Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrie en Operationele Research

Comparing clustering methods on (non)dimensionally reduced High Dimensional Low Sample Size data

Wout Peters (531818)

Supervisor:	Jeffrey Durieux
Second assessor:	Marina Khismatullina
Date final version:	1st July 2023

Abstract

Clustering High-Dimensional, Low Sample Size (HDLSS) data is an active area of research with many applications, including the biological sciences and computer vision. With standard methods often assumed not to be useful in HDLSS settings, alternative clustering methods and dimensionality reduction techniques have been developed, designed for clustering (non)dimensionally reduced HDLSS data specifically. Whilst their superiority over standard methods has been shown, never have these state-of-the-art methods been empirically compared to each other. Furthermore, it is generally thought that dimensionality reduction prior to clustering discards important high-dimensional information, and thus, nondimensionally reduced clustering is better. Therefore, for 9 (non)dimensionally reduced HDLSS datasets, we create clusters using several standard and HDLSS-specific methods, and we compare cluster quality across all methods with internal and external cluster validation. We conclude that in terms of performance and interpretability, dimensionally reduced clustering of HDLSS data is better than nondimensionally reduced clustering. Furthermore, we find that in practice, HDLSS-specific methods do not necessarily outperform standard methods.

1 Introduction

Clustering High-Dimensional, Low Sample Size (HDLSS) data has been an active area of research for several decades. With the increasing popularity of HDLSS microarray gene expression data analysis, where scientists try to find groups of genes out of thousands with the goal of identifying their relation with specific diseases, clustering of HDLSS data is used increasingly often in the biological sciences (Yu et al., 2014). Another increasingly relevant application of clustering HDLSS data is computer vision (Cheema et al., 2015), where clustering is used to group visual patterns. Because Hall et al. (2005) show that theoretically, classical clustering methods suffer in HDLSS conditions, researchers have created numerous methods that enable good clustering performance over the years. Though most papers compare the novel method's performance with classical methods, a comparison of the performance to other novel methods is often overlooked. Our research is, to the best of our knowledge, therefore the first to extensively compare state-of-the-art HDLSS-specific methods.

Inspired by the paper of Renjith et al. (2021), we aim to compare the performance of clustering methods on (non)dimensionally reduced HDLSS data. In their paper, they provide a road map for a comparison of clustering methods and Dimensionality Reduction Techniques (DRTs). In our research, we first try to reproduce their work using the same data. After, we use the road map they suggest as inspiration for an equivalent experiment using HDLSS data. A similar comparison of methods for high dimensional data has been made in Bouveyron and Brunet-Saumard (2014). In their paper, they argue that dimensionality reduction should be avoided when clustering high-dimensional data because lower-dimensional mappings inevitably discard information about the geometric structure of the high-dimensional data, which they believe is destructive for clustering performance. As an alternative, they promote the usage of methods that perform implicit dimensionality reduction; parsimonious models, subspace clustering methods and variable selection methods designed for clustering. Though the context in their paper is high dimensional data rather than HDLSS data, their argument should also hold for HDLSS

settings. Performing our research almost a decade after their paper was published, a series of DRTs have been developed that, in combination with standard clustering methods, show great clustering performance for HDLSS data. Furthermore, clustering methods have been developed that are designed to work well on HDLSS data. Therefore, while we do use the hypothesis from Bouveyron and Brunet-Saumard (2014) that clustering nondimensionally reduced HDLSS data is better than clustering dimensionally reduced HDLSS data, it is of interest to research whether this holds using state-of-the-art methods in HDLSS settings. Our main research question thus is: To what extent do clustering methods designed for HDLSS data outperform standard clustering methods for dimensionally reduced HDLSS data?

To answer the research question, we use 9 HDLSS gene expression microarray datasets, obtained from the popular R package `datamicroarray` (Ramey, 2016), and we evaluate whether clusters created with HDLSS data are better than clusters created with dimensionally reduced HDLSS data. We do this with internal and external cluster validation, analysing clustering performance based on both geometric structure and classification accuracy. Additionally, we try to find (combinations of) methods that consistently perform well throughout the experiments. For each dataset, we create dimensionally reduced versions, with the DRTs used in Renjith et al. (2021) and several state-of-the-art DRTs, designed for HDLSS settings, which are introduced in Section 2. On the dimensionally reduced data, we apply the standard clustering methods proposed in Renjith et al. (2021). On the non-dimensionally reduced data, we use, besides the latter standard clustering methods, several state-of-the-art, HDLSS-specific clustering methods. We then show that both in terms of performance and interpretability, clustering dimensionally reduced HDLSS data is actually better than clustering nondimensionally reduced HDLSS data, contradicting our hypothesis. Furthermore, we show that HDLSS-specific DRTs and clustering methods do not necessarily outperform (all) standard methods.

The paper continues as follows. In Section 2, we introduce additional literature on which our research is based. Section 3 describes the data used in more detail. Then, in Section 4, we explain all researched DRTs, clustering methods and validation indices in detail. Furthermore, we provide specific information about how our research was performed and show in detail how we replicate and extend Renjith et al. (2021). In Section 5, the results of the research are extensively covered. Finally, we exhibit our concluding findings, present the implications of our research, and make suggestions for further research in Section 6.

2 Theory

To get an idea of how HDLSS-specific clustering methods outperform standard clustering methods for dimensionally reduced HDLSS data, we first want to answer the question: Which combinations of DRTs and clustering methods result in the best cluster quality for HDLSS data? Multiple DRTs for HDLSS data have been developed and have been shown to be superior to classical DRTs for clustering purposes. Mahmud et al. (2021) show that clustering of dimensionally reduced HDLSS data using a Variational Autoencoder gives superior results over clustering of dimensionally reduced HDLSS data using a series of DRTs, including all methods in Renjith et al. (2021). While they do not invent the method, nor its use for dimensionality reduction, they are the first to research the superiority of a Variational Autoencoder over classical DRTs.

Alternatively, Kosztyán et al. (2022) present a new method, Network-based Dimensionality reduction Analysis, and show that it outperforms some classical DRTs in both non-HDLSS and HDLSS settings. Finally, Nakayama et al. (2021) theoretically and experimentally show that dimensionality reduction using Gaussian kernel Principal Component Analysis is effective for clustering HDLSS data, though not comparing its performance to classical DRTs in practical applications. Alternatively, several methods have been developed that both dimensionally reduce and cluster the data simultaneously. One recent method is given in Cai et al. (2023), which uses tensors to encode high-order relations to lower dimensions, and clusters the data accordingly. Another method, often used specifically for microarray gene expression data, is Biclustering (Mitra & Banka, 2006). While it simultaneously clusters both genes (features) and observations, Q. Liu et al. (2014) show Biclustering's limitations, and propose an improved version using Sparse Clustering. Though interesting and supposedly useful for the data in our research, we do not feel they fit the research question at hand, for which reason we leave their investigation up to further research.

After identifying the combinations of DRTs and clustering methods that result in the best cluster quality, we want to answer the question: Which clustering methods result in the best cluster quality for nondimensionally reduced HDLSS data? Hall et al. (2005) proof that due to the geometric representation of data points in HDLSS settings, popular clustering methods like k-Means and hierarchical clustering suffer, due to their dependence on the Euclidean distance. A first method proposed to overcome this issue is given in Ahn et al. (2012), where they propose to use the Maximal Data Piling (MDP) distance instead of the Euclidean distance. Alternatively, Sarkar and Ghosh (2019) present a new dissimilarity index called the Mean Absolute Difference of Distances (MADD), and they show that MADD-based clustering methods have better performance than clustering algorithms based on the MDP distance or the Euclidean distance. Modarres (2022) creates a modified version of MADD called Mean Absolute Differences of Modified Distances (MADMD) and shows that it obtains similar results in HDLSS settings. A different distance measure is proposed in Terada (2013), who proposes a transformation of the Euclidean distance matrix, using Distance Vectors. In the paper, it is shown both theoretically and experimentally that Distance Vector clustering outperforms classical methods. Y. Liu et al. (2008) provide a very different approach to clustering HDLSS data, using statistical tests to determine the significance of clustering. Their method, called significance of clustering (SigClust), though often used, uses the strong assumption of normality of the data, which often poses a problem. Therefore, Valk and Cybis (2021) introduce a new clustering method using less challenging assumptions, called U-statistical Clustering (Uclust). In their paper, they demonstrate the superiority of Uclust over SigClust. All methods that are used in our research are described in detail in Section 4.

Finally, after finding the best (combinations of) methods, we want to compare the best clusters of the dimensionally reduced and nondimensionally reduced HDLSS data, trying to find if there is a significant difference between the two.

3 Data

Having introduced HDLSS data, a formal definition is given as follows. Given an $n \times p$ dataset X, X is an HDLSS dataset if $p > n$. In fact, in practice, it often holds that $p >> n$. HDLSS data is especially common in medical applications, specifically in so-called microarray data. Microarray data is a collection of gene expression levels, extracted from genetic DNA or RNA material. Using these gene expression levels, experts are able to identify certain patterns that gain insights into how genes may be related to specific diseases. While the resulting datasets contain thousands of gene expression levels, scientists often are able to collect up to only a few hundred observations, due to natural limits in time, budget and experimental subjects. Therefore, microarray data is a commonly used data source for examining the performance of (machine learning) methods on HDLSS data. Further details regarding microarray data analysis can be found in Quackenbush (2001).

The main data used in our research was extracted from the R package `datamicroarray`, from which we used 9 pre-processed, pre-classified, and ready-to-use datasets. Implemented in the package, in alphabetic order, we used the free and openly available data from Alon et al. (1999), Christensen et al. (2009), Gravier et al. (2010), Khan et al. (2001), Pomeroy et al. (2002), Shipp et al. (2002), Sørlie et al. (2001), Su et al. (2002) and West et al. (2001). For details regarding each dataset, we refer to the respective paper. An overview of the datasets is given in Table 1. Note that not all datasets refer to a specific disease, but rather some biological condition, which is denoted by N/A.

Furthermore, reproducing the work in the paper of Renjith et al. (2021), we use the Jester dataset 1 (Goldberg et al., 2001). It contains a large amount of joke ratings from users (-10 to +10), extracted from an online joke recommender system. We first delete all observations with missing data, and then randomly sample 5000 observations, similar to Renjith et al. (2021). This is no HDLSS data, and it should therefore be noted that we do not use the results in answering our research question. Further properties of the dataset are described in Table 1.

Paper	n	p	Classes	Disease
Goldberg et al. (2001)	73.421	100	Unknown	-
Alon et al. (1999)	62	2000	2	Colon cancer
Christensen et al. (2009)	217	1413	3	N/A
Gravier et al. (2010)	168	2905	2	Breast cancer
Khan et al. (2001)	63	2308	4	SRBCT
Pomeroy et al. (2002)	60	7128	2	CNS tumor
Shipp et al. (2002)	58	6817	2	Lymphoma
Sorlie et al. (2001)	85	456	5	Breast cancer
Su et al. (2002)	102	5565	4	N/A
West et al. (2001)	49	7129	2	Breast cancer

Table 1: Overview of the datasets used in this research

4 Methodology

4.1 Replication and extension of Renjith et al. (2021)

With this research, we aim to reproduce and extend the paper by Renjith et al. (2021), and we propose some adjustments to their work. In their research, they perform a comparative analysis of combinations of several often used DRTs and clustering methods and provide a general series of steps to take to perform a similar experiment for various research purposes. In particular, they suggest the following steps:

1. Determine the optimal cluster count using the `NbClust` R package.
2. Transform data using several DRTs; PCA, ICA, t-SNE and LLE. Retain a non-transformed dataset.
3. Cluster each dataset resulting from the previous step with k-Means and AGNES.
4. Internally evaluate cluster quality of all clusters.

As a first step in our research, we replicate their work using the same data and methods. We swap steps 1 and 2 in our research, as to us, it makes more sense to determine the optimal cluster count after transforming the dataset, rather than before. It should be noted that in their paper, while they do aim to offer clear guidance on how to perform a similar experiment, they often lack exact specifications of their methods. Just one example of how their paper lacks reproducibility is their use of the `AGNES` (Agglomerative Nesting) package in R. It provides a wide range of linkage methods, but it is not specified in the paper which of these methods they use. Additionally, it is not mentioned how hyperparameters were chosen for t-SNE and LLE. Also, while we try to use the packages that are used in their paper, some of these have been deprecated and/or are inapplicable to HDLSS data, one of which is `NbClust`. Therefore, we manually determine the optimal cluster count by calculating the internal indices of step 4 for all (non)dimensionally reduced clustering methods for a range of cluster counts, and we use the cluster count that leads to the best internal index value for each individual method. Then, we use a majority vote to determine the optimal cluster count.
After replicating their work, we extend it in several ways. First, we use the road map they provide and extrapolate it to an HDLSS setting, repeating a similar experiment for 9 HDLSS datasets, such that stronger conclusions about the superiority of methods can be made. Second, as we use HDLSS data in our research, we extend the DRTs and clustering methods they propose. While the methods they propose are generally good for the $n > p$ Jester data, the literature described in Section 2 tells us these methods don't perform well in HDLSS settings, and different methods should be used. These methods are described in detail in Sections 4.2 and 4.3. Third, as we have access to the true class labels with our data, we do both internal and external cluster quality evaluation. For external cluster evaluation, we skip step 1, as the optimal cluster count is known to be equal to the true number of classes.

4.2 Dimensionality Reduction Techniques

In this section, we discuss the techniques used to dimensionally reduce the HDLSS datasets prior to clustering.

4.2.1 Principal Component Analysis (PCA) and CHull

First proposed in Hotelling (1933), PCA is the most well-studied and widely used DRT. PCA decomposes the data into principal components; orthogonal, linear combinations of the original variables that preserve the variation of the original data as much as possible. Given a dataset $\mathbf{X}$, it follows that the principal components are the eigenvectors of the matrix $\frac{1}{n}\mathbf{X}\mathbf{X}^T$. We refer to Shlens (2014) for the proof and further detailed analyses. To implement PCA, we use the method `prcomp` from the `stats` R package.

There are many ways to determine the amount of principal components to retain. A numeric method that can be used to optimally choose the amount of principal components is called CHull (Wilderjans et al., 2013). As a generic method, it is designed to optimally balance the goodness of fit f and model complexity c. For PCA, the goodness of fit is the explained variance, and the model complexity is the number of principal components to retain.
For each triplet of adjacent models (m_i, m_j, m_k), m_j is excluded if $f_j \leq f_i + (c_j - c_i)\frac{f_k - f_i}{c_k - c_i}$, resulting in a set of models that lie above the line connecting the two other models m_i and m_k (i.e., lie on the upper boundary of the convex hull). Then, the model is chosen with the highest st value:

$$st_i = \frac{\frac{f_i - f_{i-1}}{c_i - c_{i-1}}}{\frac{f_{i+1} - f_i}{c_{i+1} - c_i}} \tag{1}$$

As the numerator and denominator are the slopes of the line connecting two adjacent models, a large value implies a large increase in model fit going from m_{i-1} to m_i, and a not so large increase going from m_i to m_{i+1}. Therefore, the model is chosen where the increase in goodness of fit levels off the most. An implementation (CHULL) can be downloaded from the website of the authors as MATLAB code or standalone software.

4.2.2 Independent Component Analysis (ICA)

ICA, like PCA, is one of the most common linear DRTs. ICA assumes that the observed variables $\mathbf{X}$ are linear combinations of underlying, non-Gaussian and independent components $\mathbf{S}$, and it aims to find these components. In matrix notation, ICA assumes that $\mathbf{X} = \mathbf{S}\mathbf{A}$, where $\mathbf{A}$ is a linear mixing matrix. ICA then attempts to get the underlying components $\mathbf{S}$ by estimating an un-mixing matrix $\mathbf{W}$ such that $\mathbf{X}\mathbf{W} = \mathbf{S}$. Due to the Central Limit Theorem, we know that combinations of non-Gaussian components tend to be more Gaussian. Therefore, to "reverse" the Central Limit Theorem, ICA searches for a $\mathbf{W}$ such that the non-Gaussianity of the components is maximised. For a more detailed description of ICA, we refer to Hyvärinen and Oja (2000). In that same paper, the authors propose a fast algorithm (FastICA) for maximising non-Gaussianity. We use the method `fastICA` in R to implement this fast algorithm, using all default settings, and retaining the same amount of components as PCA. Note that even though we use the "fast" algorithm, computation times were still huge and we had to skip ICA dimension reduction for several datasets for that reason.

4.2.3 t-Distributed Stochastic Neighbour Embedding (t-SNE)

Unlike ICA and PCA, t-SNE is a nonlinear DRT. For every observation, it creates a similarity score p_{ij} in the original space using $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$, where, given a Euclidean distance matrix D:

$$p_{j|i} = \frac{exp(-\|D_{ij}\|^2/2\sigma_i^2)}{\sum_{k\neq i} exp(-\|D_{ij}\|^2/2\sigma_i^2)} \tag{2}$$

It also creates a similarity score q_{ij} in the low-dimensional space using the Cauchy distribution (t-distribution with 1 degree of freedom):

$$q_{ij} = \frac{(1+\|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k\neq l}(1+\|\mathbf{y}_k - \mathbf{y}_l\|^2)^{-1}} \tag{3}$$

For both similarity scores, higher scores/probabilities correspond to similar data points. Then, the embedding onto the low-dimensional space is learned by changing the location of the $\mathbf{y}$ objects, with the objective of minimising the Kullback-Leibler divergence between the distributions q_{ij} and p_{ij}. Because this minimisation is generally computationally expensive, Van Der Maaten (2013) provide an alternative implementation called Barnes-Hut-SNE, which runs much faster at the cost of slightly less exact approximations. We make use of this Barnes-Hut-SNE implementation using the R package `Rtsne`, with the standard `theta` setting, which controls the speed/accuracy trade-off. We map the data onto the same amount of dimensions as ICA and PCA. For t-SNE, the most important parameter to tune is `perplexity`, for which we take $\sqrt{n}$, based on the intuitive analysis in Appendix B. Finally, using a plot, we change the number of iterations `max_iter` until we observe clear distinctions/clusters in the output data.

4.2.4 Locally Linear Embedding (LLE)

LLE is another nonlinear DRT, and it is the final standard DRT proposed in Renjith et al. (2021). LLE is performed in three distinct steps, with as input an $n \times p$ matrix $\mathbf{X}$. In the first step, the neighbours of each data point $\mathbf{x}_i$ are determined. The second step is then to compute the weights w_{ij} that best reconstruct each data point, minimising $E(W) = \sum_i |\mathbf{x}_i - \sum_j w_{ij}\mathbf{x}_j|^2$. Finally, in the third step, the embedding coordinates $\mathbf{y}_i$ that are best reconstructed by the weights $\mathbf{w}_{ij}$ are computed, minimising the embedding cost function $\Phi(Y) = \sum_i |\mathbf{y}_i - \sum_j w_{ij}\mathbf{y}_j|^2$. The method is implemented using the `do.lle` method in the R package `Rdimtools`. It offers built-in regularisation parameter tuning based on the literature, and we refer to Roweis and Saul (2000) for more details regarding both the algorithm and hyperparameter tuning. Again, we choose the same amount of target dimensions as PCA, ICA and t-SNE.

4.2.5 Gaussian kernel Principal Component Analysis (Gaussian kPCA)

Gaussian kPCA is the first DRT that should be effective for HDLSS data, specifically for clustering (Nakayama et al., 2021). Similar to PCA, kernel PCA (kPCA) is a DRT that relies on eigenvalue decomposition. Suppose we are given a data vector $\mathbf{x}_i \in R^n$, $i = 1, ..., n$, and Φ an implicit nonlinear mapping of the data from the data space R^n into the higher-dimensional feature space F: $R^n \to F$, $\mathbf{x} \to \Phi(\mathbf{x})$. Then, the dot product of $\Phi(\mathbf{x}_i)$ and $\Phi(\mathbf{x}_j)$ for any i, j

in F can be computed with the kernel function $k(\mathbf{x}_i, \mathbf{x}_j) = \Phi^T(\mathbf{x}_j)\Phi(\mathbf{x}_i)$. As the choice of Φ is implicit, the choice of the functional form of $k(\mathbf{x}_i, \mathbf{x}_j)$ is free. We use the Gaussian kernel, such that $k(\mathbf{x}_i, \mathbf{x}_j) = exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/\gamma), \gamma > 0$. Then, the eigenvectors of the resulting matrix $\mathbf{K}$ are determined, and the eigenvectors with the largest eigenvalues are retained. For a more detailed analysis of kPCA, we refer to Zheng et al. (2005).
In their paper, Nakayama et al. (2021) study asymptotic properties of kPCA in HDLSS settings, and they prove that for 2 and 3 underlying clusters, asymptotically, as $p \to \infty$ with n fixed, observations can be perfectly clustered based on the sign of their PC scores. While no proofs are given for populations with more than 3 underlying clusters, they imply that the same results should apply for $k \geq 4$. Furthermore, the authors show that the scale parameter γ is important for Gaussian kPCA to give good results for clustering purposes. In their paper, they provide R code to optimally choose this parameter, and this code was used in our research. The number of eigenspaces we retain is the same as for the previously mentioned DRTs.

4.2.6 Variational Autoencoder (VAE)

A very different approach to dimensionality reduction uses a Variational Autoencoder (VAE). It is a neural network-based method, with applications in generative modelling of images and videos, image manipulation, outlier detection, data imputation and finally, dimensionality reduction. Given an input dataset $\mathbf{X}$, the goal of VAE is to map this dataset to a latent space $\mathbf{Z}$, from which PDF $P(\mathbf{z})$ we can then sample to get, with high probability, samples that are close to the original data in $\mathbf{X}$. Mathematically, we want to optimise the parameters θ in a family of deterministic functions $f(\mathbf{z}; \theta)$, $f : \mathbf{Z} \times \mathbf{\Theta} \to \mathbf{X}$ such that we can sample $\mathbf{z}$ from $P(\mathbf{z})$ and get $f(\mathbf{z}; \theta)$ similar to $\mathbf{X}$. Therefore, we want to maximise $P(\mathbf{X})$, given by:

$$P(\mathbf{X}) = \int P(\mathbf{X}|\mathbf{z}; \theta)P(\mathbf{z})d\mathbf{z} \tag{4}$$

Where $P(\mathbf{X}|\mathbf{z}; \theta) = N(\mathbf{X}|f(\mathbf{z}; \theta), \sigma^2 I)$. The optimisation of the integral in Equation 4 uses a new function $Q(\mathbf{z}|\mathbf{X})$, which takes a value of $\mathbf{X}$ as input, and produces a distribution over $\mathbf{z}$ values that are likely to produce similar values to $\mathbf{X}$. Starting from the definition of the Kullback-Leibler divergence (KL) between $P(\mathbf{z}|\mathbf{X})$ and $Q(\mathbf{z})$, for some arbitrary Q:

$$KL(Q(\mathbf{z})\|P(\mathbf{z}|\mathbf{X})) = E_{\mathbf{z}\sim Q}(logQ(\mathbf{z}) - logP(\mathbf{z}|\mathbf{X})) \tag{5}$$

Doersch (2016) shows that this can be rewritten to:

$$logP(\mathbf{X}) - KL(Q(\mathbf{z}|\mathbf{X})\|P(\mathbf{z}|\mathbf{X})) = E_{\mathbf{z}\sim Q}(logP(\mathbf{X}|\mathbf{z})) - KL(Q(\mathbf{z}|\mathbf{X})\|P(\mathbf{z})) \tag{6}$$

Relating this to Equation 4, we see that the equation on the left in Equation 6 is the term that we want to maximise; $logP(\mathbf{X})$ plus an error term, which makes Q produce $\mathbf{z}$'s such that they can reproduce $\mathbf{X}$ well. It is usual practice to take $Q(\mathbf{z}|\mathbf{X}) = N(\mathbf{z}|\mu(\mathbf{X}; \tilde{\theta}), \Sigma(\mathbf{X}; \tilde{\theta}))$, where μ and Σ are deterministic functions that are learned via neural networks. Using the technique called the "reparameterization trick", we can optimise Equation 6, learning the functions μ and Σ, such that we can use the function $Q(\mathbf{z}|\mathbf{X})$ to dimensionally reduce our data $\mathbf{X}$ to a lower-dimensional

latent space $\mathbf{Z}$.

In their paper, Mahmud et al. (2021) implement a VAE as a DRT and compare its performance to traditional DRTs. In their research, they use similar microarray data and the structure and hyperparameter choices of their VAE are described in detail. Focusing their research on the VAE, they optimise the structure and hyperparameters for optimal results, and therefore, given that the data is of a similar nature as ours, we copy their VAE structure. We first "regularly" normalise our data. Then, in the first layer of the encoder, we apply batch normalisation (Santurkar et al., 2018). We use a second intermediate layer, mapping the data to $0.1p$, with p the initial amount of features. Then, we map the data to the low-dimensional latent space $\mathbf{Z}$, where the dimension of $\mathbf{Z}$ is one of $[2, 10, 50, 100, 200, 300]$. The decoder has the inverse structure (without batch normalisation). We use a batch size of 100, 200 epochs, and Adam optimisation with a learning rate of 0.0005. Based on external validation indices, we use the number of latent dimensions that results in the best cluster quality. Even though we store more information in the dimensionally reduced data than the other DRTs this way, due to the mathematical nature of the method, we believe this is the right procedure for choosing the optimal VAE. The VAE was built using the `Keras` package in Python, using Google Colab's free GPU connection.

4.2.7 Network-based Dimensionality reduction Analysis (NDA)

The final DRT we will use is NDA. Being a network-based method, it is again very different from previously proposed DRTs. Introduced in Kosztyán et al. (2022), it is the first nonparametric DRT solution for HDLSS data. NDA is performed in three steps.

In the first step, the correlation graph between features is specified. Denote here $G(\mathbf{N}, \mathbf{A}, \mathbf{W})$ as the undirected weighted correlation graph, where $\mathbf{N}$ is the set of nodes, $\mathbf{A}$ the set of arcs, $\mathbf{W}$ the set of arc weights, and node i represents feature i, $i = 1, 2, ..., p$. The weight of an arc is defined as the squared correlation between two features ($\rho_{i,j}^2 = w_{i,j} \in \mathbf{W}$). For $\rho_{i,j}$, we take Pearson's correlation coefficient between two features $(\mathbf{v}_i, \mathbf{v}_j)$:

$$\rho_{i,j} = \frac{E((\mathbf{v}_i - \mu_i)(\mathbf{v}_j - \mu_j))}{\sigma_i \sigma_j} \tag{7}$$

In the second step, we apply modularity-based community detection to $G(\mathbf{N}, \mathbf{A}, \mathbf{W})$, which aims to find network modules, subgraphs whose vertices are more likely to be connected than those outside the graph. NDA uses the modularity measure introduced in Newman (2006), in combination with the Leiden algorithm for community detection (Traag et al., 2019). For details regarding Newman's modularity measure and the Leiden algorithm, we refer to the respective papers.

The third and final step involves calculating the latent variables using the eigenvector centrality (EVC). The EVC measures the influence of a node in a network and assigns relative scores to all nodes in the network. A high eigenvector score means that that particular node is connected to many nodes with high eigenvector scores themselves. The EVC for feature $\mathbf{v}_i$ is given by $c_i = \frac{1}{R} \sum_j r_{i,j} c_j$, where R is a constant, and $r_{i,j}$ the edge weight, here the squared correlation, between nodes (features) i and j. Then, the latent variable score LV for the module C_I is given

by:

$$LV_I = \frac{\sum_{i \in C_I} c_i \mathbf{z}_i}{\sum_{i \in C_I} c_i} \tag{8}$$

where $\mathbf{z}_i = (\mathbf{v}_i - \mu_i)/\sigma_i$ is the standardised feature $\mathbf{v}_i$.
Due to the nonparametric nature of modularity-based community detection, the number of modules can not be influenced (negatively). Further dimensionality reduction can be achieved by feature selection though, but as we do not perform feature selection for any other method, we did not use this procedure to further reduce the number of dimensions to retain. Therefore, while we have restricted other DRTs (except for the VAE) to retain the same number of dimensions for a fair comparison, this is not appropriate for NDA. Furthermore, NDA allows for the usage of orthogonal rotation methods (Browne, 2001). While these rotation methods aim to achieve a more clearly separated factor structure, this may give adverse effects if the unrotated LV matrix is highly correlated. As this is unclear prior to performing the method, we use both a rotated and non-rotated dataset. NDA was implemented using the R package `nda`.

4.3 Clustering methods

In this section, we discuss the clustering methods, where standard clustering methods (k-Means and AGNES) are used to cluster both dimensionally reduced and nondimensionally reduced HDLSS data, and the other HDLSS-specific clustering methods are used to cluster nondimensionally reduced HDLSS data only.

4.3.1 k-Means Clustering

k-Means is the most often used clustering method, first introduced in Steinhaus et al. (1956). Given a set of p-dimensional observations $(\mathbf{x}_1, ..., \mathbf{x}_n)$, k-means clusters these observations into k sets $\mathbf{S} = \{\mathbf{S}_1, ..., \mathbf{S}_k\}$, where the within-cluster sum of squares is minimised:

$$min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \in \mathbf{S}_i} \|\mathbf{x} - \mu_i\|^2 \tag{9}$$

Where $\|\cdot\|$ is the Euclidean norm. We use the specific k-Means algorithm of Hartigan et al. (1979), which we refer to for details. It is implemented in the R package `stats` using the function `kmeans`. This is the same algorithm used in Renjith et al. (2021).

4.3.2 k-Means Clustering with Mean of Absolute Differences of Distances (MADD)

While k-Means is intuitive and easy to implement, Hall et al. (2005) show that common clustering methods based on the Euclidean distance suffer in HDLSS settings. Therefore, Sarkar and Ghosh (2019) propose a new dissimilarity index, MADD, which they show is effective in HDLSS situations. As they show that MADD is a dissimilarity index, rather than a distance metric or a norm, we need a new definition for the k-Means criterion described in Section 4.3.1, such that we can use any dissimilarity matrix to perform k-Means, an example of which is given in Vera and Macías (2021). Consider $\mathbf{\Delta}$ any $n \times n$ symmetric dissimilarity matrix related to the initial $n \times p$ data matrix $\mathbf{X}$ and denote $\mathbf{\Delta^2}$ for the squared dissimilarities matrix. Furthermore, let $\mathbf{E}$

be an $n \times k$ partition matrix of $\mathbf{\Delta}^2$ in k clusters, whose elements e_{ik} are 1 if row i belongs to cluster k and zero otherwise. Then, they show that under certain conditions, it holds that:

$$min_{\mathbf{E}} W(\mathbf{E}|k, \mathbf{\Delta}^2) = min_{\mathbf{E}} \sum_{l=1}^{k} \frac{1}{2n_l} \sum_{i=1}^{n} \sum_{j=1}^{n} e_{il} e_{jl} \delta_{ij}^2 \tag{10}$$

defines an equivalent k-Means clustering criterion for $\mathbf{\Delta}^2$ to that of $\mathbf{X}$. Intuitively, and equivalent to regular k-Means, Equation 10 shows that we minimise the within-cluster sum of squared dissimilarities. The MADD dissimilarities δ_{ij} proposed in Sarkar and Ghosh (2019) between rows $\mathbf{x}_i$ and $\mathbf{x}_j$ are given by:

$$\delta_{ij} = \frac{1}{n-2} \frac{1}{\sqrt{p}} \sum_{z \in \mathbf{X} \setminus \mathbf{x}_i, \mathbf{x}_j} | \, \|\mathbf{x}_i - \mathbf{z}\| - \|\mathbf{x}_j - \mathbf{z}\| \, | \tag{11}$$

We implemented k-Means with MADD using the R package `HDLSSkST`, with the `gMADD` method using all default settings.

4.3.3 Agglomerative Hierarchical Clustering (AGNES)

AGNES is a bottom-up hierarchical clustering method; each observation starts as a cluster itself, then clusters are merged moving up the hierarchy until one large cluster remains. At every iteration, the two nearest clusters are combined, based on the distance between each cluster, called the linkage. There exist a series of linkage methods, both simple and more advanced. Distances between single observations are determined with methods such as the Euclidean distance, which is our distance measure of choice. As the linkage method, we use Unweighted average linkage:

$$\frac{1}{|\mathbf{A}| \cdot |\mathbf{B}|} \sum_{\mathbf{a} \in \mathbf{A}} \sum_{\mathbf{b} \in \mathbf{B}} d(\mathbf{a}, \mathbf{b}) \tag{12}$$

Where $\mathbf{A}$ and $\mathbf{B}$ are two clusters, and $d(\mathbf{a}, \mathbf{b})$ is the Euclidean distance. For further details, we refer to Kaufman and Rousseeuw (2009). AGNES is implemented in the R package `stats` using the function `agnes`, the choice of methods is, again, the same as Renjith et al. (2021).

4.3.4 Distance Vector Clustering

Quite similar to MADD, Distance Vector clustering proposes a new distance measure, replacing the standard Euclidean distance in standard clustering methods. Terada (2013) proposes this new, Euclidean distance-based measure, and proposes to use it with standard hierarchical clustering, using (for example) Unweighted Average or Single Linkage. Given a centred $n \times p$ data matrix $\mathbf{X}$, and its $n \times n$ Euclidean distance matrix $\mathbf{D}$, the paper proposes the distance matrix $\mathbf{\Xi} = (\xi_{ij})_{n \times n}$, with $\xi_{ij} = \sqrt{\sum_{t \neq i,j} (d_{it} - d_{jt})^2}$. Then, using this distance measure, we use both Unweighted Average and Single Linkage to cluster with AGNES. With $\mathbf{A}$ and $\mathbf{B}$ two clusters, and $d(\cdot, \cdot)$ the Distance Vector distance between two observations, the Single Linkage criterion is given by $min_{\mathbf{a} \in \mathbf{A}, \mathbf{b} \in \mathbf{B}} d(\mathbf{a}, \mathbf{b})$. The Unweighted Average criterion is given in Equation 12. We calculated the Distance Vector distance matrix in R using the above-described algorithm, and hierarchical clustering was performed using the R function `agnes`.

4.3.5 U Clustering (Uclust) and U-Hierarchical Clustering (UHclust)

An alternative approach to clustering, Uclust, uses statistical tests to iteratively find clusters in the data with significant differences. Let $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ be a sample of p-dimensional vectors divided into groups $\mathbf{G}_1$ and $\mathbf{G}_2$ of sizes n_1 and n_2, where $n = n_1 + n_2$. Valk and Cybis (2021) then show that an unbiased estimator of the within-group distance is a generalised one-sample U-statistic $U_{n_g}^{(g)}$ (Hoeffding & Robbins, 1948). Similarly, an unbiased estimator of the between-group distance is a generalised two-sample U-statistic $U_{n_1,n_2}^{(1,2)}$. Then, they show that it holds that:

$$U_n = \sum_{g=1}^{2} \frac{n_g}{n} U_{n_g}^{(g)} + \frac{n_1 n_2}{n(n-1)}(2U_{n_1 n_2}^{(1,2)} - U_{n_1}^{(1)} - U_{n_2}^{(2)}) = W_n + B_n \tag{13}$$

Which leads to the statistic $B_n = \frac{n_1 n_2}{n(n-1)}(2U_{n_1 n_2}^{(1,2)} - U_{n_1}^{(1)} - U_{n_2}^{(2)})$. It's easy to see that high values of B_n correspond to high values for the between-group distance, and low values for the within-group distances. Using a series of assumptions explained in Valk and Cybis (2021), the authors then show that as $p \to \infty$:

$$\frac{B_n}{\sqrt{var(B_n)}} \to N(0, 1) \tag{14}$$

Uclust uses this asymptotic property of B_n to test for homogeneity among each partition. Eventually, the partition with the highest value for $B_n/\sqrt{var(B_n)}$ among all significant partitions is chosen, and the initial sample $\mathbf{X}$ is split into two groups $\mathbf{G}_1$ and $\mathbf{G}_2$. If we wish to split the sample into a known number of k groups, we can iteratively perform Uclust on each group, choosing the partition with the highest B_n statistic, until we reach k groups. This procedure is close to what the hierarchical version of Uclust, UHclust, does; it keeps partitioning the groups moving up the hierarchy, choosing the partition with the highest B_n statistic until no significant partitions remain. For further details and proofs regarding Uclust and UHclust, we refer to Valk and Cybis (2021).

Uclust and UHclust are implemented in the R package `uclust`. For external cluster validation, we used the method `uclust`, and iteratively split the sample, partitioning with the largest B_n statistic, until we reached the known number of classes. For internal cluster validation, we used the method `uhclust`, testing at `alpha` $= 0.05$, splitting until no significant partition was found.

4.4 Cluster Validation

With our research, we try to answer to what extent clustering methods designed for HDLSS data outperform standard clustering methods for dimensionally reduced HDLSS data. Therefore, we need cluster validation indices to assess clustering performance. Because we have access to the true class labels of our data, we can perform external cluster validation, which assigns a value to the classification of observations, relative to the true classifications. More often used in practice, because in general one does not have access to the true class labels of observations, we also perform internal cluster validation, which assigns a value to the geometric structure of clusters. A wide range of internal indices exist, though it is unclear which of these indices is best. Therefore, we use 4 popular internal indices, which are the same indices that are used in Renjith et al. (2021). For external validation, Steinley (2004) shows the properties of

the Adjusted Rand Index for cluster validation and argues its superiority over other often-used external indices. Therefore, we only use the Adjusted Rand Index for external cluster validation.

4.4.1 External validation

Consider a set of n observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ and suppose that $\mathbf{U} = \{\mathbf{U}_1, ..., \mathbf{U}_k\}$ are the resulting clusters of the clustering methods, and $\mathbf{V} = \{\mathbf{V}_1, ..., \mathbf{V}_k\}$ are the true clusters, according to each observations' class label. Then, consider each $\binom{n}{2}$ possible combinations of pairs of observations $\mathbf{x}_i$ and $\mathbf{x}_j$ in $\mathbf{X}$. We can then define the following values:

- a: the number of pairs in $\mathbf{X}$ that are in the *same* cluster in $\mathbf{U}$ and in the *same* cluster in $\mathbf{V}$
- b: the number of pairs in $\mathbf{X}$ that are in the *same* cluster in $\mathbf{U}$ and in *different* clusters in $\mathbf{V}$
- c: the number of pairs in $\mathbf{X}$ that are in *different* clusters in $\mathbf{U}$ and in the *same* cluster in $\mathbf{V}$
- d: the number of pairs in $\mathbf{X}$ that are in *different* clusters in $\mathbf{U}$ and in *different* clusters in $\mathbf{V}$

Note that these definitions can be related to True (False) Positive (Negative) classifications. Then, a first intuitive index is the Rand Index, where $RI = \frac{a+d}{a+b+c+d} = \frac{a+d}{\binom{n}{2}}$. It is easily observed that this can be interpreted as the percentage of correctly clustered observations. Its corrected-for-chance version proposed in Hubert and Arabie (1985) is called the **Adjusted Rand Index (ARI)**, which is the index that we will be using for external validation. The ARI can take values between -1 and 1, with 0 implying equal performance to random clustering, and 1 perfect clustering. Using the previously defined definitions, the ARI is given by:

$$ARI = \frac{\binom{n}{2}(a+d) - [(a+b)(a+c)+(c+d)(b+d)]}{\binom{n}{2}^2 - [(a+b)(a+c)+(c+d)(b+d)]} \tag{15}$$

We implemented the ARI using the `ClusterR` R package, with the `external_validation` method.

4.4.2 Internal validation

Now, we consider a set of n observations $\mathbf{X} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ and suppose that $\mathbf{U} = \{\mathbf{U}_1, ..., \mathbf{U}_k\}$ are the resulting clusters of the clustering methods, using the number of clusters that we have found to be optimal. The first internal validation index we use is the **Silhouette Index**, introduced in Rousseeuw (1987). For an observation $\mathbf{x}_i \in \mathbf{U}_I$, define $a(\mathbf{x}_i)$ as the mean (Euclidean) distance of $\mathbf{x}_i$ to all other observations in the same cluster $\mathbf{U}_I$. Furthermore, define $b(\mathbf{x}_i)$ as the *smallest* mean dissimilarity of $\mathbf{x}_i$ to another cluster $\mathbf{U}_J$, using the mean distance from $\mathbf{x}_i$ to all observations in $\mathbf{U}_J$, where $\mathbf{U}_J$ is the cluster that has the minimum mean distance:

$$b(\mathbf{x}_i) = min_{J \neq I} \frac{1}{|\mathbf{U}_J|} \sum_{\mathbf{x}_j \in \mathbf{U}_J} d(\mathbf{x}_i, \mathbf{x}_j) \tag{16}$$

Then, the Silhouette value of observation $\mathbf{x}_i$ is given by:

$$s(\mathbf{x}_i) = \frac{b(\mathbf{x}_i) - a(\mathbf{x}_i)}{max\{a(\mathbf{x}_i), b(\mathbf{x}_i)\}}, \text{if } |\mathbf{U}_I| > 1 \tag{17}$$

Finally, the Silhouette Index is obtained by taking the mean Silhouette value of all observations. Small values for the Silhouette Index indicate small between-cluster dissimilarities and large within-cluster dissimilarities. Therefore, with boundaries $[-1, 1]$, higher values indicate better clustering. We implemented the Silhouette Index using `ClusterR`'s method `silhouette_of_clusters`.

The second internal index is the **Dunn Index**, proposed in Dunn (1973). It is defined as the ratio of the smallest distance between observations that are not in the same cluster, to the largest within-cluster distance. Defining the maximum within-cluster (Euclidean) distance as $\Delta_I = max_{\mathbf{x}_i, \mathbf{x}_j \in \mathbf{U}_I} d(\mathbf{x}_i, \mathbf{x}_j)$, and the minimum distance between observations not in the same cluster as $\delta(\mathbf{U}_I, \mathbf{U}_J) = min_{\mathbf{x}_i \in \mathbf{U}_i, \mathbf{x}_j \in \mathbf{U}_j} d(\mathbf{x}_i, \mathbf{x}_j)$, where $I \neq J$, the Dunn Index of the clustering is then given by:

$$Dunn = \frac{min_{I,J}\delta(\mathbf{U}_I, \mathbf{U}_J)}{max_I \Delta_I} \tag{18}$$

With a lower boundary at 0, larger values indicate better clustering. The Dunn Index was implemented using the package `clValid`, with the `dunn` method.

The third internal index, proposed in Caliński and Harabasz (1974), is the **Calinski-Harabasz Index**. It is based on the ratio between the between-cluster covariance and the within-cluster covariance. Defining the between-cluster covariance as $B(k) = \sum_{I=1}^{k} |U_I| \|\bar{\mathbf{x}_I} - \bar{\mathbf{x}}\|^2$, and the within-cluster covariance as $W(k) = \sum_{I=1}^{k} \sum_{J \neq I} \|\bar{\mathbf{x}_I} - \bar{\mathbf{x}_J}\|^2$, the Calinski-Harabasz Index is then given by:

$$Calinhara = \frac{B(k)(n-k)}{W(k)(k-1)} \tag{19}$$

Higher values for the index imply high between-cluster differences and small within-cluster differences. Therefore, higher values indicate better clustering. The Calinski-Harabasz Index was implemented using the `fpc` package, with the method `calinhara`.

The final index is **Davies-Bouldin's Index**, proposed in Davies and Bouldin (1979). Given $\mathbf{A}_I$ the centroid of cluster $\mathbf{U}_I$, define the average within-cluster centroid distance $S_I = \frac{1}{|\mathbf{U}_I|} \sum_{\mathbf{x}_i \in \mathbf{U}_I} \|\mathbf{x}_i - A_I\|$, where $\|\cdot\|$ denotes the Euclidean norm. Furthermore, define the between-cluster centroid distance as $M_{IJ} = \|\mathbf{A}_I - \mathbf{A}_J\|$. Then, Davies-Bouldin's Index is defined as:

$$DB = \frac{1}{k} \sum_{I=1}^{k} max_{J \neq I} \frac{S_I + S_J}{M_{IJ}} \tag{20}$$

Here, low values for the Davies-Bouldin Index imply low within-cluster distances to the clusters' centroids and high distances between the centroids of each cluster. Therefore, lower values indicate better clustering. The Davies-Bouldin Index was implemented using the `clusterSim` package, with the `index.DB` method.

5 Results

In this section, the results of our research are presented. First, we briefly discuss the results of our replication of the research in Renjith et al. (2021). Then, we discuss the clustering results of the (non)dimensionally reduced HDLSS datasets, which are the results used to answer our research question: To what extent do clustering methods designed for HDLSS data outperform standard clustering methods for dimensionally reduced HDLSS data?

5.1 Replication of Renjith et al. (2021)

The results of our replication of the research in Renjith et al. (2021) are given in Table 2. Comparing the results of our replication to their work, we find some clear inconsistencies. In their research, they find that using the Jester dataset, dimensionality reduction with t-SNE results in the best clusters, with the best values for the Silhouette Index, the Dunn Index, and the Calinski-Harabasz Index. While in our results, the performance of dimensionality reduction with t-SNE is in general good, performance across indices is quite different and it is not easily said which DRT, or combination of DRT and clustering method, is best. Furthermore, in our research, it was found that the optimal cluster count is 2, while in their research the same optimal cluster count is 3. The reason why our results differ compared to their results may be due to the lack of explanation on the precise execution of their methods, which we discussed in Section 4.1. Furthermore, differences may be due to randomness. The main issue here lies with the fact that a random sub-sample of 5000 observations is taken from the initial dataset, and the random seed they use is unknown.

			PCA		ICA		t-SNE		LLE	
	k-Means	AGNES	k-Means	AGNES	k-Means	AGNES	k-Means	AGNES	k-Means	AGNES
Silhouette	0,14	0,35	0,37	**0,57**	0,22	0,41	0,46	0,43	0,22	0,43
Dunn	0,15	**0,32**	0,02	0,12	0,01	0,14	0,03	0,04	0,01	0,14
Calinski-Harabasz	188,18	14,56	828,43	29,65	273,49	4,64	**1397,76**	1055,18	276,16	4,86
Davies-Bouldin	2,30	1,15	1,09	**0,38**	1,87	0,46	0,84	0,80	1,90	0,45

Table 2: Internal cluster validation indices for clustering (non)dimensionally reduced data from the Jester dataset

5.2 Clustering (non)dimensionally reduced HDLSS data

A complete collection of computed cluster validation indices for all methods and datasets used are found in Appendix A, in Tables 6 to 14. We first want to answer the question: Which combinations of DRTs and clustering methods result in the best cluster quality for HDLSS data? An overview of the best 3 combinations of DRTs and clustering methods for both internal and external cluster validation for each dataset is given in Table 3. Clearly, it can be concluded that t-SNE is the best-performing DRT in our experiment. Both with internal and external validation, clustering data which is dimensionally reduced using t-SNE often results in one of the best clusters, more than any other DRT. Additionally, NDA, PCA and Gaussian kPCA

		Alon et al. (1999)	Christensen et al. (2009)	Gravier et al. (2010)	Khan et al. (2001)	Pomeroy et al. (2002)	Shipp et al. (2002)	Sorlie et al. (2001)	Su et al. (2002)	West et al. (2001)
Internal Validation Rank	1	NDA Rotated AGNES	PCA AGNES	t-SNE k-Means	Gaussian kPCA AGNES	NDA AGNES	t-SNE AGNES	t-SNE AGNES	t-SNE k-Means	NDA AGNES
	2	ICA AGNES	ICA AGNES	t-SNE AGNES	Gaussian kPCA k-Means	t-SNE AGNES	t-SNE k-Means	Gaussian kPCA k-Means	t-SNE AGNES	NDA Rotated AGNES
	3	NDA k-Means	ICA k-Means	NDA AGNES	t-SNE AGNES	Gaussian kPCA AGNES	PCA AGNES	LLE k-Means	NDA k-Means	PCA AGNES
External Validation Rank	1	Gaussian kPCA k-Means	PCA AGNES	Gaussian kPCA k-Means	t-SNE k-Means	t-SNE k-Means	VAE AGNES	PCA k-Means	t-SNE AGNES	t-SNE k-Means
	2	VAE k-Means	ICA AGNES	ICA k-Means	LLE k-Means	NDA Rotated k-Means	t-SNE AGNES	Gaussian kPCA AGNES	PCA AGNES	t-SNE AGNES
	3	t-SNE k-Means	ICA k-Means	PCA AGNES	NDA Rotated k-Means	NDA Rotated AGNES	t-SNE k-Means	t-SNE k-Means	PCA k-Means	LLE AGNES

Table 3: Top 3 best-performing combinations of dimensionality reduction techniques and clustering methods with internal and external cluster validation, for clustering of each dimensionally reduced dataset

often produce good clustering performance, both with internal and external clustering. While NDA without rotation has good clustering results with internal cluster validation, it is not once found to be in the top 3 of best combinations of DRTs and clustering methods for external cluster validation. Interestingly, NDA with rotation performs worse than NDA without rotation with internal validation, while it is clearly better with external validation. Also, while having obviously better performance than both LLE and VAE, the worst performing methods, ICA underperforms compared to the first mentioned DRTs. It should be noted though that ICA was excluded from the experiment for certain datasets, due to infeasible computation times. Lastly, while for internal validation, AGNES clustering clearly outperforms k-Means clustering, k-Means outperforms AGNES clustering with external validation. These differences in performance across internal and external validation imply that in general, better geometric structures of clusters, assessed with internal validation, do not necessarily mean more accurately classified clusters, assessed with external validation. Which of the two is more important depends on the context, and is up to the user to decide. Furthermore, Table 3 shows that we should not refrain from using classical DRTs for dimensionally reducing HDLSS data for clustering, and in fact, t-SNE and PCA show better or similar performance compared to most DRTs designed for dimensionally reducing data in HDLSS settings. The HDLSS-specific DRTs NDA and Gaussian kPCA show good performance as well, rather than VAE.

Next, we want to answer the question: Which clustering methods result in the best cluster quality for nondimensionally reduced HDLSS data? An overview of the 3 best clustering methods for HDLSS data, for both internal and external cluster validation for each dataset is given in Table 4. An obvious observation here is that AGNES has the best performance for internal cluster validation. For external validation though, its performance lacks, and its HDLSS-adjusted version using Distance Vectors is clearly better. Distance Vector clustering seems to be the best

		Alon et al. (1999)	Christensen et al. (2009)	Gravier et al. (2010)	Khan et al. (2001)	Pomeroy et al. (2002)	Shipp et al. (2002)	Sorlie et al. (2001)	Su et al. (2002)	West et al. (2001)
Internal Validation Rank	1	AGNES	AGNES	AGNES	AGNES	AGNES	k-Means MADD	AGNES	k-Means	AGNES
	2	Distance Vector Single	Distance Vector Avg	Distance Vector Single	Uclust	Distance Vector Avg	AGNES	k-Means	k-Means MADD	Distance Vector Avg
	3	k-Means	Distance Vector Single	Distance Vector Avg	Distance Vector Single	Distance Vector Single	Distance Vector Avg	Uclust	AGNES	Distance Vector Single
External Validation Rank	1	Distance Vector Avg	Distance Vector Avg	k-Means MADD	k-Means	Distance Vector Avg	k-Means	Uclust	k-Means	AGNES
	2	Distance Vector Single	Distance Vector Single	k-Means	Uclust	Distance Vector Single	Uclust	k-Means	Uclust	Distance Vector Avg
	3	k-Means MADD	Uclust	Distance Vector Avg	Distance Vector Avg	Uclust	Distance Vector Avg	Distance Vector Avg	AGNES	Distance Vector Single

Table 4: Top 3 best-performing clustering methods with internal and external cluster validation, for clustering of each nondimensionally reduced dataset

clustering method overall, showing good performance with both internal and external cluster validation. Generally, Distance Vector clustering using Unweighted average linkage is better than Distance Vector clustering using Single linkage, especially with external validation. On the contrary, k-Means clustering has better performance than its HDLSS-adjusted version using MADD as a dissimilarity index, k-Means with MADD having the least good performance out of any of the clustering methods. Uclust finally has good performance with external cluster validation, though it underperforms compared to all other methods with internal cluster validation. The issue with Uclust is that one needs access to the true number of clusters, as with external cluster validation, and use that to iteratively split the sample according to the split with the highest B_n statistic. Therefore, if we do not have access to the true number of clusters, as with internal cluster validation, UHclust can be used to split the sample until no significant partitions are found. In our experiment though, we have found that the number of clusters is grossly overestimated for all datasets. Similar to the results in Table 3, we again find significant differences between performances with external and internal validation, and it depends on the context of the problem which of the two is more appropriate. Overall, Table 4 shows that hierarchical clustering, either with AGNES or Distance Vector clustering, is better than k-Means clustering. If internal validation is more appropriate, AGNES has the best performance for clustering HDLSS data, while in the cases where external validation is equally or more appropriate, Distance Vector clustering using Unweighted average linkage is best. Additionally, we again find, similar to clustering dimensionally reduced HDLSS data, that HDLSS-specific methods do not necessarily outperform classical methods. Finally, it is found that Uclust is a good method if and only if the primary goal is to accurately classify observations, and the number of true clusters is known.

Finally, we aim to answer our research question: To what extent do clustering methods designed for HDLSS data outperform standard clustering methods for dimensionally reduced HDLSS data? Given the knowledge that clustering methods designed for HDLSS data do not neces-

sarily outperform classical methods in HDLSS settings, we are mainly interested in evaluating whether clustering HDLSS data has better overall performance than clustering dimensionally reduced HDLSS data. An overview of the best cluster validation indices for clustering all dimensionally and nondimensionally reduced HDLSS datasets is given in Table 5. With better index

		Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin	ARI
Alon et al. (1999)	no DR	0,36	0,50	22,22	0,90	0,00
	DR	**0,60**	**0,60**	**62,30**	**0,31**	**0,40**
Christensen et al. (2009)	no DR	0,41	**0,55**	121,44	1,01	0,99
	DR	**0,81**	0,37	**1490,75**	**0,31**	**1,00**
Gravier et al. (2010)	no DR	0,46	**0,80**	14,23	0,43	0,05
	DR	**0,63**	0,77	**500,92**	**0,32**	**0,15**
Khan et al. (2001)	no DR	0,17	0,65	8,06	0,73	**0,34**
	DR	**0,88**	**1,19**	**98798,58**	**0,10**	0,25
Pomeroy et al. (2002)	no DR	0,32	**0,66**	14,46	0,53	-0,01
	DR	**0,53**	0,38	**871,77**	**0,44**	**0,04**
Shipp et al. (2002)	no DR	**0,63**	**1,13**	13,05	**0,27**	0,16
	DR	0,42	0,24	**1012,37**	0,69	**0,18**
Sorlie et al. (2001)	no DR	0,11	**0,62**	7,68	0,74	**0,62**
	DR	**0,46**	0,43	**2175,79**	**0,69**	0,57
Su et al. (2002)	no DR	0,16	0,70	20,65	0,76	**0,95**
	DR	**0,69**	**0,95**	**262,73**	**0,40**	0,92
West et al. (2001)	no DR	0,31	0,81	5,63	0,56	0,00
	DR	**0,79**	**0,90**	**69,50**	**0,19**	**0,03**

Table 5: The best cluster validation indices for clustering each dataset, with and without dimensionality reduction (DR)

values highlighted in Table 5, it is evident that overall, dimensionally reduced clustering has better performance than nondimensionally reduced clustering, both with internal and external validation. It is interesting to see that there are usually large differences in internal index values between dimensionally reduced and nondimensionally reduced clustering, while the ARI does not differ as much. This is partly due to the nature of certain DRTs. For example, t-SNE produces highly unstable low-dimensional encodings, with slight adjustments in hyperparameters resulting in large scale differences, which may bias internal index values. In our research, it is found that especially the Calinski-Harabasz Index seems to be highly sensitive to these scale differences, resulting in enormous differences in index values across methods, while other indices only show small differences compared to other methods. Even though internal validation may be biased towards dimensionally reduced clustering, clustering dimensionally reduced HDLSS data generally seems to be slightly better than clustering nondimensionally reduced HDLSS data with external validation as well. Regardless, the results show that we can conclude that in terms of clustering performance, clustering methods on HDLSS data do not outperform standard clustering methods for dimensionally reduced HDLSS data.

Besides clustering performance, interpretation plays an important role in the clustering of HDLSS data in certain practical implementations, particularly with microarray data. In practice, the goal of microarray data clustering mainly lies with "class discovery" (Dopazo, 2006), both of similar samples and closely related genes. In that case, true class assignments are unknown, and therefore unsupervised, internal cluster validation is used. Furthermore, as it is important to get an idea of how certain genes interact, it is important that the observations from thousands of individual genes are somehow grouped to enable human interpretation. Therefore, given the practical use of cluster analysis on microarray data, clustering of dimensionally reduced data makes more sense. Together with the finding that in general, the clustering quality of dimensionally reduced microarray data is better than that of nondimensionally reduced microarray data, we recommend using combinations of DRTs and standard clustering methods for clustering of microarray data. In general, we believe that for any application where the interpretation of the clustered observations plays an important role, the results of our research suggest that one should use standard clustering methods on dimensionally reduced HDLSS data, instead of using HDLSS-specific clustering methods on non-dimensionally reduced HDLSS data.

6 Conclusions

In this paper, we try to answer the question of whether, and to what extent, clustering methods designed for HDLSS data outperform standard clustering methods on dimensionally reduced HDLSS data. Firstly, we found that clustering methods designed for HDLSS data do not necessarily outperform standard clustering methods in HDLSS settings. Secondly, we found that in general, clustering on HDLSS data does not outperform clustering on dimensionally reduced HDLSS data. In fact, we found that in terms of both performance and interpretability, clustering dimensionally reduced HDLSS data is better than clustering nondimensionally reduced HDLSS data, both with internal and external cluster validation.

Clustering dimensionally reduced HDLSS data, we found that with internal and external cluster validation, dimension reduction using t-SNE generally results in the best clusters. Other good DRTs are NDA, Gaussian kPCA and PCA. This goes to show that one should not refrain from using classical DRTs in clustering situations, as their performance is often similar to or better than HDLSS-specific methods. Furthermore, we found big differences between internal and external cluster validation performances across methods, which demonstrates that clusters with better geometric structures do not necessarily imply more accurately classified clusters.

Clustering nondimensionally reduced HDLSS data, we found that overall, hierarchical clustering is better than k-Means clustering. For applications where internal validation is most appropriate, the best clustering method is AGNES. For applications where external and internal validation is equally important, or external validation is more important, the best clustering method is Distance Vector clustering with Unweighted average linkage. Furthermore, we found that, similar to clustering dimensionally reduced HDLSS data, classical methods are not necessarily outperformed by their HDLSS-specific versions, and there are significant differences in performances between internal and external validation. Finally, we demonstrated that Uclust should only be used if the true clusters are known, and the goal is to cluster with the best accuracy.

Other than the performance of clustering methods on (non)dimensionally reduced HDLSS data,

we found that the Calinski-Harabasz Index is highly sensitive to scale. Therefore, our research demonstrates that the Calinski-Harabasz Index should not be used when comparing the internal performance of combinations of DRTs and clustering methods, as some DRTs, such as t-SNE, show large scale differences with slight hyperparameter adjustments.

Theoretically, our research contradicts the conclusion of Bouveyron and Brunet-Saumard (2014), who recommend refraining from dimensionality reduction when clustering high-dimensional data. Furthermore, our research contradicts the hypothesis that standard dimensionality reduction techniques and clustering methods should not be used on HDLSS data. This is perhaps one of the most interesting findings, given the vast amount of research that uses this hypothesis to build new methods that are supposed to perform well on HDLSS data specifically. Additionally, trying to replicate Renjith et al. (2021), we demonstrated how the lack of detailed method specification in their paper results in it being very hard, if not impossible, to replicate.

Further researchers may use the solutions in this research for choices failed to mention in Renjith et al. (2021), alongside our adjusted version of the road map they provide, for a similar analysis. Furthermore, as our research was limited both in terms of computing power and time, certain choices were made to speed up the research process. For example, while a vast amount of free-to-use HDLSS datasets exist, we were only able to repeat the research for 9 relatively small-sized datasets, with a number of columns up to about 7000. Even then, for the larger datasets used, ICA was not feasible due to extremely long computation times. Further research may use better hardware to be able to scale the experiment using a larger variety of HDLSS data, including datasets with larger dimensionality. Under the same time and computing power restrictions, hyperparameters, if applicable, were not tuned to optimality. Our implementation of the VAE copies the architecture and hyperparameters of previous research and was not tuned for each dataset. Also, we use a simple analysis for the hyperparameter optimisation of t-SNE. Therefore, further research may involve an even more thorough tuning of hyperparameters for every dataset. Moreover, showing substantial empirical proof that the Calinski-Harabasz Index is inappropriate for a similar experiment to ours, it would be of our interest to investigate whether there is a classification to be made in internal indices, with the aim of finding one index that has the most desirable properties, similar to ARI. Finally, it would be of our interest to investigate whether methods that simultaneously reduce dimensionality and cluster observations, such as Biclustering, result in better or worse cluster quality than the methods we have discussed. Including this type of clustering method would be an interesting subject for further research as well.

References

Ahn, J., Lee, M. H. & Yoon, Y. J. (2012). Clustering high dimension, low sample size data using the maximal data piling distance. *Statistica Sinica*, 443–464.

Alon, U., Barkai, N., Notterman, D., Gish, K., Ybarra, S., Mack, D. & Levine, A. (1999, June). Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proceedings of the National Academy of Sciences*, *96*(12), 6745–6750.

Bouveyron, C. & Brunet-Saumard, C. (2014). Model-based clustering of high-dimensional data: A review. *Computational Statistics & Data Analysis*, *71*, 52–78.

Browne, M. W. (2001). An overview of analytic rotation in exploratory factor analysis. *Multivariate behavioral research*, *36*(1), 111–150.

Cai, H., Qi, F., Li, J., Hu, Y., Zhang, Y., Cheung, Y.-m. & Hu, B. (2023). Uniform tensor clustering by jointly exploring sample affinities of various orders. *arXiv preprint arXiv:2302.01569*.

Caliński, T. & Harabasz, J. (1974). A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, *3*(1), 1–27.

Cheema, M. S., Eweiwi, A. & Bauckhage, C. (2015). High dimensional low sample size activity recognition using geometric classifiers. *Digital Signal Processing*, *42*, 61–69.

Christensen, B. C., Houseman, E. A., Marsit, C. J., Zheng, S., Wrensch, M. R., Wiemels, J. L., ... Kelsey, K. T. (2009, August). Aging and Environmental Exposures Alter Tissue-Specific DNA Methylation Dependent upon CpG Island Context. *PLOS Genetics*, *5*(8), e1000602.

Davies, D. L. & Bouldin, D. W. (1979). A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*(2), 224–227.

Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Dopazo, J. (2006). Functional interpretation of microarray experiments. *Omics: a journal of integrative biology*, *10*(3), 398–410.

Dunn, J. C. (1973). A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters.

Goldberg, K., Roeder, T., Gupta, D. & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, *4*, 133–151.

Gravier, E., Pierron, G., Vincent-Salomon, A., Gruel, N., Raynal, V., Savignoni, A., ... others (2010). A prognostic dna signature for t1t2 node-negative breast cancer patients. *Genes, chromosomes and cancer*, *49*(12), 1125–1134.

Hall, P., Marron, J. S. & Neeman, A. (2005). Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *67*(3), 427–444.

Hartigan, J. A., Wong, M. A. et al. (1979). A k-means clustering algorithm. *Applied statistics*, *28*(1), 100–108.

Hoeffding, W. & Robbins, H. (1948). The central limit theorem for dependent random variables.

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, *24*(6), 417.

Hubert, L. & Arabie, P. (1985). Comparing partitions. *Journal of classification*, *2*, 193–218.

Hyvärinen, A. & Oja, E. (2000). Independent component analysis: algorithms and applications. *Neural networks*, *13*(4-5), 411–430.

Kaufman, L. & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Khan, J., Wei, J. S., Ringnér, M., Saal, L. H., Ladanyi, M., Westermann, F., ... Meltzer, P. S. (2001, June). Classification and diagnostic prediction of cancers using gene expression

profiling and artificial neural networks. *Nature Medicine*, *7*(6), 673–679.

Kosztyán, Z. T., Kurbucz, M. T. & Katona, A. I. (2022). Network-based dimensionality reduction of high-dimensional, low-sample-size datasets. *Knowledge-Based Systems*, *251*, 109180.

Liu, Q., Chen, G., Kosorok, M. R. & Bair, E. (2014). Biclustering via sparse clustering. *arXiv preprint arXiv:1407.3010*.

Liu, Y., Hayes, D. N., Nobel, A. & Marron, J. S. (2008). Statistical significance of clustering for high-dimension, low–sample size data. *Journal of the American Statistical Association*, *103*(483), 1281–1293.

Mahmud, M. S., Huang, J. Z., Fu, X., Ruby, R. & Wu, K. (2021). Unsupervised adaptation for high-dimensional with limited-sample data classification using variational autoencoder. *Computing & Informatics*, *40*(1).

Mitra, S. & Banka, H. (2006). Multi-objective evolutionary biclustering of gene expression data. *Pattern Recognition*, *39*(12), 2464–2477.

Modarres, R. (2022). A high dimensional dissimilarity measure. *Computational Statistics Data Analysis*, *175*, 107560. Retrieved from `https://www.sciencedirect.com/science/article/pii/S0167947322001402` doi: https://doi.org/10.1016/j.csda.2022.107560

Nakayama, Y., Yata, K. & Aoshima, M. (2021). Clustering by principal component analysis with gaussian kernel in high-dimension, low-sample-size settings. *Journal of Multivariate Analysis*, *185*, 104779.

Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, *103*(23), 8577–8582.

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., ... Golub, T. R. (2002, January). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, *415*(6870), 436–442.

Quackenbush, J. (2001). Computational analysis of microarray data. *Nature reviews genetics*, *2*(6), 418–427.

Ramey, J. A. (2016). datamicroarray: Collection of data sets for classification [Computer software manual]. (https://github.com/ramhiser/datamicroarray, http://ramhiser.com)

Renjith, S., Sreekumar, A. & Jathavedan, M. (2021). A comparative analysis of clustering quality based on internal validation indices for dimensionally reduced social media data. *Research in Transportation Business & Management*, 1047–1065.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, *20*, 53–65.

Roweis, S. T. & Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, *290*(5500), 2323–2326.

Santurkar, S., Tsipras, D., Ilyas, A. & Madry, A. (2018). How does batch normalization help optimization? *Advances in neural information processing systems*, *31*.

Sarkar, S. & Ghosh, A. K. (2019). On perfect clustering of high dimension, low sample size data. *IEEE transactions on pattern analysis and machine intelligence*, *42*(9), 2257–2272.

Shipp, M. A., Ross, K. N., Tamayo, P., Weng, A. P., Kutok, J. L., Aguiar, R. C. T., ...

Golub, T. R. (2002, January). Diffuse large B-cell lymphoma outcome prediction by gene-expression profiling and supervised machine learning. *Nature Medicine*, *8*(1), 68–74.

Shlens, J. (2014). A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*.

Sørlie, T., Perou, C. M., Tibshirani, R., Aas, T., Geisler, S., Johnsen, H., ... Børresen-Dale, A.-L. (2001, September). Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, *98*, 10869–10874.

Steinhaus, H. et al. (1956). Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci*, *1*(804), 801.

Steinley, D. (2004). Properties of the hubert-arable adjusted rand index. *Psychological methods*, *9*(3), 386.

Su, A. I., Cooke, M. P., Ching, K. A., Hakak, Y., Walker, J. R., Wiltshire, T., ... Hogenesch, J. B. (2002, April). Large-scale analysis of the human and mouse transcriptomes. *Proceedings of the National Academy of Sciences of the United States of America*, *99*(7), 4465–4470.

Terada, Y. (2013). Clustering for high-dimension, low-sample size data using distance vectors. *arXiv preprint arXiv:1312.3386*.

Traag, V. A., Waltman, L. & Van Eck, N. J. (2019). From louvain to leiden: guaranteeing well-connected communities. *Scientific reports*, *9*(1), 5233.

Valk, M. & Cybis, G. B. (2021). U-statistical inference for hierarchical clustering. *Journal of Computational and Graphical Statistics*, *30*(1), 133–143.

Van Der Maaten, L. (2013). Barnes-hut-sne. *arXiv preprint arXiv:1301.3342*.

Vera, J. F. & Macías, R. (2021). On the behaviour of k-means clustering of a dissimilarity matrix by means of full multidimensional scaling. *psychometrika*, *86*(2), 489–513.

West, M. M., Blanchette, C. C., Dressman, H. H., Huang, E. E., Ishida, S. S., Spang, R. R., ... Nevins, J. R. J. (2001, September). Predicting the clinical status of human breast cancer by using gene expression profiles. *Proceedings of the National Academy of Sciences of the United States of America*, *98*(20), 11462–11467.

Wilderjans, T. F., Ceulemans, E. & Meers, K. (2013). Chull: A generic convex-hull-based model selection method. *Behavior research methods*, *45*, 1–15.

Yu, Z., Chen, H., You, J., Liu, J., Wong, H.-S., Han, G. & Li, L. (2014). Adaptive fuzzy consensus clustering framework for clustering analysis of cancer data. *IEEE/ACM transactions on computational biology and bioinformatics*, *12*(4), 887–901.

Zheng, W., Zou, C. & Zhao, L. (2005). An improved algorithm for kernel principal component analysis. *Neural Processing Letters*, *22*, 49–56.

A Supplementary tables

		External Validation	Internal Validation			
DRT	Clustering Method	ARI	Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin
	k-Means	-0,03	0,31	0,39	**22,22**	1,58
	k-Means MADD	-0,03	0,32	0,35	15,89	1,87
	AGNES	-0,05	**0,36**	**0,50**	12,40	1,23
	Distance Vector	**0,00**	0,31	0,29	18,09	1,80
	Avg Distance Vector	-0,03	0,30	0,38	4,92	**0,90**
	Single Uclust	-0,03	0,05	0,26	7,35	1,84
PCA	k-Means	-0,03	0,40	0,16	36,27	1,23
	AGNES	-0,05	0,45	0,31	29,36	1,11
ICA	k-Means	-0,04	0,16	0,06	10,43	2,29
	AGNES	-0,01	0,50	**0,60**	6,16	0,40
t-SNE	k-Means	0,01	0,42	0,19	52,98	0,91
	AGNES	0,00	0,40	0,11	50,58	1,01
LLE	k-Means	-0,03	0,15	0,13	10,60	2,35
	AGNES	-0,01	0,41	0,55	4,48	0,46
Gaussian kPCA	k-Means	**0,40**	0,17	0,18	12,21	2,21
	AGNES	-0,01	0,32	0,41	8,50	0,80
VAE	k-Means	0,06	0,06	0,19	4,09	2,65
	AGNES	-0,01	0,26	0,31	2,47	0,63
NDA	k-Means	-0,02	0,51	0,11	**62,30**	0,92
	AGNES	-0,05	0,55	0,24	57,83	0,85
NDA Rotated	k-Means	0,00	0,29	0,07	18,34	1,79
	AGNES	-0,01	**0,60**	0,58	9,88	**0,31**

Table 6: Cluster validation indices for clustering (non)dimensionally reduced data from Alon et al. (1999)

		External Validation	Internal Validation			
DRT	Clustering Method	ARI	Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin
	k-Means	0,64	0,35	0,31	**121,44**	1,21
	k-Means MADD	0,79	0,31	0,23	48,78	1,91
	AGNES	0,24	**0,41**	**0,55**	62,20	**1,01**
	Distance Vector	**0,99**	0,39	0,36	119,79	1,09
	Avg Distance Vector	**0,99**	0,39	0,36	119,79	1,09
	Single Uclust	0,98	0,03	0,15	20,29	1,86
PCA	k-Means	0,66	0,65	0,09	206,65	0,54
	AGNES	**1,00**	**0,81**	**0,37**	**1490,75**	**0,31**
ICA	k-Means	**1,00**	0,77	0,32	1057,96	0,36
	AGNES	**1,00**	0,77	0,32	1057,96	0,36
t-SNE	k-Means	0,79	0,62	0,36	589,35	0,53
	AGNES	0,79	0,62	0,36	589,35	0,53
LLE	k-Means	0,47	0,60	0,09	243,53	0,73
	AGNES	0,05	0,59	0,14	240,01	0,73
Gaussian kPCA	k-Means	0,68	0,64	0,07	435,67	0,57
	AGNES	0,67	0,66	0,14	487,63	0,57
VAE	k-Means	0,97	0,39	**0,37**	120,11	1,09
	AGNES	0,97	0,39	**0,37**	120,11	1,09
NDA	k-Means	0,31	0,44	0,06	138,66	1,01
	AGNES	0,13	0,40	0,15	44,71	0,66
NDA Rotated	k-Means	0,27	0,37	0,12	0,12	1,26
	AGNES	0,01	0,38	0,18	42,72	0,95

Table 7: Cluster validation indices for clustering (non)dimensionally reduced data from Christensen et al. (2009)

		External Validation	Internal Validation			
DRT	Clustering Method	ARI	Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin
	k-Means	**0,05**	0,09	0,21	**14,23**	3,40
	k-Means MADD	0,10	0,20	0,37	5,55	5,28
	AGNES	0,01	**0,46**	**0,80**	5,41	**0,43**
	Distance Vector	**0,05**	0,36	0,55	6,25	3,07
	Avg Distance					
	Vector	0,01	**0,46**	**0,80**	5,41	**0,43**
	Single Uclust	0,00	-0,08	0,15	3,22	2,44
PCA	k-Means	0,00	0,37	0,05	85,66	1,38
	AGNES	0,02	0,60	0,35	27,27	0,63
ICA	k-Means	0,10	0,32	0,05	61,88	1,62
	AGNES	0,02	0,55	0,22	33,24	0,83
t-SNE	k-Means	0,00	**0,63**	0,36	**500,92**	0,57
	AGNES	0,00	**0,63**	0,36	**500,92**	0,57
LLE	k-Means	0,00	0,24	0,06	46,40	1,80
	AGNES	0,02	0,30	0,13	21,77	1,23
Gaussian	k-Means	**0,15**	0,29	0,05	62,63	1,60
kPCA	AGNES	-0,01	0,31	0,11	27,92	1,23
VAE	k-Means	0,00	0,03	0,17	4,85	5,83
	AGNES	0,00	-0,12	0,20	1,00	3,46
NDA	k-Means	0,00	0,21	0,12	23,49	2,22
	AGNES	0,01	0,59	**0,77**	9,58	**0,32**
NDA	k-Means	0,00	0,08	0,07	12,56	3,63
Rotated	AGNES	0,01	0,56	0,75	8,37	0,34

Table 8: Cluster validation indices for clustering (non)dimensionally reduced data from Gravier et al. (2010)

		External Validation	Internal Validation			
DRT	Clustering Method	ARI	Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin
	k-Means	**0,34**	0,10	0,44	**8,06**	2,56
	k-Means MADD	0,09	0,08	0,34	1,93	5,24
	AGNES Distance	-0,01	**0,17**	**0,65**	1,84	**0,73**
	Vector Avg Distance	0,10	0,07	0,44	3,55	2,40
	Vector Single	0,06	0,03	0,49	2,39	1,18
	Uclust	0,17	0,13	0,48	5,07	1,53
PCA	k-Means	0,07	0,49	0,14	94,61	0,83
	AGNES	0,07	0,50	0,30	92,57	0,61
ICA	k-Means	0,07	0,50	0,14	89,12	0,71
	AGNES	0,07	0,51	0,32	94,28	0,61
t-SNE	k-Means	**0,25**	0,54	0,27	5123,52	0,64
	AGNES	0,18	0,54	0,32	6024,26	0,52
LLE	k-Means	**0,25**	0,45	0,17	84,71	0,79
	AGNES	0,08	0,47	0,25	76,43	0,67
Gaussian kPCA	k-Means	-0,03	0,84	0,13	**98798,58**	0,20
	AGNES	-0,03	**0,88**	**1,19**	64666,87	**0,10**
VAE	k-Means	0,05	0,00	0,36	2,45	4,88
	AGNES	0,01	-0,04	0,07	2,37	4,28
NDA	k-Means	0,14	0,33	0,25	27,95	1,03
	AGNES	-0,02	0,33	0,27	26,46	0,97
NDA Rotated	k-Means	0,23	0,32	0,22	19,95	1,05
	AGNES	0,02	0,24	0,30	13,44	1,01

Table 9: Cluster validation indices for clustering (non)dimensionally reduced data from Khan et al. (2001)

		External Validation	Internal Validation			
DRT	Clustering Method	ARI	Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin
	k-Means	**-0,01**	0,18	0,38	**14,46**	1,99
	k-Means MADD	-0,03	0,19	0,38	12,08	2,05
	AGNES	-0,02	**0,32**	**0,66**	3,42	**0,53**
	Distance Vector	**-0,01**	0,30	0,59	6,87	1,19
	Avg Distance Vector	**-0,01**	0,30	0,59	6,87	1,19
	Single Uclust	**-0,01**	0,07	0,49	5,70	1,62
PCA	k-Means	-0,01	0,45	0,11	53,59	0,83
	AGNES	-0,01	0,43	0,18	45,85	0,83
ICA	k-Means	NA	NA	NA	NA	NA
	AGNES	NA	NA	NA	NA	NA
t-SNE	k-Means	**0,04**	0,42	0,14	277,56	1,03
	AGNES	-0,02	**0,53**	0,34	**871,77**	0,66
LLE	k-Means	-0,01	0,43	0,09	43,10	0,83
	AGNES	-0,03	0,41	0,22	40,23	0,86
Gaussian kPCA	k-Means	0,00	0,31	0,07	22,55	1,11
	AGNES	-0,02	0,46	0,18	49,80	0,78
VAE	k-Means	0,00	0,07	0,26	4,94	3,35
	AGNES	0,02	0,12	0,31	3,45	3,10
NDA	k-Means	0,00	0,39	0,15	38,04	0,88
	AGNES	-0,01	0,47	**0,38**	15,84	**0,44**
NDA Rotated	k-Means	**0,04**	0,28	0,16	18,51	1,25
	AGNES	0,03	0,27	0,21	15,22	1,19

Table 10: Cluster validation indices for clustering (non)dimensionally reduced data from Pomeroy et al. (2002)

		External Validation	Internal Validation			
DRT	Clustering Method	ARI	Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin
	k-Means	**0,16**	0,39	0,37	11,82	1,70
	k-Means MADD	0,05	**0,63**	**1,13**	**13,05**	**0,27**
	AGNES	0,05	**0,63**	**1,13**	**13,05**	**0,27**
	Distance Vector	0,05	0,54	0,78	8,07	0,35
	Avg Distance Vector	0,05	0,54	0,78	8,07	0,35
	Single Uclust	0,12	-0,01	0,30	4,86	1,96
PCA	k-Means	0,11	0,33	0,08	26,40	1,15
	AGNES	0,05	0,25	**0,24**	14,10	0,71
ICA	k-Means	NA	NA	NA	NA	NA
	AGNES	NA	NA	NA	NA	NA
t-SNE	k-Means	0,15	**0,42**	0,09	**1012,37**	0,84
	AGNES	0,17	**0,42**	0,19	780,49	**0,69**
LLE	k-Means	0,06	0,26	0,15	23,89	1,17
	AGNES	-0,02	0,20	0,21	8,80	0,83
Gaussian kPCA	k-Means	0,00	0,30	0,18	25,07	1,30
	AGNES	0,09	0,34	0,19	35,23	1,00
VAE	k-Means	0,12	0,03	0,14	2,99	4,91
	AGNES	**0,18**	0,26	0,23	7,08	1,76
NDA	k-Means	0,00	0,32	0,08	38,75	1,08
	AGNES	0,05	0,32	0,14	19,62	0,78
NDA Rotated	k-Means	-0,01	0,31	0,09	36,88	1,00
	AGNES	0,05	0,36	0,14	15,50	0,71

Table 11: Cluster validation indices for clustering (non)dimensionally reduced data from Shipp et al. (2002)

		External Validation	Internal Validation			
DRT	Clustering Method	ARI	Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin
	k-Means	0,54	0,08	0,41	**7,68**	2,58
	k-Means MADD	0,13	0,04	0,36	3,01	4,92
	AGNES Distance	-0,01	**0,11**	**0,62**	1,87	**0,74**
	Vector Avg Distance	0,18	0,02	0,44	3,74	3,09
	Vector Single	0,02	-0,12	0,31	1,87	1,17
	Uclust	**0,62**	0,04	0,38	3,89	2,24
PCA	k-Means	**0,57**	0,41	0,06	87,20	0,82
	AGNES	0,34	0,40	0,14	67,65	0,75
ICA	k-Means	0,45	0,42	0,09	91,33	0,83
	AGNES	0,24	0,37	0,12	57,32	0,76
t-SNE	k-Means	0,50	0,43	0,08	2024,44	0,84
	AGNES	0,41	**0,46**	0,17	**2175,79**	**0,69**
LLE	k-Means	0,41	0,45	0,05	121,94	0,72
	AGNES	0,30	0,40	0,14	99,16	0,80
Gaussian kPCA	k-Means	0,41	0,46	0,13	114,39	0,73
	AGNES	0,55	0,38	0,15	71,84	0,76
VAE	k-Means	0,20	-0,01	0,30	3,53	4,27
	AGNES	0,23	0,01	**0,43**	3,43	4,34
NDA	k-Means	0,24	0,21	0,18	22,77	1,32
	AGNES	0,19	0,18	0,21	12,04	1,07
NDA Rotated	k-Means	0,39	0,26	0,18	25,50	1,22
	AGNES	0,12	0,22	0,20	14,78	0,99

Table 12: Cluster validation indices for clustering (non)dimensionally reduced data from Sorlie et al. (2001)

		External Validation	Internal Validation			
DRT	Clustering Method	ARI	Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin
	k-Means	**0,95**	**0,16**	0,66	**20,65**	1,70
	k-Means MADD	0,63	**0,16**	0,66	**20,65**	1,70
	AGNES	0,68	0,14	0,68	1,69	**0,76**
	Distance Vector	0,31	0,12	**0,70**	8,14	1,21
	Avg Distance Vector	0,31	0,12	**0,70**	8,14	1,21
	Single Uclust	**0,95**	0,05	0,44	5,76	2,26
PCA	k-Means	0,65	0,43	0,42	66,55	0,80
	AGNES	0,68	0,43	0,42	66,55	0,80
ICA	k-Means	0,37	0,08	0,54	7,23	3,03
	AGNES	0,58	0,08	0,54	5,67	2,46
t-SNE	k-Means	0,61	**0,69**	**0,95**	**262,73**	**0,40**
	AGNES	**0,92**	**0,69**	**0,95**	**262,73**	**0,40**
LLE	k-Means	0,59	0,19	0,13	19,10	2,26
	AGNES	0,55	0,24	0,24	22,16	1,71
Gaussian kPCA	k-Means	0,27	0,19	0,14	19,15	2,21
	AGNES	0,59	0,23	0,23	13,13	0,99
VAE	k-Means	0,49	0,12	0,55	10,81	2,89
	AGNES	0,37	0,10	0,54	9,50	2,40
NDA	k-Means	0,49	0,45	0,13	90,64	0,91
	AGNES	0,40	0,45	0,16	83,08	0,84
NDA Rotated	k-Means	0,44	0,25	0,09	31,42	1,74
	AGNES	0,43	0,27	0,14	19,64	1,20

Table 13: Cluster validation indices for clustering (non)dimensionally reduced data from Su et al. (2002)

		External Validation	Internal Validation			
DRT	Clustering Method	ARI	Silhouette	Dunn	Calinski-Harabasz	Davies-Bouldin
	k-Means	-0,02	**0,31**	0,74	**5,63**	1,84
	k-Means MADD	**0,00**	**0,31**	0,74	**5,63**	1,84
	AGNES Distance	**0,00**	**0,31**	**0,81**	3,11	**0,56**
	Vector Avg Distance	**0,00**	**0,31**	0,74	**5,63**	1,84
	Vector Single	**0,00**	**0,31**	0,74	**5,63**	1,84
	Uclust	**0,00**	0,03	0,47	3,43	2,29
PCA	k-Means	0,00	0,32	0,07	17,00	1,49
	AGNES	0,01	0,63	0,52	20,99	0,51
ICA	k-Means	NA	NA	NA	NA	NA
	AGNES	NA	NA	NA	NA	NA
t-SNE	k-Means	**0,03**	0,53	0,47	**69,50**	0,73
	AGNES	**0,03**	0,53	0,47	**69,50**	0,73
LLE	k-Means	0,00	0,25	0,17	16,84	1,53
	AGNES	0,02	0,26	0,31	6,77	0,79
Gaussian kPCA	k-Means	0,01	0,30	0,12	17,81	1,53
	AGNES	0,00	0,49	0,47	18,83	0,84
VAE	k-Means	0,00	0,31	0,74	5,63	1,84
	AGNES	0,00	0,31	0,74	5,63	1,84
NDA	k-Means	0,00	0,32	0,02	10,08	1,94
	AGNES	0,00	**0,79**	**0,90**	26,65	**0,19**
NDA Rotated	k-Means	-0,01	0,30	0,03	10,83	1,92
	AGNES	0,00	0,71	0,82	15,49	0,25

Table 14: Cluster validation indices for clustering (non)dimensionally reduced data from West et al. (2001)

B t-SNE perplexity tuning

This analysis is based on the analysis in https://towardsdatascience.com/how-to-tune-hyperparameters-of-tsne-7c0596a18868.

The goal of t-SNE is to minimise the Kullback-Leibler divergence between the distributions q_{ij} and p_{ij}. Furthermore, we know that in general, larger N implies larger optimal $Perplexity$. Plotting the KL divergence as a function of perplexity, while keeping all other parameters fixed, Figure 1 shows that the KL divergence decreases monotonically, behaving like $KL = 1/Perplexity$. If we want to optimise with respect to $Perplexity$, we should create a function that looks like the following:

$$Score \sim \frac{1}{Perplexity} + Penalty \tag{21}$$

An intuitive penalty term, that leads to a function we can minimise, is taking $Penalty = Perplexity$. Though this formulation will be dominated by the penalty term, which naturally increases with N, and therefore we should normalise the penalty term:

$$Score \sim \frac{1}{Perplexity} + \frac{Perplexity}{N} \tag{22}$$

Minimising this function with respect to $Perplexity$, we get:

$$\frac{\partial Score}{\partial Perplexity} = -\frac{1}{Perplexity^2} + \frac{1}{N} = 0 \tag{23}$$

Which leads to $Perplexity \sim \sqrt{N}$.

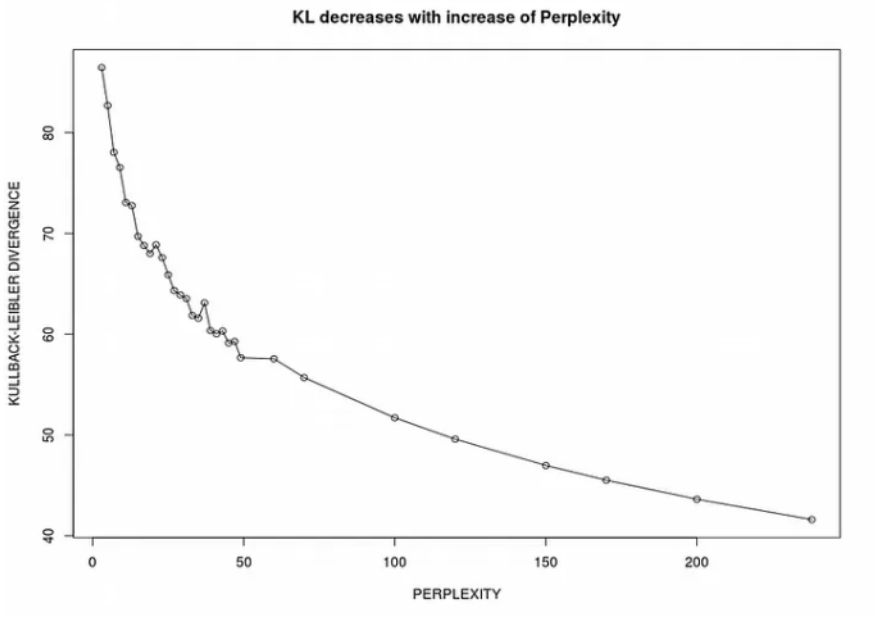

Figure 1: Kullback-Leibler divergence against Perplexity, keeping all other parameters fixed