Uncertain Choices in Method Comparisons: An Illustration with t-SNE and UMAP

Bachelor's Thesis
Faculty of Mathematics, Informatics and Statistics
Department of Statistics
Ludwig–Maximilian–University
Munich

Submitted by
Philipp Weber
Supervisor: Dr. Sabine Hoffmann

Munich, 21 April 2023

I hereby declare that this thesis was composed by myself and that the work contained herein is my own except where explicitly stated otherwise in the text.

Munich, 21 April 2023 Signature

Abstract

To help with the visualization of high dimensional data, dimension reduction techniques have become essential. Two such techniques that have gained a lot of popularity in the last years are t-Distributed Neighbor Embedding (t-SNE) and Uniform Manifold Approximation and Projection (UMAP).

In this thesis we examined these two algorithms, first theoretically describing and comparing them and then analyzing their performance and the influence of certain parameters. We ran t-SNE and UMAP with different parameter settings on six datasets and calculated three quality measures for each outcome. We then analyzed these results through linear models and compared them with different plots.

Most of the parameters we examined in this thesis influenced the quality of the embedding. For some parameters one setting was clearly superior, while other parameters were more of a trade-off between different quality measures. In our analysis, t-SNE performed a bit better than UMAP regarding all three quality measures, which is surprising, since UMAP is often regarded to better preserve the global structure of the data (Becht et al., 2018).

Contents

Chapter 1

Introduction

Dimensionality reduction is an important task in the field of data analysis and visualization, as it creates a low dimensional representation of high dimensional data, enabling one to understand and analyze complex datasets more easily. Two popular algorithms for dimensionality reduction are t-Distributed Stochastic Neighbor Embedding (t-SNE) by van der Maaten and Hinton (2008) and Uniform Manifold Approximation and Projection (UMAP) by McInnes et al. (2018), which have been shown to be effective and produce visually appealing results across many domains, such as art (Vermeulen et al., 2021), music (Philippe Hamel and Douglas Eck, 2010), finance (Greengard et al., 2020) and biology (Kobak and Berens, 2019).

However, contrary to other well known dimension reduction techniques, such as Principal Component Analysis (PCA), t-SNE and UMAP have a number of user-defined hyperparameters that influence the result. Many of these parameters are unfortunately not very intuitive, so for people who are unfamiliar with them, it can be hard to understand what each parameter does and how different parameter settings will affect the outcome, which makes selecting the values a challenging task.

The main goal of this thesis is to investigate which parameters affect the quality of the outcome the most, if there are certain settings that always produce good results and whether one of these two techniques is generally superior. To achieve this, a number of parameter setting combinations will be applied to different datasets and the quality of each output will be assessed via multiple quality measures.

The structure of this thesis will be the following: The second chapter will be a description and comparison of the algorithms, strengths, weaknesses and hyperparameters of UMAP and t-SNE. In the third chapter, the methods used to examine the influence of the parameters will be explained, with the results being presented in the fourth chapter and discussed in the fifth.

Chapter 2

Background

2.1 t-SNE

The dimension reduction technique t-SNE (t-Distributed Stochastic Neighbor Embedding) by van der Maaten and Hinton (2008) is a variation of the Stochastic Neighbor Embedding (SNE) technique by Hinton and Roweis (2002). The main differences between SNE and t-SNE are in the cost function: To make the optimization easier, t-SNE uses a symmetric version of the SNE cost function that has simpler gradients and, instead of the Gaussian used by SNE, t-SNE uses a t-distribution to calculate the similarities in the low dimensional space (van der Maaten and Hinton, 2008).

2.1.1 Algorithm

This description of the t-SNE algorithm is based on van der Maaten and Hinton (2008) and, for better comparability with the UMAP algorithm, McInnes et al. (2018).

High Dimensional Similarities

The t-SNE algorithm starts by computing pairwise similarities in the high dimensional input space X. First, for every pair of datapoints, x_j and x_i, a Gaussian similarity with respect to the Euclidean distance between the two points is calculated:

$$v_{j|i} = \exp(-\|x_i - x_j\|^2/2\sigma_i^2)$$

where σ_i^2 is the variance parameter of the underlying Gaussian distribution (McInnes et al., 2018).
After that, these similarities are normalized and converted into conditional probabilities:

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

which can be interpreted as the conditional probability that "x_i would pick x_j as its neighbor if neighbors were picked in proportion to their probability density under

a Gaussian centered at x_i"(van der Maaten and Hinton, 2008). It follows that for points, which are close together in X, the value of $p_{j|i}$ will be relatively high and for points that are far from each other, it will be extremely low. Since only pairwise similarities are of interest to us, the value of $p_{i|i}$ is set to 0 (van der Maaten and Hinton, 2008).

In order to calculate $p_{j|i}$, the value of σ_i is needed. Depending on the density of the data around x_i, different values for σ_i are appropriate. The denser the region surrounding x_i, the lower σ_i should be. The exact value of σ_i, which produces a probability distribution, P_i, over the other datapoints, is chosen through a binary search for a P_i with a certain perplexity. The perplexity is a user-defined hyperparameter:

$$Perp(P_i) = 2^{H(P_i)}$$

with $H(P_i)$ being the Shannon entropy, a measure of average information of a variable, of P_i measured in bits (meaning the base of the logarithm is 2):

$$H(P_i) = -\sum_j p_{j|i} \log_2 p_{j|i}$$

The typical values for the perplexity range from 5 to 50 and it can be interpreted as "a smooth measure of the effective number of neighbors" (van der Maaten and Hinton, 2008).

The next step is to symmetrize and further normalize the similarities to get a joint probability distribution P. These similarities are given by:

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

to ensure that $\sum_j p_{ij} > \frac{1}{2n}, \forall x_i \in X$, so that each datapoint contributes significantly to the cost function, which will be discussed later (van der Maaten and Hinton, 2008).

To speed up the process, these high dimensional similarities are often not calculated on the original data. Instead, a Principal Component Analysis is used, to reduce the number of dimensions (van der Maaten and Hinton, 2008).

Initialization

The initialization of the low dimensional space is done randomly. The points are sampled from the Gaussian $\mathcal{N}(0, 10^{-4}I)$ (van der Maaten and Hinton, 2008).

Low Dimensional Similarities

Next, we have to calculate the pairwise similarities in the low dimensional space Y. Instead of a Gaussian like in the high dimensional space, a t-distribution with one degree of freedom is used on the squared Euclidean distance (van der Maaten and Hinton, 2008).

As the first step we calculate w_{ij}:

$$w_{ij} = (1 + \|y_i - y_j\|^2)^{-1}$$

which is then normalized to get the low dimensional similarities q_{ij}:

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

Like in the high dimensional case, the value of q_{ii} is set to 0 (van der Maaten and Hinton, 2008).

Optimization

To measure how faithful the low dimensional similarities q_{ij} represent the high dimensional similarities p_{ij}, t-SNE uses the Kullback-Leibler divergence between the high dimensional probability distribution P and the low dimensional probability distribution Q (van der Maaten and Hinton, 2008).

Therefore, the cost function is given by:

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

which can be arranged into constant and non-constant contributions:

$$C = \sum_{i \neq j} p_{ij} \log p_{ij} - p_{ij} \log q_{ij}$$

It is optimized via gradient descent. Gradient descent is an algorithm to minimize a function $f(x), x \in \mathbb{R}^d$, that takes the following steps (Konečný and Richtárik, 2013):

1. Compute the gradient $f'(x)$ of the function
2. Pick (random) initial values x_1
3. Update parameter values: $x_{k+1} = x_k - hf'(x_k)$ with h being the user-defined learning rate
4. Repeat step 3 until the gradient is almost zero or for a user-defined number of times

The gradient with respect to y_i is given by:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + \|y_i - y_j\|^2)^{-1}.$$

The gradient can be interpreted as a set of springs between y_i and every other datapoint in the low dimensional space. These "springs" repel other points if the modeled low dimensional distance is too small and attract them if it is too big (van der Maaten and Hinton, 2008).

A relatively large momentum term is added to the gradient for a faster optimization and to avoid poor local minima. This updated gradient is given by:

$$\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}),$$

where $\mathcal{Y}^{(t)}$ is the solution of the t-th iteration, η the learning rate and $\alpha(t)$ the momentum at iteration t. The values used by van der Maaten and Hinton (2008) are the following: $\alpha(t) = 0.5$ for $t < 250$ and $\alpha(t) = 0.8$ for $t \geq 250$, $T = 1000$, with T being the number of iterations and $\eta = 100$. t-SNE uses the adaptive learning rate scheme by Jacobs (1988), which means that the learning rate is updated after every iteration (van der Maaten and Hinton, 2008).

A way to further improve the optimization is called "early exaggeration", where the p_{ij}'s are multiplied by an exaggeration factor at the beginning of the optimization, which changes the cost function to the following:

$$C = \sum_{i \neq j} \text{exageration factor} \cdot p_{ij} \log \frac{\text{exageration factor} \cdot p_{ij}}{q_{ij}}$$

This leads to widely separated but tight clusters, because the model is now encouraged to model the big p_{ij}'s with big q_{ij}'s, but the q_{ij}'s still only add up to 1 (van der Maaten and Hinton, 2008). An exaggeration factor of 4 for the first 50 iterations was chosen by van der Maaten and Hinton (2008).

2.1.2 Hyperparameters

In this thesis we used the FIt-SNE implementation by Linderman et al. (2019),which is a faster approximation of the standard t-SNE algorithm. It has a number of user-defined hyperparameters that influence the outcome, which will be covered in this section. The focus lies on the three main parameters used by Kobak and Linderman (2021) to improve the quality of the output. A short description of the other parameters can be found in the Appendix A.

Initialization

This parameter determines the initial placement of the points in the low dimensional embedding. If set to "pca", the first k principal components will be used for initialization, with k being the number of dimensions of the embedding. The alternatives are "random", for a random initialization or to provide an array for a custom initialization. According to Kobak and Linderman (2021), a PCA initialization helps to better preserve the global structure of the data. The default setting is "pca".

Perplexity

Perplexity is a measure for information. In t-SNE, the perplexity is used to set the number of effective nearest neighbors. Smaller values will preserve the local structure better, while larger values help to better preserve the global structure. According to van der Maaten and Hinton (2008), the typical values range from 5 to 50. For large data sets Kobak and Berens (2019) suggest using a multi-scale approach, where multiple perplexity values are used simultaneously to preserve both local and global structure. This is done by calculating the $p_{j|i}$'s for all perplexity values and averaging them. Whenever $n/100 \gg 30$ they recommend a perplexity combination of 30 and $n/100$. The default value is 30.

Learning rate

The learning rate determines the step size at each iteration of the gradient descent used during the optimization. A high learning rate converges faster but might skip minima because the step size is too big, while a low learning rate might take too long to converge or get stuck in a suboptimal local minimum (Buduma and Locascio, 2017). The default value is "auto", which sets the learning rate to n/exaggeration_factor, or to 200 if n/exaggeration_factor < 200.

2.1.3 Strengths and Weaknesses

Compared to other dimension reduction techniques, like the PCA, t-SNE excels at data visualization, specifically of the local structure of high dimensional data. This focus on retaining the local structure is especially helpful if the data lies on or near a non-linear manifold, in which case a linear dimension reduction technique that focuses more on retaining the global structure of the data would not be able to produce a good visualization that accurately represents the non-linear structure of the data (van der Maaten and Hinton, 2008).

However, visualization is the only intended use for t-SNE. In many other uses for dimensionality reduction, the interpretability of the dimensions of the embedding space is of great importance but not given for t-SNE. Also, because of the behavior of the t-distribution in high dimensional spaces, the local structure of the data might not be preserved well if t-SNE is used for a more general dimensionality reduction to $d > 3$ dimensions (van der Maaten and Hinton, 2008).
Because of the non-convexity of the t-SNE cost function, the constructed solution is only a local optimum and it depends on several user-defined hyperparameters. The solution will also be different each time t-SNE is run on the same dataset, since the initialization is done randomly (van der Maaten and Hinton, 2008).
Since the t-SNE algorithm mainly utilizes the local structure of the data, it is susceptible to the curse of dimensionality. The local linearity assumption that is implicitly made by the usage of the Euclidean distance may be violated in data sets with a high intrinsic dimensionality, which can cause t-SNE to be less successful (van der Maaten and Hinton, 2008).
Lastly, the focus on accurately representing the local structure comes at the cost of retaining the global structure. t-SNE does reveal some global structure, but if that is of primary interest, then t-SNE may not be the best suited technique for the task (van der Maaten and Hinton, 2008).

2.2 UMAP

UMAP (Uniform Manifold Approximation and Projection) is a dimension reduction technique by McInnes et al. (2018). It has theoretical foundations based in Riemannian geometry and algebraic topology, which won't be covered here but can be found in (McInnes et al., 2018). Instead of the typical description as a graph based algorithm, a version using the same similarity notation as the t-SNE algorithm based on

(McInnes et al., 2018, Appendix C) will be presented here. This is done for easier comparability later.

2.2.1 Algorithm

High Dimensional Similarities

The first step in calculating the high dimensional similarities is to compute the k nearest neighbors of every datapoint x_i under a metric d, which is usually, but not necessarily, the Euclidean distance (McInnes et al., 2018).

To define the high dimensional similarities we need to specify two parameters, ρ_i and σ_i. For each x_i, ρ_i is given by:

$$\rho_i = \min\{d(x_i, x_{i_j}) | 1 \leq j \leq k, d(x_i, x_{i_j}) > 0\},$$

and we set σ_i such that:

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(x_i, x_{i_j}) - \rho_i)}{\sigma_i}\right) = \log_2(k).$$

Now we can calculate the high dimensional similarities $v_{j|i}$ between every x_i and its k nearest neighbors with:

$$v_{j|i} = \exp\left(\frac{-d(x_i, x_j) - \rho_i}{\sigma_i}\right),$$

and set $v_{j|i} = 0$ for all other x_j. ρ_i ensures that every point has at least one other point with a high dimensional similarity of one, which helps with the representation on very high dimensional data. σ_i is a normalization factor, which depends on the density of the region surrounding x_i (McInnes et al., 2018).

The last step is to symmetrize the similarities via the fuzzy set union using the probabilistic t-conorm:

$$v_{ij} = (v_{j|i} + v_{i|j}) - v_{j|i}v_{i|j}$$

Fuzzy sets are a generalization of classical sets, where the elements have degrees of membership valued in the real unit interval $[0, 1]$. A fuzzy union or t-conorm, the terms can be used interchangeably, is one of many different possible generalizations of the classical set union (Klir and Yuan, 1995).

Initialization

UMAP uses Spectral Embedding (Laplacian Eigenmaps), a graph based dimension reduction algorithm, for the initialization. The following brief description of the algorithm is based on (Belkin and Niyogi, 2003).

Given k points $x_1, ..., x_k \in \mathbb{R}^l$, there are three main steps:

1. **Constructing the adjacency graph:** "Close" vertices are connected by an edge. Determining which points are close can be done via a n nearest neighbor approach, where each vertex is connected to its n nearest neighbors based on the squared Euclidean distance, or a ϵ-neighborhood approach, where vertices are connected if the squared Euclidean distance is less than ϵ.

2. **Choosing the weights:** There are, again, two possible ways to do this. The simple way is to set $W_{ij} = 1$, if the vertices i and j are connected by an edge, and $W_{ij} = 0$ if not. The other way is to set $W_{ij} = \exp\left(\frac{\|x_i - x_j\|^2}{t}\right)$ for connected vertices, where the parameter $t \in \mathbb{R}$ needs to be chosen.

3. **Obtaining the Eigenmaps:** The next step is to solve the generalized eigenvector problem:

$$Lf = \lambda Df$$

with $D_{ii} = \sum_j W_{ji}$ and $L = D - W$. L is called the Laplacian matrix.

Let $f_0, ..., f_{k-1}$ be the eigenvectors to this problem, in ascending order according to their eigenvalues. The eigenvector f_0 corresponding to the eigenvalue 0 is left out and the next m eigenvectors are used for obtaining the m-dimensional representations $y_1, ..., y_k$:

$$y_i = (f_1(i), ..., f_m(i))$$

Low Dimensional Similarities

UMAP uses a modified t-distribution to calculate the low dimensional similarities, which are given by:

$$w_{ij} = (1 + a\|y_i - y_j\|^{2b})^{-1},$$

with a and b being positive user-defined values. The default values are $a \approx 1.929$ and $b \approx 0.7915$ (McInnes et al., 2018).

Optimization

UMAP uses the cross entropy as a cost function, to measure how well the low dimensional similarities w_{ij} represent the high dimensional similarities v_{ij}. It is given by:

$$C = \sum_{i \neq j} v_{ij} \log\left(\frac{v_{ij}}{w_{ij}}\right) + (1 - v_{ij}) \log\left(\frac{1 - v_{ij}}{1 - w_{ij}}\right),$$

which can be arranged into constant (containing only v_{ij}) and non-constant (containing w_{ij}) contributions:

$$C = \sum_{i \neq j} v_{ij} \log v_{ij} + (1 - v_{ij}) \log(1 - v_{ij}) - v_{ij} \log w_{ij} - (1 - v_{ij}) \log(1 - w_{ij}).$$

The cost function is optimized (minimized) via stochastic gradient descent (SGD) (McInnes et al., 2018). Generally, the difference between SGD and regular gradient descent is that SGD, instead of using the whole data set every time, randomly chooses a data point at every iteration of the parameter update to compute the gradient. This reduces the amount of computations and thus the run time dramatically (Konečný and Richtárik, 2013).

To be more precise, the way it works specifically for UMAP is that points in the low dimensional space are moved one at a time. When a point is selected to be moved it is attracted by one of its high dimensional neighbors and repulsed by a sampling of other points (McInnes et al., 2018).

2.2.2 Hyperparameters

In this thesis the "umap" package by Konopka (2022) will be used. It has a number of user-defined hyperparameters, which will be described below. The focus is on three main parameters, which will be covered in this section. A short description of the other parameters can be found in the Appendix A

Initialization

The "init" parameter determines the initial placement of the points in the low dimensional embedding. If set to "spectral", Spectral Embedding is used. Alternatively, it is also possible to choose "random", for a random initialization, or to provide a matrix with coordinates for the initialization. The default setting is "spectral".

Number of nearest neighbors

This parameter sets the number of nearest neighbors to consider when calculating the high dimensional similarities. It is comparable to the perplexity from t-SNE and can be seen as a trade-off between retaining the local and global structure of the data (McInnes et al., 2018). The default value is 15.

Minimum distance

This parameter controls how tightly the points in the low dimensional embedding can be packed together. It influences, together with the "spread" parameter, the calculation of the a and b values used to alter the t-distribution when calculating the low dimensional similarities. Low values will result in clumpier embeddings, which can be useful for clustering. Higher values will force the points to spread out more and can help prevent overplotting issues (McInnes et al., 2018). Overplotting describes a problem where multiple data points with similar or identical values overlap, making the individual observations non-distinguishable (Dang et al., 2010). The default value is 0.1.

2.2.3 Strengths and Weaknesses

Similarly to t-SNE, UMAP is mostly used for visualization of (very) high dimensional data and focuses on retaining the local structure of the data, which is especially useful if the data lies on a non-linear manifold (McInnes et al., 2018; van der Maaten

and Hinton, 2008).
Also, in contrast to t-SNE, all decisions regarding the UMAP algorithm are based on mathematical theory, instead of being derived through experimentation (McInnes et al., 2018).

Although it is sometimes used for other tasks than visualization, UMAP has the same interpretability problem as t-SNE. The dimensions of the embedding often need to be interpretable, in which case a linear dimension reduction technique such as PCA is more suitable (McInnes et al., 2018).
As with t-SNE, the non-convex optimization problem of UMAP means that a solution is only a local optimum. Also, because of the probabilistic component from the SGD, the solution may be different every time UMAP is run on the same data set (McInnes et al., 2018).
Another weakness UMAP shares with t-SNE is the retention of the global structure. Although UMAP is often said to retain the global structure better than t-SNE, mostly due to the Spectral Embedding used for the initialization, its focus is still more on accurately representing local structure. Therefore, if the preservation of the global structure of the data is the primary concern, other techniques may be better suited (McInnes et al., 2018; Kobak and Linderman, 2021).
Lastly, one has to be careful when using UMAP on small datasets. The UMAP algorithm makes some approximations to improve the computational efficiency, which can result in suboptimal embeddings for datasets with less than 500 samples (McInnes et al., 2018).

2.3 Comparison

In this section we will compare the two techniques. The first part will be a theoretical comparison of the two algorithms and the second part will be an empirical comparison, where the current state of research will be presented.

2.3.1 Theoretical Comparison

In this part, the theoretical differences and similarities between t-SNE and UMAP in the four steps of the algorithms will be discussed.

High Dimensional Similarities

UMAP, unlike t-SNE, only calculates the high dimensional similarities between each point and its k nearest neighbors instead of all the points, which avoids some unnecessary computations.

To calculate the high dimensional similarities $v_{j|i}$, t-SNE uses a Gaussian with respect to the Euclidean distance between x_i and x_j. UMAP does not use a Gaussian and the metric used can be freely chosen. σ_i, a normalizing factor based on the density around x_i, is used in both algorithms and although it is calculated differently it fulfills a similar function (McInnes et al., 2018).

After calculating $v_{j|i}$, t-SNE first normalizes the similarities before symmetrizing them, while UMAP symmetrizes them right away.

Initialization

In t-SNE, the initialization is done randomly, whereas UMAP uses spectral embedding. According to McInnes et al. (2018), this is a major factor that helps UMAP better retain the global structure of the data.

Low Dimensional Similarities

To calculate the low dimensional similarities w_{ij}, t-SNE uses a t-distribution with one degree of freedom on the squared Euclidean distances. UMAP uses a slightly different formula with hyperparameters a and b to modify the t-distribution. Again, t-SNE normalizes the similarities while UMAP does not.

Optimization

The two algorithms optimize different cost functions. t-SNE uses the Kullback-Leibler divergence between the high dimensional and the low dimensional probability distribution and UMAP uses the cross entropy.

The way the cost functions are optimized is also different. t-SNE uses regular gradient descent, while UMAP utilizes stochastic gradient descent, a faster approximation of the gradient descent algorithm. In the low dimensional space, t-SNE moves every point at every iteration, whilst UMAP moves one point at a time.

2.3.2 Empirical Comparison

In the UMAP publication paper, McInnes et al. (2018) compared their new algorithm with a number of other dimension reduction techniques, including t-SNE. They came to the conclusion that, while UMAP and t-SNE retain the local structure equally well, UMAP has several advantages, including the retention of the global structure, the stability under sub-sampling and the run-time. These results make UMAP seem like the clearly superior alternative, but should be regarded with caution, as the supposed superiority of new algorithms over their existing competitors is often over-optimistic, due to the authors being (intentionally or not) biased towards their own algorithm (Ullmann et al., 2022; Buchka et al., 2021). It is therefore important to also look at other, more neutral comparisons, which, in this case, often come to the same conclusions. For example, Becht et al. (2018) claim that UMAP is faster, better reproducible and preserves the global structure better than t-SNE, with Yang et al. (2021) and Wu et al. (2019) coming to similar conclusions.

However, the FIt-SNE implementation by Linderman et al. (2019) solves the run-time deficits of t-SNE, as both Becht et al. (2018) and Kobak and Linderman (2021) agree that it is at least as fast as UMAP. Additionally, Kobak and Linderman (2021) come to the conclusion that UMAP's superiority in terms of reproducibility and preserving the global structure stems only from the initialization. They showed

that when t-SNE is run with an informed initialization, a PCA initialization in this case, it performed as well as UMAP.

Chapter 3

Methods

The aim of this thesis is to examine which parameters affect the quality of the output from t-SNE and UMAP the most, how they affect it and whether one algorithm generally performs better than the other. This Chapter will provide an overview of the methods used to test these questions.

3.1 Parameters

The main idea is to select the most important parameters, choose a few reasonable settings for each parameter and run UMAP/t-SNE for every combination. Unfortunately, due to long computing times, it was necessary to keep the number of total combinations relatively small. For this reason, only three parameters with at most four different settings were selected for each technique. Every other parameter not presented in this section was left at the default setting except max_iter from t-SNE, which was set to 1000, following the recommendations from Kobak and Berens (2019).

3.1.1 t-SNE

Initialization: According to Kobak and Linderman (2021), an informed initialization is the main reason why UMAP is supposedly better able to preserve the global structure of the data than standard t-SNE. To test this, and also to see how the initialization affects other quality measures as well, the two settings used are "pca" and "random".

Perplexity: For large datasets, Kobak and Berens (2019) suggest using a perplexity combination of 30 and $n/100$ to best preserve both the local and global structure. To test the trade-off in preserving the local or global structure the values of 30, $n/100$, and the combination of both were selected.

Learning rate: Kobak and Berens (2019) recommend using n/exaggeration_factor. To test the influence this might have, the standard value of 200 and n/exaggeration_factor $= n/12$ were selected.

3.1.2 UMAP

Initialization: To test whether the supposedly superior preservation is due to an informed initialization, the standard spectral embedding and a random initialization were selected.

Number of nearest neighbors: The default value is 15. There seem to be no recommendations what values to use depending on the size of the dataset. To see how different settings affect the outcome, the values 5, 15, 40 and 100 were selected.

Minimum distance: Again, there seems to be no guideline what settings to use, so the values 0.02, 0.1 (default value) and 0.5 were selected to see how different values affect the result.

3.2 Data Sets

All the combinations of parameter settings for both t-SNE and UMAP were computed for multiple datasets, which are described below:

MNIST: This dataset contains 70000 images of handwritten digits with $28 \times 28 = 784$ pixels. The first 784 columns are the grayscale values of each pixel, ranging from 0 to 255, and the last column is a label indicating which digit is depicted (Deng, 2012).

F-MNIST: This dataset is very similar to MNIST, as it is intended to be a direct replacement. It contains 70000 grayscale images of fashion items with 28×28 pixels. Again, the grayscale value of each pixel, ranging from 0 to 255, is a column and a label column assigns each image to one of ten classes (Xiao et al., 2017).

Statlog (Shuttle): This is a NASA dataset that contains various data about space shuttles. It has 58000 points with 9 numeric attributes and a label column assigning each point to one of 7 classes (Dua and Graff, 2017).

COIL-20: This dataset contains 1440 grayscale images of 20 objects in 72 slightly rotated poses spanning 360 degrees. The value of each of the $128 \times 128 = 16384$ pixels, ranging from 0 to 1, is a column and a label column indicates which of the 20 objects is depicted. (Nene et al., 1996a)

COIL-100: This dataset is similar to COIL-20. It contains 7200 128×128 pixel images of 100 objects in 72 poses. These are color images so the first $16384 \cdot 3 = 49152$ columns are the pixel values from the red, green and blue channels, ranging from 0 to 1. The last column is a label indicating which object is depicted (Nene et al., 1996b).

Olivetti-Faces: This dataset contains 400 images of 40 persons faces in ten different poses with $64 \times 64 = 4096$ pixels. The first 4096 columns are the grayscale values of each pixel, ranging from 0 to 255, and the last column is a label indicating the person (Melville, 2022).

3.3 Quality Measures

To numerically assess the different aspects of quality of the outcomes, quality measures are needed. In this thesis, we used the same measures as recommended by Kobak and Berens (2019).

k-nearest neighbors (KNN): For every point the k-nearest neighbors are computed in the high dimensional original data and the low dimensional embedding. Then, the fraction of points that are neighbors in both spaces is calculated. This is done for every point and then averaged over the whole dataset. As recommended by Kobak and Berens (2019), we set $k = 10$. This is a measure for the preservation of the local or microscopic structure.

k-nearest classes (KNC): For every class the mean is calculated in the high and low dimensional space. Then, the fraction of the k-nearest class means that are the same in the high and low dimensional space is computed for every class and then averaged over all classes. As done by Kobak and Berens (2019), we set k to 25% of the total number of classes for each dataset. This is a measure for the preservation of the mesoscopic structure.

Correlation between pairwise distances (CPD): 1000 points are sampled randomly. Then, the spearman correlation between the pairwise distances in the high dimensional and low dimensional space is computed. This is done 10 times and then averaged. It is a measure for the preservation of the global or macroscopic structure.

3.4 Analysis

3.4.1 Linear Models

To analyze the influence of the parameters on the quality of the outcome, we used linear models on the data generated by running t-SNE/UMAP with all the parameter combinations mentioned above on all datasets and calculating the three quality measures. We computed a linear model for each quality measure where it was the dependent variable and the parameter settings and dataset used in each instance were the independent variables. For t-SNE, such a model looks like this:

$$\begin{aligned}\text{quality\_measure}_i = \beta_0 &+ \beta_1 \text{initialization\_random}_i + \beta_2 \text{perplexity\_combined}_i \\ &+ \beta_3 \text{perplexity\_n/100}_i + \beta_4 \text{learning\_rate\_n/12}_i \\ &+ \beta_5 \text{dataset\_coil-20}_i + \beta_6 \text{dataset\_f-mnist}_i \\ &+ \beta_7 \text{dataset\_olivetti}_i + \beta_8 \text{dataset\_mnist}_i + \beta_9 \text{dataset\_shuttle}_i\end{aligned}$$

and for UMAP like this:

$$\begin{aligned}\text{quality\_measure}_i = \beta_0 &+ \beta_1 \text{initialization\_random}_i + \beta_2 \text{n\_neighbors\_5}_i \\ &+ \beta_3 \text{n\_neighbors\_40}_i + \beta_4 \text{n\_neighbors\_100}_i + \beta_5 \text{min\_dist\_0.02}_i \\ &+ \beta_6 \text{min\_dist\_0.5}_i + \beta_7 \text{dataset\_coil-20}_i + \beta_8 \text{dataset\_f-mnist}_i \\ &+ \beta_9 \text{dataset\_olivetti}_i + \beta_{10} \text{dataset\_mnist}_i + \beta_{11} \text{dataset\_shuttle}_i\end{aligned}$$

for each of the three quality measures.

Since some parameter settings are dependent on the size of the dataset, it makes sense to divide the datasets into "big" and "small" ones and analyze them separately. This was done by categorizing all datasets with $n/100 > 30$ as big and the rest as small, with $n/100$ being one setting for the perplexity value and 30 being the default value. In addition to the models given above, we computed similar models using only the big or small datasets respectively.

The parameters were coded as factors, with the default setting as the reference category. For the datasets, COIL-100 was used as the reference category, except when using only the small datasets, where COIL-20 was used.

For each of the different models, an analysis of variance (ANOVA) was used to calculate how much each variable contributes to the total variance.

3.4.2 Density Plots

To compare the overall performance we used overlayed density plots of t-SNE and UMAP for each quality measure, with a vertical line denoting the average for each technique. To create these plots we used the "geom_density" function with default settings from the "ggplot2" package by Wickham (2016).

3.4.3 Box Plots

For a more accurate comparison of the two techniques we used box plots. For each quality measure we constructed a graphic with box plots for t-SNE and UMAP for every dataset. A star in every box plot marks the value obtained using the default parameter settings.

3.4.4 Scatter Plots

Since t-SNE and UMAP are primarily used for visualization, it is important to examine the visualizations they produce. As an overview, we created a scatter plot of the best and worst embedding from t-SNE and UMAP for each dataset.

Chapter 4

Results

In this chapter, the analysis results will be presented, starting with the microscopic or local structure.

4.1 Local Structure

In this section, we will examine how well the different embeddings were able to retain the local structure of the data, as measured by the KNN. First, with the help of the linear models describe above, we will analyze how the different parameter settings influenced the retention of the local structure. Then, we will compare the results of the KNN values via density and box plots.

Linear Models

For the sake of brevity, we will only show some of the linear models here. The rest can be found in the Appendix A.

It has to be noted that the models, due to the artificial data generation process, may not meet the assumptions for linear regression, so the p-values should only be interpreted with great caution. That applies for all the linear models covered in this or other sections. The diagnostic plots for all models can be found in the Appendix A.

t-SNE

Since some parameters depend on the size of the dataset it makes more sense to focus on the models using either only the big or small datasets, instead of the models containing all datasets.

Table 4.1 shows the variance decomposition for all the t-SNE models with KNN as the dependent variable. It shows clearly that the initialization had no effect on the retention of the local structure. The variance explained by the perplexity is relatively small and the variance explained by the learning rate is even smaller. By far the largest part of the variance can be explained by the dataset, with values over 90% for all three models. The residual part of the variance was relatively small.

Source	KNN		
	all	**big**	**small**
initialization	0.00	0.00	0.02
perplexity	1.95	4.53	0.32
learning rate	1.06	2.32	0.02
dataset	93.62	91.17	99.36
residual	3.38	1.98	0.28

Table 4.1: Variance decomposition (percent values) for the t-SNE KNN models

KNN - big datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.809	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	0.000	0.986
$\text{perplexity}_{\text{combined}}$	-0.032	0.029
$\text{perplexity}_{n/100}$	-0.128	$2.18 \cdot 10^{-11}$
$\text{learning\_rate}_{n/12}$	0.078	$3.12 \cdot 10^{-8}$
$\text{dataset}_{\text{F-MNIST}}$	-0.580	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{MNIST}}$	-0.572	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{shuttle}}$	-0.235	$< 2 \cdot 10^{-16}$

Table 4.2: Estimates of the coefficients and p-values of the linear model for t-SNE with KNN as the dependent variable using only the big datasets

Table 4.2 shows the coefficient estimates and p-vales of the linear model for the local structure using only the big datasets. Especially the effect of the perplexity setting is of interest here. A perplexity of $n/100$ and a combined perplexity both influenced the retention of the local structure negatively, with the $n/100$ setting having a much stronger negative effect A learning rate of $n/12$ helped to better retain the local structure of the data.

KNN - small datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.807	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.003	0.320
$\text{perplexity}_{\text{combined}}$	-0.006	0.117
$\text{perplexity}_{n/100}$	0.011	0.012
$\text{learning\_rate}_{n/12}$	0.004	0.246
$\text{dataset}_{\text{Olivetti}}$	-0.247	$< 2 \cdot 10^{-16}$

Table 4.3: Estimates of the coefficients and p-values of the linear model for t-SNE with KNN as the dependent variable using only the small datasets

Table 4.3 shows the results from the model using only the small datasets, where $n/100 < 30$. One could reasonably assume the perplexity settings to have the opposite effect of the KNN model for big datasets. The perplexity however, as well as all other parameters, have almost no effect in this model.

UMAP

Since, in contrast to the t-SNE models, the parameter settings do not depend on the size of the dataset, we will focus on the models using all datasets when examining the specific coefficient estimates.

Source	KNN		
	all	**big**	**small**
initialization	0.05	0.04	0.32
n_neighbors	5.57	5.28	38.83
min_dist	0.97	1.68	1.62
dataset	89.71	87.40	54.21
residual	3.70	5.59	5.03

Table 4.4: Variance decomposition (percent values) for the UMAP KNN models

Table 4.4 shows the variance decomposition for all UMAP models with KNN as the dependent variable. Similarly to the t-SNE models, by far the largest part of the variance in most models is explained by the dataset. As expected, the initialization did not influence the retention of the local structure. The number of neighbors had a small to medium sized effect on the KNN, except in the model with the small datasets, where the number of neighbors explained a big part of the variance. Again, the datasets explained most of the variance, although it was much less in the model with the small datasets.

KNN - all datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.595	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.009	0.201
n_neighbors_{100}	-0.127	$< 2 \cdot 10^{-16}$
n_neighbors_{40}	-0.077	$2.40 \cdot 10^{-11}$
n_neighbors_{5}	-0.007	0.543
$\text{min\_dist}_{0.02}$	0.003	0.723
$\text{min\_dist}_{0.5}$	-0.045	$2.61 \cdot 10^{-6}$
$\text{dataset}_{\text{COIL-20}}$	0.123	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{F-MNIST}}$	-0.422	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{Olivetti}}$	-0.029	0.026
$\text{dataset}_{\text{MNIST}}$	-0.434	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{shuttle}}$	-0.110	$3.48 \cdot 10^{-14}$

Table 4.5: Estimates of the coefficients and p-values of the linear model for UMAP with KNN as the dependent variable using all datasets

Table 4.5 shows the coefficient estimates and p-vales of the UMAP model for the local structure. The number of neighbors setting of 100 had a relatively big negative influence on the retention of the local structure. The effect of considering 40 nearest neighbors was still negative, but not as big. Lowering the setting even more to a value of 5, compared to the reference category of 15, had almost no effect. The

minimum distance setting of 0.5 had a small negative influence on the retention of the local structure compared to the other two settings of 0.02 and 0.1.

Plots

Next, we will compare the performance of t-SNE and UMAP regarding the retention of the local structure of the data with the help of different plots.

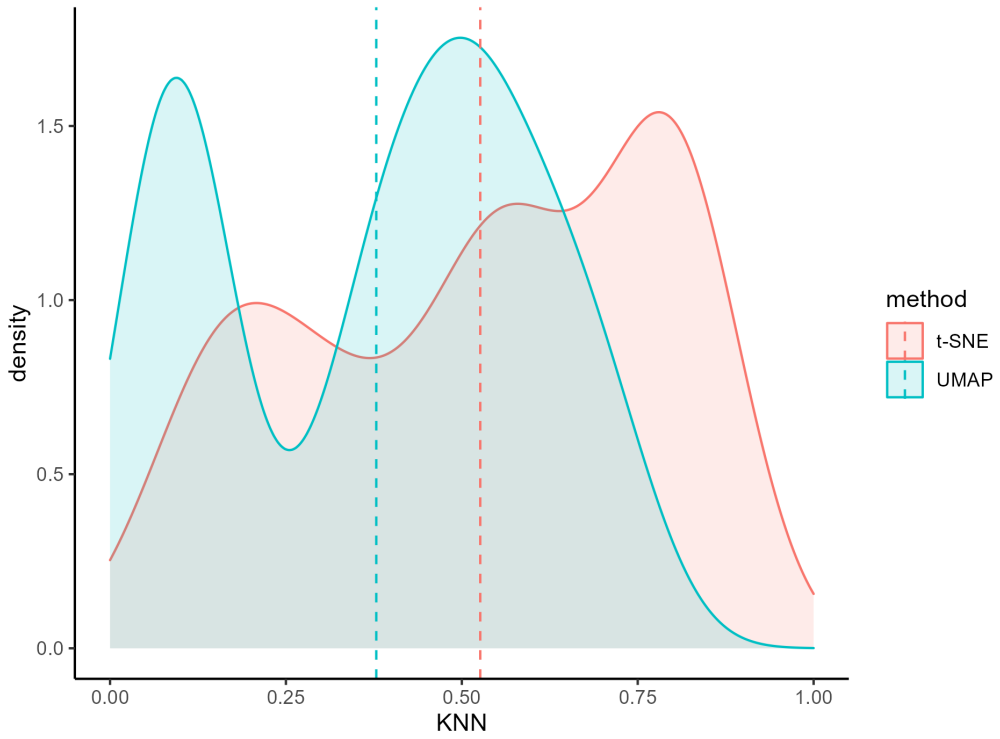

Figure 4.1: Density plot of the KNN values for t-SNE and UMAP

Figure 4.1 shows the KNN density plots for t-SNE and UMAP. Both distributions are multimodal, which is likely due to the fact that the values are heavily influenced by the different datasets, as seen in the linear models shown above. Generally, t-SNE was able to retain the local structure better than UMAP. The mean KNN value is notably higher, and it is clear to see that the t-SNE density is higher for especially large and lower for very small KNN values. The UMAP density curve also flattens noticeably sooner, so the largest KNN values are only reached by t-SNE embeddings.

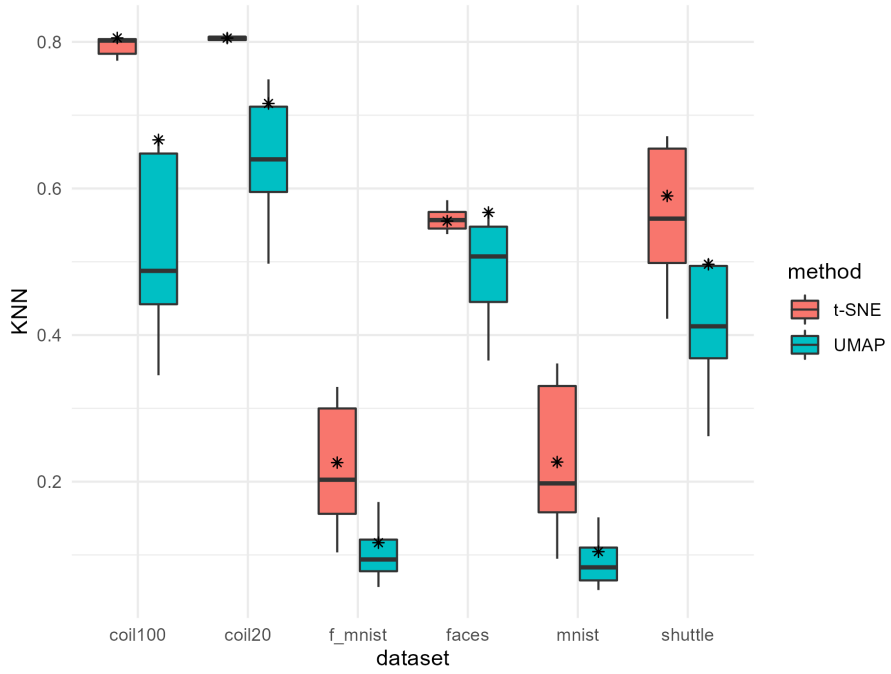

Figure 4.2: Box plots of the KNN values for every dataset for t-SNE and UMAP

Figure 4.2 shows the box plots of the KNN values of the t-SNE and UMAP embeddings for each dataset, with a star marking the value from the embedding where the default settings were used. The median of the t-SNE embeddings is higher than the UMAP median in every dataset. The maximum value obtained by t-SNE is also larger than the best UMAP value in all datasets. The size of the boxes varies massively between the different datasets. In two datasets, COIL-100 and COIL-20, the plots do not overlap, meaning that no matter what parameter settings were used, the t-SNE embeddings were always superior regarding the KNN. In five of the six datasets, the t-SNE embeddings using the default settings achieved a better KNN value than the UMAP default embeddings. All things considered, it can be said that in our analysis, t-SNE was better able to retain the local structure of the data as measured by the KNN.

4.2 Mesoscopic Structure

In this section, we will examine the retention of the mesoscopic structure of the data, as measured by the KNC, beginning with the linear models and then comparing the performance of t-SNE and UMAP with different plots.

Linear Models

As in the last section, we will not show all the models here. All the models not shown in the main part of this thesis can be found in the Appendix A.

t-SNE

For t-SNE, we will examine the models using only the big or only the small datasets.

Source	KNC		
	all	**big**	**small**
initialization	5.04	6.96	1.26
perplexity	2.08	4.98	36.63
learning rate	0.11	0.12	0.08
dataset	67.52	70.03	41.07
residual	25.25	17.91	20.96

Table 4.6: Variance decomposition (percent values) for the t-SNE KNC models

Table 4.1 shows the variance decomposition for all the t-SNE models with KNC as the dependent variable. The initialization had a small to medium sized effect on the retention of the mesoscopic structure. The influence of the perplexity is relatively small for the models containing all or only the big datasets, but much larger for the model containing only the small datasets. The learning rate did not influence the retention of the mesoscopic structure, explaining less than 1% of the variance in all three models. As with the KNN, the datasets also explained a lot of the variance of the KNC. The residual part of the variance was also relatively big.

KNC - big datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.614	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.086	$3.16 \cdot 10^{-4}$
$\text{perplexity}_{\text{combined}}$	0.077	0.006
$\text{perplexity}_{n/100}$	0.077	0.007
$\text{learning\_rate}_{n/12}$	-0.011	0.602
$\text{dataset}_{\text{F-MNIST}}$	0.186	$4.08 \cdot 10^{-7}$
$\text{dataset}_{\text{MNIST}}$	0.072	0.024
$\text{dataset}_{\text{shuttle}}$	-0.188	$3.22 \cdot 10^{-7}$

Table 4.7: Estimates of the coefficients and p-values of the linear model for t-SNE with KNC as the dependent variable using only the big datasets

As we can see in Table 4.7, a PCA initialization helped to better retain the mesoscopic structure of the data. A perplexity of $n/100$ and a combined perplexity both had a positive effect on the retention of the mesoscopic structure for the big datasets. The learning rate did not have any noteworthy influence.

Table 4.8 shows the results from the t-SNE model for the small datasets with KNC as the dependent variable. Surprisingly, the initialization had a much smaller effect in this model than in the KNC model for the big datasets. A PCA initialization still had a positive effect on the retention of the mesoscopic structure, but it is relatively small. A combined perplexity had almost no effect, while a perplexity of $n/100$ negatively influenced the KNC. Again, the learning rate had almost no effect.

KNC - small datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.744	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.016	0.313
$\text{perplexity}_{\text{combined}}$	-0.008	0.677
$\text{perplexity}_{n/100}$	-0.097	$8.22 \cdot 10^{-5}$
$\text{learning\_rate}_{n/12}$	0.004	0.793
$\text{dataset}_{\text{Olivetti}}$	-0.093	$1.28 \cdot 10^{-5}$

Table 4.8: Estimates of the coefficients and p-values of the linear model for t-SNE with KNC as the dependent variable using only the small datasets

UMAP

Source	**KNC**		
	all	**big**	**small**
initialization	12.50	16.45	5.56
n_neighbors	13.57	4.28	64.41
min_dist	0.05	0.02	0.16
dataset	37.18	47.08	5.34
residual	36.71	32.17	24.53

Table 4.9: Variance decomposition (percent values) for the UMAP KNC models

Table 4.9 shows the variance decomposition of all UMAP models with KNC as the dependent variable. The initialization had a relatively big influence on the retention of the mesoscopic structure for the models containing all and only the big datasets, and a much smaller influence for the model containing only the small datasets. The proportion of the variance explained by the number of neighbors parameter varies greatly between the models, ranging from about 4% for the model with only big datasets to 64% for the model with only the small datasets. The datasets explain a big proportion of the total variance except for the model containing only the small datasets. The residual part of the variance was also very big for all models.

As shown in Table 4.10, using spectral embedding as initialization positively influenced the retention of the mesoscopic structure. The number of nearest neighbors parameter also had a big effect on the KNC. A setting of 100 neighbors had the biggest positive effect, 40 had a smaller positive effect and the setting of 5 nearest neighbors had a negative effect on the retention of the mesoscopic structure. The minimum distance parameter had almost no influence.

KNC		
Coefficient	**Estimate**	**P-Value**
Intercept	0.488	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.124	$5.37 \cdot 10^{-10}$
n_neighbors_{100}	0.108	$6.49 \cdot 10^{-5}$
n_neighbors_{40}	0.088	0.001
n_neighbors_{5}	-0.051	0.053
$\text{min\_dist}_{0.02}$	-0.006	0.781
$\text{min\_dist}_{0.5}$	0.003	0.891
$\text{dataset}_{\text{COIL-20}}$	0.089	0.007
$\text{dataset}_{\text{F-MNIST}}$	0.263	$2.24 \cdot 10^{-13}$
$\text{dataset}_{\text{Olivetti}}$	0.159	$2.28 \cdot 10^{-6}$
$\text{dataset}_{\text{MNIST}}$	0.161	$1.69 \cdot 10^{-6}$
$\text{dataset}_{\text{shuttle}}$	-0.059	0.068

Table 4.10: Estimates of the coefficients and p-values of the linear model for UMAP with KNC as the dependent variable using all datasets

Plots

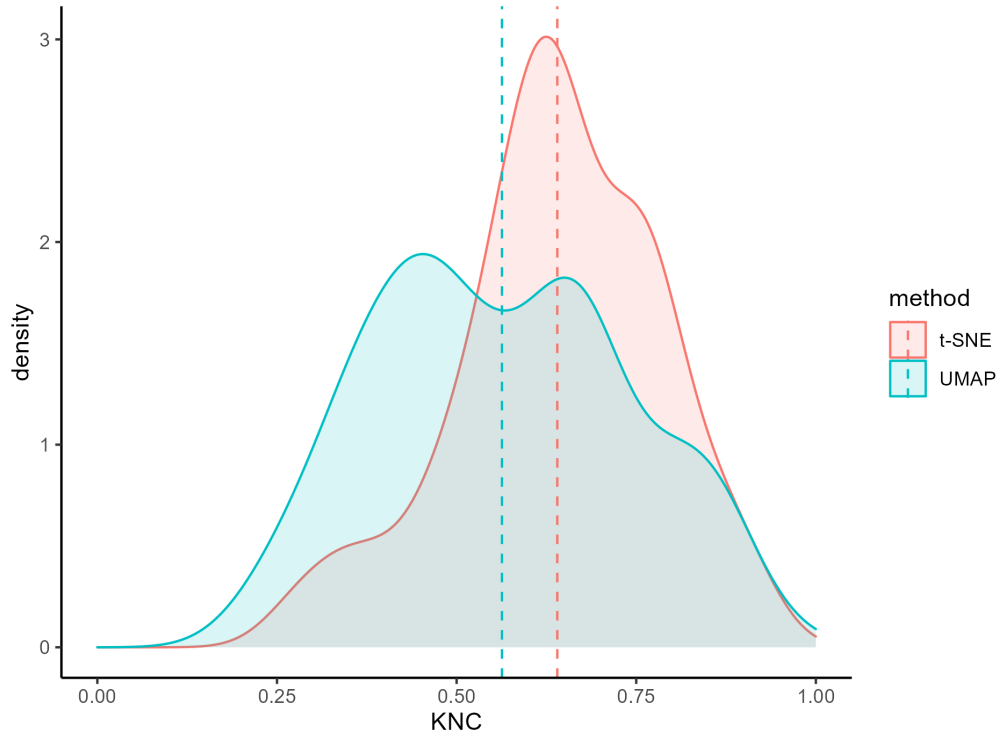

Figure 4.3: Density plot of the KNC values for t-SNE and UMAP

Figure 4.3 shows the KNC density plots. Again, the two distributions are multimodal. Both curves cover approximately the same area, so there is no range of high KNC values that only one method was able to achieve. However, the t-SNE density is much higher than the UMAP density for the bigger KNC values and lower for the small values. The mean KNC value for all t-SNE embeddings is higher than the

mean for UMAP. Considering all of this, it can be said that t-SNE generally performed better than UMAP regarding the preservation of the mesoscopic structure.

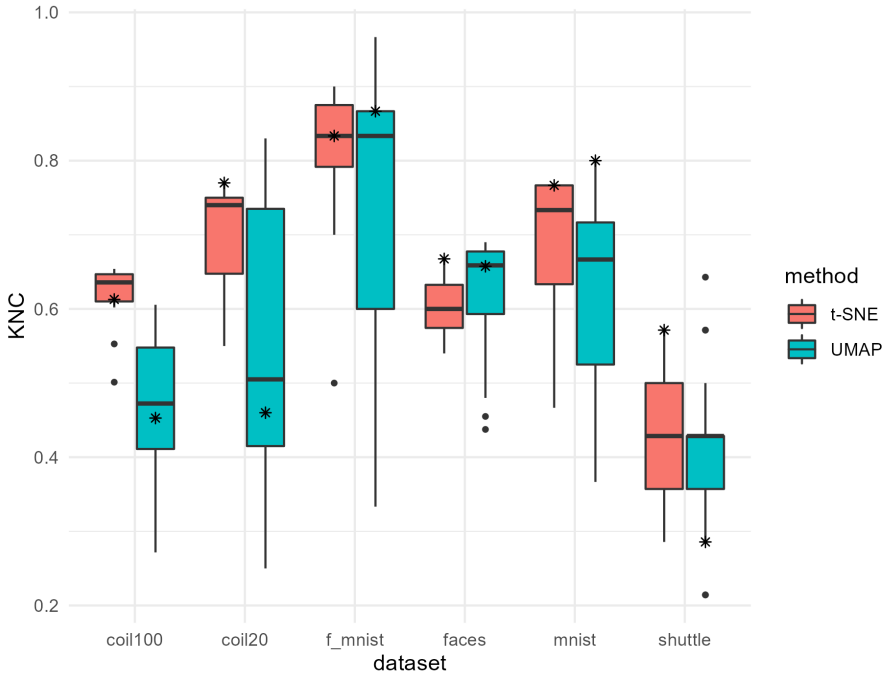

Figure 4.4: Box plots of the KNC values for every dataset for t-SNE and UMAP

Figure 4.4 shows the KNC box plots. In three of the six datasets, the t-SNE median is higher, in two datasets, the t-SNE and UMAP median are the same and only in the Olivetti Faces dataset, the UMAP median is higher. The maximum value, however, is achieved by UMAP embeddings in five datasets, while COIL-100 is the only dataset, where the best KNC value stems from a t-SNE embedding. The plots overlap for all datasets. Using the default settings, t-SNE performed better in four datasets and UMAP in two. In contrast to the density plots, in this graphic, neither t-SNE nor UMAP are shown to clearly better retain the mesoscopic structure.

4.3 Global Structure

In this section we will examine the retention of the global, or macroscopic structure of the data, measured by the CPD.

Linear Models

Again, for the sake of brevity, we will not discuss all linear models here. The models not shown in this section can be found in the Appendix A.

t-SNE

Source	CPD		
	all	**big**	**small**
initialization	4.88	6.61	2.37
perplexity	9.31	27.46	69.98
learning rate	0.23	0.04	2.38
dataset	45.15	47.01	10.16
residual	40.43	18.88	15.12

Table 4.11: Variance decomposition (percent values) for the t-SNE CPD models

Table 4.11 shows the variance decomposition for all the t-SNE models with CPD as the dependent variable. The initialization had a small to medium sized influence on the retention of the global structure. The effect of the perplexity is bigger and varies massively between the models. The learning rate explained a small amount of the variance in the model using only the small datasets and had almost no effect in the other models. Compared to the other models, the datasets explained surprisingly little of the variance of the model using only the small datasets. The residual part of the variance was relatively big for all models, especially for the model using all datasets.

CPD - big datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.364	$5.22 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.074	$5.74 \cdot 10^{-4}$
$\text{perplexity}_{\text{combined}}$	0.157	$8.80 \cdot 10^{-8}$
$\text{perplexity}_{n/100}$	0.161	$5.12 \cdot 10^{-8}$
$\text{learning\_rate}_{n/12}$	0.006	0.772
$\text{dataset}_{\text{F-MNIST}}$	0.197	$1.48 \cdot 10^{-8}$
$\text{dataset}_{\text{MNIST}}$	-0.064	0.028
$\text{dataset}_{\text{shuttle}}$	0.093	0.002

Table 4.12: Estimates of the coefficients and p-values of the linear model for t-SNE with CPD as the dependent variable using only the big datasets

As we can see in Table 4.12, a PCA initialization positively influenced the retention of the global structure. The perplexity also had a big influence, with the "combined" and the "$n/100$" setting having a similar positive effect on the CPD for the big datasets. The learning rate did not influence the preservation of the global structure.

The coefficient estimates for the CPD model using only the small datasets are shown in Table 4.13. Compared to a random initialization, a PCA initialization had a small positive influence on the retention of the global structure. As expected, a small perplexity of $n/100$ negatively influenced the preservation of the global structure, while a combined perplexity had a much smaller negative effect. A learning rate of $n/12$

had a small positive effect in the CPD.

CPD - small datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.594	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.026	0.111
$\text{perplexity}_{\text{combined}}$	-0.023	0.245
$\text{perplexity}_{n/100}$	-0.158	$1.14 \cdot 10^{-7}$
$\text{learning\_rate}_{n/12}$	0.026	0.110
$\text{dataset}_{\text{Olivetti}}$	0.053	0.003

Table 4.13: Estimates of the coefficients and p-values of the linear model for t-SNE with CPD as the dependent variable using only the small datasets

UMAP

Source	**CPD**		
	all	**big**	**small**
initialization	10.64	20.05	2.41
n_neighbors	11.73	4.40	36.64
min_dist	0.26	0.21	0.47
dataset	54.19	54.99	48.65
residual	23.18	20.35	11.83

Table 4.14: Variance decomposition (percent values) for the UMAP CPD models

Table 4.14 shows the variance decomposition for all UMAP models with CPD as the dependent variable. The influence of the initialization on the retention of the global structure of the data varies greatly, explaining only about 2% of the variance in the model using only the small datasets, while explaining about 20% in the model using the big datasets. The number of nearest neighbors parameter had a big effect on the CPD in the model containing only the small datasets, but only a small effect in the model using only the big datasets. The datasets explained about half of the variance in all models and the residual part of the variance was also relatively big in all models.

As shown in Table 4.15, an informed initialization had a big positive influence on the retention of the global structure. The number of nearest neighbors parameter also had a big influence. The more neighbors were considered, the better the global structure of the data was preserved. The minimum distance parameter had almost no effect.

CPD		
Coefficient	**Estimate**	**P-Value**
Intercept	0.167	$7.78 \cdot 10^{-8}$
$\text{initialization}_{\text{random}}$	-0.132	$1.79 \cdot 10^{-12}$
n_neighbors_{100}	0.153	$2.40 \cdot 10^{-9}$
n_neighbors_{40}	0.095	$1.15 \cdot 10^{-4}$
n_neighbors_{5}	-0.015	0.536
$\text{min\_dist}_{0.02}$	-0.007	0.732
$\text{min\_dist}_{0.5}$	0.017	0.409
$\text{dataset}_{\text{COIL-20}}$	0.125	$3.99 \cdot 10^{-5}$
$\text{dataset}_{\text{F-MNIST}}$	0.327	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{Olivetti}}$	0.441	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{MNIST}}$	0.142	$3.49 \cdot 10^{-6}$
$\text{dataset}_{\text{shuttle}}$	0.313	$< 2 \cdot 10^{-16}$

Table 4.15: Estimates of the coefficients and p-values of the linear model for UMAP with CPD as the dependent variable using all datasets

Plots

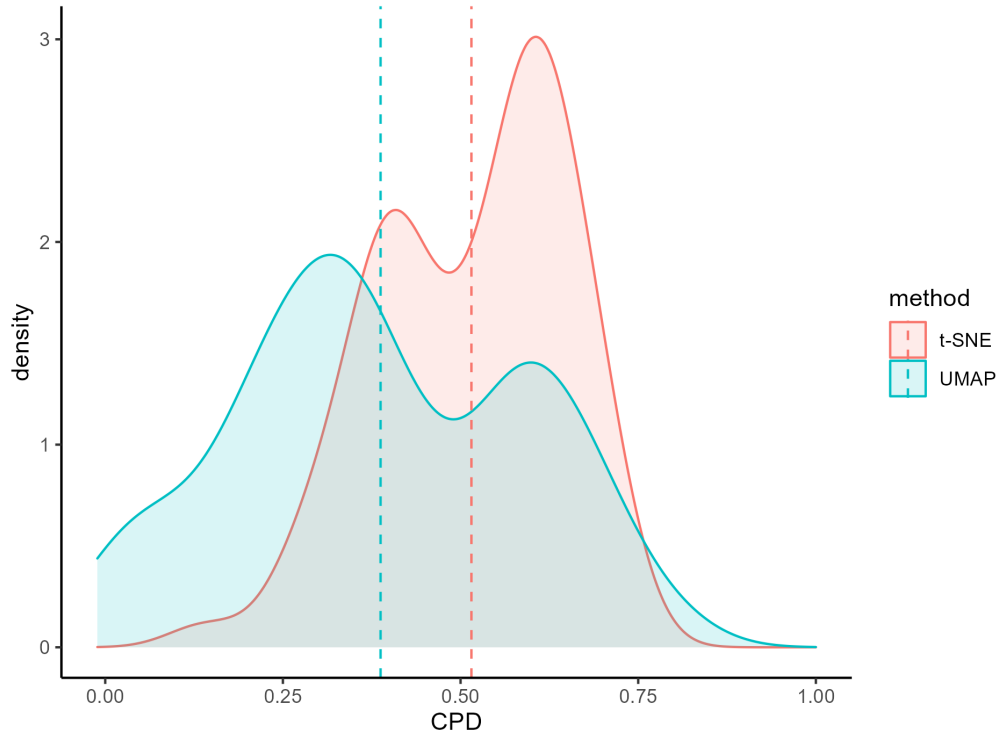

Figure 4.5: Density plot of the CPD values for t-SNE and UMAP

Figure 4.5 shows the CPD density plots for t-SNE and UMAP. The mean CPD value of the t-SNE embeddings is higher than the UMAP mean and the t-SNE density is generally higher for big and lower for small CPD values. However, it has to be mentioned that the UMAP density is a bit higher for the biggest CPD values. All things considered, it can still be said that t-SNE was generally able to better pre-

serve the global structure of the data.

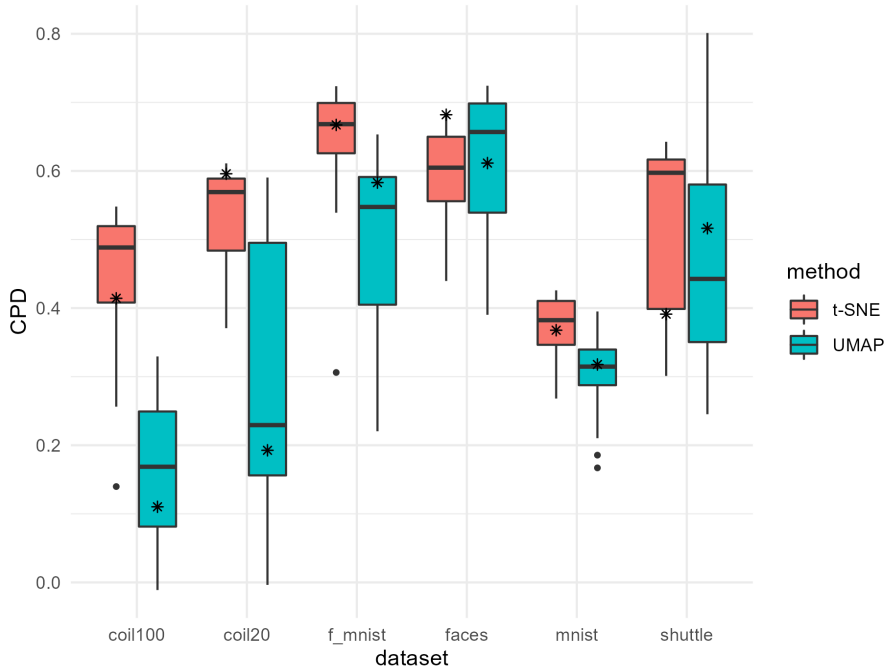

Figure 4.6: Box plots of the CPD values for every dataset for t-SNE and UMAP

Figure 4.6 shows the CPD box plots. In five datasets, the median CPD value is higher for the t-SNE embeddings, with the Olivetti Faces dataset being the only exception where the UMAP median is higher. The maximum value in four out of the six datasets comes from a t-SNE embedding. The Shuttle dataset was the only one where the UMAP default settings performed better than the t-SNE defaults. In the other five datasets the t-SNE embeddings using default settings had better CPD scores. Overall, t-SNE performed better in our analysis regarding the CPD measure, meaning it was better able to retain the global structure of the data.

4.4 Scatter Plots

For the sake of brevity, we will only examine the plots from the MNIST and COIL-20 dataset in this section. The plots for the other datasets can be found in the Appendix A.

4.4.1 MNIST

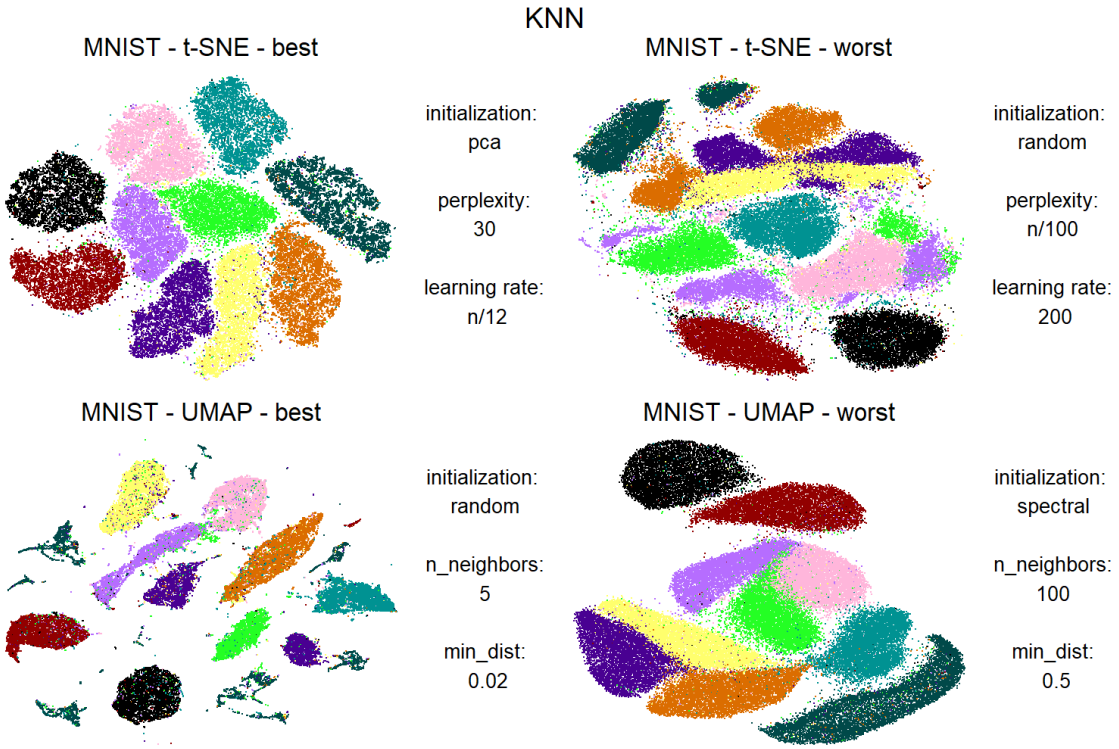

Figure 4.7: Scatter plots of the t-SNE/UMAP embedings of the MNIST dataset with the best and worst KNN values

Figure 4.7 shows the visualizations of the best and worst MNIST embeddings from t-SNE and UMAP regarding the retention of the local structure. In all four plots, the different digits form clusters, as indicated by the color of the points. In both t-SNE plots, the clusters are relatively large and close together, with little space in between. In the best t-SNE plot, with a smaller perplexity, there are ten clear clusters, one for every digit. The partition of clusters is not as perfect in the worst t-SNE plot, where sometimes multiple different clusters for the same digit exist.
The best UMAP plot has small dense clusters with much more space between the clusters compared to the t-SNE plots. The number of nearest neighbors setting of 5 explains the focus an the local structure and good KNN result but also leads to multiple separated clusters for some digits. Unsurprisingly, the worst UMAP embedding regarding the KNN used a number of neighbors setting of 100, focusing more on the global structure. This plot resembles the t-SNE plots more with ten bigger clusters that are closer together, which is most likely due to the high min_dist setting.

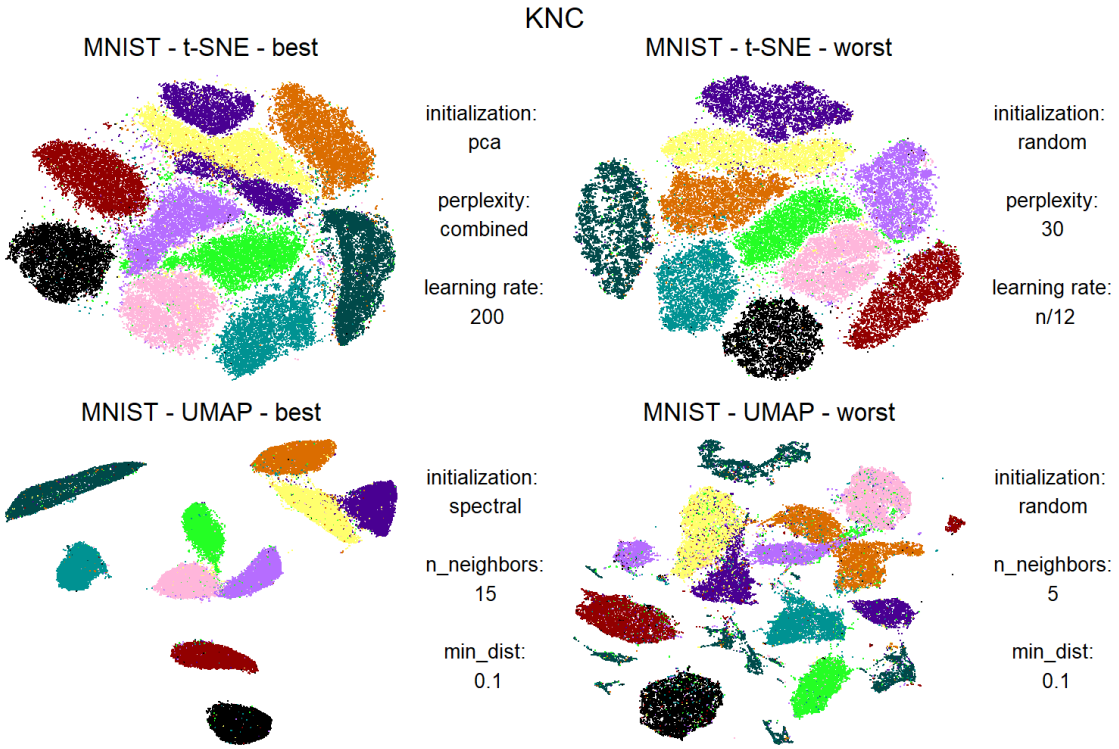

Figure 4.8: Scatter plots of the t-SNE/UMAP embedings of the MNIST dataset with the best and worst KNC values

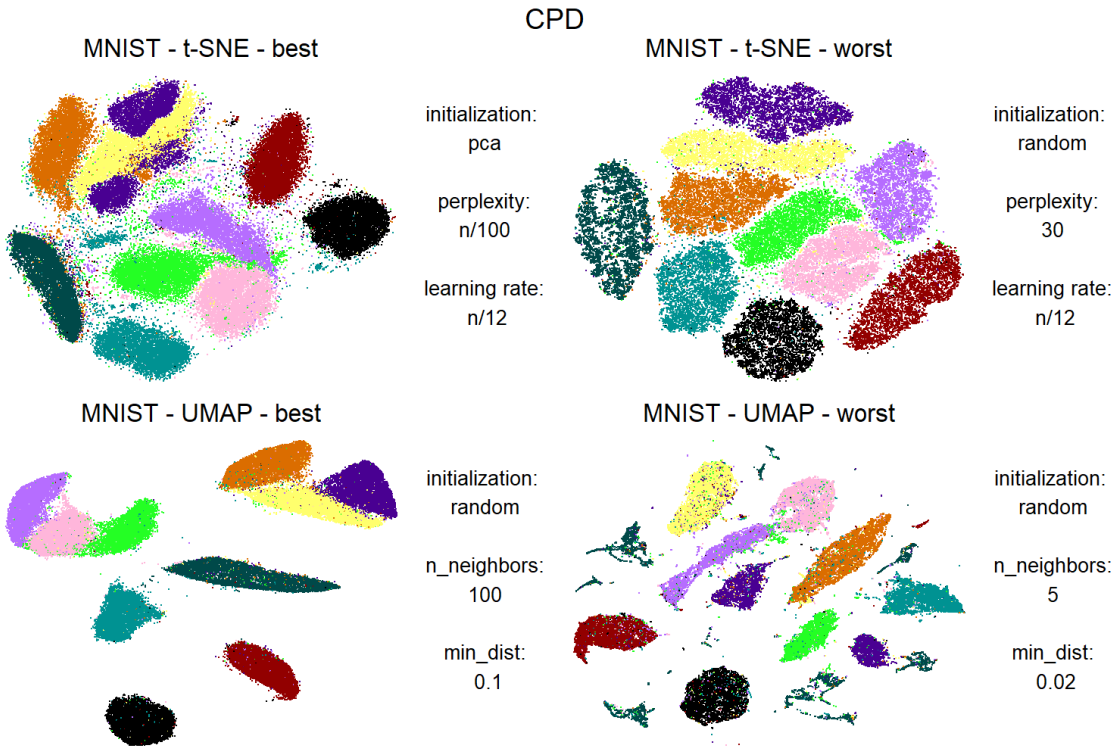

Figure 4.9: Scatter plots of the t-SNE/UMAP embedings of the MNIST dataset with the best and worst CPD values

Figure 4.8 shows the best and worst embeddings regarding the retention of the mesoscopic structure. The t-SNE plots look fairly similar, with big clusters that are close together. There is one cluster for every digit in the plot with the worst KNC, while the best plot is not as clean, with one digit being split across two clusters. The settings that lead to the best and worst KNC values are not surprising but seem to have little effect on the visuals.
The best UMAP plot has small dense clusters, which are mostly clearly separated but some stick closer together. There are only ten clusters, one for every digit. The worst UMAP embedding used a number of nearest neighbors setting of 5, which again lead to some digits being divided into multiple small clusters.

Figure 4.9 shows the MNIST embeddings with the best and worst retention of the global structure. The worst t-SNE embedding is the same as for the KNC. For some digits, the best t-SNE plot has multiple clusters. There are also more regions between the clusters, where many colorful points representing different digits are located. Otherwise it has the same characteristics as all t-SNE plots, with mostly bigger clusters that are relatively close together.
The plot of the best UMAP embedding looks similar to the best UMAP plot regarding the KNC, with some clusters sticking together and some being very isolated. The UMAP embedding with the worst CPD is the same as the one with the best KNN.

4.4.2 COIL-20

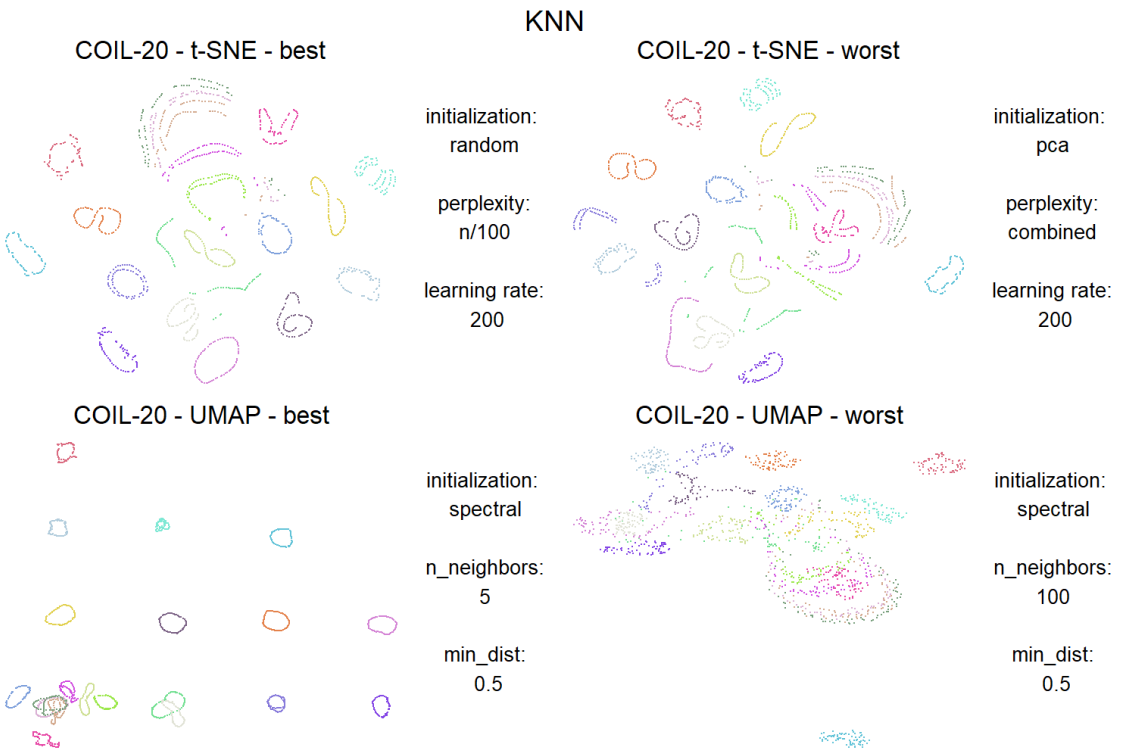

Figure 4.10: Scatter plots of the t-SNE/UMAP embedings of the COIL-20 dataset with the best and worst KNN values

Figure 4.10 shows the scatter plots of the best and worst embeddings of the COIL-20 dataset regarding the preservation of the local structure. The twenty different objects are marked by color and form mostly well separated clusters in the shape of rings or lines. Both t-SNE plots look extremely similar, with one bigger cluster where the points from three different objects form a big semi-circle and otherwise well separated clusters.
In the best UMAP embedding, the different ring-clusters are arranged like a grid and only in the bottom left corner multiple clusters share a space and overlap. The worst UMAP embedding looks noticeably different from all other plots. The points are all close together and most clusters do not have the distinct ring shape seen in the other plots.

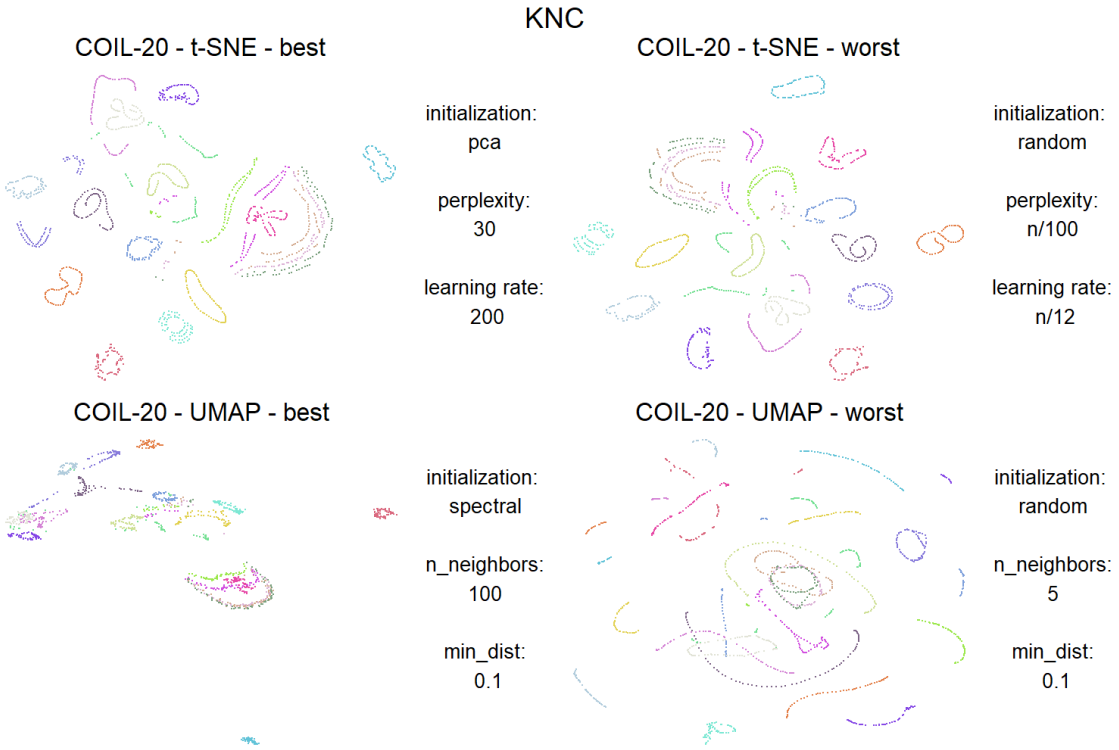

Figure 4.11: Scatter plots of the t-SNE/UMAP embedings of the COIL-20 dataset with the best and worst KNC values

Figure 4.11 shows the scatter plots for the mesoscopic structure. Again, both t-SNE plots look extremely alike and show essentially the same characteristics as both KNN plots shown in Figure 4.10.
In the best UMAP embedding, most clusters clump together in the upper left corner of the plot. Many clusters have the line or semi-circle shape seen in the other COIL-20 plots, but no clusters appear in the ring shape that is also typical for this dataset. The plot of the worst UMAP embedding looks more fractured than the other plots. The clusters take the typical line and ring shape and most are well separated, but many objects are split into multiple small clusters.

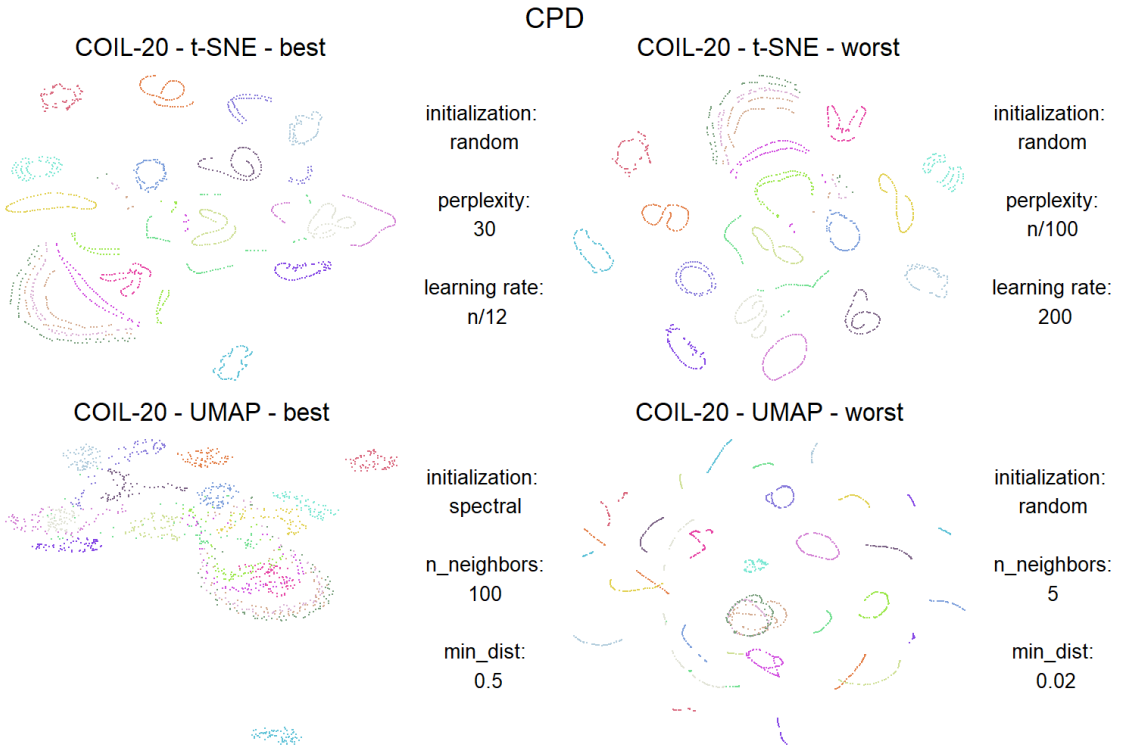

Figure 4.12: Scatter plots of the t-SNE/UMAP embedings of the COIL-20 dataset with the best and worst CPD values

Figure 4.12 shows the best and worst COIL-20 embeddings regarding the retention of the global structure. The worst t-SNE embedding was the best regarding the KNN. The plot of the best t-SNE embedding looks a lot like all the other t-SNE plots for the COIL-20 dataset, with the well separated ring and line clusters and the same semi-circle where three clusters clump together.
The best UMAP embedding regarding the CPD is also the worst in terms of the KNN, which we have already discussed earlier. The worst CPD plot looks similar to the worst KNC plot, with many small clusters in the typical line or ring shape.

Chapter 5

Discussion

In this thesis we examined the two dimension reduction techniques t-SNE and UMAP, especially focusing on the effects of different parameter settings on three quality measures across multiple datasets as well as comparing the overall performance of the two methods.

For both t-SNE and UMAP an informed initialization (PCA for t-SNE and Spectral Embedding for UMAP) helped to better retain the mesoscopic and global structure (higher KNC and CPD values), while not affecting the retention of the local structure (no effect on the KNN). The other two t-SNE parameters we examined were the perplexity and the learning rate. In the big datasets, a large perplexity value of $n/100$ helped to better retain the global structure of the data at the cost of the local structure. A combined perplexity of 30 and $n/100$ had almost the same benefits as the $n/100$ setting with a much smaller negative influence on the retention of the local structure. For the small datasets, a reduction in the perplexity to $n/100$ only had a very small positive effect on the preservation of the local structure but worsened the retention of the mesoscopic and global structure by a large amount. A combined perplexity had no effect on the retention of the local structure. The learning rate only had very small to no effects most of the time except for the big datasets, where a larger learning rate of $n/12$ positively influenced the retention of the local structure. Overall, a PCA initialization is always preferable to a random initialization, while the optimal setting of the perplexity depends on whether the retention of the local or global structure is of more importance. A perplexity combination of 30 and $n/100$, as recommended by Kobak and Linderman (2021), seems like a good middle ground. Since the setting of the learning rate had almost no effect most of the time, it is not possible to give a recommendation.

Similar to the perplexity value from t-SNE, the number of nearest neighbors UMAP parameter is a trade-off, where higher values better capture the global structure at the cost of the local structure. There is no comparable setting to the combined perplexity that helps to better retain the global structure without sacrificing too much of the local structure. The minimum distance parameter had almost no effect on the quality measures except that a very large value negatively influenced the retention of the local structure. Again, an informed initialization only has advantages over a random initialization, so the spectral embedding initialization should always be used. A recommendation for the optimal number of neighbors is not possible, as it

depends strongly on whether the focus is more on the local or global structure of the data. Although it had almost no effect on the quality measures, the minimum distance parameter is still important, since it has a big influence on the resulting visualizations. It is not possible to recommend one value here, but it could be helpful to run UMAP multiple times with different minimum distance settings to see which one looks best in a given case.

The density plots show that t-SNE generally performed better than UMAP for all three quality measures in our analysis. This is especially interesting, since it was claimed by Becht et al. (2018) that UMAP generally outperforms t-SNE in terms of preserving the global structure of the data. The box plots give a more detailed overview and show that the performances varied greatly between the datasets. The plots overlap for most datasets and quality measures, which means that whether t-SNE or UMAP performed better depended on the parameter settings used. This also shows how easy it would be to make one method look much better or worse than it actually is by selecting specific parameter settings and datasets, which can be problematic, since new methods are often presented over-optimistically (Ullmann et al., 2022). That is why neutral comparison studies are essential for an objective assessment and comparison of different methods (Boulesteix et al., 2013). This thesis was an attempt at such a neutral comparison study, but it has to be noted that the observed superiority of t-SNE in this thesis does not mean that it is better in general, especially since we used t-SNE parameter settings that were already shown to produce good results by Kobak and Berens (2019), while there were no specific recommendations for UMAP settings.

Both t-SNE and UMAP were able to produce good visualizations where the points belonging to the same group formed clear clusters. While all t-SNE scatter plots for the same dataset looked very similar, the UMAP visualizations varied a lot more depending on the parameter settings used.

This thesis also has some limitations. As we already mentioned, the artificial data generation process means that the model assumptions for linear regression are not met. Another limitation was computing power. Due to relatively long computing times it was not possible to include more parameters, parameter settings and datasets in this thesis. It could be interesting to explore these things more deeply in the future.

Appendix A

A.1 Linear Models

t-SNE

KNN		
Coefficient	**Estimate**	**P-Value**
Intercept	0.805	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.001	0.936
$\text{perplexity}_{\text{combined}}$	-0.023	0.107
$\text{perplexity}_{n/100}$	-0.082	$2.48 \cdot 10^{-7}$
$\text{learning\_rate}_{n/12}$	0.051	$4.35 \cdot 10^{-5}$
$\text{dataset}_{\text{COIL-20}}$	0.010	0.635
$\text{dataset}_{\text{F-MNIST}}$	-0.580	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{Olivetti}}$	-0.238	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{MNIST}}$	-0.572	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{shuttle}}$	-0.235	$< 2 \cdot 10^{-16}$

Table A.1: Estimates of the coefficients and p-values of the linear model for t-SNE with KNN as the dependent variable using all datasets

KNC		
Coefficient	**Estimate**	**P-Value**
Intercept	0.630	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.063	0.001
$\text{perplexity}_{\text{combined}}$	0.049	0.029
$\text{perplexity}_{n/100}$	0.019	0.393
$\text{learning\_rate}_{n/12}$	-0.009	0.614
$\text{dataset}_{\text{COIL-20}}$	0.082	0.010
$\text{dataset}_{\text{F-MNIST}}$	0.186	$9.73 \cdot 10^{-8}$
$\text{dataset}_{\text{Olivetti}}$	-0.011	0.713
$\text{dataset}_{\text{MNIST}}$	0.072	0.023
$\text{dataset}_{\text{shuttle}}$	-0.188	$7.31 \cdot 10^{-8}$

Table A.2: Estimates of the coefficients and p-values of the linear model for t-SNE with KNC as the dependent variable using all datasets

CPD		
Coefficient	**Estimate**	**P-Value**
Intercept	0.408	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.058	0.008
$\text{perplexity}_{\text{combined}}$	0.097	$3.68 \cdot 10^{-4}$
$\text{perplexity}_{n/100}$	0.055	0.038
$\text{learning\_rate}_{n/12}$	0.012	0.558
$\text{dataset}_{\text{COIL-20}}$	0.098	0.010
$\text{dataset}_{\text{F-MNIST}}$	0.197	$1.14 \cdot 10^{-6}$
$\text{dataset}_{\text{Olivetti}}$	0.151	$1.09 \cdot 10^{-4}$
$\text{dataset}_{\text{MNIST}}$	-0.064	0.087
$\text{dataset}_{\text{shuttle}}$	0.093	0.014

Table A.3: Estimates of the coefficients and p-values of the linear model for t-SNE with CPD as the dependent variable using all datasets

UMAP

KNN - big datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.593	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.009	0.411
n_neighbors_{100}	-0.115	$1.41 \cdot 10^{-11}$
n_neighbors_{40}	-0.071	$5.61 \cdot 10^{-6}$
n_neighbors_{5}	-0.009	0.533
$\text{min\_dist}_{0.02}$	0.004	0.741
$\text{min\_dist}_{0.5}$	-0.054	$5.91 \cdot 10^{-5}$
$\text{dataset}_{\text{F-MNIST}}$	-0.422	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{MNIST}}$	-0.434	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{shuttle}}$	-0.110	$5.92 \cdot 10^{-11}$

Table A.4: Estimates of the coefficients and p-values of the linear model for UMAP with KNN as the dependent variable using only the big datasets

KNC - big datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.511	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.151	$2.81 \cdot 10^{-9}$
n_neighbors_{100}	0.057	0.080
n_neighbors_{40}	0.071	0.030
n_neighbors_{5}	-0.022	0.506
$\text{min\_dist}_{0.02}$	-0.004	0.874
$\text{min\_dist}_{0.5}$	0.002	0.931
$\text{dataset}_{\text{F-MNIST}}$	0.263	$2.56 \cdot 10^{-12}$
$\text{dataset}_{\text{MNIST}}$	0.161	$2.91 \cdot 10^{-6}$
$\text{dataset}_{\text{shuttle}}$	-0.059	0.069

Table A.5: Estimates of the coefficients and p-values of the linear model for UMAP with KNC as the dependent variable using only the big datasets

CPD - big datasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.196	$4.66 \cdot 10^{-10}$
$\text{initialization}_{\text{random}}$	-0.162	$1.88 \cdot 10^{-14}$
n_neighbors_{100}	0.098	$1.85 \cdot 10^{-4}$
n_neighbors_{40}	0.071	0.006
n_neighbors_{5}	0.026	0.309
$\text{min\_dist}_{0.02}$	-0.011	0.611
$\text{min\_dist}_{0.5}$	0.009	0.674
$\text{dataset}_{\text{F-MNIST}}$	0.327	$< 2 \cdot 10^{-16}$
$\text{dataset}_{\text{MNIST}}$	0.142	$1.70 \cdot 10^{-7}$
$\text{dataset}_{\text{shuttle}}$	0.313	$< 2 \cdot 10^{-16}$

Table A.6: Estimates of the coefficients and p-values of the linear model for UMAP with CPD as the dependent variable using only the big datasets

KNN - small daatasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.722	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.012	0.120
n_neighbors_{100}	-0.153	$< 2 \cdot 10^{-16}$
n_neighbors_{40}	-0.089	$1.32 \cdot 10^{-10}$
n_neighbors_{5}	-0.001	0.928
$\text{min\_dist}_{0.02}$	0.001	0.883
$\text{min\_dist}_{0.5}$	-0.027	0.004
$\text{dataset}_{\text{Olivetti}}$	-0.152	$< 2 \cdot 10^{-16}$

Table A.7: Estimates of the coefficients and p-values of the linear model for UMAP with KNN as the dependent variable using only the small datasets

KNC - small daatasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.532	$< 2 \cdot 10^{-16}$
$\text{initialization}_{\text{random}}$	-0.072	0.005
n_neighbors_{100}	0.211	$1.89 \cdot 10^{-7}$
n_neighbors_{40}	0.121	0.001
n_neighbors_{5}	-0.111	0.002
$\text{min\_dist}_{0.02}$	-0.010	0.729
$\text{min\_dist}_{0.5}$	0.005	0.877
$\text{dataset}_{\text{Olivetti}}$	0.070	0.005

Table A.8: Estimates of the coefficients and p-values of the linear model for UMAP with KNC as the dependent variable using only the small datasets

CPD - small daatasets		
Coefficient	**Estimate**	**P-Value**
Intercept	0.233	$5.31 \cdot 10^{-8}$
$\text{initialization}_{\text{random}}$	-0.071	0.007
n_neighbors_{100}	0.265	$2.81 \cdot 10^{-9}$
n_neighbors_{40}	-0.143	$1.92 \cdot 10^{-4}$
n_neighbors_{5}	-0.096	0.009
$\text{min\_dist}_{0.02}$	0.001	0.981
$\text{min\_dist}_{0.5}$	0.033	0.277
$\text{dataset}_{\text{Olivetti}}$	0.317	$9.29 \cdot 10^{-16}$

Table A.9: Estimates of the coefficients and p-values of the linear model for UMAP with CPD as the dependent variable using only the small datasets

A.2 Diagnostic Plots

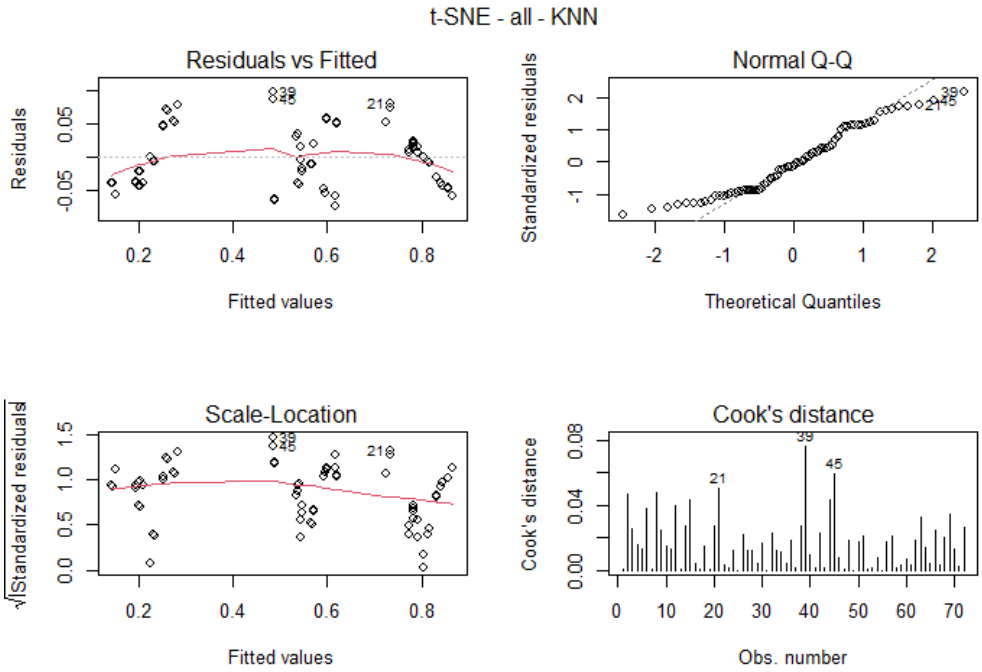

Figure A.1: Diagnostic plots of the linear model for t-SNE with KNN as the dependent variable using all datasets

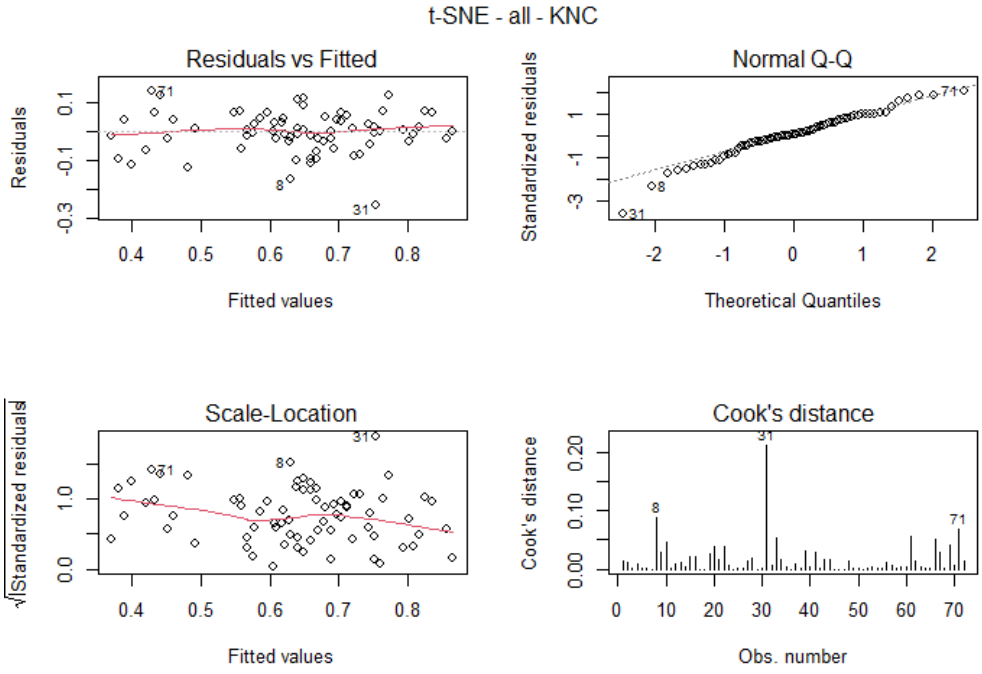

Figure A.2: Diagnostic plots of the linear model for t-SNE with KNC as the dependent variable using all datasets

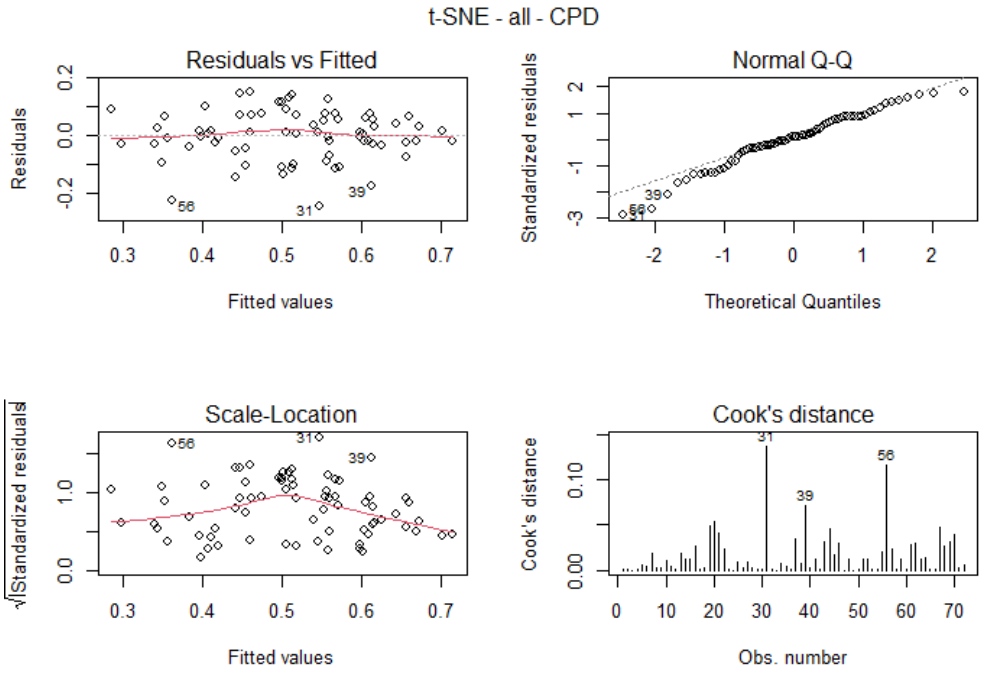

Figure A.3: Diagnostic plots of the linear model for t-SNE with CPD as the dependent variable using all datasets

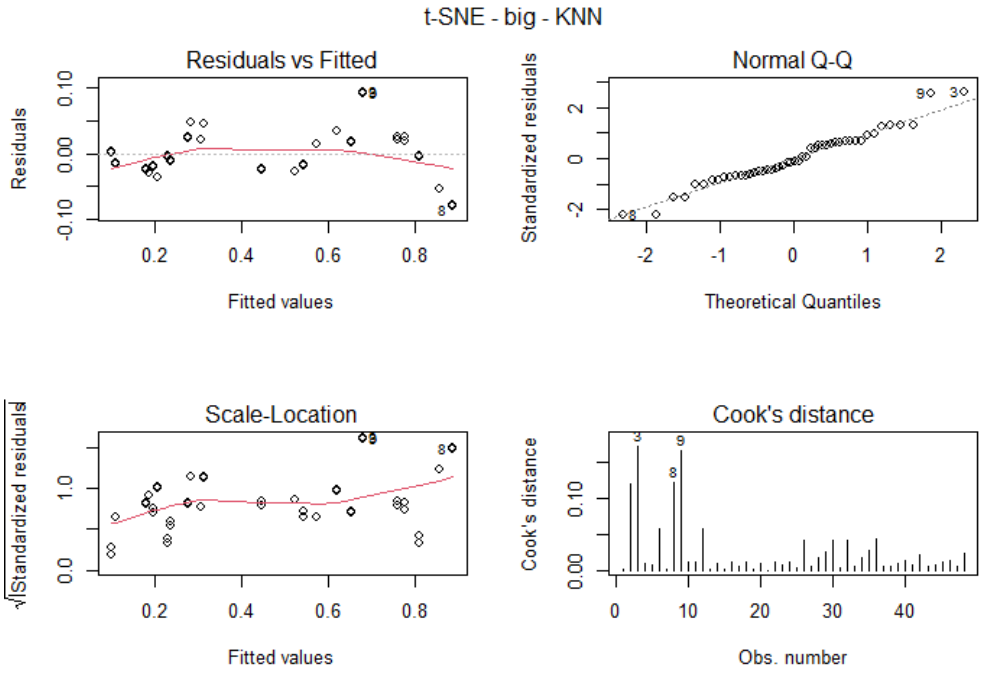

Figure A.4: Diagnostic plots of the linear model for t-SNE with KNN as the dependent variable using only the big datasets

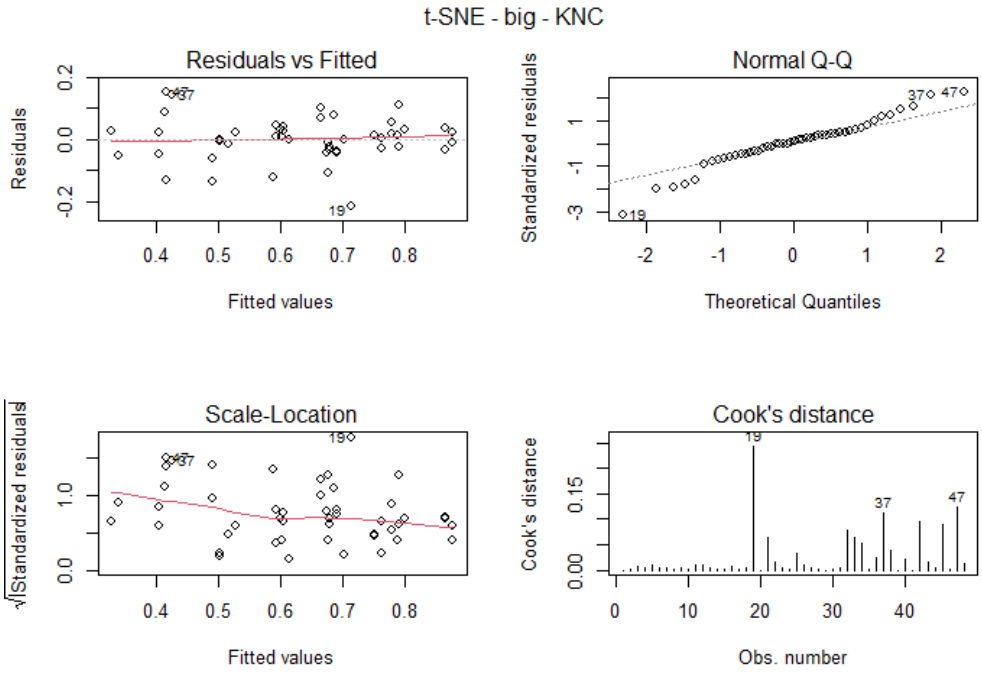

Figure A.5: Diagnostic plots of the linear model for t-SNE with KNC as the dependent variable using only the big datasets

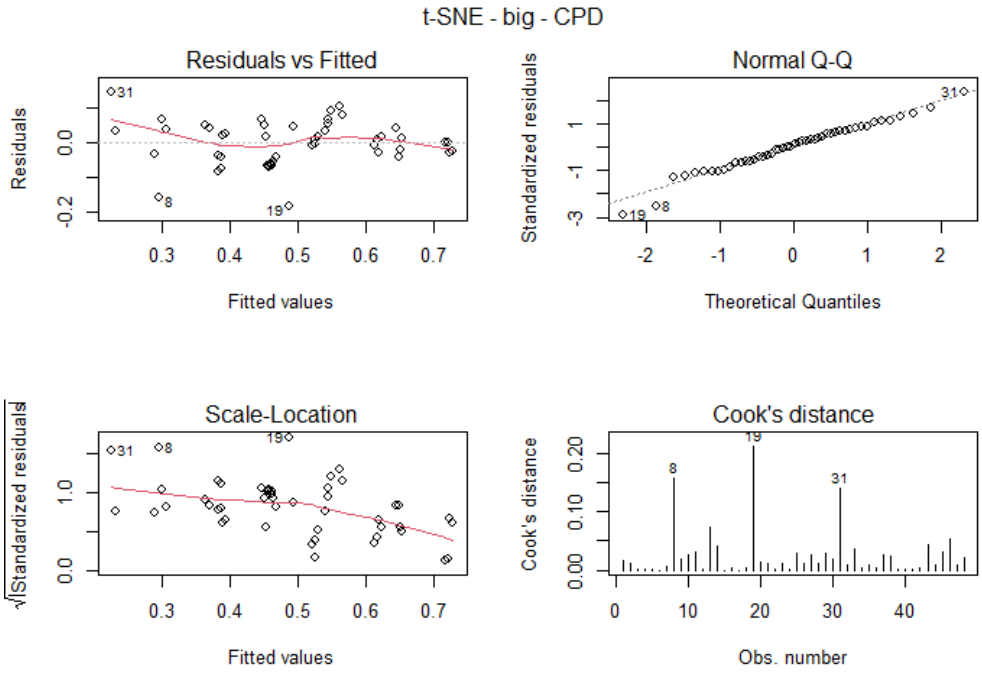

Figure A.6: Diagnostic plots of the linear model for t-SNE with CPD as the dependent variable using only the big datasets

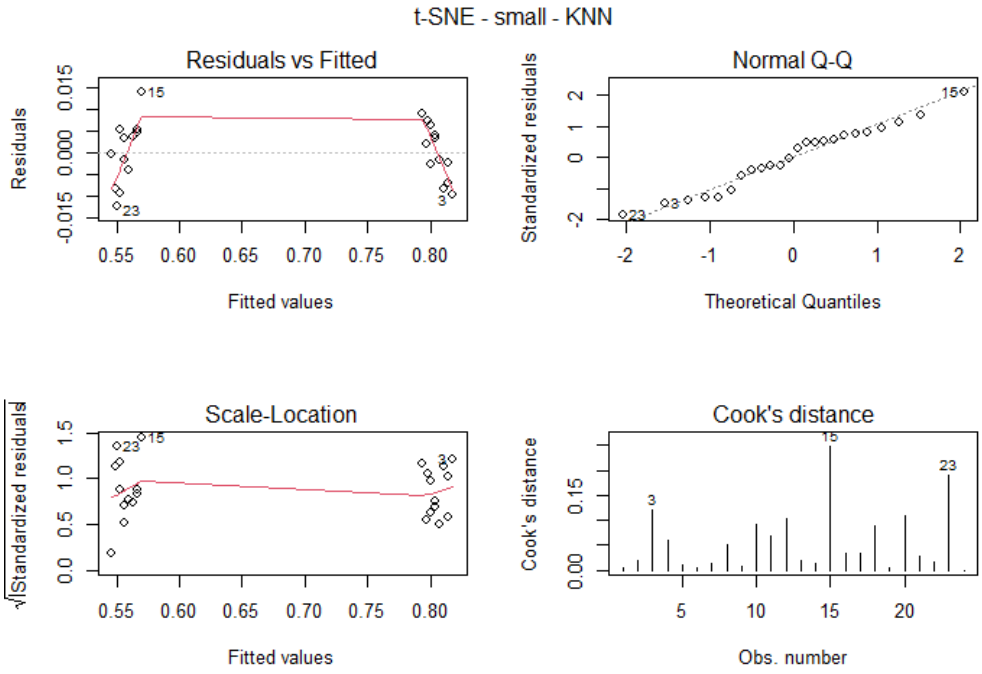

Figure A.7: Diagnostic plots of the linear model for t-SNE with KNN as the dependent variable using only the small datasets

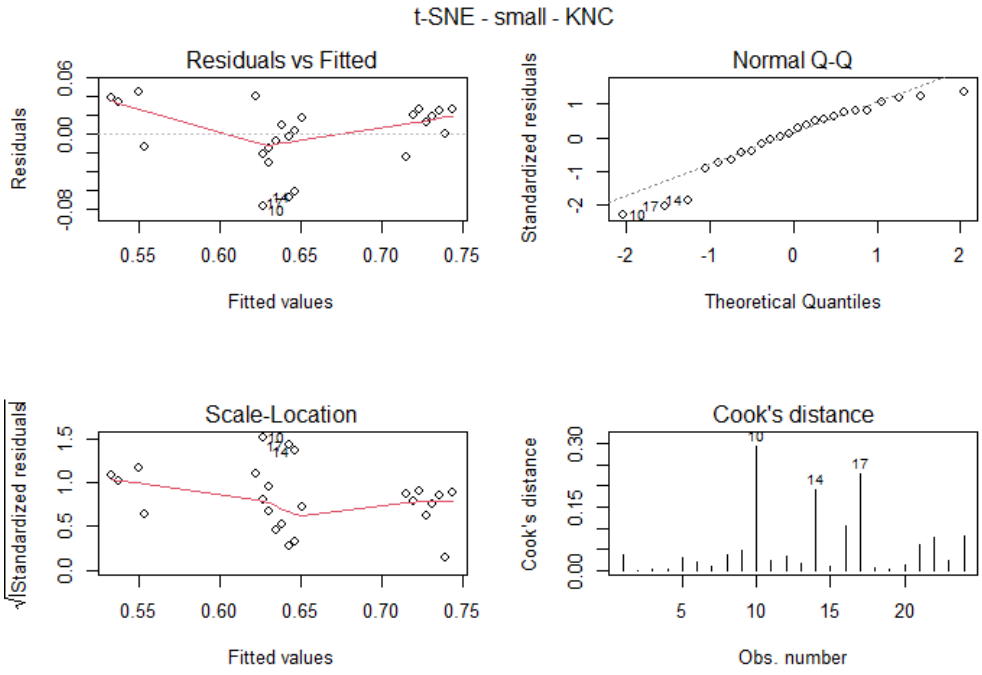

Figure A.8: Diagnostic plots of the linear model for t-SNE with KNC as the dependent variable using only the small datasets

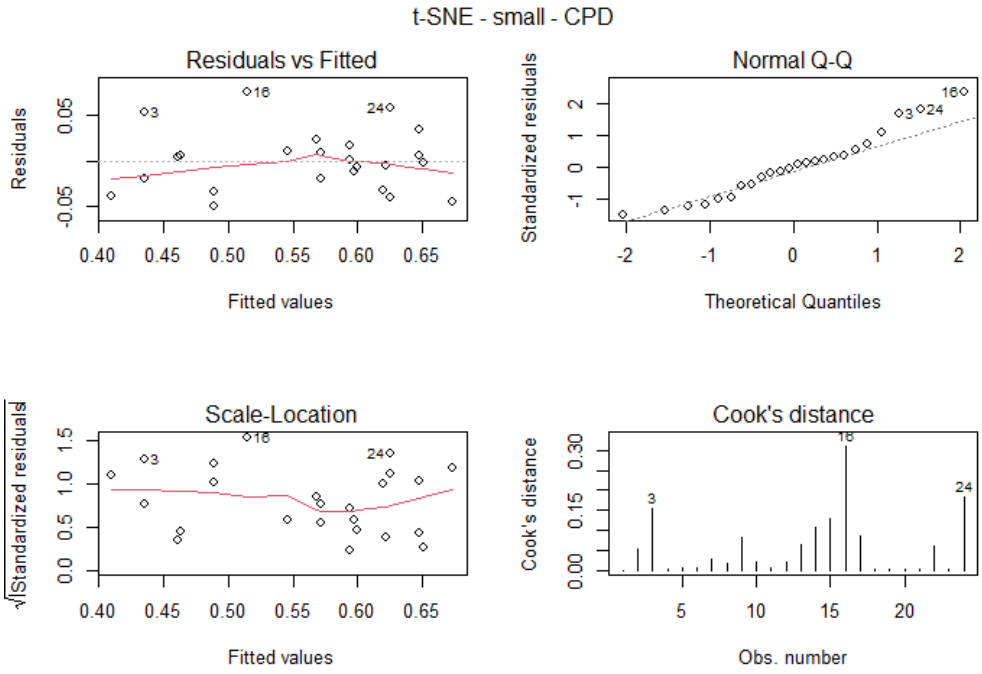

Figure A.9: Diagnostic plots of the linear model for t-SNE with CPD as the dependent variable using only the small datasets

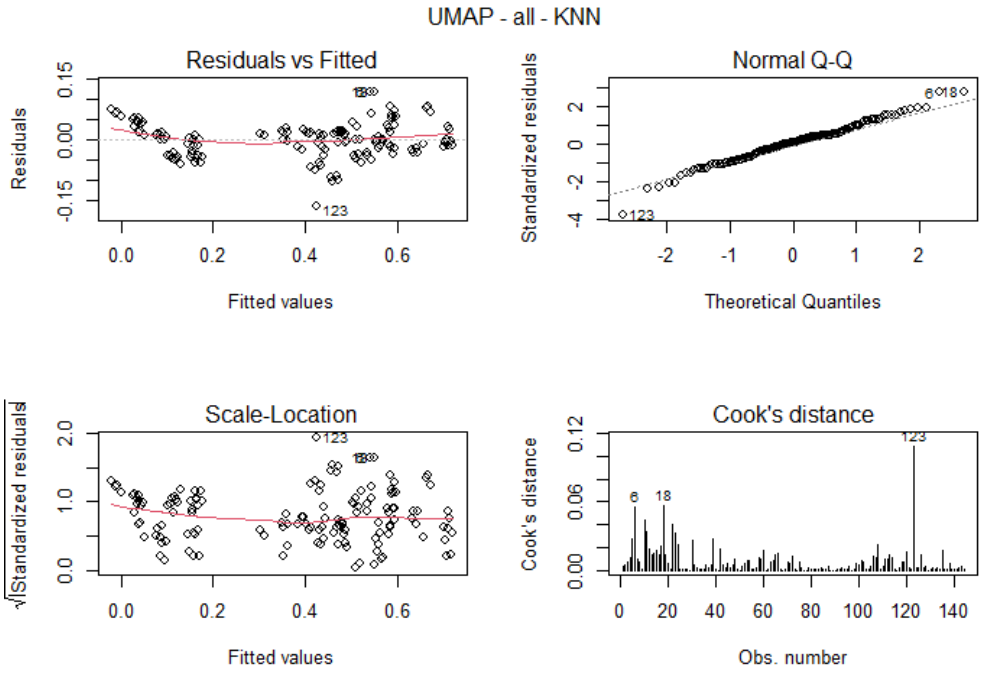

Figure A.10: Diagnostic plots of the linear model for UMAP with KNN as the dependent variable using all datasets

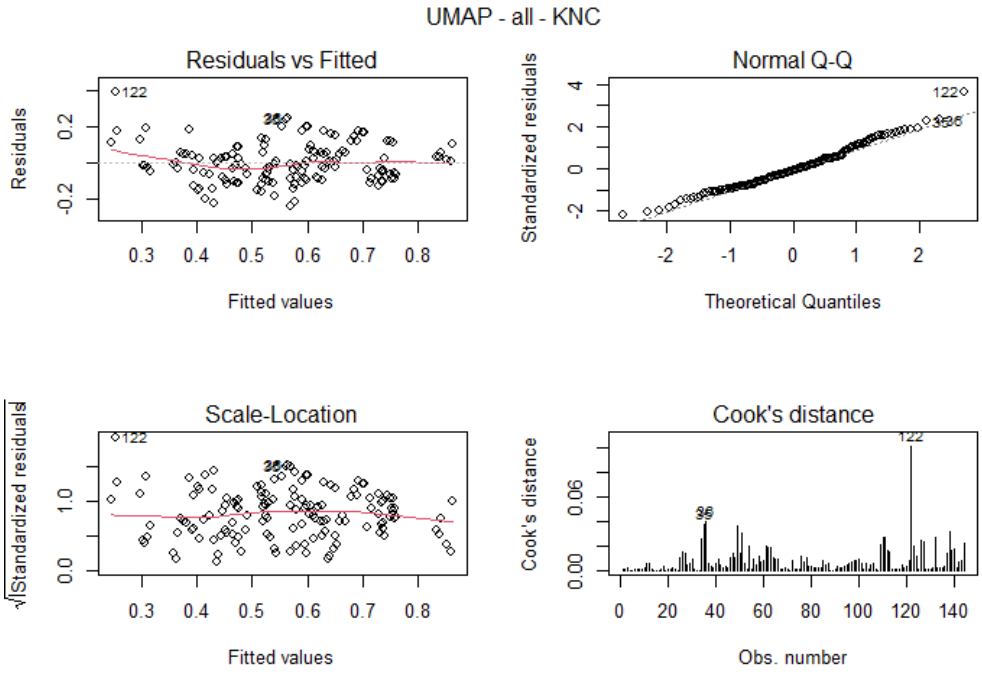

Figure A.11: Diagnostic plots of the linear model for UMAP with KNC as the dependent variable using all datasets

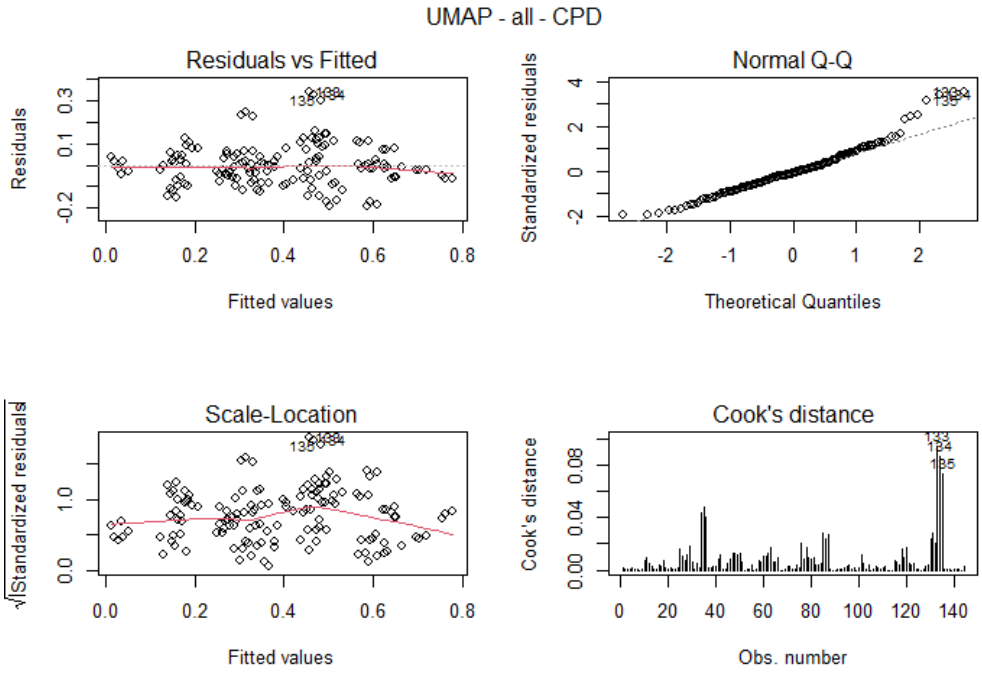

Figure A.12: Diagnostic plots of the linear model for UMAP with CPD as the dependent variable using all datasets

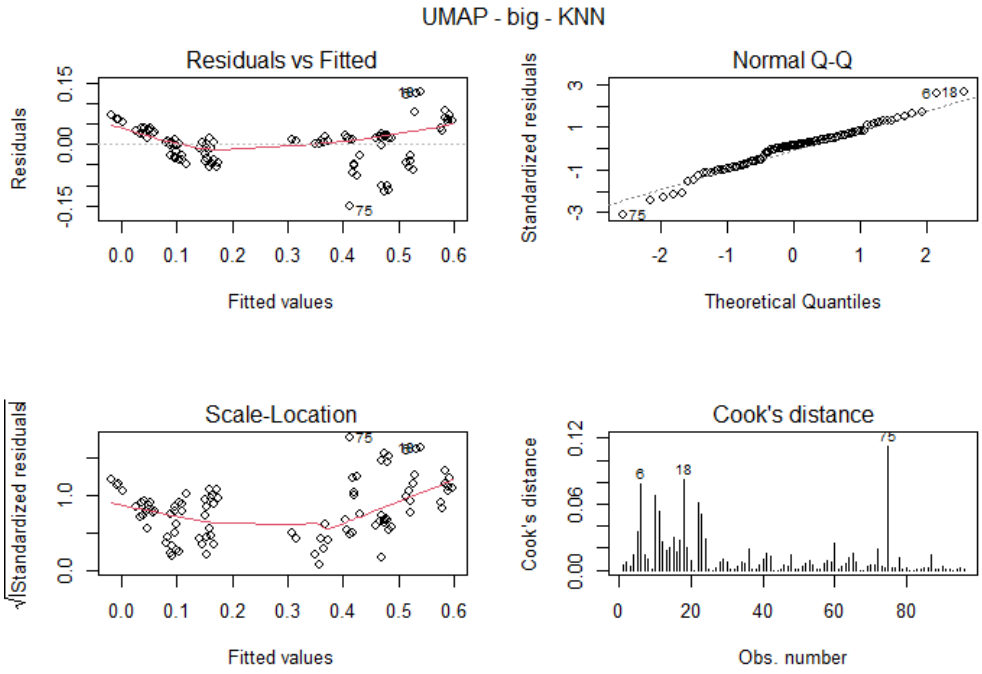

Figure A.13: Diagnostic plots of the linear model for UMAP with KNN as the dependent variable using only the big datasets

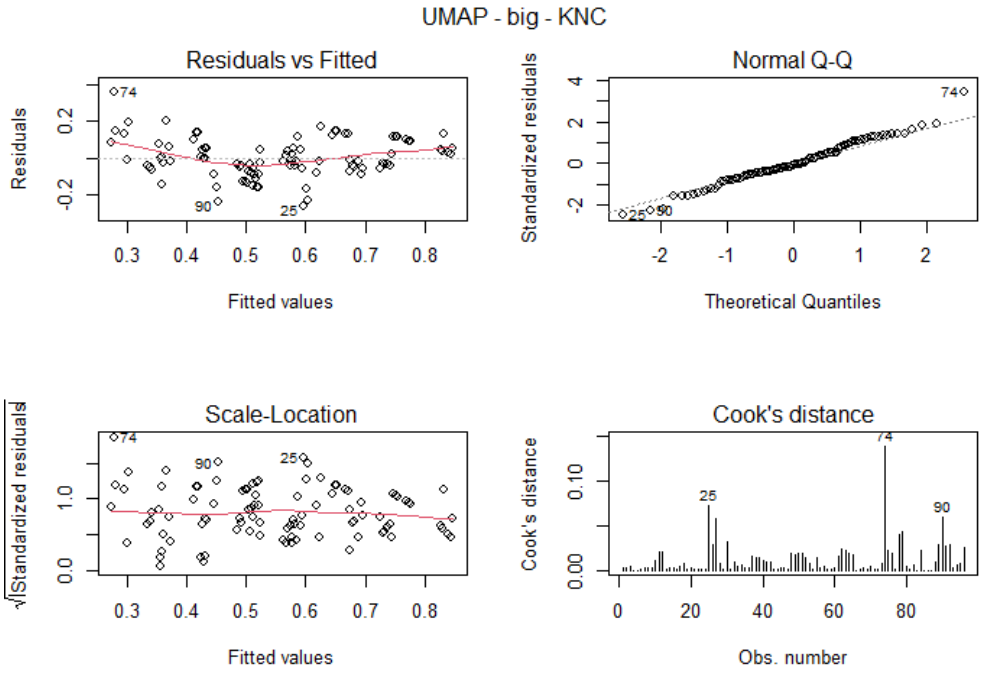

Figure A.14: Diagnostic plots of the linear model for UMAP with KNC as the dependent variable using only the big datasets

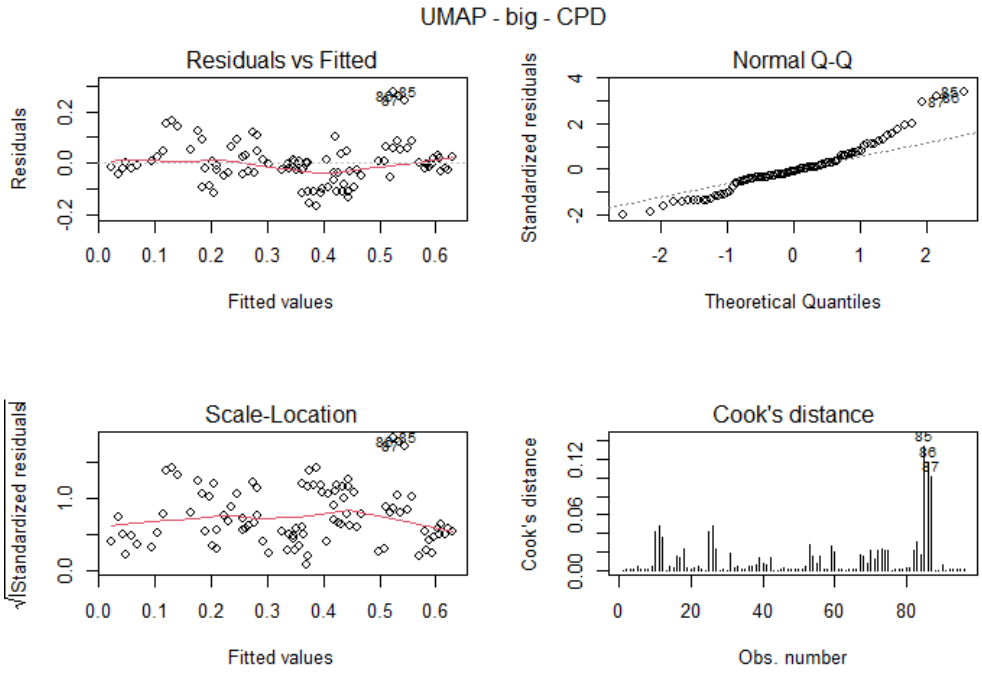

Figure A.15: Diagnostic plots of the linear model for UMAP with CPD as the dependent variable using only the big datasets

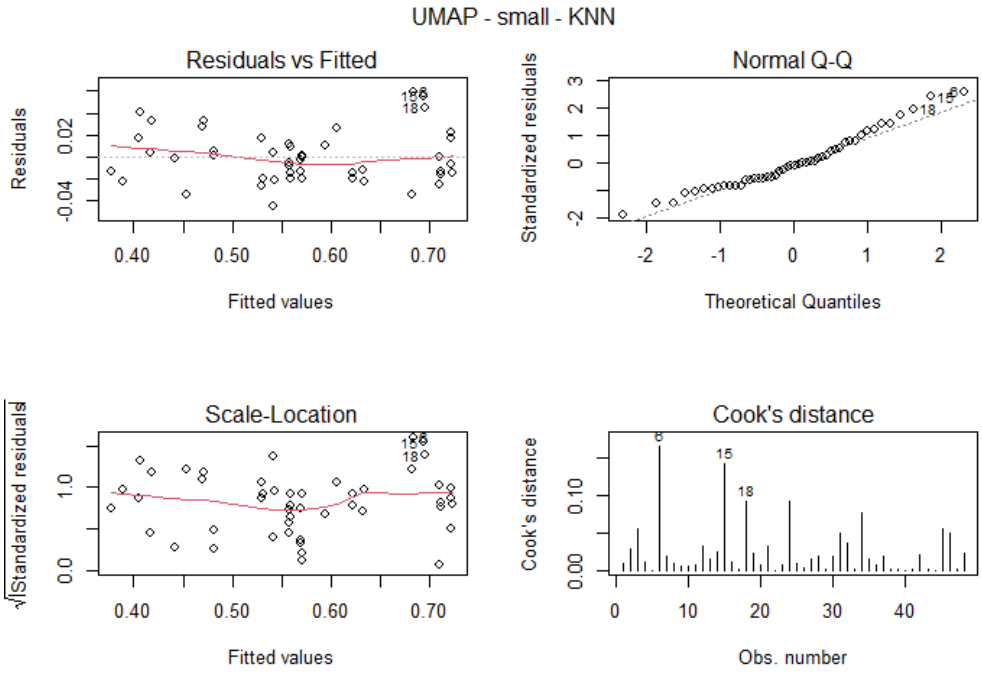

Figure A.16: Diagnostic plots of the linear model for UMAP with KNN as the dependent variable using only the small datasets

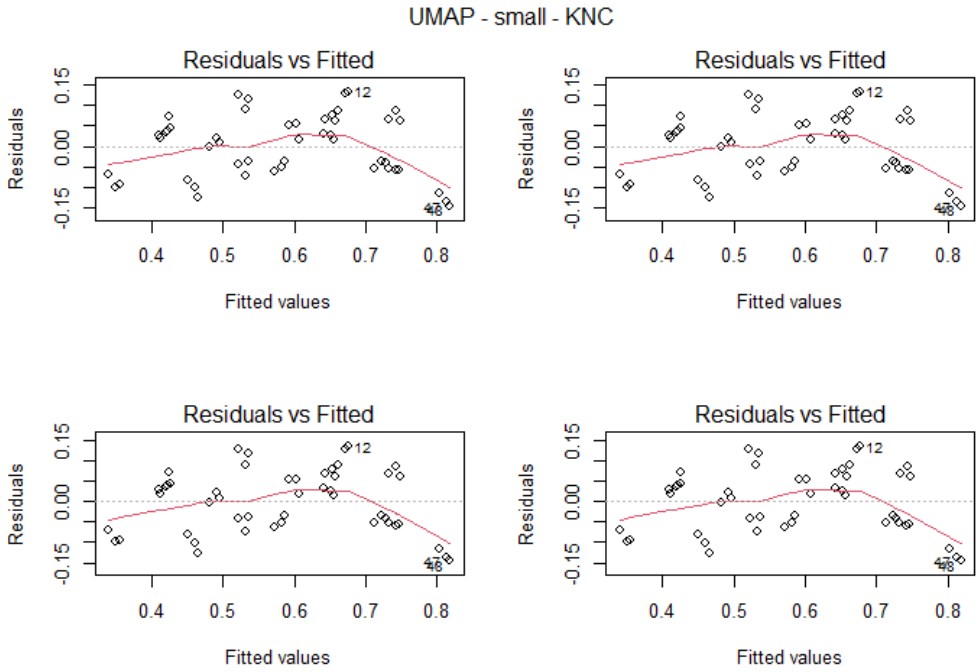

Figure A.17: Diagnostic plots of the linear model for UMAP with KNC as the dependent variable using only the small datasets

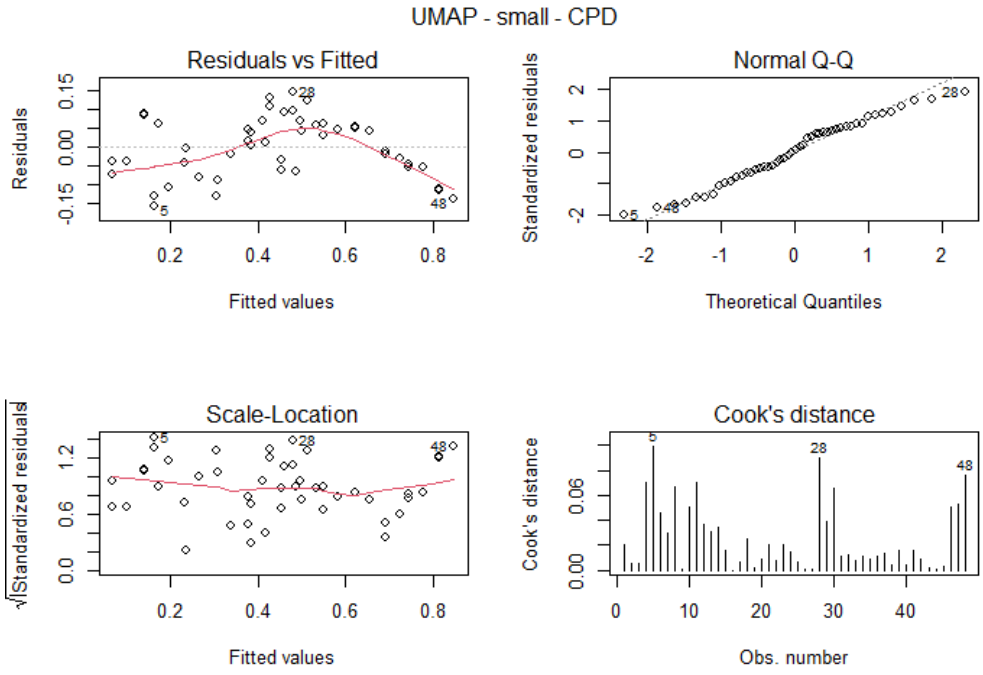

Figure A.18: Diagnostic plots of the linear model for UMAP with CPD as the dependent variable using only the small datasets

A.3 Scatter Plots

F-MNIST

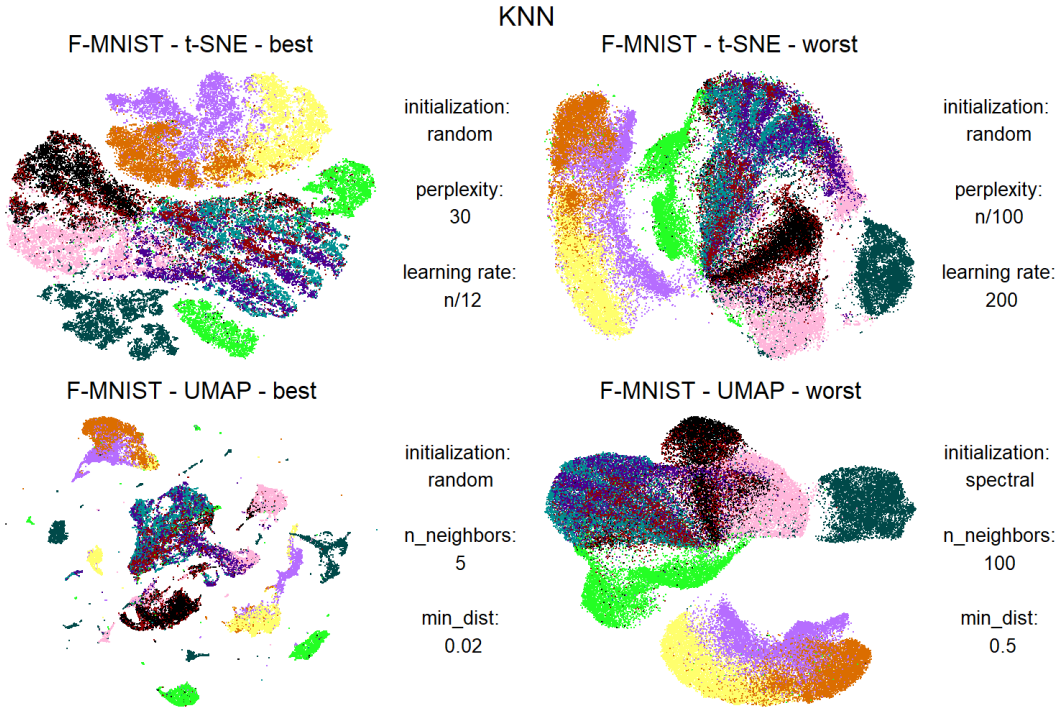

Figure A.19: Scatter plots of the t-SNE/UMAP embedings of the F-MNIST dataset with the best and worst KNN values

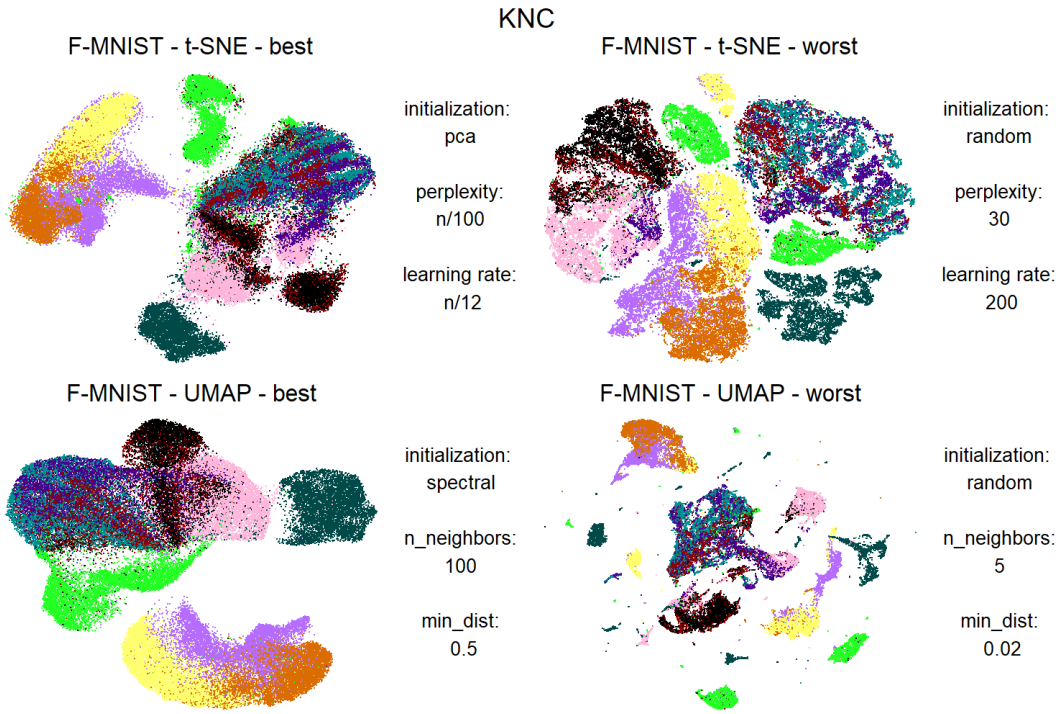

Figure A.20: Scatter plots of the t-SNE/UMAP embedings of the F-MNIST dataset with the best and worst KNC values

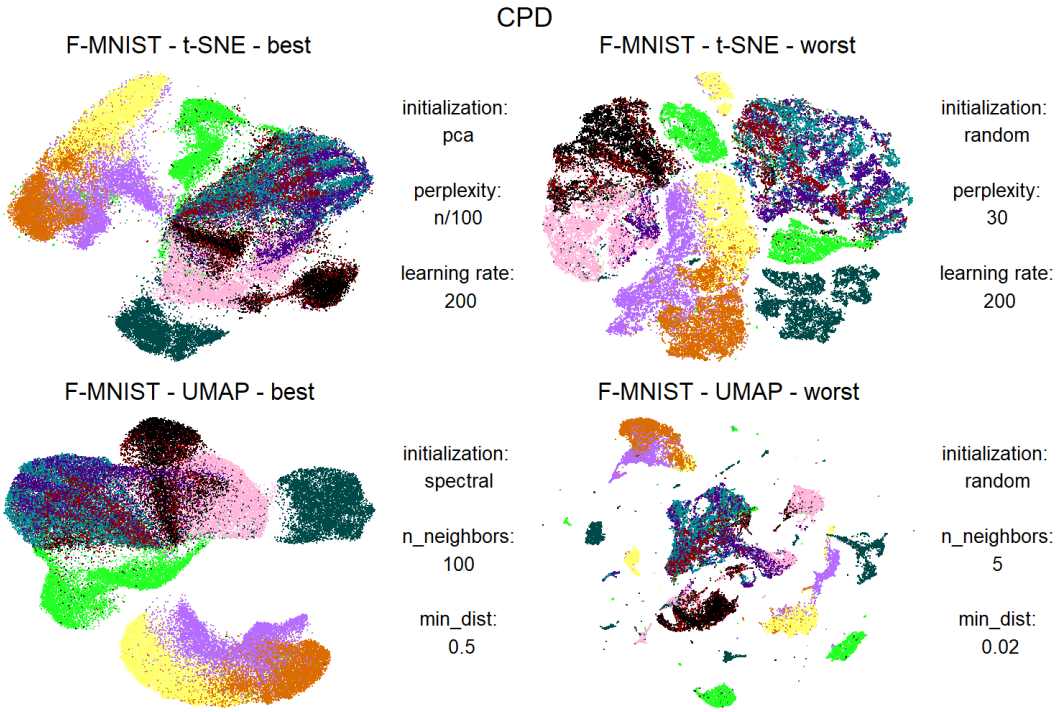

Figure A.21: Scatter plots of the t-SNE/UMAP embedings of the F-MNIST dataset with the best and worst CPD values

Olivetti Faces

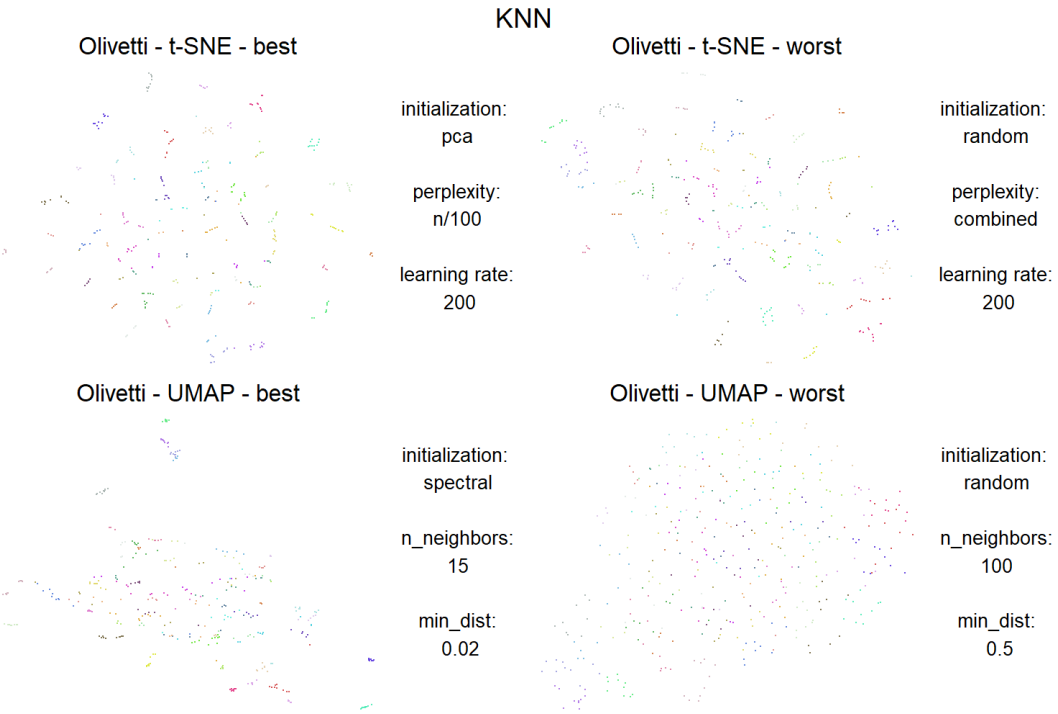

Figure A.22: Scatter plots of the t-SNE/UMAP embedings of the Olivetti Faces dataset with the best and worst KNN values

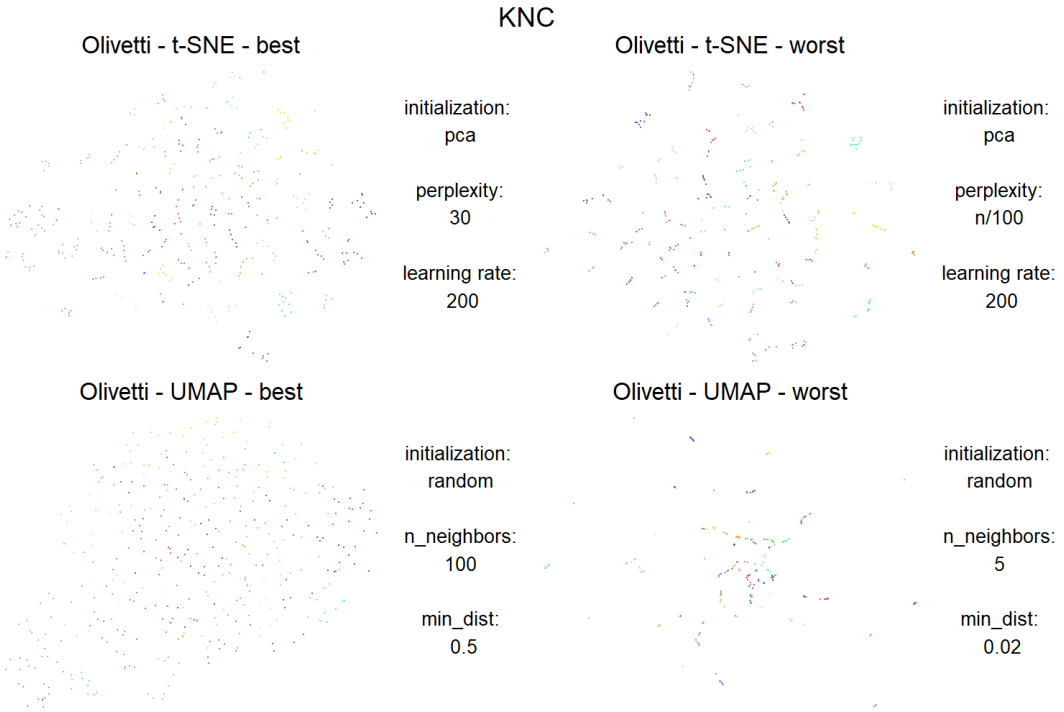

Figure A.23: Scatter plots of the t-SNE/UMAP embedings of the Olivetti Faces dataset with the best and worst KNC values

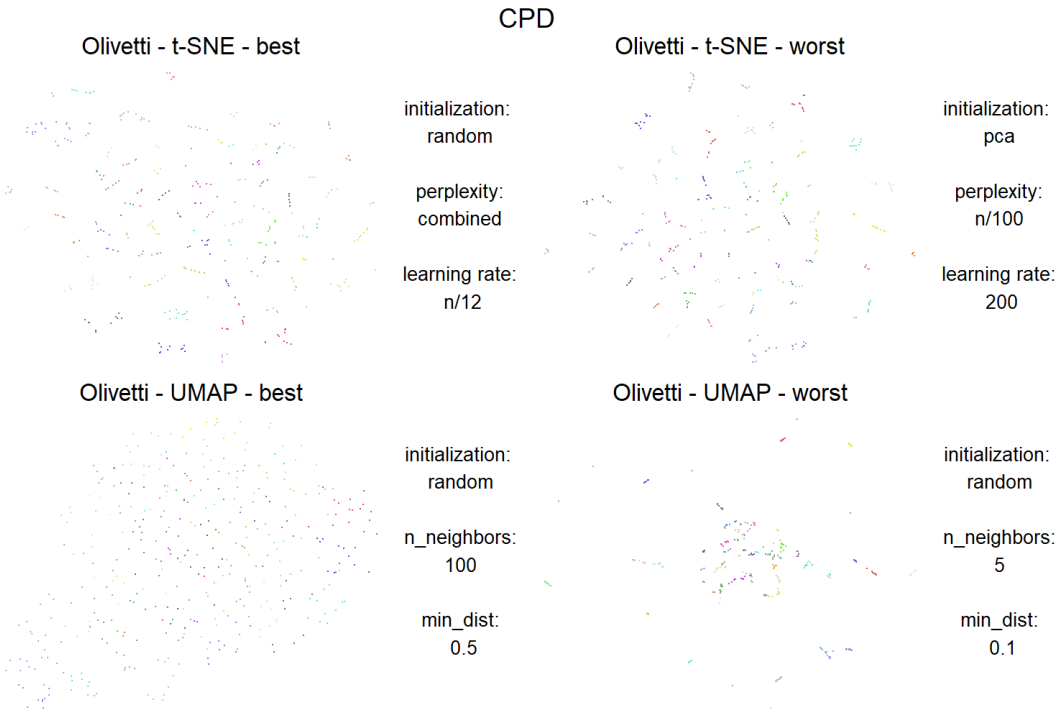

Figure A.24: Scatter plots of the t-SNE/UMAP embedings of the Olivetti Faces dataset with the best and worst CPD values

Shuttle

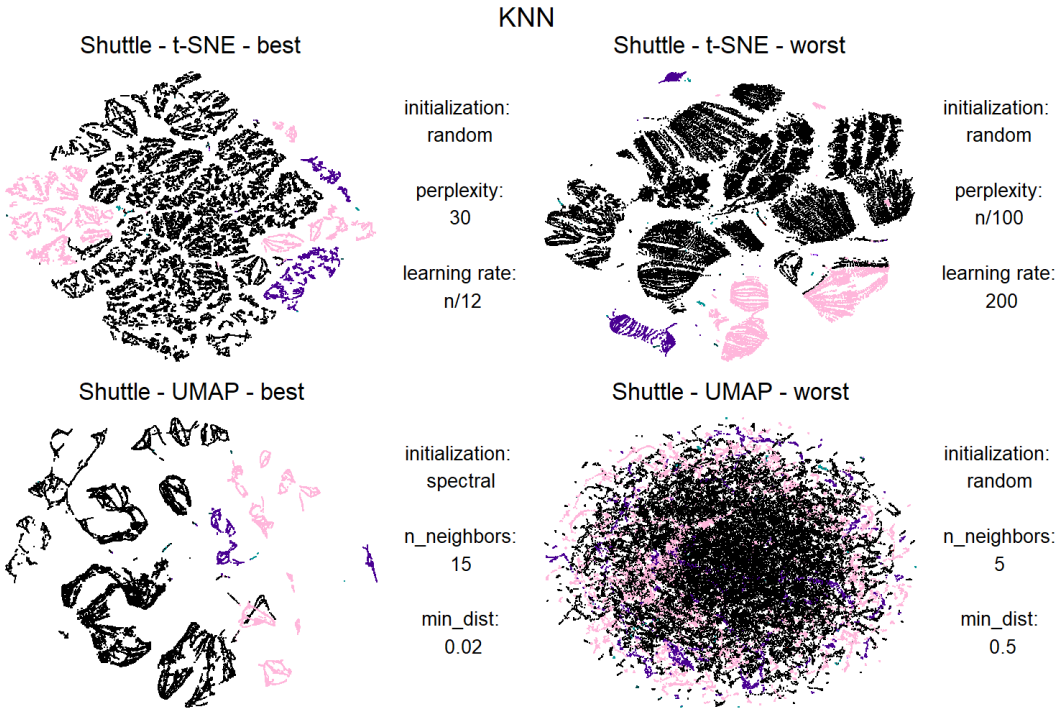

Figure A.25: Scatter plots of the t-SNE/UMAP embedings of the Shuttle dataset with the best and worst KNN values

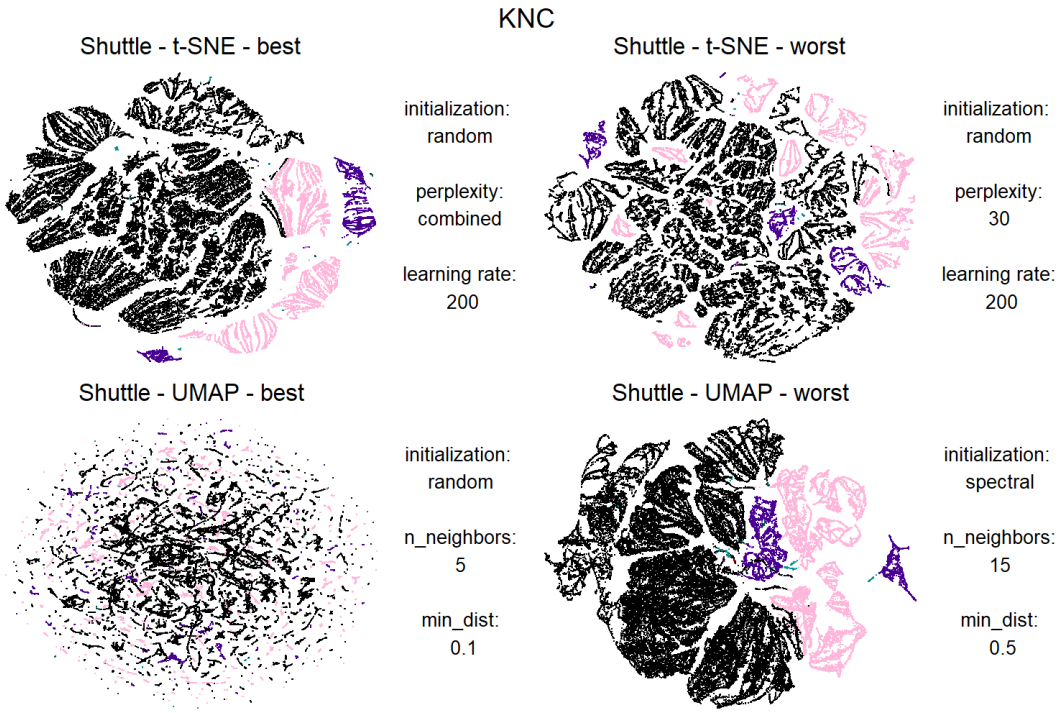

Figure A.26: Scatter plots of the t-SNE/UMAP embedings of the Shuttle dataset with the best and worst KNC values

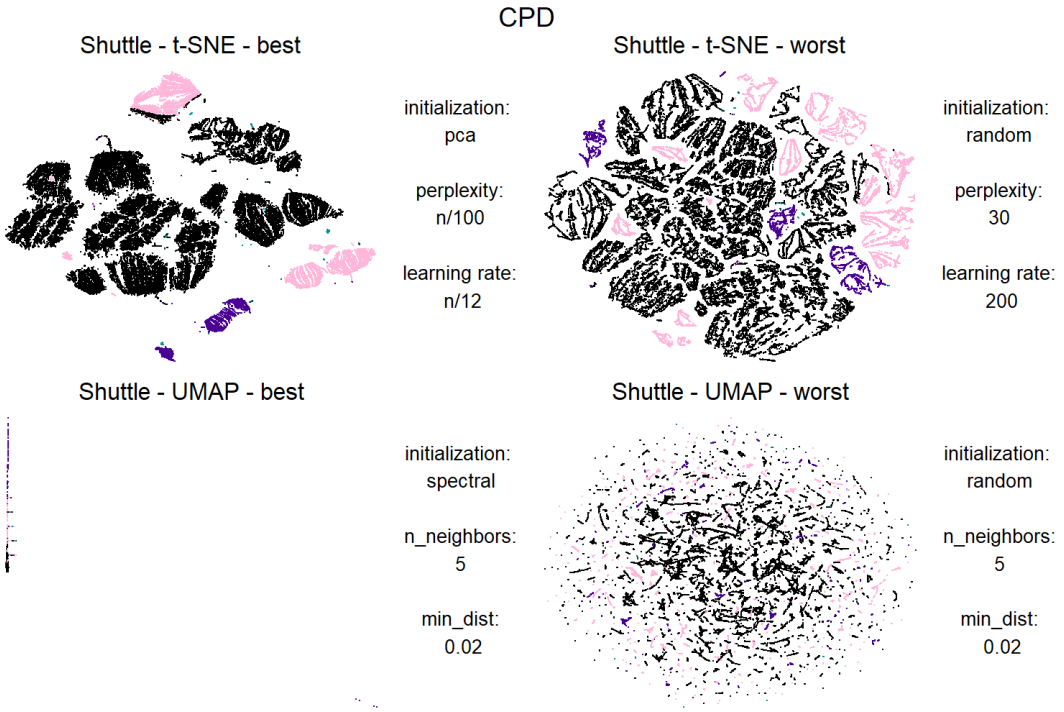

Figure A.27: Scatter plots of the t-SNE/UMAP embedings of the Shuttle dataset with the best and worst CPD values

COIL-100

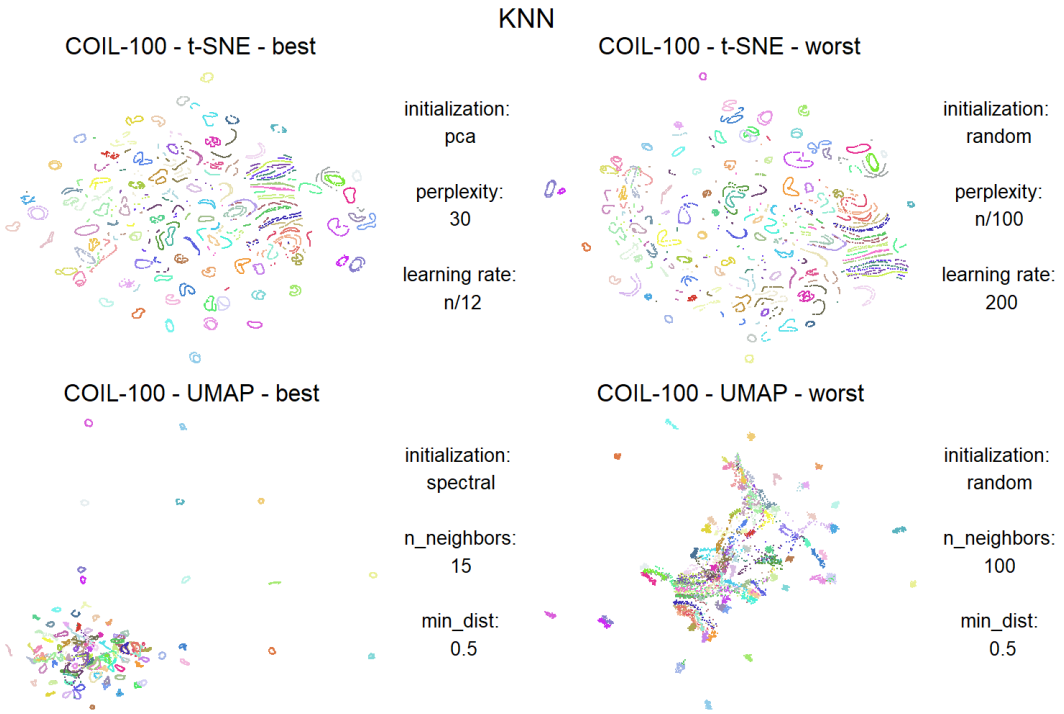

Figure A.28: Scatter plots of the t-SNE/UMAP embedings of the COIL-100 dataset with the best and worst KNN values

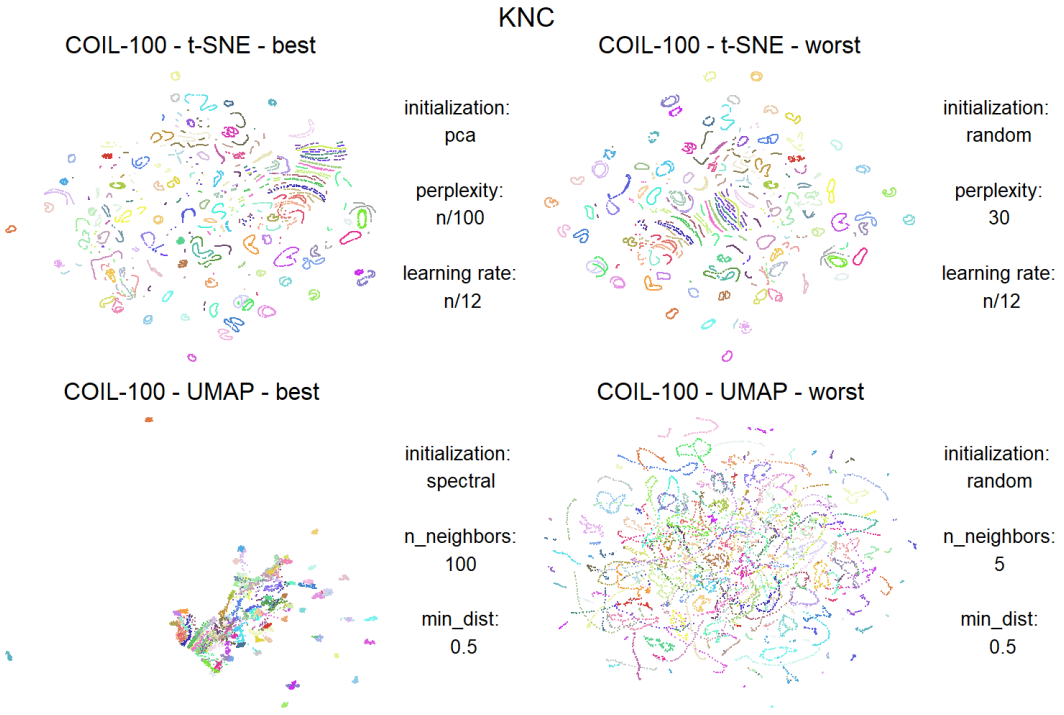

Figure A.29: Scatter plots of the t-SNE/UMAP embedings of the COIL-100 dataset with the best and worst KNC values

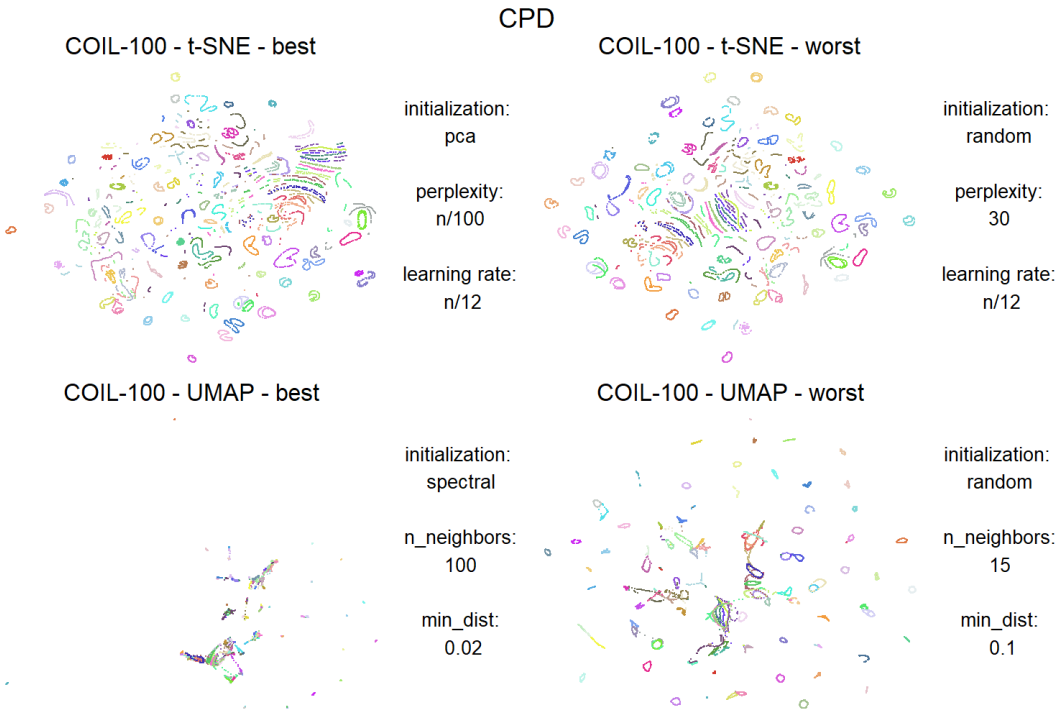

Figure A.30: Scatter plots of the t-SNE/UMAP embedings of the COIL-100 dataset with the best and worst CPD values

A.4 Parameters

t-SNE

dims: Dimensionality of the output space. Default: 2.
theta: If set to 0 regular t-SNE will be used. If non-zero, either FIt-SNE or Barnes Hut will be used. If Barnes Hut is used, theta determines the accuracy of the approximation. Default: 0.5.
max_iter: Number of iterations to run. Default: 750.
fft_not_bh: Determines whether to use FIt-SNE or Barnes Hut approximation if theta is nonzero. Default: "True" for FIt-SNE.
ann_not_vptree: Determines whether to use vp-trees (as in Barnes Hut) or approximate nearest neighbors. Default: "True" for approximate nearest neighbors.
exaggeration_factor: Coefficient used for early exaggeration. Must be > 1. Default: 12.
no_momentum_during_exag: If set to "True", no momentum is used during exaggeration. Default: "False".
stop_early_exag_iter: When to stop early exaggeration. Default: 250.
start_late_exag_iter: When to begin late exaggeration. "auto" means that, unless late_exag_coeff > 0, late exaggeration is not used. If that is the case, start_late_exag_iter is set to stop_early_exag_iter. Default: "auto".
late_exag_coeff: Coefficient used for late exaggeration. If set to -1 late exaggeration is not used. Default: -1.
max_step_norm: Maximum distance that a point can move on one iteration. If set to -1 this is deactivated. Default: 5.
nterms: Number of interpolation points per sub-interval if FIt-SNE is used. Default: 3.
intervals_per_integer: See min_num_intervals.
min_num_intervals: let maxloc $= \lceil \max(\max(X)) \rceil$ and minloc $= \lfloor \min(\min(X)) \rfloor$. The number of intervals in each dimension is either min_num_intervals or $\lceil (\text{maxloc} - \text{minloc})/\text{intervals\_per\_integer} \rceil$, depending on which is larger. min_num_intervals and intervals_per_integer must both be integers > 0. Defaults: min_num_intervals=50 and intervals_per_integer = 1.
sigma: Fixed sigma value to use when perplexity==-1. Set to -1 for none. Default: -1.
K: Number of nearest neighbors to get when using a fixed sigma value. Set to -30 to use none. Default: -30.
load_affinities: If set to 1, input similarities are loaded from a file and not computed. If set to 2, input similarities are saved into a file. If set to 0, affinities are neither saved nor loaded. Default: NULL.
perplexity_list: Perplexity combination that will be used if perplexity==0. Default: NULL.
df: Degrees of freedom of the t-distribution. Default: 1.0.

UMAP

n_components: Dimensionality of the output space. Default: 2.
metric: Metric used to calculate distances between the datapoints. Available settings are: "euclidean", "manhattan", "cosine", "pearson", "pearson2" or a custom

metric given as a function. Default: "euclidean".
n_epochs: Number of training epochs to use during the optimization. Default: 200.
input: "data" or "dist". Controls whether input is treated as a data matrix or distance matrix. Default: "data".
set_op_ratio_mix_ratio: Used during the construction of a fuzzy simplicial graph. Range: $[0, 1]$. Default: 1.
local_connectivity: Used during the construction of fuzzy simplicial sets. Default: 1.
bandwidth: Used during the construction of fuzzy simplicial sets. Default: 1.
alpha: Initial value of the learning rate used in the layout optimization. Default: 1.
gamma: Influences, together with alpha, the learning rate. Default: 1.
negative_sample_rate: Number of non-neighbor points used per point and iteration during the optimization. Default: 5.
spread: Influences, together with the min_dist parameter, the calculation of the a and b values. Default: 1.
a: Manually sets the a value. Default: NA.
b: Manually sets the b value. Default: NA.
random_state: Seed used for random number generation during umap(). Default: NA.
transform_state: Seed used for random number generation during predict(). Default: NA.
knn: Possibility to provide precomputed nearest neighbors. Default: NA.
knn_repeats: Determines how often to restart the knn search. Default: 1.
verbose: Determines whether or not to show progress. Default: "False".
umap_learn_args: Arguments to the python package "umap-learn". Default: NA.

List of Figures

List of Tables

Bibliography

Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel W. H. Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W. Newell. Dimensionality reduction for visualizing single-cell data using umap. *Nature biotechnology*, 2018. doi: 10.1038/nbt.4314.

Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003. ISSN 0899-7667. doi: 10.1162/089976603321780317.

Anne-Laure Boulesteix, Sabine Lauer, and Manuel J. A. Eugster. A plea for neutral comparison studies in computational sciences. *PLoS ONE*, 8(4):e61562, apr 2013. doi: 10.1371/journal.pone.0061562. URL `https://doi.org/10.1371%2Fjournal.pone.0061562`.

Stefan Buchka, Alexander Hapfelmeier, Paul P. Gardner, Rory Wilson, and Anne-Laure Boulesteix. On the optimistic performance evaluation of newly introduced bioinformatic methods. *Genome Biology*, 22(1):152, 2021. ISSN 1474-760X. doi: 10.1186/s13059-021-02365-4.

Nikhil Buduma and Nicholas Locascio. *Fundamentals of deep learning: Designing next-generation machine intelligence algorithms*. O'Reilly Media, Sebastopol, CA, first edition edition, 2017. ISBN 9781491925614. URL `https://ebookcentral.proquest.com/lib/kxp/detail.action?docID=4865877`.

Tuan Nhon Dang, Leland Wilkinson, and Anushka Anand. Stacking graphic elements to avoid over-plotting. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1044–1052, 2010. doi: 10.1109/TVCG.2010.197.

Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

Philip Greengard, Yukun Liu, Stefan Steinerberger, and Aleh Tsyvinski. Factor clustering with t-sne. *SSRN Electronic Journal*, 2020. doi: 10.2139/ssrn.3696027.

Geoffrey E. Hinton and Sam Roweis. Stochastic neighbor embedding. *Advances in Neural Information Processing Systems*, 15, 2002. URL `https://proceedings.neurips.cc/paper/2002/file/6150ccc6069bea6b5716254057a194ef-Paper.pdf`.

Robert A. Jacobs. Increased rates of convergence through learning rate adaptation. *Neural Networks*, 1(4):295–307, 1988. ISSN 08936080. doi: 10.1016/0893-6080(88)90003-2.

George J. Klir and Bo Yuan. *Fuzzy sets and fuzzy logic: Theory and applications*. Prentice Hall, Upper Saddle River, NJ, 1. print edition, 1995. ISBN 0131011715.

Dmitry Kobak and Philipp Berens. The art of using t-sne for single-cell transcriptomics. *Nature communications*, 10(1):5416, 2019. doi: 10.1038/s41467-019-13056-x.

Dmitry Kobak and George C. Linderman. Initialization is critical for preserving global data structure in both t-sne and umap. *Nature biotechnology*, 39(2):156–157, 2021. doi: 10.1038/s41587-020-00809-z.

Jakub Konečný and Peter Richtárik. Semi-stochastic gradient descent methods, 2013.

Tomasz Konopka. *umap: Uniform Manifold Approximation and Projection*, 2022. URL `https://CRAN.R-project.org/package=umap`. R package version 0.2.9.0.

George C. Linderman, Manas Rachh, Jeremy G. Hoskins, Stefan Steinerberger, and Yuval Kluger. Fast interpolation-based t-sne for improved visualization of single-cell rna-seq data. *Nature methods*, 16(3):243–245, 2019. doi: 10.1038/s41592-018-0308-4.

Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction, 2018.

James Melville. *snedata: SNE Simulation Dataset Functions*, 2022. R package version 0.0.0.9000.

Samer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (coil-20). Technical Report CUCS-005-96, Department of Computer Science, Columbia University, February 1996a.

Samer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (coil-100). Technical Report CUCS-006-96, Department of Computer Science, Columbia University, February 1996b.

Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks, 2010.

Theresa Ullmann, Anna Beer, Maximilian Hünemörder, Thomas Seidl, and Anne-Laure Boulesteix. Over-optimistic evaluation and reporting of novel cluster algorithms: an illustrative study. *Advances in Data Analysis and Classification*, 2022. ISSN 1862-5347. doi: 10.1007/s11634-022-00496-5.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

Marc Vermeulen, Kate Smith, Katherine Eremin, Georgina Rayner, and Marc Walton. Application of uniform manifold approximation and projection (umap) in spectral imaging of artworks. *Spectrochimica acta. Part A, Molecular and biomolecular spectroscopy*, 252:119547, 2021. doi: 10.1016/j.saa.2021.119547.

Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL `https://ggplot2.tidyverse.org`.

Duoduo Wu, Joe Yeong Poh Sheng, Grace Tan Su-En, Marion Chevrier, Josh Loh Jie Hua, Tony Lim Kiat Hon, and Jinmiao Chen. Comparison between umap and t-sne for multiplex-immunofluorescence derived single-cell data from tissue sections. *bioRxiv*, 2019. doi: 10.1101/549659. URL `https://www.biorxiv.org/content/early/2019/02/15/549659`.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Yang Yang, Hongjian Sun, Yu Zhang, Tiefu Zhang, Jialei Gong, Yunbo Wei, Yong-Gang Duan, Minglei Shu, Yuchen Yang, Di Wu, and Di Yu. Dimensionality reduction by umap reinforces sample heterogeneity analysis in bulk transcriptomic data. *Cell Reports*, 36(4):109442, 2021. ISSN 2211-1247. doi: https://doi.org/10.1016/j.celrep.2021.109442. URL `https://www.sciencedirect.com/science/article/pii/S2211124721008597`.