SlowMoMan: A web app for discovery of important features along user-drawn trajectories in 2D embeddings[⋆]

Kiran Deol[1], Griffin M. Weber[2], and Yun William Yu[3,4]

[1] University of Alberta, Edmonton AB T6G 2R3, Canada
[2] Harvard Medical School, Boston MA 02115, USA
[3] University of Toronto Scarborough, Toronto ON M1C 1A4, Canada
[4] Carnegie Mellon University, Pittsburgh PA 15213, USA
ywyu@cmu.edu

Abstract. Nonlinear low-dimensional embeddings allow humans to visualize high-dimensional data, as is often seen in bioinformatics, where data sets may have tens of thousands of dimensions. However, relating the axes of a nonlinear embedding to the original dimensions is a nontrivial problem. In particular, humans may identify patterns or interesting subsections in the embedding, but cannot easily identify what those patterns correspond to in the original data. Thus, we present SlowMoMan (SLOW Motions on MANifolds), a web application which allows the user to draw a 1-dimensional path onto a 2-dimensional embedding. Then, by back-projecting the manifold to the original, high-dimensional space, we sort the original features such that those most discriminative along the manifold are ranked highly. We show a number of pertinent use cases for our tool, including trajectory inference, spatial transcriptomics, and automatic cell classification.
Software availability: `https://yunwilliamyu.github.io/SlowMoMan/`
Code availability: `https://github.com/yunwilliamyu/SlowMoMan`

Keywords: visualization · embedding · manifolds

[⋆] K.D. was supported by the University of Toronto Data Sciences Institute as a Summer Undergraduate Data Science (SUDS) Research Scholar. We acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC), (NSERC grant RGPIN-2022-03074)

1 Introduction

The surge of big data in bioinformatics often means data sets may contain hundreds of thousands of dimensions/features [11,16,25]. Such data is impossible to visualize without low-dimensional embeddings. A number of linear and non-linear embedding algorithms exist, such as PCA [21], UMAP [2], and t-SNE [14]. Visual analysis tools can help users discover insights in their embeddings through visual enrichment of the embedding (such as by adding color maps, labels, annotations, etc.) [19]. Such tools allow users to manually identify areas of significance that may not be flagged as important by quantitative metrics alone. However, understanding the behaviour of the original features within such areas is nontrivial when the number of dimensions is too large to be manually examined. In particular, a user may wish to know if any of the original features exhibit special or notable behaviour within the area of interest.

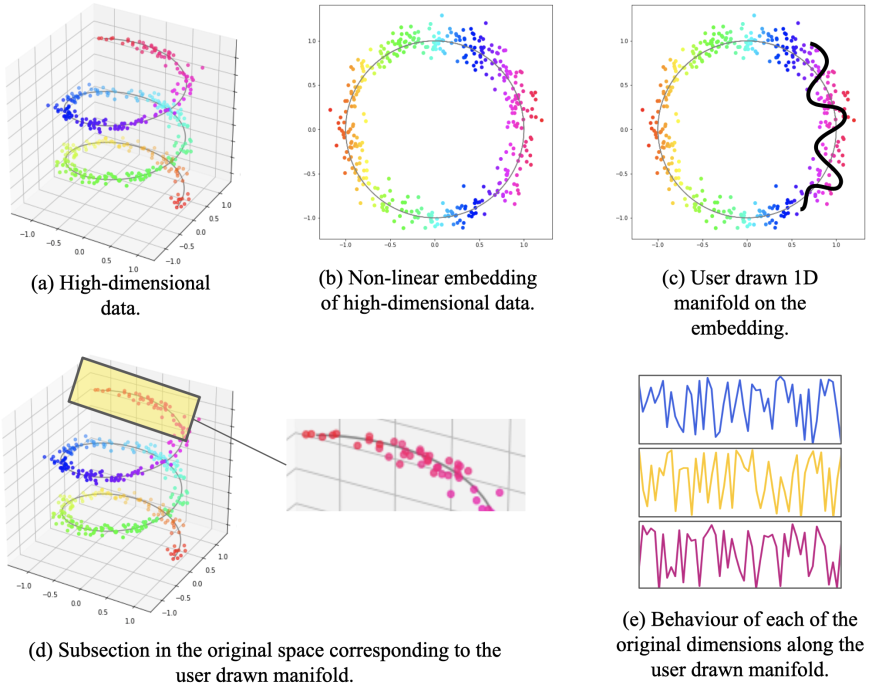
Automatically determining discriminative features on labelled data (and then using those features for downstream tasks, such prediction of labels on unlabelled data points) is a classic problem in machine learning, and has been solved in many ways using tools such as regression analysis [8], support vector machines (SVMs) [18], and deep neural networks [13]. However, in this paper we seek to identify those features which best determine/account for a particular spatial pattern in an embedding (as opposed to features which are most important in say, predicting the class of a point). For linear embedding methods like PCA, determining the most discriminative features is straightforward. For nonlinear methods, it is not clear how to determine which of the original features is most discriminative. As a proxy for discriminative ability over a particular spatial pattern, we propose a novel pipeline for ranking features by their periodicity along a user-defined path (1-dimensional manifold). In other words, we assume that features which best account for certain spatial patterns in an embedding must have some level of repeating structure along said manifold. Such a heuristic allows us to abstract away from the exact embedding algorithm used and also allow users to define completely custom paths. Additionally, our approach avoids making any decisions about which behaviours should be considered ‘discriminative’. Rather, we search for any kind of repeating structure.

To formalize the study of custom paths, we exploit the perspective of spatial analysis, i.e. studying the behaviour of features over a set of spatial parameters [27]. To illustrate, consider a two-dimensional city map with the two variables of rainfall and temperature overlaid. Analogously, we use a 2D embedding of high-dimensional data, and then seek to understand behavior of the features in the original space as a function of a some trajectory in the embedding space. The use of one-dimensional trajectories in our analysis enables us to leverage standard fast and efficient methods used in time series analysis to determine the importance of features.

Hence, in this manuscript, we seek to automatically extract features on non-categorical labels corresponding to user-specified trajectories/pseudo-time in datasets. What constitutes a notable area is dependent on the user’s domain knowledge, expectations, and purpose. For example, single cell trajectory inference seeks to identify continuous processes in single cell data, often corresponding to a notion of pseudotime in cell differentiation [12]. Often, such trajectories are near-to-obvious to the human eye when embedded in 2D, but the relationship between the 2D embedding and the original feature space may be complicated. Users can quickly identify potential trajectories in the embedded, but cannot easily link those trajectories back to feature space. Furthermore, users need to remain cognizant of spurious path-like structures in an embedding, which may appear due to the optimization criteria of various embeddings—t-SNE is well-known to in certain parameter ranges to promote long thread-like regions, even when the underlying data do not strongly support it [28].

Thus, we present SlowMoMan, an in-browser tool which allows users to draw a custom, 1-dimensional path (a.k.a. manifold) over notable areas of their embedded data. By examining the visualization of the embedding and the distribution of the original features along possible trajectories in the embedding, users can quickly draw potential trajectories which they can compare and analyze. Once the user has drawn a path over a notable area, we run a nearest-neighbours algorithm on the points of the path to locate the nearest points in the embedding and then back-project those points to their original feature space (see. Fig. 1d). Along this back-projection, we examine the behaviour of each of the original features and for each of these features, we compute a user-chosen metric for discriminative ability.

Importantly, because the user has drawn a path, rather than provided a binary label, ‘discrimination’ here corresponds to measuring how important that feature seems to be to determining the position along the path (as opposed to how important a feature is in determining the label of some sample). Users may choose from autocorrelation or the harmonic sum of the magnitudes of the fast Fourier transform, two metrics we have found to be both quick-to-compute and that correspond well with a human intuition of path-importance. Based on the metric scores, we sort the original features. Using metrics to sort the original features provides a fast filter that does not impose any assumptions on the data set (as opposed to wrapper or embedded feature selection methods, which rely on generally expensive learning algorithms to rank features' importance) [25]. The features ranked highly are then listed and their exact distributions along the manifold are displayed for the user, who can then manually validate the feature selection.



(a) High-dimensional data.

(b) Non-linear embedding of high-dimensional data.

(c) User drawn 1D manifold on the embedding.

(d) Subsection in the original space corresponding to the user drawn manifold.

(e) Behaviour of each of the original dimensions along the user drawn manifold.

Fig. 1: In most use cases, data with hundreds or thousands of dimensions would be best suited for SlowMoMan. (a) However, to visualize an example of high-dimensional data, we present a toy 3-dimensional dataset (the typical Swiss Roll dataset with 3 dimensions/features). (b) A non-linear embedding allows visualization in 2 dimensions, but it is not clear how these axes relate to the original features. (c) Users may use SlowMoMan to draw a path in the 2D embedding, which ideally corresponds to something close to a 1D manifold in the original space. (d) We can then cross-reference the points from the 2D embedding to their original representations in the high-dimensional space. (e) We analyze the behavior of the original features along that path, highlighting and sorting by the features that vary the slowest or are most autocorrelated.

2 Methodology

2.1 Overview

SlowMoMan runs locally in-browser using the Javascript engine and relies on the user's machine for all computation and storage. Thus, users do not need to install any additional software and can process their data sets without any third-party servers ever seeing their data. We have tested SlowMoMan on commodity personal laptops using Safari, Firefox, and Chrome. A 2017 Lenovo laptop with an Intel i7-8550U CPU and 16 GB of RAM is sufficiently fast to run SlowMoMan interactively with only a few seconds of loading delay for large data files, and any modern browser works. SlowMoMan will accept CSV files of the user's 2D embedding and the original data, where columns contain the features and rows contain the samples. Once uploaded, the user will see the visualization of the embedded data. On this visual, the user can draw their own 1-dimensional manifold over any areas of interest (Fig. 1c). A nearest neighbours algorithm is then used to identify the points of the embedding which are nearest to the user's custom manifold. Those points are then back-projected to their original space (Fig. 1d). Next, we observe the behaviour of each of the original dimensions along the back-projection of the user's manifold (Fig. 1e).

Using the behaviour of each of the original dimensions along the back-projection of the user's manifold, SlowMoMan will apply a number of possible metrics to sort the original dimensions, such that the dimensions most discriminative along the user's manifold are ranked highest for further examination. Users will see a graph of the value of each of those recommended discriminative variables plotted against the time parameterization of the drawn trajectory. Users can also select specific variables to plot, if they have expert knowledge on which of the highest ranking variables are likely to be important.

2.2 Canvas and back-projection

In our implementation, we used the HTML Canvas API to create a fixed canvas of size 512x512 pixels, and the 2D embedding is resized to fit into that space. This design choice was made for a combination of efficiency and reproducibility: (1) by setting the visual display to match the 2D embedding, determining the appropriate back-projection is made easier, and (2) we provide the user-drawn path as a series of key points in the canvas in a text box, and fixing the canvas size for all users ensures that those key points remain correct and portable.

However long the user-drawn path, we internally convert the path into 512 evenly spaced points on the canvas approximating that path. This is necessary because the mouse-over events vary across browsers, OS mouse sensitivity, and the mice themselves. If a user slowly drags a path, there will be many more mouse-over events than if a user draws the same path quickly; hence the normalization. The choice of $n = 512$ was largely arbitrary, but chosen to match the canvas size, and as a power of 2 for efficiency of the FFT computation.

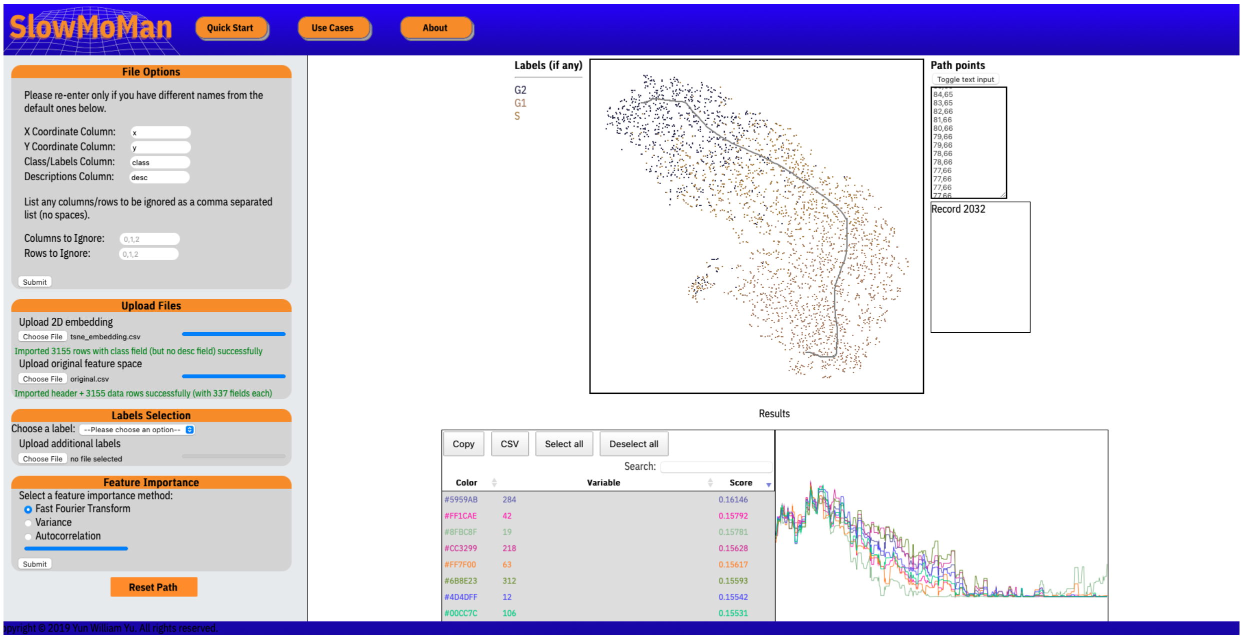
Because SlowMoMan internally discretizes both the path and the canvas, back-projection via nearest neighbors is fairly straight-forward. For each point along the path, we find the nearest data point on the canvas, and store the original features for that data point. Because of the discretization, there may sometimes be multiple points in the embedding with equal distance to a point in the path. In such cases we simply average features across all of those points. One might argue that it would be better to arbitrarily choose a point instead of averaging features, but in practice, it does not seem to make a difference as this is a sufficiently rare event.

2.3 Feature-ranking metrics

Depending on the data or the goals of the user, which features are most discriminative may vary. We explored a variety of possible metrics for users to choose from, though most performed fairly poorly (e.g. variance)—while variance is an accessible metric, its inability to account for the spatial/time-dependency along a path yields inconsistent results. The metrics that were most helpful were those that find periodicity along the path. As such, the metrics available in SlowMoMan include autocorrelation and the harmonic sum of the fast Fourier transform magnitudes (*FFT Score*).

Implementation details Although SlowMoMan is presented as a website, there are no server-side components. All of SlowMoMan is implemented in client-side HTML5, Javascript, and CSS. The choice to use web technologies was made to take advantage of ready-to-use user interface elements, including the HTML5 Canvas, and Mouseover events for drawing. For the FFT computation, we use a 2014 version of the Project Nayuki Javascript FFT library [17], adapted by Cannam [3]. The autocorrelation computation does not use any libraries and is done directly in-app.

Autocorrelation The autocorrelation of a variable x can intuitively be thought of as the level of correlation that x has with a delayed version of itself [10]. Within the domain of spatial analytics, the spatial autocorrelation is used to formalize the notion that areas near each other are more likely to be correlated. Thus, SlowMoMan uses the autocorrelation metric to test whether proximal points in the embedding are correlated within the original feature space. In other words, SlowMoMan uses autocorrelation to pick which



Please re-enter only if you have different names from the default ones below.

X Coordinate Column: x
Y Coordinate Column: y
Class/Labels Column: class
Descriptions Column: desc

List any columns/rows to be ignored as a comma separated list (no spaces).

Columns to Ignore: 0,1,2
Rows to Ignore: 0,1,2

Upload 2D embedding
Choose File tsne_embedding.csv
Imported 3155 rows with class field (but no desc field) successfully
Upload original feature space
Choose File original.csv
Imported header + 3155 data rows successfully (with 337 fields each)

Choose a label: --Please choose an option--
Upload additional labels
Choose File no file selected

Select a feature importance method:
● Fast Fourier Transform
○ Variance
○ Autocorrelation

84,65
83,65
82,66
81,66
80,66
79,66
79,66
78,66
78,66
77,66
77,66
77,66
77,66

Record 2032

Search:

Color	Variable	Score
#5959AB	284	0.16146
#FF1CAE	42	0.15792
#8FBC8F	19	0.15781
#CC3299	218	0.15628
#FF7F00	63	0.15617
#6B8E23	312	0.15593
#4D4DFF	12	0.15542
#00CC7C	106	0.15531

(a) In-app screen capture of SlowMoMan

Fig. 2: SlowMoMan identifies the most informative nodes within a neural network trained to process images of Jurkat cells in different stages of the cell cycle [7]. The t-SNE embedding of the neural network activation space representations of each cell is presented near the center of the screen capture. Notice the clusters already have a fairly clear trajectory and are indeed in their correct order. The top 7 most important nodes identified by SlowMoMan are displayed in the left half of the lower two boxes. The behaviour of the top 7 nodes along the path of the trajectory is displayed in the right half of the lower two boxes. Notably, all of the nodes move in a fairly similar pattern, as to be expected with a good trajectory.

features repeat themselves along the user-drawn path. Such periodicity may suggest that the feature is one of the determinants of the path. Formally, the autocorrelation is defined as

$$r_k = \frac{\sum_{t=k+1}^{n}(x_t - \bar{x})(x_{t-k} - \bar{x})}{\sum_{t=1}^{n}(x_t - \bar{x})^2} \qquad (1)$$

where $\bar{x}$ is the mean of x and k is the lag value. Thus, SlowMoMan computes r_k for each of the original dimensions along the back-projection of the embedding, using a lag value of $k = 22$.

However, one disadvantage of using autocorrelation is that we need to specify a lag value in advance. The choice of lag value $k = 22$ was found empirically to work reasonably well in our experimentation (close to the square root of the 512 path points), so that is the default lag, but that varies by application. Due to the need to set a parameter, autocorrelation should only be used by advanced users with a reasonable guess as to the lag that should be set.

Fast Fourier Transform That of course leads us to desire a parameter-less way to find features that have some periodicity. Thanks to the decision to rely on a single parameter in our spatial analysis, we can leverage well-known tools used in time series analysis. To this end, we turn to standard signal processing tools and the Fourier transform. The fast Fourier transform is an efficient algorithm for computing the discrete Fourier transform [5]. The discrete Fourier transform decomposes a sequence of values into a sum of discrete sinusoidal waves, each with a distinct magnitude and frequency. Formally, if we have a sequence of k values $\{x_n\}$, we can approximate it with another sequence $\{F_m\}$ of sinusoids.

$$F_m = \sum_{n=0}^{k-1} x_n * [cos(\frac{2\pi}{k} * n * m) - i * sin(\frac{2\pi}{k} * n * m)] \qquad (2)$$

Each of these new sinusoids will have a respective amplitude, which is the absolute height of the sinusoid. Across all of the amplitudes m_n of each of the sinusoids for a specific sequence $\{x_n\}$, we can take a harmonic sum S over descending m_n

$$S = \sum_{n=1}^{k-1} m_n * \frac{1}{n} \tag{3}$$

which is then used as the custom "FFT Score" metric. Notice the metric is maximized when a feature can be decomposed into sinusoids with large amplitudes - as opposed to features with many small amplitude sinusoids, which may be indicative of noise. In other words, the harmonic sum inherently lends greater weight to the sinusoids with lower frequencies—although, we naturally omit the constant m_0 term as we are only interested in periodic behavior. Intuitively, we are not as interested in high-frequency variables because they are not strongly correlated with position on our path. Instead, low-frequency variables (i.e. slow motions) are features where if you know their values, you can roughly predict your position along the path. This allows us to simultaneously capture periodicity at all scales, while more heavily weighting longer periods, overcoming the shortcoming of having to set a lag parameter in autocorrelation.

3 Evaluation

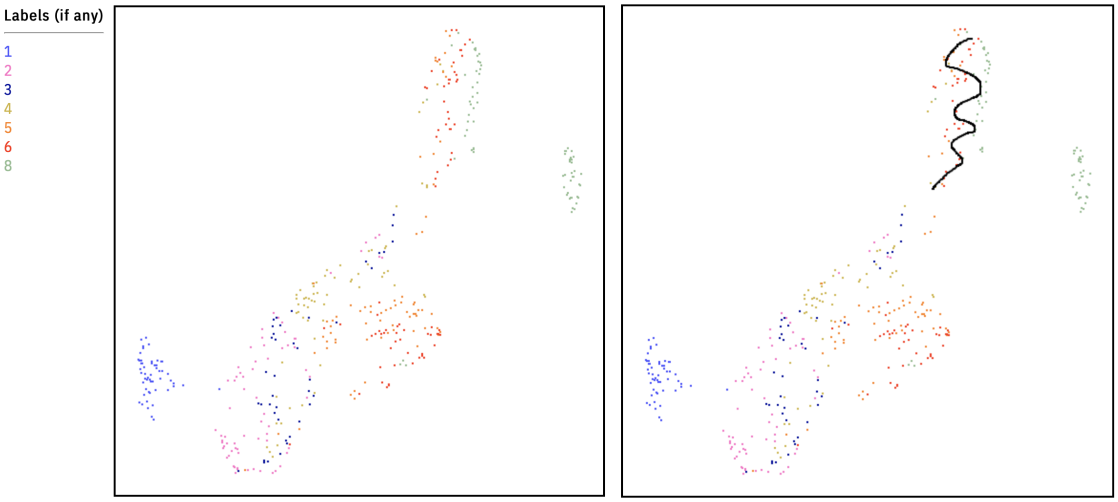
We demonstrate the utility of SlowMoMan through case studies and empirical analysis. For all use cases presented below, we have made available associated data and trajectory information on the SlowMoMan website (see "Use Cases" and from there "Google Drive artifacts"). In the first task, our prototypical use case, we show how SlowMoMan can be used to help interpret trajectories drawn on a 2D embedding. In this task, we consider two trajectories, where the features are actual genes in single-cell transcriptomic data. In the second task, we demonstrate using SlowMoMan on a more classic task related not to trajectories, but to simple clusters instead—we find important features disambiguating between two clusters of spatial transcriptomic cell data. In the last task, we use SlowMoMan to better characterize the nature of errors in automatic cell classification using single-cell transcriptomic data.

3.1 Trajectory Inference/Pseudotemporal Ordering

We apply SlowMoMan to datasets taken from Paul et al. [20] (Fig. 4) and Eulenberg et al. [7] (Fig. 2). Physical limitations in the technologies used to obtain gene expression data often make it impossible to track dynamic processes in single cells, such as the cell cycle or cell differentiation. Trajectory inference methods seek to infer such processes by analyzing similarities in the gene expression of cells to order them along a continuous interval. A large number of algorithms exist for such a purpose and Saelens et al. [22] benchmark 45 of these methods on their accuracy, scalabilty, stability, and usability. SlowMoMan inherently complements such methods due to its focus on trajectories, as opposed to standard tools which often focus on analyzing entire clusters. In particular, SlowMoMan may be used to identify important features along a given trajectory or to quickly explore alternative, user-defined trajectories.

Trajectory Inference in Hepatoblast Differentiation of Mouse Liver Cells Data collected by Yang et al. [32] was retrieved from a Kaggle dataset [4]. The raw data contains 24,748 gene counts across 447 liver cells obtained from mice. Each cell is labelled by its embryonic date, starting from embryonic day 10.5 (E10.5) to embryonic day 17.5 (E17.5), with E16.5 excluded. For our analysis, we changed the labelling of embryonic days to a simple scalar where E10.5 became 1, E11.5 became 2, and so fourth (although E16.5 is not present in the dataset, E17.5 is still labelled 8). Although the authors used PCA to obtain their 2D embedding, we chose to use UMAP. The linearity of PCA makes it a feature importance method in its own right [23] (thus making its usage with SlowMoMan somewhat redundant). Nonlinear methods like UMAP, which do not have clear means of identifying the contribution of each feature, are better suited to SlowMoMan. As a simple sanity check, we demonstrate a run of SlowMoMan with the FFT metric.

For the first run, we present an irregular scrawl drawn over an interesting cluster of the UMAP embedding (see Fig.3b). This cluster is off to the top right of the "main" cluster seen below it. This cluster is particularly interesting in that it only contains days 5, 6, and 8 (recall day 7 is not in the dataset) and each of those days



(a) Original UMAP Projection of Liver Cells

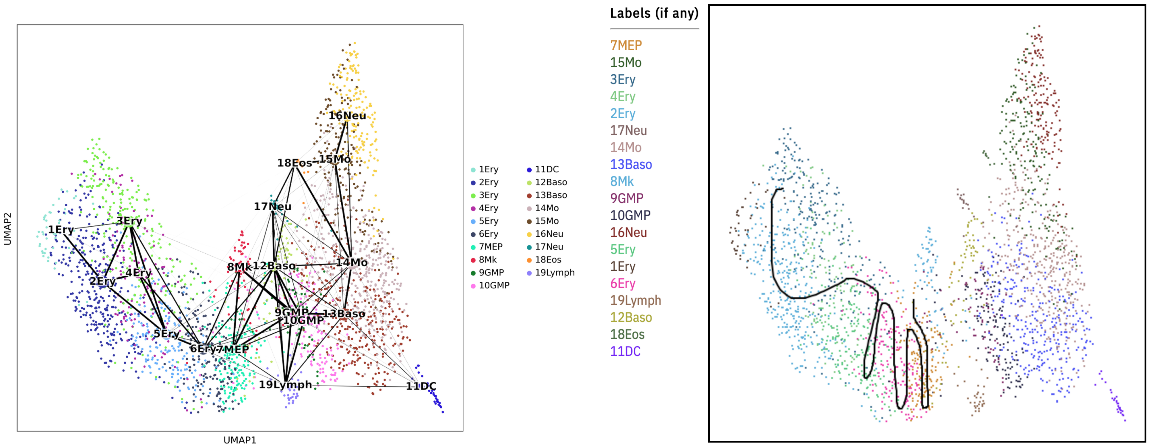
(b) UMAP Projection with Custom Trajectory used for FFT Analysis

Fig. 3: In (a) we present the 2D embedding displayed by SlowMoMan, where the labels on the right correspond to embryonic date. In (b) we show the 2D with the user-drawn trajectory. Notice the trajectory is drawn over a fairly isolated cluster of cells containing only days 4, 5, 6, and 8.

present themselves in roughly vertical lines in roughly the correct order (days 4 and 5 are intermixing, but days 6 and 8 are quite distinct). The choice to use a "squiggle" shape comes from the fact that since each day presents roughly vertically, the overall progression from day 4 to 5, and 5 to 6, etc. is therefore horizontal. However, since this cluster is not very wide, a single horizontal line would be not be representative of the entire cluster. Thus, the squiggle manages to create multiple horizontal lines on the cluster. After drawing this, we compute the FFT metric. Recall that there are nearly 25,000 genes and we rely on the FFT metric to return a list of the top 100 genes who present periodic behaviour along our manifold. The analysis of the genes marked significant by the FFT metric is of course limited to the amount of domain knowledge and time for exploration that one has available.

We now discuss a few notable genes flagged by the FFT metric. Ranked 2nd was FABP1 (fatty acid binding protein 1), which is known to have expression in the liver as it encodes the for fatty acid binding protein of the liver [24]. Ranked 25th was SERPIND1, which is known to have highly biased expression in the liver [24]. Ranked 27th was CYP2D26 which the original authors identified as notable among the E15.5 and E17.5 hepatocytes (see fig 2 of [32]). This aligns well with the fact that the cluster being analyzed here contained days 6 and 8 most prominently. Ranked 78th was CPS1, which like SERPIND1 is known to have biased expression in the liver and was also identified by the original authors as being highly expressed among E15.5 and E17.5 hepatocytes (see fig 2 of [32]). Ranked 81st was KLF6 (Kruppel-Like Factor 6), which is known to have some expression in liver tissue [24] and was also found to have a potential association with advanced nonalcoholic liver disease in humans [15]. Although this evaluation is inherently limited by our domain knowledge and exploration, the ability to flag such genes as significant (out of 25,000) is remarkable.

Trajectory Inference in Hematopoiesis of Mouse Bone Marrow Cells Data collected by Paul et al. [20] was retrieved through the *paul15()* method of [29]. The raw data contains 3,451 gene counts across 2,730 labelled bone marrow cells of mice. The clusters correspond to cell type. Numbers are prefixed to the cluster labels to distinguish between cells that may belong to the same cell type, but are at different levels of hematopoietic differentiation. In order to identify potential trajectories we use PAGA [30], one of the best overall trajectory inference methods in [22]. From a k-nearest neighbours (kNN) graph of single-cells and the associated clusters (partitions) of these cells, PAGA will compute a new graph representing the connectivities of the partitions. Obtaining the data, preprocessing, and running PAGA on this dataset were



(a) PAGA Graph Overlaid onto UMAP Embedding. (b) Custom Trajectory as used in SlowMoMan.

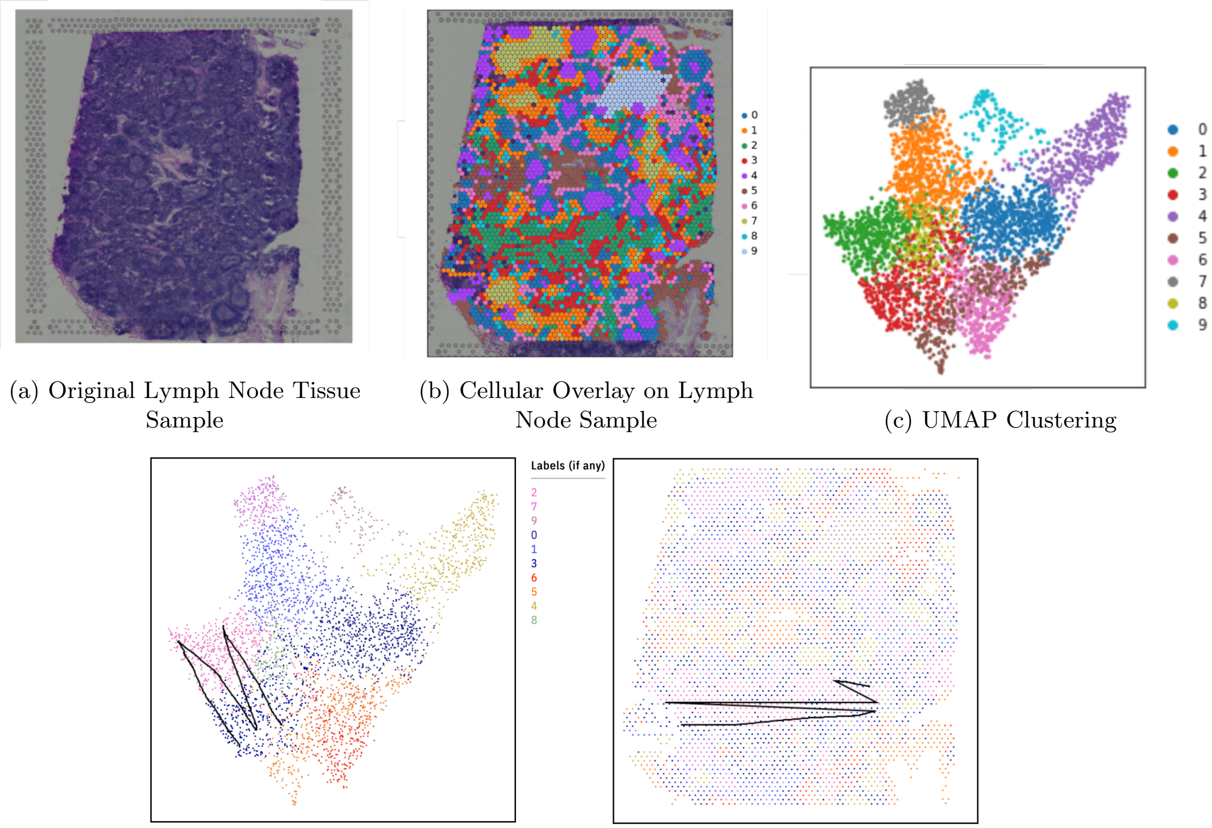
Fig. 4: The results of the PAGA algorithm on the hematopoiesis data of Paul et al, 2015 [20] highlight strong connectivities between clusters 1-6. We use SlowMoMan to explore important features along a custom manifold between clusters 1-7. (a) The UMAP embedding of the hematopoiesis data of Paul et al, 2015 [20] with the PAGA connectivities between clusters represented as black edges. Thicker edges indicate stronger connectivities. Clusters correspond to cell type, with labels corresponding to those of Paul et al, 2015 [20]. Note also the prominent connectivities between clusters 1-7. (b) A screen capture of SlowMoMan with the user-drawn manifold used in the analysis.

all accomplished through the Python package Scanpy, which contains methods for trajectory inference [29]. In particular, we use UMAP with the euclidean distance metric to compute a single-cell connectivity graph with a neighbourhood size of 20. We use the existing clusters of Paul et al. [20] as the partitions. We then run PAGA to obtain the graph of connectivities between partitions, as shown in Fig. 4.

A notable set of connectivities are those among clusters 1 to 7, which Paul et al. [20] hypothesize represent a continuum of erythrocyte differentiation. SlowMoMan is used to draw a trajectory between these clusters (see Fig.4) and using the FFT score to sort, obtain a list of the 100 most important features out of 3451. Notably, a number of key transcription factors identified in Fig. 1C of Paul et al. [20] are ranked within the top 20 genes of this trajectory by SlowMoMan, includinig *Apoe*, *CPOX*, *Hba-a2*, and *Car2*. Within the top 100 genes, SlowMoMan also identified *Gata2* and *Car1*. Although not previously associated with erythropoiesis, Paul et al. [20] noted the significance of *Cited4* and *Phf10* in clusters 1-7. Interestingly, SlowMoMan also ranks these genes within the top 50 most important genes of the trajectory. Although SlowMoMan was able to identify a number of key genes, due to these genes being spread across the list of top 100 features, a user without prior knowledge may not realize their significance. Indeed, users with prior knowledge as which features are most important may focus on using SlowMoMan to experiment with possible trajectories that capture as many of the important features as possible, as illustrated in the current example. In contrast, users without prior expectations on the features may instead use SlowMoMan is to gather a list of important features along a specific trajectory which may be investigated further with more conclusive methods.

3.2 Spatial Transcriptomics

We apply SlowMoMan to a Human Lymph Node Spatial Gene Expression Dataset by Space Ranger 1.0.0, as provided by 10x Genomics [9] (Fig. 5). A recent advancement in bioinformatics is the ability to associate gene expression data with its spatial distribution in a sample. Specifically, spatial transcriptomics enables researchers to locate the position of individuals cells (and the respective gene expression data of each cell) in a sample. Spatial transcriptomics is a strong candidate field for SlowMoMan, as clusters in a tissue sample are unlikely to be perfectly distinct. Indeed, while the clusters of the corresponding embedding of the cells of tissue sample cells tend to remain fairly coherent, the clusters in the tissue sample itself are often intermingled



(a) Original Lymph Node Tissue Sample

(b) Cellular Overlay on Lymph Node Sample

(c) UMAP Clustering

(d) UMAP clustering with user-drawn trajectory as used in SlowMoMan

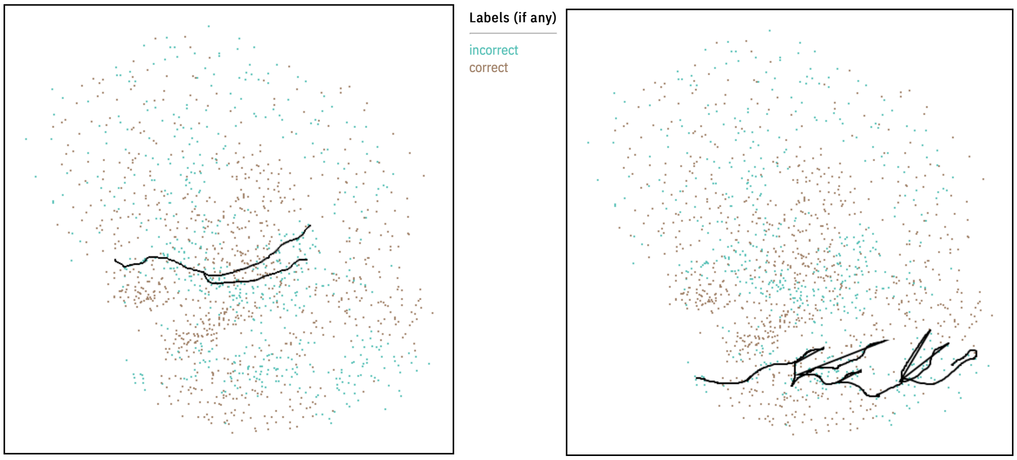
(e) Cellular Overlay on Lymph Node Sample with user-drawn trajectory as used in SlowMoMan

Fig. 5: SlowMoMan applied to a spatial transcriptomics dataset to determine important features distinguishing between two nearby clusters. (a) The original image of the human lymph node tissue sample [9]. (b) The human lymph node tissue sample with the cells and their respective UMAP classes overlaid. (c) The UMAP clustering of the single cells identified in the tissue sample. (d) The UMAP clustering of the single cells identified in the tissue sample, with a simplified overlay of the lines used during the analysis with SlowMoMan drawn as black lines. (e) The original image of the human lymph node tissue sample, with a simplified overlay of the lines used during the analysis with SlowMoMan drawn as white lines.

are broken into a number of subclusters, as illustrated in 5. In such cases, analyzing the features of entire clusters would not be as informative as analyzing the individual subsections of clusters and comparing them.

Thus, we apply SlowMoMan to a Human Lymph Node Spatial Gene Expression Dataset, provided by 10x Genomics [9] (Fig. 5). In particular, 10x Genomics used their Visium Spatial Gene Expression tool to analyze a sample of human lymph node tissue. Obtaining the data, preprocessing, visualization, and running the embedding algorithm were all accomplished through the Python package Scanpy, which also contains methods for spatial transcriptomics [29].

This dataset from 10x Genomics contains the spatial coordinates of each of the individually identified cells in the tissue sample, along with the gene expression levels of each of these cells. From the gene expression data, we use UMAP on a 50-dimensional PCA embedding to compute a graph with neighbourhoods of size 15. This neighbourhood graph is then used in the Leiden clustering algorithm of [26] to identify 10 within the data. Although these clusters are distinct (see 5), we cannot yet make any claims as to what known lymph cell subtypes these clusters may translate to. However, we can identify patterns between the spatial distribution of the cells and their UMAP distribution and use SlowMoMan to explain the genetic underpinnings for such patterns.



(a) kNN errors on pancreatic data with path covering large erroneous region

(b) kNN errors on pancreatic data with user-drawn path covering other large erroneous region

Fig. 6: SlowMoMan is used to understand the types of errors made by the kNN algorithm (with a neighbour size of 9) on an automatic cell classification task in single-cell transcriptomics [31]. We use UMAP to obtain a 2-dimensional embedding. The clusters represent which cells were classified correctly vs. incorrectly by the trained kNN model. Of the 581 incorrectly labelled cells, 78% were beta cells, 14% were gamma cells, and 8% were delta cells. None of the incorrectly identified cells were alpha cells. (a) A screen capture of SlowMoMan with the user-drawn trajectory over a fairly distinct subcluster of errors. Notable genes identified along this trajectory were NPY, PPY, and ZFP36. (b) Similar to (a), but with a user-drawn trajectory covering a different, but also fairly distinct subcluster of errors. Notable genes identified along this trajectory were G6PC2 and SCGB2A1.

One notable observation is that clusters 2 and 3 (green and red) not only exist beside each other in the UMAP embedding, but are also frequently seen together in the tissue sample, with red appearing almost as a border to the green sections. In contrast to the embedding, the red cluster is broken up into many separate, smaller subclusters throughout the tissue sample. Computing the feature importance of the red cluster in its entirety would fail to capture any behaviour unique to aforementioned area of the red cluster which borders the green cluster. Thus, we first use SlowMoMan without considering any subclusters and analyze the red and green clusters in the UMAP embedding. Then, we compare the results to SlowMoMan being run on the aforementioned subcluster of the spatial distribution. Notice in Fig. 5 that the path drawn goes back-and-forth between the two clusters, unlike in the previous example; this artifically creates the periodicity that SlowMoMan tries to detect, allowing it to find not just discriminating features for paths, but also clusters.

Between the two runs of SlowMoMan shown in Fig. 5, we used the FFT score to obtain the top ten most important genes in both instances and found that genes IGHG1, IGHG2, IGHG3, IGHG4, and IGKC were all identified among the top 10 most important genes (out of 19,686) in both the trajectory drawn on the UMAP embedding and the trajectory drawn on the spatial distribution. Notably, the Entrez Gene summary for each of these genes predicts that they are important to immunoglobulin receptor binding. Immunoglobulins are produced by B-lymphocytes, which are located primarily in the cortex of the lymph nodes. Given that the red and green clusters were observed mainly in one region of the tissue sample, we could the explain the importance of IGHG1, IGHG3, IGHG4, and IGKC as the result of the cells of the red and green clusters being located in the cortex of the lymph node tissue sample.

3.3 Automatic Cell Classification

We apply SlowMoMan to a single-cell transcriptomic dataset from [31] (Fig. 6). In single cell analysis, data sets are often too large for manual annotation and methods for automated classification of cell types are a promising alternative. A number of challenges arise when using such methods, however, such as choosing which method to use or evaluating the accuracy of a given method. SlowMoMan augments such methods by allowing users to interpret regions where multiple classification methods may differ, or by interpreting regions where a classification method fails.

For example, the pancreatic data obtained by [31] examines the transcriptomes of 1492 islet cells across 18 donors and includes four classes of islet cells: alpha, beta, gamma, and delta. As part of a survey of automatic cell classification methods, [1] apply a number of classification methods to this dataset, including the kNN algorithm ($k = 9$). Replication of these methods found that kNN had an accuracy score of 59.10%, supporting the results of [1]. SlowMoMan augments this method by enabling the interpretation of regions where kNN fails.

One explanation for the poor performance of the kNN algorithm was the imbalance in data: 60% were alpha cells, 32% were beta cells, 5% were gamma cells, and 3% were delta cells. This would also explain why all of the incorrectly identified cells were either beta, gamma, or delta cells. However, we also use SlowMoMan to understand the possible genomic causes for these errors.

In particular, we run SlowMoMan on two large and fairly distinct subclusters of kNN errors (see 6). Akin to the previous example, analyzing the feature importances across the entire cluster of errors would fail to capture any information unique to the two identified subclusters. Thus, we perform two runs of SlowMoMan, one for each of the previously identified subclusters of errors.

In the first subcluster (Fig. 6a), we identified 3 genes of interest: NPY, PPY, and ZFP36. In the second subsection, (Fig. 6b), we identified 2 genes of interest: G6PC2 and SCGB2A1. We note that in both subclusters, the aforementioned genes were all within the top 100 most important genes (out of 33890) of their respective trajectories. We also note that in this case, sorting with the autocorrelation metric led to more notable genes being identified than with the usual FFT score. The PPY gene is a known marker gene of gamma cells. Thus, its importance is to be expected, given the inability of the kNN model to recognize any of the gamma cells. The importance of the remaining four genes, however, can be explained by the discovery of four beta cell subtypes by [6]. Indeed, they identify the four subtypes as β_1, β_2, β_3, β_4. Within which there is the $ST8SIA1^+$ β_3/β_4 subtype, $ST8SIA1^-$ β_1/β_2 subtype, $CD9^+$ β_2/β_4 subtype, and the $CD9^-$ β_1/β_3 subtype. Notably, NPY and ZFP36 were found by [6] to be differentially expressed between the $CD9^-$ β_1/β_3 and $CD9^+$ β_2/β_4 subtypes. In particular, NPY had larger expression levels in the $CD9^-$ β_1/β_3 subtype, whereas ZFP36 had larger expression levels in the $CD9^+$ β_2/β_4 subtype. Thus, although the cells in the first subsection of errors (Fig. 6a), were primarily beta cells, the possibility of multiple subtypes of beta cells existing in that area may have contributed to the poor results obtained by the kNN model when classifying beta cells. Similarly, in the second subsection analyzed (Fig. 6b), the genes G6PC2 and SCGB2A1 were also identified by [6] as being differentially expressed between the $ST8SIA1^+$ β_3/β_4 and $ST8SIA1^-$ β_1/β_2 subtypes. Therefore, even with an imbalance in the data set, the analysis with SlowMoMan suggests the possibility that the unspecified subtypes within the beta cells may have contributed to the poor performance of the kNN model on classifying beta cells.

4 Conclusion

In this manuscript, we presented a new interactive web app, SlowMoMan, for visualization and exploration of important variables in low-dimensional embeddings. This web app runs entirely client-side using Javascript, HTML 5, and CSS, and allows for real-time usage on commodity laptops. By back-projecting user-drawn paths and ranking variables by proxies for their periodicity, users are able to identify the original features that are most informative for those paths.

The major advance of our work is in allowing users to draw paths, rather than simply apply categorical labels, which we showed in our two trajectory analysis examples. As shown in our latter two examples, users are still able to examine features important for binary classifications by simply drawing appropriate paths doubling back on themselves. However, the paths lend themselves particularly well in cases where there is some underlying notion of paths and time, such as in cell differentiation and cell cycles. We hope tools like SlowMoMan will assist scientists in interpreting and analyzing low-dimensional embeddings.

References

1. Abdelaal, T., Michielsen, L., Cats, D., Hoogduin, D., Mei, H., Reinders, M.J.T., Mahfouz, A.: A comparison of automatic cell identification methods for single-cell rna sequencing data. Genome Biology **20**(1), 194 (Sep 2019). https://doi.org/10.1186/s13059-019-1795-z, `https://doi.org/10.1186/s13059-019-1795-z`
2. Becht, E., McInnes, L., Healy, J., Dutertre, C.A., Kwok, I.W., Ng, L.G., Ginhoux, F., Newell, E.W.: Dimensionality reduction for visualizing single-cell data using umap. Nature biotechnology **37**(1), 38–44 (2019)
3. Cannam, C.: Javascript dsp tests. `https://code.soundsoftware.ac.uk/projects/js-dsp-test/repository/show/fft`, accessed: 2019-08-01
4. Chervov, A.: scrna-seq trajectory inference (2022), `https://www.kaggle.com/datasets/alexandervc/trajectory-inference-single-cell-rna-seq`
5. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex fourier series. Mathematics of computation **19**(90), 297–301 (1965)
6. Dorrell, C., Schug, J., Canaday, P.S., Russ, H.A., Tarlow, B.D., Grompe, M.T., Horton, T., Hebrok, M., Streeter, P.R., Kaestner, K.H., Grompe, M.: Human islets contain four distinct subtypes of β cells. Nature Communications **7**(1), 11756 (Jul 2016). https://doi.org/10.1038/ncomms11756, `https://doi.org/10.1038/ncomms11756`
7. Eulenberg, P., Köhler, N., Blasi, T., Filby, A., Carpenter, A.E., Rees, P., Theis, F.J., Wolf, F.A.: Reconstructing cell cycle and disease progression using deep learning. Nature Communications **8**(1), 463 (Sep 2017)
8. Freund, R.J., Wilson, W.J., Sa, P.: Regression analysis. Elsevier (2006)
9. 10x Genomics: Human lymph node spatial gene expression dataset by space ranger 1.0.0. `https://support.10xgenomics.com/spatial-gene-expression/datasets/1.0.0/V1_Human_Lymph_Node?`, accessed: 2022-08-01
10. Gubner, J.A.: Probability and random processes for electrical and computer engineers. Cambridge University Press (2006)
11. Hilario, M., Kalousis, A.: Approaches to dimensionality reduction in proteomic biomarker studies. Briefings in bioinformatics **9**(2), 102–118 (2008)
12. Ji, Z., Ji, H.: Tscan: Pseudo-time reconstruction and evaluation in single-cell rna-seq analysis. Nucleic acids research **44**(13), e117–e117 (2016)
13. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
14. Van der Maaten, L., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(11) (2008)
15. Miele, L., Beale, G., Patman, G., Nobili, V., Leathart, J., Grieco, A., Abate, M., Friedman, S.L., Narla, G., Bugianesi, E., Day, C.P., Reeves, H.L.: The kruppel-like factor 6 genotype is associated with fibrosis in nonalcoholic fatty liver disease. Gastroenterology **135**(1), 282–291.e1 (Jul 2008). https://doi.org/10.1053/j.gastro.2008.04.004, `https://doi.org/10.1053/j.gastro.2008.04.004`
16. Moore, J.H., Asselbergs, F.W., Williams, S.M.: Bioinformatics challenges for genome-wide association studies. Bioinformatics **26**(4), 445–455 (2010)
17. Nayuki, P.: Free small fft in multiple languages. `https://www.nayuki.io/page/free-small-fft-in-multiple-languages`, accessed: 2022-08-01
18. Noble, W.S.: What is a support vector machine? Nature biotechnology **24**(12), 1565–1567 (2006)
19. Nonato, L.G., Aupetit, M.: Multidimensional projection for visual analytics: Linking techniques with distortions, tasks, and layout enrichment. IEEE Transactions on Visualization and Computer Graphics **25**(8), 2650–2673 (2019). https://doi.org/10.1109/TVCG.2018.2846735
20. Paul, F., Arkin, Y., Giladi, A., Jaitin, D.A., Kenigsberg, E., Keren-Shaul, H., Winter, D., Lara-Astiaso, D., Gury, M., Weiner, A., David, E., Cohen, N., Lauridsen, F.K.B., Haas, S., Schlitzer, A., Mildner, A., Ginhoux, F., Jung, S., Trumpp, A., Porse, B.T., Tanay, A., Amit, I.: Transcriptional heterogeneity and lineage commitment in myeloid progenitors. Cell **163**(7), 1663–1677 (2015). https://doi.org/https://doi.org/10.1016/j.cell.2015.11.013, `https://www.sciencedirect.com/science/article/pii/S0092867415014932`
21. Ringnér, M.: What is principal component analysis? Nature biotechnology **26**(3), 303–304 (2008)
22. Saelens, W., Cannoodt, R., Todorov, H., Saeys, Y.: A comparison of single-cell trajectory inference methods. Nature Biotechnology **37**(5), 547–554 (May 2019). https://doi.org/10.1038/s41587-019-0071-9, `https://doi.org/10.1038/s41587-019-0071-9`
23. Song, F., Guo, Z., Mei, D.: Feature selection using principal component analysis. In: 2010 International Conference on System Science, Engineering Design and Manufacturing Informatization. vol. 1, pp. 27–30 (2010). https://doi.org/10.1109/ICSEM.2010.14
24. Stelzer, G., Rosen, N., Plaschkes, I., Zimmerman, S., Twik, M., Fishilevich, S., Stein, T.I., Nudel, R., Lieder, I., Mazor, Y., Kaplan, S., Dahary, D., Warshawsky, D., Guan-Golan, Y., Kohn, A., Rappaport, N., Safran, M., Lancet, D.: The genecards suite: From gene data mining to disease genome sequence analyses. Current Protocols in Bioinformatics **54**(1), 1.30.1–1.30.33 (2016). https://doi.org/https://doi.org/10.1002/cpbi.5, `https://currentprotocols.onlinelibrary.wiley.com/doi/abs/10.1002/cpbi.5`
25. Townes, F.W., Hicks, S.C., Aryee, M.J., Irizarry, R.A.: Feature selection and dimension reduction for single-cell rna-seq based on a multinomial model. Genome biology **20**(1), 1–16 (2019)

26. Traag, V.A., Waltman, L., van Eck, N.J.: From louvain to leiden: guaranteeing well-connected communities. Scientific Reports **9**(1), 5233 (Mar 2019)
27. Unwin, D.J., Hepple, L.W.: The statistical analysis of spatial series. Journal of the Royal Statistical Society. Series D (The Statistician) **23**(3/4), 211–227 (1974), http://www.jstor.org/stable/2987581
28. Wattenberg, M., Viégas, F., Johnson, I.: How to use t-sne effectively. Distill **1**(10), e2 (2016)
29. Wolf, F.A., Angerer, P., Theis, F.J.: Scanpy: large-scale single-cell gene expression data analysis. Genome Biology **19**(1), 15 (Feb 2018). https://doi.org/10.1186/s13059-017-1382-0, https://doi.org/10.1186/s13059-017-1382-0
30. Wolf, F.A., Hamey, F.K., Plass, M., Solana, J., Dahlin, J.S., Göttgens, B., Rajewsky, N., Simon, L., Theis, F.J.: Paga: graph abstraction reconciles clustering with trajectory inference through a topology preserving map of single cells. Genome Biology **20**(1), 59 (Mar 2019). https://doi.org/10.1186/s13059-019-1663-x, https://doi.org/10.1186/s13059-019-1663-x
31. Xin, Y., Kim, J., Okamoto, H., Ni, M., Wei, Y., Adler, C., Murphy, A.J., Yancopoulos, G.D., Lin, C., Gromada, J.: Rna sequencing of single human islet cells reveals type 2 diabetes genes. Cell Metabolism **24**(4), 608–615 (2016). https://doi.org/https://doi.org/10.1016/j.cmet.2016.08.018, https://www.sciencedirect.com/science/article/pii/S155041311630434X
32. Yang, L., Wang, W.H., Qiu, W.L., Guo, Z., Bi, E., Xu, C.R.: A single-cell transcriptomic analysis reveals precise pathways and regulatory mechanisms underlying hepatoblast differentiation. Hepatology **66**(5), 1387–1401 (2017). https://doi.org/https://doi.org/10.1002/hep.29353, https://aasldpubs.onlinelibrary.wiley.com/doi/abs/10.1002/hep.29353