university of groningen

faculty of mathematics and natural sciences

# DimRedPlot: A Generic Visualisation Tool for Dimensionality Reduced Data

Master's Thesis Computing Science

21st January 2016

Student: K.L. Winter

Primary supervisor: Prof. Dr. A. Telea

Secondary supervisor: Prof. Dr. M. Biehl

External supervisor: Dr. B. Broeksema

## ABSTRACT

Dimensionality reduction techniques can transform datasets with a large number of variables to simpler two or three-dimensional datasets, while preserving distances and structure in the original data as much as possible. This makes these techniques very useful when dealing with large datasets. Unfortunately, the results they produce can be abstract, making it hard to fully understand how these results relate to the original data. As a result, many researchers treat these techniques as simple black boxes, which means they severely underutilise their potential. Most of them also are only capable of either analysing numerical or categorical data, which makes analysing mixed datasets a difficult challenge. This thesis presents DimRedPlot, a tool which, when combined with more general visualisation techniques, allows users to easily see the relation between the results of linear dimensionality reduction techniques and their original data. The focus on linear techniques, such as Principal Component Analysis, is due to the fact that they have been widely used for decades in a wide range of applications. Because of the support of both Principal Component Analysis, capable of analysing numerical data, (Multiple) Correspondence Analysis, capable of analysing categorical data, and the ability to combine these analyses on one screen, DimRedPlot greatly simplifies working with mixed datasets. DimRedPlot has been designed and evaluated at the Luxembourg Institute of Science and Technology, or LIST, and it has been integrated into the larger RParcoords environment developed there. The evaluation was performed using two datasets generated and used at the institute, and DimRedPlot continues to be used by researchers at the LIST.

# CONTENTS

# 1 INTRODUCTION

High-dimensional data is nowadays a common occurence in both research and industry. This data is characterised by having a high number of features or variables per observation in a dataset. The number of variables can easily run into the thousands or higher in some of these datasets. The availability of such datasets has been facilitated by several factors. The increase in available computing power has enabled their creation as a results of, e.g., large simulations. An example would be global climate simulations, which can easily result in datasets containing thousands of locations as observations and thousands of variables such as temperatures at many different time points. Similarly, the development of cheaper and better sensors have also made it easier than ever to create large datasets containing the output of potentially hundreds of sensors at different time points. Finally, databases in all kinds of areas, ranging from social media to insurance companies, are continuously increasing in size.

Having access to large datasets can be crucial for research. Many areas of research, such as the climate, are complex and have many aspects to them. Generating large datasets that encompass as much of this complexity as possible allows researchers to gain more insight about the topic at hand and to gain more sound conclusions about it. The same can be said for business, where insight gained from large datasets can be instrumental in determining company policy.

Unfortunately, analysing a high-dimensional dataset can be quite hard. Due to the sizes these datasets have, it is hard to determine where to look in the data to find the insight and conclusions researchers are looking for. Especially the presence of high numbers of variables is what makes it difficult to gain information from them. Some variables may not be important and can be ignored, while others are very important for the overall structure in the data. It is, however, not trivial at all to find out which variable is which. Furthermore, complex relations can be hidden in groups of variables, which are not easily found when just looking at one or two variables at a time.

To attack this problem, dimensionality reduction techniques are widely used. These techniques project the observations in high-dimensional datasets onto a manageable two or three axes. This means that large datasets can suddenly be much easier to explore and analyse. There are many different dimensionality reduction techniques and the amount in which each are used varies. One of the oldest and well known techniques, Principal Component Analysis, or PCA, is used in almost every scientific discipline and has been used at least as early as 1933 [1]. Techniques such as PCA can be used for many different goals, but simplifying datasets for analysis is one of the more common usages.

Although the number of different dimensionality reduction techniques is large, in this thesis we focus only on Principal Component Analysis, Correspondence Analysis (CA), and Multiple Correspondence Analysis (MCA). These techniques are all linear, meaning that the transformations they apply to observations to project them onto new axes are linear. Non-linear techniques also exist, such as Multi Dimensional Scaling and t-SNE, but due to the transformations they perform being more mathematically complex, they are harder to understand and interpret. The linear techniques, and especially PCA, are also widely accepted by many researchers and they have been in use for decades in a large number of fields. Due to this, the primary focus of this thesis is on these three linear techniques.

Even though dimensionality reduction is widely used, several such techniques are insufficiently well understood by many researchers. The results produced by them are abstract. It is not clear why the generated projected points are close together or how the projections relate to the original data. This makes it hard to interpret them. As a result, dimensionality reduction techniques are often used as black boxes where datasets are given as input and a 2 or 3-dimensional scatterplot is created as output. Treating them as simple black boxes can still yield some new insight into the original data; however, a better understanding of how these techniques work and how their results link to the original data can vastly increase the amount of insight into that can be gathered. This lack of understanding can have the effect that even though dimensionality reduction techniques can potentially give the user the insight or answer he or she is looking for, the user is unable to actually find these insights and answers.

Beyond the treatment of black boxes that is given to these techniques, there is also the problem that many datasets can not be analysed with just one of these methods. Whereas PCA can analyse only purely numerical data, CA and MCA are designed to be used purely with categorical data. However, many datasets are more complex than merely numerical or categorical and contain a mix of both types of variables. These datasets can not be analysed using just one method, which makes it hard for researchers to use dimensionality reduction to see how these two parts of a dataset relate to each other. There are of course general visualisation techniques that allow both types of variables to be shown, such as parallel coordinates; however, these are usually severely limited in the number of variables they can display.

In this thesis we present DimRedPlot, a visual analytics tool designed to visualise the results of Principal Component Analysis, Correspondence Analysis, and Multiple Correspondence Analysis. DimRedPlot visualises the results of dimensionality reduction in a scatterplot, as users will be familiar with and used to, but it combines these scatterplots with a set of features that help users understand the technique they are using and how what they see can be explained in terms of their original data.

Although DimRedPlot can be used as a stand alone tool, it has been designed to be used in combination with more general visualisation techniques, such as parallel coordinates, that show the original data. In particular, DimRedPlot has been integrated into RParcoords, a parallel coordinates visualisation tool developed and used at the Luxembourg Institute of Science and Technology[1], or LIST, as a exploratory environment for multivariate data. However, DimRedPlot can, due to its general and modular design, theoretically be used with any visualisation tool. The interactions designed between DimRedPlot and RParcoords allow a user of these tools to quickly obtain insight about how the projected dimensionality reduced structure seen in DimRedPlot relates to the original data. It also makes iterative dimensionality reduction very easy, which means that even if the first attempt at dimensionality reduction bears no fruit, the parameters used can quickly be refined in order to obtain clearer or more useful results.

RParcoords also allows multiple DimRedPlot instances to be shown at once. This means that a user can use one DimRedPlot instance to display the results of PCA for the numerical variables in the data, while at the same time another DimRedPlot instance can show the results of MCA for the categorical variables in the data. The different DimRedPlot instances are interactive, meaning that through user interaction users can explore how the numerical part of their data relates to the categorical part.

1 http://www.list.lu

RParcoords and DimRedPlot have been slightly modified to serve as a bioinformatics tool for contig binning. The tool is known as ICoVeR [2], and it is available online through `http://www.github.com/bbroeksema/ICoVeR`.

In general we can say that, using DimRedPlot and RParcoords, this thesis tries to answer the following question:

> How can we, through linked visual metaphors, support the exploration and interpretation of dimensionality reduction on complex high-dimensional datasets?

The structure of the rest of this thesis is as follows. In Chapter 2 we discuss related work by looking at general visualisation techniques, analytical tools, such as PCA, and existing visual analytics tool that visualise the results of dimensionality reduction. For full understanding of the design of DimRedPlot, some background in PCA, CA, and MCA is needed. These techniques are more deeply explored in Chapter 3.

Chapter 4 discusses the design of DimRedPlot and the user interactions it offers. DimRedPlot itself has been integrated into RParcoords and both RParcoords and the user interaction that comes with this integration are the topic of Chapter 5.

The combination of RParcoords and DimRedPlot has been evaluated using datasets that are being used by researchers at the Luxembourg Institute of Science and Technology. The results of these evaluations are discussed in Chapter 6.

Finally, in Chapter 7 we end the thesis with the conclusion, along with a discussion of several features that were thought out but were not added to DimRedPlot and RParcoords due to time constraints.

# 2 RELATED WORK

In many fields of study, researchers have to explore and analyse complex high-dimensional datasets. Where simpler datasets can be explored and analysed using simpler techniques such as scatterplots and t-tests, high-dimensional datasets are too complex to be dealt with this way. More complex visualisation and analysis techniques are necessary to gain insight into these datasets.

One way to gain insight into high-dimensional datasets is by trying to visualise all or most of the variables in the data at the same time. Many visualisation techniques exist that try to achieve this, but often there are simply to many variables to fit on a screen. And even if all the variables can be visualised at once, exploring them may still not be an easy task.

Another approach is to apply analytical techniques that can analyse the dataset and reduce its dimensionality. After applying such a dimensionality reduction technique on the dataset, visual analytics tools can be used to explore these reduced datasets using visualisation techniques and user interaction. This allows for complex high-dimensional datasets to be strongly simplified and makes it easier for researchers to analyse their datasets.

In the next sections, we begin with discussing several existing visualisation techniques that try to visualise as many variables in the dataset as possible in one go. After this we take a look at both some dimensionality reduction techniques that can simplify large datasets and at some of the existing visual analytics tools that visualise the results of these techniques.

## 2.1 HIGH-DIMENSIONAL VISUALISATION TECHNIQUES

As mentioned, several visualisation techniques exist that try to visualise an entire dataset or a large portion of a dataset at once. This has the benefit that users do not necessarily have to make a selection of variables or observations before visualising them, which is useful because users often do not know in advance what variables or observations are interesting or important. In the following sections we explore some of the existing high-dimensional visualisation methods and we discuss why these techniques on their own are not sufficient solutions for the problem we are trying to solve.

### 2.1.1 *Permutation matrix*

A permutation matrix [3] is essentially a data-table, only instead of showing numbers in every cell, a bar is used. The height of each bar indicates what numerical value each cell has. Compared to a regular data-table, a permutation matrix is also transposed, as the rows represent the variables, while the columns represent the observations. An example of a permutation matrix can be seen in Figure 1.

The usage of bars instead of numbers makes it easier to detect patterns in the data, since bars are easier to visually compare than numbers. The permutation matrix as a visualisation technique is, however, very flexible and there are many different ways each value can be encoded instead of rectangle height. Examples of encodings are by making the rectangles equally sized and colouring them or by making the rectangles the same shape and varying their sizes. To detect actual patterns in the data, the rows and columns in the permutation matrix are moved around, or permutated, until patterns appear.

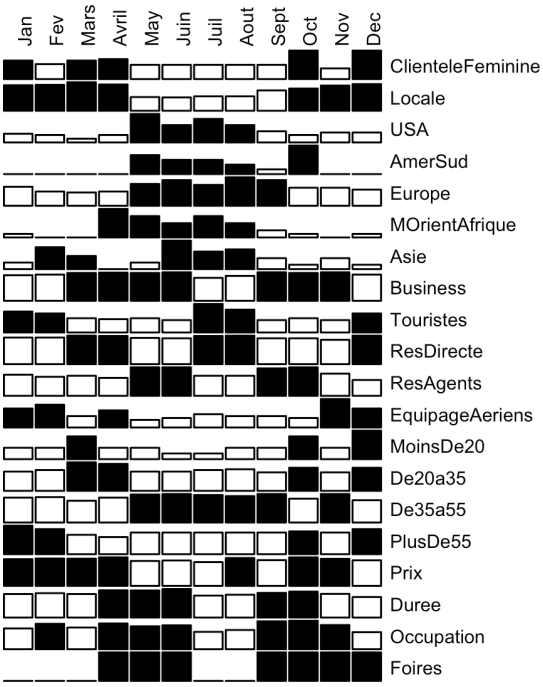


Figure 1: Permutation matrix showing a data set about hotel occupation throughout the year.

Although permutation matrices are very flexible in the data they can visualise and how the visualisation takes place, they are quite limited in how much data can be visualised at once. The rectangles are limited in how small they can be, which means that screen space will quickly run out. Also, to find patterns in the data, the user needs to keep permutating the rows and columns in the hopes of finding some. Depending on the size of the data-set this can be very time-consuming, without guarantees of success even if patterns in the data exist.

### 2.1.2 *Table lens*

Similarly to permutation matrices, table lenses [4, 5] visualise a dataset as a data-table. Table lenses show the observations on the vertical axis and the variables on the horizontal axis. Numerical variables are represented using vertical bar plots where every bar represents the associated value of the observation at that vertical height. This way several variables can be placed next to each other. Categorical variables are represented using dots for every observation, whose horizontal distance from the start of the variable indicates the category that observation is in. Figure 2 shows an example of a table lens with both numerical variables and a categorical variable. By default, every observation will occupy one pixel of vertical space, unless more space is available, making the bar plots' bars one pixel high. This means that a table lens can easily display a large number of observations at the same time. Every variable is given just enough horizontal space to display the pattern of the observation bars. To obtain more detailed information about an observation or a variable, a user can zoom in onto a selection of observations and variables, as has happened in Figure 2. Zooming in makes the titles of the observations easily readable and the data values easily discernible.

Although table lenses offer both the possibility of visualising a large number of observations and showing detailed information about them, the number of variables that can be shown on the screen at once is limited,

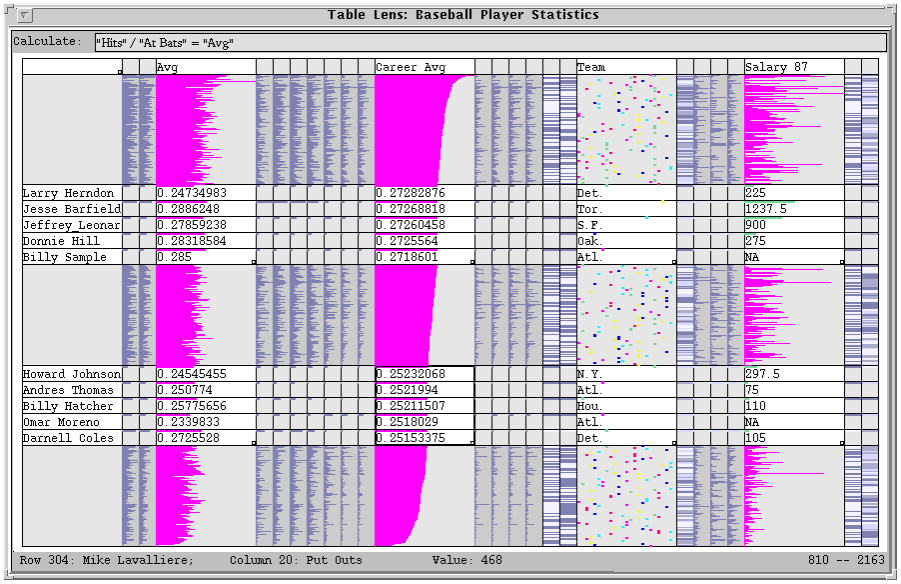


Figure 2: Table lens showing both numerical and categorical variables.

due to screen-size limitations. Furthermore, to find patterns in the data, the observations need to be sorted based on their associated values for a variable, as otherwise a variable will just display a set of bars with seemingly random lengths. Although sorting on a variable is possible, and subsequent sorting of other variables, grouped first by the earlier sorted variables is also a possibility [6], this does require the user to know which variables to sort and to focus on, which is not always something a user knows in advance.

### 2.1.3 *Scatterplot matrix*

The scatterplot matrix technique plots every variable against every other variable in scatter-plots. This means that if we have $N$ variables, we obtain $N \cdot N - N$ scatterplots. These scatterplots are then displayed in a matrix layout, where every row and column represents a variable and every position in the matrix shows the scatterplot between the variables of its row and column. An example of a scatterplot matrix can be seen in Figure 3. The technique is useful to quickly spot the relationships between individual variables. Unfortunately, with a high number of variables, screen space can easily run out. This means that scatterplot matrices can not display datasets with large numbers of variables at once. Furthermore, although scatterplots can show the relationships between two different variables, more complicated relationships involving many variables are harder to find.

### 2.1.4 *Mosaic display*

Mosaic displays [7] are visualisations designed to visualise datasets with categorical variables. The displays consists of a rectangle for each possible combination of categories. This means that if there are 3 variables with 2 categories each, the total number of rectangles is 8. The size of the rectangles corresponds to the frequency at which the respective combination of categories occurs in the dataset. The rectangles are created by iteratively splitting them into a different group for each category. We can see how this works by looking at figure 4. The figure shows a mosaic display visualising data regarding extra-marital and pre-marital sex and the marital status of both men and women, e.g., for every participant it is recorded whether he or

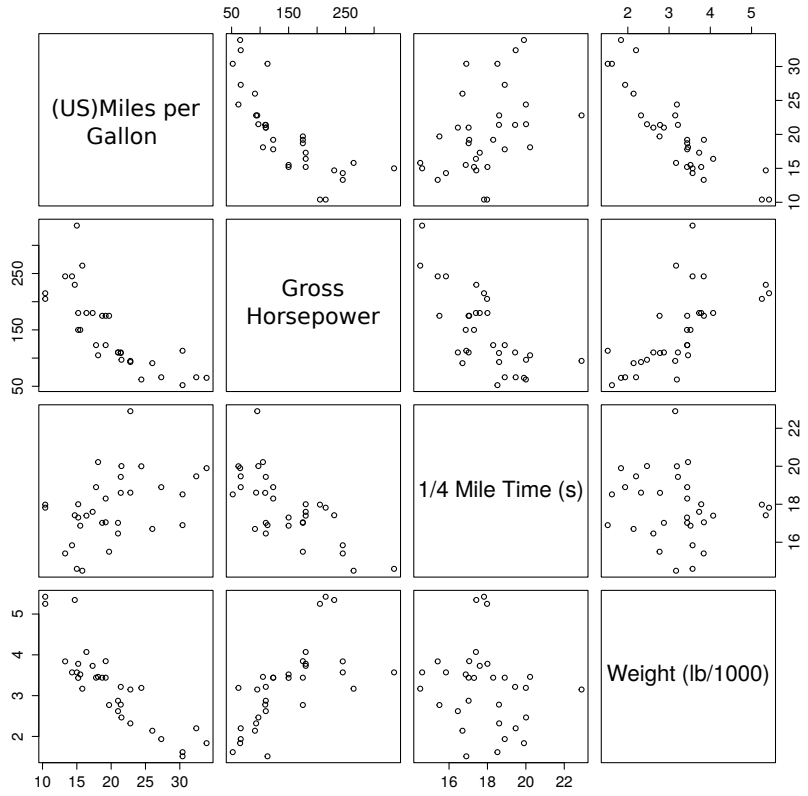


Figure 3: Scatterplot matrix showing a data set from the 1974 Motor Trend US magazine.

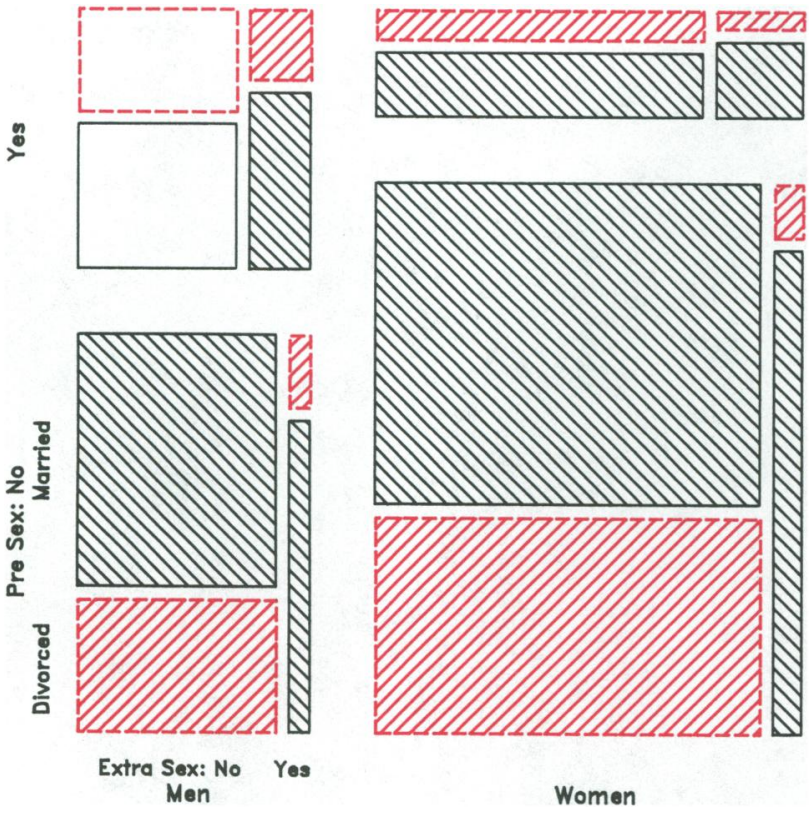


Figure 4: Mosaic display showing data regarding extra-marital and pre-marital sex and the marital status of both men and women.

she is divorced or married, whether he or she has had pre-marital sex, and whether he or she has had extra-marital sex. We can see that the rectangles are first split horizontally over gender, then vertically over pre-marital sex, then each quadrant is split horizontally over extra-marital sex, and finally each quadrant is split over marital status.

As we can see by looking at Figure 4, mosaic displays make detecting correlations in the data easy. For example, when we look at the figure we can quickly see that more women than men participated because the women column is wider. It is also clear that people having had pre-marital or extra-marital sex are more often divorced, and pre-marital sex is more common in men.

Unfortunately, the number of categories that can feasibly be shown at the same time is lacking. The shown dataset contains 8 different categories. In a more complex dataset it is possible to have tens or even hundreds of categories which would make the rectangles very fragmented and hard to interpret. Also, the order in which the categories were used to split the display influences the easiness with which the display can be studied; however, a logical order is not necessarily obvious to a user when exploring a dataset. Finally, mosaic displays only support the visualisation of categorical variables, while we are interested in visualising complex datasets that contain both numerical and categorical variables.

### 2.1.5 *Parallel coordinates*

Parallel coordinates [8] displays every variable as a vertical line, where every vertical line represent the domain of that variable. Every observation is rendered as a poly-line through the vertical lines. The location of the intersection between the vertical lines and the poly-line indicate what value an observation has for the associated variables. An example of this can be seen in Figure 5. Unlike techniques such as scatterplot matrices, parallel coordinates allows for seeing more complicated relationships through multiple variables. However, some inherent data-structure that can easily be seen in scatterplots, such as clusters, may be hard to see in parallel coordinates.

Parallel coordinates can display a large number of both categorical and numerical variables. Unfortunately however, there is still a limit to the number of visualised variables when it comes to screen space, and displaying more then 10 to 20 variables can easily lead to a cluttered visualisation.

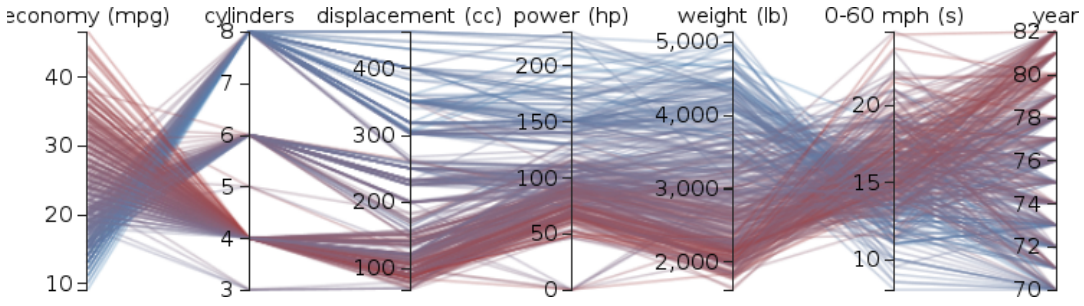


Figure 5: Parallel coordinates showing a datasets from the 1974 Motor Trend US magazine.

### 2.1.6 *Parallel sets*

Bendix et al. [9] introduced an adaptation to parallel coordinates called parallel sets. Parallel sets is designed for use with categorical datasets. Here, the categorical variables are no longer drawn using axes with tick marks,

as is the case in Figure 5. Instead, every category of a variable is rendered as a rectangle along the axis of the variable, with every rectangle's size corresponding to the frequency of that category. The polylines used in the parallel coordinates visualisation are replaced by bands whose thickness represents the number of observations in the band. An example of this can be seen in Figure 6.

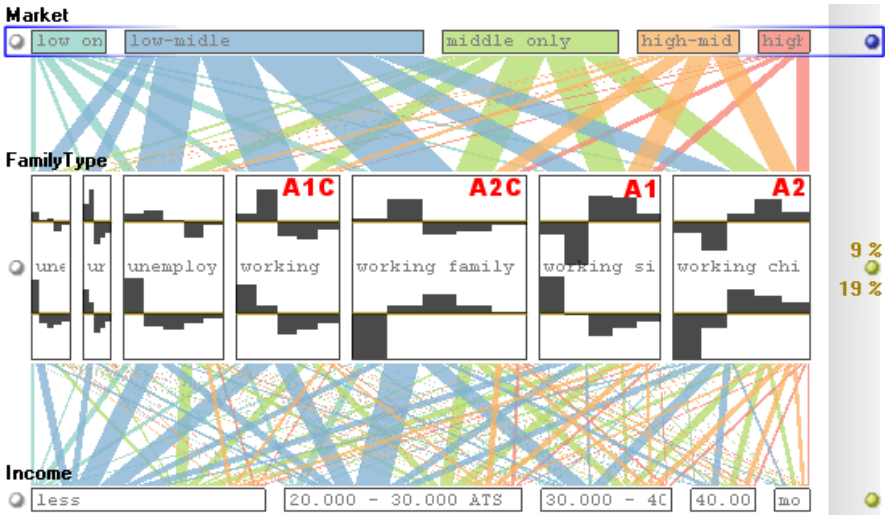


Figure 6: A parallel sets visualisation showing the relationship between the income of a family (Market), the employment of the (Family Type), and their income.

Unfortunately, similarly to parallel coordinates, a parallel sets visualisation is limited in the number of variables that can fit on screen. In fact, due to bands of observations potentially splitting up at every variable, as can be seen happening in Figure 6 between Family Type and Income, parallel sets is more readable when as few variables as possible are used. Too many variables will make the visualisation chaotic and hard to read. Furthermore, parallel sets only support categorical data, while we are also interested in visualising numerical data.

## 2.2 DIMENSIONALITY REDUCTION TECHNIQUES

Discussing the many different visualisation techniques in the previous section made it clear that all visualisation techniques are limited in the amount of data that can be visualised by them. All techniques are limited in the number of variables that can be shown at the same time and most techniques are also limited in the number of observations. When a visualisation has too many observations to show, they can relatively easily be subsampled or aggregated in order to reduce their numbers. Unfortunately, when dealing with too many variables, reducing the variables the same way is hard if not impossible.

One way to deal with the issue of too many variables is to reduce the number of variables that have to be examined. This is called dimensionality reduction and there are two main ways in which it can be achieved. The first way is called feature selection, and it reduces the number of variables by removing variables that are, for example, not interesting enough for the particular use case. The second way is called feature extraction, and it creates new variables based on the original variables and projects the observations onto those new variables. In general, the number of new variables is lower than the number of old variables, which makes it easier to explore the dataset.

The new variables are created such that certain metrics are maximised for the first few new variables. An example of such a metric is how well the distances on the first new variables match the distances between the points in the original dataset. Another metric is the amount of variance in the data described by the new variables. By maximising these metrics, most of the structure in a dataset can often be plotted on just two or three variables, making it possible to use many of the above-mentioned visualisation techniques, and even simple scatterplots, to visualise the data.

Although feature selection can be useful and there are many algorithms for it [10], the focus of this thesis is to make it easier to explore and interpret the results of feature extraction techniques. As such, any further mentions of dimensionality reduction in this thesis will refer to feature extraction. However, as we show in Section 5.2.4 and Section 5.2.5, overlap in feature selection and feature extraction is not uncommon as the results of feature extraction can be used to perform feature selection.

Several variants of feature extraction exist. In this thesis we focus on three of these method, Principal Component Analysis, Correspondence Analysis, and Multiple Correspondence Analysis, which are discussed in the following sections. These three techniques have in common that they are all linear. This means that the transformations they apply to a dataset to obtain new variables and projections on those variables are all linear in nature. Besides linear techniques there are also more complex techniques that are non-linear, such as t-Distributed Stochastic Neighbour Embedding [11] and Multi Dimensional Scaling. In the book Nonlinear Dimensionality Reduction [12], Lee et al. offer a general overview of variants of multi dimensional scaling and other non-linear techniques.

### 2.2.1 *Principal Component Analysis*

Principal Component Analysis, or PCA, is one of the most used and most famous techniques for feature extraction. The term "principal components" in this context was first coined by Hotelling [1] as early as 1933. PCA takes a data table and creates a new set of variables, called principal components or eigenvectors, to project the observations from this data table onto. The generated eigenvectors are orthogonal to each other and they are aligned in such a way that the first eigenvector is aligned with as much of the data-variance as possible.

Figure 7 shows an example of what PCA can do. The top plot in the figure shows the observations from a dataset with two variables plotted in a scatterplot. The bottom plot shows the same observations projected onto newly generated eigenvectors. We can see that the eigenvectors are aligned with the variance present in the data. Although this example uses a dataset with only two variables, PCA does not have a limit, other than a computational one, to the number of variables that can be analysed.

### 2.2.2 *Correspondence Analysis*

Correspondence Analysis, or CA, is a generalisation of PCA developed by Benzécri in 1973 [13]. Where PCA can be applied to any numeric data table, CA is designed to be applied to contingency tables. Contingency tables show the frequency distributions of two categorical variables. Table 1 shows a contingency table. In the table the frequencies of two categorical variables, farms and farm animals, are shown against each other.

Just like PCA, CA will generate a new set of eigenvectors aligned with the variance in the data. However, unlike PCA, CA projects the categories of both categorical variables, the rows and the columns of the table, onto the new

eigenvectors, instead of just the observations. As a result of using contingency tables, the results of CA focus on the difference between observations and columns in terms of their frequency distributions in the data table, whereas the results of PCA focus on the difference between observations with regards to their actual values in the data tables.

MULTIPLE CORRESPONDENCE ANALYSIS CA only supports the analysis of two categorical variables at the same time. To analyse more than two variables Multiple Correspondence Analysis [14], or MCA, can be used. MCA is an extension to CA that theoretically allows an unlimited number of categorical variables to be analysed. MCA works with categorical data tables, where every column is a categorical variable. MCA transforms the data table

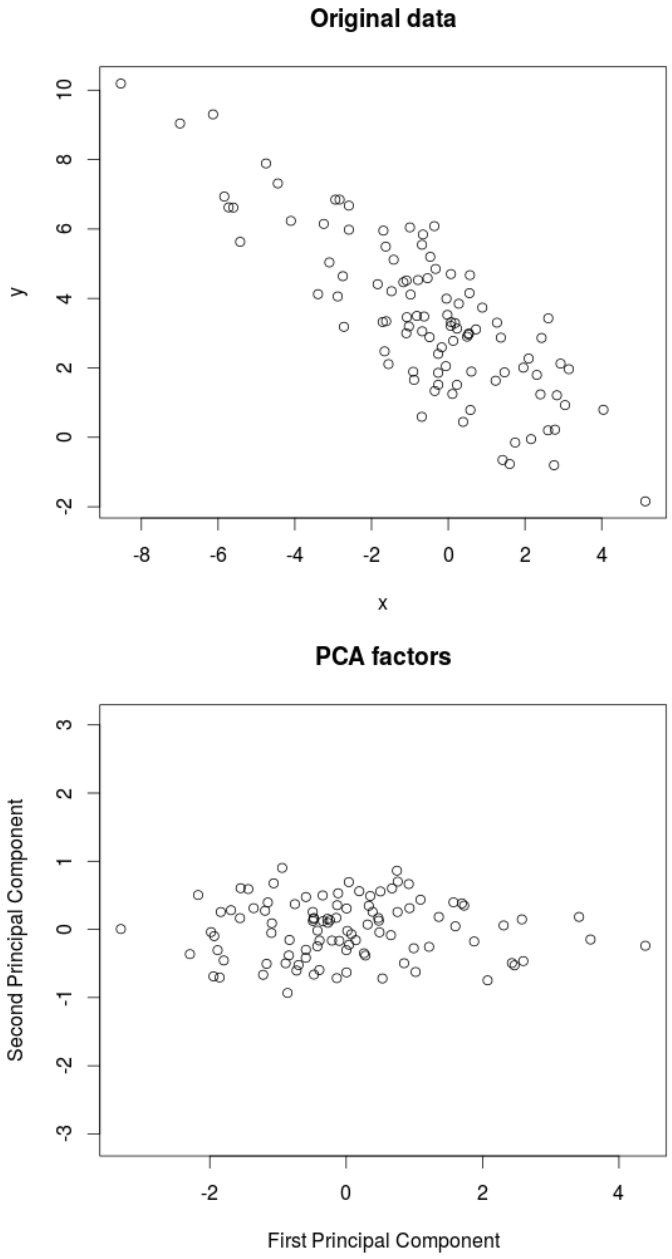


Figure 7: Example of PCA. Top: observations projected onto two variables. Bottom: same observations projected onto principal components generated by PCA.

| Farm | Cows | Horses | Pigs |
|---|---|---|---|
| Olterterp | 125 | 40 | 54 |
| Gorredijk | 10 | 5 | 7 |
| Nij beets | 0 | 30 | 10 |
| Wijnjewoude | 0 | 45 | 0 |

Table 1: A contingency table showing four farms with differing numbers of livestock.

by turning every category into a binary variable, after which the binary data table can be analysed using regular CA.

## 2.3 VISUAL ANALYTICS APPROACHES

When visualising the results of dimensionality reduction techniques, researchers often use simple scatterplots where the first two newly generated eigenvectors are plotted against each other. Work by Sedlmair et al. [15] and Brehmer et al. [16] has, however, indicated that researchers do not always get the expected results from these techniques, and they often have trouble understanding the projections that they are looking at. In order for researchers to get more out of dimensionality reduction techniques, there has been some work that focuses on visualising their results in more detail than mere scatterplots.

### 2.3.1 *iPCA*

iPCA [17] is an interactive tool designed to combine the visualisation of the original data with the visualisation of the results of PCA. As shown in Figure 8, it combines two parallel coordinates visualisations, one showing the original data and one showing the data projected onto eigenvectors, with a scatterplot of two eigenvectors and a correlation matrix. The user can change the eigenvectors used for the scatterplot to any of the seven most important eigenvectors. Unfortunately, iPCA only supports PCA, meaning that it can not analyse datasets consisting of both numerical and categorical variables.

The tool is designed to give users a better idea of how PCA works and how the results generated by PCA relate to their original data. It does this by allowing the user to change the data in one of the visualisations manually. When a user makes manual changes, the other visualisations are updated in real-time to reflect this change. An example of this would be the user dragging a point in the scatterplot, which would result in both parallel coordinates visualisations updating as well.

Although the interactive element present in iPCA can give the user a good idea of how PCA relates to their data, it is a rather indirect approach. If a user, for example, would like to find out which variables are most responsible for the structure in the scatterplot, the user could move a point in the scatterplot and see along which variables in the parallel coordinates this point changes the most. In contrast, using contribution bar plots, discussed in Section 3.1.3 and Section 4.2, a user can answer this question immediately without the need for extensive interaction.

### 2.3.2 *Dimstiller*

In contrast with the specific visualisation target of iPCA, dimstiller [18] is a much more general tool which allows the visualisation of both PCA and MDS.

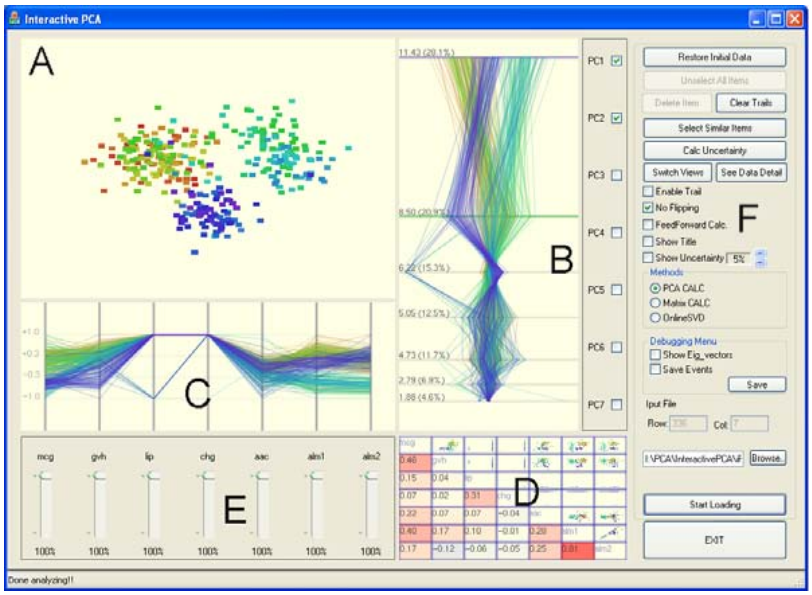


Figure 8: iPCA overview. (A) PCA projection view. (B) PCA eigenvectors as dimensions in a parallel coordinates visualisation. (C) Original data in parallel coordinates. (D) Correlation matrix of original data. (F) Controls.

The tool supports a range of different actions, which can be chained to get the eventual desired result. Some of these different actions are: culling variables with low variance, performing PCA or MDS, rendering a scatterplot matrix, and several more. The action chains, or workflows, that are created this way can be saved and easily replayed later on.

Although this tool allows for a lot of flexibility in how the dimensionality reduction is executed, the eventual visualisation of the PCA and MDS results consists of simple scatterplots. As such, the visualisation is not necessarily that much more helpful in understanding dimensionality reduction techniques and relating their results to the original data.

### 2.3.3 *Decision Exploration Lab*

Broeksema et al. [19, 20] present a visual analytics tool designed to visualise the results of MCA. Their solution is designed to be used by analysts and business users.

Although the developed visualisation is geared towards a specific user base, the techniques used in the visualisation of the MCA results are very interesting. First off, the focus of the visualisation is not just the projection of observations onto the eigenvectors generated by MCA, but the projection of variables onto these eigenvectors. By looking at the distance between different projected variables and how they are clustered, conclusions are drawn about relationships between the variables. Similarly to biplots [21], the proposed tool allows both variables and observations to be projected onto the same eigenvectors

Shown in the corners of the projections view in Figure 9, the tool also uses bar plots, partially based on the work by Oeltze et al. [22], to display which of the variables are important to the eigenvectors. These bar plots can be used to explain the meaning of the eigenvectors, and a similar technique has been used in this thesis as described in Section 4.2.1.

A downside to the tool is the fact that it only supports MCA, which means it can only analyse categorical data. Even though it is still possible to analyse

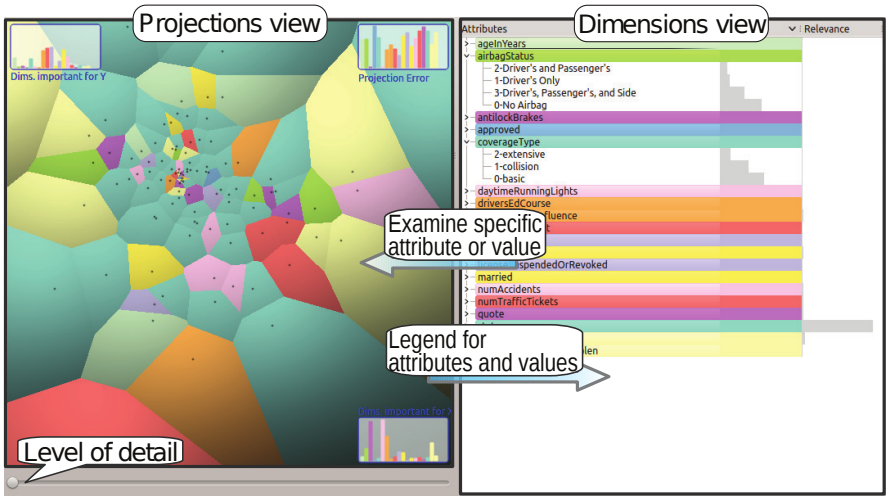


Figure 9: Decision Exploration Lab. The projected variables can be seen in the left view, while the variables are listed by name in the right view.

datasets that also contain numerical data by binning the numerical variables in multiple categorical bins, this approach effectually reduces the precision of numerical datasets which is often not desirable. It is also not clear how large the percentage of data-variance is that the eigenvectors used as scatterplot axes describe, which means that conclusions drawn from the visualisation may be less solid than one might suspect purely from the plot.

### 2.3.4 *Explaining three-dimensional dimensionality reduction plots*

Coimbra et al. [23] extend upon the work by Broeksema et al. with techniques to visualise the results of any dimensionality reduction technique using a 3D scatterplot. The original variables are also shown in this scatterplot, although not as points but as potentially non-linear axes. The bar plots shown by Broeksema et al. are used here as well to indicate how the original variable relate to the current x-axis and y-axis. Using these bar plots users can interactively rotate the 3D projection to align one of the axes with a variable displayed in the bar plot. Beyond these features, many other features such as colouring and a sphere which shows all the potential viewpoints that are available. An example of the proposed visualisation can be seen in Figure 10.

### 2.3.5 *Attribute-based Visual Explanation of Multidimensional Projections*

Recently, da Silva et al. [24] proposed a visualisation where the projected observations generated by dimensionality reduction are coloured based on which variable, or set of variables, best explains the placement of those observations. Figure 11 shows an example of this, where 6497 wine samples have been coloured based on 12 variables. The colouring that is added to the 2D projection makes it very easy and straight-forward to explain the placement of a group of variables.

Although the proposed system is more a visualisation technique than a full blown visual analytics tool, it is a very interesting approach to explaining projections, and it could be integrated into other tools.

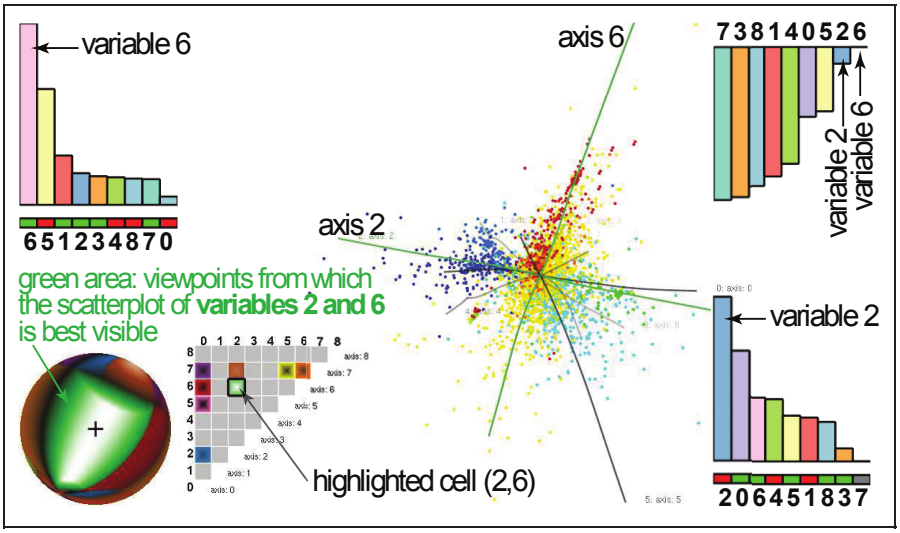


Figure 10: An example of the work presented by Coimbra et al. Some of the rendered axes are clearly curved because of the non-linearity of the dimensionality reduction method used.

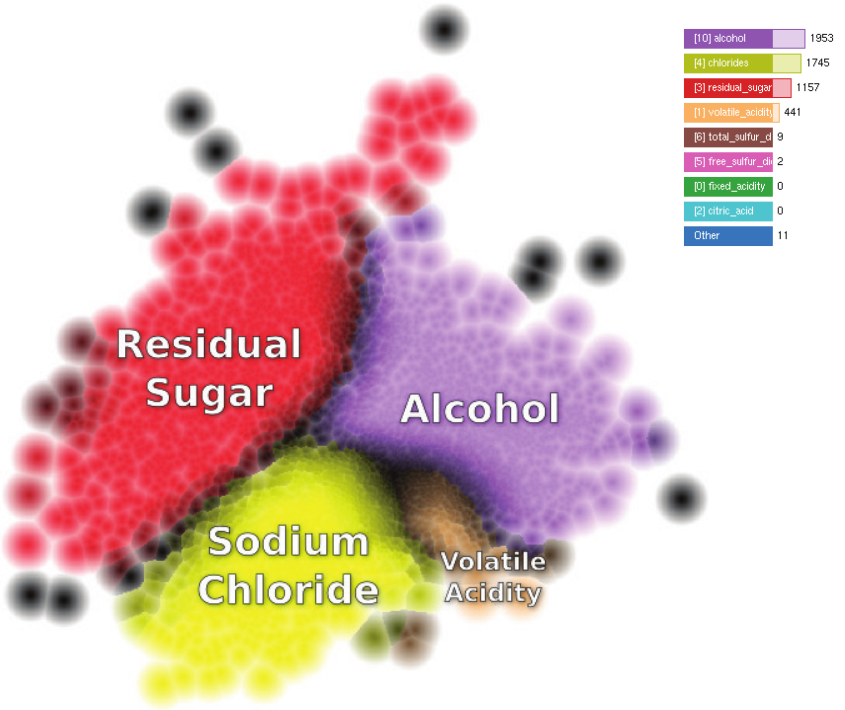


Figure 11: 6497 wine samples coloured by the variables that explain the placement of the samples best.

## 2.4 DISCUSSION

As we have seen, there exist many techniques aimed at visualising and analysing high-dimensional data. One big group of techniques tries to visualise an entire dataset at once. Unfortunately, these techniques are all severely limited in the number of variables they can display at once. A solution to this problem is to use dimensionality reduction to reduce the number of variables to just 2 or 3 variables. This is commonly done using dimensionality reduction methods such as PCA. However, here the problem arises that the

results produced by these methods are highly abstract, making interpretation of these result hard even for users familiar with the inner workings of these techniques. New tools are needed that can make the results of dimensionality reduction easier to interpret and explore. Chapter 4 and Chapter 5 present new tools to address these issues.

# 3 DIMENSIONALITY REDUCTION TECHNIQUES

In Section 2.2 we looked into several linear dimensionality reduction techniques. DimRedPlot, discussed in Chapter 4, is designed to visualise the analyses produced by these techniques. To support this visualisation several features of PCA, CA, and MCA have been used in DimRedPlot, such as loadings and contributions. Because understanding these features is crucial in understanding how DimRedPlot works, this chapter looks at how PCA, CA, and MCA work and what results they produce.

## 3.1 PCA

Before PCA is applied to a data-table, the columns of the data-table, the variables, are centred in such a way that their means are equal to 0. Furthermore, the columns are often standardised, e.g., the values in the columns are divided by the standard deviation of the column. This is only needed when the different columns are not in the same space, such as `mg` or `pH`, and can not be compared in a sensible way.

As explained in Section 2.2, PCA produces so called principal components or eigenvectors upon which the rows of the data-table are projected. The values the rows obtain for every principal component are called factor scores. To obtain these factor scores we need to solve the singular value decomposition, or SVD, as described in Equation 1, where $X$ is the data-table.

$$X = P\Delta Q^T \tag{1}$$

For an explanation of how the SVD can be solved, the paper on SVD by Abdi et al. [25] can be used. After solving the SVD, we can obtain the factor scores as shown in Equation 2, where $F$ is the matrix containing the factor scores, with every column being a principal component.

$$F = P\Delta \tag{2}$$

The factor scores could now be plotted by taking two columns from $F$ and using these as $x, y$ coordinates. However, there is more information that can be obtained from PCA than just the factor scores. The following sections will describe some of the different results that PCA can give. For a more thorough and deeper explanation of how PCA works, the paper on PCA by Abdi et al. [26] is a good source.

### 3.1.1 *Principal components*

As explained in the previous section, after solving the SVD, we are left with a matrix $F$, where every column is a principal component. The principal components are obtained in such a way that the first principal component explains as much of the variability or variance in the data as possible. Every next principal component explains as much of the variance in the data with the constraint that it is orthogonal to the previous principal components. This means that just like our original variables, all the principal components are orthogonal to each other. In fact, the principal components are linear combinations of our original variables, and they can be seen as a rotation of the original variables.

Principal components are also called eigenvectors. The reason for this is that the principal components are the eigenvectors of the eigendecomposition

of $X^T X$, where $X$ is again the original data-table. The data-variance that every principal component or eigenvector explains is called their inertia and it is equal to their corresponding eigenvalue. We can take the sum of all these inertias and divide each inertia by this sum to obtain percentages. Using these percentages we can say for every eigenvector how much of the data-variance it describes in percentages. This is useful, as we can use these percentages to tell a user how much of the data-variance is actually being looked at when showing a scatterplot with factor scores.

Unfortunately, only PCA uses the terms principal components when referring to the axes it generates. To avoid any confusion in the rest of this thesis, we will be talking about eigenvectors instead of principal components, as this means the same in PCA, CA, and MCA.

### 3.1.2 *Loadings*

After performing PCA on a data-table, we are left with a set of eigenvectors and factor scores. However, what we do not know at this point is what these eigenvectors mean in terms of our original variables. Loadings can help us understand these relations.

A loading, in the context of PCA, is the correlation between an eigenvector and a variable. This correlation tells us something about the amount of information that is shared between an eigenvector and a variable. The loadings have values lying in the $[-1, 1]$ interval. Since every variable has a loading to every eigenvector, it is possible to plot every variable based on their loadings to the eigenvectors used as plot axes. An example of such a plot can be seen in Figure 12, which shows 7 variables plotted onto two eigenvectors.

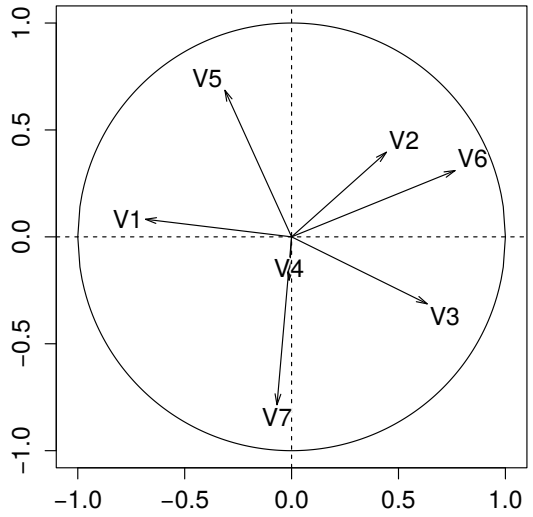


Figure 12: Loadings of variables projected onto eigenvectors generated by PCA.

The loading plot can be interpreted as follows: the closer a loading is to one, or minus one, the more information the variable and eigenvector of that loading share. To take the plot as an example, variable V7's arrow is strongly aligned with the eigenvector of the vertical axis and V7's distance to the centre is almost 1, which means that V7 has a strong negative correlation with the vertical axis' eigenvector and thus shares a lot of information with the eigenvector. Variable V2 is in between the two eigenvectors in orientation and closer to the centre of the plot. This means that the correlation of V2 to both eigenvectors is similar. The short distance of V2 to the centre, means that V2 is also sharing information with eigenvectors beyond the first two.

This information can be very useful to find out what the meaning is of an eigenvector in terms of the variables in the original data, and through this, which variables are important for the structure in the observations as projected onto the eigenvectors.

Looking at the plot, we can see that every plotted variable lies within a unit circle around $(0, 0)$. The reason for this is that the loadings are normalised in such a way that the sum of the squared loadings for a variable are equal to one. When the sum of the squared elements of a vector is equal to one, the length of that vector must be one as well. As such when looking at the loadings of a variable as a vector, the end of that vector must lie on the edge of a $L$-dimensional sphere, where $L$ is the number of eigenvectors.

We only discuss loadings in the context of PCA. This is mostly because CA and MCA have alternative ways to project variables onto generated eigenvectors. However, loadings can be calculated for other dimensionality reduction techniques as well, even more complex non-linear techniques. A generalisation of loadings that will work independently of dimensionality reduction technique is discussed by Coimbra et al. [23].

### 3.1.3 *Contributions*

The way the eigenvectors are calculated depends on how the original data is shaped. This means that for a certain eigenvector there are some observations which might be very important for the calculation of the eigenvector, while there are others that might not be important at all. The importance an observation has for the calculation of a certain eigenvector is called the contribution, as it is essentially the contribution of an observation to a eigenvector.

To calculate the contributions of an observation to the eigenvectors we need the factor scores of that observation and the eigenvalues or inertias of the eigenvectors. We will call these factor scores $f_{i,l}$, with $l$ being the eigenvector and $i$ being the observation, and we will call the eigenvalues $\lambda_l$. The contributions of observation $i$ are then equal to:

$$contribution_{l,i} = \frac{f_{i,l}^2}{\lambda_i} \tag{3}$$

The inertia, or eigenvalue, of an eigenvalue can also be calculated through the following formula:

$$\lambda_l = \sum_i f_{i,l}^2 \tag{4}$$

Because of this, the contribution always lies in the $\langle 0, 1]$ interval. Furthermore, the sum of all contributions to an eigenvector is always equal to 1. Thanks to these properties we can translate contribution to a percentage. Using this percentage we can then make statements such as: a certain observation is 45% responsible for a certain eigenvector.

When looking at observations, contributions may not be an inherently useful statistic. Observations with a high contribution are those with the most extreme values, and we can easily spot them by merely looking at a scatterplot of the observations. However, since we can project variables onto the eigenvectors using loadings, we can also calculate a contribution for them by using their loadings as $f_{i,l}$. When doing this, contribution tells us approximately the same as the squared loadings. The higher the contribution of a variable to an eigenvector, the more information is shared between the eigenvector and the variable. Since, we already have the loading statistic one might think that contributions are not that interesting. However, CA and MCA do not have loadings and as such contribution is a useful statistic there.

## 3.2 CA

CA, or correspondence analysis, is a generalised form of PCA. As mentioned before, CA is designed to be operated on contingency tables, although it has been used for many other purposes since its inception.

The benefit of using CA on contingency tables, instead of simply using PCA, can be shown with an example. Table 1 shows a count of cows, horses, and pigs for four farms. If PCA is applied to this table, the result would be that Olterterp would be an outlier from the rest. The reason for this is that Olterterp has a larger number of livestock than the rest of the farms. However, if we were to apply CA to this table, Olterterp and Gorredijk would actually be relatively close together. This is because even though the number of livestock differ quite significantly between the first two farms, the ratio of livestock on both farms is quite similar.

As mentioned, CA will transform a contingency table and project its observations onto a set of new axes. Unlike PCA, these axes are called factors or eigenvectors, instead of principal components. Furthermore, CA can not only project the rows of a contingency table onto new axes, it can also project the columns onto these axes. This has to do with the fact that the variables in a contingency table all have the same type and domain. Because of this, if we transpose the contingency table, it is still a sensible table, and we can project our new rows, which were previously columns, onto new axes as well. Since we have only transposed our contingency table, the resulting eigenvalues and eigenvectors do not change, which means that our new axes are the same for both the original contingency table and the transposed contingency table.

Similarly to PCA, CA can also be performed by solving an SVD, only in this case it is a generalised singular value decomposition [25]. In the next sections a short explanation of the GSVD will be given. GSVD makes use of properties of the columns and the rows called mass, which will be talked about first. After this the actual explanation of the GSVD is presented.

### 3.2.1 *Mass*

When performing CA, every column and row in the original data-table has a mass. This mass indicates the proportion of a row or column in the total table.

In order to find the masses of the rows of data-table $X$, we first need to know the sum of all elements in $X$, which we shall call $s$. For Table 1, $s = 326$. Second, we need the sums of the elements in each row. In the case of Table 1, the matrix of row sums, $S$, is as follows:

$$S = [219, 22, 40, 45]^T$$

Using the row sums, sum of $X$, and Equation 5, we can find the matrix of row masses, $R$:

$$R = \frac{1}{s} S \tag{5}$$

Applying this formula to Table 1 results in the following row masses:

$$R = [0.672, 0.067, 0.123, 0.138]^T$$

As we can see from this matrix, the masses add up to 1. This means that the masses essentially tell us what fraction of the total livestock count each farm has.

| Fruit | Colour | Colour:red | Colour:yellow | Colour:orange |
|---|---|---|---|---|
| apple | red | 1 | 0 | 0 |
| banana | yellow | 0 | 1 | 0 |
| tomato | red | 1 | 0 | 0 |
| orange | orange | 0 | 0 | 1 |

Table 2: The colour of different fruits, using both a categorical variable and binary variables.

The masses of the columns are calculated the same, except that $\frac{1}{s}$ is multiplied with the sums of the elements in each column. The resulting column masses look like this:

$$C = [0.414, 0.368, 0.218]^T$$

### 3.2.2 *Solving the GSVD*

Before we solve the GSVD, we first normalise and centre the data-table. The normalisation is done by dividing each row by its sum. The centring is done by subtracting the average of the rows from every row.

After obtaining the normalised data-table we can solve the GSVD. What makes the GSVD different is that there are some extra constraints regarding the masses of the rows and columns, as can be seen in Equation 6.

$$X = P\Delta Q^T \quad | \quad P^T R P = Q^T C^{-1} Q = I \tag{6}$$

After solving the GSVD we can retrieve the factor scores through Equation 2. The factor scores for the variables can be calculated using the exact same progress, except for the fact that our data-table first needs to be transposed. For a more thorough explanation of CA, the paper on CA by Abdi et al. [27] is a good source.

## 3.3 MCA

MCA works by taking a data-table with categorical variables and converting it into a data-table with binary variables. This conversion is done by creating a new binary variable for every category of the original categorical variables. The new binary variables will be 1 for every row that has that specific category, and 0 for the other rows. An example of this can be seen in Table 2, which contains both the original variable, `Colour`, and the new binary variables, `Colour:red`, `Colour:yellow`, and `Colour:orange`.

After the original categorical data-table has been converted to a binary data-table, regular CA can be applied to the binary data-table. Because of this, we get the same results when we apply MCA as when we apply CA. The only difference is that the mass of every observation is the same. We can see this in Table 2. The total sum of every observation is equal to one, because every observation can only have one colour at the same time. This will be true for every categorical variable, which means that the total sum of every observation is the number of categorical variables. Since observation mass is directly correlated to the observation sums, the mass will be the same. This makes observation mass in MCA quite meaningless.

## 3.4 DISCUSSION

In the introduction we discussed that many researchers that use linear dimensionality reduction techniques only interpret them by looking at the

resulting projected observations in a scatterplot. However, as we have shown in this chapter, there is a lot more information, such as loadings, variance, and contributions, that these methods can provide. When used correctly, this information can be used to make the results of linear techniques much easier to interpret.

# 4

# DIMREDPLOT

DimRedPlot is a visual analytics tool designed to visualise the results of three particular dimensionality reduction techniques, PCA, CA, and MCA. DimRedPlot can be used as a stand-alone tool, but it is designed to be used in combination with other visualisation techniques such as parallel coordinates, as is done in Chapter 5. An example of DimRedPlot can be seen in Figure 13. Looking at the figure, we can see that the visualisation consists of several parts, highlighted in the image with red rectangles. The function of each highlighted part of the image is as follows:

- EIGEN-BAR This part of the image consists of a long bar with blue rectangles in it. The bar represents the eigenvectors generated by the used dimensionality reduction technique, as described in Section 3.1.1. The eigen-bar can be used by a user to change the eigenvectors used as scatterplot axes in the scatterplots below. Beyond this, the bar also gives the user a quick intuition of how much of the original data is being looked at in the scatterplots and thus how strong any conclusions drawn here are.

- OBSERVATION SCATTERPLOT This scatterplot displays the observations of the used dataset projected onto eigenvectors using their factor scores. The observation scatterplot will be what most users of dimensionality reduction techniques are used to. Using the observation scatterplot, users can find out what structure is present in their data and what the general shape of their data looks like.

- VARIABLE SCATTERPLOT This scatterplot displays the variables onto the eigenvectors using either the loadings of the variables, as described in Section 3.1.2, or using the factor scores of the variables. The structure of the variables in the data can be explored using this scatterplot. The proximity between variables tells us about the similarity between variables, which means that the plot can be used to quickly find both variables that are very dissimilar from the rest and groups of variables that are very similar to each other.

- CONTRIBUTION BAR PLOTS The bar plots in this part of the image depict different values for the variables shown in the right-most scatterplot. The four bar plots show the contributions that the variables have to the generated eigenvectors, as explained in Section 3.1.3. The bar plots allow a user to find out in a direct manner what variables are responsible for the eigenvectors used as scatterplot axes. This means that they tell the user which variables are responsible for the structure seen in the observation scatterplot.

  When a selection of observations is made, as described in Section 4.3.4, a fifth barplot appears in this part of the image, which shows the amount in which a variable discriminates a selection of observations from the rest of the observations. An example of this fifth barplot is shown later on in Figure 23.

In the following sections, we look at the implementation details of DimRedPlot and we describe how the individual parts of the visualisation work in more detail. We also have a look at the interactions that are possible between the individual elements.

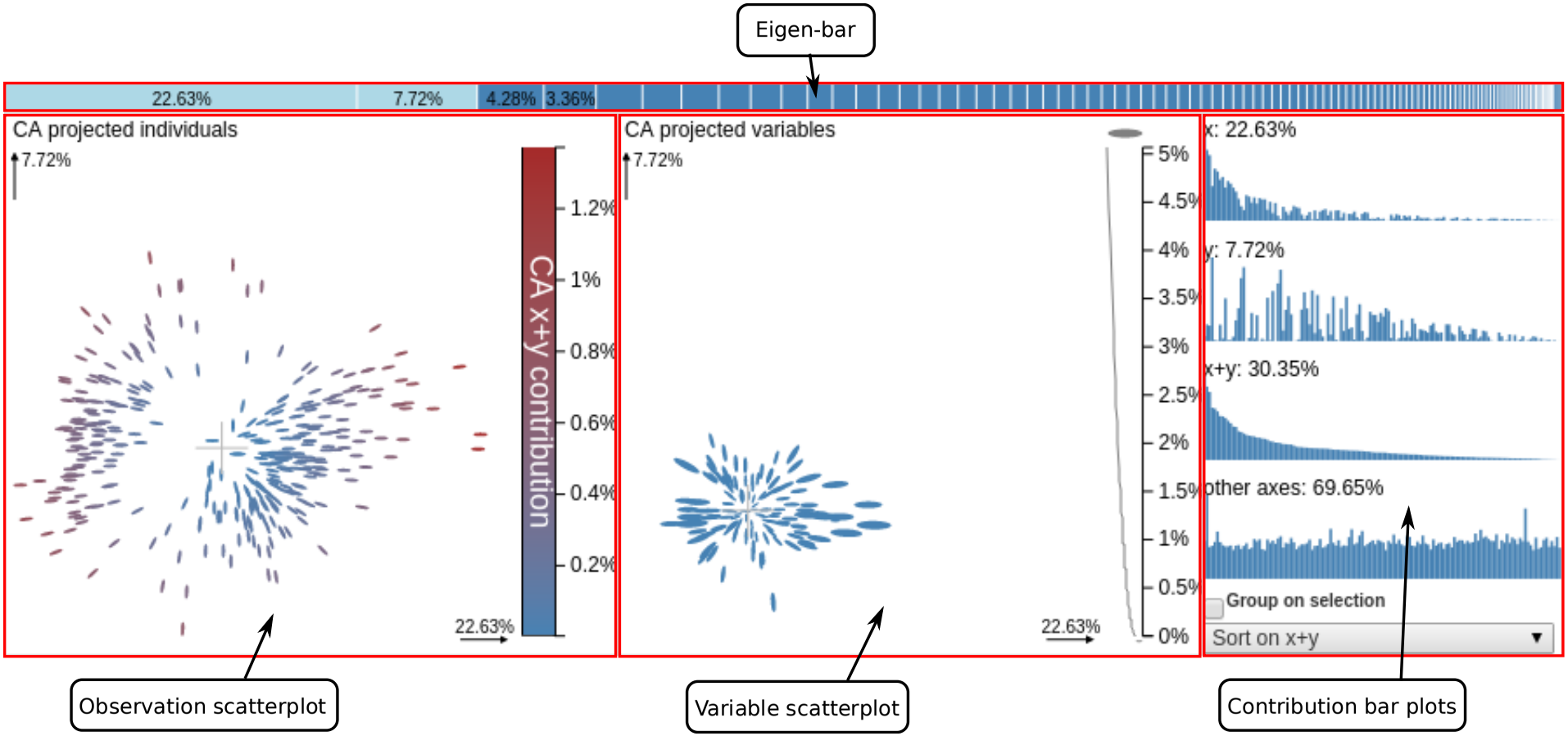


Figure 13: DimRedPlot visualising results from CA. Eigen-bar: Each rectangle in the bar represents a generated eigenvector and the data-variance it describes. Observation scatterplot: The observations projected onto 2 eigenvectors. Variable scatterplot: The variables projected onto the same 2 eigenvectors. Contribution bar plots: Several bar plots detailing how important each variable is to the used eigenvectors.

## 4.1 EIGEN-BAR

As described, the thin long bar on top of the visualisation in Figure 13 represents the eigenvectors generated by the used dimensionality reduction technique. This bar can be used to select the two eigenvectors that are consequently used as scatterplot axes. The bar tells the user how much data-variance each eigenvector describes, allowing users to select their eigenvectors such that a desired amount of data-variance is shown in the scatterplot.

Every eigenvector generated by PCA, MCA, or CA describes a certain percentage of the variance in the original data. The total variance of all the eigenvectors adds up to 100%. Every rectangle in the eigen-bar represents one of the eigenvectors generated by the used dimensionality reduction technique. The total length of the eigen-bar represents 100% of the variance, while the length of the blue rectangles represents the variance percentage of each eigenvector. For example, if the length of one of the rectangles is half the total length of the bar, the eigenvector represented by that rectangle describes 50% of the total variance in the data. To make the variance described by every eigenvector extra clear, the specific percentages are also displayed within the rectangles. The percentages can also be obtained by hovering the mouse cursor over the rectangles and reading them from the resulting tooltip.

At any time, only two eigenvectors are used as axes for the scatterplots. To distinguish the rectangles that represent these eigenvectors, they are coloured a lighter blue than the rest of the rectangles.

Often, when the results of PCA are visualised using a scatterplot, only the first two eigenvectors are used as scatterplot axes. Instead, we chose to give the user access to all generated eigenvectors. This is helpful when the first two eigenvectors only describe a small amount of data-variance, or when the first few eigenvectors describe a very similar amount of data-variance. In both cases structure in the data may be found on eigenvectors beyond the first two, making it worth it to explore other eigenvectors.

Even though it can be interesting to look at eigenvectors beyond the first two, there are generally also a large number of them that are not interesting. This is due to them describing only a low percentage of the data-variance. Nonetheless, these eigenvectors are still shown. Their presence in the bar helps the intuition of how much data-variance the first ones describe compared to the total.

### 4.1.1 *Alternative visualisation*

In another potential design the eigenvectors are shown using a vertical bar plot. How this looks can be seen in Figure 14, which shows both the eigen-bar as Part A and the vertical bar plot annotated as Part B. Unfortunately, a bar plot does not give an immediate intuitive feeling of how much of the total variance is described by one or two bars. As such, the eigen-bar is used instead of the vertical bar plot design. An additional benefit is reduced screen-space usage due to removal of unused white space. Unless all bars in a bar plot are the same length, it can waste a considerable amount of space.

In the design of the vertical bar plot, it is also possible for users to filter out a set of eigenvectors. This is helpful as a large number of eigenvectors can result in bars with a very low height, making it hard to distinguish the bars. It also makes showing percentages on the bar as text impossible. However, the current visualisation does not have these problems and as such the reasons for filtering do not apply anymore.

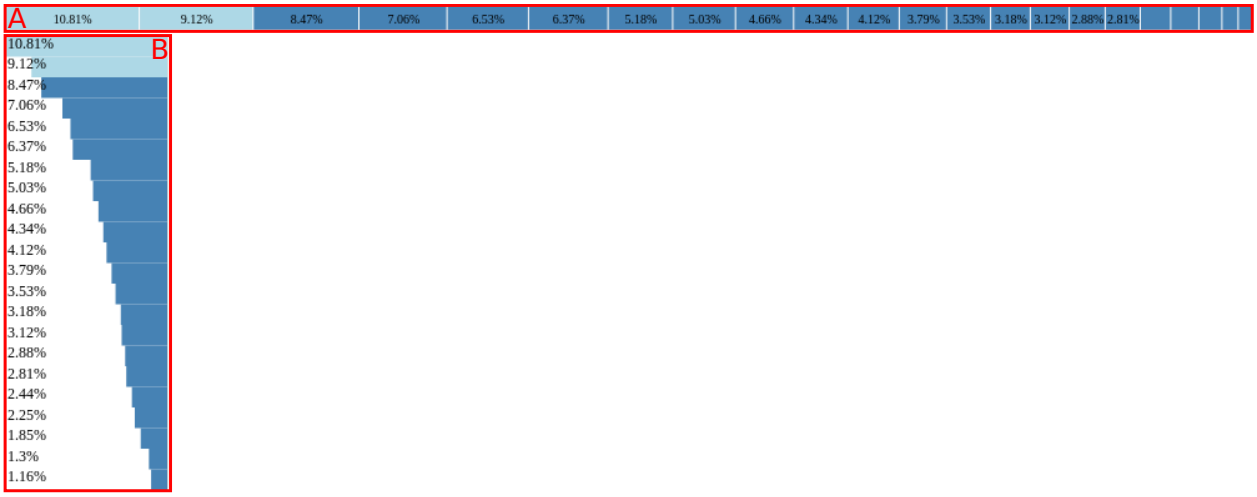


Figure 14: A comparison of different visualisations of described data-variance by eigenvectors. (A) The eigen-bar. (B) The alternative vertical bar plot.

### 4.1.2 *User interaction*

To change the eigenvectors that are currently used as axes in the scatterplots there are several interactions in place. The first and most obvious interaction is the possibility to simply select an eigenvector not used as axis using the mouse cursor. On doing this the selected eigenvector is used as axis in the scatterplots. To determine which of the axis to change, the axis is picked that was changed least recently.

The second interaction is the possibility to select an eigenvector that is used as axis and drag it over the bar. Whenever the user stops dragging, the dragged axis is changed to the eigenvector the user currently hovers over with the mouse cursor.

Finally, it is also possible to simply hover the mouse cursor over the eigen-bar and scroll the mouse wheel. This results in both axes changing their eigenvectors. When scrolling, the eigenvectors used by both axis will shift one to the right or the left depending on the scroll direction.

Whenever the user changes the eigenvectors used as scatterplot axis, the observations in the scatterplots move smoothly from their old location to their new location. This is very useful, as it allows users to see how structure in the scatterplot changes between different eigenvectors. This functionality is, in fact, very similar to the work by Elmqvist et al. [28], albeit without the explicitly rotating cube. An example of this can be seen in Figure 15. Scatterplot A shows the structure before changing the axes' eigenvectors. In the scatterplot, three distinct clusters can be seen. When changing the eigenvectors the projected observations move, and scatterplot B shows the state of the scatterplot halfway through this movement. Already we can see that a new cluster appears on the right and moves upward. The final situation can be seen in scatterplot C. We can now see four clear clusters, with the new cluster encircled with a red ellipse. The smooth animation makes it immediately clear which cluster moved where and that one cluster splits up into two different ones.

In general, we can say that the eigen-bar is designed to be used as follows. When a user starts an instance of DimRedPlot he or she may be faced with the following two situations:

1. The first two eigenvectors describe a, for that user, significant amount of data-variance, with both eigenvectors describing significantly more

data-variance than the third one. If interesting structure, such as clusters, occurs on the first two eigenvectors, the user can select parts of the structure using the scatterplots and use any of the described interactions to change the used eigenvectors. This way it is possible to find out whether that same structure can be seen on other eigenvectors. If so, the user can draw more sound conclusions about the structure than if it occurs only on two low variance eigenvectors.

2. The first two eigenvectors are very similar in the data-variance they describe as the next couple of eigenvectors. In this case, if the user does not see any structure in the data interesting to that user's use case, it is possible to change the eigenvectors to explore combinations of the first $N$ similar eigenvectors in search of interesting structure.

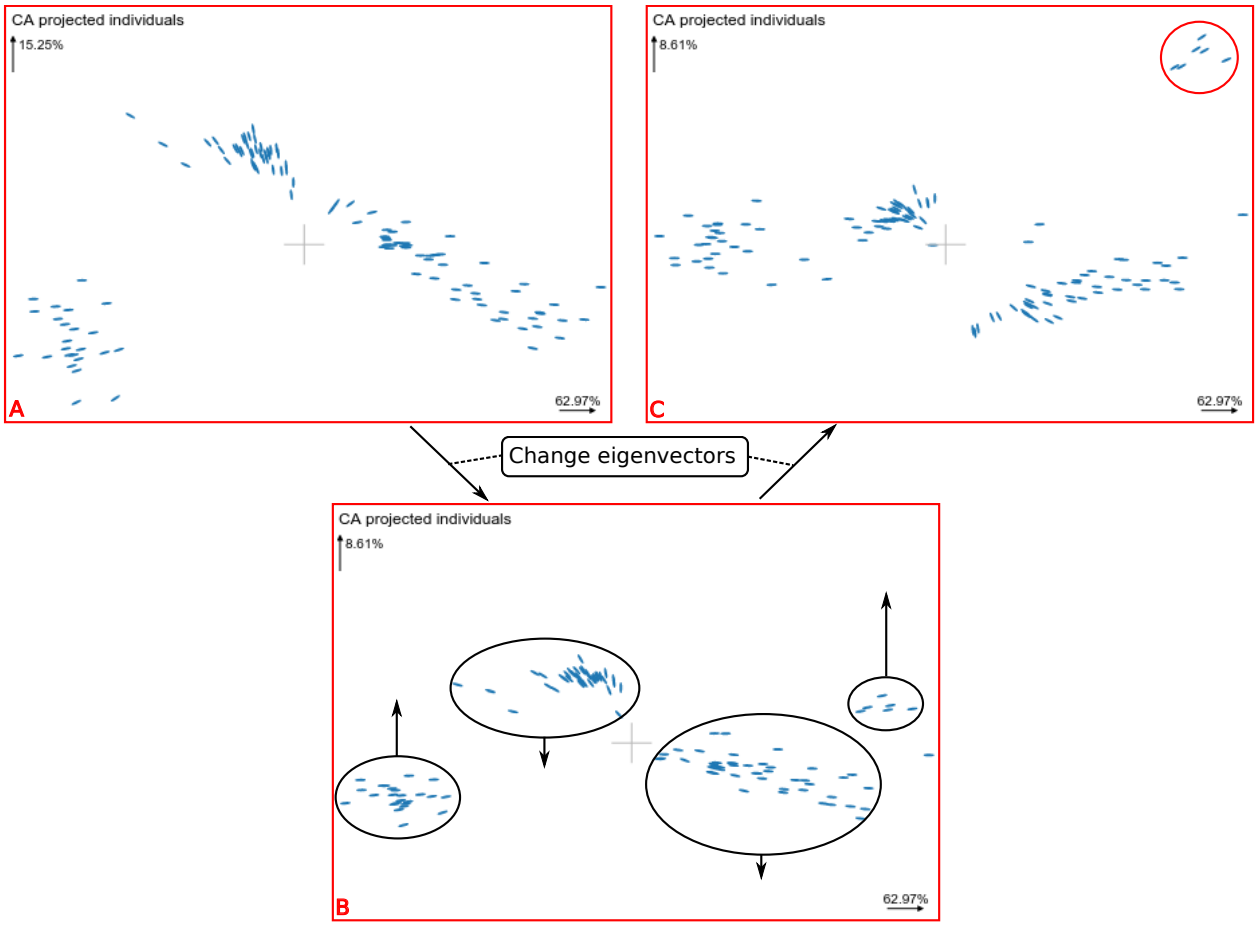


Figure 15: A scatterplot transition showing the change of the y-axis' eigenvector. A: The scatterplot before the transition. Three main clusters can be seen. B: Halfway through the change of the y-axis' eigenvector. The arrows indicate in which direction the groups of observations are moving. The right-most group we see moving was not its own group in part A. C: The scatterplot after the transition. We can now see four main clusters. Notice how the red encircled cluster was not separate in part A.

## 4.2 VARIABLE BAR PLOTS

The bar plots on the right of DimRedPlot, as shown in Figure 13, depict either contribution per variable or the discrimination of a selection that a variable can supply. All the bar plots work the same and only the data they visualise is different. Figure 16 shows the five bar plots that can be shown annotated with red rectangles. Bar plots A to D show contribution per variable, and bar plot F, which is optional, shows discrimination per variable. In the next sections we discuss each bar plot in more detail.

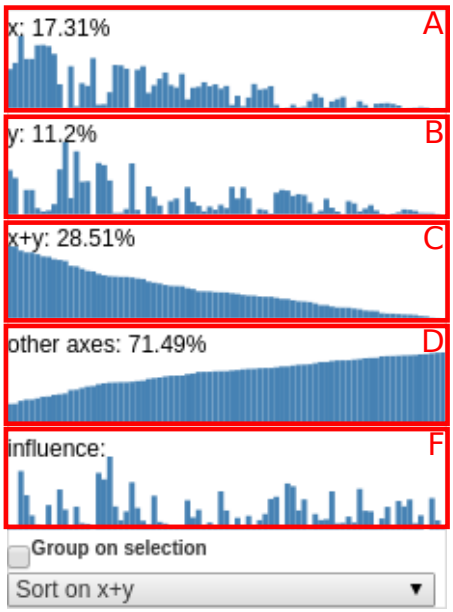


Figure 16: Bar plots depicting several metrics for each variable. (A-D) The contribution each variable has to different sets of the generated eigenvectors. (F) The amount in which each variable can discriminate between selected and non-selected observations.

### 4.2.1 *Contribution bar plots*

As mentioned in Section 3.1.3, contribution is a property that every observation and variable has for every eigenvector. Contribution is expressed in percentages and tells how much a variable or observation influenced the forming of a specific eigenvector. Contribution is not necessarily an interesting metric for the observations; however, the contributions of the variables are very interesting. When variables have a high contribution to a certain eigenvector, it essentially means that we can explain the meaning of that eigenvector using those variables. If that eigenvector is used as scatterplot axis, we can use the variables that describe its meaning to explain the structure we see in the observation scatterplot. For example, if there is interesting structure in the data along a certain eigenvector, such as a set of clusters, the variables with strong contributions to that eigenvector are good at distinguishing the different clusters from each other.

In total there are four contribution bar plots present in DimRedPlot, each serving a different purpose, and they are discussed in the next paragraphs.

SINGLE EIGENVECTOR CONTRIBUTIONS The two top-most bar plots, bar plot A and bar plot B in Figure 16, show the contributions of the variables to the eigenvectors that are currently used as scatterplot axes. The user has the option to sort any of the bar plots. When sorting, the visualisation

makes sure that the ordering of variables is the same for all bar plots. This is useful to compare the contributions of the same variable. An example of sorting can be seen in Figure 16 where the x+y contribution bar plot is sorted, which is the default behaviour. Sorting one of the top two bar plots, makes it very easy for a user to see which variables are important to those eigenvectors. Unfortunately, the bars in the bar plots are too small to contain the variable names. To solve this the mouse cursor can be hovered over the bars to retrieve the contribution percentage and the name of the variable. Section 7.1.3 discusses a possible solution to this problem.

To help the user visually link the axes of the variable scatterplot to these two bar plots, a label above each bar plot indicates the axis represented by the bar plot. Next to the axis name, the percentage of data-variance along the axis is displayed. The other two contribution bar plots, discussed in the next two paragraphs, also have a label detailing the axes the bar plots represent and the total data-variance those axes' eigenvectors describe.

COMBINED EIGENVECTOR CONTRIBUTIONS The first two bar plots are useful to find out which variables are important for the structure along the eigenvectors used as axes. However, to find out which variables are important to the structure along both eigenvectors, a user would have to look for bars that are high in both bar plots, which is very time consuming or even near to impossible when the number of variables is large. To solve this issue, the third bar plot from above, bar plot C in Figure 16, displays the contributions of the variables to both used eigenvectors. To calculate this cumulative contribution we have to be careful not to just add the contribution percentages to the two eigenvectors together. This is because the two shown eigenvectors may not describe the same data-variance. An extreme example of this case can be seen in Figure 17. A variable low on the y-axis may contribute just as much to that axis as another variable to the x-axis, but that does not mean they are just as important. To add the contributions together we can first scale them by the data-variance described by their eigenvectors. After doing this, we have to make sure that the contributions are scaled again to add up to 100%.

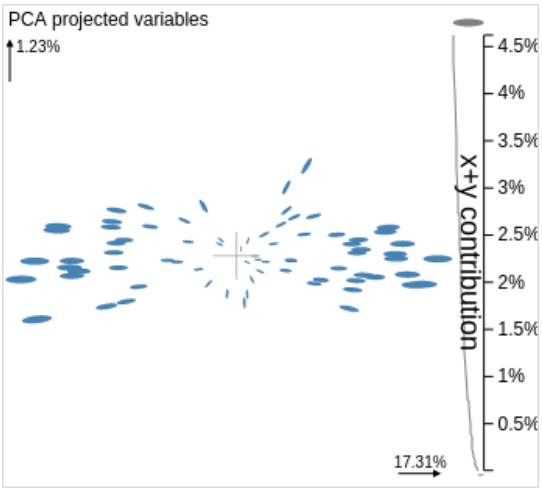


Figure 17: Scatterplot using both an eigenvector describing high data-variance and one describing low data-variance as axes.

OTHER EIGENVECTOR CONTRIBUTIONS The fourth bar plot from the top, bar plot D in Figure 16, shows the contribution that each variable has to the eigenvectors not currently shown. This can be interesting, as a variable that has a low value in this bar plot has most of its variance described by the eigenvectors currently used as scatterplot axis. The variance of variables that

have a high value in the bar plot must instead be found on other eigenvectors. It must however be noted that when using PCA, the contributions of each variable to all eigenvectors add up to 100%, meaning that this contribution bar plot is simply the inverse of the combined contribution bar plot.

### 4.2.2 *Discrimination bar plot*

The fifth bar plot, bar plot F in Figure 16, does not necessarily show anything related to contribution. As mentioned, it shows a metric depicting the ability of the original variables to distinguish a selection of observations. This means that if a user were to make a selection of observations through whatever means, this metric shows which variables are good at discriminating those observations from the observations that are not selected. This can be useful to find out what the meanings of certain structures in the data are, e.g., which variables cause them to be separated from the rest.

The metric used to determine this discrimination differs slightly per dimensionality reduction method. For PCA and CA, the metric uses the normalised data on which the techniques are performed and calculates for every variable the average value and the standard deviation of both the selected and the non-selected observations. As explored by Turkay et al. [29], when the average and standard deviation of both the selected and non-selected observations over the same variable are similar, it means that that variable would be bad at discriminating between the two sets of observations. If instead the averages and standard deviations are different, that variable is good at separating the two sets of observations. The metric calculates the difference in averages and standard deviations and adds these values together. The result is that each variable will be assigned a value indicating the difference in average and standard deviation, which is then used to create the discrimination bar plot.

When working with categorical data and MCA, it is impossible to calculate averages or standard deviations. Instead the metric looks at how well the different categories of a variable separate the selected observations from the non-selected observations. In the case that, for example, non of the categories are assigned both to selected and non-selected observations, the variable is perfect at discriminating between the two sets of observations. If categories are assigned to observations in both sets, the variable is bad at discriminating between the observations.

Using one of the methods described above, every variable is assigned a value determining its quality at discriminating between the selected and non-selected observations. These values are then shown as a bar plot below the contribution bar plots. After this, the bar plot can be used to determine which variables are responsible for certain structures in the data, such as clusters. To do this, a user has to select a set of observations in the observation scatterplot, and the bar plot will be updated based on the used discrimination metric.

It is also possible to use other discrimination metrics instead of the ones described here, although currently only the described metrics have been implemented into DimRedPlot. For some other metrics that could be used here, the paper by Rauber et al. [30] is a good source, in which various discrimination metrics are used.

### 4.2.3 *User interaction*

All of the bar plots offer the possibility to select variables. The user can use the mouse cursor to click and drag over the bar plot, which results in the selection of the variables represented by the dragged over bars. A different selection can be made on each bar plot, and the variables selected this way are the union of all the bar plot selections. When a selection is made on a

bar plot, the percentages of the selected bars are added up and displayed after the data-variance percentage in the label. This percentage is the data-variance described by the selected variables over the axes represented by the respective bar plot.

In Figure 18 a selection has been made in the $x + y$ contribution bar plot, which can be seen by the black outline around the selected bars.


Figure 18: Comparison of bar plots without selection (left image) and with selection (right image).

As mentioned, the user can choose to sort any of the bar plots through a drop-down menu. In combination with selecting variables, sorting makes it easy to select the set of $N$ variables with the highest values in the sorted bar plots. When doing this for the top-three bar plots, it means selecting the variables responsible for the structure over the x-axis, the structure over the y-axis, or the structure over both axes.

Both the contribution and discrimination metrics can help to find the variables that are responsible for the structure seen in the observation scatterplot. Because of this, selecting these variables can be useful to simplify or focus further analyses when DimRedPlot is used in combination with other tools. For example, the selection of variables could be used to refine the selection of variables on which dimensionality reduction is applied, to highlight the variables in another visualisation, or to reduce the number of variables shown or used in another visualisation. Some of these possibilities have been implemented in combination with RParcoords and are discussed in Section 5.2.4 and Section 5.2.5.

Apart from making variable selections using the bar plots it is also possible to make selections using the variable scatterplot. When this is done, the selected bars are coloured red in the bar plots. Chances are that the red bars are not all concentrated on one part of the bar plots, but are instead spread out, as shown in the left image of Figure 19. This spread makes it hard to do anything else with these selections using the bar plot, such as refining the selection. To remedy this problem, a checkbox is present below the bar plots that allows a user to first order the bars on selection and then sort the bars based on the bar plot sorting chosen in the drop-down menu, as shown in the right image of Figure 19.

## 4.3 OBSERVATION SCATTERPLOT

The main part of DimRedPlot consists of two scatterplots, one for the observations and one for the variables. The observation scatterplot displays the observations projected onto eigenvectors using their factor scores. The eigenvectors that the observations are projected on are the ones selected using the eigen-bar. This is the part of DimRedPlot that most users of techniques such


Figure 19: Variable bar plots with some variables selected through the scatterplot. (Left) The bar plots before grouping on selection. (Right) The bar plots after grouping on selection.

as PCA will be familiar with, since the results of PCA are usually shown with a scatterplot.

Although there are other methods for displaying the values of observations on several axes, such as parallel coordinates, using scatterplots allows a user to directly see the structure in the explored data, whereas parallel coordinates makes this step potentially harder. An example of this would be the situation where the observations form a circular structure on the selected axes. A scatterplot will immediately show this structure, whereas parallel coordinates will show the observations as crisscrossing lines with little immediately visible structure. Furthermore, since the observations in parallel coordinates are spread out over a line, the different observations can easily occlude each other through being close together or through their lines crisscrossing each other. Scatterplots spread the observations over a two-dimensional area, which means that less occlusion will occur in scatterplots. Finally, the familiarity that users have with data being plotted on scatterplots means that users can quickly start working with the visualisation.

Due to restrictions in the way the observations are rendered, there is a limit on the number of observations rendered in this scatterplot. If more observations are available than the limit, the observation scatterplot will simply not be rendered. Currently, the limit is set at 2500 observations, as rendering more will cause the `HTML` page to run sluggishly in all popular browsers. However, in the future this limit could be removed by, for example, using WebGL or `HTML canvas` rendering.

Although a simple scatterplot can be helpful in exploring the projection of the observations onto the eigenvectors, it is rather limited in the information it provides. The scatterplot in DimRedPlot offers several additional features, such as excentric labelling and colouring to assist the user further in exploring the projections. These features are discussed in the following sections.

### 4.3.1 *Axes scaling*

Initially, tick marks were rendered on each of the axes in both scatterplots to indicate what values the observations had on the eigenvectors. Each scatterplot had a standard size and the projected observations were scaled in such a way that they filled the scatterplot. An example of how this looks can be seen in Figure 20. Unfortunately, this can have the effect that vertical and horizontal distance in the scatterplot have different meanings, making the scatterplot hard to interpret. Furthermore, when looking at different combinations of eigenvectors as plot axes, the fact that the entire plot is

always filled makes it hard to get an intuition for the difference in data-variance each eigenvector describes. Finally, the tick mark values do not have a straightforward interpretation, making the displayed numbers quite meaningless to users.

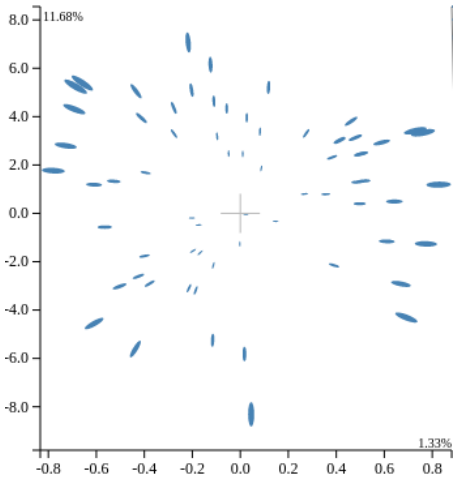


Figure 20: DimRedPlot with axes rendered.

To address these issues, the scaling of the observations is done in such a way that all eigenvectors are using the same domain when used as axes. To achieve this scaling we locate the observations with the minimal and maximal factor scores on all eigenvectors, and their values are used as axis domain. This has the effect that when plotting an eigenvector describing a high data-variance against an eigenvector describing a low data-variance, the observations along the second eigenvector will lie in a narrow part of the plot. An example of this can be seen in Figure 17.

Due to this axes scaling approach, no matter what eigenvectors are chosen as plot axes, the ratio between pixel distance and observation distance in the projection is constant. Consequently, there is no longer a reason to still show the axes, and as such the axes are not rendered anymore. Instead, two small arrows are rendered to represent the eigenvectors used as plot-axes. Both arrows also display the data-variance described by their eigenvectors. This makes it easy to see which one is used where as axis, as this would otherwise not have been entirely clear. To make it clear where the point $(0, 0)$ resides, which is where observations with average data values lie, a small plus sign is rendered there.

There is unfortunately a downside to having the same axis scale for all eigenvectors. When the two eigenvectors used as axes both describe low variance, the observations are usually rendered in a small cluster in the center of the plot. The reason for this is that the factor scores of observations along those eigenvectors are generally small, while the axes scaling is based on the minimal and maximal factor scores of all observations along all eigenvectors. This makes it very hard to discern any detail in the structure projected onto these eigenvectors. To remedy this, an option has been added to DimRedPlot to temporarily use the full scatterplot area to render the observations. When this mode is enabled, the axes arrows are coloured red to indicate the change to the user. Whenever the used eigenvectors are changed, this mode is disabled again to make sure that users do not accidentally get stuck in this mode. An example of this mode disabled and enabled can be seen in Figure 21. Looking at the images we notice that the zoomed image is actually smaller than the non-zoomed image. The reason for this is that in this mode, DimRedPlot will only use as much space as necessary, which in the zoomed case is less than normal.

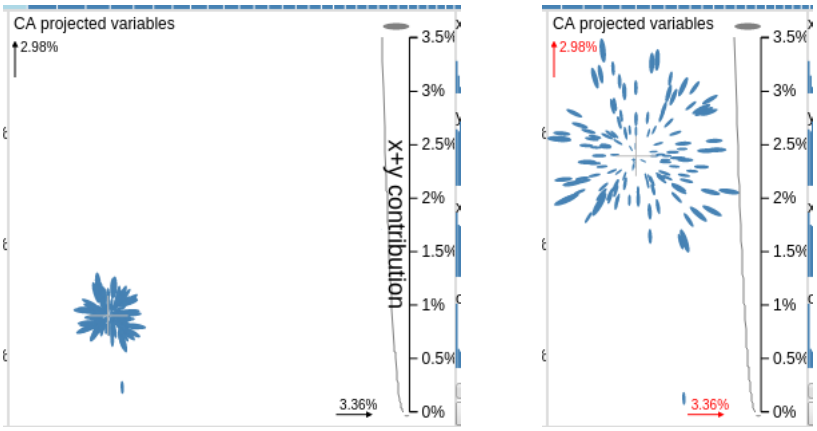


Figure 21: The difference between using unified axes, the left image, and using the full space, the right image.

### 4.3.2 *Colouring*

The observations in the observation scatterplot can be coloured. By default, and as shown in Figure 23, the observations are coloured using the combined contribution on the two used eigenvectors. However, the colouring can also be set from outside DimRedPlot, as discussed in Section 5.2.2, which can be used to colour the points based on an original variable.

A colourmap is shown next to the scatterplot, on which it is written what is used for colouring the points, e.g., contribution or a variable name. Interaction with the colourmap is also possible. Dragging the mouse cursor over the colourmap will result in selecting the observations that have the colours that are dragged over.

### 4.3.3 *Rotated ellipses*

In Section 4.2.1 we discussed the fact that the contribution of the observations to the eigenvectors is often not that interesting. This is mostly because a lot of datasets consist of a select set of different variables that describe the data, while there are countless observations that are individually not that interesting. However, this is not necessarily always the case; observations can very well be individually interesting, especially when dealing with contingency tables. In this case it can be useful to know which observations are responsible for the formation of specific eigenvectors.

We already saw in the previous section that observations can be coloured based on their combined contribution to both eigenvectors used as scatterplot axes. However, this can not tell a user how much of that contribution is to the x-axis' eigenvector and how much is to the y-axis' eigenvector. To solve this, the observations are rendered as fixed size ellipses with aspect ratios of 1 to 4 that can be rotated. This rotation is based on the ratio dictating how much of the combined contribution can be attributed to the two eigenvectors individually. The way this works is that every ellipse that only contributes to the x-axis' eigenvector is aligned with the x-axis in orientation. While every ellipse that only contributes to the y-axis is aligned to the y-axis' eigenvector. If an ellipse contributes to both eigenvectors, its orientation will be in between the two axes.

### 4.3.4 *User interaction*

There are several user interactions possible within the scatterplot. When hovering the mouse cursor over the scatterplot area a circle is rendered centred on the mouse cursor, as shown in Figure 22. Using the scroll wheel the radius of the circle can be adjusted. Using the circle it is possible to both show labels for the plotted observations and to select them. We explore both possibilities in the following paragraphs.

LABELLING Looking at the observations, it is not clear what they mean. Initially, labels were shown for the selected observations, or for the observations with a sufficiently high contribution value. These labels were simply rendered at the coordinates at which they were projected, which easily resulted in overlapping labels, making them unreadable. As such, we needed a better way to place the labels.

Label placement is a difficult problem that is often worked on in the context of geographical maps. The biggest reason this is such a difficult problem is the fact that it is an NP-hard problem. Although a lot of work has been spent in finding fast labelling algorithms, as is shown by J. Christensen et al. [31], most of the work is focused on solving the problem for geographical maps. Here, it is generally areas that are labelled, not points. Geographical maps often have some restrictions on how small the areas can get, making it easier to find a labelling for these maps. In our case, we are trying to label individual points without there being any restriction on how densely the points may be clustered together. In fact, close clustering of points is to be expected when we are dealing with a high number of them. As such, most of the proposed algorithms are not applicable to our problem.

In order to label the observations, we have instead chosen to use excentric labelling [32]. Any observations that fall within the earlier described circle will have their labels drawn on either the left or right side of the circle. Lines are rendered from the observations to their labels. To avoid this from becoming an entangled unreadable mess of lines and labels, the labels are vertically centred around the circle. This means that if the circle is moved, the labels are moved as well. The motion of the lines makes it intuitively clear which label belongs to which observation. An example of the excentric labels can be seen in Figure 22.

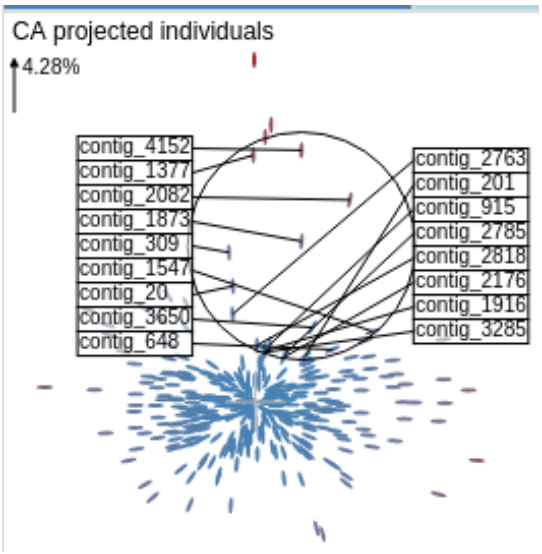


Figure 22: Excentric labels, shown around the mouse cursor.

When the number of observations rendered is high, there are too many labels to show using excentric labels. To combat this, the excentric labels are sorted on the combined contribution to both axes. This has the effect that if too many labels have to be rendered, the observations with the highest contribution are still labelled. These are generally the most interesting since they contribute the most to the data-variance shown in the scatterplot. To still make it possible to see the label of any observation desired, the user can scroll the mouse wheel when the circle around the mouse cursor is shown. This scrolling will result in the circle changing size. The changing in size allows a user to show the labels of a smaller subset of observations.

Beyond the excentric labelling, it is also possible for a user to obtain more detailed information about a single observation. This can be done by making sure that the circle around the mouse cursor only encompasses one observation. When this happens, a tooltip is used to show the factor scores of the observation, its contributions, and in case of using CA or MCA, its mass.

SELECTION The circle around the mouse cursor can also be used as a brush to select observations with. By moving the circular brush while pressing the left mouse button, all observations in the circle all selected. Unless `CTRL` or `SHIFT` is pressed, any previous selection is removed. Earlier versions used a rectangle that could be drawn with the mouse cursor; however, sometimes a wanted selection does not fit a simple rectangle. The circular brush allows for much more complex selections.

Selecting observations will result in them being given the colour black or white, depending on whether the background is respectively light or dark. Black and white have been chosen because they are never part of the colourmap that is being used, as these colours might not be visible depending on the background. The size of their ellipses is also slightly increased when selected. This makes the selection extra clear as it is can be hard to see the difference between, for example, black and dark blue.The result can be seen in Figure 23.

When the user makes a selection, the selected observations are also rendered on the colourmap as small lines. This allows the user to see more exactly what the contributions, or variable values, of the selected observations are. The colour of the lines are, again, black or white.

## 4.4 VARIABLE SCATTERPLOT

The scatterplot showing the projected variables is placed on the right side of the observation scatterplot. The placement of the two scatterplots has been chosen to make sure that the bar plots, described in Section 4.2, are close to the scatterplot showing the variables. This is because both the bar plots and the variable scatterplot describe variables. Since the bar plots are on the right side, the variable scatterplot is as well.

There are several elements in the variable scatterplot which are the same as in the observation scatterplot. The axes scaling and ellipse rotation work the same for both scatterplots. User interaction involving selection and labelling using the circle around the mouse cursor can also be used in the variable scatterplot. The variable scatterplot differs in the fact that the variables can not be colour mapped, but are instead “sizemapped”. The colour used for selected variables is also different.

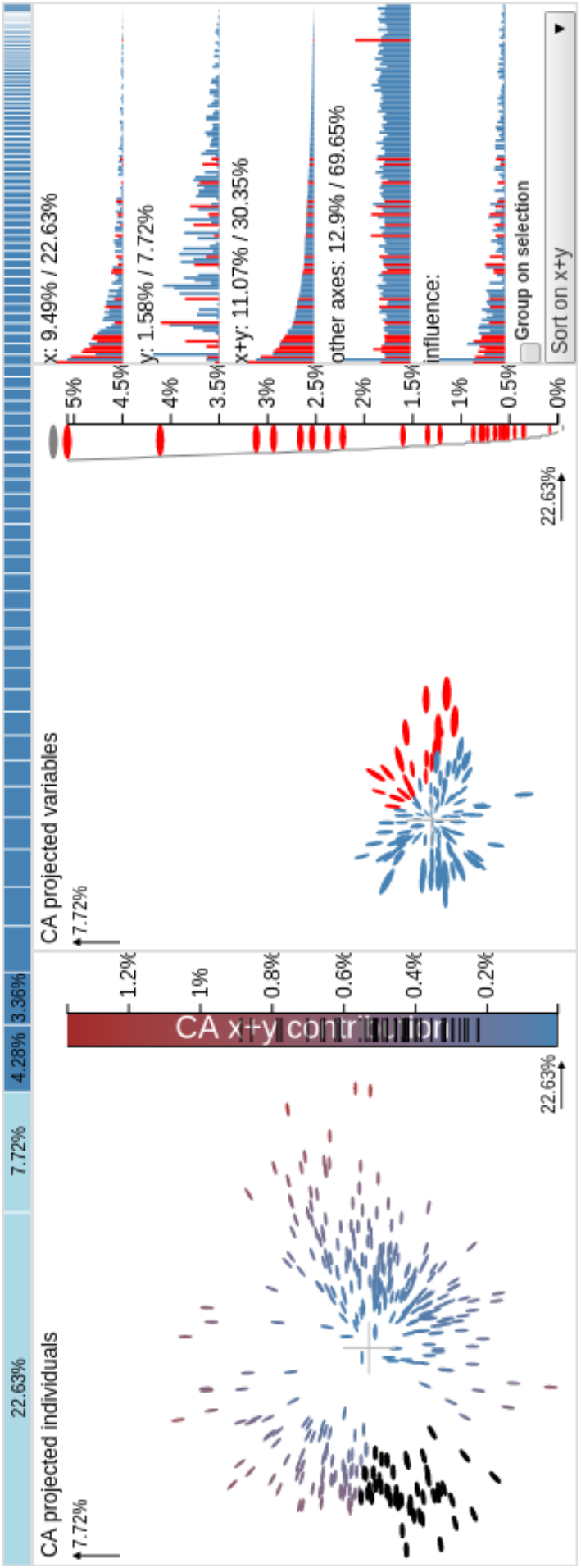


Figure 23: Both variables and observations selected in DimRedPlot. The selections have been made in the 2 scatterplots and they are reflected in the colourmap, the sizemap, and the bar plots.

Unique elements in both scatterplots, such as the colouring, were introduced to make the two scatterplots sufficiently different. Originally, the two scatterplots were very similar. Early tests with users showed that users were confused by this and had trouble understanding that the variable scatterplot showed variables and not observations. With the introduced difference in both scatterplots the concept of a variable scatterplot has been much easier to grasp for users. Besides this reason for difference, there is also the reason that both scatterplots have slightly different use cases.

### 4.4.1 *Size mapping*

As described in Section 4.2.1, the contribution variables have to eigenvectors can help explain those eigenvectors and the structure of observations projected onto those eigenvectors. Although the bar plots in DimRedPlot can be used to find how much each variable contributes to the eigenvectors, it is hard to visually link the information displayed in the bar plots to the variables displayed in the scatterplot. It would be possible to use a combination of colouring and rotated ellipses to solve this problem, but using colouring in both this scatterplot and the observation scatterplot turned out to be confusing to users.

Instead of using colouring, the combined contribution to both the x and y-axis' eigenvectors is encoded in the size of its ellipse. Here, the contribution is linear in the area of the ellipses. Originally, the contribution was linear in the length of the ellipses. However, this had the disadvantage that the ellipses close to the centre were all very similar in size, while most generally reside there. Furthermore, when the area of a glyph is not linear in the metric it encodes, users can have difficulty interpreting the glyph size, as shown in Chapter 2 of The Visual Display of Quantitative Information [33].

To help the user translate a certain ellipse size to a certain contribution, a "sizemap" is shown to the right of the scatterplot. This sizemap works the same way as a colourmap in that it shows for every size what contribution belongs to it. Above and below the sizemap grey ellipses are shown that represent the largest and smallest possible ellipses. This helps the user intuitively understand what the sizemap is meant to represent. Similarly to the colourmap in the observation scatterplot, interaction with the sizemap is possible. Dragging the mouse cursor over the sizemap will result in selection of the variables that have the colours that are dragged. The sizemap also shows the text "x+y contribution" below it, to make it clear to the user what the sizemap is representing.

### 4.4.2 *Alternative visualisation*

The current way of visualising the variables in the scatterplot is not the only way that has been experimented with. Other visualisations were developed but ultimately rejected in favour of the current method of rendering variables.

The first way that the variables were visualised was by using rectangles. The rectangles were centred around the projected variable. The height of the rectangles was linear in the contribution of the variable to the y-axis, while the width was linear in the contribution to the x-axis. Figure 24A shows an example of variables rendered as rectangles.

The idea of the visualisation was to encode for every variable what the contributions to the x-axis and y-axis were. However, it can be hard to visually compare the width to height ratio of two rectangles that are far away to each other.

A second method visualised the variables as ellipses, just like the current method. The difference here was that the ellipses were not all the same shape.

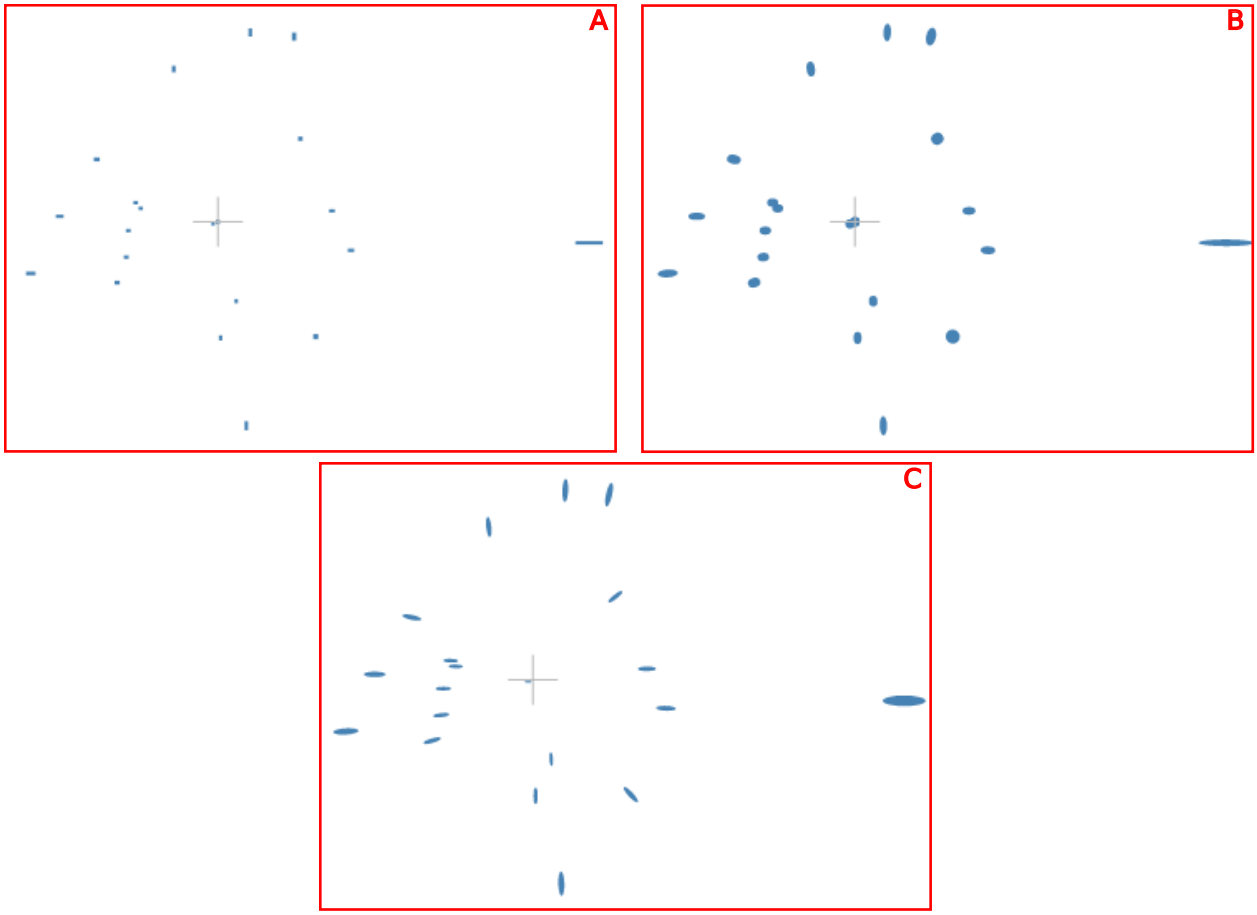


Figure 24: Comparison of different variable render methods. (A) Points rendered using rectangles, with width and height corresponding to respectively x and y contribution. (B) Points rendered using ellipses, with width and length corresponding to x and y contribution. (C) The current variable rendering. This uses ellipses with a predetermined shape. Ellipse area corresponds to $x + y$-contribution.

The ellipses were all rotated the same way as they currently are, but their length was linear in the maximum of the x-axis and y-axis contribution, while their width was linear in the minimum of the x-axis and y-axis contribution. The result can be seen in Figure 24B. The downside to this method is that due to the way that variables are naturally distributed, e.g., many near the average and few extremes, many variables near the average are rendered as small circles, while the extreme ones are rendered as very long and narrow ellipses or slightly bigger circles. This makes it hard to easily differentiate between the different variables.

### 4.4.3 *User Interaction*

Selecting variables will result in them being given the colour red, instead of black or white. Red was chosen since it is a strong contrast with the regular blue of the ellipses. While choosing selection colours it was also made sure that the selection colour of the variable scatterplot and the observation scatterplot were sufficiently different. This was done since a selection of variables means something different as an observation selection.

A selection made in this scatterplot is also reflected in the sizemap and the bar plots, as can clearly be seen in Figure 23. The sizemap will display

the selected ellipses on their appropriate size. The bar plots will render any variable selected as a red bar instead of the regular bar. Care was taken to assure that the colouring in both the scatterplot, bar plots, and sizemap is the same, e.g., blue for non-selected and red for selected. This helps make it intuitive that all these items represent the same thing. Figure 23 shows both observations and variables selected, to illustrate the difference between observation and variable selections.

## 4.5 IMPLEMENTATION

DimRedPlot is written using `JavaScript` and `D3` [34]. `D3` is a library to easily create visualisations from data. A prevalent pattern in `D3` is to design visualisations as reusable charts. One of the benefits of reusable charts is that they are easy to deploy and integrate in different systems. DimRedPlot is also written as a reusable chart. This means that besides being used in RParcoords, it could very easily be used in other environments.

## 4.6 DISCUSSION

Using elements such as the eigen-bar, contribution bar plots, and excentric labelling allows DimRedPlot to both study a larger part of the dimensionality reduced space than other methods generally do, and it makes finding the relation between the dimensionality reduced data and the original data easier. DimRedPlot could be used as a stand-alone tool. However, some of the interactions build into DimRedPlot, such as selection of variables, are only useful when DimRedPlot is used in combination with other visualisation techniques such as parallel coordinates. Other features, such as colouring and observation selection can be useful on their own, but can be greatly expanded in usefulness when combined with other tools. This is discussed in the next chapter, where we look at the integration of DimRedPlot into the larger RParcoords environment.

Currently, the visualisation has been used with PCA, CA, and MCA. DimRedPlot does not contain any code specific to PCA, CA, or MCA. Instead, it expects its input to be in a certain format. All three dimensionality reduction techniques can have their results outputted in this format and can thus be used in DimRedPlot. There are other dimensionality reduction techniques, such as Multiple Factor Analysis, of which the output can also be given in the expected format. As such, these techniques are theoretically also supported by DimRedPlot. The downside to this approach is that certain features of techniques are not given a lot of attention because they are unique to one method. An example of this is row and column mass in the case of CA, which are not encoded into the visualisation even though this may be useful. Other techniques, such as non-linear dimensionality reduction have output that does not fit the expected input format to begin with, and changes would have to be made to DimRedPlot and the expected input format in order to support techniques like these. Section 7.1.2 discusses with some more detail what changes would have to be made in this case.

# 5 RPARCOORDS

In Chapter 4 we discussed the design of DimRedPlot. DimRedPlot is meant to be used in combination with more general visualisation techniques, such as parallel coordinates or scatterplot matrices. Combining DimRedPlot with such visualisations means we can link views in such a way that the relationship between the dimensionality reduced data and original data becomes clear. In our case, DimRedPlot has been integrated into a larger high-dimensional data exploration tool called RParcoords, which is centered around parallel coordinates, described in Section 2.1.5. This chapter focuses on how RParcoords works and how DimRedPlot has been integrated with it.

We start this chapter with a discussing of the functionality and design on RParcoords alone. After this, we look into the many design decisions of DimRedPlot that were made in the context of the larger visualisation, and how the interaction between the two visualisations works. At the end, we have a look at the implementation details of RParcoords.

## 5.1 DESIGN AND FEATURES

Figure 25 shows an example of RParcoords. In the image, RParcoords is visualising the biogas dataset described in Section 6.2. The biogas dataset contains abundance levels off about 30000 DNA fragments in several biogas reactors. The abundance levels have been measured at 7 different time-points. The dataset is used to find out what bacteria are present in the biogas reactors. Looking at the figure we can see that the visualisation is divided into several parts. On top is a bar with some colouring options, some information on how much data we are looking at, and an option to switch between a light and a dark theme. Below the top bar is the actual parallel coordinates visualisation, spanning the width of the page. Finally, below the parallel coordinates is an collapsible options menu, from which the visualisation can be controlled.

Through the options menu and the parallel coordinates, users of RParcoords have access to a large number of features, such as colouring, selections, and clustering. The following sections describe which different features RParcoords offers and why they have been added to RParcoords.

### 5.1.1 *Parallel coordinates*

RParcoords is, as the name suggests, centred around a parallel coordinates visualisation. Using parallel coordinates allows RParcoords to support a variety of scientific domains and problems in the exploring and analysing of high-dimensional datasets.

When RParcoords is started the parallel coordinates visualisation displays a set of variables which is based on the used dataset. Every variable is represented as a vertical line with the variable name as label and an axis depicting the variable values. Lines, representing the observations, are drawn from left to right going through the variable lines. The places along the variable axes where the observation lines intersect indicate the associated values of the variables.

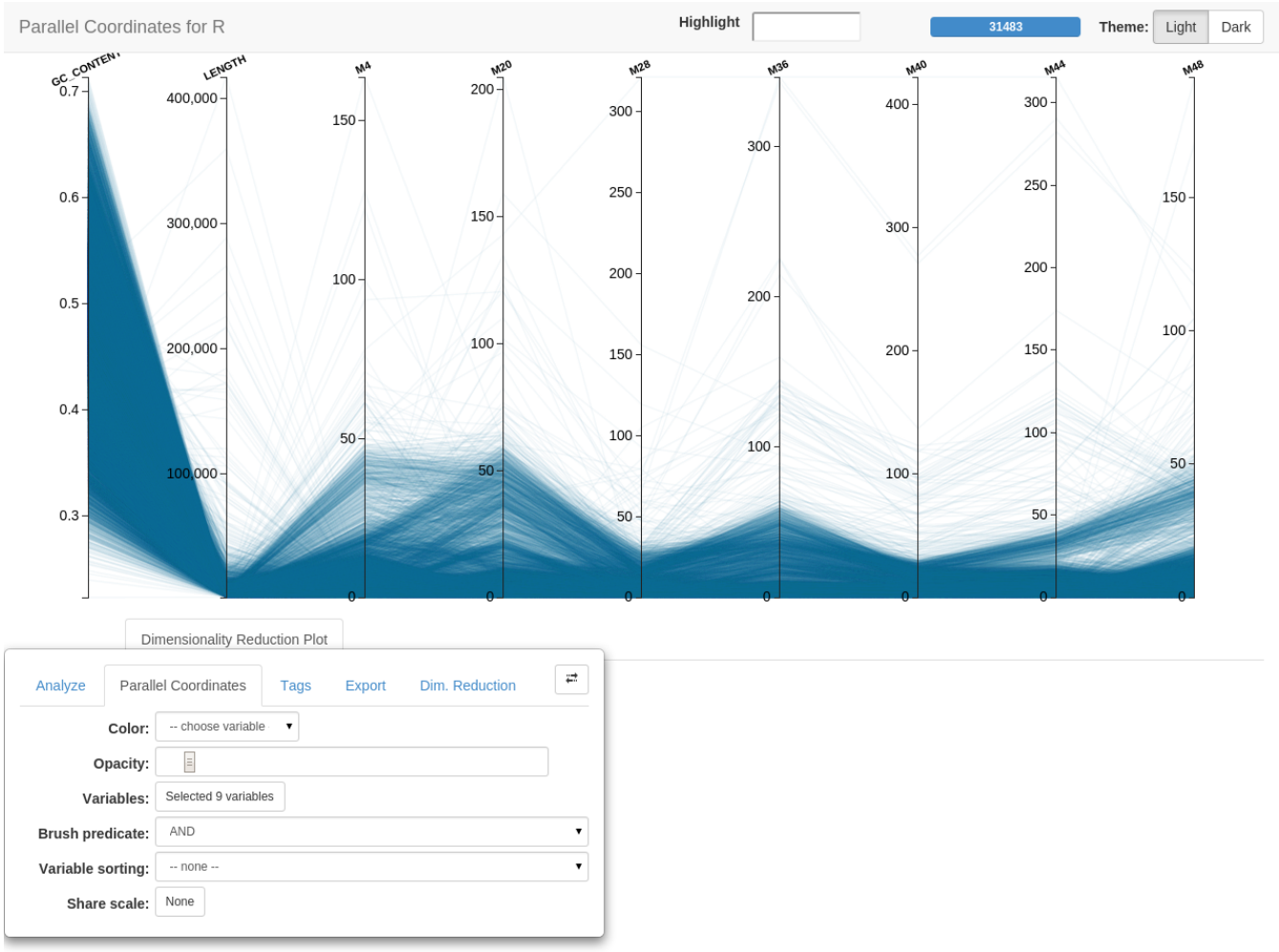


Figure 25: An example of RParcoords showing a dataset consisting of small DNA sequences.

The variable labels are rotated at a slight angle to avoid overlap when a large number of variables are shown at the same time. A user can change the orientation of the labels by scrolling while hovering over the labels. The user can also reorder variables by dragging the vertical variable lines around. This is useful when the default ordering is not the most natural for the particular dataset.

### 5.1.2 *Selection and filtering*

Often, the dataset is too big to be studied all at once, or it might simply be interesting for the use case to focus on a smaller part of the dataset. In order to focus on a subset of the dataset, it is possible both to select observations to highlight them and to remove observations from the dataset.

Selecting can be done by dragging the mouse cursor along one of the variable lines in the parallel coordinates. The area of selection made by the user will be indicated with a grey rectangle over the variable line. Since the selection rectangle does not disappear until the user makes a single click on the variable line with the rectangle on it, it is possible to make multiple selections at the same time on different variables. In fact, it is even possible to make multiple selections on the same variable. When multiple selections are made the final selection is a logical intersection between all of the smaller selections. This can be changed by the user to be a logical union.

To make it clear to the user which observations are currently selected, selected observations are rendered black or white, depending on whether the theme is light or dark respectively. The selection colours used are, not coincidentally, the same as those used in DimRedPlot's observation scatterplot, as described in Section 4.3.2. Both the parallel coordinates and the scatterplot represent the same observations. Turning the visualisations into linked views, through amongst other things using the same colours, helps the user to understand this. This idea of linked views is further explored in Section 5.2.

Filtering allows a user to actually remove observations from the parallel coordinates altogether. When data is selected, two options appear in the options menu: "keep selected" and "remove selected". Selecting these options will either remove the non-selected observations or remove the selected observations. The advantage of using filtering is that any further analyses are performed as if the removed observations do not exists. The most obvious example of this is that the parallel coordinates are redrawn and it will make full use of the vertical space. If, for example, outliers have been removed from the data, the remaining observations are spread out over a larger area making it easier to study the data. It also means that when clustering or dimensionality reduction is applied, it is applied only on the remaining data.

### 5.1.3 *Tags*

Sometimes, it is useful to come back to a previously made selection later on in the same session or even in another session. To accommodate this, RParcoords contains a tagging system. Before this system was in place, researchers had to write down precisely how they got to a selection in order to retrieve that selection later on. With the tagging system, researchers can tag a selection, which is then stored on by the R backend.

The tagging interface, seen in Figure 26, only shows up when tags are available or when a selection is made. When a selection of observations is made through whatever means, the user is presented with the option to enter a new name to create a tag or to enter an existing name to add the selected observations to a tag or to overwrite a tag. The name of a tag can be given

in a text-input field which supports autocompletion. This makes it easy for a user to find and select an existing tag without the risk of misspelling its name.

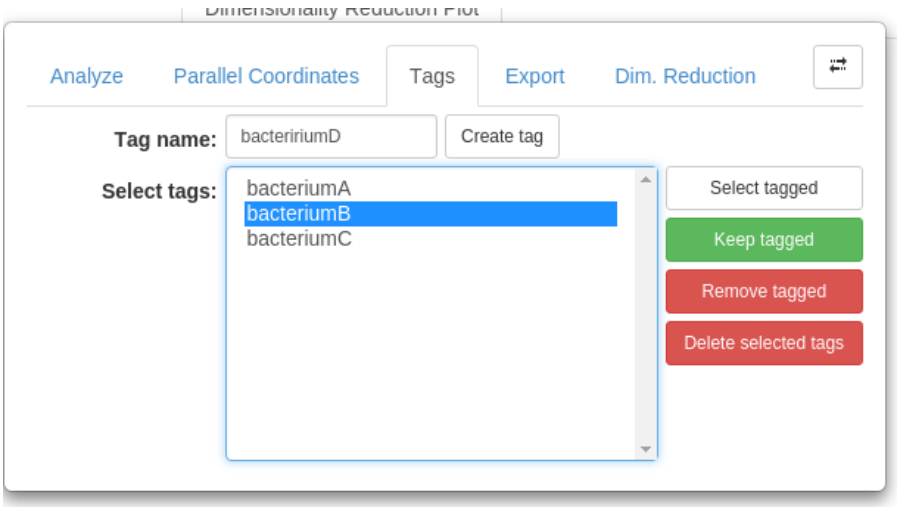


Figure 26: User interface for creating tags.

When a tag is created, it is added to a list containing all available tags. The list of tags is shown below the tag creation interface. Users can select multiple tags in the list and select the observations represented by those tags in the parallel coordinates and DimRedPlot. It is also possible to keep the tagged selections or remove the tagged selections from the dataset entirely. Finally, a user can also remove existing tags if those are no longer required. To keep things clear, when a subset of the dataset is being shown because filtering was applied, only those tags are shown in the list which contain observations that are in the subset. This means that any tags of which observations are not shown can not be used to select those observations.

### 5.1.4 *Transparency*

By default, transparency is applied to the lines rendered in the parallel coordinates. The level of transparency can be modified through the options menu. There are different situations in which transparency is a useful feature. When the dataset being visualised is quite big, any structure in the dataset can be hard to see as it might be buried under lines rendered later. Also, because it is hard to both draw a large number of lines and make lines distinguishable from each other, the eventual visualisation may simply look like a big blue blob. Using transparency, areas where only a few number of lines reside have a more empty look, while areas with more lines look fuller, making it easier to distinguish different structures in the data. Figure 27 shows the parallel coordinates with transparency in the top image and without transparency in the bottom image. In the top image some of the clearly visible structure has been highlighted. When comparing the top image to the bottom one, we can see that almost none of the structure visible in the top image is visible in the bottom image.

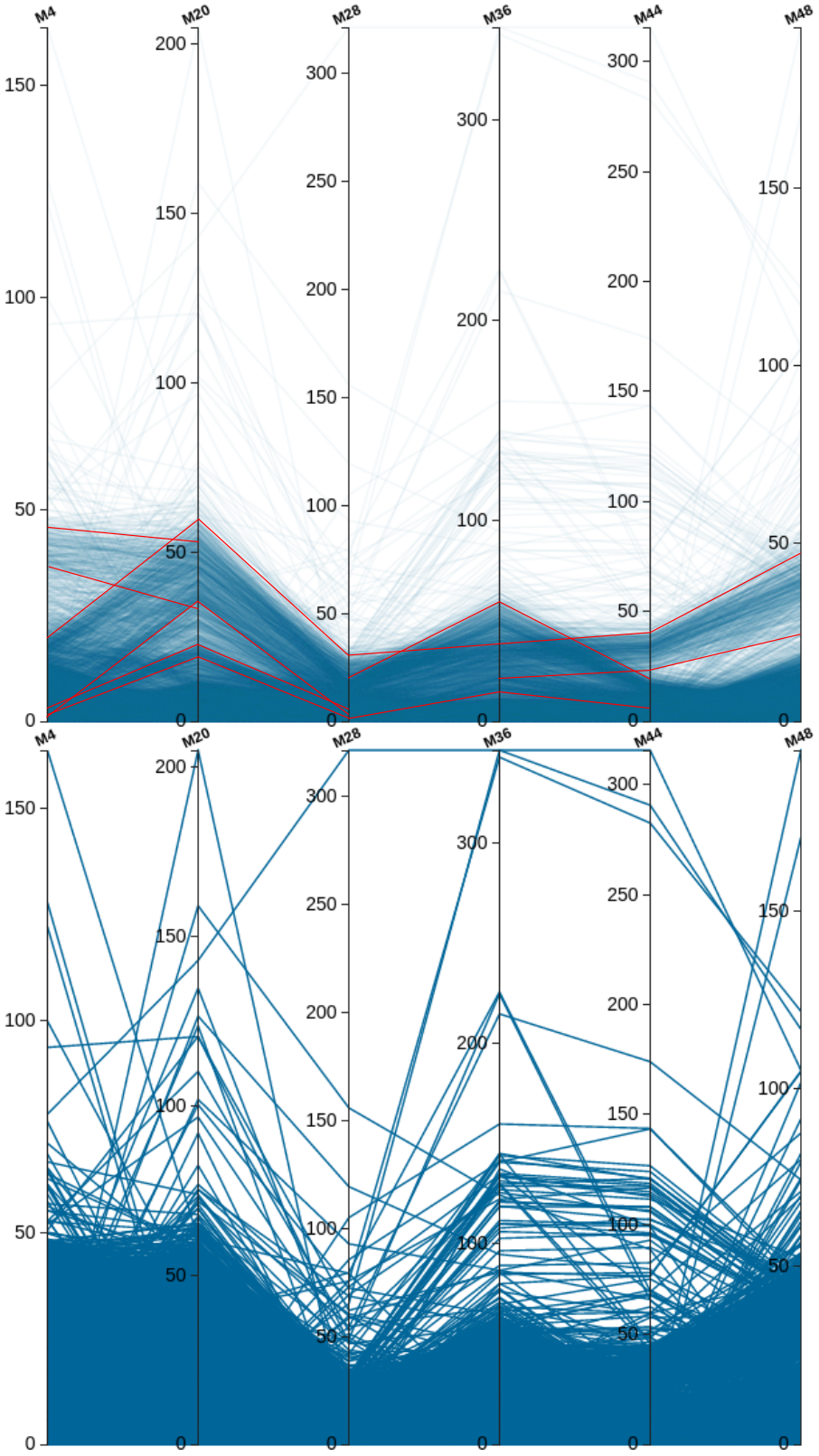


Figure 27: Comparison between rendering parallel coordinates using transparency and without using transparency. (Top) Parallel coordinates with transparency, some of the clearly distinguishable data features have been highlighted with red lines. (Bottom) Parallel coordinates without transparency. Almost non of the earlier marked structure can be distinguished here.

Another benefit of using transparency occurs when visualising a low number of observations. When only a small number of observations are rendered most lines can be distinguished in the parallel coordinates. However, when two or more lines overlap because their observations partially have the same variable values, it can be hard to see that overlapping is happening. Using transparency the lines that overlap seem thicker, which allows a user to spot the overlapping lines. An example of this is described in Section 6.1.3.

The amount of transparency used can be configured in the options. This is needed as more transparency is needed when more observations are rendered.

### 5.1.5 *Highlighting*

Sometimes it is needed to find the line representing a specific observation in the parallel coordinates. Especially when a lot of observations are being rendered at the same time, individual lines can be hard to find.

To support the task of finding a specific observation, a text field resides in the top bar. When the name of an observation is entered in the top bar, the line representing the observation lights up, while the rest of the lines are faded out.

### 5.1.6 *Colouring*

Beyond using transparency, selections, and highlighting to discern different patterns in the data, it is also possible to apply colouring to the lines in the parallel coordinates. There are several colour schemes and methods that can be used to obtain different colouring effects. Besides being useful for discerning patterns in the data, colouring is also useful in combination with the dimensionality reduction visualisation as described in Section 5.2.2.

To apply colouring, a user can go to the options menu and select "manual selection", a variable in the data, or a generated variable such as a clustering or contribution. When a categorical variable is chosen, including clustering variables, the user can select a colour set from several sets to apply to the different categories. The colour sets have been chosen using ColorBrewer [35]. Except for one, the colour schemes are chosen to use many easily discernible different colours. The exception is a colour map designed to gradually go from one colour to another. This is useful when the data contains categorical variables where the categories have a natural order.

Choosing a numerical variable allows the user to choose between 3 continuous colourmaps and 5 decile colourmaps. The decile colourmaps have been chosen using ColorBrewer, whereas the continuous colourmaps have no particular source.

Manual selection allows the user to make selections in the data and then choose a colour for that selection. The same colour sets available when doing categorical variable colouring can be chosen to select colours from. When a selection is made, the colour button in the top bar can be selected to show a drop-down menu with the colours from the chosen colour set, as seen in Figure 28. Selecting one of the colours will colour all the observations in the selection with that colour.

The control this colouring mode gives is useful in combination with DimRedPlot in distinguishing interesting data features such as clusters, as described in Section 5.2.2. Beyond this, manual selection is sometimes useful to use instead of categorical variable colouring. When using categorical variable colouring there is no control over which categories get which colours. Using manual selection gives a user more control over this which is useful when a user suffers from colourblindness and can only discern specific colours,

or when the standard colour schemes are not discerning enough in themselves. In this case, a user has to make sure the categorical variable is shown in the parallel coordinates, select a category of the variable in the parallel coordinates, and apply a colour from the drop-down menu.

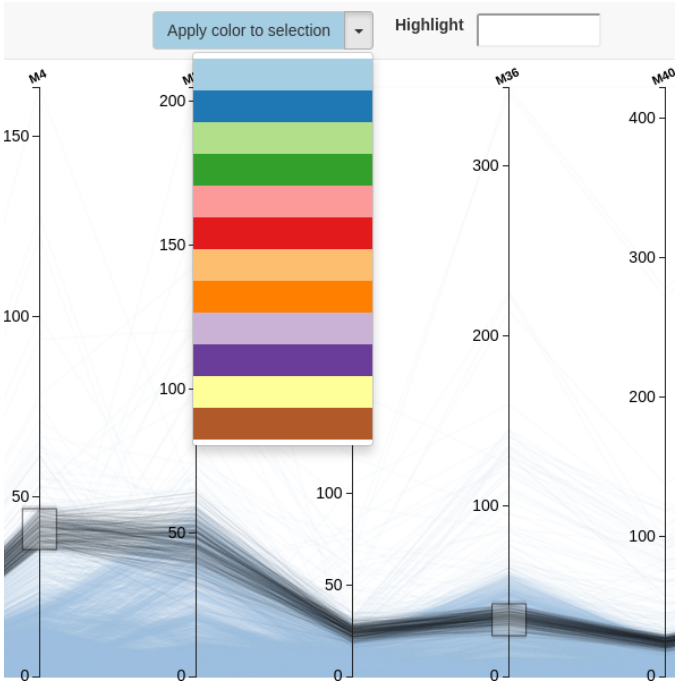


Figure 28: Colouring a selection in the parallel coordinates using the "manual selection" mode.

### 5.1.7 *Variable ordering*

One of the problems when dealing with parallel coordinates is that the order of the variables is by default arbitrary, which can make it hard to detect patterns within the data. To combat this, an option is present that allows a user to order the variables based on some criterium. Currently, the variables can only be ordered on one criterium. This criterium is the screen-space height in pixels to which the average value of the variable would be mapped. The effect of ordering the variables is that the variables with low valued outliers are shown to the left, while variables with high valued outliers are shown to the right. In theory there should not necessarily be a correlation between two variables with the same type of outliers; however, neighbouring variables will now share the part of their domain where most observations reside, which makes it easier to visually compare them. By default, variables may be ordered in a highly chaotic way, having the interesting parts of their domain constantly jump up and down visually.

Even though variable ordering can make parallel coordinates less chaotic, when a selection is made in the parallel coordinates, the selection can still move very chaotically through the parallel coordinates. Because of this, when a user makes a selection with variable ordering turned on, the variables are ordered on the selection instead of on the entire dataset. This allows a user to quickly see where the values of the selection is highest or lowest on average and in variance. Figure 29 shows an example of variable ordering based on a selection. Here we can clearly see that the selection on the left-most and right-most variables is lower in variance but has higher and lower average

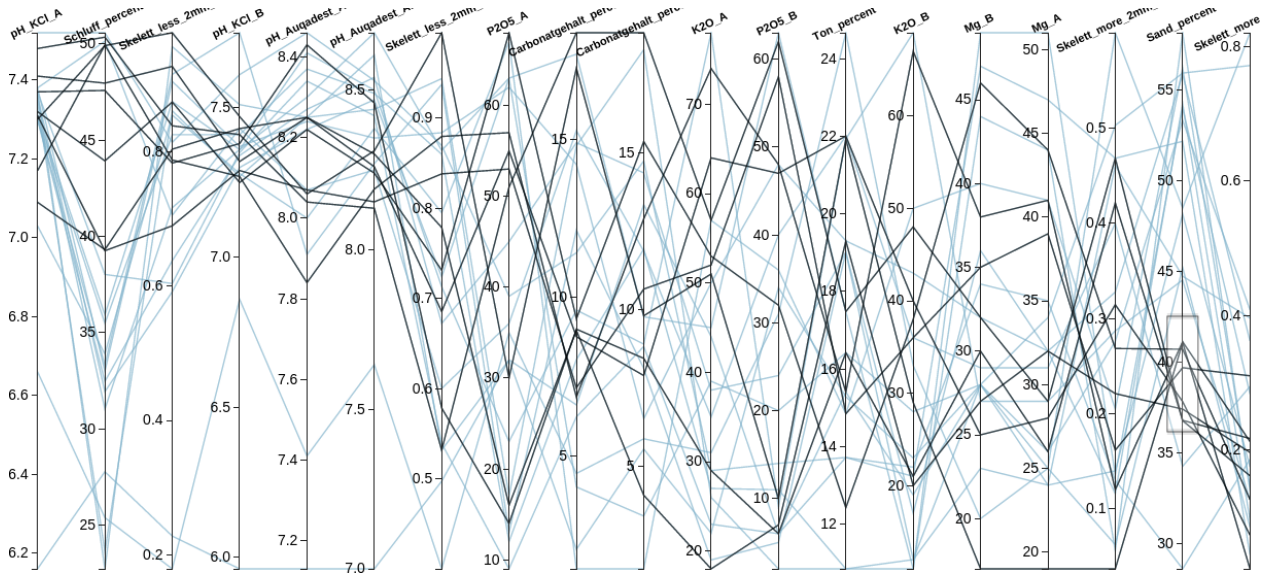


Figure 29: Variable ordering applied to parallel coordinates based on the pixel height of the average selected value.

values, while the selection has a higher variance on the variables in the centre.

### 5.1.8 *Clustering*

Due to certain use cases requiring it, there is clustering support within RParcoords. Although the support for clustering is not really relevant in the context of DimRedPlot, clustering is used in the evaluation so a short explanation of the option will be given here.

From the options menu both k-means clustering and hierarchical clustering can be performed. The k-means clustering is performed on standardised data using an Euclidean distance metric. The hierarchical clustering uses the Pearson correlation coefficient as metric. Several parameters can be chosen for the clustering methods, such as the number of clusters for k-means clustering. The actual clustering is performed on the observations in the dataset that are currently visible, e.g., the filtered out observations are ignored. The clustering is performed by the R backend, and the resulting clusterings are stored as categorical variables in the dataset. Every observation in the dataset is given a number for this variable indicating the cluster that observation is in. Storing the clusterings in the backend means that the clusterings can be used in the visualisation even after closing it and later starting it again.

## 5.2 DIMREDPLOT INTERACTION

RParcoords contains support for displaying DimRedPlot instances in its visualisation. One of the pages in the options menu is devoted to performing dimensionality reduction on the dataset currently being visualised. To perform dimensionality reduction, a user will first have to select the desired method, PCA, CA, or MCA, after which variables can be selected from a list of appropriate variables for the selected techniques. The resulting DimRedPlot instance is rendered below the parallel coordinates aligned with the right screen border.

Several interactions between the features in RParcoords, such as parallel coordinates and colouring, and DimRedPlot have been development to allow DimRedPlot to be used effectively in combinating with RParcoords. Figure 30

displays an example of RParcoords with multiple DimRedPlot instances. Some of the core interactions between RParcoords and DimRedPlot have been outlined in the image and are discussed in the following sections.

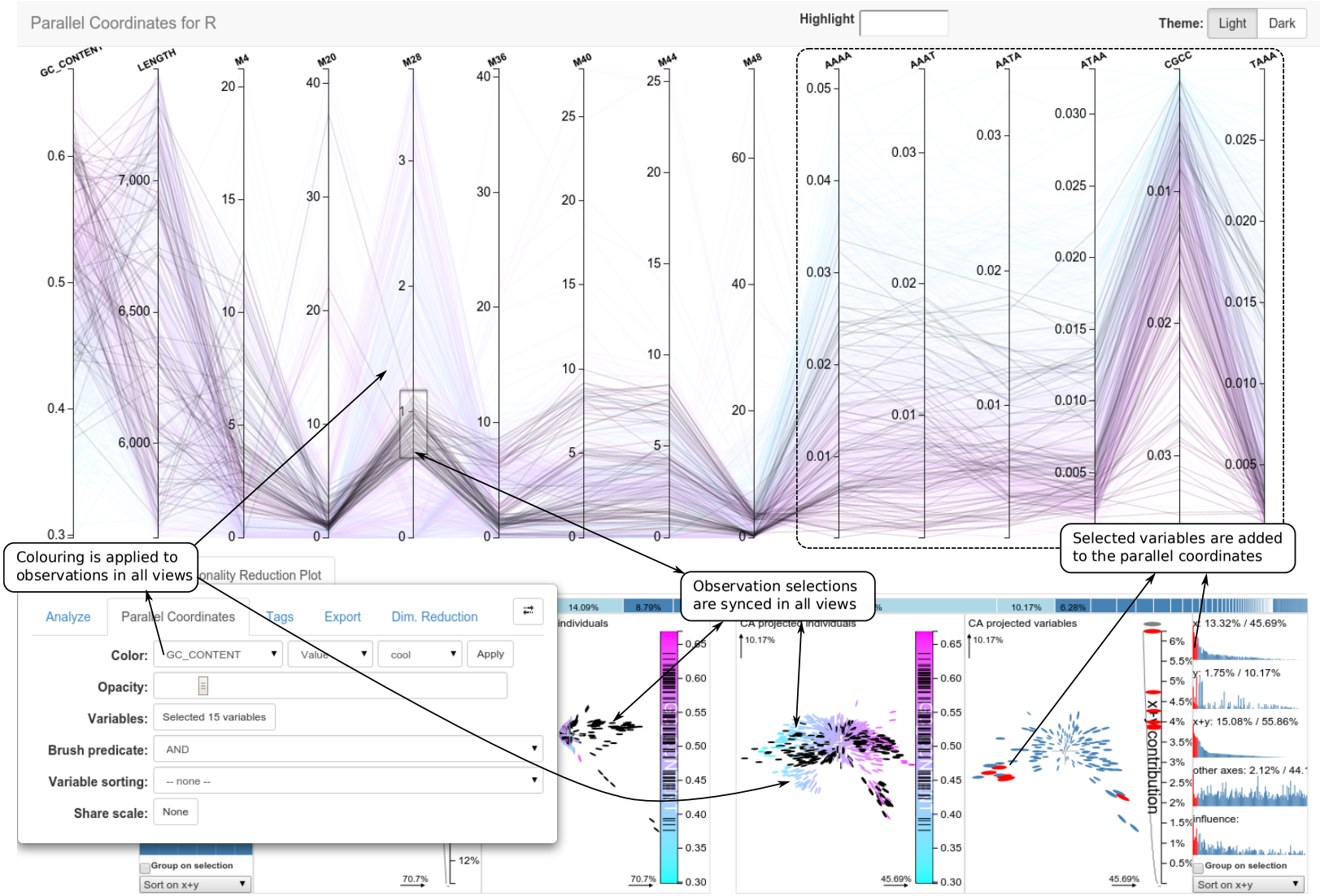


Figure 30: RParcoords with two DimRedPlot instances. Some of the most important interactions between the different views are shown. Observation selections and colouring are synced among the parallel coordinates and DimRedPlot instances, and variable selections in DimRedPlot can be used to extend the parallel coordinates.

### 5.2.1 *Multiple DimRedPlot instances*

A user can do multiple dimensionality reductions and the resulting DimRedPlot instances will be placed next to each other. This is the main reason that the options menu is collapsible as there would otherwise not be enough horizontal space. There is no practical limit to the number of DimRedPlot instances being shown to each other; however, when a user performs dimensionality reduction using PCA, CA, or MCA, any existing DimRedPlot instance showing the results from the same technique will be overwritten. For example, if there is already a DimRedPlot instance displaying PCA results and PCA is performed again, the existing DimRedPlot instance will be filled with the new PCA results. As a result, it is only possible to show three DimRedPlot instances next to each other.

Even though three DimRedPlot instances can be shown next to each other, it would be unwise. Unless a extremely wide screen is used, the DimRedPlot instances would only have a small horizontal area and would thus be greatly compressed. This would make it hard to distinguish the different features of the visualisation.

Having multiple DimRedPlot instances next to each other can be very useful. Any selection of observations made in one of the DimRedPlot instances is synced with the other instances, which allows for easy interaction between the different visualisations. An example of when this is useful is when a user wants to study both categorical and numerical variables at the same time. None of the supported dimensionality reduction techniques allow both numerical and categorical data to be processed at the same time. However, by showing the results of MCA and PCA next to each other, it is possible to see the relationship between the two analyses.

To support users' understanding of the relationships between observations in different dimensionality reduction analyses, the visualisation of the DimRedPlot instances is slightly modified. Whenever a second DimRedPlot instance is added, the left-most instance will be horizontally reversed. This means that the bar plots will be drawn on the left, the variable scatterplot will be drawn in the centre, and the observation scatterplot will be drawn on th right. The effect of this is that the observation scatterplots of both DimRedPlot instances are next to each other. Because the observations in both scatterplots are the same, it makes sense to have them closer together. The variables in both instances will always be disjunct to each other, which makes it logical to render them far from each other. Not only is this setup more intuitive, it also lessens the screenspace that the eyes have to traverse to compare a selection of observations in one DimRedPlot instance with a selection in the other instance. An example of this setup can be seen in Figure 35 and Figure 36 in the evaluation.

### 5.2.2 *Colouring*

In Section 4.3.2 it is mentioned that DimRedPlot supports colourmapping of the plotted observations. Using this colouring support, any colouring that is applied to the parallel coordinates, as described in Section 5.1.6, is also applied to DimRedPlot. This way, the observations in both DimRedPlot and the parallel coordinates are always coloured the same way. This helps not only to make things clear for the user as one colour will always mean the same thing, but also to better see the relationship between the observations in both visualisations.

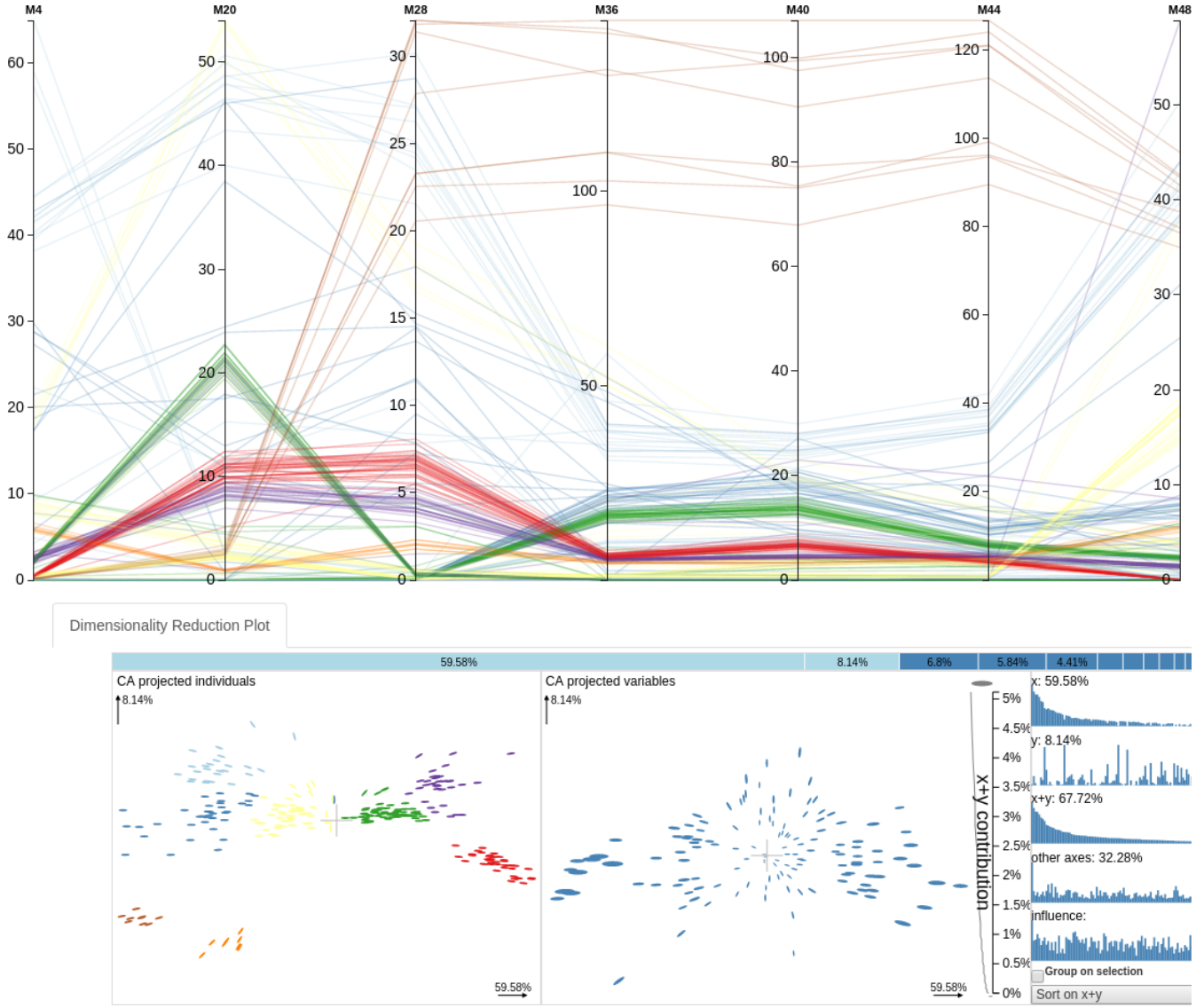


Figure 31: Clusters in DimRedPlot have each been given a different colour, which are then also applied to the same observations in the parallel coordinates. Through the colours, we can clearly see how the clusters in DimRedPlot relate to the clusters in the parallel coordinates.

Colouring observations in DimRedPlot can serve several purposes. Colouring the observations based on some variable is useful to help the user understand the relationship between the original data shown by the parallel coordinates and the dimensionality reduction results. An example would be if the dataset contains a variable that should be or is important for the particular use case. To find out what the relationship is between the observations in the scatterplot and this variable, a user can look at the contribution bar plots described in Section 4.2. However, this can still be a bit abstract whereas colouring the observations based on the variable can feel much more natural to the user as the relationship between the dimensionality reduction and the original data is directly visible.

Another use case of colouring is to link structure in DimRedPlot to structure in the Parallel Coordinates. An example of this would be the use case where clusters can be seen in the dimensionality reduction visualisation. A user can in this case use "manual selection" colouring and choose a colour set. After this, the user can select a cluster in DimRedPlot using the circular brush, select a colour from the drop-down menu, and repeat until all clusters are coloured. The lines in the parallel coordinates will be coloured the same as in DimRedPlot. By giving each cluster a different colour, the clusters can easily be discerned and studied in the parallel coordinates. Figure 31 shows an example where clusters in DimRedPlot have been coloured. The colouring makes it clear how the clusters shown in DimRedPlot relate to clusters in the parallel coordinates.

When dealing with two DimRedPlot instances, the structure in both DimRedPlot instances can be compared the same way as with structure in the parallel coordinates. Simply colouring observations in one of the DimRedPlot instances will apply the colouring in the other DimRedPlot as well. This makes it very easy to compare how structure in one DimRedPlot instance is related to structure in the other instance. An example of usage would be the case where both numerical variables and categorical variables have been dimensionality reduced, and the user wants to find out whether clusters in one of the DimRedPlot instances match clusters in the other instance.

Finally, where colouring can also help is when the observations in a DimRedPlot instance are coloured by variables that were not used to perform the dimensionality reduction. For example, say that clusters can be seen in the scatterplot after DimRedPlot has been initialised with numerical data. An interesting question might be how these clusters correspond to certain categorical variables, as it could be the case that every cluster aligns with a category in a categorical variable. Colouring based on the categorical variable can very easily and quickly show whether such a relationship exists, as the colours of the different categories will also be applied to the observations projected in DimRedPlot.

### 5.2.3 *Selections*

Similarly to colouring, any selection of observations in either DimRedPlot or the parallel coordinates is also synchronised. Selected observations are in both cases coloured the same, either black or white depending on the background colour. Synchronising the selections is most useful in getting a quick overview of the relationship between DimRedPlot and the parallel coordinates. It is for example simple to find out how a cluster in DimRedPlot appears in the parallel coordinates. Although this can also be done using colouring, if only a quick overview is needed, making a simple selection is much faster.

When the dataset that is being visualised is quite big, it is infeasible to show names for every line in the parallel coordinates, which otherwise could be

done as a categorical variable. However, by selecting the observations in the parallel coordinates, the same observations will be selected in DimRedPlot, and the excentric labelling can be used to figure out what the names are of the selected observations.

Selections are also useful to find out more detailed information about individual observations or small groups of observations shown in DimRedPlot. Since selections in DimRedPlot show up in the parallel coordinates, small groups of observations can be selected in DimRedPlot, after which a user can quickly see what the actual values are of those observations on the shown variables. For example, if DimRedPlot contains outliers, they can be selected, which will make it clear which values those outliers have on the parallel coordinates. This information can then be used by the user to explain why the outliers are outliers.

All in all, selections are useful to quickly see the relationship between the dimensionality reduction and the actual data. Furthermore, selections can be used to gain insight on observations in both the parallel coordinates and in DimRedPlot which could otherwise only be gained through just one of the visualisations.

### 5.2.4 *Variable selection*

In Chapter 2 we discussed the fact that many general visualisation techniques, such as parallel coordinates are severely limited in the number of variables they can display. In order to have such visualisations be useful, it is important that users have a good way of selecting which variables are shown based on criteria important to them. In DimRedPlot, this task can be achieved through the selection of variables.

Selecting variables in DimRedPlot can be done through the variable scatterplot, the sizemap, or the bar plots. When variables are selected they are added to the current set of variables shown in the parallel coordinates. This is not permanent, as any new selection of variables will remove the previously added variables and add the newly selected variables. Adding variables dynamically may not always be desirable so it can be turned off in the options menu.

To determine which variables are interesting to add to the parallel coordinates, a user can, for example, look at the variable bar plots within DimRedPlot. As described in Section 4.2, DimRedPlot contains bar plots showing the contributions of variables to the eigenvectors used as axis. Essentially, these bar plots tell us which variables are responsible for the data structure projected onto the eigenvectors.

Which bar plot to use for selecting variables depends on which eigenvectors define the structure that is interesting to the user. If most of the structure is aligned with the $x$-axis' eigenvector, the $x$-bar plot should be used. For structure aligned with the $y$-axis, the $y$-bar plot should be used, and when the structure is not aligned with any of the two eigenvectors individually, the $x + y$-bar plot should be used. Once the user knows which bar plot is needed, he or she can drag the mouse cursor over the bar plot to select the highest bars. This workflow is visualised in Figure 32. The width can depend on several reasons. It could be that the user has a number of variables in mind, in which case it is possible to just select a set-size matching that number. Another reason is that the user wants to select a number of variables cumulatively describing a high enough data-variance. This can be determined by the percentage that appears above the bar plot on selection.

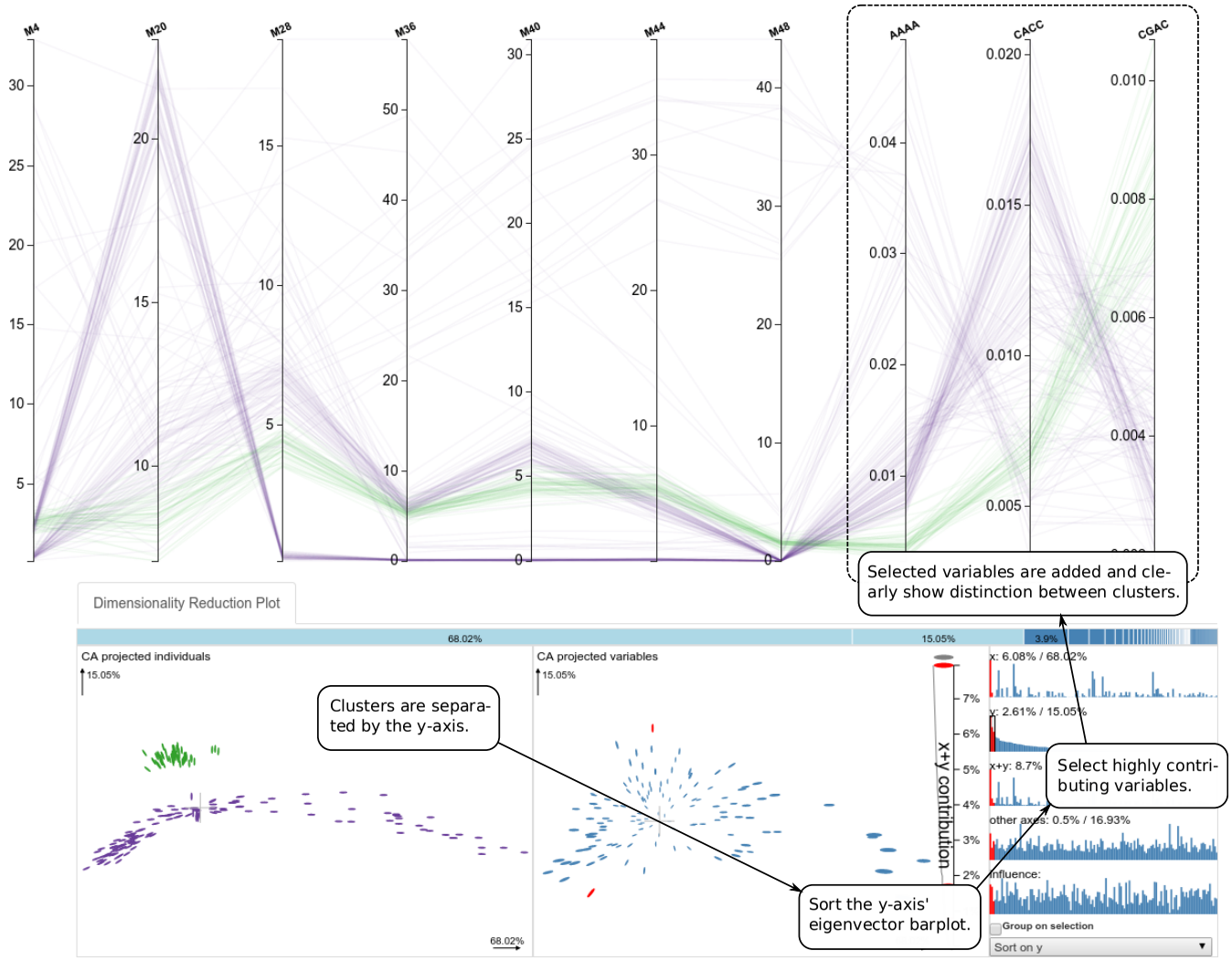


Figure 32: A potential workflow a user could take through RParcoords and DimRedPlot to find out what variables are important for certain observation structures. First, a user finds which eigenvector is defining the interesting structure. After this, the bar plot corresponding to this eigenvector can be sorted and the variables contributing most to the eigenvector can be selected in the bar plot. Finally, after selection the variables are automatically added to the parallel coordinates which confirms the influence the variables have on the observation structure.

Another use case is to use this technique to confirm conclusions drawn from DimRedPlot. For example, say that DimRedPlot displays three clusters, which are mostly split by the second eigenvector, as is the case in Figure 32. To find out which variables are responsible for these cluster we can examine the contribution bar plot of the second eigenvector. By selecting the variables with the highest contributions to this eigenvector, the variables will be shown on the parallel coordinates. Now a user can easily confirm that the selected variables are in fact responsible for the clusters by selecting the clusters and looking at the added variables in the parallel coordinates. The clusters should be easily distinguishable from each other across the selected variables. This distinction can also clearly be seen in the final step in Figure 32.

### 5.2.5 *Iterative dimensionality reduction*

Sometimes, when a user performs dimensionality reduction, the resulting DimRedPlot instance gives the user all the information he or she wants and nothing more has to be done. However, often this is not the case and dimensionality reduction needs to be performed iteratively until the desired results are obtained. There are two ways in which RParcoords supports this.

SELECTING VARIABLES: Whenever a selection of variables is made using DimRedPlot, the variable selection list in the option menu, used for determining the variables to be analysed using PCA, CA, or MCA, is updated to reflect this selection. All the previous selections in this list are removed and the current selection is added. This allows the user to make a selection of variables, press the reduce button, and dimensionality reduction will be performed on the selection of variables that is made.

Redoing dimensionality reduction on a user selection is useful for several reasons. One reason is that a user may simply be interesting in a smaller set of variables based on domain knowledge. In this case the first dimensionality reduction acts to give more of a global overview, after which the user can zoom in on a subset of the data. There is no clear flow to be described for this use case, as it is highly depended on the dataset being used and the use case associated with the dataset.

Another reason for redoing dimensionality reduction is to simplify the analysis. For example, say that two separate clusters can be distinguished on the first two eigenvectors. The user can in this case select the variables with a high value in the $x + y$ contribution bar plot, which will be responsible for the clustering, and redo dimensionality reduction. This will simplify the analysis as there will be less generated eigenvectors, and it also means that the clusters might be a bit more refined since noise from other variables is removed. Decreasing the number of variables to study will also simplify any further analyses. For example, a smaller set of variables may fit on the parallel coordinates, whereas the larger set did not. The use case here is very similar in nature compared to the one Rauber et al. [30] focus on. They use several potential metrics, instead of just contribution, to score variables and based on these scores select a smaller set of variables. This process is done iteratively until a sufficiently small but accurate selection of variables is obtained. The reduced number of variables makes it easier to do further analysis or classification using the data.

FILTERING OBSERVATIONS: Using the filtering described in Section 5.1.2 dimensionality reduction can easily be redone on the observations remaining after the filter operation. After filtering, the user can press the reduce button, resulting in dimensionality reduction to be redone on the filtered data.

A potential problem of techniques such as PCA is that if there are extreme outliers in the data, the variance in the rest of the data does not always show on the first few eigenvectors. In this case, the results are dominated by the outliers. Although it is possible for a user to change the eigenvectors to find the variance in the rest of the data, this can be a lot of work and there is no guarantee that the variance in the data the user is looking for is aligned with just one or two eigenvectors. Instead, a user can brush the outliers using the circular brush in the observation scatterplot and press "remove selected" in the options to filter the outliers out of the data. After removing the outliers the user can redo dimensionality reduction and study the structure in the rest of the data.

More generally we can say that to find out how a subset of observations projected onto the first eigenvectors is structured, there are two things a user can do. A user can either look through the other eigenvectors or select the subset of observations he or she is interested in, press keep selected, and redo dimensionality reduction.

Finally, the user may have a good reason to redo dimensionality reduction because of the use case and dataset. For example, say that ten different plants have each been measured ten times for specific chemicals, resulting in 100 observations. A user might be interested in seeing PCA being done on all the observations, but also on the observations belonging to a certain plant or to a certain time-point. In this case, the same procedure can be used regarding filtering and redoing dimensionality reduction.

## 5.3 IMPLEMENTATION

As the name RParcoords suggests, the backend of the visualisation is written using the R programming language [36]. The frontend is written using JavaScript and AngularJS. The parallel coordinates visualisation uses an existing D3 parallel coordinates implementation [37]. To have the JavaScript communicate with the R backend, OpenCPU [38] has been used. OpenCPU offers a RESTful API that allows any application to call R functions and retrieve the result as a JSON structure.

The R backend is responsible for several tasks. It stores and retrieves the used datasets, which is done using SQLite. Dimensionality reduction is performed using FactoMineR [39] for PCA and CA, and the `ca` package [40] is used to perform MCA. Finally, clustering is also performed in the R backend.

The fact that RParcoords has been written using JavaScript makes deployment to the end-users easy, since all the end-users need is a webbrowser and an address to the webserver serving RParcoords through OpenCPU.

## 5.4 DISCUSSION

By combining DimRedPlot with the larger visualisation tool RParcoords, many user interaction that were not that useful in DimRedPlot as standalone tool have become very useful. Interactions such as observation selection and colouring in both the parallel coordinates and DimRedPlot can offer extra insight into the projections generated by dimensionality reduction and also into the original data itself. Using multiple DimRedPlot instances and allowing interaction between them allows users to quickly and easily see the relationship between categorical and numerical dimensionality reduced data. Furthermore, selecting variables in DimRedPlot makes it possible for users to utilise the parallel coordinates much more effectively because of the greater control on the displayed variables. Finally, the easiness of iteratively performing dimensionality reduction should make analyses flexible, as no new complicated setup is needed to obtain new results.

Of course, the question remains how useful actual users will experience the many features discussed in this chapter and the previous chapter on DimRedPlot. As such, the next chapter discusses evaluations performed with users in order to discuss the actual usefulness of the features shown.

# 6 EVALUATION

To evaluate the developed visualisation and interactions, several researchers at the Luxembourg institute of Science and Technology, or LIST, have used the visualisation with data of their own. This gave us the chance to evaluate RParcoords and DimRedPlot using actual use cases and datasets in a setting for which they are designed to be used.

Two datasets have been studied using RParcoords and DimRedPlot, resulting in two evaluations. Between the two evaluations there is a big difference in scope, use cases, and results. The first evaluation concerns vineyards in Luxembourg, and the second evaluation concerns biogas production and contig binning. The evaluations are structured as workflows that start at the start-up of the visualisation and end with the users obtaining some result or coming to some conclusion.

## 6.1 VINEYARDS IN LUXEMBOURG

At the LIST, researchers are studying the properties of several vineyards and of the wine produced there. In total 23 vineyards have been studied. 21 of the vineyards are located in Luxembourg along the Moselle river and 2 are located in Germany and have been added as a control group. The vineyards are located in different regions, called terroirs in the context of vineyards. These terroirs are distinguishable by different environmental properties, such as the soil and the climate. Measurements have been taken at the 21 Luxembourgish vineyards and the 2 German vineyards. The measurements include properties of the soil, properties of the wine, and other properties such as which plant the plants are cloned from or how much space each plant has.

The goal of the research is to find out whether wines originating from different terroirs can be distinguished based on their chemical properties and taste. At the time of the evaluation, the dataset was not complete, as some chemical data regarding the wine was still missing and the wines had not been tasted yet. However, the dataset was complete enough to perform an exploratory analysis, verify assumptions underlying the research, and to test some initial hypotheses.

As we are trying to find out if one group of variables influences another, we can group the variables into independent and dependent variables. The dependent variables are those describing properties of the wine, while the independent variables describe properties that might influence the wine. The independent variables are the variables describing the soil, but also other variables such as the ones describing what clones the plants are and how much space the plants have. These other variables are also called covariates.

A problem that this dataset faces is the fact that not all variables are of the same type, some are numerical while others are categorical. The dependent variables are all numerical, while the independent variables are both numerical and categorical. Since PCA can only analyse numerical data and MCA can only analyse categorical data, the two techniques have to be combined in order to study the dataset.

The strategy the researchers normally use to study the dataset is divided up into the following four steps:

1. Determine whether the different terroirs can be distinguished.
2. Determine whether there are different distinguishable wines.

3. Find a link between different terroirs and different wines.
4. Find out if covariates are influencing the different wines.

It is known that the vineyards are located on different terroirs. If the first step fails, this means that even though different terroirs do exist, the data is apparently not sufficient enough to show the distinction between these terroirs. If, according to the data, no different terroirs exist, they can not possibly be linked to different wines.

Since the question is whether the terroirs influence the wine, there needs to be a difference in the wines. If the second step fails and the wines are indistinguishable, then it is impossible to determine whether something is influencing the wines.

If there are both different terroirs and at the same time there is a difference in the wines, the next question would be if the different terroirs are influencing this difference in wines. This is what the third step tries to figure out, and it would answer the main question of the study.

Finally, it is also interesting, especially if the previous steps have weak results, to see if the covariates are influencing the different wines. It could be that things like who grows the wine, which clone the plants are, or how much space a plant has influences the chemical properties of the wine more than the soil the plants are grown on.

### 6.1.1 *Evaluation setup*

Four people were present during the evaluation. Two of the attendees were researchers working on the vineyard research, while we were the other two attendees as developers of RParcoords and DimRedPlot. Fitting four people behind one monitor is a bit hard, so a beamer setup was used instead. RParcoords was shown on a beamer screen with the two researchers standing next to it. We controlled the visualisation through a connected computer and made notes of the evaluation. During the evaluation, the researchers asked for certain actions to be performed in the visualisation, while we every now and then gave suggestions of useful actions that could be performed using RParcoords. It would have been possible to also have let the researchers control the visualisation, but the researchers had limited to no previous exposure to RParcoords. Controlling the visualisation ourself based on instructions from the researchers allowed us to slowly show how RParcoords works and what actions are possible, without spending a large amount of time explaining every aspect of RParcoords and DimRedPlot.

The evaluation itself consisted of performing the previously mentioned four steps on the provided dataset. A detailed explanation of the four steps and the findings that were made by the researchers are detailed in the following sections.

### 6.1.2 *Distinguishing the terroirs*

In the region that the vineyards are located, two major geological soil types or terroirs exist, muschelkalk and keuper. Figure 33 shows a map with the studied Luxembourgish vineyards and the terroirs that they are in. It should be the case that the variables describing the soil form a structure in which the two geological types can be distinguished. Should this not be the case, it will be impossible to find a link between the wines and the terroirs they are growing in.

To answer the question whether the terroirs can be distinguished, MCA was performed on the soil variables. Not all categorical variables are soil variables, so a selection had to be made before MCA could be performed.

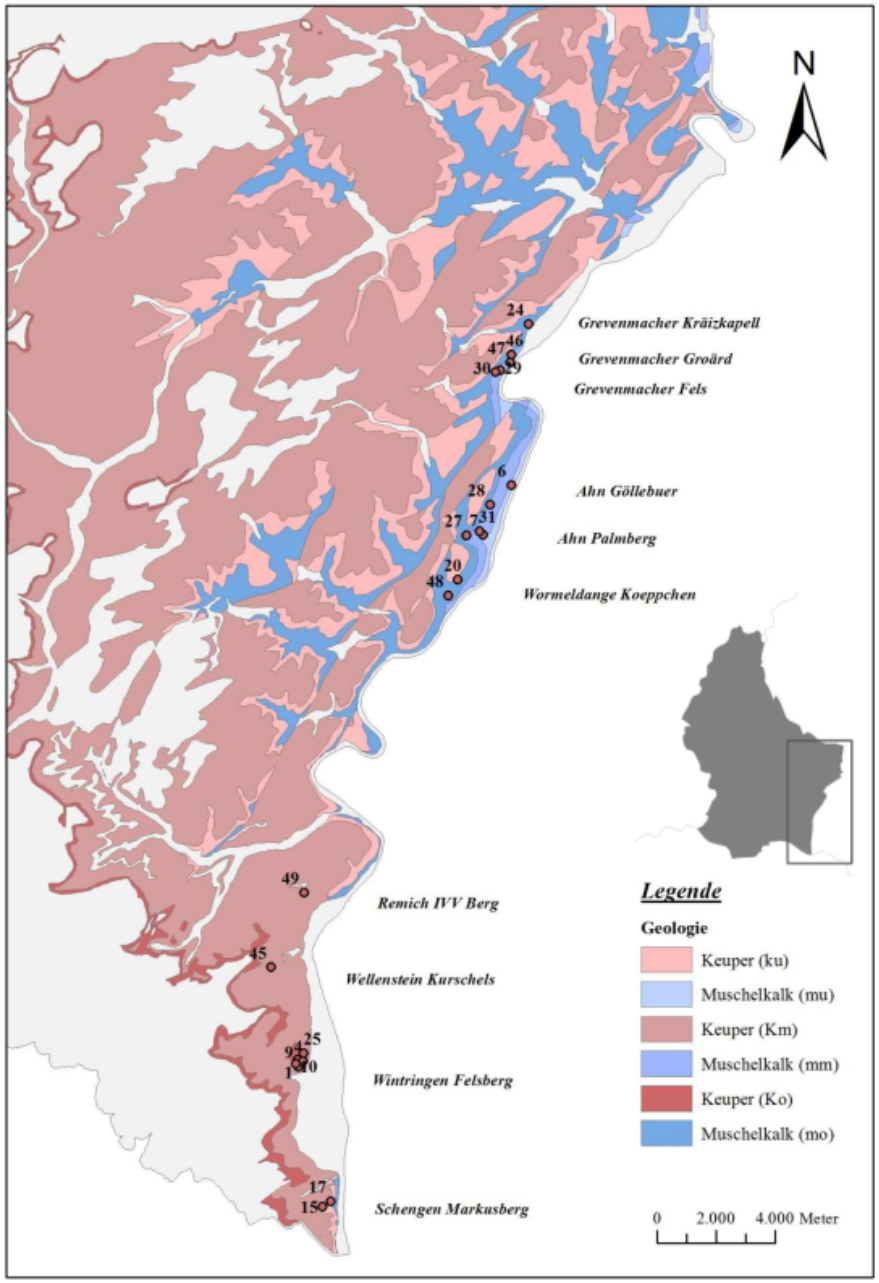


Figure 33: Location and geology of the studied vineyards in Luxembourg. Source: Schumacher [41].

The initial results showed a cluster of points with three distinct outliers, seen in Figure 34A. Using the circular mouse brush, it was found that two of the outliers were in fact the two vineyards from Germany put in as a control group. These vineyards have been marked with a red "Germany" in the image. As the soil type of these vineyards is quite different they are shown as outliers. As described in Section 5.2.5, outliers can mess up dimensionality reduction by hiding other structure in the data. As such, the German outliers were selected and filtered out, and the MCA was redone, resulting in Figure 34B.

The second MCA showed two large clusters and three outliers. The researchers wanted to know how the clusters matched up with the different terroirs. According to the researchers the information about the terroir a vineyard is in was stored in the geology variable, which is a categorical

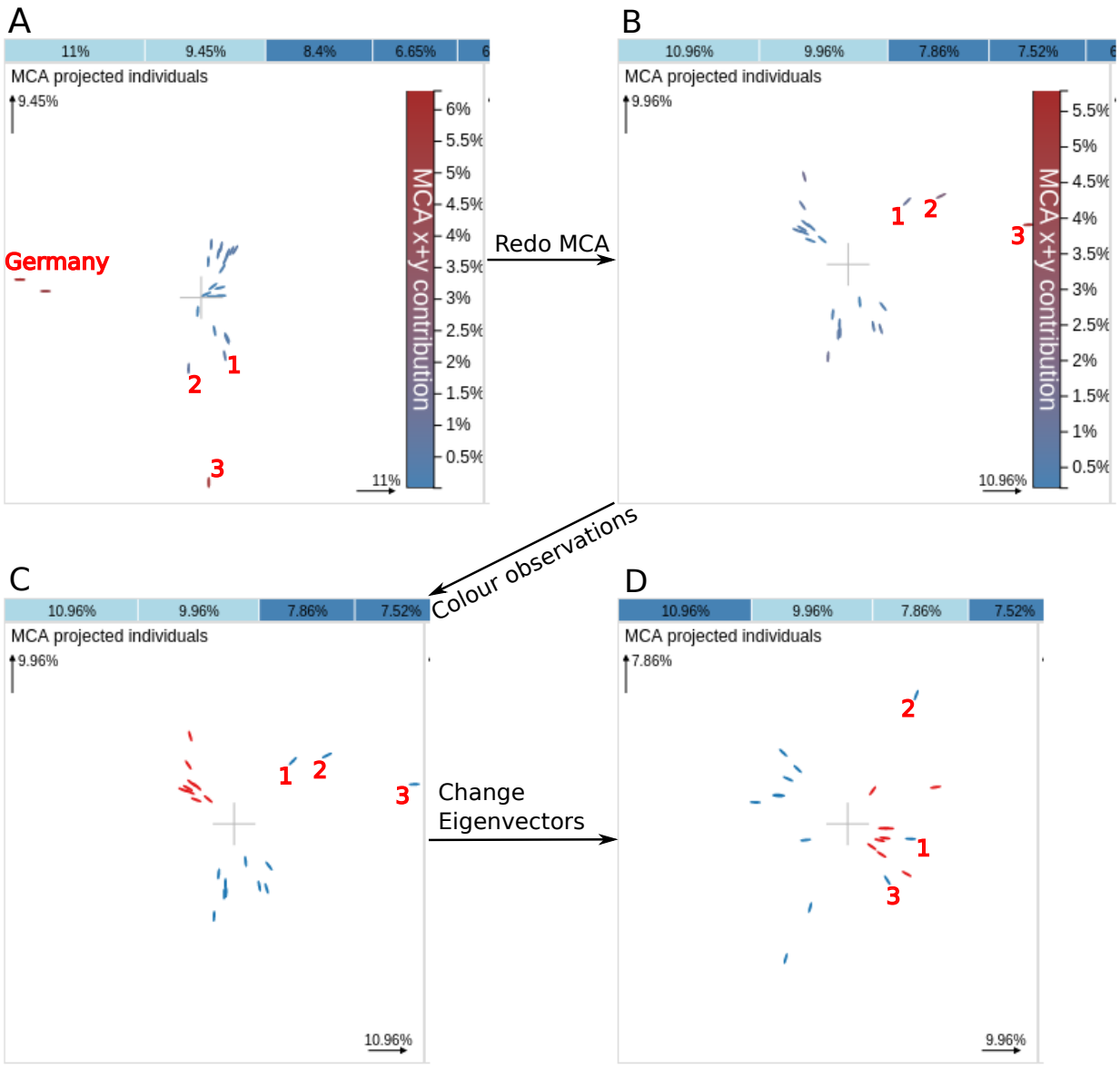


Figure 34: Four stages of analysis. (A) Initial MCA on all observations. The outliers marked with the red 1 and 2 are positionally between the two main terroirs. The outlier marked with the red 3 has imported soil. (B) Outliers from Germany, tagged in A with a red "Germany", have been removed. (C) Observations are coloured on geology. (D) The eigenvectors used as projection axis have been changed.

variable with the different terroirs as categories. As such, we suggested to use colouring based on the geology variable. To colour the points, the geology variable was added to the parallel coordinates and the different categories were selected and coloured from there using the manual colouring option. Figure 34C shows this colouring, and it can clearly be seen that the two clusters separate the muschelkalk vineyards, blue, from the keuper vineyards, red. The three outliers, marked throughout the images with a 1, 2, and 3, could also be explained by the researchers in the same terms. The two outliers marked 1 and 2 are actually positionally between the muschelkalk and the keuper regions in, and the outlier marked 3 uses soil that has been imported from different regions over the years.

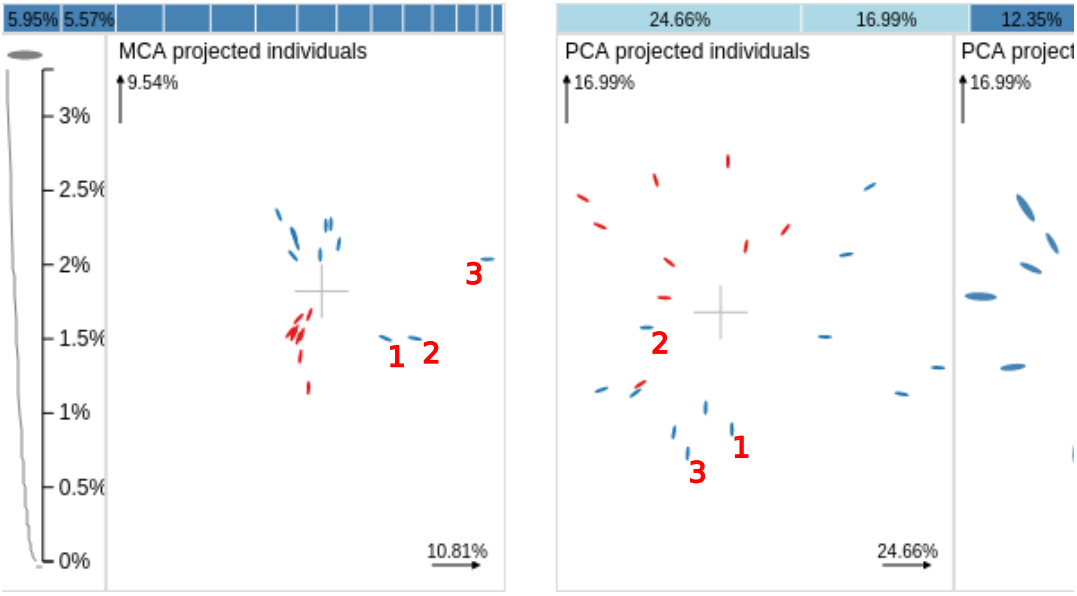


Figure 35: MCA on the categorical terroir variables next to PCA on the numerical terroir variables. The Luxembourgish outliers have again been marked with a red 1, 2, and 3.

At this point it was noted that the geology variable was actually part of the MCA, which could influence the analysis. However, after removing it and redoing MCA the resulting scatterplot looked the same. The amount of data-variance described by the first two eigenvectors was a little less than 20%, which is not a lot. To see if the divide between the terroirs was also present on other eigenvectors, the eigenvectors were rotated through, as described in Section 4.1.2, and it was found that the third eigenvector shows the same structural divide. Projecting the points onto the other eigenvectors showed a mixing of the two geological groups. In total the structure was visible in a little less than 30% of the data-variance.

Finally, although the categorical soil variable showed that it was possible to distinguish between the different terroirs, there were also some numerical soil variables. Because it would be interesting to see if the same could be said here, PCA was performed on the numerical soil variables. As RParcoords supports multiple DimRedPlot instances next to each other it was easy to link the numerical soil variables with the categorical soil variables. The result can be seen in Figure 35. In the PCA plot, the same separation of terroirs could be seen, albeit with some minor miss-classifications. This helped solidify the researchers' hypothesis that the different terroirs are distinguishable using the measured soil variables.

### 6.1.3 *Distinguishing wines and linking to terroirs*

After having found that the terroirs can be distinguished using the measured soil properties, we had a look at the variables describing the wines. All the wine variables are numerical so PCA was performed on them and the results replaced the current PCA results. Unlike the MCA results for soil variables, no clear structure could be seen in the wine variables, which made the researchers wonder whether the addition of the then missing variables would change this.

The PCA results did show a divide between the two geological soil types; however, the separation was not perfect. The top-right part of the plot only

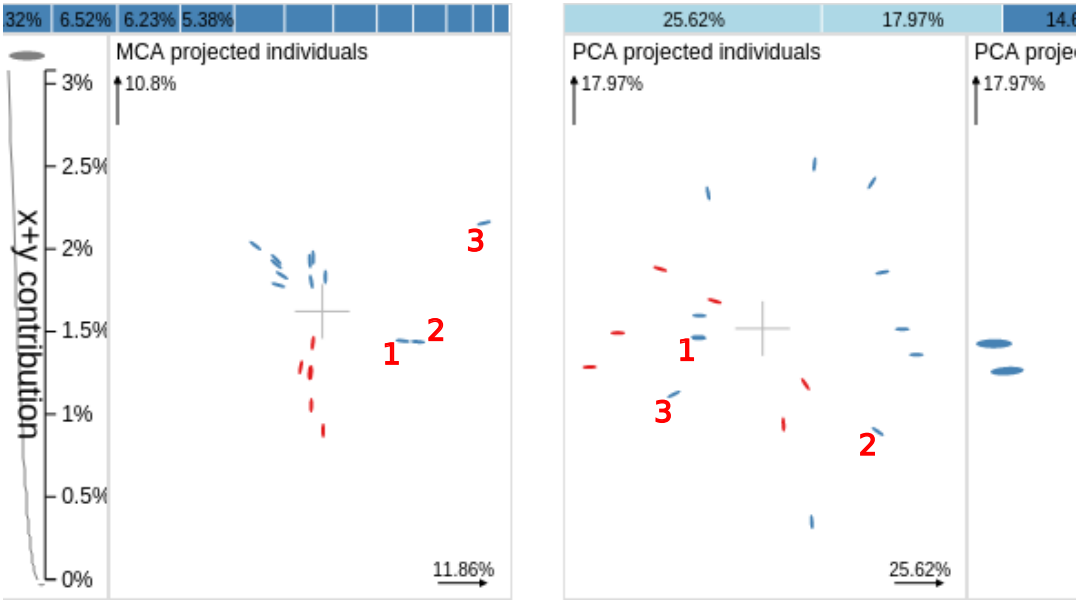


Figure 36: MCA on the terroir variables next to PCA on the wine variables. The Luxembourgish outliers have again been marked with a red 1, 2, and 3.

showed blue points, while the bottom-left part of the plot showed more of a mix between red and blue points, as seen in Figure 36.

In order to get a better idea of the structure in this part of the data, we suggested that the variables in the PCA plot be selected to show them on the parallel coordinates. Next, the variables were ordered on average value, as described in Section 5.1.7. The resulting parallel coordinates can be seen in Figure 37. The figure quite noticeably shows that one line through the parallel coordinates is thicker than the rest. This is the result of the fact that that line consists of three vineyards with the same values. Had transparency not been used in the parallel coordinates, this would have been overlooked. By selecting the lines in the parallel coordinates we could see their names in the PCA DimRedPlot instance. Based on these names, we knew that these vineyards were given averages as values, as the grapes in these vineyards had been harvested before measurements could be taken on them. These vineyards were filtered out and PCA and MCA were redone. However, the results were not changed significantly because of this.

### 6.1.4 *Influence of covariates*

Seeing the slightly disappointing results of the previous step, the researchers wanted to know what the influence on the wines was of the covariate variables. Unfortunately, the covariates are part categorical and part numerical. As such, there is little point in performing MCA, and performing PCA would overwrite the current PCA DimRedPlot instance. Instead, the covariates were selected in the options menu to be shown on the parallel coordinates.

The two red outliers at the bottom of the PCA plot in Figure 36 were selected to see which values they have on‘ the covariates. The selection could be seen as black lines on the parallel coordinates, and it showed that both points were similar in the amount of distance between the plants and the yield of the plants. This may suggest that sun-exposure has an influence on the chemicals, which leads to similar properties of wines in different geologies.

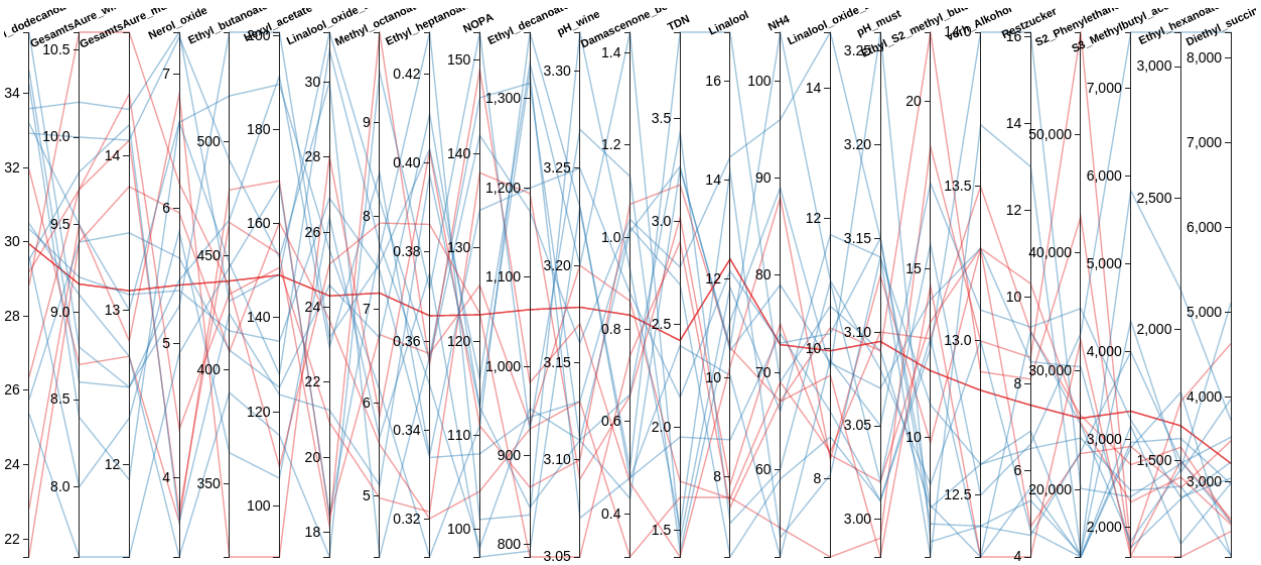


Figure 37: Dependent variables as parallel coordinates. The variables have been ordered on average value. The thick red line are three observations with the same average data values

### 6.1.5 *Final results and remarks*

The total session took about 2 hours and the results of the analysis were useful to the researcher to get an idea of how their data is structured and what needs to be done to advance with their research. The notes taken during the evaluation will be used to form a report on the progress of the research.

The reaction of the researchers to RParcoords and DimRedPlot was very positive. The tools made investigating the vineyards data easy and fast. It was especially noted that the tools made the exploring of relationships between numerical variables and categorical variables easy, as this can otherwise be a hard thing to do.

The following is a quote from one of the researchers regarding RParcoords and DimRedPlot: "The tool helped us by allowing a quick screen, which data are worth a closer look and which ones are not. I really appreciated that no re-arrangement of the data was needed to check multiple hypothesis in sequence. Furthermore, methods that are separated in other tools were combined, and objects and variables were traceable in different plots / across different analyses. Compared to other tools it is more interactive, flexible, and visually driven. My feeling was that persons who are at least superficially familiar with PCA and MCA should not have much difficulties understanding the visualisation.

There were some possible improvements I could think of. From time to time I wondered if the parallel coordinates are really superior over a multiple scatter plot approach. I could imagine that replacing the parallel coordinates with multiple scatter plots could enhance the intuitive comprehension of the information in complex data sets, but I fear that conventional monitors are too small to have all information on the same screen. So far, I have no idea how difficult it is to upload data, because you did it. Allowing easy upload of data by users is crucial for wide use. Finally, Some kind of statistical test that gives the user an idea if a specific sub-group of data is over- or under-represented in another sub-group would be useful and valuable for cases where the distinction / grouping of objects or variables is not perfect and overlaps are found. Maybe Chi-square?"

## 6.2 DNA CONTIG BINNING

The research in the second evaluation concerns biogas production. Biogas is gas produced by feeding bacteria organic matter, such as food waste and manure. Anaerobic digestion within the bacteria turns the organic matter into a mixture of different gases such as methane and $CO_2$. For this research several biogas reactors have been set-up containing an estimated 10000 to 30000 different bacteria, many of which are low-abundant. The researchers want to find out what the most prominent bacteria are and what there function is in the process of producing methane.

At different times during the experiment samples of bacteria were taken from the reactors and their genomes were analysed using shotgun sequencing, resulting in many strands of smaller DNA sequences. By looking at overlap in the different sequences, existing software tries to combine these small DNA sequences into larger sequences or contigs. Unfortunately, combining the sequences is a combinatorial problem and it can not be solved reasonably to obtain full DNA strands. In order to still find out what bacteria were present in the reactors, the individual contigs have to be grouped together some other way in order to form a complete genome, which is also known as contig binning. To perform the contig binning, RParcoords is used.

The dataset resulting from the shotgun sequencing contains approximately 30000 contigs as observations. The dataset has a variable dictating the length of each contig and a variable with counts of the number of occurrences of `CG`'s in the contigs. Beyond this there are 7 variables describing relative contig abundance levels. These abundance levels indicate the number of contigs that were present in the containers with every variable being at a different timepoint. Finally, the dataset contains 128 variables that contain the normalised frequency of four-letter DNA combinations, or tetranucleotides, such as ATCG and AAAT, in the contigs. These variables combine structural and behavioural information about the genomes of bacteria in the reactors.

The dataset generated by this research had already been studied for quite some time before this evaluation took place. One way this was done is by looking at the abundance levels. If contigs belong to the same bacterium their growth or decline in abundance over time should be similar. To study this growth and decline, the parallel coordinates in RParcoords are used. Parallel coordinates allow for showing the different abundance variables next to each other, making the different abundance flows very apparent. An example of this can be seen in Figure 38, which shows some easily spotted abundance flows in RParcoords.

Unfortunately, parallel coordinates alone are not enough to find the different bacterial genomes. To further support this task, k-means clustering and correlation clustering has been used on the abundance variables. When clustering is used on the abundance variables, every cluster is likely to contain a selection of contigs with a similar abundance flow, but with a distinct abundance flow when compared to other clusters.

Just like abundance variables, the tetranucleotides frequency variables can also be used to distinguish different genomes from each other. The contigs of one genome generally have similar tetranucleotide frequencies. This distinction is however not as clear as with the abundance variables. Before the addition of DimRedPlot to RParcoords, the tetranucleotides variables had only been used a little bit in order to distinguish different genomes from each other.

Finally, selections of contigs are also tested for the presence of essential copy genes. Essential copy genes are a set of 107 genes that are the bare minimum a bacterium needs to be viable. Most bacteria only have one of each of these genes in their genome. This knowledge can be used to find out if a selection of contigs belongs to the genome of one bacterium. The

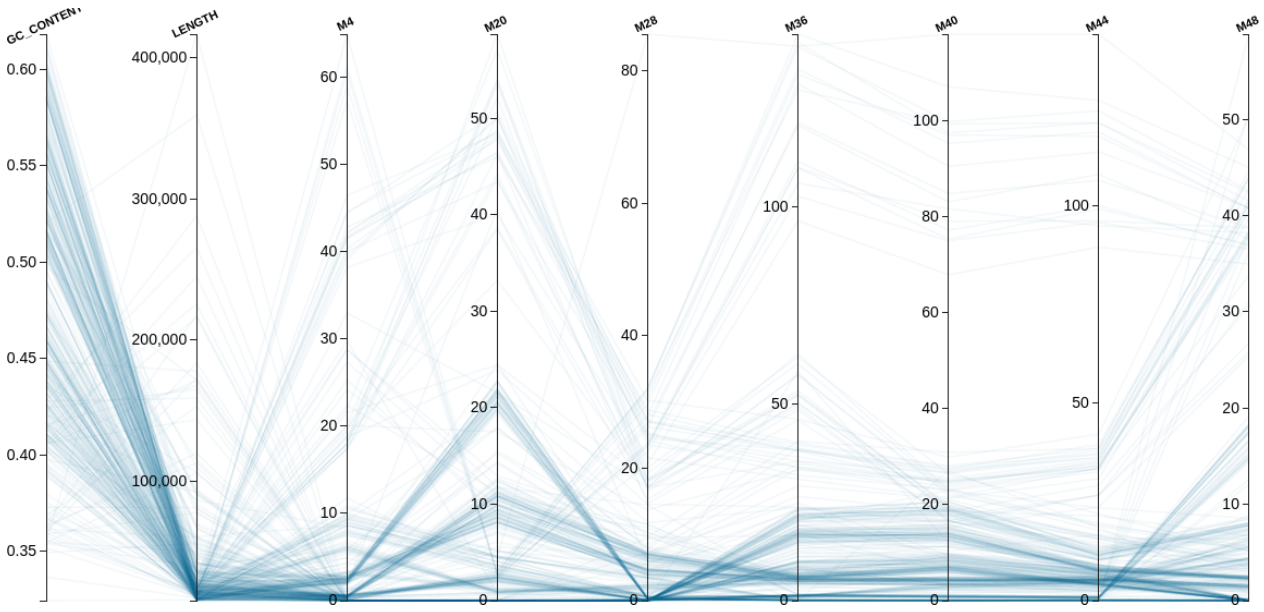


Figure 38: Parallel coordinates showing contigs following different flows through the abundance variables.

selections should have a sufficient number of essential copy genes and it should not contain many or any duplicates of those genes.

### 6.2.1 *Evaluation setup*

For this evaluation, two persons were present. The first attendee was a researcher working on the described project, while the second attendee was one of the developers of DimRedPlot and RParcoords. As mentioned, the researchers working on the described research had already been working with RParcoords before the evaluation. As a result, RParcoords and DimRedPlot were controlled by the researcher, while the developer took notes and every now and then gave the researcher some directions on the usage of DimRedPlot, as DimRedPlot had not been used before by the researcher. However, little explanation was needed for the researcher to get a good grasp of how DimRedPlot worked.

Because the dataset was already extensively studied, the evaluation mostly involved using the tetranucleotides frequencies to solidify previously made conclusions. For example, if a selection of contigs is thought to belong to one genome but it has two distinct distributions of tetranucleotides frequencies, the selection is probably made out of two genomes instead of one. During the evaluation multiple existing conclusions or open questions have been addressed using the addition of DimRedPlot to RParcoords. Because the tetranucleotides variables are frequencies, CA was used as dimensionality reduction technique on them.

### 6.2.2 *One selection might be two genomes*

The first case looked at in the evaluation concerned a selection of contigs that formed one cluster in the correlation clustering, but two in the k-means clustering. The selected contigs looked very similar in the parallel coordinates, but the k-means clusters had some overlap in essential copy genes, which created the suspicion that there must be more than one genome in the selection.

The first thing to do was to find the selection of contigs again, which was done through showing the clusterings as variables in the parallel coordinates and selecting the clusters in question. After making the selection and filtering the rest of the contigs out of the visualised dataset, CA was performed, which resulted in the contigs being projected as one big cluster without any distinct structure. At first glance, this suggested that the selection was in fact just one genome. There were two outlier contigs present in the projection, which were removed to make sure they were not hiding any structure, as described in Section 5.2.5. However, redoing CA yielded the same results as before.

To further investigate the relation between the projected structure of the contigs' tetranucleotide frequencies and the contigs in the k-means clusters, the individual k-means clusters were selected and manually given different colours. As a result the contigs in DimRedPlot were also coloured, which can be seen in Figure 39. As the image shows, both k-means clusters show up as different clusters in the projection. Although both clusters are occupying the same area, one of the clusters is much denser and limited to one area, while the other cluster is much wider. This showed that the two groups of contigs were in fact different when it comes to their tetranucleotides frequency distributions, supporting the original suspicion that there were in fact two genomes. As a result of this additional analysis, the researcher could conclude that the contigs form two genomes of new bacteria from the phylum Firmicutes.

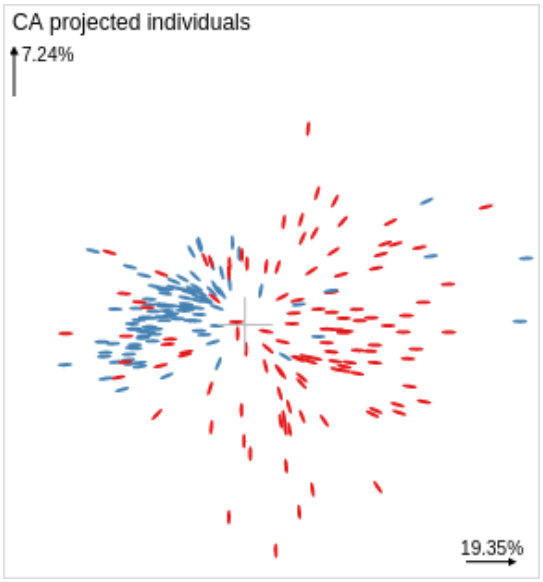


Figure 39: Contigs shown in DimRedPlot. Uncoloured the contigs look like one big cluster. But when colouring is applied, it is clear that there are two different clusters with different means and standard deviations.

### 6.2.3 *A selection should be one genome*

In the second case, the domain expert had a selection that contained 105 essential copy genes out of a potentially total of 107. This, and the fact that the contigs behaved similarly in the parallel coordinates, made the selection highly likely to be a new genome. However, the clustering divided the selection into three parts for which the researcher wanted to find an explanation.

After filtering out the irrelevant contigs and performing CA, some structure could definitely be found, as seen in Figure 40. In the image, the different

contigs are not clustered together but seem to be spread out over two or three different clusters lying close together. The domain expert used the k-means clusters to refine the selection. Upon redoing CA, it shows one outlier which had been marked as outlier in the parallel coordinates as well.

The researchers selected some of the clusters in DimRedPlot and found one cluster of contigs that made a narrow selection in the parallel coordinates. This selection was also quite complete based on the number of essential copy genes it had. The selection can be seen selected in Figure 40 in black. The genome it forms is of a new bacterium from a candidate division WWE1.

### 6.2.4 *Many low abundant contigs*

A large part of the contigs in the dataset are low abundant and belong to a large set of bacteria. This means that there are only a couple of contigs for each bacterium. The domain expert was wondering whether CA could be used to distinguish the different bacteria is this large set of contigs.

Upon filtering out the rest of the contigs and performing CA no contigs were projected. There were too many contigs to be visualised due to limitations in the HTML, as described is Section 4.3. Making a smaller selection of contigs resulted in a CA that simply shows one big cluster with all contigs in it. Perhaps there were many different clusters of contigs hidden inside this big cluster; however, there were too many different bacteria to see any structure in the projection.

### 6.2.5 *Study a selection with duplicated essential genes*

A previously made selection had a pretty complete set of essential copy genes, 98 of 107. However, 3 genes were duplicated, meaning that out of those 98 genes, only 95 were unique. Duplicate genes are not impossible to occur, but it might be that studying the tetranucleotides will show that they are outliers. The selection consisted of contigs that were both in a certain correlation cluster and a certain k-means cluster.

For completeness, the domain expert kept all the contigs in the correlation cluster and performed CA on it. The projected contigs were structured as a big cluster with many outliers. The outliers were removed and CA was redone. The result is shown in Figure 41. In the image, the contigs from the mentioned k-means cluster have been selected and are coloured black. As we can see, the big cluster did not only contain contigs from the k-means cluster, but also from other k-means clusters. Also, some of the outliers were part of the original k-means cluster. This suggests that some of the contigs believed to be part of the selection are not part of it, while others might be part of it.

To find out where the three duplicated essential copy genes resided in the projection, the highlight function was used to highlight them in the parallel coordinates. Brushing them in the parallel coordinates showed that they were in fact in the big cluster in DimRedPlot, indicating that they have similar tetranucleotides frequencies compared to the rest of the selection. Based on this evidence, the domain expert concluded that this particular bacterium happens to be one with several duplicate essential copy genes.

### 6.2.6 *A selection may have to be extended*

Finally, we looked at a rough selection of contigs that was not finished. The domain expert suspected that there were more contigs that needed to be added to the selection. The selection in its then current form was made out of contigs that were in a certain correlation cluster and in one of a total of two k-means clusters.

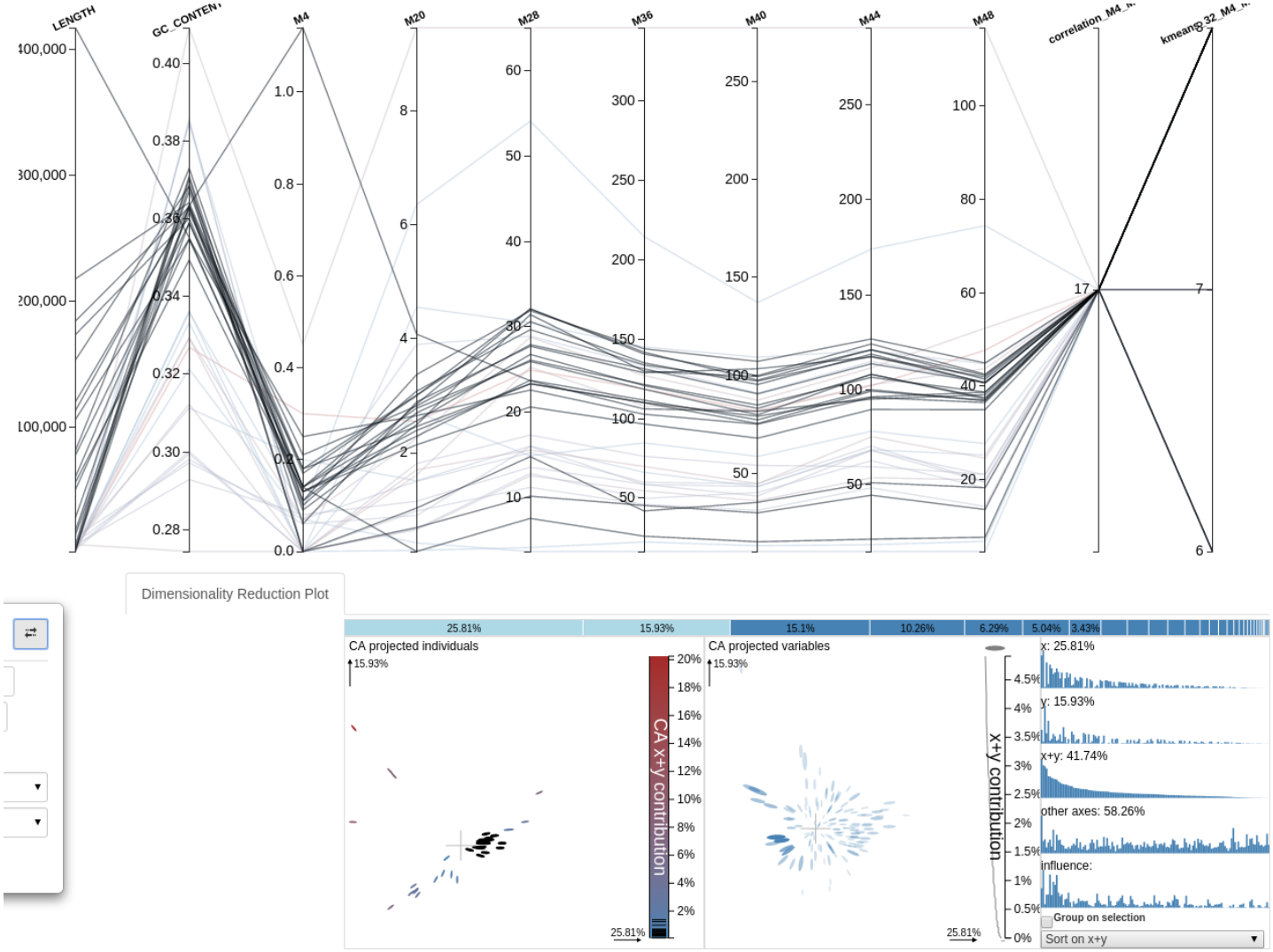


Figure 40: A selection in RParcoords shows a spread out set of contigs that are all approximately following the same growth and decline in abundance over time. DimRedPlot shows that there are different tetranucleotide structures within the selection. One cluster in DimRedPlot has been selected and in RParcoords this selection turns out to be quite narrow.

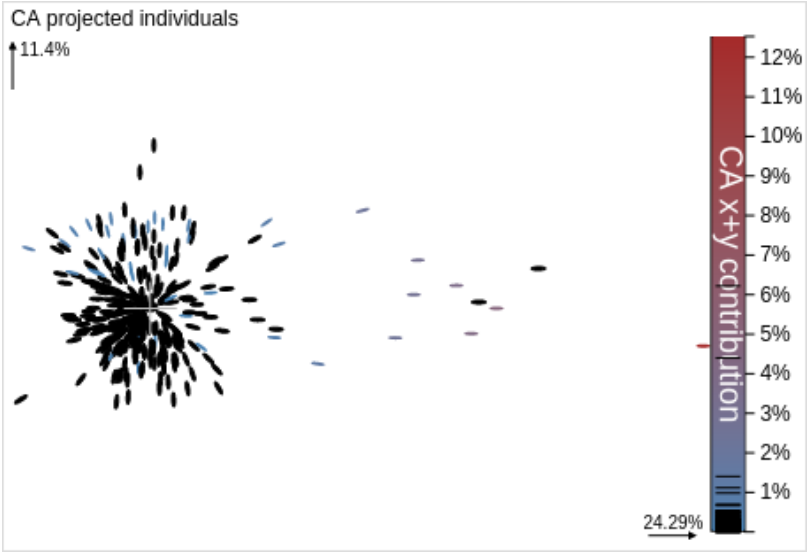


Figure 41: Contigs shown in the DimRedPlot. The big cluster on the left does not correspond to the contigs in the k-means cluster, which have been coloured black.

The domain expert started by selecting all contigs in the two k-means clusters using the parallel coordinates. Next some outliers were removed in the parallel coordinates and CA was performed. The resulting CA showed a single cluster. To study the k-means clusters, both clusters were coloured manually and both clusters turned out to have approximately the same mean and variance in the projection. However, when the two k-means clusters were individually selected in the parallel coordinates they showed slightly differing patterns, especially in the first few measuring points. This can be seen in the parallel coordinates in Figure 42 by looking at the variables M4 to M48. Purely looking at the parallel coordinates, the difference in the abundance at the first few measuring points initially caused some uncertainty about the selection being just one bacterium. However, the addition of DimRedPlot solidified the hypothesis that the selection was in fact one genome.

To make this similarity in the tetranucleotides frequencies more visible, the domain expert used the contribution bar plots to select the tetranucleotides with the highest contribution to the first two eigenvectors, as shown in the DimRedPlot instance in Figure 42. This displayed the selected tetranucleotides as variables on the parallel coordinates, as can also be seen in the figure, which showed both k-means clusters to have a similar pattern along these variables.

To further study the contigs in both k-means clusters the domain expert used a separate tool to check the contigs of the two clusters for essential copy genes. The first cluster contained 86 non duplicated essential copy genes, which is a strong indicator that the contigs belong to a single bacterium. The second cluster contained 9 non duplicated essential copy genes which were not present in the earlier 86 essential copy genes of the first cluster. This is a strong indication that the, in total, 95 essential copy genes belong to a single bacterium. Using this information, the researcher concluded the genome to be a new bacteroidetes.

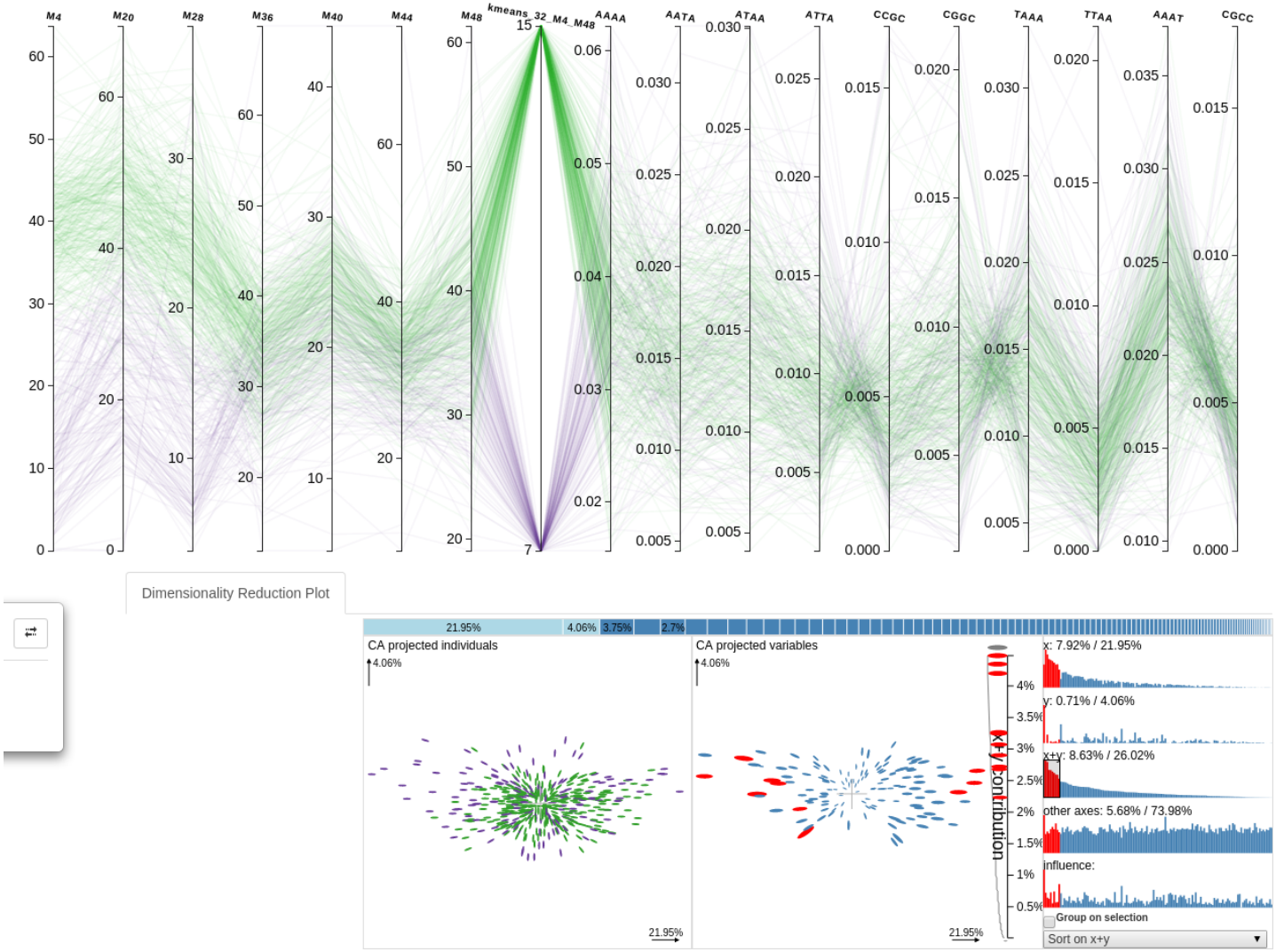


Figure 42: The contigs in two k-means clusters are shown in RParcoords. The contigs in the two clusters are coloured respectively green and purple. The abundance levels show that at first the two clusters are different (M4 to M28) but after that the two clusters are similar (M36 to M48). The DimRedPlot instance shows similar tetranucleotide frequencies for both clusters, as is verified by showing the highly contributing tetranucleotides on the parallel coordinates.

### 6.2.7 *Final results and remarks*

Although the research talked about was already in full swing and was already using RParcoords, the addition of DimRedPlot proofed to be a great help in identifying and discovering bacteria. The interaction between the parallel coordinates and the results from CA allowed the researcher to easily connect the CA results to the actual data in the parallel coordinates.

This evaluation was the first time the domain expert worked with DimRedPlot, but working with the visualisation turned out to be easy after some instruction. The evaluation in total took about 2 hours to complete. The domain expert at some point during the evaluation mentioned that DimRedPlot would have saved a lot of time had it been accessible earlier.

The researcher we worked with provided us with the following feedback: "I think that concerning the prototype application in metagenomics, with all the added parameters it is now quite easy and fast to get quite complete and clean microbial genomes from the metagenomic soup. For the moment we are still evaluating the software in comparison to others in the domain, and it looks quite advanced, although this is a very fast moving field, and at the time when we started developing it, no software was available and now there are several.

What I really like in the tools is the visual aspect, and that I can select the data the way I want (although there are still some interesting parameters missing, that could be added afterwards).

To understand the visualisation and interactions was relatively easy for me, since initially the software was developed according to our needs and ideas. It was exactly what we asked for, and of course thanks to all the work of you guys involved in the project, it developed very nicely."

## 6.3 DISCUSSION

Chapter 4 and Chapter 5 introduced a large number of features and interactions to be used by users to gain insight into their data. Some of these features have been added to accommodate the use cases that were envisioned RParcoords and DimRedPlot would target, while others were added on request of people that used the tools to perform research with or that performed initial testing of the tools. During this evaluation we have seen that some of the features added indeed proved to be very useful, while others were slightly ignored or not used all that much. In this section we discuss what and why features proved useful and what the reasons could have been that other features were used less.

SELECTION AND COLOURING One of the most clearly useful features turned out to be the fact that the parallel coordinates and DimRedPlot instances are all linked together. E.g., a selection or colouring in one view can also be seen in other views. In both evaluations the researchers made quite some use of these features, both as interaction between the parallel coordinates and DimRedPlot and as interaction between different DimRedPlot instances.

In both evaluations the linked selections were mostly used to quickly see the relationship between observations in multiple views, or to gain some information about an observation that could not be gained through just one view, such as an observation's name. The colouring was mostly used when the relationships between several different groups of observations in different views had to be compared. Because, unlike selections, colouring is permanent until turned off, colouring allowed users to colour certain structures and then continue with other interactions while the colouring remained.

The fact that the same colouring in different views meant the same thing, such as selected observations always being black, meant that this functionality was easy to grasp for users.

ITERATIVE DIMENSIONALITY REDUCTION Iterative dimensionality reduction was used quite a lot in the evaluations, although to keep the evaluation succinct some of the occurrences were left out of the second evaluation. In the first evaluation iterative analysis was mostly used for small changes, e.g., some variables should not be used in DimRedPlot or some observations should be removed because they are outliers.

The second evaluation went further in this regard. Often, although not always stated, the researcher started by performing dimensionality reduction on a larger selection than the researcher was interested in. This was done to get a more complete view of the situation. After this, the researcher would filter out the irrelevant contigs and redo dimensionality reduction. The final selection was then often updated again to remove outliers or to include contigs not considered earlier. Beyond changing the selection of contigs to perform dimensionality reduction on, the researcher also reduced the number of variables used for dimensionality reduction once by selecting a smaller set through the contribution bar plots. The resulting DimRedPlot instance was then used as an extra confirmation that the remaining variables were indeed responsible for the observation structure seen in DimRedPlot.

Due to the fact that dimensionality reduction can easily be redone by first either changing the selection of observations or making a selection of variables and second pressing the button to perform dimensionality reduction, the feature was easy for users to understand and use.

SCATTERPLOTS The observation scatterplot in DimRedPlot was used by the users as the core of the dimensionality reduction visualisation. The scatterplot was easy for users to work with, mostly because it is a visualisation the users were familiar with. Features such as colouring and selecting observations were used quite a lot, especially in the context of linked views. The excentric labelling was also seen by users as a useful and easy to use tool to get a good idea of what the ellipses that they were looking at represented.

The variable scatterplot was unfortunately mostly ignored. A variable plot can most certainly provide valuable information, such as what groups of variables are similar and which variables are extreme in their differences, but during the two evaluations done in this thesis these questions never really came up. This left the variable scatterplot mostly ignored. There is also the problem that a plot of variables is, to many people, still quite an abstract concept, making it harder to understand the potential such a scatterplot can offer.

EIGEN-BAR Although the eigen-bar was used occasionally to actually change the eigenvectors used as scatterplot axes, it was not a feature that was really used a lot without our suggestion. When asking about this, users often said to be satisfied with the data-variance described by the first two eigenvectors, and it was often the case that other eigenvectors besides the first two described very small data-variance. It might, however, also very well be that users did not want to go through the difficulty of interpreting yet another eigenvector. If this really is a problem, a 3D projection instead of the current 2D projection might make this a smaller step for users.

The eigen-bar was used a lot to determine how much data-variance users were looking at. Although not specifically mentioned in the evaluation, especially in the second evaluation, the amount of data-variance described by the first two eigenvector was almost always looked at by the researcher.

The found data-variance influenced how serious the researchers took the conclusions drawn from the dimensionality reduction plot.

CONTRIBUTION BAR PLOTS The contribution bar plots were only used a couple of times, mostly in the second evaluation. Here it was both used to redo dimensionality reduction on a smaller set of variables, a use case described in Section 5.2.5, and to add variables to the parallel coordinates. The reason it was not used more probably had to do with that the evaluations just did not ask for it much. In the first evaluation the researchers were focused mainly on how the structure in the data was aligned with the geology variable. Although, looking back, using the bar plots more would have probably been useful in explaining the structure in the data when it was not well aligned with this variable. In the second evaluation the focus was mostly on what the structure in the tetranucleotides was and not so much on which specific variables were responsible for that.

In the cases where the bar plots were used, the interactions necessary were not seen as complicated by users.

TAGGING The tagging interface, discussed with quite some detail in Section 5.1.3, has not been used in the evaluation at all. The feature has actually been asked for by researchers and as we saw in the second evaluation, before the addition of the tagging system, the researchers kept selections by writing down which clusters were part of the selections. Tagging would make keeping these selections much easier, but for the evaluation it made little sense to copy all the different selections to the new tagging system. The usefulness of the tagging interface is not really doubted by us because of this. It would, however, have been nice to evaluate this interface as well.

# 7 CONCLUSION

As we have discussed in Chapter 1, dimensionality reduction is widely used in scientific research in order to analyse high-dimensional data. However, due to their abstractness the methods used are often not well understood by researchers and as a result they are treated as black boxes. This undermines the ability of researchers to fully grasp the extra information and insight that dimensionality reduction can offer. Furthermore, the fact that these techniques focus on either numerical or categorical variables makes analysing more complex datasets that contain both of these variables hard. This thesis has tried to solve these issues, condensed in the introduction as the following question:

> How can we, through linked visual metaphors, support the exploration and interpretation of dimensionality reduction on complex high-dimensional datasets?

In this thesis we have developed a solution to this problem with the creation of DimRedPlot, a new visual analytics tool that helps users get the most out of dimensionality reduction. Through elements such as the eigen-bar we created support for further exploration and understanding of the space generated by dimensionality reduction techniques.

By using linked visual metaphors such as selections and colouring to combine DimRedPlot with parallel coordinates, the results of dimensionality reduction have been made much easier to interpret. Users can with little effort understand how results shown in DimRedPlot are related to their original data shown in the parallel coordinates, which in turns helps users to get more insight into their data.

Using the same linked visual metaphors when multiple DimRedPlot instances are shown, we have also given users the possibility to combine dimensionality reduction on categorical and numerical data, so that the relationship between these two parts of complex datasets can easily be understood.

To validate our solution, RParcoords and DimRedPlot have been used to analyse and explore actual datasets produced and studied by researchers at the Luxembourg Institute of Science and Technology. The evaluation showed that using these tools researchers could indeed obtain new insight into their high-dimensional datasets quickly, even when dealing with complex datasets consisting of both numerical and categorical data. DimRedPlot and RParcoords continue to be used to support researchers at the LIST.

## 7.1 FUTURE WORK

Several features and ideas have not been implemented due to a lack of time or a lack of importance. This chapter lists the features that did not make it to the version of DimRedPlot outlined in this thesis but that could be implemented in the future to improve DimRedPlot.

### 7.1.1 *Distance preservation*

Some dimensionality reduction techniques try to reduce dimensionality by projecting observations onto new axes such that the distances between the observations are as close as possible as they were in the original dataset. Multi Dimensional Scaling methods are examples of techniques that do this. In

fact, although not its primary goal, PCA also has the effect that distances are preserved as much as possible within the constraint that the original data is linearly transformed. This means that when looking at observations projected onto new axes, as is done in DimRedPlot, when points are close together they are probably also close together in the original dataset. However, this can not be guaranteed, which can make scatterplots as used in DimRedPlot misleading to users. Points may seem to be close together, but in reality they can be quite far apart.

Martins et al. [42] propose several techniques to indicate what the difference is between projected distance between points and real distance between points. Integrating one of these techniques into DimRedPlot would greatly improve the certainty with which users can draw conclusions from the projected points.

### 7.1.2 *Supporting other dimensionality reduction techniques*

Although DimRedPlot is designed to visualise the results of PCA, MCA, and CA, DimRedPlot does not have any specific code for these methods and simply accepts a structured file with the results in them. Any dimensionality reduction technique whose results can be translated to this format can be visualised using DimRedPlot. Because of this, with only small changes to DimRedPlot it is possible for other techniques to be supported as well.

For a method to be supported it needs to output a couple of things. First off, the new axes that it creates need to be given a score in order for them to be displayed in the eigen-bar. This can be done for most methods by determining how well each axis confirms to the metric that the technique tries to optimise. In the case of multi dimensional scaling this could be done by looking at how the distances between points along each axis correspond to the real distances between points.

Second off, although not strictly necessary, DimRedPlot assumes both projected observations projected variables to be on the same axes. This essentially means that it should be possible to create a biplot with the results of the used dimensionality reduction technique. In Biplots in Practice [43] Greenacre explores the potential of many analysis technique, including multi dimensional scaling, to be visualised using biplots.

Finally, the method needs to have some metrics that determine the relationship between the generated axes and the variables in the dataset. These metrics are then shown in the current contribution bar plots. This is the most difficult point for adopting support for other dimensionality reduction techniques. The contribution metric is a specific metric for PCA-like techniques. This means that for different techniques similar metrics would have to be found. This can be done either with metrics specific to the used technique or with generic metrics that can be used in combination with any dimensionality reduction techniques, such as those discussed by Coimbra et al. [23]. In any case, when such metrics are used, care must be taken that the interpretation of the metrics does not change. The information alternative metrics show may be similar, but the difference in how they are calculated can have subtle difference in what they mean and how they are interpreted.

When no similar metrics exist, it would be possible to change DimRedPlot to make the contribution bar plots optional. Another option is to add more metrics that are not technique specific. The discrimination metric described in Section 4.2.2, for example, is not PCA specific and can be used with any dataset.

### 7.1.3 *Contribution table lens*

Right now, the contributions of the variables are displayed using several bar plots. The downside to using bar plots is that it is not immediately apparent what variable a certain bar is referring to. Writing the names on the variables is usually impossible since the bars are too low and thin for text to be displayed on it and still be readable. A solution to this problem would be to use a table lens instead. This table lens could either show the five metrics shown in the bar plots or show one metric at a time.

Using a table lens would allow the user to see the metrics for all variables on the screen at the same time. It would, however, also be possible to zoom in, and actually see the names of the variables displayed on the bars. If only one metric is shown in the table lens at a time, the metrics could be switched between using the same dropdown menu that can now be used to sort a specific bar plot. The downside to replacing the bar plots with a table lenses is that a table lens with all metrics in it would probably not have enough horizontal space, making it harder to show all metrics on the screen at the same time without horizontal scrolling.

# BIBLIOGRAPHY


[1] H. Hotelling. "Analysis of a complex of statistical variables into principal components." In: *Journal of Educational Psychology* 24.6 (1933), pp. 417–441.

[2] B Broeksema, M Calusinka, F McGee, X Goux, K Winter, M Ghoniem and P Delfosse. "ICoVeR - A novel interactive visualization interface for contig-bin verification and refinement". In: *Bioinformatics* (in review).

[3] J. Bertin. *La graphique et le traitement graphique de l'information*. Flammarion: Paris, 1977.

[4] R. Rao and S. K. Card. "The table lens: merging graphical and symbolic representations in an interactive focus+ context visualization for tabular information". In: *Proceedings of the SIGCHI conference on Human factors in computing systems* (1994), pp. 318–322.

[5] P. Pirolli and R. Rao. "Table lens as a tool for making sense of data". In: *Proceedings of the workshop on Advanced visual interfaces* June (1996), pp. 67–80.

[6] A. Telea. "Combining Extended Table Lens and Treemap Techniques for Visualizing Tabular Data". In: *Proceedings of the Eighth Joint Eurographics/IEEE VGTC conference on Visualization* (2006), pp. 51–58.

[7] M. Friendly. "Mosaic Displays for Multi-Way Contingency Tables". In: *Journal of the American Statistical Association* 89.425 (1994), pp. 190–200.

[8] A. Inselberg. "The plane with parallel coordinates". In: *The Visual Computer* 1.2 (1985), pp. 69–91.

[9] F. Bendix, R. Kosara and H. Hauser. "Parallel sets: Visual analysis of categorical data". In: *IEEE Symposium on Information Visualization, INFO VIS* (2005), pp. 133–140.

[10] H. Liu and L. Yu. "Toward integrating feature selection algorithms for classification and clustering". In: *IEEE Transactions on Knowledge and Data Engineering* 17.4 (2005), pp. 491–502.

[11] L. Van der Maaten and G. Hinton. "Visualizing data using t-SNE". In: *Journal of Machine Learning Research* 9 (2008), pp. 2579–2605.

[12] J. A. Lee and M. Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

[13] J. Benzécri. *L'Analyse des données*. Vol. 2. Dunod: Paris, 1973.

[14] H. Abdi and D. Valentin. "Multiple correspondence analysis". In: *Encyclopedia of measurement and statistics* (2007), pp. 651–657.

[15] M. Sedlmair, M. Brehmer, S. Ingram and T. Munzner. *Dimensionality Reduction in the Wild: Gaps and Guidance*. Tech. rep. University of British Columbia, 2012, p. 10.

[16] M. Brehmer, M. Sedlmair, S. Ingram and T. Munzner. "Visualizing dimensionally-reduced data: Interviews with analysts and a characterization of task sequences". In: *Proceedings of the Fifth Workshop on Beyond Time and Errors: Novel Evaluation Methods for Visualization* (2014), pp. 1–8.

[17] D. H. Jeong, C. Ziemkiewicz, B. Fisher, W. Ribarsky and R. Chang. "iPCA: An Interactive System for PCA based Visual Analytics". In: *Computer Graphics Forum* 28.3 (2009), pp. 767–774.

[18] S. Ingram, T. Munzner, V. Irvine, M. Tory, S. Bergner and T. Möller. "DimStiller: Workflows for dimensional analysis and reduction". In: *IEEE Symposium on Visual Analytics Science and Technology (VAST)* (2010), pp. 3–10.

[19] B. Broeksema, A. C. Telea and T. Baudel. "Visual analysis of multi-dimensional categorical data sets". In: *Computer Graphics Forum* 32.8 (2013), pp. 158–169.

[20] B. Broeksema, T. Baudel, A. C. Telea and P. Crisafulli. "Decision exploration lab: A visual analytics solution for decision management". In: *IEEE Transactions on Visualization and Computer Graphics* 19.12 (2013), pp. 1972–1981.

[21] K. R. Gabriel. "The biplot-graphical display of matrices with applications to principal components analysis". In: *Biometrika* 58 (1971), pp. 453–467.

[22] S. Oeltze, H. Hauser and J. Kehrer. "Interactive Visual Analysis of Perfusion Data". In: *IEEE Transactions on Visualization and Computer Graphics* 13.6 (2007), pp. 1392–1399.

[23] D. B. Coimbra, R. M. Martins, T. T. Neves, a. C. Telea and F. V. Paulovich. "Explaining three-dimensional dimensionality reduction plots". In: *Information Visualization* (2015), pp. 1–9.

[24] R. R. O. Silva, P. E. Rauber, R. M. Martins, R. Minghim and A. C. Telea. "Attribute-based Visual Explanation of Multidimensional Projections". In: *EuroVis Workshop on Visual Analytics* (2015).

[25] H. Abdi. "Singular Value Decomposition (SVD) and Generalized Singular Value Decomposition". In: *Encyclopedia of measurement and statistics* (2007), pp. 907–912.

[26] H. Abdi and L. J. Williams. "Principal component analysis". In: *Wiley Interdisciplinary Reviews: Computational Statistics* 2.4 (2010), pp. 433–459.

[27] H. Abdi and L. Williams. "Correspondence Analysis". In: *Encyclopedia of Research Design* (2010), pp. 1–20.

[28] N. Elmqvist, P. Dragicevic and J.-D. Fekete. "Rolling the Dice : Multidimensional Visual Exploration using Scatterplot Matrix Navigation To cite this version : Rolling the Dice : Multidimensional Visual Exploration using Scatterplot Matrix Navigation". In: *IEEE Transactions on Visualization and Computer Graphics* 14.6 (2008), pp. 1141–1148.

[29] C. Turkay, P. Filzmoser and H. Hauser. "Brushing dimensions-A dual visual analysis model for high-dimensional data". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2591–2599.

[30] P. E. Rauber, R. R. O. Silva, S. Feringa, M. E. Celebi, A. X. Falcão and A. C. Telea. "Interactive Image Feature Selection Aided by Dimensionality Reduction". In: *EuroVis Workshop on Visual Analytics (EuroVA)* (2015).

[31] J. Christensen, J. Marks and S. Shieber. "An empirical study of algorithms for point-feature label placement". In: *ACM Transactions on Graphics* 14.3 (1995), pp. 203–232.

[32] J.-D. Fekete. "Excentric Labeling: Dynamic Neighborhood Labeling for Data Visualizaition". In: *Proceedings of the SIGCHI conference on Human Factors in Computing Systems* 09 (1999), pp. 512–519.

[33] E. R. Tufte. *The Visual Display of Quantitative Information*. Vol. 2. Graphics press LLC, 2007, p. 197.

[34] M. Bostock, V. Ogievetsky and J. Heer. "D3; Data-Driven Documents". In: *IEEE Transactions on Visualization and Computer Graphics* 17.12 (2011), pp. 2301–2309.

[35] M. Harrower and C. A. Brewer. "ColorBrewer.org: An Online Tool for Selecting Colour Schemes for Maps". In: *The Cartographic Journal* 40.1 (2003), pp. 27–37.

[36] R. Ihaka and R. Gentleman. "R: A Language for Data Analysis and Graphics". In: *Journal of Computational and Graphical Statistics* 5.3 (1996), pp. 299–314.

[37] K. Chang. *Parallel Coordinates, A visual toolkit for multidimensional detectives*. `https://syntagmatic.github.io/parallel-coordinates/`.

[38] J. Ooms. *The OpenCPU System : Towards a Universal Interface for Scientific Computing through Separation of Concerns*. 2014. arXiv: `1406.4806v1`.

[39] S. Lê, J. Josse and F. Husson. "FactoMineR : An R package for multivariate analysis". In: *Journal of Statistical Software* 25.1 (2008), pp. 1–18.

[40] O. Nenadic and M. Greenacre. "Correspondence analysis in R, with two- and three-dimensional graphics: the ca package". In: *Journal of Statistical Software* 20.3 (2007), pp. 1 –13. arXiv: `1501.0228`.

[41] T. Schumacher. "Terroir an der Luxemburgischen Mosel, Eigenschaften von Weinbergsböden". MA thesis. Universität Trier, 2014.

[42] R. M. Martins, D. B. Coimbra, R. Minghim and A. C. Telea. "Visual analysis of dimensionality reduction quality for parameterized projections". In: *Computers and Graphics* 41 (2014), pp. 26–42.

[43] M. Greenacre. *Biplots in Practice*. Fundacion BBVA, 2010, p. 219.