Calibrating dimension reduction hyperparameters in the presence of noise

Justin Lin[1] and Julia Fukuyama[2]

[1]Department of Mathematics, Indiana University, Bloomington, Indiana, United States of America

[2]Department of Statistics, Indiana University, Bloomington, Indiana, United States of America

Abstract

The goal of dimension reduction tools is to construct a low-dimensional representation of high-dimensional data. These tools are employed for a variety of reasons such as noise reduction, visualization, and to lower computational costs. However, there is a fundamental issue that is discussed in other modeling problems that is often overlooked in dimension reduction — overfitting. In the context of other modeling problems, techniques such as feature-selection, cross-validation, and regularization are employed to combat overfitting, but rarely are such precautions taken when applying dimension reduction. Prior applications of the two most popular non-linear dimension reduction methods, t-SNE and UMAP, fail to acknowledge data as a combination of signal and noise when assessing performance. These methods are typically calibrated to capture the entirety of the data, not just the signal. In this paper, we demonstrate the importance of acknowledging noise when calibrating hyperparameters and present a framework that enables users to do so. We use this framework to explore the role hyperparameter calibration plays in overfitting the data when applying t-SNE and UMAP. More specifically, we show previously recommended values for perplexity and n_neighbors are too small and overfit the noise. We also provide a workflow others may use to calibrate hyperparameters in the presence of noise.

Author Summary

In our infinitely complex world, perfect data rid of noise is an unattainable ambition. Hence, our goal is to coerce meaningful information, or the signal, from data inevitably riddled with unwanted, random variation. Advances in technology have allowed us to collect and process biological data of increasing size and complexity, so it is now more important than ever to acknowledge noise in our analyses to ensure random structures are not confused for significant patterns. Many algorithms and ideas have been suggested, some more cognizant of noise than others, but it is still unclear how noise should be handled in various situations. Our experiments, however, indicate typical calibrations of popular analysis methods are inadequately handling noisy, complex biological data. In response, we show and explain how alternate calibrations perform better in the presence of noise and lead to results more faithful to the data. By providing evidence of mishandled noise and presenting solutions, we hope to further the discussion on handling noise in biological data.

1 Introduction

In recent years, non-linear dimension reduction techniques have been growing in popularity due to their usefulness when analyzing high-dimensional data. Biologists use these techniques for a variety of visualization and analytic purposes, including exposing cell subtypes [1], checking for batch effects

[2], and visualizing the trajectories of differentiating cells [3]. The most popular non-linear dimension reduction methods are t-distributed Stochastic Neighbor Embedding (t-SNE, [4]) and Uniform Manifold Approximation and Projection (UMAP, [5]). Both methods have been applied to various types of data within biology ([1], [6], [7]).

Since the introduction of t-SNE and UMAP, hyperparameter calibration has proven to be a difficult task. The most crucial hyperparameters, t-SNE's perplexity and UMAP's n_neighbors, control how large a neighborhood to consider around each point when determining its location in low dimension. Calibration is so troublesome, that perplexity-free versions of t-SNE have been proposed [8]. It is also an extremely important task, since both methods are known to produce unfaithful results when mishandled [9]. For t-SNE, the original authors suggested perplexities between 5 and 50 [4], while recent works have suggested perplexities as large as one percent of the sample size [10]. [11] studied the inverse relationship between perplexity and Kullback-Leibler divergence to design an automatic calibration process that "generally agrees with experts' consensus." For UMAP, the original authors make no recommendation for optimal values of n_neighbors, but their implementation defaults to n_neighbors = 15 [5]. Manual tuning of perplexity and n_neighbors requires a deep understanding of the t-SNE and UMAP algorithms, as well as a general knowledge of the data's structure.

The primary purpose of dimension reduction is to simplify data in a way that eliminates superfluous or nonessential information, i.e. noise. Each dimension reduction method does this slightly differently, but most require hyperparameter calibration. For example, the classical linear method, PCA, requires tuning of the number of principal components. A more contemporary method in biology, PHATE (Potential of Heat-diffusion for Affinity-based Trajectory Embedding) [3], requires tuning of a hyperparameter named diffusion time scale t. PHATE represents the structure of the data by computing local similarities then walking through the data using a Markovian random-walk diffusion process. t determines the number of steps taken in a random walk and "provides a tradeoff between encoding local and global information in the embedding" [3]. Perplexity and n_neighbors serve the same purpose in their respective algorithms. Hence, we believe t-SNE and UMAP are capable of handling noise, but naïve calibrations that disregard noise often result in overfitting.

To assess dimension reduction performance in the presence of noise, we must acknowledge noise during the evaluation process. When the data's structure is available, we can visualize the results and choose the representation that best captures the hypothesized structure. In supervised problems, for example, we look for low-dimensional representations that cluster according to the class labels. For

unsupervised problems, however, the structure is often unknown, so we cannot visually assess each representation. In these cases, we must resort to quantitative measures of performance to understand how well the low-dimensional representation reproduces the high-dimensional data. While this strategy is heavily discussed in the machine learning literature, many prior works disregard the possibility of overfitting when quantitatively measuring performance.

In this paper, we present a framework for studying dimension reduction methods in the presence of noise (Section 3). We then use this framework to calibrate t-SNE and UMAP hyperparameters in both simulated and practical examples to illustrate how the disregard of noise leads to miscalibration (Section 4). We also discuss how other researchers may use this framework in their own work (Section 5) and present a case study that walks the reader through the application of the framework to a modern data set (Section 6).

2 Background

2.1 t-SNE

t-distributed Stochastic Neighbor Embedding (t-SNE, [4]) is a nonlinear dimension reduction method primarily used for visualizing high-dimensional data. The t-SNE algorithm captures the topological structure of high-dimensional data by calculating directional similarities via a Gaussian kernel. The similarity of point x_j to point x_i is defined by

$$p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)}.$$

Thus for each point x_i, we have a probability distribution P_i that quantifies the similarity of x_i to every other point. The scale of the Gaussian kernel, σ_i, is chosen so that the perplexity of the probability distribution P_i, in the information theory sense, is equal to a pre-specified value also named perplexity,

$$\text{perplexity} = 2^{-\sum_{j \neq i} p_{j|i} \log_2 p_{j|i}}.$$

Intuitively, perplexity controls how large a neighborhood to consider around each point when approximating the topological structure of the data. As such, it implicitly balances attention to local and global aspects of the data with high values of perplexity placing more emphasis on global aspects. For the sake of computational convenience, t-SNE assumes the directional similarities are symmetric by

defining

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}.$$

The p_{ij} define a probability distribution P on the set of pairs (i, j) that represents the topological structure of the data.

The goal is to then find an arrangement of low-dimensional points $y_1, \ldots, y_n$ whose similarities q_{ij} best match the p_{ij} in terms of Kullback-Leibler divergence,

$$D_{KL}(P||Q) = \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}.$$

The low-dimensional similarities q_{ij} are defined using the t distribution with one degree of freedom,

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}}.$$

The primary downsides of t-SNE are its inherent randomness, unintuitive results, and sensitivity to hyperparameter calibration. The minimization of KL divergence is done using gradient descent methods with incorporated randomness to avoid stagnating at local minima. As a result, the output differs between runs of the algorithm. Hence, the traditional t-SNE workflow often includes running the algorithm multiple times at various perplexities before choosing the best representation. t-SNE is also known to produce results that are not faithful to the true structure of the data, even when calibrated correctly. For example, cluster sizes and inter-cluster distances aren't always consistent with the original data [13]. Such artifacts of the t-SNE algorithm can be confused for significant structures by inexperienced users.

2.2 UMAP

Uniform Manifold Approximation and Projection (UMAP, [5]) is another nonlinear dimension reduction method that has been rising in popularity. Originally introduced as a more computationally efficient alternative to t-SNE, UMAP is a powerful tool for visualizing high-dimensional data that requires user calibration. While its underlying ideology is completely different from that of t-SNE, the UMAP algorithm is very similar architecturally to the t-SNE algorithm — high-dimensional similarities are computed and the resulting representation is the set of low-dimensional points whose low-dimensional similarities best match the high-dimensional similarities. See [5] for details. The

largest difference is UMAP's default initialization process. UMAP uses Laplacian eigenmaps to initialize the low-dimensional representation, which is then adjusted to minimize the cost function. Most t-SNE implementations use PCA during the initialization process. The initialization process is the primary benefit of the default implementation of UMAP, but t-SNE and UMAP have been shown to perform similarly with identical initializations [7]. Modern implementations of both algorithms are also comparable in speed.

UMAP shares similar disadvantages with t-SNE. It can create unfaithful representations that require experience to interpret and is sensitive to hyperparameter calibration [14].

3 Methods

3.1 Dimension Reduction Framework

Prior works quantitatively measure how well low-dimensional representations match the high-dimensional data. However, if we consider data as a composition of signal and noise, we must not reward capturing the noise. Therefore, we should be comparing the low-dimensional representation against the signal underlying our data, rather than the entirety of the data.

Suppose the underlying signal of our data is described by an r-dimensional matrix $Y \in \mathbb{R}^{n \times r}$. In the context of dimension reduction, the signal is often lower dimension than the original data. Let $p \geq r$ be the dimension of the original data set, and let $\text{Emb} : \mathbb{R}^r \to \mathbb{R}^p$ be the function that embeds the signal in data space. Define $Z = \text{Emb}(Y)$ to be the signal embedded in data space. We then assume the presence of random error. The original data can then be modeled by $Z + \epsilon$ for $\epsilon \sim N_p(0, \Sigma)$. The dimension reduction method φ is applied to $Z + \epsilon$ to get a low-dimensional representation $X \in \mathbb{R}^{n \times q}$. See Figure 1.

Fig 1: Dimension reduction framework

3.2 Reconstruction Error Functions

The remaining piece is a procedure for measuring dimension reduction performance. Suppose we have a reconstruction error function $f(D_1, D_2)$ that quantifies how well the data set D_2 represents the data set D_1. Prior works like [9], [10] , [15], and [16] use various reconstruction error functions to quantify performance; only, they study $f(Z + \epsilon, X)$ to measure how well the constructed representation X

represents the original data $Z + \epsilon$. We argue it is more appropriate to compare X against the signal Y by examining $f(Y, X)$.

Prior works in dimension reduction have suggested various quantitative metrics for measuring dimension reduction performance. In line with recent discussions of perplexity ([10] and [15]), we employ two different metrics — one that measures local performance and one that measures global performance.

For local performance, we use a nearest-neighbor type metric called trustworthiness [17]. Let n be the sample size and $r(i, j)$ the rank of point j among the k nearest neighbors of point i in high dimension. Let $U_k(i)$ denote the set of points among the k nearest neighbors of point i in low dimension, but not in high dimension. Then

$$f_{trust}(D_1, D_2) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{j \in U_k(i)} [r(i, j) - k] .$$

For each point, we are measuring the degree of intrusion into its k-neighborhood during the dimension reduction process. The quantity is then re-scaled, so that trustworthiness falls between 0 and 1 with higher values favorable. Trustworthiness is preferable to simply measuring the proportion of neighbors preserved because it's more robust to the choice of k. For very large values of n, we can get an estimate by only checking a random subsample of points $i_1, \ldots, i_m$. In this case,

$$f_{trust}(D_1, D_2) \approx 1 - \frac{2}{mk(2n - 3k - 1)} \sum_{l=1}^{m} \sum_{j \in U_k(i_l)} [r(i_l, j) - k] .$$

Local performance is the primary concern when applying t-SNE and UMAP, so our experiments focus on maximizing trustworthiness.

For global performance, we use Shepard goodness [16]. Shepard goodness is the Spearman correlation, a rank-based correlation, between high and low-dimensional inter-point distances,

$$f_{\text{Shep}}(D_1, D_2) = \sigma_{\text{Spearman}}(||z_i - z_j||, ||\varphi(z_i) - \varphi(z_j)||).$$

Again for very large values of n, we can get an approximation by calculating the correlation between inter-point distances of a random subsample.

3.3 Using this framework

When using this framework to model examples, three components must be specified: $Z + \epsilon$, Y, and Emb(). These elements describe the original data, the underlying signal, and the embedding of the signal in data space, respectively. When simulating examples, it's natural to start with the underlying signal Y then construct $Z + \epsilon$ by attaching extra dimensions and adding Gaussian noise. The Emb() function is then given by $\text{Emb}(y) = (y, 0, \ldots, 0)$ so that

$$Z + \epsilon = \begin{bmatrix} Y & | & 0 \end{bmatrix} + \epsilon.$$

Practical examples are more tricky because we do not have the luxury of first defining Y. Instead, we are given the data $Z + \epsilon$ from which we must extract Y, or at least our best estimate. This process is dependent on the specifics of the problem and should be based on a priori knowledge of the data. If there is no specific signal of interest, a more general approach can be taken. We used a PCA projection of the data to represent the signal, $Y = \text{PCA}_r(Z + \epsilon)$, where r is the dimension of the projection. For a reasonably chosen r, we expect the first r principal components to contain most of the signal, while excluding most of the noise. Another advantage to using PCA is it gives rise to a natural Emb() function — the PCA inverse transform. If Y is centered, then we may define

$$Z = \text{invPCA}_r(Y) = (Z + \epsilon)V_r V_r^T,$$

where $V_r \in \mathbb{R}^{p \times r}$ contains the first r eigenvectors of $(Z + \epsilon)^T(Z + \epsilon)$ as column vectors.

4 Results

4.1 Simulated Examples

We first looked at simulated examples with explicitly defined signal structures – three low-dimensional examples (Figure 2) and one high-dimensional example. The links example and the high-dimensional example are explored here. See Table 1 and the Supporting Information (SI.1 and SI.2) for the other simulated examples.

Fig 2: Low-Dimensional Simulated Examples

4.1.1 Links Data Set

For the links example, the signal Y consisted of two interlocked circles, each containing 250 points, embedded in three dimensions. $Z + \epsilon$ was constructed by adding seven superfluous dimensions and isotropic Gaussian noise. Various degrees of noise were tested ($sd = 0.5, 1, 1.5, 2, 2.5, 3$).

t-SNE was run using the R package *Rtsne* [19] at varying perplexities. For each perplexity, the algorithm was run 40 times to mimic the ordinary t-SNE workflow. If the distinction between signal and noise were disregarded, a plot of $f_{\text{trust}}(Z + \epsilon, X)$ vs. perplexity could be used to maximize local performance. To avoid overfitting the noise, a plot of $f_{\text{trust}}(Y, X)$ vs. perplexity should be used instead. See Figure 3 for examples of these plots for the $sd = 1$ case. Both plots depict an increase in local performance followed by a decrease as perplexity increases. This cutoff point, however, varies between the two plots. When comparing against the original data, the trustworthiness-maximizing representation was constructed with a perplexity of 40, which is consistent with the original authors' suggestion of 5 to 50 for perplexity [4]. When comparing against the signal, the trustworthiness-maximizing representation was constructed with a perplexity of 80.

Fig 3: Trustworthiness vs. Perplexity (Links $sd = 1$)

With the signal structure known, we are also able to visually assess the trustworthiness-maximizing representations. Figure 4 shows the trustworthiness-maximizing representations for the $sd = 1$ case. Notice the larger perplexity was able to successfully separate the circles in the presence of noise, while the smaller perplexity was not. By using the signal as the frame of reference, our framework correctly rewarded the representation that was able to successfully separate the two links.

Fig 4: Trustworthiness-Maximizing Representations (Links $sd = 1$)

The same pattern held true for other levels of noise. The optimal perplexity was consistently larger when comparing against the signal, rather than the original data (Figure 5).

Fig 5: Optimal Perplexity (Links)

These results suggest larger perplexities perform better in the presence of noise, both quantitatively and qualitatively. We hypothesize t-SNE tends to overfit the noise when the perplexity is too small. Intuitively, small perplexities are more affected by slight perturbations of the data when only

considering small neighborhoods around each point, leading to unstable representations. Conversely, larger perplexities lead to more stable representations that are more robust to noise.

4.1.2 High-Dimensional Clusters

The signal Y consisted of seven Gaussian clusters, each containing 50 points, in seven dimensions. The clusters were drawn from multivariate normal distributions with mean $10e_i$ and random diagonal covariance matrices, where e_i is the i^{th} standard basis vector. The data set $Z + \epsilon$ was constructed from Y by adding 53 superfluous dimensions and isotropic Gaussian noise to all 60 dimensions. Various degrees of noise were tested ($sd = 2, 2.5, 3, 3.5, 4, 4.5$).

When $sd = 3$, local performance peaked at different perplexities when changing the frame of reference (Figure 6). When comparing against the original data, trustworthiness was maximized at a perplexity of 55. When comparing against the signal, trustworthiness was maximized at a perplexity of 60. See Figure 7 for the trustworthiness-maximizing representations. Visually, both representations maintain the original clustering to some extent, but the higher-perplexity representation shows less mixing between the clusters and had a larger average silhouette width (0.178) than the lower-perplexity representation (0.121). This suggests the higher-perplexity representation better maintained the original clustering.

Fig 6: Trustworthiness vs. Perplexity (High-Dimensional Clusters $sd = 3$)

Fig 7: Trustworthiness-Maximizing Representations (High-Dimensional Clusters $sd = 3$)

Figure 8 shows the optimal perplexities for different levels of noise. Again, the trustworthiness-maximizing perplexity was larger when comparing against the signal for all levels of noise.

Fig 8: Optimal Perplexity (High-Dimensional Clusters)

4.2 Practical Examples

In addition to simulated data sets, we looked at three practical data sets: a single-cell RNA sequencing data set [20], a cytometry by time-of-flight (CyTOF) data set [21], and a microbiome data set [22]. For each data set, we compared the optimal perplexity for locally replicating the original data versus the estimated signal. We explore the scRNA-seq data set in detail here. The results of the other two

practical examples can be found in Table 1. The details can be found in the Supporting Information (SI.1 and SI.2).

The scRNA-seq data set was generated from induced pluripotent stem cells collected from three different individuals. The original data includes 864 units and 19,027 readings per unit. To process this zero-inflated count data, columns containing a large proportion of 0's (20% or more) were removed before a log transformation was applied. This step reduced the number of dimensions to 5,431. A PCA pre-processing step further reduced the number of dimensions to 500, which still retained 88% of the variance of the log-transformed data. Hence, the processed data set consisted of 864 observations in 500 dimensions, $Z + \epsilon \in \mathbb{R}^{864\times 500}$. To determine the dimensionality of the signal, we drew a scree plot (Figure 9). Note, the first eigenvalue (2359.357) was cut to fit the plot. A conservative estimate is five dimensions, so Y was extracted by taking the first five principal components, $Y = \text{PCA}_5(Z+\epsilon)$. We computed the t-SNE representations for perplexities ranging from 10 to 280. For each perplexity, 20 different t-SNE representations were computed.

Fig 9: Scree Plot for scRNA-seq Data Set

As with the simulated examples, there is a difference in trend when switching the frame of reference (Figure 10). When compared against the original data, trustworthiness is maximized at a perplexity of 40, which is consistent with [4]'s recommendation of 5 to 50. When compared against the signal, trustworthiness is maximized at a larger perplexity of 120, reinforcing the hypothesis that lower values of perplexity may be overfitting the noise.

Fig 10: Trustworthiness vs. Perplexity for r = 5 (scRNA-seq)

Visual inspection of the trustworthiness-maximizing representations reveals the effect of increasing the perplexity (Figure 11). A hierarchical clustering of the high-dimensional data was computed, then projected onto the trustworthiness-maximizing representations. Both representations depict a similar a structure, but the relative positioning of clusters differs. For example, the Class 1 cluster is the rightmost cluster in the perplexity = 40 representation, while the Class 5 cluster is the rightmost cluster in the perplexity = 120 representation. Furthermore, the left-to-right order of the Class 3, Class 8, and Class 9 clusters is reversed in both representations. In terms of local structure, the Class 3 and Class 7 clusters are better preserved in the perplexity = 40 representation, while the Class 12 cluster is better preserved in the perplexity = 120 representation. The perplexity = 40 representation also suggests

the Class 2 cluster could potentially contain two separate clusters, but this is not consistent with the high-dimensional data according to the dendrogram and higher-order clusterings. The perplexity = 120 representation does not mislead in this way. Overall, the lower perplexity leads to tighter-knit clusters as expected. However, further investigation reveals the over-clustering may be unfaithful to the original data.

Fig 11: Trustworthiness-Maximizing Representations for r = 5 (scRNA-seq)

If we, instead, decide to be more conservative and use the first 10 principal components to represent the signal, we still see a similar trend (Figures 12,13). Trustworthiness still increases then decreases with perplexity. When compared against the original data, trustworthiness is maximized at a perplexity of 50 (Note the optimal perplexity when compared against the original data differed between the two experiments, even though it should theoretically be independent of the chosen signal dimension. This is due to the inherent randomness of the t-SNE algorithm). When compared against the signal, trustworthiness is maximized at a perplexity of 60. By including three extra principal components in the signal, we're assuming the data contains less noise, allowing the model to be more aggressive during the fitting process.

Fig 12: Trustworthiness vs. Perplexity for r = 10 (scRNA-seq)

Fig 13: Trustworthiness-Maximizing Representations for r = 10 (scRNA-seq)

4.3 Summary of Results

See Table 1 for a summary of the results. n, p, and r represent the sample size, dimension of the (post PCA-processed) data, and dimension of the extracted signal, respectively. The optimal perplexity when comparing against the signal was greater than the optimal perplexity when comparing against the original data for every example.

				Optimal Perplexity	
Data Set	n	p	r	signal + noise	signal
Links [13]	500	10	3	40	80
Trefoil [13]	500	10	3	35	100
Mammoth [18]	500	10	3	30	80
High-Dimensional Clusters	210	60	10	55	60
scRNA-seq [20]	864	500	5	40	120
scRNA-seq [20]	864	500	10	50	60
CyTOF [21]	5,000	30	5	50	110
CyTOF [21]	5,000	30	8	45	65
Microbiome [22]	280	66	5	50	90
Microbiome [22]	280	66	8	60	85

Table 1: Summary of Results

4.4 UMAP and n_neighbors

If n_neighbors functions similarly to perplexity, we'd expect small values of n_neighbors to overfit the data as well. An identical experiment was run using the Python package *umap-learn* [5] on the scRNA-seq data. n_neighbor values ranging from 10 to 300 were tested and an n_neighbors value of 190 maximized trustworthiness when comparing against the original data, but an n_neighbors value of 300 maximized trustworthiness when comparing against the signal (Figure 14).

Fig 14: Trustworthiness vs. n_neighbors for UMAP (scRNA-seq)

5 Application

To apply this framework in practice, one must decide how to extract the signal from the data. The signal should include the features of the data one desires to retain throughout the dimension reduction process. When using a PCA projection to serve as the signal, one could draw a scree plot or employ a component selection algorithm such as parallel analysis [23] to determine the dimension of the signal.

With a signal constructed, it remains to compute t-SNE/UMAP outputs at varying perplexities/n_neighbors. It's recommended that at least a couple outputs are computed for each perplexity/n_neighbors to account for randomness in the algorithms. For each output, one must calculate the trustworthiness and Shepard goodness with respect to the signal. From there, one can choose

the representation with the desired balance of local and global performance. A summary is given in Algorithm 1. Sample code is available at `https://github.com/JustinMLin/DR-Framework/`.

Algorithm 1 Measuring Performance in the Presence of Noise

Require: original data $Z + \epsilon$, perplexities $\{p_1, \ldots, p_m\}$ to test, and neighborhood size k

$Y \Leftarrow \text{PCA}_r(Z + \epsilon)$
perplexities $\Leftarrow \{p_1, \ldots, p_m\}$
for perplexity in perplexities **do**
    loop
        $X\_tsne \Leftarrow \text{Rtsne}(Z + \epsilon, \text{perplexity})$
        $trust \Leftarrow \text{trustworthiness}(Y, X\_tsne, k)$
        $shep \Leftarrow \text{Shepard\_goodness}(Y, X\_tsne)$
    end loop
end for
Plot trustworthiness and Shepard goodness values
Choose output with desired balance of local and global performance

It is worth noting that computational barriers may arise, especially for very large data sets. To alleviate such issues, trustworthiness and Shepard goodness can be approximated by subsampling before calculation. Furthermore, t-SNE and UMAP are generally robust to small changes in perplexity and n_neighbors, so checking a handful of values is sufficient. If computing multiple low-dimensional representations is the limiting factor, one can try calibrating the hyperparameters for a subsample before extending to the full data set. [24] found that embedding a ρ-sample, where $\rho \in (0, 1]$ is the sampling rate, with perplexity Perp' gives a visual impression of embedding the original data with perplexity $\text{Perp} = \frac{\text{Perp}'}{\rho}$. With these concessions, applying this framework to calibrate hyperparameters should be feasible for data sets of any size.

6 Case Study

To demonstrate how one might apply this framework, we walk through a detailed case study on a modern scRNA-seq data set.

6.1 Data

Cryopreserved human peripheral blood mononuclear cells (PBMCs) from a healthy female donor aged 25 were obtained by 10x Genomics from AllCells. Granulocytes were removed by cell sorting, followed by nuclei isolation. Paired ATAC and Gene Expression libraries were generated from the isolated nuclei and sequenced. See [25] for details.

6.2 Pre-Processing

Pre-processing was completed using the R package *BPCells* and the steps followed the provided tutorial [26] closely. Low quality cells (those that did not meet the required number of RNA reads, the required number of ATAC reads, or TSS Enrichment cutoffs) were filtered out before a matrix normalization was applied. The cleaned dataset contained 2,600 cells and 1,000 genes. The number of dimensions was then reduced to 500 using PCA, which retained 86% of the original variance. The processed data set to be analyzed contained 2,600 observations in 500 dimensions, $\mathbb{Z} + \epsilon \in \mathbb{R}^{2,600 \times 500}$.

6.3 Determining the Signal

To determine the number of signal dimensions, a scree plot was drawn (Figure 15). The first eigenvalue was approximately 188 but was trimmed to fit the plot. Four dimensions, a relatively conservative estimate, were chosen to represent the signal, $Y \in \mathbb{R}^{2,600 \times 4}$.

Fig 15: Scree Plot for PBMC Data Set

6.4 Results

UMAP was applied with multiple values of n_neighbors. 20 representations were computed for each value, and trustworthiness was measured with respect to both the entire data and the signal. Trustworthiness was maximized at a n_neighbors value of 50 when comparing against the entire data and a value of 70 when comparing against the signal (Figure 16). Cell types (B, T, Monocyte, NK, Dendritic cell, CD8 T) were assigned to each cluster by exploring marker genes (Figure 17). See [26] for details.

Fig 16: Trustworthiness vs. n_neighbors for UMAP (PBMC)

Fig 17: Cell Types (PBMC)

6.5 Analysis

In all three representations, the primary division of cell types is between monocytes and some of the dendritic cells vs. the T, B, CD8 T, and NK cells. In the default UMAP representation, the B cells form a cluster that is quite distinct from the T, CD8 T, and NK cells. As we increase n_neighbors to 50 and 70, the B cell cluster moves closer to the T/CD8 T/NK cell cluster. The closer proximity

of the B cells to the T, CD8 T, and NK cells in the n_neighbors = 70 representation is consistent with the over-arching categorization of T, CD8 T, NK, and B cells as lymphocytes, as opposed to monocytes.

Perhaps a starker difference between the representations concerns the dendritic cells (DCs). In the n_neighbors = 15 and n_neighbors = 50 representations, there are three distinct clusters of DCs, whereas there are only two in the n_neighbors = 70 representation (Figure 18). Principal component analysis of the DCs alone suggests that the DCs are either two clusters, one of which is more diffuse than the other, or three clusters, two of which are fairly close together (Figure 19). Standard metrics for determining the number of clusters suggest the same. The silhouette width metric suggests two clusters, while the gap statistic suggests three (Supporting Information SII.3). However, the three-cluster solution given by k-means and visual inspection of the principal components plot does not align with the three clusters in the n_neighbors = 15 or n_neighbors = 50 representation. The green and orange clusters are represented faithfully, but the third, more diffuse, purple cluster is split across two DC clusters in the n_neighbors = 15 and n_neighbors = 50 representations (Figure 20). The degree of separation is lesser in the n_neighbors = 70 representation. Therefore, the n_neighbors = 15 and n_neighbors = 50 representations inaccurately represent the dendritic cells in a way that the n_neighbors = 70 representation does not.

Fig 18: Plot of Dendritic Cells (PBMC)

Fig 19: PCA Applied to Dendritic Cells (PBMC)

Fig 20: Dendritic Cells Colored According to PCA Projection (PBMC)

7 Discussion

We have illustrated the importance of acknowledging noise when performing dimension reduction by studying the roles perplexity and n_neighbors play in overfitting data. When using the original data to calibrate perplexity, our experiments agreed with perplexities previously recommended. When using the signal, however, our experiments indicated that larger perplexities perform better. Low perplexities/n_neighbors lead to overly-flexible models that are heavily impacted by the presence of noise, while higher perplexities/n_neighbors exhibit better performance due to increased stability.

These considerations are especially important when working with heavily noised data, which are especially prevalent in the world of single-cell transcriptomics [27].

We have also presented a framework for modeling dimension reduction problems in the presence of noise. This framework can be used to study other hyperparameters and their relationships with noise. In the case when a specific signal structure is desired, this framework can be used to determine which dimension reduction method best preserves the desired structure. Further works should explore alternative methods for extracting the signal a in way that preserves the desired structure.

8 Data Availability

All data and code are freely available at `https://github.com/JustinMLin/DR-Framework/`.

References

[1] Amir et al. viSNE enables visualization of high dimensional single-cell data and reveals phenotypic heterogeneity of leukemia. *Nature Biotechnology 31 545-552*, 2013.

[2] Orly Alter, Patrick O. Brown, and David Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *PNAS 97(18) 10101-10106*, 2000.

[3] Moon et al. Visualizing structure and transitions in high-dimensional biological data. *Nat Biotechnology 37(12):1482-1492*, 2019.

[4] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research 9:2579 – 2605*, 2008.

[5] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for dimension reduction. *arXiv preprint arXiv:1802.03426v3*, 2020.

[6] Becht et al. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology 37 38-44*, 2019.

[7] Dmitry Kobak and George C. Linderman. Initialization is critical for preserving global data structure in both t-SNE and UMAP. *Nature Biotechnology 39 156-157*, 2021.

[8] Francesco Crecchi, Cyril de Bodt, Michel Verleysen, John A. Lee, and Davide Bacciu. Perplexity-free parametric t-SNE. *arXiv preprint arXiv:2010.01359v1*, 2020.

[9] Haiyang Huang, Yingfan Wang, Cynthia Rudin, and Edward P. Browne. Towards a comprehensive evaluation of dimension reduction methods for transcriptomic data visualization. *Communications Biology, 5:716*, 2022.

[10] Dmitry Kobak and Philipp Berens. The art of using t-SNE for single-cell transcriptomics. *Nature Communications, 10:5416*, 2019.

[11] Yanshuai Cao and Luyu Wang. Automatic selection of t-SNE perplexity. *arXiv preprint arXiv:1708.03229.v1*, 2017.

[12] Ronald R. Coifman and Stéphane Lagon. Diffusion maps. *Applied and Computational Harmonic Analysis 21:1 5-30*, 2006.

[13] Martin Wattenberg, Fernanda Viégas, and Ian Johnson. How to Use t-SNE Effectively. *Distill*, 2016.

[14] Andy Coenen and Adam Pearce for Google PAIR. Understanding UMAP. *https://pair-code.github.io/understanding-umap/*.

[15] Tara Chari and Lior Pachter. The specious art of single-cell genomics. *PLoS Computational Biology 19(8):e1011288*, 2023.

[16] Mateus Espadoto, Rafael M. Martins, Andreas Kerren, Nina S. T. Hirata, and Alexandru C. Telea. Towards a quantitative survey of dimension reduction techniques. *IEEE Transactions on Visualization and Computer Graphics 27:3*, 2021.

[17] Jarkko Venna and Samuel Kaski. Visualizing gene interaction graphs with local multidimensional scaling. *European Symposium on Artificial Neural Networks*, 2006.

[18] Yingfan Wang, Haiyang Huang, Cynthia Rudin, and Yaron Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-SNE, UMAP, TriMap, and PaCMAP for data visualization. *Journal of Machine Learning Research 22*, 2021.

[19] Jesse H. Krijthe. Rtsne: T-Distributed Stochastic Neighbor Embedding using a Barnes-Hut Implementation. *https://github.com/jkrijthe/Rtsne*, 2015.

[20] Po-Yuan Tung, John D. Blischak, Chiaowen Joyce Hsiao, David A. Knowles, Jonathan E. Burnett, Jonathan K. Pritchard, et al. Batch effects and the effective design of single-cell gene expression studies. *Scientific Reports 7:39921*, 2017.

[21] Dara M. Strauss-Albee, Julia Fukuyama, Emily C. Liang, Yi Yao, Justin A. Jarrell, Alison L. Drake, et al. Human NK cell repertoire diversity reflects immune experience and correlates with viral susceptibility. *Science Translational Medicine 7:297*, 2015.

[22] Manimozhiyan Arumugam, Jeroen Raes, Eric Pelletier, Denis Le Paslier, Takuji Yamada, Daniel R. Mende, et al. Enterotypes of the human gut microbiome. *Nature 473 174-180*, 2011.

[23] Horn, John L. A rationale and test for the number of factors in factor analysis. *Psychometrika 30:2 179-185*, 1965.

[24] Martin Skrodzki, Nicolas Chaves-de-Plaza, Klaus Hildebrandt, Thomas Höllt, and Elmar Eisemann. Tuning the perplexity for and computing sampling-based t-SNE embeddings. *arXiv preprint arXiv:2308.15513v1*, 2023.

[25] Cell Ranger ARC 2.0.0. Single Cell Multiome ATAC + Gene Expression Dataset. *https://www.10xgenomics.com/datasets/pbmc-from-a-healthy-donor-granulocytes-removed-through-cell-sorting-3-k-1-standard-2-0-0*, 2021.

[26] Benjamin Parks. BPCells: Single Cell Counts Matrices to PCA. *https://bnprks.github.io/BPCells/articles/pbmc3k.html*, 2023.

[27] Shih-Kai Chu, Shilin Zhao, Yu Shyr, and Qi liu. Comprehensive evaluation of noise reduction methods for single-cell RNA sequencing data. *Briefings in Bioinformatics 23:2*, 2022.

[28] Ehsan Amid and Manfred K. Warmuth. TriMap: Large-scale dimensionality reduction using triplets. *arXiv preprint arXiv:1910.00204v2*, 2022.

[29] John A. Lee and Michel Verleysen. Quality assessment of dimensionality reduction: Rank-based criteria. *Neurocomputing 72:1431 – 1443*, 2009.

[30] Tobias Schreck, Tatiana von Landesberger, and Sebastian Bremm. Techniques for precision-based visual analysis of projected data. *Sage 9:3*, 2012.

9 Supporting information

SI Simulated Examples

SI.1 Trefoil Plots

SI.2 Mammoth Plots

SII Practical Examples

SII.1 CyTOF Data Set

SII.2 Microbiome Data Set

SIII PBMC Data Set