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 highly discussed in other modeling problems, but almost entirely ignored in the dimension reduction literature: overfitting. If we interpret data as a combination of signal and noise, prior works judge dimension reduction techniques on their ability to capture the entirety of the data, i.e. both the signal and the noise. In the context of other modeling problems, techniques such as feature-selection, cross-validation, and regularization are employed to combat overfitting, but no such precautions are taken when performing dimension reduction. In this paper, we present a framework that models dimension reduction problems in the presence of noise and use this framework to explore the role perplexity and number of neighbors play in overfitting data when applying t-SNE and UMAP. More specifically, we show previously recommended values for perplexity and number of neighbors are too small and tend to overfit the noise. We also present a workflow others may use to calibrate hyperparameters in the presence of noise.

1 Introduction

In recent years, non-linear dimension reduction techniques have been growing in popularity due to their usefulness when analyzing high-dimensional data. To date, they are the most effective tools for visualizing high-dimensional data. Visualization plays a key role in identifying and understanding relationships within the data. It exposes patterns, trends, and outliers that can't be discerned from the raw data. The most popular non-linear dimension reduction technique is t-distributed stochastic neighbor embedding (t-SNE, [3]).

Since the introduction of t-SNE, hyperparameter calibration has proven to be a difficult task. The most crucial hyperparameter, known as perplexity, controls how large a neighborhood to consider around a point when determining its location. Perplexity calibration is so troublesome, that perplexity-free versions of t-SNE have been proposed [1]. It is also an extremely important task, since t-SNE is known to produce unfaithful results when mishandled [2]. The original authors suggested values between 5 and 50 [3], while recent works have suggested perplexities as large as one percent of the sample size [4]. [5] studied the inverse relationship between perplexity and Kullback-Leibler divergence to design an automatic calibration process that "generally agrees with experts' consensus." Manual tuning of perplexity requires a deep understanding of the t-SNE algorithm and general knowledge of the data's structure. When the data's structure is available, we can visualize the results and choose the perplexity 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 the well the low-dimensional representation represents the high-dimensional data. While this strategy is heavily discussed in the machine learning literature, prior works disregard the possibility of overfitting when 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).

2 Background

2.1 t-SNE

t-distributed stochastic neighbor embedding (t-SNE, [3]) 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 every other point to x_i. 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 [3]. 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 main downsides of t-SNE are its inherit randomness and sensitivity to hyperparameter calibration. The minimization of the 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.

2.2 UMAP

Uniform Manifold Approximation and Projection (UMAP, [6]) is another nonlinear dimension reduction method that has been rising in popularity. Like t-SNE, UMAP is a powerful tool for visualizing high-dimensional data that requires user calibration. The architecture of the UMAP algorithm is similar to that of t-SNE's — 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. The cost function and the formulas for high/low-dimensional similarities differ from t-SNE. See [6] for details.

UMAP has a hyperparameter called n_neighbors that is analogous to t-SNE's perplexity. It determines how large a neighborhood to use around each point when calculating high-dimensional similarities. The original authors make no recommendation for optimal values of n_neighbors, but their implementation defaults to n_neighbors = 15 [6].

3 Methods

3.1 Dimension Reduction Framework

Prior works quantitatively measure how well low-dimensional representations match the high-dimensional data. However, if we view the original data as a composition of signal and noise, we must not reward capturing the noise. This would lead to overfitting. 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 captured by an r-dimensional matrix $Y \in \mathbb{R}^{n \times r}$. In the context of dimension reduction, the underlying 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

$$Z + \epsilon \text{ for } \epsilon \sim N_p(0, \Sigma).$$

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.

$$\begin{tikzcd}
 & & & Z + \epsilon \arrow[llld, "\varphi"'] \\
X & & Y \arrow[r, hook, "\text{Emb}"'] & Z \arrow[u] \\
{} \arrow[rrr, "\text{Dimension (low to high)}"'] & & & {}
\end{tikzcd}$$

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 [7], [8], [2], and [4] 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 underlying 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 ([4] and [7]), we focus on 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 [9]. 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

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

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 $i_1, \dots, i_m$. In this case,

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

For global performance, we use Shepard goodness [8]. Shepard goodness is the Kendall correlation (a rank-based correlation) between high and low-dimensional inter-point distances,

$$f_{\text{Shep}}(D_1, D_2) = \sigma_{\text{Kendall}}(||z_i - z_j||^2, ||\varphi(z_i) - \varphi(z_j)||^2).$$

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 our 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

$$\text{Emb}(y) = (y, 0, \dots, 0),$$

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 left to the researcher 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 reasonable number of dimensions r, we would 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

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

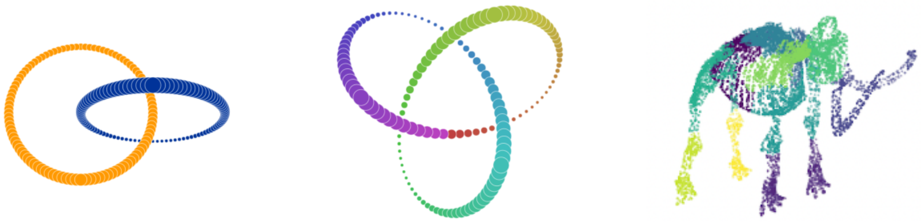

Fig 2: Low-Dimensional Simulated Examples; links and trefoil from [10], mammoth from [11]

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.

For the three low-dimensional examples, the signal Y consisted of 500 points in three dimensions. The dataset $Z+\epsilon$ was constructed from Y by adding seven superfluous dimensions and isotropic Gaussian noise to all 10 dimensions. We then ran t-SNE using the R package *Rtsne* with perplexities ranging from 10 to 160. For each value of perplexity, we ran the algorithm 20 times to mimic the ordinary t-SNE workflow. If we were to disregard the distinction between signal and noise, a plot of $f_{\text{Shep}}(Z+\epsilon, X)$ vs. $f_{\text{trust}}(Z+\epsilon, X)$ could be used to calibrate perplexity. To avoid overfitting the noise, a plot of $f_{\text{Shep}}(Y, X)$ vs. $f_{\text{trust}}(Y, X)$ should be used instead. See Figure 3a for plots of the links example. Both plots depict an increase in global performance as perplexity increases. Both plots also depict an increase in local performance followed by a decrease as perplexity increases. This trend is more evident when we plot trustworthiness vs. perplexity (Figure 3b).

Past a certain perplexity, local performance begins to deteriorate with increasing perplexity, exhibiting

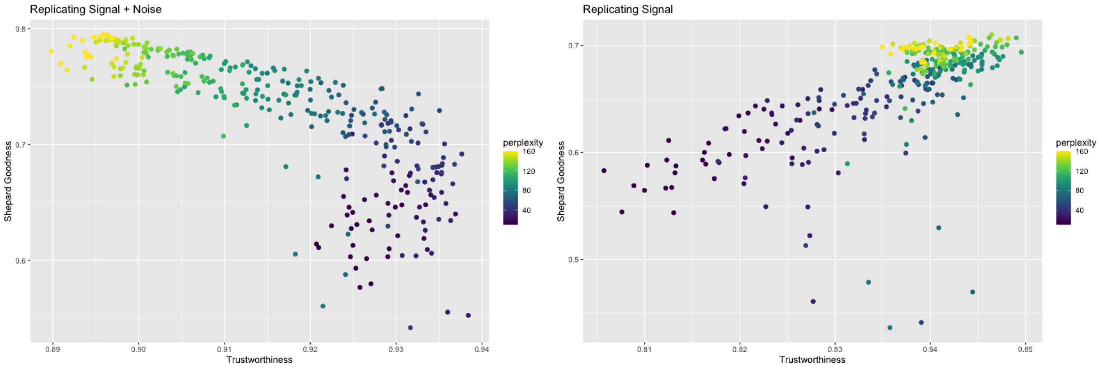

Fig 3a: Shepard Goodness vs. Trustworthiness (Links)

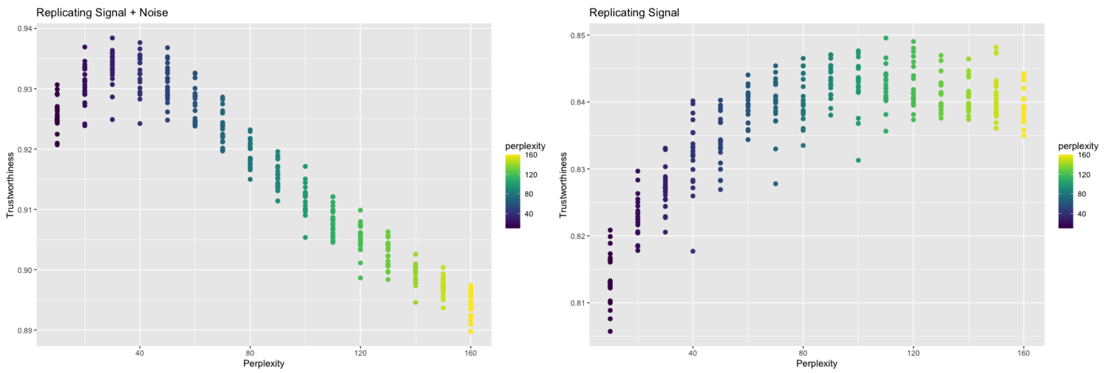

Fig 3b: Trustworthiness vs. Perplexity (Links)

the trade-off between global and local performance. This cutoff point, however, varies between the two plots. When comparing against the original data, a perplexity of 30 maximizes trustworthiness, which is consistent with the original authors' suggestion of 5 to 50 for perplexity [3]. When comparing against the signal, a perplexity of 110 maximizes trustworthiness. We hypothesize t-SNE tends to overfit the noise when the perplexity is too low. 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 less affected by noise. The other two low-dimensional simulated examples exhibit a similar trend (See Table 1 and Supporting Information).

The high-dimensional example was constructed by generating seven 10-dimensional Gaussian clusters with varying centers. More specifically, the signal Y contained 210 points in 10 dimensions. The dataset $Z + \epsilon$ was constructed from Y by adding 50 superfluous dimensions and isotropic Gaussian noise to all 60 dimensions. Again, local performance peaked at different perplexities when changing the frame of reference (Figure 4). When comparing against the original data, trustworthiness was maximized at a perplexity of 40. When comparing against the underlying signal, trustworthiness was maximized at a

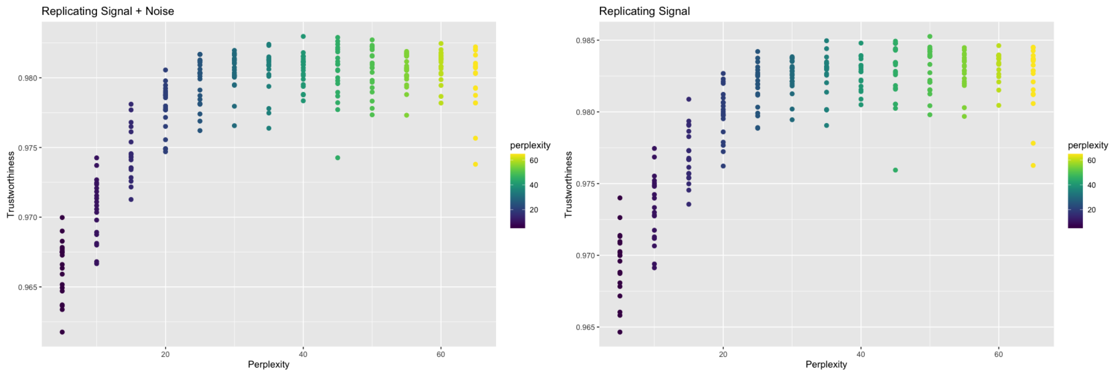

Fig 4: Trustworthiness vs. Perplexity (High-Dimensional Clusters)

perplexity of 50.

4.2 Practical Examples

In addition to simulated datasets, we looked at three practical datasets: a cytometry by time-of-flight (CyTOF) dataset [12], a single-cell RNA sequencing dataset [13], and a microbiome dataset [14]. For each dataset, we compared the optimal perplexity for locally replicating the original data versus just the signal at varying numbers of signal dimensions. We explore the CyTOF dataset in detail here. Summaries for the other two practical examples can be found in Table 1, and the data processing details for the other two practical examples can be found in the Supporting Information.

The CyTOF dataset contained 239,933 observations in 49 dimensions [12]. To reduce the computational load, a subset of 5,000 observations was sampled. In line with the ordinary t-SNE workflow, a log transformation was followed by a PCA pre-processing step to reduce the number of dimensions to 30, which still retained 77% of the variance in the original data. The processed data set to be studied consisted of 5,000 observations in 30 dimensions,

$$Z + \epsilon \in \mathbb{R}^{5,000 \times 30}.$$

To determine the dimension of the signal, we drew a scree plot (Figure 5), which depicts flatening after the fifth eigenvalue. And thus, Y was extracted by taking the first five principal components,

$$Y = \mathrm{PCA}_5(Z + \epsilon).$$

We computed the t-SNE representations for perplexity values ranging from 10 to 300. For each perplexity value, 20 different t-SNE representations were computed. Figure 6a contains the plots for $f_{\mathrm{Shep}}(Z+\epsilon, X)$ vs. $f_{\mathrm{trust}}(Z+\epsilon, X)$ and $f_{\mathrm{Shep}}(Y, X)$ vs. $f_{\mathrm{trust}}(Y, X)$.

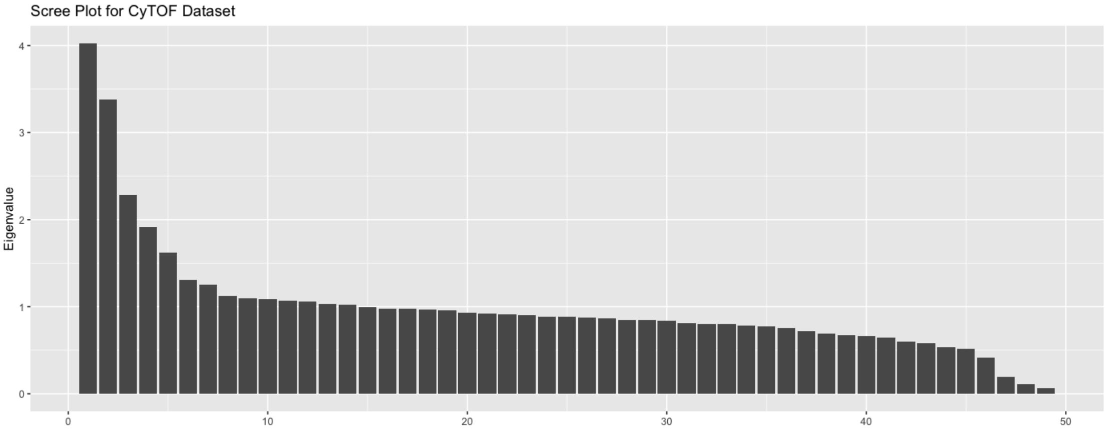

Fig 5: Scree Plot for CyTOF Dataset

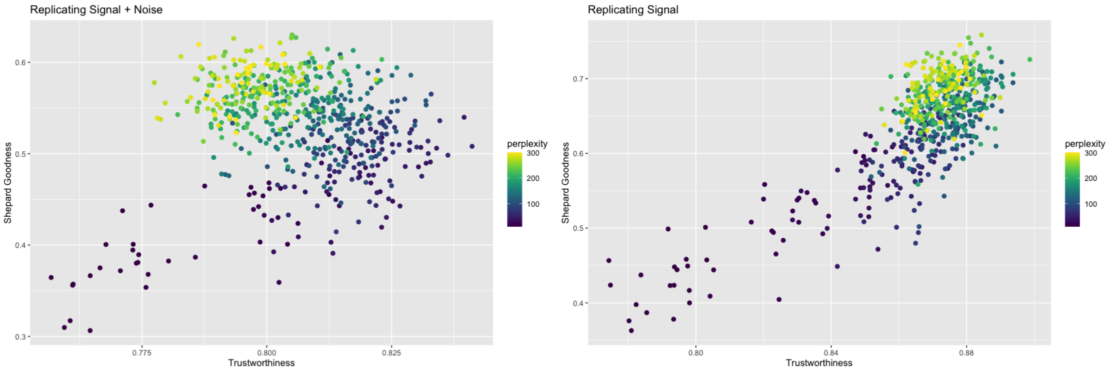

Fig 6a: Shepard Goodness vs. Trustworthiness (CyTOF)

As with the simulated examples, there is a difference in trend when the frame of reference is the original data versus when it's just the underlying signal (Figure 6b). When compared against the original data, trustworthiness is maximized at a perplexity of 50, which is consistent with [4]'s recommendation of setting perplexity to $n/100$. When compared against the underlying signal, trustworthiness is maximized at a larger perplexity of 220, reinforcing the hypothesis that lower values of perplexity may be overfitting the noise.

If we, instead, decide to be a little more conservative and use the first eight principal components to represent the signal, we still see a similar trend (Figure 7). Trustworthiness still increases then decreases with perplexity. When compared against the original data, trustworthiness is maximized at a perplexity of 45. When compared against the signal, trustworthiness is maximized at a perplexity of 65. 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.

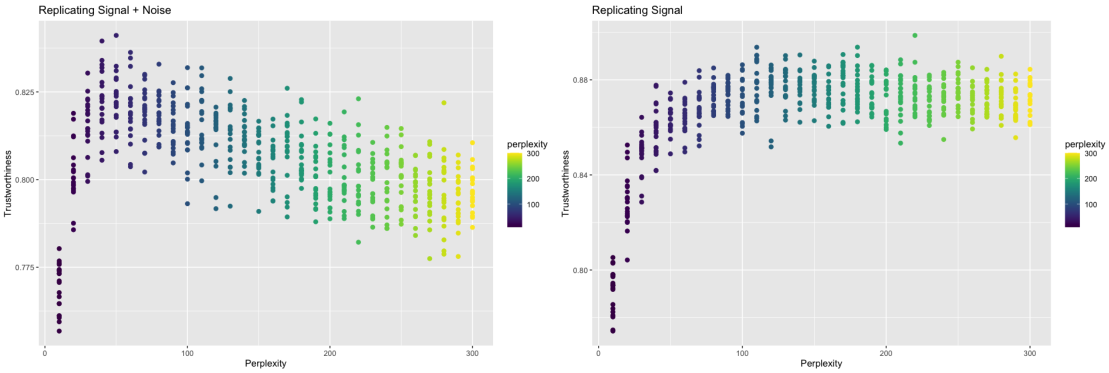

Fig 6b: Trustworthiness vs. Perplexity (CyTOF)

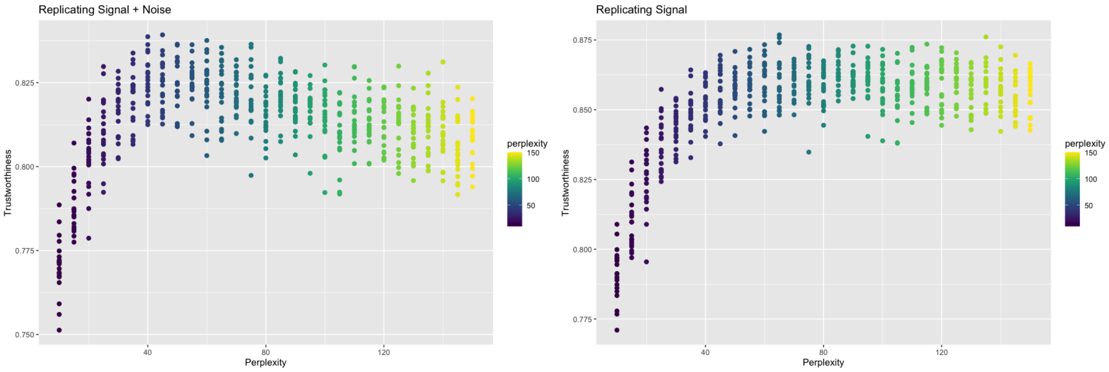

Fig 7: Trustworthiness vs. Perplexity for $r = 8$ (CyTOF)

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.

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* on the CyTOF data. n_neighbor values ranging from 10 to 300 were tested on the same CyTOF data set. An n_neighbors value of 120 maximized trustworthiness when comparing against the original data, but an n_neighbors value of 160 maximized trustworthiness when comparing against the underlying signal. See Supporting

				Optimal Perplexity	
Dataset	n	p	r	signal + noise	signal
Links [10]	500	10	3	30	110
Trefoil [10]	500	10	3	30	110
Mammoth [11]	500	10	3	20	100
High-Dimensional Clusters	210	60	10	40	50
CyTOF [12]	5,000	30	5	50	220
CyTOF [12]	5,000	30	8	45	65
scRNA-seq [13]	864	500	5	40	120
scRNA-seq [13]	864	500	10	50	60
Microbiome [14]	280	66	5	50	90
Microbiome [14]	280	66	8	60	85

Table 1: Summary of results

Information for plots.

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 [15] to determine the dimension of the signal.

With a signal constructed, it remains to compute t-SNE outputs at varying perplexities. It's recommended that at least a couple outputs are computed for each perplexity to account for t-SNE's inherit randomness. 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 desirable balance of local and global performance. A summary is given in Algorithm 1. Sample code is available at `https://github.com/JustinMLin/DR-Framework/`.

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 is generally robust to small changes in perplexity [3], so checking a handful of different perplexities is sufficient. If computing the t-SNE representations is the limiting factor, the perplexity can be calibrated for a subsample of the data, instead. [16] 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 perplexity is feasible for data sets of any size.

Algorithm 1 Measuring Performance in the Presence of Noise

```
Require: original data $Z + \epsilon$, perplexities $\{p_1, \dots, p_m\}$ to test, and neighborhood size $k$
$Y \Leftarrow \text{PCA}_r(Z + \epsilon)$
perplexities $\Leftarrow \{p_1, \dots, p_m\}$
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
```

6 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 just the signal, however, our experiments indicated that larger perplexities performed better. Low perplexities lead to overly-flexible t-SNE models that are heavily impacted by the presence of noise, while higher perplexities 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 [17].

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 in way that preserves the desired structure.

7 Data Availability

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

References

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

[2] 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.

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

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

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

[6] Leland McInnes, John Healy, and James Melville. *arXiv preprint arXiv:1802.03426v3*, 2020.

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

[8] 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.

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

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

[11] 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.

[12] 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.

[13] 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.

[14] 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.

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

[16] 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.

[17] 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.

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

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

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

8 Supporting information

S1 Fig. Trefoil Plots

S2 Fig. Mammoth Plots

S3 Fig. UMAP Plots

S4 Fig. scRNA Plots (r = 5)

S5 Fig. scRNA Plots (r = 10)

S6 Fig. Microbiome Plots (r = 5)

S7 Fig. Microbiome Plots (r = 8)