Dimensionality Reduction in Genomic Big Data

Dekkers, Diederik

Award date:
2021

Eindhoven University of Technology

Bachelor Final Project

Dimensionality Reduction in Genomic Big Data

Author:
D.M.H. Dekkers *(1021655)*

Supervisors:
prof. dr. E.R. van den Heuvel
prof. dr. P.J.H. Daas

August 9, 2021

Abstract

The objective of this paper is to investigate different dimensionality reduction methods in the context of a genomic big data set. From literature research, an overview is provided of which methods are useful as dimensionality reduction techniques for the data set introduced in this thesis. After this different methods are evaluated based on their ability to reduce the dimensionality of the data set while retaining as much relevant information as possible.

Two methods are tested on this data set, both using entropy measures. The paper of Dash et al. (2000) starts off with a filter method, which is expanded upon by a wrapper method. The paper of Varshavsky et al. (2006) uses only a filter method. It is shown that both of these methods show good results, dependent on the type of data pre-processing that is applied to the data set.

The dimensionality of the data set was reduced from 47,582 genes to 600 while improving the cluster quality.

Keywords— Genomic Big Data, Dimensionality Reduction, Feature Selection, Feature Extraction

Contents

1 Introduction

Over the past decades, the rate at which data has become available has rapidly increased. It has become much cheaper and easier to gather data. For instance, in the field of genomics, it took over 13 years and more than 1000 scientists and $3 billion to sequence the first human genome, while today it can be done in under 40 hours with an associated cost of less than $1000.00 [1]. This implies that the available amount of data related to genomic big data has increased rapidly. It is expected that within the next decade, researchers will analyze the genomes of all living creatures making genomics an enormous and hugely important generator of data. This has shifted the focus in the field of genomics from sequencing the data to making use of the data. Dimensionality reduction can play a key role in helping to negate these problems.

Dimensionality reduction can be characterized in many different ways. In the broadest sense, dimensionality reduction is the process of reducing the dimensionality of a data set while retaining as much relevant information as possible. What information is relevant depends on the context of the problem.

There are multiple ways in which dimensionality reduction can be divided. Firstly, dimensionality reduction can be divided into feature selection and feature extraction. Feature selection is the process in which a subset of features is chosen from the original set of features. In feature extraction new features can be constructed using the original data set, meaning that these do not have to be a subset of the original data set.

According to Tadist et al. [14], feature selection is believed to become a game-changer that can help to substantially reduce the complexity of the data, making it easier to analyze and translate data into useful information.

Dimensionality reduction has four main objectives [8]:

- Providing a better understanding of the underlying process that generated the data. It can be difficult to notice trends or clusters among the data when there has not been any dimensionality reduction. By reducing the number of features, these trends can become easier to notice. Note that this is not possible for all forms of dimensionality reduction. An example would be genomic data, where the features are genes. Applying feature selection to this data set results in a ranking of the most important genes. In bioinformatics analysis, there is a cost associated with measuring the expression of genes. A ranking would allow for a smaller selection of genes to be measured.
- Improving the prediction performance of the predictors. For some data sets, the goal is to build a predictive model. The curse of dimensionality refers to certain problems that arise when analyzing high-dimensional data. An example of this is overfitting. Overfitting is *the production of an analysis that corresponds too closely or exactly to a particular set of data, and may therefore fail to fit additional data or predict future observations reliably* [2]. This means that a certain predictive model would be too specifically trained for this data set, which would result in less accuracy when applied to a different data set. Dimensionality reduction reduces the number of input variables in such a model, which means that the likelihood of overfitting decreases.
- Lowering the computational costs. The data sets used for feature selection are often highly dimensional. This means that the data set becomes very hard to handle as it is computationally costly to use in further research. An example of this is the distance matrix of a data set. Often used in data analysis, this matrix shows the distance between pairs of objects of the data set. A distance matrix exponentially increases in size relative to the number of variables. This means that it is quickly too big to effectively use when no feature selection is applied.
- Reducing the amount of storage space required for storing the data set.

There are many different ways in which feature selection methods can be divided. One of the most cited works in this category is the work of Kohavi and John (1997) [10], which gives a way to divide feature selection methods. Although this is not the only way in which these methods can be divided, it is by far the most used division. That is why it will also be used in this thesis. Kohavi and John (1997) [10] divide feature selection methods into filter, wrapper, and embedded methods. The definitions are the following.

- Filter methods are the most straightforward of the feature selection methods. Filter methods select the most relevant features using only the data itself, using a certain statistic to 'score' or 'rank' the features. This makes filter methods relatively easy and quick. An example is a correlation coefficient. Using correlation, all features can be given a certain correlation score. After this, k features with the largest correlation can be selected or all features with correlation greater than M. Here k and M should be selected by the user.
- Wrapper methods evaluate feature subsets using the results of a specific clustering algorithm or predictive model. Clustering algorithms are used in unsupervised learning and a predictive model can only be used in

supervised learning. These methods are generally more computationally costly compared to filter methods and more prone to overfitting, but do often achieve better results. A search algorithm is required for a wrapper method, for instance, forward search. An example of a wrapper method is a search algorithm that sequentially adds features. These features are used in a k-means clustering algorithm. The performance of the clustering algorithm is then evaluated and the most relevant features can be selected.

- Embedded methods are a combination of filter and wrapper methods, and thus try to make use of the efficiency and speed of the filter methods and the effectiveness of wrapper methods.

The goal of this thesis is split up into two parts. The first goal is to find and summarize relevant dimensionality reduction methods that can be applied to the data set introduced in this paper. This is accomplished by doing literature research. The second goal is to apply feature dimensionality methods found in the literature research to the data set that will be introduced later on in this thesis. This results in the following research questions.

- What is the most relevant work done so far in the field of dimensionality reduction relevant to the data set that is introduced in this paper?
- Can one or more of these methods found in the literature research be used for the potato data sets and how do these methods compare?

2 Data set

2.1 Data set origin

The data set that is used in this thesis consists of eight commercial and twelve experimental potato varieties which were grown in the Netherlands, and two extra commercial varieties grown in the United Kingdom. The potatoes in the Netherlands were harvested in September 2010, 2011, and/or 2012, and the varieties in the United Kingdom in 2004 or 2004 and 2005. For each type of potato, different amounts of samples were taken, ranging from 2 to 20. This means that for the potatoes more than one sample was obtained, related to different environmental conditions. These conditions include different field trials but do not include technical and biological variations, meaning there was no variation in the year of harvest, location, and the time until the potato was analyzed since harvest. All this was done to get an unbiased, clear transcriptome profile of all the potatoes. The result is that a data set is obtained within total 104 profiles of potatoes. In table 1 it can be seen for instance that the 'Desireé' potato has 2 profiles while the 'Bintje' potato has 17. Each potato profile is based on scores for 47,582 different genes, which covers the majority of the potato transcriptome. The focus of this paper is on the mathematical and statistical analysis of this data set. Therefore the biological aspect of how this data set is obtained is covered in minimal detail. This data set has been studied in other papers, and the process of transcriptomics and metabolomics is described in much greater detail in the paper of E. Kok et al [11].

Figure 1: Summary table of the potato data used

Variety	Type	Year of harvest	Year of data sequencing	# transcriptome profiles
Desirée	Parent (non-GM counterpart)	2012	2014	2
Bintje	Commercial	2011–12	2012–13-14	17
Biogold	Commercial	2010	2012	2
Fontane	Commercial	2010	2012	2
Innovator	Commercial	2010–12	2012–13-14	9
Lady Balfour	Commercial	2004	2012	2
Lady Rosetta	Commercial	2010–12	2012–13	5
Maris Piper	Commercial	2010	2012	2
Nicola	Commercial	2011–12	2012–13	12
Sante	Commercial	2004–5	2012	5
A01-20	GM	2012	2014	2
94	Experimental	2011–12	2012–13	4
HZ94	Experimental	2011–12	2012–13	4
IVP06	Experimental	2011–12	2012–13	6
IVP4X-154	Experimental	2011	2012	2
IVP159	Experimental	2011	2012	2
RH386	Experimental	2011–12	2012–13	4
RH29	Experimental	2011–12	2012–13	4
RH36	Experimental	2011–12	2012–13	4
RH4X-054	Experimental	2011–12	2012–13	4
RH753	Experimental	2011–12	2012–13	6
RH90	Experimental	2011–12	2012–13	4

Taken from E. Kok et al [11]

2.2 Rough data set inspection

The data set consists of a total of 104 transcriptome profiles of different types of potatoes, and per transcriptome profile, there are 47,582 gene transcripts. This means there are 104 rows and 47,582 columns. The gene expression of these potatoes ranges from 0 to 923,563. Approximately 64% of the data set consists of 0-entries, and approximately 25% of the rows of the data sets are rows containing only zeroes. More precisely, there are a total of 37,858 features that do not contain only zeroes. This does not mean however that most features are not relevant. When we make a subset of the original data set of features that have a total gene transcript larger or equal than 1000, 20,422 features remain. This gives the idea that the discrepancy between different potatoes or potato samples is very large. In figure 2 the average and standard deviation of the gene expressions are displayed. These are divided into two groups, commercial and experimental potatoes. The GMO-type potatoes are left out of these graphs because there are only two samples available of the GMO-type. In figure 3 the percentage of genes that are equal to 0 is displayed, once again divided into commercial and experimental potato types. In figures 4, 5, and 6 these statistics are also displayed, but divided per potato type.

Figure 2: Average and standard deviation for commercial and experimental potato types

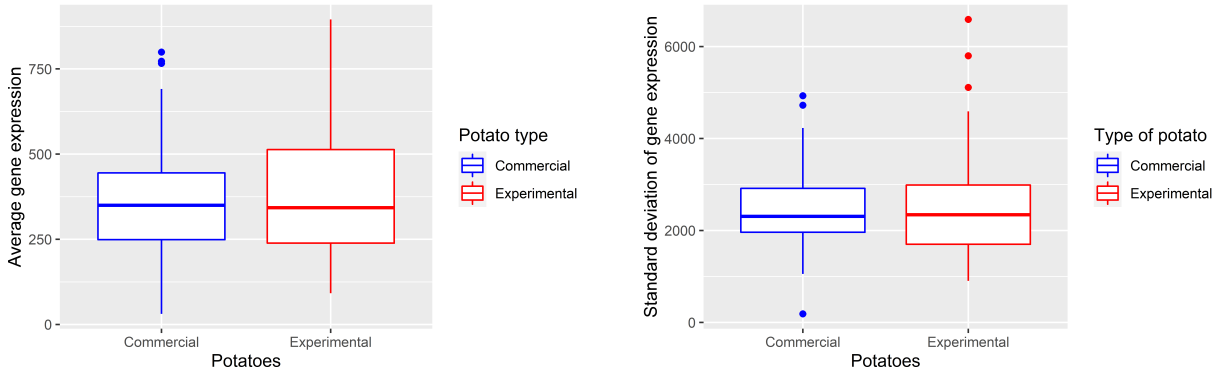
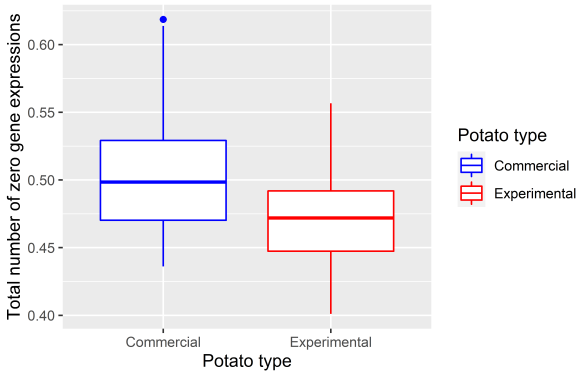


Figure 3: Percentage of gene expression which are equal to 0 divided by commercial and experimental types



GMO-type potatoes are excluded from these graphs

Figure 4: Average gene expression for commercial and experimental potato types

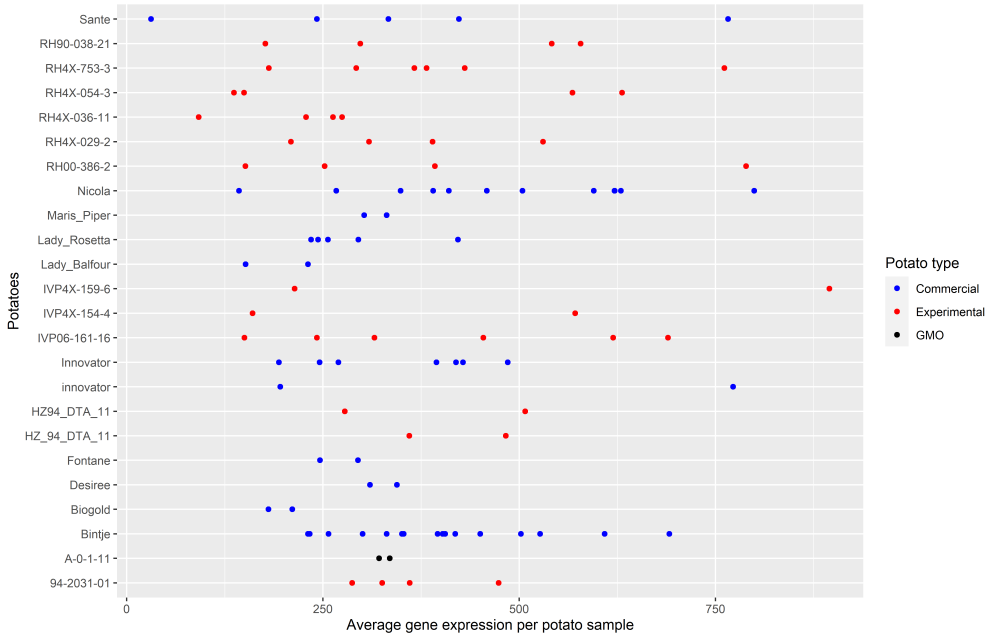


Figure 5: Standard deviation of the gene expression for commercial and experimental potato types

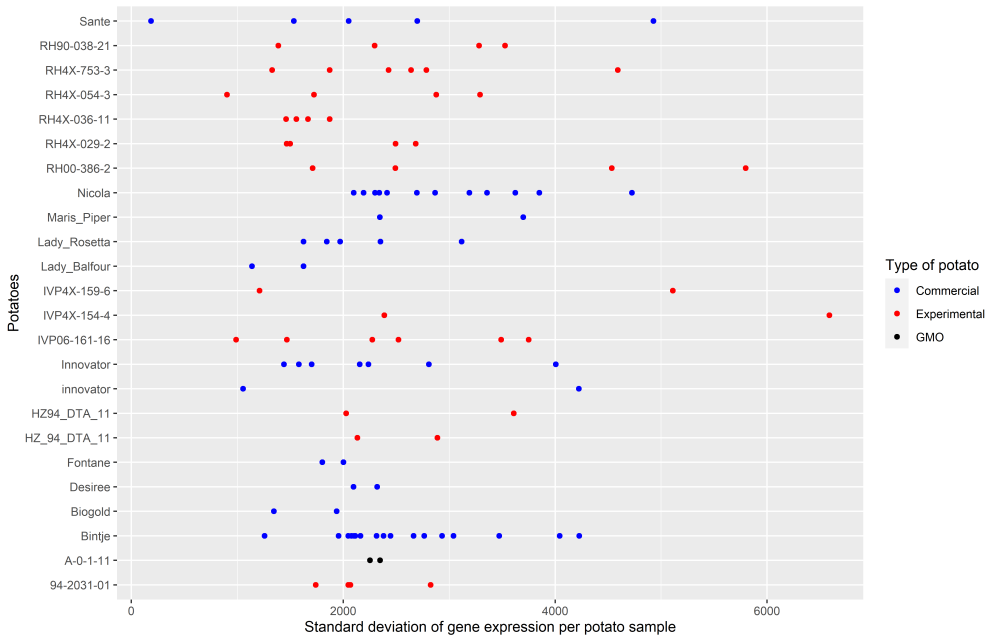
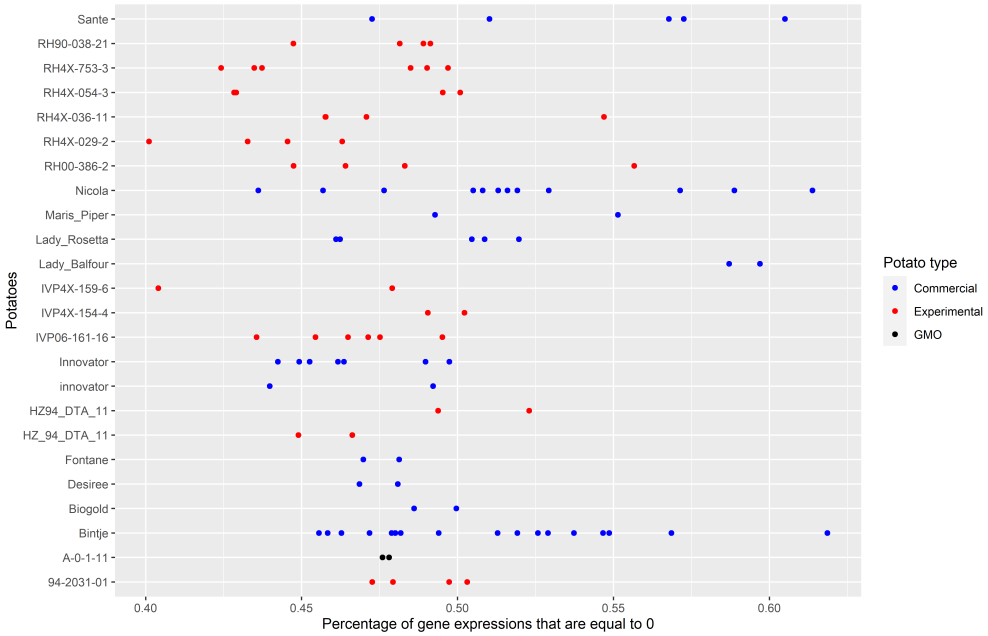


Figure 6: Number of gene expressions per potato sample which are equal to 0 for commercial and experimental potato types



3 Literature research

3.1 Search queries and synonyms

To get a good overview of current methods used in feature selection, a literature review was done. This literature review focused mainly on papers that gave an overview of already used feature selection methods, such as Guyon et al. [8], Tadist et al. [14], and Solorio-Fernandez et al. [13]. Literature research was mainly done using Scopus and Google Scholar. To get good results, multiple terms were often used to form an appropriate search query. These terms are provided in table 1, which summarizes the used terms and their synonyms used in the literature research. The list of search terms was set up to get a broad result of papers and to limit the chance of missing relevant work. It is important to note that not all terms are direct synonyms. This was done because it was found that authors do not concisely use certain terms, and the probability was high that relevant work would not be found if some of these search terms would not be included. An example of a search query would be the following.

(Feature Selection OR variable selection OR attribute selection)
AND
(Genomics OR micro-array data)
AND
(Overview OR review)

Table 1: Overview of search terms and their synonyms

Terms	**Synonyms**
Dimensionality reduction	Feature reduction, dimension reduction
Feature selection	Unsupervised feature selection, variable selection, attribute selection, variable subset selection
Feature extraction	Feature construction, projection onto a low dimensional subspace, feature representation, variable extraction
Unsupervised	Unlabeled, non-labeled
Supervised	Labeled
Genomic	Genomics, micro-array data, bioinformatics
Big data	High-dimensional data, large data set
Overview	Outline, systemic review, summary

3.2 Selection criteria

To choose relevant work found using the search terms, a few selection criteria were used. The most important ones were the following.

- Number of references of a paper.
- Type of paper. Overview papers were more thoroughly researched compared to normal papers.
- References of a paper within overview studies, with the corresponding reputation by the author of the overview paper. Some papers were mentioned by different authors in overview papers, with a very positive annotation. Some of these papers did not have the highest amount of citations, but due to this positive recommendations still investigated.
- The reputation of the source, e.g. the journal in which the paper was published.
- Relevance to the data set used in this paper.

4 Dimensionality Reduction Methods

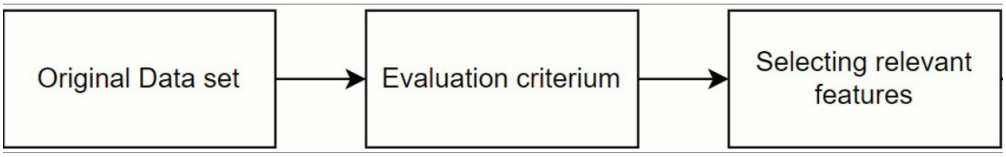
As mentioned in Chapter 1, dimensionality reduction methods can be divided into feature selection and feature extraction. Furthermore, feature selection can be divided into filter, wrapper, and embedded methods.

Dimensionality reduction methods can also be divided into supervised and unsupervised problems. This is the case for both feature selection and feature extraction. In supervised dimensionality reduction (Kotsiantis et al. [12]) the data set is labeled, which informs us about what group (or cluster) the data belongs to. In unsupervised dimensionality reduction, the data does is not labeled or the labeling of the data is not used. Unsupervised feature reduction is also used when the data is labeled. This is because unsupervised dimensionality reduction is less prone to overfitting (Guyon et al. [8]).

This chapter will look at the different methods of dimensionality reduction, and explain in greater detail how they work. Advantages and disadvantages will be given and where necessary examples will be provided to illustrate how these methods work.

4.1 Filter methods

Figure 7: Overview of filter feature selection method



Filter methods are a type of feature selection. Filter methods refer to the feature selection methods which select features independently of the model that is being used. This means that the selected features are based only on the general characteristics of the data. These methods can be used on their own or used as a predecessor step in combination with another dimensionality reduction method. Filter methods can be divided into two categories, univariate and multivariate. Univariate filter methods are effective at identifying and removing irrelevant features, but not so much at removing redundant features [13]. This is because univariate methods do not take into account dependency among features, as the features are evaluated on a dual basis, not in groups. Multivariate filter methods work by evaluating the relevance of features based on groups of features. This means that multivariate methods can handle both relevancy and redundancy, and will thus perform better than univariate filter methods most of the time. The downside is that multivariate methods are more costly to run.
Univariate filter methods rely on a certain statistic to rank the features according to this statistic. If a ranking is achieved using a criterion, a subset of features can be chosen based on this ranking. There are two ways to do this, depending on the knowledge one has of the data set. When one has a-priori knowledge of the data set, it is often best to use this knowledge in unsupervised filter methods as there is no output variable. This knowledge can be used to select the number of features required, which gives the final subset of features together with the ranking. The most important features can then be chosen when the amount of features that are needed, is known. Often, however, this knowledge is not present which makes the choice of features more difficult. Then other methods are required to overcome the lack of knowledge. In the potato data set examined in this paper, there is little to no relevant domain knowledge. That is why the algorithms applied to this data set will assume no prior knowledge. In some practical applications, however, it can be easy to note a certain trend amongst the features. If this is the case, features can also be chosen based on this principle. This is however not a method that works for all data sets, and the only way to know for certain is to closely inspect the ranking of the features and to plot them.

Table 2: Advantages and disadvantages of filter methods

Method	Advantages	Disadvantages
Univariate	Scalable Fast Independent of classifier	Ignores dependencies Features evaluated individually
Multivariate	Can handle dependencies Computationally more efficient than wrapper methods	Computationally costlier than univariate methods Less scalable than univariate methods

It is difficult to determine a 'best' statistic that can be used in filter methods. Currently, there is no clear information on microarray data which filter method criterion is the best [18]. That is why there are still multiple criteria used. Below a few common criteria are named, together with their advantages and disadvantages.
One statistic that is often used and has a solid theoretical background, is Pearson's correlation coefficient. Shown below is the Pearson correlation coefficient. Here X is the matrix containing the data and Y is the output variable.

$$\mathcal{R}_i = \frac{cov(X_i, Y)}{\sqrt{var(X_i)var(Y)}} \tag{1}$$

The estimate of $\mathcal{R}$ is given by:

$$R_i = \frac{\sum_{k=1}^{m}(x_{k,i} - \bar{x}_i)(y_k - \bar{y})}{\sqrt{\sum_{k=1}^{m}(x_{k,i} - \bar{x}_i)^2 \sum_{k=1}^{m}(y_k - \bar{y})^2}} \tag{2}$$

The squares of this estimate can be used further. R_i^2 is more generally known as the R^2 coefficient, or the coefficient of determination. This coefficient is a percentage that determines the fraction of the total movement of the regression line which is explained by a change in the input variable. This metric can thus be used to construct a ranking according to the goodness-of-fit of linear variables. This method has two immediate disadvantages. First and foremost it relies on an output variable Y. As explained earlier in the context of the potato data set this is not very useful and is the reason why this coefficient will not be explained further. The second downside is that R_i can only detect linear dependencies between variables. To combat this fact, different measures can be taken of this correlation coefficient. Two prominent examples are Kendall's tau and Spearman's Rho coefficient. Both of these can be formulated as a special case of the general correlation coefficient.
The second class of methods often used as filter methods for feature selection are methods based on Information Theory. I. Guyon et al. [8] note that many information-theoretic criteria rely on empirical estimates of mutual information between each variable and its target. An information criterion is also proposed:

$$I(i) = \int_{x_i}\int_{y} p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)} dxdy \tag{3}$$

Where $p(x_i)$ and $p(y)$ are the probability density functions of respectively input variable x and output variable y and $p(x_i, y)$ is the joint probability density. This information criteria is for continuous probabilities. For the discrete case the following criterion is defined.

$$I(i) = \sum_{x_i}\sum_{y} P(X = x_i, Y = y) \log\Big(\frac{P(X = x_i, Y = y)}{P(X = x_i)P(Y = y)}\Big) \tag{4}$$

A notable disadvantage to this method is that these probabilities densities are very hard to determine, and often even impossible to determine.

Another method often used in feature selection is entropy, or contributed entropy. There are several methods proposed to use this in feature selection, most notably In Varshavsky et al. [15] and in Dash et al. [7]. Entropy is loosely defined to be, in the context of information theory, the average level of uncertainty inherent to the possible outcomes of the variables. This means that entropy is a way to measure the influence certain variables have on the outcome of the uncertainty of the system. Entropy is mathematically defined by the following formula.

$$E = -\sum_{X_i}\sum_{X_j} P(X_{ij}) \log P(X_{ij}) \tag{5}$$

The methods proposed in these papers work by calculating the entropy [15] of every feature. A sample is taken from the data set, based on the search algorithm or random choice. Sequentially every feature is removed one by one, after which the entropy of the system is calculated. Using this method, the contribution of every feature to the entropy of the system can be calculated and used in determining a ranking. Looking at formula 5, we can see that it also uses the probability densities, same as in equation 3. There are different methods to overcome this obstacle. In the paper of Dash et al. [7], the probability density functions are replaced by similarity measures, for numeric and nominal data. For nominal data, the distance measure used is the Euclidean distance. Formally, this is defined as follows.

$$D_{i_1,i_2} = \left(\sum_{k=1}^{M} \left(\frac{x_{i_1 k} - x_{i_2 k}}{max_k - min_k} \right)^2 \right)^{\frac{1}{2}} \tag{6}$$

Here i_1 and i_2 are two data points, with both M features. This distance is used to get a similarity measure. This measure is the following.

$$S_{i_1,i_2} = e^{-\alpha D_{i_1,i_2}}$$

(7)

Here α is defined to be the following.

$$\alpha = \frac{-\ln(0.5)}{\bar{D}} \tag{8}$$

Where $\bar{D}$ is the average distance among all data points. For nominal data, not Euclidean distance is used but Hamming distance. Hamming distance is $|x_{i_1 k} = x_{i_2 k}|$ is 1 if $x_{i_1 k}$ is equal to $x_{i_2 k}$ and 0 otherwise. Using this distance measure, the similarity measure is easily derived from the distance measure.

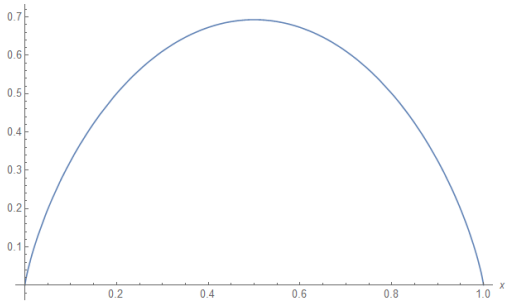
$$S_{i_1,i_2} = \frac{\sum_{k=1}^{M} |x_{i_1 k} = x_{i_2 k}|}{M} \tag{9}$$

Now that for both numeric and nominal data the similarity has been defined, we can define the entropy. This is the similarity measure substituted in equation 5.

$$E = -\sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \left(S_{i_1,i_2} \log(S_{i_1,i_2}) + (1 - S_{i_1,i_2}) \log(1 - S_{i_1,i_2}) \right) \tag{10}$$

Similarity S_{i_1,i_2} between x_{i_1} and x_{i_2} is high if the two points are very close, and low if the points are far away. Entropy E_{i_1,i_2} will be low if S_{i_1,i_2} is either high or low, and E_{i_1,i_2} will be high otherwise. This is explained in the graph below. On the x-axis is similarity S_{i_1,i_2}, and on the y-axis the entropy E_{i_1,i_2}.

Figure 8: Entropy plotted against similarity



In the paper of Varshavsky et al. [15], another feature selection method based on entropy is defined. The main difference between this method and the previous method is that this method uses the Singular Value Decomposition (SVD) of the data. Using this SVD, the singular values of the data set can be calculated. The squares of these singular values are the eigenvalues of the AA^T matrix, with A being the data. The resulting matrix AA^T is then a n by n matrix. The amount of singular values that are calculated using SVD, is $p = \min(n, m)$. Here n is the number

of samples, and m is the number of features. Using the potato data set, this results in 104 singular values calculated. Using these singular values s_j^2, V_j can be calculated with the following formula.

$$V_j = \frac{s_j^2}{\sum_k s_k^2} \tag{11}$$

Here s_j^2 is the j^{th} singular value of A. This formula is derived from Wall et al. [16] and can be seen as the normalized values of the singular values. Using these normalized values, the entropy can be calculated. Therefore the following equation is used, from Alter et al. [4].

$$E = -\frac{1}{\log(N)} \sum_{j=1}^{N} V_j \log(V_j) \tag{12}$$

Here N is the number of features in the data set. Now that the method to calculate the entropy has been defined, the 'contributed entropy' can be defined. This is simply the entropy of the system, minus the entropy of the system without a certain feature. Mathematically, this is the following.

$$CE_i = E(A_m) - E(A_{m-1}) \tag{13}$$

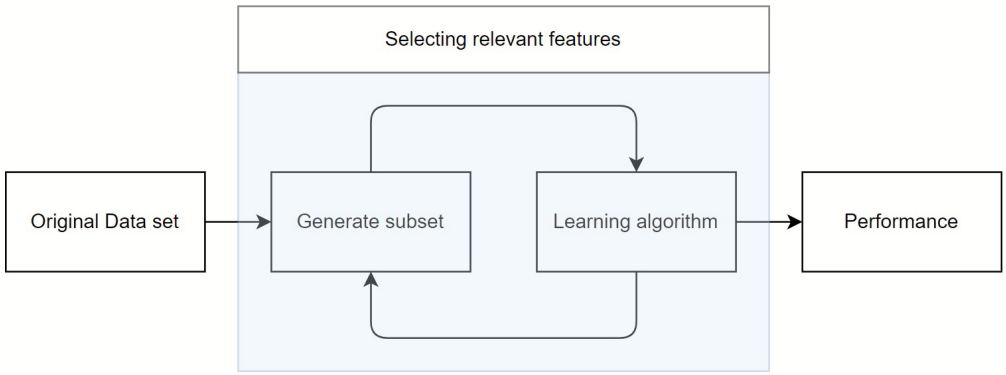
Here A_{m-1} means the matrix A with one feature removed. In the paper of Varshavsky et al. [15], a method is proposed to select features based on a simple ranking. The paper differentiates between three distinct groups of features.

1. $CE_i > c + d$
2. $c + d > CE_i > c - d$
3. $CE_i < c - d$

Where c is the average of all CE and d is the standard deviation. Features that are in the first group contribute to an increase in entropy, hence these are important. The features in the second group are neutral, as the entropy changes very little when removing these features. The third group consists of features for which the entropy decreases. These features contribute uniformly and are also not selected. Both of the entropy methods are applied to the potato data sets in section 5. Here the algorithms used will be described in greater detail and will be compared based on their strengths and weaknesses.

4.2 Wrapper methods

Figure 9: Overview of wrapper feature selection method



Wrapper methods in feature selection were first proposed by Kohavi et al. [10]. According to Guyon and Elisseeff (2003), wrapper methods are defined as follows. *In its most general formulation, the wrapper methodology consists in using the prediction performance of a given learning machine to assess the relative usefulness of subsets of variables.* A wrapper works by selecting a subset of features using a search algorithm, after which this subset of features is used in a learning algorithm. This learning algorithm is then evaluated, after which the usefulness of this subset of features can be evaluated. Wrapper methods are generally more difficult than filter methods to implement, and more computationally costly. Even though wrapper methods are more prone to overfitting, they generally perform better than filter methods. A wrapper method consists of the following three things that need to be defined.

1. Determine a search algorithm to find the variable subsets
2. Determine the learning algorithm, or in the case of unsupervised wrapper methods, the clustering algorithm
3. How to evaluate the learning or clustering algorithm

The paper of Dash et al. (2000) extends upon the filter proposed in the previous chapter, by introducing a wrapper method. Normally, a search algorithm is required to find suitable feature subsets. But because the paper first uses a filter method to get an ordered ranking, no search method is needed in this case. This significantly reduces the complexity of the wrapper method, because now features can be selected one by one, with an already existing ordered ranking. The method uses a k-means algorithm to cluster the data. K-means is an iterative cluster algorithm that tries to make k amount of clusters from the data. Random points are chosen, after which each data point is connected to this random point, which becomes the centroids of the clusters. In the next step, the data points are put into a new cluster if there is another cluster closer to the one the data point is currently in. Closer in this context is dependent on the distance metric used, in the case of the paper of Dash et al. it is Euclidean distance. The centroids are then again moved to the centers of the clusters. A certain amount of iterative steps are done until the k-means algorithm is completed. In the paper of Dash et al., this k-means is first done with the most important feature. Thereafter the two most important features. The clustering quality is measured using an invariant criterion. We start by defining the scatter matrix for the jth cluster $P_j = \sum_{X_i \in \chi_j} (X_i - m_j)(X_i - m_j)^T$. Here X_i are the data points within cluster χ_j, and m_j is the mean vector for the jth cluster. This matrix bears similarity to the sum of squares matrix. Then we define the sum of the matrices of all clusters to be the following $P_W = \sum_{j=1}^{k} P_j$. This matrix is defined to be the 'within-cluster' scatter matrix. We also define the 'between-cluster' scatter matrix, $P_B = \sum_{j=1}^{c} (m_j - m)(m_j - m)^T$. Here m is the mean vector for all the clusters. Using these two matrices, we compute the final scatter criteria.

$$tr(P_W^{-1} P_B) \tag{14}$$

Where tr means the trace of the matrix, the sum of the diagonal elements. Using this criterion, we can choose the features in two ways. We can set a certain threshold for the cluster quality, or we can graph the results and see if the quality of the clusters stops increasing after adding a certain amount of features.

The search algorithm is used to explore the combinatorial space of feature subsets. The problem for exhaustive search is proven to be NP-hard, (Amaldi and Kann [5]). This makes exhaustive search only practical for small data sets. A few search algorithms which are commonly used are the following [5].

1. Forward selection. This search method is characterized by starting with an empty or sparse variable subset. During each iterative step, the remaining variables are evaluated in a small sample. The most promising variables are then added to the subset until a satisfactory subset has been found.
2. Backward elimination starts with the entire subset of features from the original data set. Same as in forward selection, an evaluation method is iteratively run. This method then discards the least promising variables.
3. Simulated annealing. Simulated annealing is a type of heuristic search algorithms, and is often used when the problem is discrete. It starts in a certain state. In this state, all neighboring states are evaluated based on a certain metric. It probabilistically then decides whether or not to move to this new state, or staying in the same state. This iteration goes on until a sufficient state is reached or until a certain computational budget has been expended. A famous example where simulated annealing is used is the traveling salesman problem.

Another clustering method used for unsupervised learning is hierarchical clustering. Hierarchical clustering is a method that tries to cluster the data in different hierarchies. There are two different types of hierarchical clustering, agglomerative and divisive. Agglomerative clustering is also called 'bottom-up' because it starts with all observations in separate clusters. When moving up the hierarchy, similar clusters are merged as you move up the hierarchy. In divisive clustering, also called 'top-down' clustering, the reverse is true. All observations start in one cluster, and then as you move down the hierarchy, observations are split into separate clusters. A notable advantage of hierarchical clustering compared to k-means is that it has a clear visual understanding in the form of a dendrogram. Especially with genomic big data, hierarchical clustering is often used due to this fact.

Two other types of learning machines are simple least-squares and support vector machines. Least-squares is one of the most used forms of prediction. This method works by minimizing the square of the residuals.

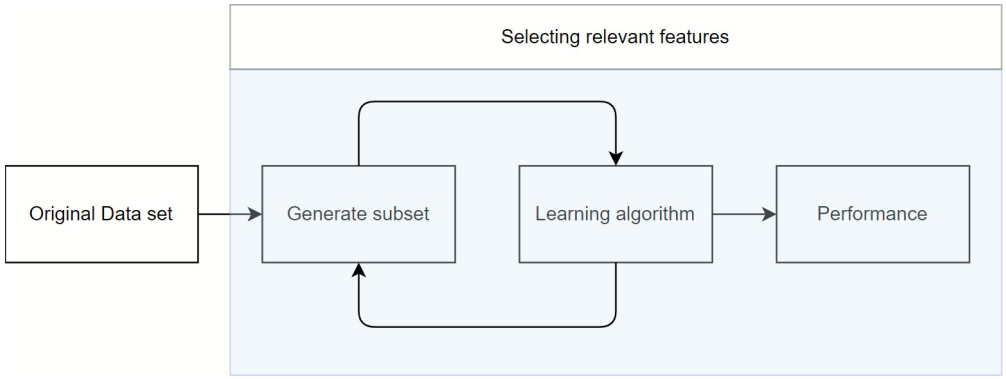
$$SS = \sum_{i=1}^{n} r_i^2 = \sum_{i=1}^{n} (y_i - f(x_i, \beta))^2 \tag{15}$$

Support vector machines work by creating hyperplanes which separate different classes, and maximizes the margin. Support vector machines are one of the most used forms of predictors. The downside is that it can not be used for unsupervised feature selection.

Lastly, the chosen learning or clustering algorithm needs to be evaluated. This can be done in multiple ways, the most common way is a validation set. This is a type of validation in which the training model is trained on a certain set, and then a different set is used to validate the outcome of the model. This can be done using internal or external validation. In internal validation, a certain subset is chosen from the data set to act as the validation set. The downside of this method is that the validation set is not independent, but it is easy to generate. External validation does not have this problem, but the downside is that this external validation set needs to be available which is often not the case, especially in unsupervised dimensionality reduction.

4.3 Embedded methods

Figure 10: Overview of embedded feature selection method



Embedded methods are a combination of filter and wrapper methods, and are very similar to wrapper methods. Embedded methods are the methods that are the most dependent on the problem and the data set. That is why it is difficult to generalize embedded methods. Because embedded methods are a combination of filter and wrapper methods, they generally use the same methods. The algorithm of Dast et al. (2000), is not an embedded method, even though it uses a filter and wrapper method. This is because the filter and wrapper methods are used in different stages of the algorithm, and do not work simultaneously. Embedded methods do use filter and wrapper methods simultaneously.

An example of an embedded method is LASSO (least absolute shrinkage and selection operator) [9]. When given the problem of a regression model, with x_{ij} the standardized predictors and y_i the centred response value, LASSO solves the following optimization for β.

$$\beta_j = \sum_{i=1}^{N}(y_i - \sum_j x_{ij}\beta - j)^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{16}$$

This is almost equal to minimizing the least square, with an added extra constraint of the form $\sum |\beta_j| \leq s$. It uses cross-validation to choose the penalty factors so that LASSO will still perform well with large data samples.

4.4 Feature extraction methods

Feature extraction methods are different compared to feature selection methods, as feature extraction constructs new features instead of selecting them. There are two types of feature extraction, linear and non-linear [6]. Linear feature extraction methods tend to be faster and easier to interpret. Non-linear methods are more complicated but are better at constructing features of complex data structures.

One of the most used techniques for feature extraction is principal component analysis (PCA) [17]. PCA works by creating a linear mapping of the data to a lower-dimensional space, in such a manner that the variance in the lower dimension to which it is projected, is maximized. Usually, this is done by constructing the covariance matrix of the data. Thereafter the eigenvectors of this covariance matrix are computed. The eigenvectors which correspond to the largest eigenvalues, the principal components, can now be used for PCA to reconstruct the large data set. Because the eigenvalues are ranked by magnitude, the original data set can largely be reconstructed by these eigenvectors. Principal component analysis is very efficient and widely used in statistics and other fields. PCA has been used successfully in the past on microarray data of cancer patients. These methods were *highly effective in identifying important features of the data* [6]. A drawback of PCA is that it cannot easily construct principal components of data which has non-linear underlying relations.
PCA is closely related to Singular Value Decomposition (SVD). In SVD a matrix an mxn matrix A is transformed to the following.

$$A = U\Sigma V^T \tag{17}$$

Here the columns of V^T are the principal directions, and the columns of $U\Sigma$ are the principal components. Furthermore the singular values are related to the eigenvalues of the covariance matrix via the following equation.

$$\lambda_i = \frac{s_i^2}{n-1} \tag{18}$$

Another linear feature extraction method is Linear Discriminant Analysis (LDA). LDA seeks to optimally separate samples in a data set by their class value. That is why LDA can only be used for a supervised data set. LDA works more specifically by finding a linear combination of the input variables which optimally separates the samples between the classes. Optimally means maximum separation between classes and minimum separation within each class. There are different ways to implement an LDA algorithm. Most often it is implemented with a matrix factorization technique, like PCA. LDA is closely related to ANOVA (Analysis of Variance). The difference is that ANOVA uses categorical independent variables and a continuous dependent variable. As opposed to LDA, which uses continuous independent variables and a categorical dependent variable.
Related to PCA is a non-linear feature extraction method, kernel-PCA. Kernel-PCA is an extension of PCA, and used for data sets which are not linearly separable. Kernel-PCA makes use of the kernel of the matrix.

$$\mathcal{K}(x) = \phi(x)\phi^T(x) \tag{19}$$

Here $\phi(x)$ is the non-linear mapping from the data set to a lower dimension, $m > n$. The kernel function can then be used in the same way as the covariance matrix in PCA. This means the eigenvalues of the kernel function are computed, together with the eigenvectors, which form the principal components. Kernel-PCA is generally more difficult than PCA, but the benefit is that it also works for data with underlying non-linear relations.

5 Methods

5.1 Motivation for methods used

As stated before, unsupervised feature selection is generally harder than supervised feature selection. For the data set that is examined in this paper, there is no access to the following. There is no domain knowledge of the subject. This means that it is difficult to evaluate chosen features based on a-priori knowledge. Secondly, there is no training data or target variable. An unsupervised method must thus be chosen. That is why is chosen for an unsupervised filter method, as opposed to a wrapper method. Methods based on entropy are chosen because from the literature it seemed that entropy-based methods were most promising as unsupervised filter methods.

In this chapter, two algorithms will be applied to the potato data set. These are the algorithms proposed in the papers of Dash et al. (2000) and Varhavsky et al. (2006). For ease of notation, the algorithm of Dash et al. (2000) will be referred to as the 'Dash algorithm'. For the algorithm of Varshavsky et al. (2006), the name will be 'Varshavsky algorithm'. Both methods will be explained and researched in more detail in their effectiveness. Extra focus will be laid on the influence of manipulating the data beforehand.

5.2 Methods: explanation of the 'Dash' and 'Varshavsky' algorithms

In the first example, we will look at an unsupervised, univariate filter method proposed in the paper of Dash et al. [7]. The way this method works is explained in section 4.1. After that, the algorithm of the wrapper method is introduced in section 4.2. Here the implementation in R will be discussed.

The method consists of two parts, named in the paper as the RANK and SELECT algorithms. The RANK algorithm is a filter method that ranks a subset of features, whereas the SELECT algorithm is the wrapper method. The RANK algorithm is not directly used, but rather the SRANK algorithm, short for Scalable RANK. This is because the distance metric is used for this method. Therefore if RANK was immediately ran on the entire data set, the size of the distance matrix would be $\sum_{i=1}^{47,581} i \approx 450,000,000$. This is not a viable option on a normal computer, therefore it is done in small subsamples instead of the entire data set in on time. In table 3 it can be seen that the time it takes to run the RANK algorithm quickly balloons as the size of the subset increases. The paper suggests taking subsamples with sizes 0.25%, 0.5%, and 1% of the total features. The amount of times the SELECT algorithm is run is chosen to be at least 35, as *35 is often considered the minimum number of samples for large sample procedures* [3].

Table 3: Computation times of RANK algorithm R's SYSTEM.TIME function

Size of subset	**Computation time (s)**
50	0.09
100	0.36
500	36.75

Algorithm 1: RANK algorithm: filter method to rank features based on entropy

set **number of runs** to be at least 35;
set **sub sample size** either 0.25%, 0.5% or 1%;
create **results** a vector where the results will be stored;
m is the number of features;
for *i in 1:* ***number of runs*** **do**
  sample data into sub samples of **sub sample size** of the features, for all potato samples;
  for *i in 1:* ***number of samples*** **do**
    remove i^{th} feature from matrix;
    calculate **distance** of the sub sample with i^{th} feature removed using either Euclidean or Hamming distance ;
    compute **similarity measure** as defined in equation 10;
    store **Entropy** in **results**;
    place i^{th} feature back in matrix;
  end
end
Return(results);

In their paper, Dash et al. [7], state that a reasonable small sub-set still retains the original cluster information in most cases. This is a dubious claim, which will be shown in the results. The following algorithm, the SELECT algorithm is the wrapper method, explained in section 4.2.

Algorithm 2: SELECT Algorithm: wrapper method using k-means

set **feature amount threshold** as the maximum features that are required, this is an estimated guess. If too small, make it bigger in the next run;
input is the **ranked list of features**;
for *i in 1:* ***feature amount threshold*** **do**
    take the first i features from the ordered ranking;
    run k-means algorithm using these features from the data set only;
    return **trace criterion defined in equation 14**;
end
plot **trace criterion** versus amount of features and determine using the graph how many features are required;

Finally, the implementation of the algorithm is proposed in the paper of Varshavsky et al. (2006). This method is explained in section 4.1.

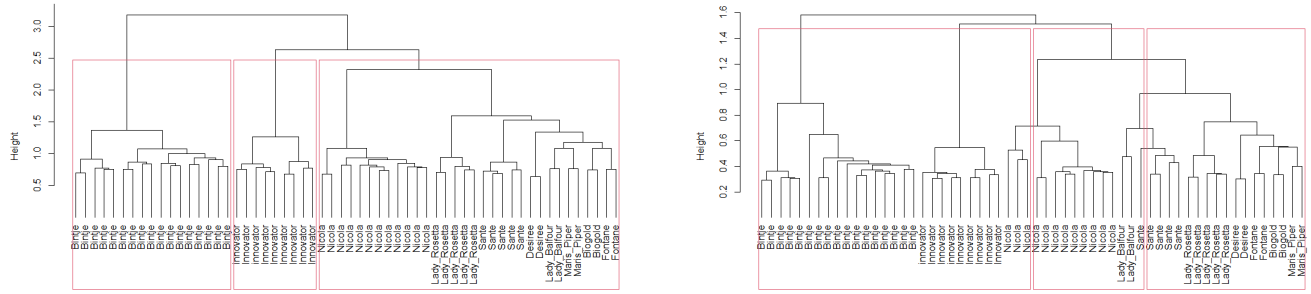
Algorithm 3: Varshavsky algorithm: entropy-based filter method using SVD

create **resultVector** this will be the result vector;
m is the number of features;
calculate **Entropy** with all features;
for *i in 1:* ***m*** **do**
    remove i^{th} feature from data;
    calculate singular values of data using SVD;
    calculate V_j;
    compute **Entropy** using equation 12;
    store CE_i in **resultVector**;
    put i^{th} feature back;
end
Return(resultVector);

5.3 Performance measure of methods

To measure the performance of the different methods, hierarchical clustering is used. Hierarchical clustering is a form of clustering, explained in greater detail in section 4.2. Both of the methods will be compared to each other, and no dimensionality reduction. The hierarchical clustering algorithm will be ran using the dissimilarity matrix, which equals $1 - C$, where C is the correlation matrix. Below an example of hierarchical clustering is shown.

Figure 11: Hierarchical clustering using two different methods with 3 clusters shown



In figure 11 two different hierarchical clusters are shown, both with 3 clusters highlighted. In the left figure, we can see that all commercial potatoes are grouped in the same cluster. This means that the cluster method works well,

and the number of faults is 0. In the right figure, this is not the case. Here it can be seen that some potato types, for instance, the 'Nicola' potato, are split into different clusters. In the right figure, it can be seen that there are 4 potatoes placed in the wrong cluster (3x Nicola and 1x Sante). This method is used in the following section, the results. In Appendix section A; 9.1 more examples are shown of this hierarchical clustering.

6 Results

Below the results of the hierarchical clustering explained in the previous section are applied to different types of data. In the table for each data pre-processing, the entire data set is used, and the Dash and Varshavsky methods. In the brackets, the amount of features is shown which are used for corresponding methods. The results start by first applying the hierarchical clustering algorithm to the raw data. These results are presented below.

Table 4: Amount of potatoes wrongly clustered using raw data

Features used / # clusters	All features (47,582)	Dash et al. (600)	Varshavsky et al. (42)
2	7	10	13
3	11	19	13
4	15	21	15
5	19	25	20
7	23	27	23
10	23	27	23

What can be seen is that for both methods, and all the data, the results are not very good. Because the data has not been scaled, there is a large difference between the expression of genes. Some genes have averages of around 0, while some are higher than 100,000. This large difference makes it very difficult to cluster the data or to use dimensionality reduction.

When the data is scaled, the results are different. There are many ways that data can be scaled, one of the ways that is most often used is by scaling the data in the range of [0,1] using the following formula for gene X_i.

$$\bar{X_i} = \frac{X_i - \max X_i}{\max X_i - \min X_i} \tag{20}$$

Besides this scaling, the genes are removed which only have 0 entries. This is because these genes do not affect the methods used. The results are shown in the table below.

Table 5: Amount of potatoes wrongly clustered using data which has been scaled in the range [0,1]

Features used / # clusters	All features (35,091)	Dash et al. (600)	Varshavsky et al. (4,377)
2	13	13	2
3	14	13	2
4	16	13	2
5	17	13	2
7	17	14	7
10	17	14	13

The results for the entire data set and the Dash algorithm do increase compared to no data pre-processing. But the real difference is in the Varshavsky algorithm. This method provides much better results than no scaling or using the entire data set. It can be seen that the amount of features has been reduced from 47,582 to 4,377 while improving the cluster quality. The paper of Dash et al. proposes another similarity metric based on the Hamming distance instead of the Euclidean distance 4.1. This gave rise to the idea of transforming the data to binary form, meaning that 0 entries stay 0 and entries larger than 0 become 1. This provides the following results.

Table 6: Amount of potatoes wrongly clustered using data which has been scaled to binary form

Features used / # clusters	All features (21,221)	Dash et al. (600)	Varshavsky et al. (2,014)
2	3	0	0
3	4	0	0
4	11	0	0
5	12	0	0
7	17	5	5
10	19	7	9

This method provides the best results. The algorithm of Dash and Varshavsky perform almost equally well, but the algorithm of Dash does it with much fewer features. Transforming the data into binary form, removing all genes which are either 0 or fully expressed, and then running the Dash algorithm has the effect of improving the cluster quality massively with much fewer features (600 instead of 47,582). A notable advantage of the second method is that SVD is very efficient. This means that it is not required for this problem to separate the features into sub-sets. When looking at algorithm 3 this can be seen. This means that the contributed entropy of the features is not dependent on the sub-set in which it is in. The entropy of the SVD algorithm takes into account the interplay between all features. Therefore the entropy calculated in this method takes not only the variance into account but also correlations between features. Besides this, the algorithm is quicker.

7 Discussion

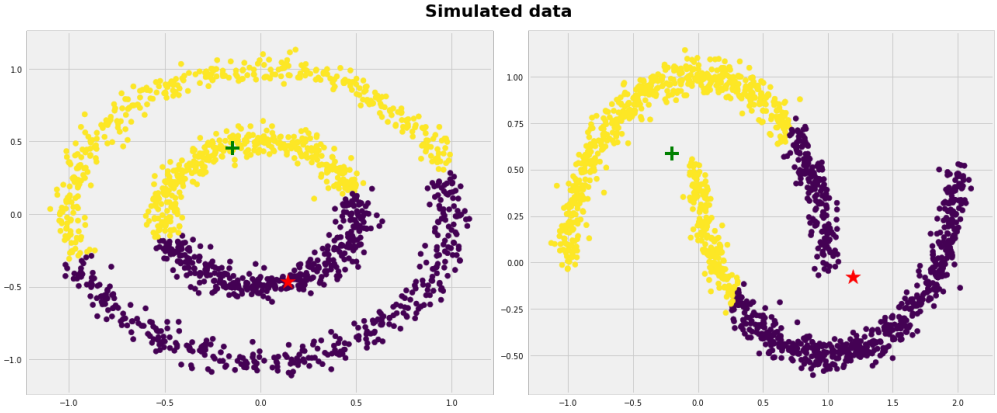
7.1 Unexpected results

One of the things that are interesting to note is the large difference in features selected by the algorithm of Varshavsky et al. For the raw data, 42 features were selected. For the [0,1]-scaled data and binary data, there were respectively 4,377 and 2,014 features selected. This is a very large difference, and important to understand why this happens. For the raw data, the variance in the data is very high. Both scaling methods help to reduce this variance. Because the variance is so high for the raw data, the criterion to select features is also very high. Recall that this criterion is $CE_i > c + d$ where c is the average of the contributed entropy, and d the standard deviation. There are only a few features larger than this criterion, and this leads to lesser results.

7.2 Limitations

One of the limitations of this thesis is the k-means algorithm that was used in the paper of Dash et al. (2000). As explained before, this wrapper method uses k-means to determine the number of features that need to be selected. K-means is a clustering algorithm that works very well when the data has spherical structures, but not so much when the data has complex structures. Due to this fact, the k-means method did not work very well and hence was replaced by the hierarchical clustering method. Below a figure is presented which illustrates an example where k-means is not the right choice for clustering purposes. A limitation of this thesis is that this problem was not researched further because hierarchical clustering was the better choice here. It does not mean that k-means clustering can not work for this problem, but more research is required.

Figure 12: An example where k-means clustering is not working properly. The two colors indicate the two clusters. As can clearly be seen, the colors do not match the clusters.



Two problems arise surrounding the hierarchical clustering method. First of all, the hierarchical clustering method was used in replacement of the k-means wrapper method in the algorithm of Dash et al. Besides this, it was also used to evaluate this method. This means there is some form of bias between this method and the evaluation criterion. Besides this, the dissimilarity matrix was used for the hierarchical clustering based on the correlation matrix. Though this provided good results, this is not the only way in which hierarchical clustering algorithms can be run. There are more ways to calculate the (dis)-similarity between data points.

The data scaling method in the range of [0,1] is often used for data pre-processing. The scaling of the data to binary form is not that common. It was heuristically derived and does therefore not have a solid theoretical foundation. This is a limitation of this thesis, as even though this scaling method did provide good results, it does not have a solid biological/theoretical basis. Besides this, more different types of data pre-processing could be used. For instance the removal of features with very low variance/standard deviation.

For a clearer visual understanding, commercial potatoes were only displayed in this thesis for the hierarchical clustering. Something that could be researched further, is the addition of supervised learning to this thesis. This could be done by labeling the data for instance as 'commercial' and 'experimental'. This was not done in this thesis as it did not fit within the scope of this project.

All these limitations of this thesis could serve as guiding points for further research into this topic.

8 Conclusion

In section 1, the following research questions were formulated.

- What is the most relevant work done so far in the field of feature selection relevant to the data set that is introduced in this paper?
- Can a method found in the literature research be used for the potato data set, and what are the results of this?

Even though unsupervised dimensionality reduction methods are generally of less interest to scientists than supervised feature selection methods, there are still plenty of methods available. For both feature selection and feature extraction, there are options available. It is difficult, however, to rank the feature selection methods. All methods have notable advantages and disadvantages and are largely dependent on the area of application, whether there is a-priori knowledge, and the goal of feature selection.

Comparing both methods that are applied in this paper, a conclusion can be drawn. The method of Varshavsky performs better for [0,1]-scaled data, but the method of Dash performs better for the binary scaled data. The method of Dash does this with fewer features, so this method can be regarded as the most successful method.

For this data set, the conclusion is to transform the data to binary and use the algorithm of Dash to select features, using hierarchical clustering instead of k-means clustering. The methods introduced in this paper worked on a similar basis, but one presented much better results.

References

[1] Genomics and big data – unlocking the code to new therapies.

[2] *Oxford Dictionary*.

[3] *Probability and Statistics for Engineering and Sciences*. Duxbury Press, 4th editio edition, 1995.

[4] Orly Alter, Patrick O. Brown, and David Botstein. Singular value decomposition for genome-Wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 97(18):10101–10106, 2000.

[5] Edoardo Amaldi and Viggo Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2):237–260, 12 1998.

[6] Rabia Aziz, C.K. Verma, and Namita Srivastava. Dimension reduction methods for microarray data: a review. *AIMS Bioengineering*, 4(1):179–197, 2017.

[7] Manoranjan Dash and Huan Liu. Feature selection for clustering. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 1805, pages 110–121. Springer Verlag, 2000.

[8] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection, 3 2003.

[9] K. Kayser, S. D. Jacinto, G. Böhm, P. Fritz, W. P. Kunze, A. Nehrlich, and H. J. Gabius. Application of Computer-assisted Morphometry to the Analysis of Prenatal Development of Human Lung. *Anatomia, Histologia, Embryologia*, 26(2):135–139, 1997.

[10] Ron Kohavi and George H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[11] Esther Kok, Jeroen van Dijk, Marleen Voorhuijzen, Martijn Staats, Martijn Slot, Arjen Lommen, Dini Venema, Maria Pla, Maria Corujo, Eugenia Barros, Ronald Hutten, Jeroen Jansen, and Hilko van der Voet. Omics analyses of potato plant materials using an improved one-class classification tool to identify aberrant compositional profiles in risk assessment procedures. *Food Chemistry*, 292:350–358, 9 2019.

[12] S. B. Kotsiantis. Erratum: Feature selection for machine learning classification problems: A recent overview (Artificial Intelligence Review (2011)), 5 2014.

[13] Saúl Solorio-Fernández, J. Ariel Carrasco-Ochoa, and José Fco Martínez-Trinidad. A review of unsupervised feature selection methods. *Artificial Intelligence Review*, 53(2):907–948, 2 2020.

[14] Khawla Tadist, Said Najah, Nikola S. Nikolov, Fatiha Mrabti, and Azeddine Zahi. Feature selection methods and genomic big data: a systematic review. *Journal of Big Data*, 6(1), 2019.

[15] R. Varshavsky, A. Gottlieb, M. Linial, and D. Horn. Novel Unsupervised Feature Filtering of Biological Data. *Bioinformatics*, 22(14):e507–e513, 7 2006.

[16] Michael E. Wall, Andreas Rechtsteiner, and Luis M. Rocha. Singular Value Decomposition and Principal Component Analysis. In *A Practical Approach to Microarray Data Analysis*, pages 91–109. Kluwer Academic Publishers, 12 2005.

[17] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 2(1-3):37–52, 1987.

[18] Yee Hwa Yang, Yuanyuan Xiao, and Mark R Segal. Identifying differentially expressed genes from microarray experiments via statistic synthesis. *BIOINFORMATICS ORIGINAL PAPER*, 21(7):1084–1093, 2005.

9 Appendix

9.1 Appendix A: Dendrogram

Figure 13: Dendrogram of all features, binary scaled with 5 clusters

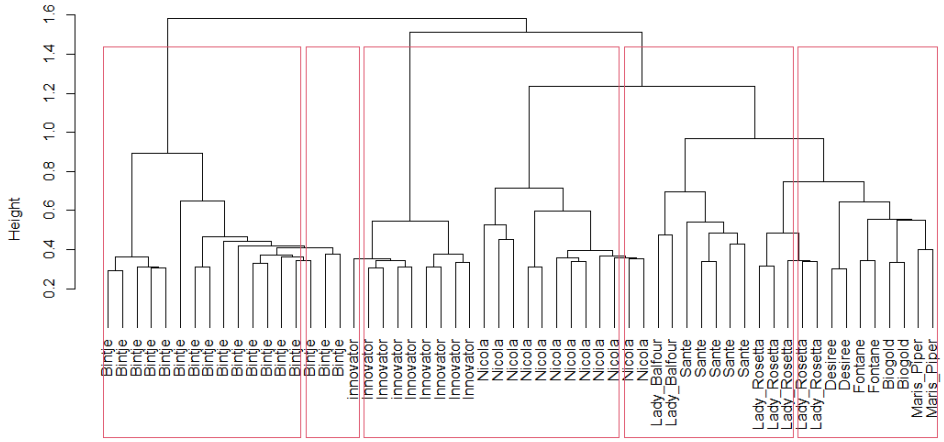


Figure 14: Dendrogram of binary scaled features selected by Dash algorithm with 5 clusters

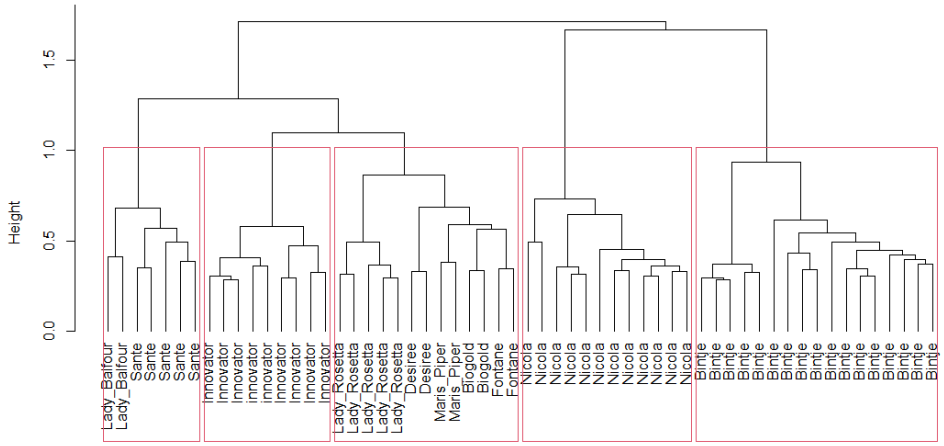


Figure 15: Dendrogram of binary scaled features selected by Varshavsky algorithm with 5 clusters