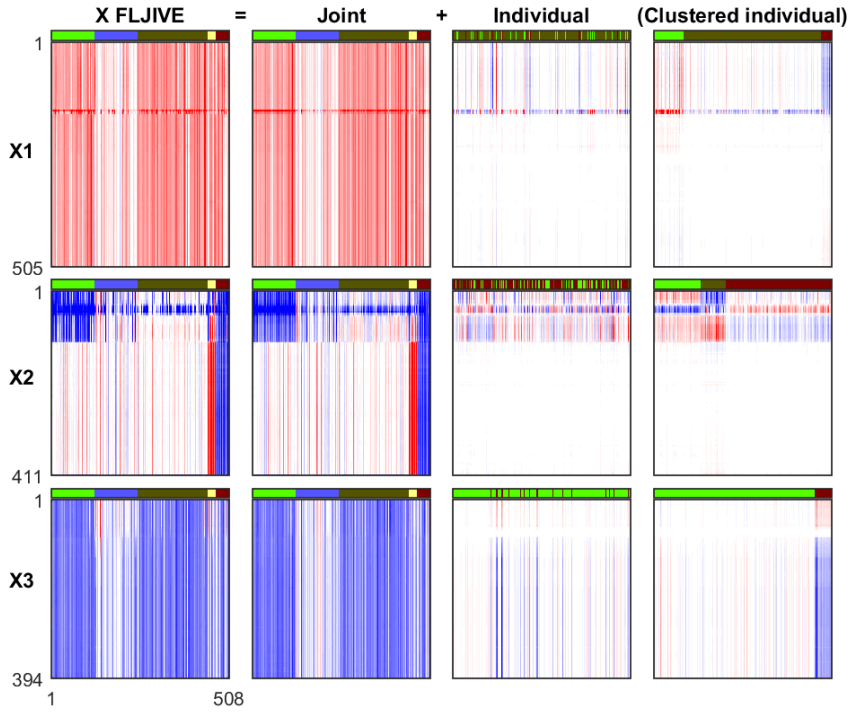
CHALMERS



Discovery of subgroup dynamics in Glioblastoma multiforme

using integrative clustering methods and multiple data types

Master's Thesis in Computer Science: Algorithms, Languages and Logics

Sebastian Ånerud

Department of Computer Science & Engineering
Chalmers University of Technology
Gothenburg, Sweden 2015

Discovery of subgroup dynamics in Glioblastoma multiforme using integrative clustering methods and multiple data types

Sebastian Ånerud

Examiner: Graham Kemp
Supervisor: Rebecka Jörnsten

Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Gothenburg
Sweden
Telephone: +46 (0)31-772 1000

Cover: Showing results of applying JIVE and JIC to chromosome 7, 9 and 10 of the TCGA GBM Copy Number data set.

Department of Computer Science and Engineering
Gothenburg, Sweden. June 16, 2015.

Abstract

An integrative data mining method, using multiple data types, called *Joint and Individual Variation Explained* (JIVE) and it's existing sparse version *Sparse JIVE* (sJIVE) are analysed and further extended. The proposed extension, called *Fused Lasso JIVE* (FLJIVE), includes the integration of a Fused Lasso penalization framework into the JIVE method. Also, a model selection tool for selecting the parameters in the JIVE model is proposed. The new model selection algorithm and the three versions of the method, JIVE, sJIVE and FLJIVE, are analysed and compared in a simulation study and later applied to the TCGA Glioblastoma Multiforme Copy Number (CNA) data which is know to have fused properties. The simulation study shows that the rank selection algorithm is successful and that FLJIVE is superior JIVE and sJIVE when the data have underlying fused properties. The results of applying the methods to the TCGA data set suggest that large parts of the underlying mutational process is shared between chromosome 7, 9 and 10. Results also suggest that chromosome 1 does not share as much of this process and that chromosome 15 is almost independent of this process.

Acknowledgements

Firstly, I would like to thank my supervisor at Chalmers University of Technology, Rebecka Jörnsten, for helping me with the composition and proposal of this Thesis. I would also like to thank her for continuous feedback, great support and for her encouragement throughout the entire project. Without her expertise and state of the art knowledge of the field this thesis would not have been as interesting and close to current research. Lastly, I would also like to thank my examiner, Graham Kemp, for helpful input, feedback and guidance during the different stages of the thesis.

I would also like to acknowledge that this is a collaborative thesis between the Computer Science and Mathematical Statistics departments at the Chalmers University of Technology.

Sebastian Ånerud, Gothenburg June 16, 2015

Contents

1

Introduction

The main topic of the thesis is an integrative data analysis method called Joint and Individual Variation Explained (JIVE). In this thesis the Fused Lasso penalization framework is integrated into the JIVE method and a novel rank selection algorithm for JIVE is presented. The methods are then evaluated in a simulation study and then applied to a real data set.

1.1 Background

In many research fields it is getting more and more common that data are measured in multiple different data types for a common set of objects. Examples of different objects and possible data types are shown in Table 1.1. The Cancer Genome Atlas (TCGA, homepage available at: *http://cancergenome.nih.gov/*) provides such data for a large set of patients diagnosed with the malignant brain tumor Glioblastoma Multiforme (GBM). Also, in an exclusive collaboration, the Nelander Lab at Uppsala university is providing Chalmers with data from the Human Glioma Cell Culture (HGCC) which are GBM cell lines grown in a laboratory at the university hospital. Both data sets include measurements for patients' (or cell lines') copy number aberrations (CNA), DNA methylation, gene expression, somatic mutations and miRNA.

Table 1.1: Showing three examples of objects and corresponding data types.

Object	**Data types**
Websites	Word frequencies, visitor demographics, linked pages
Artists (music)	Genre classifications, listener demographics, related artists
Patients	Copy number, gene expression, microRNA

However, the different cases in GBM are highly heterogeneous, and understanding the dynamics of subgroups, and how they are defined among the patients, may lead to more effective targeted therapy plans for new patients. It is also of great importance to investigate how similar the cell lines in HGCC, which are grown from early stage cancer tissue originating from patients treated in Sweden, are to the late stage cancer samples from the patients in the TCGA data set. If there is a strong connection, then one could possibly test new drugs and therapies for their effectiveness on the cell lines rather than on real patients. This opens up lots of opportunities for discovering new effective cancer therapies. However, in order to make these kind of analyses possible new statistical and data mining techniques need to be developed.

This thesis will investigate and extend a framework which is a step in the direction to where the questions mentioned above could be answered. The basis for this framework is a method called Joint and Individual Variation Explained (JIVE) [1]. The method tries to simultaneously decompose a dataset, containing multiple data types, into two different levels of structure. The first level contains structure which is shared across all data types. The second level contains individual data type-specific structure for each data type, where the individual structures are independent of the joint structure but also uniquely defined in each data type. By studying this method one can gain knowledge about it's current limitations and the soundness of it's underlying assumptions. This is important in order to, in the future, extend the model to handle multiple data sets and not just multiple data types.

1.2 Aim

To describe the purpose of this thesis, and what it tries to achieve, it is divided into three different aims. These aims will also be recurring themes in, and define the structure and flow of, the following sections of the thesis. This thesis aims to:

1. Extensively explore JIVE in order to learn more about it's limitations, underlying assumptions and the model itself. Knowledge about this is needed in order to extend the method.

2. Extend the current sparsity framework in JIVE to incorporate the underlying structure of genomic data. This can be done by integrating the Fused Lasso penalization [9] into the JIVE procedure.

3. Investigate possible ways of finding the most suitable parameters for the JIVE method. A solid model selection tool is of great importance when comparing the original model to extended versions of it.

For the first aim to be reached, general understanding of JIVE is needed. This includes the ability to draw conclusions about when it works as intended and for what kind of input it generally fails. It also includes investigating how sensitive JIVE is to the different parameters in the model. This is important when applying the method to real

data where the underlying structure is not known. It is also important since this class of integrative clustering methods, specifically derived to handle multiple data types, is new and not extensively explored and documented.

The second aim is reached by formulating and understanding the Fused Lasso sparsity framework. This must be done in order to be able to interchange the current sparsity setting in JIVE. In the current form of JIVE the sparsity is incorporated into the Principal Component Analysis (PCA) method, which is used as a subroutine within JIVE. To be able to interchange the current non-structured sparsity penalization with a structured sparsity penalization, the Fused Lasso penalty must be applied in the PCA step of the method. This means that a great part of this project is to derive an algorithm for PCA which incorporates the Fused Lasso sparsity into the principal component loadings. This new structured sparsity is simply integrated into the JIVE procedure by replacing the current PCA method with the Fused Lasso PCA (FLPCA).

The last aim comprises the challenge of how to validate the choice of parameters for this new class of integrative methods. This class started with R. Shen, A. Olshen and M. Ladanyi proposing a new method called iCluster [3] as recently as 2009. The method was designed specifically for solving the problem of clustering a common set of observation measured in many different types of data. The method was applied to Glioblastoma CNA, miRNA and methylation data with promising results. A few years later Mo, Q. et al. extended the method which resulted in iCluster+ [4]. The contribution of the underlying method for this thesis, JIVE, was done by E. Lock, et al. [1] where they applied the method to GBM, miRNA and gene expression data. K. Hellton and M. Thoresen [2] extended JIVE into a method named Joint and Individual Clustering (JIC), which also cluster the observations, as recently as November 2014. Commonly discussed in these papers are the problems of validating the results and choice of model parameters. This demonstrates a consensus solution to the model selection problem has not yet been found. It also suggests that there is a need for novel model selection tools.

1.3 Limitations

As mentioned in the background, a future aim is to extend the JIVE procedure to handle more than one data set so that one can analyse the similarities between the HGCC cell lines and the TCGA late stage cancer patients. The analysis would need the method to extract common and individual features between data sets as well as between data types. However, this thesis will not try to analyse the similarities/differences between the TCGA and the HGCC data sets, and it will not apply any of the methods to the HGCC data set. This is mostly because the HGCC data set needs more work to be assembled in the same way as the TCGA data set, and partly because such an analysis is large enough to be a paper itself.

Also, this thesis will not try to model an extension of JIVE capable of analysing multiple data sets. In order to model such a large-scale extension, more knowledge about the original model is required. The time frame of this thesis does not allow for both such an extension and the pre-work needed in order to carry it out. Therefore, this thesis is limited to the smaller extension of Fused Lasso JIVE and will lay the grounds for the more large-scale extension.

The model selection problem for JIVE will be addressed in this thesis. However, the model selection problem for the penalization parameters in sparse JIVE and Fused Lasso JIVE will not be thoroughly addressed and discussed throughout the thesis. The use of more sophisticated model selection tools for this problem cannot be fitted into the scope of this thesis, and instead, these parameters will be selected using visual guidance.

1.4 Thesis outline

In section 2 methods important to the thesis will be explained. Understanding the Principal Component Analysis and k-means methods are important in order to understand and to follow the motivation of JIVE and it's extension JIC which also clusters the joint and individual components in JIVE. Subsection 2.5 discusses the sparse PCA method and proceeds by defining the generalized Fused Lasso problem and how the split Bregmann method can be used to solve it. The Fused Lasso PCA is then proposed and derived. The topic of subsection 2.6 is model selection which introduces a novel rank selection algorithm for JIVE and discusses how model selection for the penalization parameters is done throughout this thesis. The last subsection of section 2 gives a short introduction to how the results of JIVE and JIC are visualized.

In section 3 a simulation study is conducted where the proposed rank selection algorithm is evaluated on data with known underlying parameters. The three methods JIVE, sJIVE and FLJIVE are then tested on simulated data where the assumption of FLJIVE is not true and then on a simulated data set where the assumption of FLJIVE holds.

The rank selection algorithm and the three variations of JIVE are then applied to the TCGA data set in section 4. In this section only the CNA data type will be used, and instead, each chromosome will be interpreted as its own data type. Firstly, the rank selection algorithm is applied to the data in order to estimate the most favourable parameters. Then, given the parameters suggested by the rank selection algorithm, JIVE, sJIVE and FLJIVE are fitted to three different set of chromosomes in order to discover the relationship between the underlying mutational process of the chromosomes.

In section 5 a discussion about the performance of the rank selection algorithm is held. The performance of FLJIVE is also discussed in a setting where the underlying assumption is true and in settings where the assumption is violated. The section also discusses JIVE as a method for analysing the connection between the underlying mutational processes of the chromosomes. Lastly, possible future directions are discussed.

2

Methods

The most central methods in this thesis are Joint and Individual Variation Explained (JIVE) and its extension Joint and Individual Clustering (JIC). However, there are theories important to understand in order to understand these methods. Therefore, this section begins by explaining methods important to the thesis. The section then formulates JIVE and JIC mathematically before describing the sparsity framework currently available and it's extension derived in this thesis. Also, a novel rank selection algorithm for JIVE and JIC is presented. The section ends with a short introduction to how the results of JIVE and JIC are visualized. In all formulations n refers to the number of observations (rows) and p to the number of features (columns). Also, in general, matrices are written in upper case bold letters, vectors are written in lower case bold letters and numerical variables are written as lower case normal face letters.

2.1 Principal Component Analysis (PCA)

Principal Component Analysis is a method that decomposes a matrix $\mathbf{X}$ into two matrices: the principal component scores $\mathbf{Z}$ and their loading factors $\mathbf{W}$. More formally it can be written as:

$$\mathbf{X} = \mathbf{Z}\mathbf{W}^T,$$

where $\mathbf{X}$ is an $n \times p$-matrix, $\mathbf{W}$ is a $p \times p$-matrix whose columns must be orthonormal and $\mathbf{Z}$ is an $n \times p$-matrix whose columns must be orthogonal. The $\mathbf{W}$ and $\mathbf{Z}$ are chosen such that they minimizes the following optimization problem:

$$\begin{aligned} \min_{\mathbf{Z},\mathbf{W}} \frac{1}{2}\|\mathbf{X}-\mathbf{Z}\mathbf{W}^T\|_F^2 = &\min_{\mathbf{Z},\mathbf{W}} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{p}(\mathbf{X}-\mathbf{Z}\mathbf{W}^T)_{ij}^2 \\ &\text{s.t} \\ \mathbf{W}^T\mathbf{W} = &\mathbf{I} \\ \mathbf{Z}^T\mathbf{Z} = &\begin{pmatrix} s_1 & & 0 \\ & \ddots & \\ 0 & & s_p \end{pmatrix}, \end{aligned} \tag{2.1}$$

where $\mathbf{I}$ is the $p \times p$ identity matrix and $\mathbf{Z}^T\mathbf{Z}$ is a diagonal matrix with the singular values $s_1 \geq ... \geq s_p \geq 0$ on the diagonal.

An important application of PCA is the approximation of a matrix $\mathbf{X}$ with another matrix of rank $r \leq \min(n,p)$. In many cases the rank r of a large matrix $\mathbf{X}$ is much smaller than $\min(n,p)$ which means that the matrix could be approximated well using only a few components. Let $\mathbf{W}_{[r]}$ denote the first r columns of $\mathbf{W}$, $\mathbf{Z}_{[r]}$ denote the first r columns of $\mathbf{Z}$, and let $\mathbf{X}_{(r)}$ denote the best r-rank approximation of $\mathbf{X}$. Then $\mathbf{X}_{(r)}$ can be written as:

$$\mathbf{X}_{(r)} = \mathbf{Z}_{[r]}\mathbf{W}_{[r]}^T = \sum_{i=1}^{r}\mathbf{Z}_i\mathbf{W}_i^T, \tag{2.2}$$

where $\mathbf{Z}_i$ is the i:th column of $\mathbf{Z}$ and $\mathbf{W}_i$ is the i:th column of $\mathbf{W}$. If one is interested in finding the best 1-rank approximation of $\mathbf{X}$, the optimization problem could be formulated as:

$$\min_{\mathbf{z},\mathbf{w}} \frac{1}{2}\|\mathbf{X}-\mathbf{z}\mathbf{w}^T\|_F^2, \tag{2.3}$$

where $\mathbf{z}$ is a $n \times 1$-vector and $\mathbf{w}$ is a $p \times 1$ vector. By equation 2.2 the solution to 2.3 is $\mathbf{z} = \mathbf{Z}_1$ and $\mathbf{w} = \mathbf{W}_1$. Finding the subsequent vector-pairs $\mathbf{Z}_i$,$\mathbf{W}_i$ for $i > 1$ is equivalent to finding the best 1-rank approximation of the residual matrix $\mathbf{R}_i = \mathbf{X} - \sum_{j=1}^{i-1}\mathbf{Z}_j\mathbf{W}_j^T$. The Non-linear Iterative Partial Least Squares algorithm (NIPALS) uses this fact to compute the first few components in a principal component analysis [5]. Given a sufficiently large number m, which will ensure that the algorithm will return, the algorithm for extracting the r first components is defined as follows:

Algorithm 1 PCA NIPALS

procedure PCA($\mathbf{X}$,r,ϵ,m)
  $\mathbf{R} = \mathbf{X}$.
  for ($i = 1$,...,r) **do**
    $\delta = \infty$
    $\mathbf{z} = \mathbf{R}_i$
    for $j = 1$,...m **do**
      $\mathbf{w} = \mathbf{R}^T\mathbf{z}$
      $\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
      $\mathbf{z} = \mathbf{R}\mathbf{w}$
      if $|\delta - \|\mathbf{z}\|| < \epsilon$ **then**
        break
      end if
      $\delta = \|\mathbf{z}\|$
    end for
    $\mathbf{W}_i = \mathbf{w}$
    $\mathbf{Z}_i = \mathbf{z}$
    $\mathbf{R} = \mathbf{R} - \mathbf{z}\mathbf{w}^T$
  end for
  return $\mathbf{W}$,$\mathbf{Z}$
end procedure

2.2 k-means

In this section a brief explanation of the k-means clustering method will be presented. Since k-means is a well-established and well-known algorithm with many implementations available, this section will only state the algorithm for solving the optimization problem and not discuss some of its drawbacks. Instead, this section will focus on the mathematical definitions needed in order to understand the method and its extension reduced k-means. Firstly, the original version of k-means is defined, and then the two-step PCA/k-means procedure, knows as reduced k-means, will be introduced.

2.2.1 Conventional k-means

The well known unsupervised method k-means is a clustering method for dividing a set of objects into a predefined number K cohesive groups (clusters). The name k-means comes from the fact that the algorithm finds K vectors $\boldsymbol{w}_1$,...,$\boldsymbol{w}_K$ which defines the centres of the K clusters, and given the cluster centres $\boldsymbol{w}_1$,...,$\boldsymbol{w}_K$ the cluster membership of an object $\mathbf{x}$ is then defined as:

$$C(\mathbf{x}) = \arg\min_k \|\mathbf{x} - \boldsymbol{w}_k\|^2. \tag{2.4}$$

The objective function which k-means is opting to minimize is the *within cluster sum of squares* ($WCSS$):

$$WCSS = \frac{1}{2}\sum_{k=1}^{K}\sum_{i\in C_k}\|\mathbf{x_i} - \boldsymbol{w}_k\|^2, \tag{2.5}$$

where C_k is defined as the set $\{i : C(\boldsymbol{x}_i) = k\}$. The most common implementation of the k-means algorithms alternates between finding the cluster memberships (2.4) and minimizing the objective function (2.5) with respect to $\boldsymbol{w}_1$,..., $\boldsymbol{w}_K$. With $|C_k|$ being the number of objects in cluster k the algorithm looks as follows:

Algorithm 2 k-means clustering algorithm

procedure K-MEANS($\mathbf{X}$,K)
  initialize $\boldsymbol{w}_1$,..., $\boldsymbol{w}_K$.
  while not converged **do**
    Update $C(\boldsymbol{x}_1)$,...,$C(\boldsymbol{x}_n)$
    for $k = 1$,...K **do**
      $\boldsymbol{w}_k = \frac{1}{|C_k|}\sum_{i\in C_k}\boldsymbol{x}_i$
    end for
  end while
  return $C(\boldsymbol{x}_1)$,...,$C(\boldsymbol{x}_n)$
end procedure

2.2.2 Reduced k-means (k-means via PCA)

In many applications the number of features p exceeds the number of objects n, and if p is really large, the running time of k-means will suffer. The idea with reduced k-means is to first reduce the dimensionality of the data, and then cluster it in order to find the cluster indicators. In this way the running time of the algorithm can be kept low, as long as the dimension reduction technique is fast. Instead of having the solution of k-means to be a vector of cluster belongings $C(\boldsymbol{x}_1)$,...,$C(\boldsymbol{x}_n)$, let the solution be the cluster indication matrix $\boldsymbol{Z} = (\boldsymbol{z}_1, ..., \boldsymbol{z}_{K-1})$ where:

$$\boldsymbol{z}_k^T = \frac{1}{\sqrt{|C_k|}}\left(0,\ldots,0,\underbrace{1,\ldots,1}_{|C_k|},0,\ldots,0\right),$$

and $z_{ik} > 0$ meaning that object i belongs to cluster k. $\boldsymbol{Z}$ only needs $K-1$ indicator vectors to contain all clustering information since the observations belonging to cluster K will have zeros in every column and can in that way be identified. Also, let $\boldsymbol{W} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{K-1})$. Then the objective function (2.5) of k-means can be reformulated as:

$$\begin{aligned}\min_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_K} WCSS &= \min_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_K} \frac{1}{2}\sum_{k=1}^{K}\sum_{i\in C_k}\|\mathbf{x_i}-\boldsymbol{w}_k\|^2 \\ &\Leftrightarrow \min_{\boldsymbol{w}_1,\ldots,\boldsymbol{w}_K} \frac{1}{2}\sum_{k=1}^{K}\sum_{i=1}^{n} z_{ik}\|\mathbf{x_i}-\boldsymbol{w}_k\|^2 \\ &= \min_{\boldsymbol{W}} \frac{1}{2}\sum_{i=1}^{n}\|\mathbf{x_i}-\boldsymbol{z}_i\boldsymbol{W}^T\|^2 \\ &= \min_{\boldsymbol{W}} \frac{1}{2}\|\mathbf{X}-\boldsymbol{Z}\boldsymbol{W}^T\|_F^2\end{aligned} \tag{2.6}$$

In this form the k-means objective function is undoubtedly very similar to the objective function (2.1) of PCA. In fact, it has been shown that the principal component analysis is the continuous version of the k-means minimization problem and that the principal component score matrix in PCA is the continuous version of the cluster indicator matrix in k-means [6], [7]. In order to retrieve the actual cluster indications, one can apply k-means to the principal components scores. This give rise to the following two-step procedure:

Algorithm 3 reduced-k-means clustering algorithm

1: **procedure** REDUCED-K-MEANS($\mathbf{X}$,K,ϵ,m)
2: W,Z = PCA($\mathbf{X}$,$K-1$,ϵ,m)
3: $C(\boldsymbol{x}_1)$,...,$C(\boldsymbol{x}_n)$ = k-means(Z,K)
4: **return** $C(\boldsymbol{x}_1)$,...,$C(\boldsymbol{x}_n)$
5: **end procedure**

Note that in Algorithm 3 k-means is run on Z which is a $n\times(K-1)$-matrix instead of X which is a $n\times p$ matrix. If p is large, the running time of the reduced-k-means algorithm is significantly less then the running time of the conventional k-means algorithm.

2.3 Joint and Individual Variation Explained (JIVE)

Given the ranks $r, r_1, \ldots, r_I$ JIVE is a method that decomposes multiple matrices $\mathbf{X}_1, \ldots, \mathbf{X}_I$, measuring I types of data for the same set of observations, each into two components: One joint component $\boldsymbol{J} = (\boldsymbol{J}_1, \ldots, \boldsymbol{J}_I)$ which encodes structure that is shared between all data types and one individual component $\boldsymbol{A} = (\boldsymbol{A}_1, \ldots, \boldsymbol{A}_I)$ which encodes information only present in the corresponding data type. The matrix $\boldsymbol{J}$ is constrained to have rank r and the matrices $\boldsymbol{A}_i$ are constrained to have rank r_i respectively.

Furthermore, the $\boldsymbol{J}$ and the $\boldsymbol{A}_i$ terms should be orthogonal ($\boldsymbol{J}^T\boldsymbol{A}_i = \mathbf{0}$) which ensures that $\boldsymbol{J}$ and the $\boldsymbol{A}_i$ terms do not share any structure among the objects. If the individual

structures were to share the same structure among them, it would in fact be a joint structure, and this which would be counter intuitive. Mathematically this can be written as:

$$\begin{aligned}
\mathbf{X}_1 =& \boldsymbol{J}_1 + \boldsymbol{A}_1 + \epsilon_1 \\
&\vdots \\
\mathbf{X}_i =& \boldsymbol{J}_i + \boldsymbol{A}_i + \epsilon_i \\
&\vdots \\
\mathbf{X}_I =& \boldsymbol{J}_I + \boldsymbol{A}_I + \epsilon_I \\
&\text{s.t} \\
\boldsymbol{J}^T \boldsymbol{A}_i =& \mathbf{0} \quad \forall\, 1 \leq i \leq I,
\end{aligned} \tag{2.7}$$

where $\mathbf{X}_i, \boldsymbol{J}_i, \boldsymbol{A}_i$ and ϵ_i are all $n \times p_i$-matrices, $\boldsymbol{J}_i$ is the sub-matrix of the joint structure $\boldsymbol{J}$ corresponding to data type i, $\boldsymbol{A}_i$ is the individual structure for data type i and ϵ_i represents noise specific for data type i. Note that $\boldsymbol{J}_i$ and $\boldsymbol{A}_i$ refer to the joint and individual matrices of data type i and not to a row or column.

The matrices $\boldsymbol{J}_1, \ldots, \boldsymbol{J}_I$ and $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_i$ can be further decomposed into two components each, using PCA, as:

$$\begin{aligned}
\mathbf{X}_1 =& \mathbf{Z}\mathbf{W}_1^T + \mathbf{Z}_1 \boldsymbol{V}_1^T + \epsilon_1 \\
&\vdots \\
\mathbf{X}_i =& \mathbf{Z}\mathbf{W}_i^T + \mathbf{Z}_i \boldsymbol{V}_i^T + \epsilon_i \\
&\vdots \\
\mathbf{X}_I =& \mathbf{Z}\mathbf{W}_I^T + \mathbf{Z}_I \boldsymbol{V}_I^T + \epsilon_I,
\end{aligned} \tag{2.8}$$

Note that $\mathbf{Z}$ is shared between all data types for the joint component but not for the individual component. This further decomposition is practical in three ways: Firstly, PCA could be used to estimate the matrices $\boldsymbol{J}, \boldsymbol{A}_1, ..., \boldsymbol{A}_I$ while forcing structure to be shared between the $\boldsymbol{J}_i$ but not between the $\boldsymbol{A}_i$ terms. Secondly, the rank for the matrices $\boldsymbol{J}, \boldsymbol{A}_1, \ldots, \boldsymbol{A}_I$ can be controlled by using PCA to find the best r-rank approximation of $\boldsymbol{J}$ and the best r_i-rank approximation of $\boldsymbol{A}_i$. Lastly, the reduced k-means procedure can be applied in order to find clusters. This will be further discussed in section below.

JIVE finds $\boldsymbol{J}$ and the $\boldsymbol{A}_i$ terms by minimizing the squared Frobenious norm of the following residual matrix:

$$\boldsymbol{R} = \begin{pmatrix} \boldsymbol{R}_1 \\ \vdots \\ \boldsymbol{R}_i \\ \vdots \\ \boldsymbol{R}_I \end{pmatrix}^T = \begin{pmatrix} \boldsymbol{X}_1 - \boldsymbol{J}_1 - \boldsymbol{A}_1 \\ \vdots \\ \boldsymbol{X}_i - \boldsymbol{J}_i - \boldsymbol{A}_i \\ \vdots \\ \boldsymbol{X}_I - \boldsymbol{J}_I - \boldsymbol{A}_I \end{pmatrix}^T \tag{2.9}$$

Similarly to PCA, the estimation of $\boldsymbol{J}$ and the different $\boldsymbol{A}_i$ are done iteratively by alternating between estimating $\boldsymbol{J}$ for a fixed $\boldsymbol{A}$ and estimating the $\boldsymbol{A}_i$ terms given $\boldsymbol{J}$. In fact, the $\boldsymbol{J}$ that minimizes $\|\boldsymbol{R}\|_F^2$ is $\boldsymbol{J} = \boldsymbol{Z}\boldsymbol{W}^T$ where $\boldsymbol{Z}$ and $\boldsymbol{W}$ are the scores and loadings of the r-rank PCA approximation of $\boldsymbol{X} - \boldsymbol{A}$. Similarly, the $\boldsymbol{A}_i$ that minimizes $\|\boldsymbol{R}\|_F^2$ is $\boldsymbol{A}_i = \boldsymbol{Z}_i\boldsymbol{W}_i^T$ where $\boldsymbol{Z}_i$ and $\boldsymbol{W}_i$ are the scores and loadings of the r_i-rank PCA approximation of $\boldsymbol{X}_i - \boldsymbol{J}_i$. However, this is without the orthogonality constraint between $\boldsymbol{J}$ and the $\boldsymbol{A}_i$ terms taken into account. The derivation for the correction due to the orthogonality constraint can be found in the supplementary material of the original JIVE article [1]. The JIVE algorithm is defined as:

Algorithm 4 JIVE algorithm

procedure JIVE($\boldsymbol{X}, r, r_1 \dots r_I$)
  $\boldsymbol{R} = \boldsymbol{X}$
  while not converged **do**
    $\boldsymbol{W}, \boldsymbol{Z} = \text{PCA}(\boldsymbol{R}, r)$
    $\boldsymbol{R} = \boldsymbol{X} - \boldsymbol{Z}\boldsymbol{W}^T$
    $\boldsymbol{Z}' = \text{normalize}(\boldsymbol{Z})$
    for $i = 1,...I$ **do**
      $\boldsymbol{W}_i, \boldsymbol{Z}_i = \text{PCA}((\boldsymbol{I} - \boldsymbol{Z}'\boldsymbol{Z}'^T)\boldsymbol{R}_i,\ r_i)$
      $\boldsymbol{R}_i = \boldsymbol{X}_i - \boldsymbol{Z}_i\boldsymbol{W}_i^T$
    end for
  end while
  return $\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{W}_1, \boldsymbol{Z}_1, \dots, \boldsymbol{W}_I, \boldsymbol{Z}_I$
end procedure

2.4 Joint and Individual Clustering (JIC)

The extension from JIVE into JIC is rather straightforward. In words, JIC is a combination between JIVE and reduced k-means which uses the fact that JIVE is a decomposition of the form (2.8). Given the data $\boldsymbol{X}$, the number of joint clusters c and the number of individual clusters c_i, JIC starts by doing a JIVE-decomposition and then applies k-means to the result returned by JIVE. More specifically, since $\boldsymbol{J}$ and the $\boldsymbol{A}_i$ terms are estimated via PCA, reduced k-means can be applied to them in order to find clusters

that are defined jointly among the data types and individually in each data type. In practice this is just applying k-means to the $\mathbf{Z}$ and $\mathbf{Z}_i$'s returned by JIVE. Note that the relationship between the number of clusters in JIC and the ranks in JIVE is $c = r + 1$. The algorithm for JIC looks as follows:

Algorithm 5 JIC algorithm

1: **procedure** JIC($\boldsymbol{X}, c, c_1 \dots c_I$)
2: $\boldsymbol{W}, \boldsymbol{Z}, \boldsymbol{W}_1, \boldsymbol{Z}_1, \dots, \boldsymbol{W}_I, \boldsymbol{Z}_I = \text{JIVE}(\boldsymbol{X}, c-1, c_1-1, \dots, c_I-1)$
3: **return** k-means($\boldsymbol{Z}$,c), k-means($\boldsymbol{Z}_1$,c_1) $\dots$, k-means($\boldsymbol{Z}_I$,c_I)
4: **end procedure**

2.5 Sparsity framework

In the JIVE model (2.7) the $\boldsymbol{J}$ and the different $\boldsymbol{A}_i$ does not assume the data to have any kind of structure. If the underlying true signal of the data is in fact structured, JIVE can fail to estimate this structure due to a high degree of noise. In some applications it is therefore desirable that JIVE takes some prior knowledge or assumption of the structure of the data into account. One example is the assumption that the underlying signal is sparse, i.e. a majority of the entries in the true $\boldsymbol{J}$ and $\boldsymbol{A}_i$ terms are exactly 0. In order to enforce structure in JIVE, one can enforce the structure into the underlying PCA decompositions of JIVE.

This section will define two different extensions of PCA that can be used inside the JIVE procedure in order to enforce structure on the resulting fit. Firstly, the already existing sparse PCA method (sPCA), which imposes sparsity on the loadings in the decomposition, will be described. Before defining the second PCA extension, a general definition of the Fused Lasso, which can be used to enforce anything from a regular sparse structure to a graph structure, will be introduced. Lastly, the derivation of how the Fused Lasso can be used in combination with PCA to form the Fused Lasso PCA (FLPCA) is carried out.

2.5.1 Sparse PCA (sPCA)

As described in the paper by H. Shen and J. Z. Huang [8] one can enforce sparsity on the loading factors in a principal component analysis by rewriting the optimization problem in (2.3) as:

$$\min_{\mathbf{z},\mathbf{w}} \frac{1}{2}\|\mathbf{X} - \mathbf{z}\mathbf{w}^T\|_F^2 + \lambda_1\|\mathbf{w}\|_1, \tag{2.10}$$

with $\|\mathbf{w}\|_1 = \sum_{i=1}^{p} |w_i|$ and λ_1 being a parameter determining the degree of the penalization.

In Algorithm 1, $\mathbf{w}$ is constrained to have unit length which makes it unsuitable to apply penalization directly on $\mathbf{w}$. Instead, it is more suitable to scale $\mathbf{z}$ to have unit length making $\mathbf{w}$ free of any scale constraints. As a last step in the outermost for-loop one can re-scale $\mathbf{w}$ to have unit length and $\mathbf{z}$ to be free of scale constraints. With $\mathbf{z}^T\mathbf{z} = 1$ Lemma A.1 gives that for a fixed $\mathbf{z}$ the $\mathbf{w}^*$ that minimizes equation (2.10) is:

$$\mathbf{w}^* = T_{\lambda_1}^{soft}(\mathbf{X}^T\mathbf{z}),$$

where $T_\lambda^{soft}(\mathbf{w}) = \left(t_\lambda^{soft}(w_1), ..., t_\lambda^{soft}(w_p)\right)^T$ and $t_\lambda^{soft}(w) = \text{sign}(w)\max(0, w - \lambda)$. The algorithm for sparse PCA looks as follows:

Algorithm 6 Sparse PCA NIPALS

procedure sPCA($\mathbf{X}$,r,λ_1,ϵ,m)
  $\mathbf{R} = \mathbf{X}$.
  for ($i = 1$,...,r) **do**
    $\delta = \infty$
    $\mathbf{z} = \mathbf{R}_i$
    for $j = 1$,...m **do**
      $\mathbf{z} = \frac{\mathbf{z}}{\|\mathbf{z}\|}$
      $\mathbf{w} = T_{\lambda_1}^{soft}(\mathbf{R}^T\mathbf{z})$
      $\mathbf{z} = \mathbf{R}\mathbf{w}$
      if $|\delta - \|\mathbf{z}\|| < \epsilon$ **then**
        break
      end if
      $\delta = \|\mathbf{z}\|$
    end for
    $\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
    $\mathbf{z} = \mathbf{R}\mathbf{w}$
    $\mathbf{W}_i = \mathbf{w}$
    $\mathbf{Z}_i = \mathbf{z}$
    $\mathbf{R} = \mathbf{R} - \mathbf{z}\mathbf{w}^T$
  end for
  return $\mathbf{W}$,$\mathbf{Z}$
end procedure

2.5.2 The generalized Fused Lasso

The Fused lasso was originally proposed by R. Tibshirani et al. in 2005 [9]. In a general setting the fused lasso problem can be formulated as:

$$\min_{\mathbf{w}} f(\mathbf{X},\mathbf{z},\mathbf{w}) + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{L}\mathbf{w}\|_1, \tag{2.11}$$

where $f(\mathbf{X},\mathbf{z},\mathbf{w})$ is any loss function of $\mathbf{X},\mathbf{z},\mathbf{w}$, $\lambda_1\|\mathbf{w}\|_1$ fills the same functionality as in (2.10) and the penalty term $\lambda_2\|\mathbf{L}\mathbf{w}\|_1$ penalizes the differences in the entries of $\mathbf{w}$ as specified by $\mathbf{L}$. The matrix $\mathbf{L}$ can be any $m \times p$-matrix and will specify the relationship between the coefficients $\mathbf{w}$. $\mathbf{L}$ could for example be a $p \times p$-matrix describing a graph relationship between the coefficients.

The optimization problem in (2.11) is difficult to solve because of the non-differentiability of the ℓ_1-norms. However, it can be efficiently solved using the split Bregmann method [10]. By a few steps the split Bregmann method reformulates (2.11) into a primal and a dual problem which can then be alternated between in order to find the optimal solution $\mathbf{w}^*$. Firstly, instead of formulating the optimization problem as an unconstrained problem, it can be formulated as a constrained optimization problem:

$$\begin{aligned} &\min_{\mathbf{w}} f(\mathbf{X},\mathbf{z},\mathbf{w}) + \lambda_1\|\mathbf{a}\|_1 + \lambda_2\|\mathbf{b}\|_1 \\ &\qquad \text{s.t} \\ &\mathbf{a} = \mathbf{w} \\ &\mathbf{b} = \mathbf{L}\mathbf{w}. \end{aligned} \tag{2.12}$$

The constrained version of the problem can be solved using the Lagrangian method of multipliers. The Lagrangian function of (2.12) is defined as:

$$\tilde{\mathcal{L}}(\mathbf{w},\mathbf{a},\mathbf{b},\mathbf{u},\mathbf{v}) = f(\mathbf{X},\mathbf{z},\mathbf{w}) + \lambda_1\|\mathbf{a}\|_1 + \lambda_2\|\mathbf{b}\|_1 + \mathbf{u}^T(\mathbf{w}-\mathbf{a}) + \mathbf{v}^T(\mathbf{L}\mathbf{w}-\mathbf{b}), \tag{2.13}$$

where $\mathbf{u}$ and $\mathbf{v}$ are $p \times 1$ and $m \times 1$ dual vectors for the constraints $\mathbf{a} = \mathbf{w}$ and $\mathbf{b} = \mathbf{L}\mathbf{w}$. However, the problem is more efficiently solved using the augmented Lagrangian function of (2.12). In the augmented Lagrangian function of (2.12) another two terms, penalizing the violation of $\mathbf{a} = \mathbf{w}$ and $\mathbf{b} = \mathbf{L}\mathbf{w}$, is added to (2.13):

$$\begin{aligned} \mathcal{L}(\mathbf{w},\mathbf{a},\mathbf{b},\mathbf{u},\mathbf{v}) =& f(\mathbf{X},\mathbf{z},\mathbf{w}) + \lambda_1\|\mathbf{a}\|_1 + \lambda_2\|\mathbf{b}\|_1 + \mathbf{u}^T(\mathbf{w}-\mathbf{a}) + \mathbf{v}^T(\mathbf{L}\mathbf{w}-\mathbf{b}) + \\ & \frac{\mu_1}{2}\|\mathbf{w}-\mathbf{a}\|_2^2 + \frac{\mu_2}{2}\|\mathbf{L}\mathbf{w}-\mathbf{b}\|_2^2, \end{aligned} \tag{2.14}$$

where $\mu_1, \mu_2 > 0$ are two parameters affecting the convergence rate of the method. The $\mathbf{w}^*$ that minimizes (2.11) will now satisfy the following inequality [12]:

$$\mathcal{L}(\mathbf{w}^*,\mathbf{a}^*,\mathbf{b}^*,\mathbf{u},\mathbf{v}) \leq \mathcal{L}(\mathbf{w}^*,\mathbf{a}^*,\mathbf{b}^*,\mathbf{u}^*,\mathbf{v}^*) \leq \mathcal{L}(\mathbf{w},\mathbf{a},\mathbf{b},\mathbf{u}^*,\mathbf{v}^*). \tag{2.15}$$

This inequality constraint can be solved by alternating between minimizing the primal function $\mathcal{L}(\mathbf{w},\mathbf{a},\mathbf{b},\mathbf{u}^*,\mathbf{v}^*)$ for fixed $\mathbf{u}^*,\mathbf{v}^*$ and maximizing the dual function $\mathcal{L}(\mathbf{w}^*,\mathbf{a}^*,\mathbf{b}^*,\mathbf{u},\mathbf{v})$ for fixed $\mathbf{w}^*,\mathbf{a}^*,\mathbf{b}^*$. Finding the estimates of the variables in time step $t+1$ is done by

first finding the solution to the primal problem and then finding the solution to the dual problem as follows:

$$\begin{aligned}(\mathbf{w}^{(t+1)},\mathbf{a}^{(t+1)},\mathbf{b}^{(t+1)}) &= \arg\min_{\mathbf{w},\mathbf{a},\mathbf{b}} \mathcal{L}(\mathbf{w},\mathbf{a},\mathbf{b},\mathbf{u}^{(t)},\mathbf{v}^{(t)}) \\ (\mathbf{u}^{(t+1)},\mathbf{v}^{(t+1)}) &= \arg\max_{\mathbf{u},\mathbf{v}} \mathcal{L}(\mathbf{w}^{(t+1)},\mathbf{a}^{(t+1)},\mathbf{b}^{(t+1)},\mathbf{u},\mathbf{v}).\end{aligned} \tag{2.16}$$

The solution to the dual problem is rather simple, since $\mathcal{L}$ is linear in $\mathbf{u}$ and $\mathbf{v}$, and can be found using gradient ascent. With the step parameters $\delta_1, \delta_2 > 0$, $\mathbf{u}$ and $\mathbf{v}$ are updated as follows:

$$\begin{aligned}\mathbf{u}^{(t+1)} &= \mathbf{u}^{(t)} + \delta_1(\mathbf{w}^{(t+1)} - \mathbf{a}^{(t+1)}) \\ \mathbf{v}^{(t+1)} &= \mathbf{v}^{(t)} + \delta_2(\mathbf{L}\mathbf{w}^{(t+1)} - \mathbf{b}^{(t+1)})\end{aligned} \tag{2.17}$$

The solution to the primal problem is slightly more complicated since it still contains the non-differentiable ℓ_1-terms. However, note that $\mathcal{L}$ does not contain any ℓ_1-terms involving $\mathbf{w}$ which means that the minimization of the primal problem can be split into three parts as follows:

$$\begin{aligned}\mathbf{w}^{(t+1)} &= \arg\min_{\mathbf{w}} f(\mathbf{X},\mathbf{z},\mathbf{w}) + \mathbf{u}^T(\mathbf{w}-\mathbf{a}) + \mathbf{v}^T(\mathbf{L}\mathbf{w}-\mathbf{b}) + \frac{\mu_1}{2}\|\mathbf{w}-\mathbf{a}\|_2^2 + \frac{\mu_2}{2}\|\mathbf{L}\mathbf{w}-\mathbf{b}\|_2^2 \\ \mathbf{a}^{(t+1)} &= \arg\min_{\mathbf{a}} \lambda_1\|\mathbf{a}\|_1 + \mathbf{u}^T(\mathbf{w}-\mathbf{a}) + \frac{\mu_1}{2}\|\mathbf{w}-\mathbf{a}\|_2^2 \\ \mathbf{b}^{(t+1)} &= \arg\min_{\mathbf{b}} \lambda_2\|\mathbf{b}\|_1 + \mathbf{v}^T(\mathbf{L}\mathbf{w}-\mathbf{b}) + \frac{\mu_2}{2}\|\mathbf{L}\mathbf{w}-\mathbf{b}\|_2^2\end{aligned} \tag{2.18}$$

Lemma A.2 can be applied to the minimization of $\mathbf{a}$ and $\mathbf{b}$ and gives that:

$$\begin{aligned}\mathbf{a}^{(t+1)} &= T^{soft}_{\lambda_1/\mu_1}(\mathbf{w}^{(t+1)} + \mathbf{u}^{(t)}/\mu_1) \\ \mathbf{b}^{(t+1)} &= T^{soft}_{\lambda_2/\mu_2}(\mathbf{L}\mathbf{w}^{(t+1)} + \mathbf{v}^{(t)}/\mu_2)\end{aligned} \tag{2.19}$$

The result of (2.17)-(2.19) is the split Bregmann method, which in this case is also the same as the alternating direction method of multipliers (ADMM) [13]. The algorithm is defined as follows:

Algorithm 7 Split Bregmann method for the generalized Fused Lasso problem

procedure SBFLasso($\mathbf{X}$,$\mathbf{z}$,λ_1,λ_2,μ_1,μ_2,δ_1,δ_2, ϵ)
  Initialize $\mathbf{a}^{(0)}$,$\mathbf{b}^{(0)}$,$\mathbf{u}^{(0)}$,$\mathbf{v}^{(0)}$.
  while not converged **do**
    $\mathbf{w}^{(t+1)} = \arg\min_{\mathbf{w}} f(\mathbf{X},\mathbf{z},\mathbf{w}) + \mathbf{u}^{(t)T}(\mathbf{w} - \mathbf{a}^{(t)}) + \mathbf{v}^{(t)T}(\mathbf{L}\mathbf{w} - \mathbf{b}^{(t)}) + \frac{\mu_1}{2}\|\mathbf{w} - \mathbf{a}^{(t)}\|_2^2 + \frac{\mu_2}{2}\|\mathbf{L}\mathbf{w} - \mathbf{b}^{(t)}\|_2^2$
    $\mathbf{a}^{(t+1)} = T^{soft}_{\lambda_1/\mu_1}(\mathbf{w}^{(t+1)} + \mathbf{u}^{(t)}/\mu_1)$
    $\mathbf{b}^{(t+1)} = T^{soft}_{\lambda_2/\mu_2}(\mathbf{L}\mathbf{w}^{(t+1)} + \mathbf{v}^{(t)}/\mu_2)$
    $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t)} + \delta_1(\mathbf{w}^{(t+1)} - \mathbf{a}^{(t+1)})$
    $\mathbf{v}^{(t+1)} = \mathbf{v}^{(t)} + \delta_2(\mathbf{L}\mathbf{w}^{(t+1)} - \mathbf{b}^{(t+1)})$
  end while
  return $\mathbf{w}^{(t+1)}$
end procedure

The initialization of $\mathbf{a}^{(0)}$,$\mathbf{b}^{(0)}$,$\mathbf{u}^{(0)}$,$\mathbf{v}^{(0)}$ is not discussed in the article by Gui-Bo Ye and Xiaohui Xie. In this thesis this is done by setting all entries to 0. By experimentation, other "smarter" initial values does not seem to improve the rate of convergence. The update of $\mathbf{w}^{(t+1)}$ depends directly on $f(\mathbf{X},\mathbf{z},\mathbf{w})$ and is therefore discussed in the next section where the split Bregmann method for the generalized Fused Lasso is applied to a concrete problem. In this thesis the parameters $\delta_1 = \mu_1$ and $\delta_2 = \mu_2$ will be used, as suggested by [10]. For further convergence properties of Algorithm 7 the reader is referred to the original article by Gui-Bo Ye and Xiaohui Xie.

2.5.3 Fused Lasso PCA (FLPCA)

In this section the application of generalized Fused Lasso to the loadings of a principal component analysis is derived. By (2.3) one gets that the Fused Lasso loss function in (2.11) is $f(\mathbf{X},\mathbf{z},\mathbf{w}) = \frac{1}{2}\|\mathbf{X} - \mathbf{z}\mathbf{w}^T\|_F^2$. In this thesis the focus will lie on a specific choice of penalization matrix $\mathbf{L}$, namely:

$$\mathbf{L} = \begin{pmatrix} -1 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & -1 & 1 & \dots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & -1 & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & -1 & 1 \end{pmatrix},$$

where $\mathbf{L}$ is a $(p-1) \times p$-matrix. With this loss function and choice of $\mathbf{L}$ the fused lasso problem for PCA becomes the following:

$$\begin{aligned}&\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{X} - \mathbf{z}\mathbf{w}^T\|_F^2 + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{L}\mathbf{w}\|_1 = \\ &\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{X} - \mathbf{z}\mathbf{w}^T\|_F^2 + \lambda_1\sum_{i=1}^{p}|w_i| + \lambda_2\sum_{i=2}^{p}|w_i - w_{i-1}|.\end{aligned} \tag{2.20}$$

As seen in (2.20) the choice of $\mathbf{L}$ will penalize the differences in subsequent entries in $\mathbf{w}$ and shrink the differences towards 0. This penalization will encourage smoothness in $\mathbf{w}$ and will fuse it to be a piecewise constant function for large enough λ_2.

By applying Lemma A.3, $\mathbf{w}^{(t+1)}$ in Algorithm 7 is found by solving the following system of linear equations:

$$((1+\mu_1)\mathbf{I} + \mu_2\mathbf{L}^T\mathbf{L})\mathbf{w} = \mathbf{X}^\mathbf{T}\mathbf{z} + (\mu_1\mathbf{a}^{(t)} - \mathbf{u}^{(t)}) + \mathbf{L}^T(\mu_2\mathbf{b}^{(t)} - \mathbf{v}^{(t)}) \tag{2.21}$$

With this specific choice of $\mathbf{L}$ the matrix $((1+\mu_1)\mathbf{I} + \mu_2\mathbf{L}^T\mathbf{L})$ in (2.21) is tridiagonal which means that (2.21) can be solved in the order of p iterations. This is very convenient for the running time of the algorithm. Also, using an efficient sparse matrix implementation the space complexity of the algorithm is limited to $O(np)$ with $\mathbf{X}$ being the limiting factor. Without a sparse matrix implementation the limiting factor would be $\mathbf{L}^T\mathbf{L}$ which would take up $O(p^2)$ space. This would not be practical for large p. Additionally, note that if $\mathbf{L}^T\mathbf{L}$ is not tridiagonal the space complexity of $O(np)$ is not guaranteed even with the use of a sparse matrix implementation.

As mentioned in previous section the convergence criterion of Algorithm 7 was not discussed in the original article. Since in this application $\mathbf{w}$ is free of any scale constraint, it is suitable to assume that Algorithm 7 has converged if $|\mathbf{w}^{(t+1)T}\mathbf{w}^{(t+1)} - \mathbf{w}^{(t)T}\mathbf{w}^{(t)}| < \epsilon$ is satisfied. Combining Algorithm 6, Algorithm 7 and (2.21) the algorithm for Fused Lasso PCA (FLPCA) becomes:

Algorithm 8 Fused Lasso PCA NIPALS

procedure FLPCA($\mathbf{X}$,r,λ_1,λ_2,μ_1,μ_2,ϵ_1,ϵ_2,m)
  $\mathbf{R} = \mathbf{X}$.
  for ($i = 1$,...,r) **do**
    $\delta = \infty$
    $\mathbf{z} = \mathbf{R}_i$
    for $j = 1$,...m **do**
      $\mathbf{z} = \frac{\mathbf{z}}{\|\mathbf{z}\|}$
      $\mathbf{w} = \text{SBFLasso}(\mathbf{X},\mathbf{z},\lambda_1,\lambda_2,\mu_1,\mu_2,\mu_1,\mu_2,\epsilon_2)$
      $\mathbf{z} = \mathbf{R}\mathbf{w}$
      if $|\delta - \|\mathbf{z}\|| < \epsilon_1$ **then**
        break
      end if
      $\delta = \|\mathbf{z}\|$
    end for
    $\mathbf{w} = \frac{\mathbf{w}}{\|\mathbf{w}\|}$
    $\mathbf{z} = \mathbf{R}\mathbf{w}$
    $\mathbf{W}_i = \mathbf{w}$
    $\mathbf{Z}_i = \mathbf{z}$
    $\mathbf{R} = \mathbf{R} - \mathbf{z}\mathbf{w}^T$
  end for
  return $\mathbf{W}$,$\mathbf{Z}$
end procedure

2.6 Model selection

In this section two different model selection problems will be discussed. Firstly, the problem of selecting the ranks $r, r_1, \dots r_I$ for the JIVE decomposition. Secondly, the problem of selecting the penalization parameter λ_1 for sJIVE and FLJIVE and penalization parameter λ_2 for FLJIVE.

2.6.1 Rank selection

A challenge with all supervised methods is validating them, and this is no exception for JIVE and JIC. As a matter of fact, as JIVE and JIC are both very recent methods no consensus solution to this problem has yet been agreed on in the literature. The main goal of the validation of these two methods is finding the correct ranks r,$r_1, \dots, r_I$. In this section a novel validation method, inspired by consensus clustering [11], for finding the correct ranks r,$r_1, \dots, r_I$ is presented. The validation method is based on clustering sub-samples of the original data set.

Let $\boldsymbol{X}^{(1)}, \dots, \boldsymbol{X}^{(L)}$ denote L sub-samples of the original data set $\boldsymbol{X}$ where each $\boldsymbol{X}^{(l)}$

contains all the features for a random subset of the rows of $\boldsymbol{X}$. Define $\mathcal{I}^{(l)}$, the indicator matrix of sub-sample $\boldsymbol{X}^{(l)}$, as:

$$\mathcal{I}_{ij}^{(l)} = \begin{cases} 1 & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}^{(l)} \\ 0 & \text{else} \end{cases} \tag{2.22}$$

The purpose of the indicator matrix $\mathcal{I}^{(l)}$ is to keep track of which pairs of objects are present in each sub-sample. This is necessary since the sub-samples contains a subset of the rows of $\boldsymbol{X}$, and therefore not all rows will be present in each sub-sample. Also, let $\mathcal{C}^{(l)}$ denote the $n \times n$ connectivity matrix for sub-sample $\boldsymbol{X}^{(l)}$. Given a clustering $C(\boldsymbol{X}^{(l)}) = \left(C(\boldsymbol{x}_1^{(l)}), \ldots, C(\boldsymbol{x}_{n_l}^{(l)})\right)$ the connectivity matrix $\mathcal{C}^{(l)}$ is defined as:

$$\mathcal{C}_{ij}^{(l)} = \begin{cases} 1 & \text{if object } i \text{ and } j \text{ are assigned to the same cluster in } C(\boldsymbol{X}^{(l)}) \\ 0 & \text{else} \end{cases} \tag{2.23}$$

The consensus matrix $\mathcal{C}$ is then formed by counting the number of times i and j have been clustered together and dividing it by the number of times i and j have been included in the same sub-sample. In mathematical terms the consensus matrix is defined as:

$$\mathcal{C}_{ij} = \frac{\sum_{l=1}^{L} \mathcal{C}_{ij}^{(l)}}{\sum_{l=1}^{L} \mathcal{I}_{ij}^{(l)}} \tag{2.24}$$

In a perfect scenario $\mathcal{C}$ would only contain 0 or 1 entries, and if the objects were ordered by their cluster-belongings the consensus matrix would be block diagonal with the blocks' entries equal to 1 and the rest of the entries equal to 0. However, in practise it is highly unlikely that a perfect consensus matrix occurs, and comparing two consensus matrices is not trivial. Therefore, there is a need for a consensus summary statistic which can be used to compare two, or more, different consensus matrices. In this thesis a completely new consensus statistic is derived to fit this specific rank selection problem. Future work will be to compare this new statistic to the consensus summary statistic suggested by S. Monti et al. [11] as well as to other statistics.

The new statistic is motivated by histograms over the values for different consensus matrices. A key observation for the statistic proposed in this thesis is that as the rank is more and more overfitted, the median of values larger than 0.5 starts to move from 1 to 0.5. Figure 2.1 shows histograms for $\mathcal{C}_J^2, \ldots, \mathcal{C}_J^9$ ($\mathcal{C}_J^r$ corresponds to the joint consensus matrix from rank r) where 4 is the correct rank. In the figure, this phenomenon is demonstrated rather clearly where the median of values larger than 0.5 is significantly less for $\mathcal{C}_J^5$ than for $\mathcal{C}_J^4$.

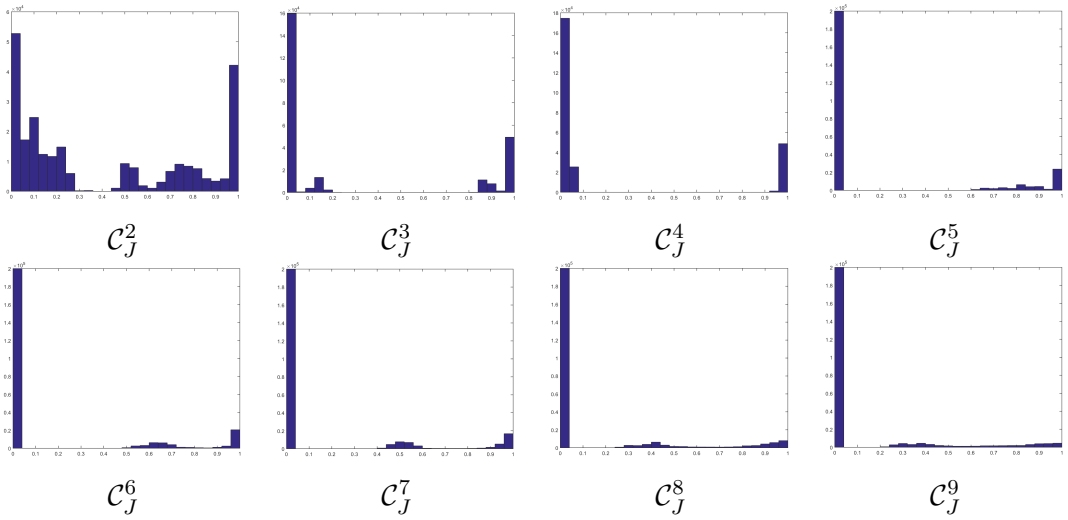


Figure 2.1: Showing histogram over the values for examples of $\mathcal{C}_J^2, \ldots, \mathcal{C}_J^9$ where the correct rank is 4. As seen in the figure, $\mathcal{C}_J^4$ is closest to containing only zeroes and ones.

Another indicator of the rank being set wrong is the presence of multiple values between 0 and 1. For $\mathcal{C}_J^2$ and $\mathcal{C}_J^3$ in Figure 2.1 this is clearly demonstrated. It is also sensible to assume that having values closer to 0.5 is worse than having values closer to 1 or 0. With $0 \leq v_1 \leq v_2 \leq 1$ define:

$$\begin{aligned} \beta_{\min} &= \min_{v_1 \leq \mathcal{C}_{ij} \leq v_2} B(\mathcal{C}_{ij}, \alpha, \beta) \\ \beta_{\max} &= \max_{v_1 \leq \mathcal{C}_{ij} \leq v_2} B(\mathcal{C}_{ij}, \alpha, \beta), \end{aligned} \tag{2.25}$$

where $B(x, \alpha, \beta)$ is the beta pdf-function for x with parameters α and β. The two indicators of bad fit can be combined into the following statistic which should be minimized:

$$T(\mathcal{C}) = \left(1 - \underset{\mathcal{C}_{ij} \geq m_1}{\text{median}}\left(\mathcal{C}_{ij}\right)\right) \sum_{v_1 \leq \mathcal{C}_{ij} \leq v_2} \left(1 - \frac{B(\mathcal{C}_{ij}, \alpha, \beta) - \beta_{\min}}{\beta_{\max} - \beta_{\min}}\right), \tag{2.26}$$

where $0 < m_1, \alpha, \beta < 1$ are parameters that can be experimented with.

It is difficult to derive a statistic that works when the correct rank is small, the correct rank is large, the clusters have a hierarchical structure and in every other possible situation. Also, in practice the correct rank will most likely not have a perfect distribution of only zeros and ones. Even though this is a difficult problem, setting $v_1 = 0.1, v_2 = 0.9, m_1 = 0.5, \alpha = 0.1, \beta = 0.1$ seems to give good results in most cases and will therefore be used throughout this thesis.

The proposed Consensus Rank selection algorithm for JIVE and JIC can be seen in Algorithm 9. On line 4 of the algorithm L sub-samples, $\boldsymbol{X}^{(l)}$, are sampled, and for each sub-sample the indicator matrix (line 5) is updated. JIVE is then run with the ranks $r, r_1, \ldots, r_I$ on line 6. On line 8-12 k-means is run with $K = j + 1$, for $j = 1, \ldots, r$ using only the j first columns of the joint scores, $Z_{J[j]}$, as input. The same procedure is done for the individual scores, $Z_{A_1[j]}, \ldots, Z_{A_I[j]}$ on line 14-20. On line 10-11 and 17-18 one can see how the result of k-means is used to form the joint and individual connectivity matrices for each sub-sample $\boldsymbol{X}^{(l)}$. The connectivity matrices is then divided with the indicator matrix, using the element-wise division operation "./", on line 23-31 in order to form the joint and individual consensus matrices. Finally, the consensus statistic (2.26) for each consensus matrix is returned. The consensus statistics can then be plotted to determine the correct rank.

The consensus rank selection algorithm is best applied in a two-step procedure. Since the joint and individual components are independent of each other, one can estimate the joint rank by fixing the individual ranks to be 0 in the first run of the rank selection algorithm. In the second step one can fix the joint rank to the suggested value from the first run and in that way find the correct individual ranks. The first step would be to call the algorithm with the parameters $\mathbf{X}$,r,0,...,0,L,m. With r^* being the joint rank suggested in the first step, the second step is to call the algorithm with the parameters $\mathbf{X}$,r^*,r_1,...,r_I,L,m.

Algorithm 9 Consensus Rank selection

procedure CRS($\mathbf{X}$,r,$r_1,\ldots,r_I$,L,m)
  $\mathcal{I}, \mathcal{C}_J^1,\ldots,\mathcal{C}_J^r, \mathcal{C}_{A_1}^1,\ldots,\mathcal{C}_{A_1}^{r_1},\ldots,\mathcal{C}_{A_I}^1,\ldots,\mathcal{C}_{A_I}^{r_I} = \mathbf{0}_{n\times n}$
  for $l = 1,\ldots,L$ **do**
    $\boldsymbol{X}^{(l)}$ = sub-sample($\boldsymbol{X}$,m) // $0 < m < 1$
    $\mathcal{I}_{ij} = \begin{cases} \mathcal{I}_{ij} + 1 & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \in \boldsymbol{X}^{(l)} \\ \mathcal{I}_{ij} + 0 & \text{else} \end{cases}$
    W_J,Z_J,W_{A_1},Z_{A_1},$\ldots$,W_{A_I},$Z_{A_I} = JIVE(\boldsymbol{X}^{(l)}$,$r$,$r_1,\ldots,r_I)$

    for $j = 1,\ldots,r$ **do**
      C = k-means($Z_{J[j]}$,$j+1$)
      $\mathcal{C}_{ij}^{(l)} = \begin{cases} 1 & \text{if object } i \text{ and } j \text{ are assigned to the same cluster in } C \\ 0 & \text{else} \end{cases}$
      $\mathcal{C}_J^j = \mathcal{C}_J^j + \mathcal{C}^{(l)}$
    end for

    for $i = 1,\ldots,I$ **do**
      for $j = 1,\ldots,r_i$ **do**
        C = k-means($Z_{A_i[j]}$,$j+1$)
        $\mathcal{C}_{ij}^{(l)} = \begin{cases} 1 & \text{if object } i \text{ and } j \text{ are assigned to the same cluster in } C \\ 0 & \text{else} \end{cases}$
        $\mathcal{C}_{A_i}^j = \mathcal{C}_{A_i}^j + \mathcal{C}^{(l)}$
      end for
    end for
  end for

  for $j = 1,\ldots,r$ **do**
    $\mathcal{C}_J^j = \mathcal{C}_J^j./\mathcal{I}$
  end for

  for $i = 1,\ldots,I$ **do**
    for $j = 1,\ldots,r_i$ **do**
      $\mathcal{C}_{A_i}^j = \mathcal{C}_{A_i}^j./\mathcal{I}$
    end for
  end for

  return $T(\mathcal{C}_J^1),\ldots,T(\mathcal{C}_J^r),T(\mathcal{C}_{A_1}^1),\ldots,T(\mathcal{C}_{A_1}^{r_1}),\ldots,T(\mathcal{C}_{A_I}^1),\ldots,T(\mathcal{C}_{A_I}^{r_I})$
end procedure

2.6.2 Selecting the penalization parameters λ_1, λ_2

As stated in the Limitations section (section 1.3) the selection of the parameters λ_1, for sPCA and FLJIVE, and λ_2, for FLJIVE, is done via visual guidance. Figure 2.2 shows an example of the first joint loading found by JIVE (green), sJIVE (blue) and FLJIVE (red). By looking at the figure one can see that the JIVE's estimation of the first loading fluctuates around 0 for values between 600 and 900 on the x-axis. The parameter λ_1 is chosen such that sJIVE and FLJIVE successfully shrinks these values to 0 without affecting the other values too much. For values between 200 and 500 on the x-axis one can see that the values for JIVE and sJIVE fluctuates around the same value on y-axis. The parameter λ_2 is chosen such that these values are smoothed out without having too much impact on the rest of the fitted loading.

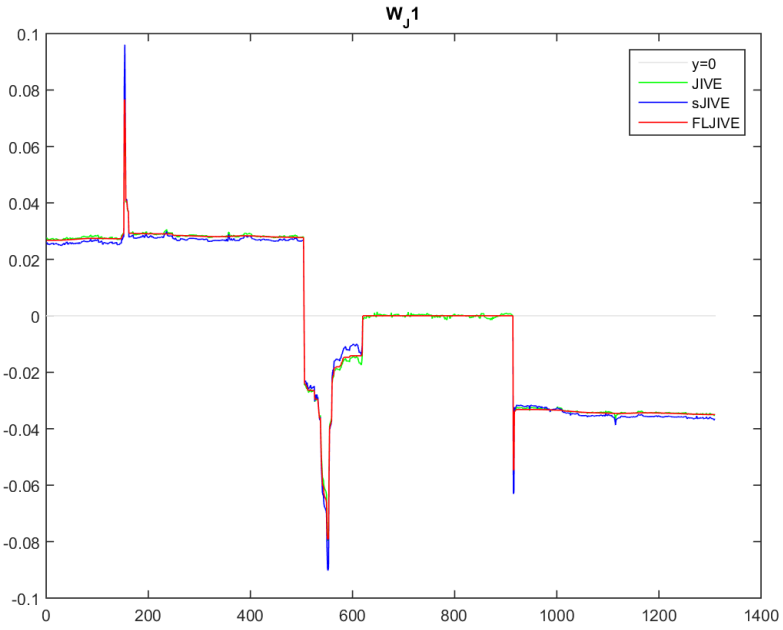


Figure 2.2: Showing an example of the first estimated joint loading for JIVE (green), sJIVE (blue) and FLJIVE (red) where the value of the loading (y-axis) is shown as a function of the genomic position (x-axis).

2.7 Visualization

All mathematics in this thesis assumes the data matrices to have the objects as rows and the corresponding features as columns ($n \times p$-matrices). The motivation behind this is that most literature assumes this form, and therefore, adapting this form would benefit the reader and improve the reader's ability to understand the methods. However, the visualization of the results from JIVE and JIC benefit from having the observations as columns and the corresponding features as rows. The main advantage is that joint

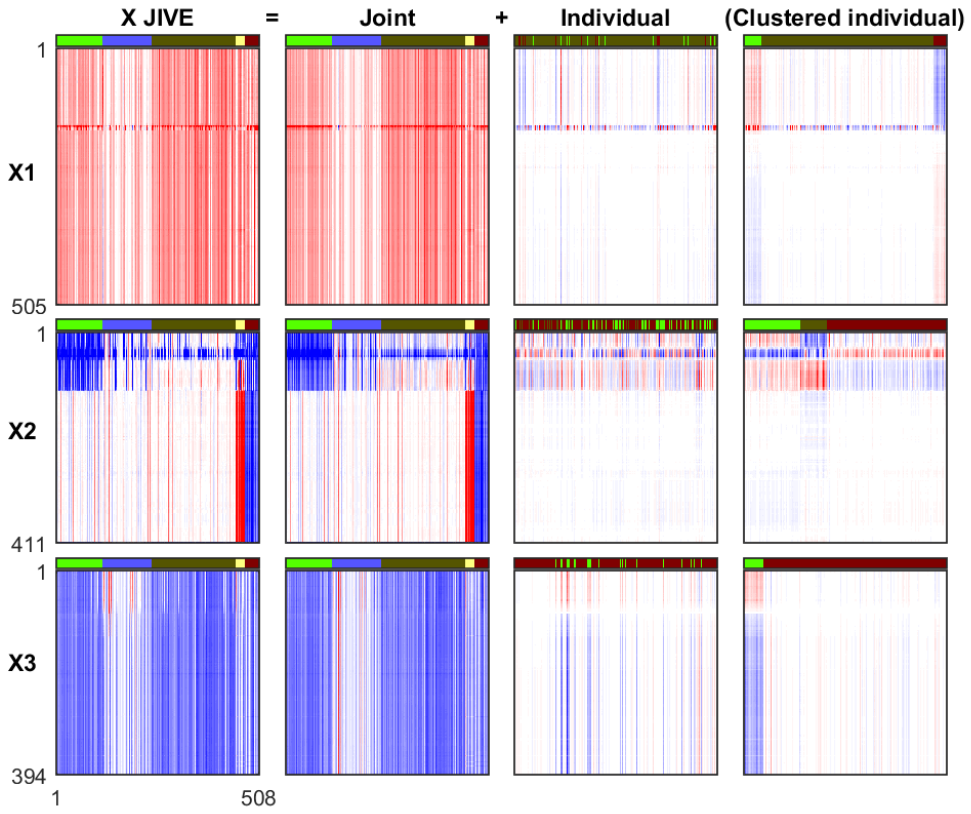


Figure 2.3: Showing an example of how the JIC result is visualized. Note that in the figure objects plotted along the columns (x-axis) while the corresponding features are plotted along the rows (y-axis).

clusters can more easily be seen and compared vertically by stacking the data types on top of each other.

Figure 2.3 shows an example of how the results from JIVE/JIC is visualized throughout the rest of the thesis. Note that in the figure the objects are plotted along the columns (x-axis) of the heatmaps and the features are plotted along the rows (y-axis) of the heatmaps. The first column of the figure shows the joint and individual components added together. The second and third columns show the joint and individual components. The three first columns are ordered according to the joint clustering. The last column shows the individual component re-ordered by it's own clustering. The actual clusters are represented by the color bars on top of each heatmap. The rows of the figure show the different data types where $X1$ correspond the first data type, $X2$ to the second data type and so on. In all plots, positive values (amplifications) are represented by red and negative values (deletions) are represented as blue. White corresponds to values equal to zero.

3

Simulation study

This section describes a study conducted on different simulated data sets. The goal of the simulation study is to compare the results of applying the consensus rank selection procedure with the different JIVE methods. The goal is also to compare the resulting fits of applying JIVE, sJIVE and FLJIVE to the simulated data sets, with the correct ranks, and to analyse how close the fits are to the original data. This section aims to provide evidence that FLJIVE exceeds sJIVE and JIVE when the data has underlying fused structure and to explore how FLJIVE compares to sJIVE and JIVE when the data has no underlying fused structure.

3.1 Data set creation

For the simulation study two different procedures are used in order to create simulated data sets. The first procedure will create a data set containing clusters where the underlying loadings have no fused properties. The second procedure will create a data set where the underlying joint loadings have fused properties, and where the data contains clusters. Both procedures create data sets according to the decomposed JIVE model (2.8).

3.1.1 No underlying fused PC loading

Given the number of objects n, features sizes p_1,...,p_I and the ranks $\boldsymbol{r} = (r, r_1, \ldots, r_I)$, the scores $\boldsymbol{Z}, \boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_I$ ($n \times r_i$-matrices) are created of the form $\boldsymbol{Z} = (\boldsymbol{z}_1, \ldots \boldsymbol{z}_r)$ and $\boldsymbol{Z}_i = (\boldsymbol{z}_1, \ldots \boldsymbol{z}_{r_i})$ where

$$\boldsymbol{z}_k^T = \frac{1}{\sqrt{n_k}} \left(0, \ldots, 0, \underbrace{1, \ldots, 1}_{n_k}, 0, \ldots, 0 \right)$$

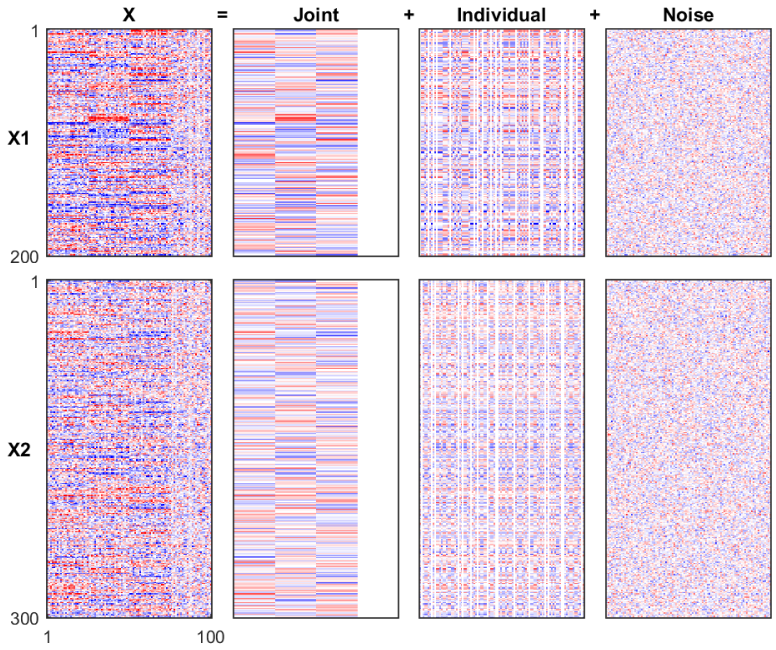


Figure 3.1: Showing a data set created as described in section 3.1.1 where $n = 100$, $p_1 = 200$, $p_3 = 200$, $r = 3$, $r_1 = 4$, $r_2 = 5$, $\sigma = 0.05$. For each heatmap in the figure, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.

and $n_k = \frac{n}{r+1}$. In this way the first column of $\boldsymbol{Z}$ will have ones on the first $\frac{n}{r+1}$ rows, the second column will have ones on the next $\frac{n}{r+1}$ rows and so on. The last $\frac{n}{r+1}$ rows of $\boldsymbol{Z}$ will have zeros for all columns and can in that way be identified as the last cluster. As the individual components must be independent of the joint component, the rows of $\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_I$ are permuted with one random permutation for each $\boldsymbol{Z}_i$.

The loadings $\boldsymbol{W}_1, \ldots \boldsymbol{W}_I$ and $\boldsymbol{V}_1, \ldots \boldsymbol{V}_I$ ($p_i \times r_i$-matrices) are taken to be the loadings from principal component analyses of equally many standard normal distributed matrices. In this way the created $\boldsymbol{Z}$s and $\boldsymbol{W}$s fulfil the PCA criteria of orthonormal loadings and orthogonal scores and the JIVE criterion of individual and joint components being orthogonal. The actual data sets are then created as (2.8) where $\epsilon_1, \ldots, \epsilon_I$ are normal distributed matrices with zero mean and σ standard deviation. An example of a data set created with this procedure can be seen in Figure 3.1. The data set in the figure was created using $n = 100$, $p_1 = 200$, $p_3 = 200$, $r = 3$, $r_1 = 4$, $r_2 = 5$, $\sigma = 0.05$. In the joint structure one can clearly see the 4 clusters ($r + 1$ clusters), but in the individual components the objects have been permuted and the clusters are not visibly clear. The individual clusters can be retrieved by clustering the individual components.

3.1.2 Underlying fused PC loading

This procedure creates the data set very similarly to the procedure in the subsection above. The only difference is that the joint loadings $\boldsymbol{W}_1, \dots \boldsymbol{W}_I$ are now created to have fused properties. The $\boldsymbol{W}_i$'s are created on the form $\boldsymbol{W}_i = (\boldsymbol{w}_1, \dots \boldsymbol{w}_{r_i})$ where

$$\boldsymbol{w}_k^T = \frac{1}{\sqrt{p^k}} \left(0, \dots, 0, \underbrace{1, \dots, 1, -1, \dots, -1}_{p^k}, 0, \dots, 0 \right)$$

and $p^k = \frac{p_i}{r_i}$. Differently from $\boldsymbol{Z}$, all rows of $\boldsymbol{W}$ have non-zero entries for one of the columns. The matrix $\boldsymbol{J}_i = \boldsymbol{Z}\boldsymbol{W}_i^T$ will now be very sparse but also have fused properties. The reason why the individual components are not created in this way is that they cannot be made truly independent of the joint component when being really sparse. This problem is due to the fact that a lack of signal, due to sparsity, is in fact a signal as well, and if both the joint and individual components are very sparse, overlap between them cannot be avoided. Therefore, the individual components are created as in previous subsection. An example of a data set created using this procedure, with $n = 50$, $p_1 = 100$, $p_3 = 200$, $r = 5$, $r_1 = 3$, $r_2 = 5$, $\sigma = 0.05$, can be seen in Figure 3.2.

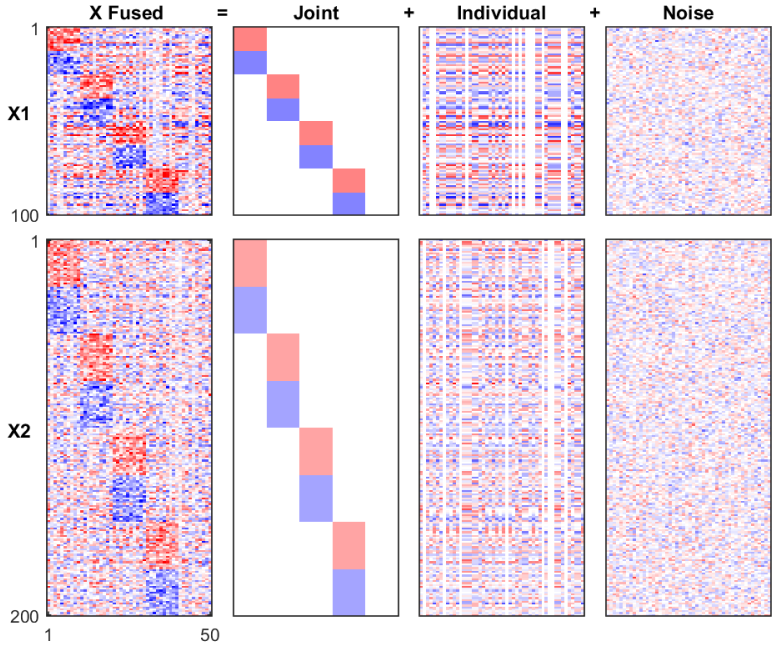


Figure 3.2: Showing a simulated data set, as described in section 3.1.2, where the joint loadings have fused properties. The data set was created with $n = 50$, $p_1 = 100$, $p_3 = 200$, $r = 5$, $r_1 = 3$, $r_2 = 5$, $\sigma = 0.05$. For each heatmap in the figure, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.

3.2 Rank selection study

In this section the consensus rank selection procedure, with JIVE, sJIVE and FLJIVE, is applied to different simulated data sets. In the first subsection the simulated data sets will have no fused properties, and in the second subsection the rank selection procedure is applied to data with fused joint structure. Common for all subsections in this rank selection study is that the consensus rank selection algorithm will be run on 10 different data sets for each set of ranks. Each data set will be sub-sampled 100 times where each sub-sample is created by drawing 90% of the original data set's objects randomly.

3.2.1 Data with no underlying fused PC loading

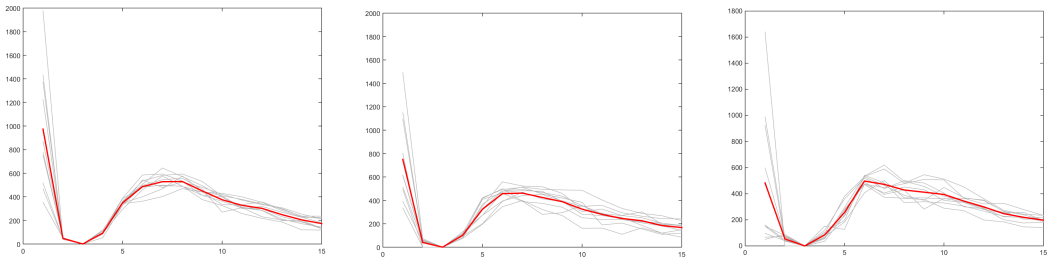
Given a vector of ranks $\boldsymbol{r}$ the simulated data sets are created as described in section 3.1.1. As described in section 2.6.1, the consensus rank selection algorithm is applied in a two-step procedure where the first step aims to find the joint rank r by setting $r_1, \ldots, r_I = 0$. The second step fixates r to the rank suggested by the first step and then aims to find the individual ranks.

Rank setup $\boldsymbol{r}_1 = (r = 3, r_1 = 4, r_2 = 5)$

Figure 3.3 shows the consensus validation statistic, for JIVE, sJIVE and FLJIVE, as a function of the joint rank r when the individual ranks are set to 0. The red line in the figure corresponds to the average statistic over 10 trials, and the grey lines represents the statistic for each trial. From the figure it is clear that the rank selection procedure succeeds in finding the correct rank $r = 3$ using all three methods.

Given that $r = 3$, the rank selection procedure is run again in order to find the individual ranks. Figure 3.4 shows the statistic as a function of the first individual rank r_1. The top three plots in the figure show the entire curves while in the bottom three plots the curves have been zoomed in order to see the minima clearer. Both JIVE and FLJIVE have minima at $r_1 = 4$, which is the correct rank. For sparse JIVE (sJIVE) the minimum occurs at $r_1 = 3$ and $r_1 = 4$. In this case the higher rank should be favoured since the statistic is by purpose defined to penalize over-fits harder than under-fits.

In Figure 3.5 the consensus statistic for r_2 can be seen. Looking at (d), (e) and (f) in the figure, one can see that the minima at $r_2 = 5$ is clearest for FLJIVE since both JIVE and sJIVE seem to have minima at $r_2 = 4$ and $r_2 = 5$.



(a) (b) (c)

Figure 3.3: Showing the consensus validation statistic (y-axis) as a function of the joint component rank, r, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c). The grey lines correspond to the statistic for each of the ten trials, and the red line shows the mean of the ten trials. The correct rank is 3 which is also where the minima occurs in the three plots.

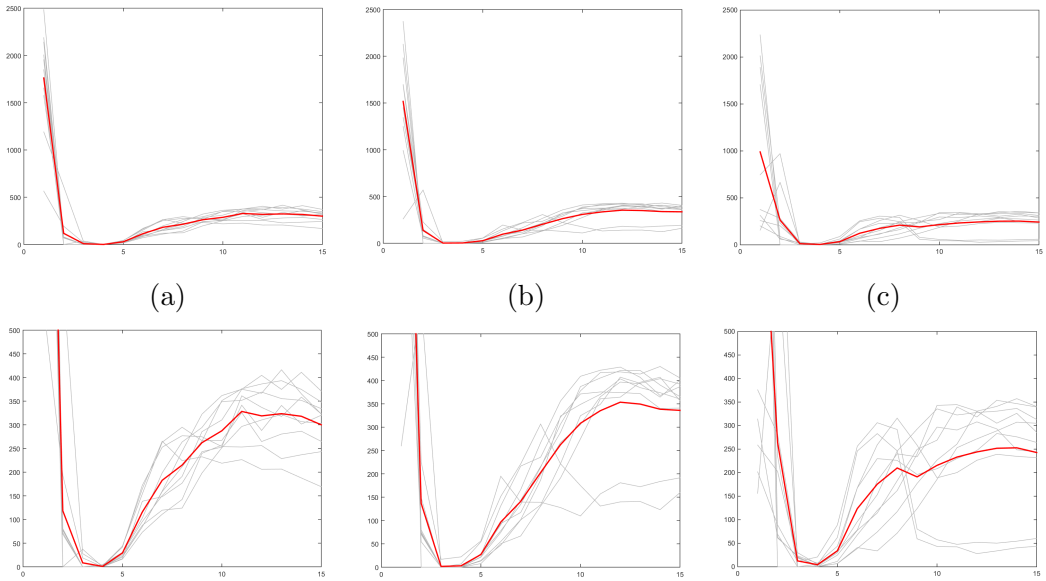


Figure 3.4: Showing the consensus validation statistic (y-axis) as a function of the first individual component rank, r_1, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c). The grey lines correspond to the statistic for each of the ten trials, and the red line shows the mean of the ten trials. The correct rank is 4 and is found by looking at the zoomed plots (d), (e) and (f).

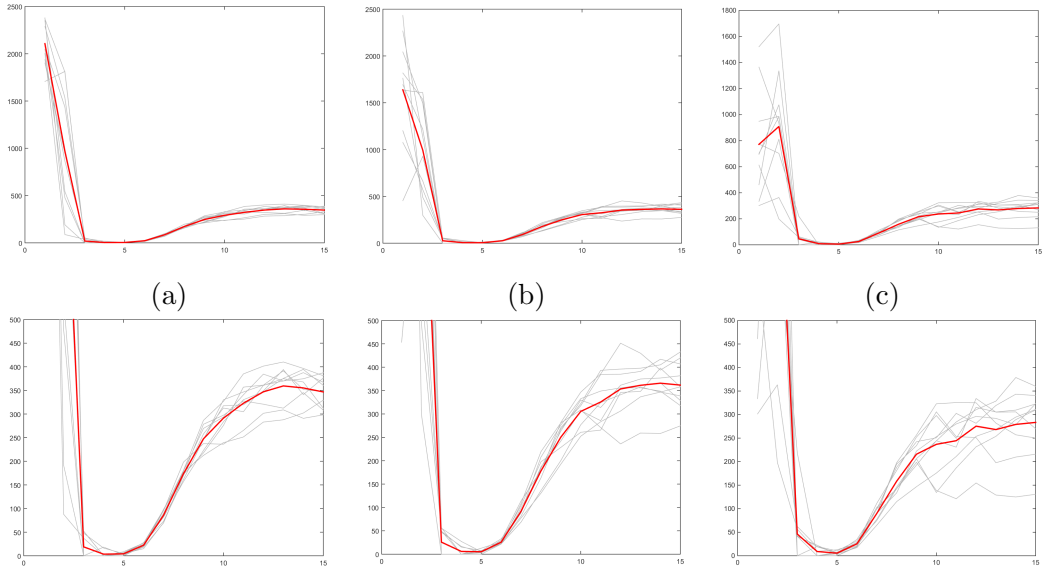


Figure 3.5: Showing the consensus validation statistic (y-axis) as a function of the second individual component rank, r_2, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c). The grey lines correspond to the statistic for each of the ten trials, and the red line shows the mean of the ten trials. The correct rank is 5 which can be found by looking at the zoomed plots (d), (e) and (f).

Rank setup $\boldsymbol{r}_2 = (r = 9, r_1 = 11, r_2 = 7)$

In Figure 3.6, the consensus rank selection statistic as a function of the joint rank r can be seen. As before, the individual ranks have been set to 0 when first trying to identify the joint rank. In the figure, (d), (e) and (f) are zoomed in versions of (a), (b) and (c). In the zoomed in versions one can see that the rank selection algorithm is successful in finding the correct rank $r = 9$ for all three methods.

Figure 3.7 and 3.8 shows plots over the statistic for the individual ranks r_1 and r_2 when r have been fixed to 9. In Figure 3.7 (d), (e) and (f), one can see that the minimum is clearest for JIVE, which is also finds the correct rank. Both sJIVE and FLJIVE seem to have two minima at $r_1 = 10$ and $r_1 = 11$, and as motivated previously, the higher rank should be favoured in this case. For the second individual rank, JIVE is again the method which gives the clearest correct minimum at $r_2 = 7$ which can be seen by looking at (d), (e) and (f) in Figure 3.8. Sparse JIVE seem to have two minima at $r_2 = 6$ and $r_2 = 7$ for which $r_2 = 7$ should be favoured. In (f) one can see that FLJIVE does in fact suggest that $r_2 = 6$ is the correct rank even though $r_2 = 7$ is not much worse.

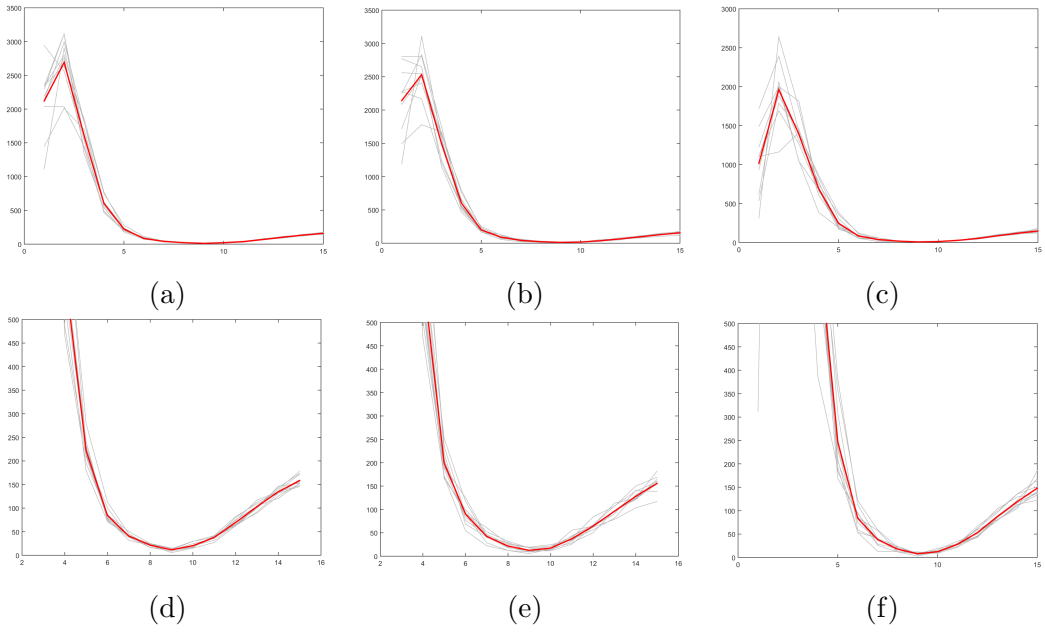


Figure 3.6: Showing the consensus validation statistic (y-axis) as a function of the joint component rank, r, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c). The grey lines correspond to the statistic for each of the ten trials, and the red line shows the mean of the ten trials. The correct rank is 9 which is also where the minima occurs in all three curves. Zoomed versions of (a), (b) and (c) are shown in (d), (e) and (f).

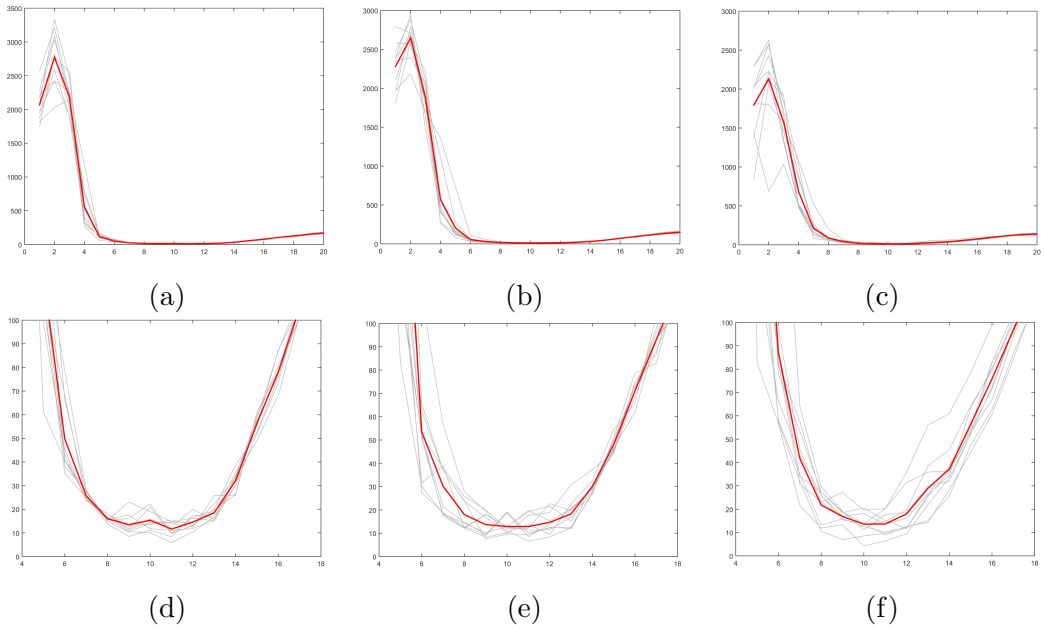


Figure 3.7: Showing the consensus validation statistic (y-axis) as a function of the first individual component rank, r_1, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c). The grey lines correspond to the statistic for each of the ten trials, and the red line shows the mean of the ten trials. The correct rank is 11 which, by looking at (d), (e) and (f), is found by JIVE. Sparse JIVE and FLJIVE have two subsequent minima for which the higher should be favoured

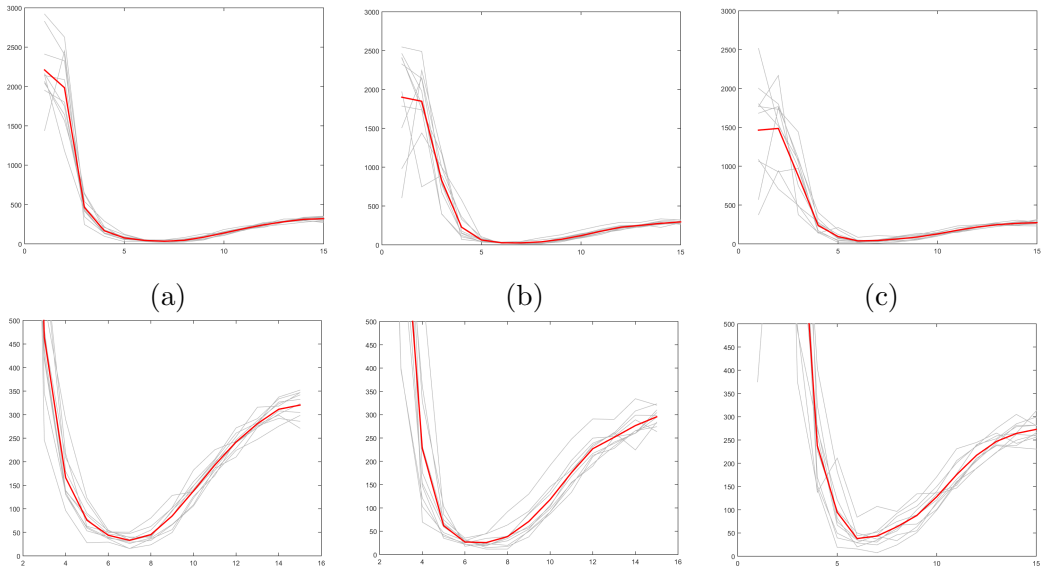


Figure 3.8: Showing the consensus validation statistic (y-axis) as a function of the second individual component rank, r_2, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c). The grey lines correspond to the statistic for each of the ten trials, and the red line shows the mean of the ten trials. The correct rank is 7 which, by looking at the zoomed plots (d), (e) and (f), is only found by JIVE. Sparse JIVE have two subsequent minima, for which the higher should be favoured, and FLJIVE suggests that 6 is the correct rank.

3.2.2 Data with underlying fused PC loading

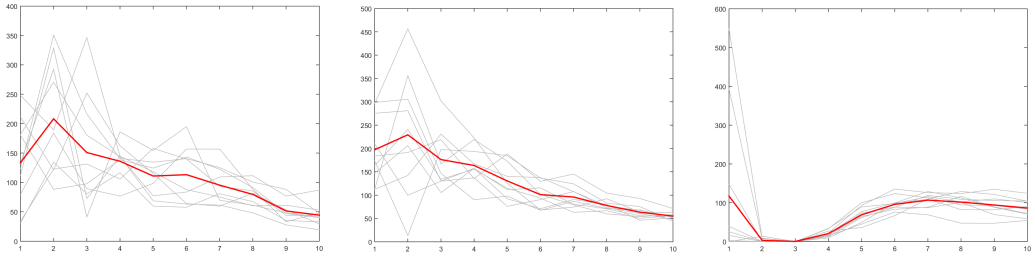
The data used in this study have fused loadings only for the joint components as described in section 3.1.2. However, it is enough for the joint component to have fused properties in order to prove the concept of FLJIVE exceeding sJIVE and JIVE when the data have such underlying properties. The argument can be extended to situations where also the individual components show fused properties.

Rank setup $\boldsymbol{r}_3 = (r = 3, r_1 = 5, r_2 = 7)$

As in previous subsection, the rank selection algorithm is first run with the individual ranks set to 0. Figure 3.9 shows the consensus statistic as a function of the joint rank r. In (a) and (b) in the figure it is clear that JIVE and sJIVE fails to find the correct rank. In (c) one can see that in this case FLJIVE is superior compared to the two other methods since it has no problem finding the correct rank $r = 3$.

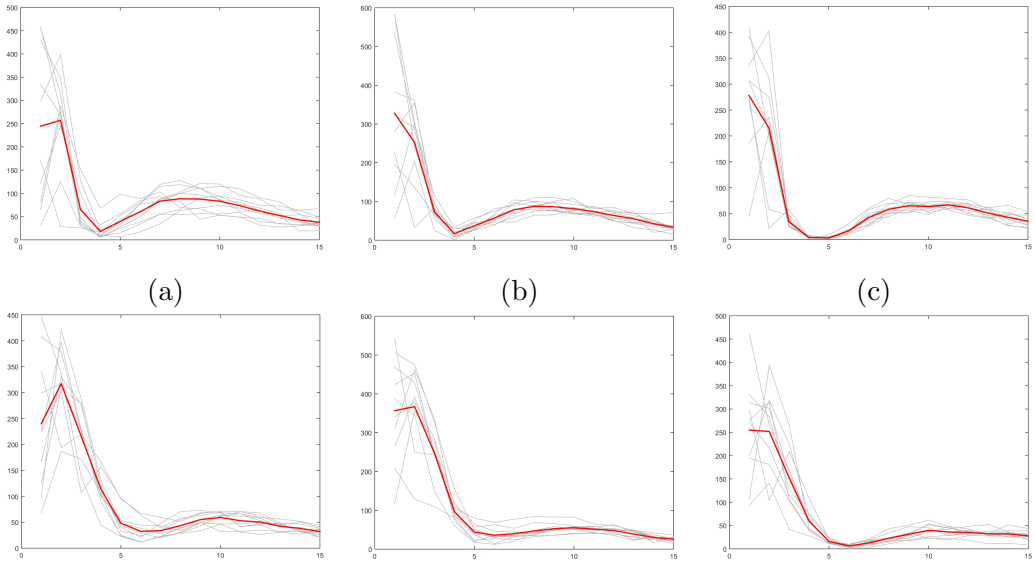
The rank selection algorithm is then run again with $r = 3$ being fixed. For JIVE, sJIVE and FLJIVE the corresponding PCA-methods are used to approximate the joint structure with $r = 3$. However, as the underlying individual structures have no fused properties the ordinary PCA is used to find the individual structures in the rank procedure. This is equivalent to setting the penalization parameters $\lambda_1, \lambda_2 = 0$ for the individual components in the sJIVE and FLJIVE methods. This can be seen as hybrid versions between sJIVE/JIVE and FLJIVE/JIVE.

Figure 3.10 shows the result for this procedure for both the first and second individual component. In (a) and (b) one can see that JIVE and the hybrid version sJIVE/JIVE suggests that the rank for the first individual component, r_1, is 4. As seen in (c) only FLJIVE/JIVE finds the correct rank which is 5. In (d), (e) and (f) of Figure 3.10 it is rather clear that all three methods, JIVE, sJIVE/JIVE, FLJIVE/JIVE, fails to find the correct rank for the second individual component, $r_2 = 7$, since they all suggests that 6 is optimal.



(a) (b) (c)

Figure 3.9: Showing the consensus validation statistic (y-axis) as a function of the joint component rank, r, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c). The grey lines correspond to the statistic for each of the ten trials, and the red line shows the mean of the ten trials. The correct rank is 3 which is only found by FLJIVE.



(a) (b) (c)

(d) (e) (f)

Figure 3.10: Showing the consensus validation statistic (y-axis) as a function of the first individual component rank, r_1, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c) and for the second individual component rank, r_2, for JIVE (d), sJIVE (e) and FLJIVE (f). The grey lines correspond to the statistic for each of the ten trials, and the red line shows the mean of the ten trials. The correct rank for the first individual component rank is 5 and for the second it is 7.

3.3 Estimation study

In this section the actual fit of the underlying components will be reviewed given that the correct ranks are known. The rank setups are the same as in the previous section.

3.3.1 Data with no underlying fused PC loading

Rank setup $\boldsymbol{r}_1 = (r = 3, r_1 = 4, r_2 = 5)$

Figure 3.11 shows the simulated data set, with $n = 100, p_1 = 200, p_3 = 300, r = 3, r_1 = 4, r_2 = 5, \sigma = 0.05$, on which the first estimation study will be performed. JIVE, sJIVE and FLJIVE will be used to estimate the underlying components of the data set. The estimated joint and individual components are then compared to the true components.

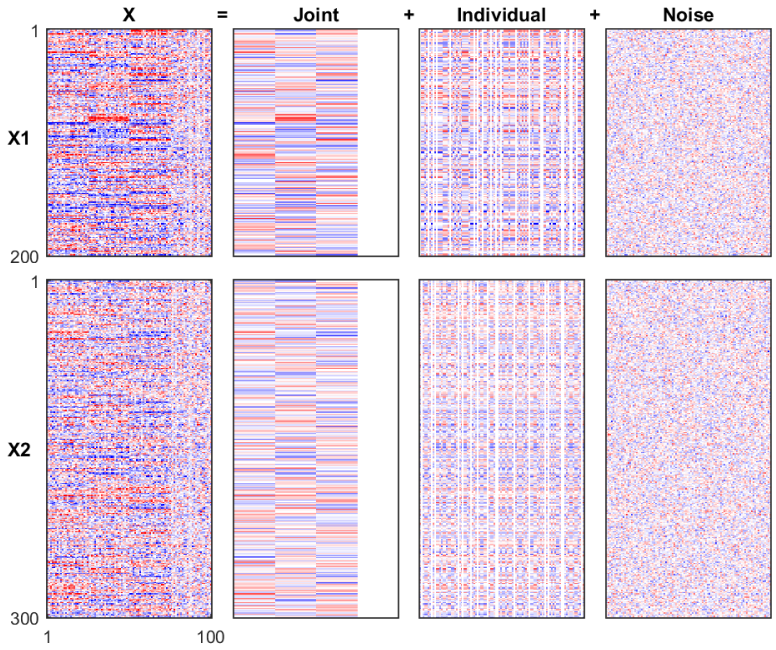


Figure 3.11: Showing the simulated data set used for the estimation study for the rank setup $\boldsymbol{r}_1 = (r = 3, r_1 = 4, r_2 = 5)$. For each heatmap in the figure, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.

The three methods, JIVE, sJIVE and FLJIVE, were applied to the data set in Figure 3.11, and the result of this can be seen in Figure 3.12. In (a) in the figure, the true data set with only the joint and individual components is shown (noise component not added). Visually, there are small differences, which can be hard to spot, between the fit of JIVE (b) and sJIVE (c). In (d) one can see that the fit of FLJIVE deviates even more

than sJIVE from the fit of JIVE. Looking carefully at the figure one can determine that JIVE actually provides the best fit in this scenario while the fit of FLJIVE is the worst.

In order to compare the fits more formally, the squared Frobenious norm of the differences between the fitted components and the true components are shown in Table 3.1. As suggested by looking at Figure 3.12, JIVE is the best at finding the true underlying components with the differences 31.45, 18.78 and 15.88 for $\boldsymbol{J}, \boldsymbol{A}_1\boldsymbol{A}_2$. The second best method is sJIVE with the corresponding differences 34.22, 22.03 and 17.91, and worst of the methods is FLJIVE with 54.63, 26.02 and 24.44. In this simulated setting FLJIVE performs on average 60% worse than JIVE while sJIVE performs only 12% worse.

Figures of the fitted joint loadings and scores of the three methods can be seen in Appendix B.

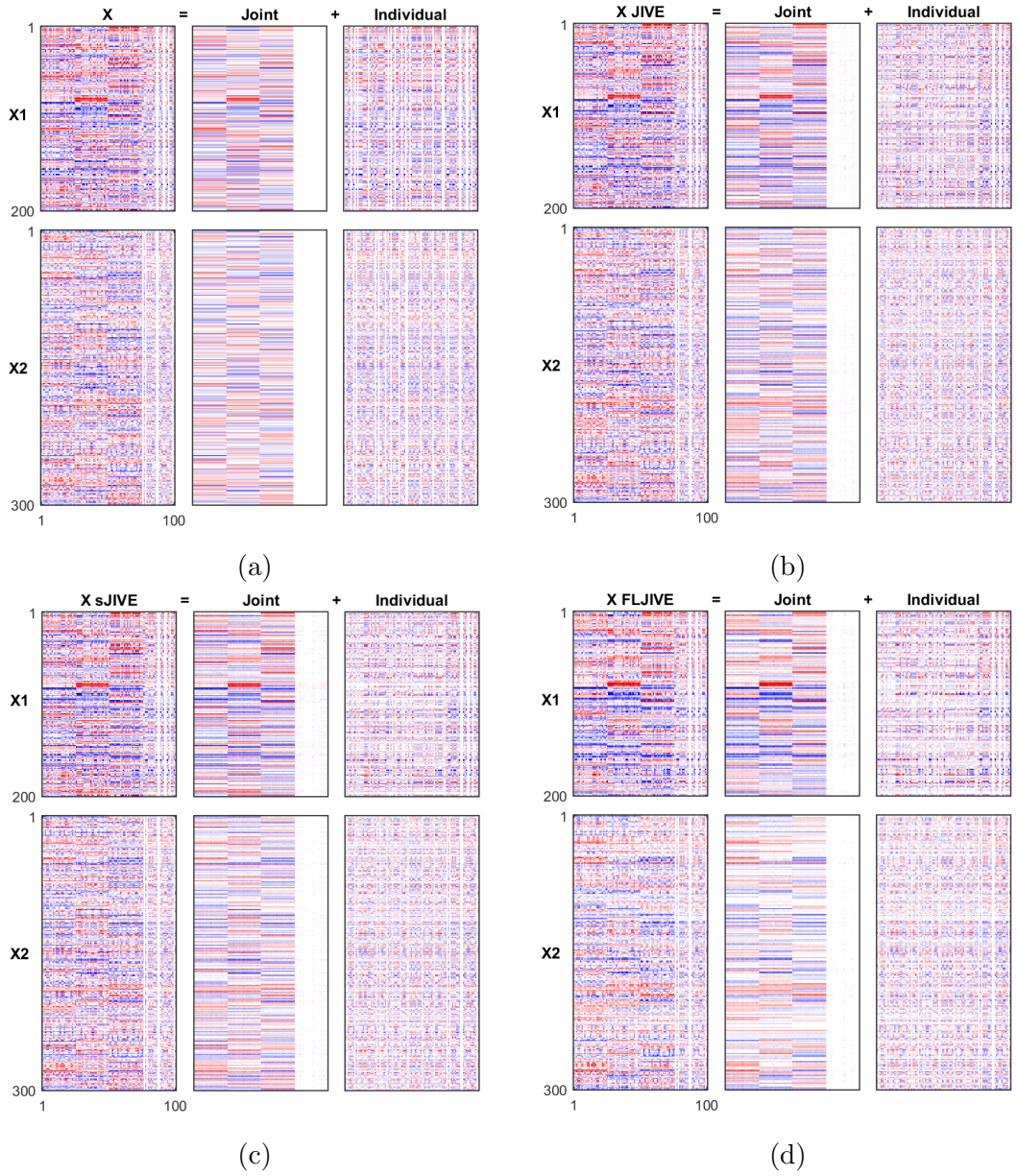


Figure 3.12: Showing the original data set from Figure 3.11 with the noise component being removed from the data (a), the fit from running JIVE on the simulated data (b), the fit from sJIVE (c) and the fit from FLJIVE (d). For each heatmap in the sub-figures, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.

Table 3.1: Showing the squared Frobenious norm of the differences between the simulated data set, with ranks $\boldsymbol{r}_1$, and the fits of the different components for JIVE, sJIVE and FLJIVE.

Component	JIVE	sJIVE	FLJIVE
$\boldsymbol{J}$	31.45	34.22	54.63
$\boldsymbol{A}_1$	18.78	22.03	26.02
$\boldsymbol{A}_2$	15.88	17.91	24.44
Average	22.03	24.72	35.03

Rank setup $\boldsymbol{r}_2 = (r = 9, r_1 = 11, r_2 = 7)$

The data set used for this estimation study is created with the parameters $n = 100, p_1 = 200, p_3 = 300, r = 9, r_1 = 11, r_2 = 7, \sigma = 0.05$. The data set including the noise component can be seen in Figure 3.13.

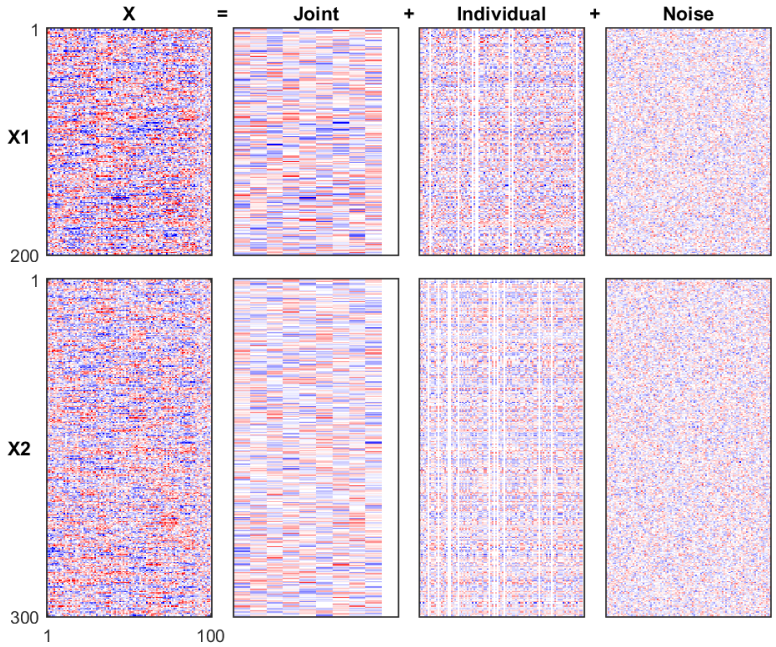


Figure 3.13: Showing the simulated data set used for the estimation study for the rank setup $\boldsymbol{r}_2 = (r = 9, r_1 = 11, r_2 = 7)$. For each heatmap in the figure, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.

Figure 3.14 shows JIVE (b), sJIVE (c), FLJIVE(d) applied to the data. In (a) one can see the data set without the noise component. As for the previous rank setup, the

difference between JIVE and sJIVE is marginal. The fit for FLJIVE is on the other hand visually different from JIVE and sJIVE. Looking at Table 3.2 this can be confirmed. In the table one can see that FLJIVE is roughly 70% percent worse in estimating all three components compared to JIVE while sJIVE is approximately 30% worse on average. Figures of the first three actual joint loadings and scores for the different methods can be seen in Appendix B.

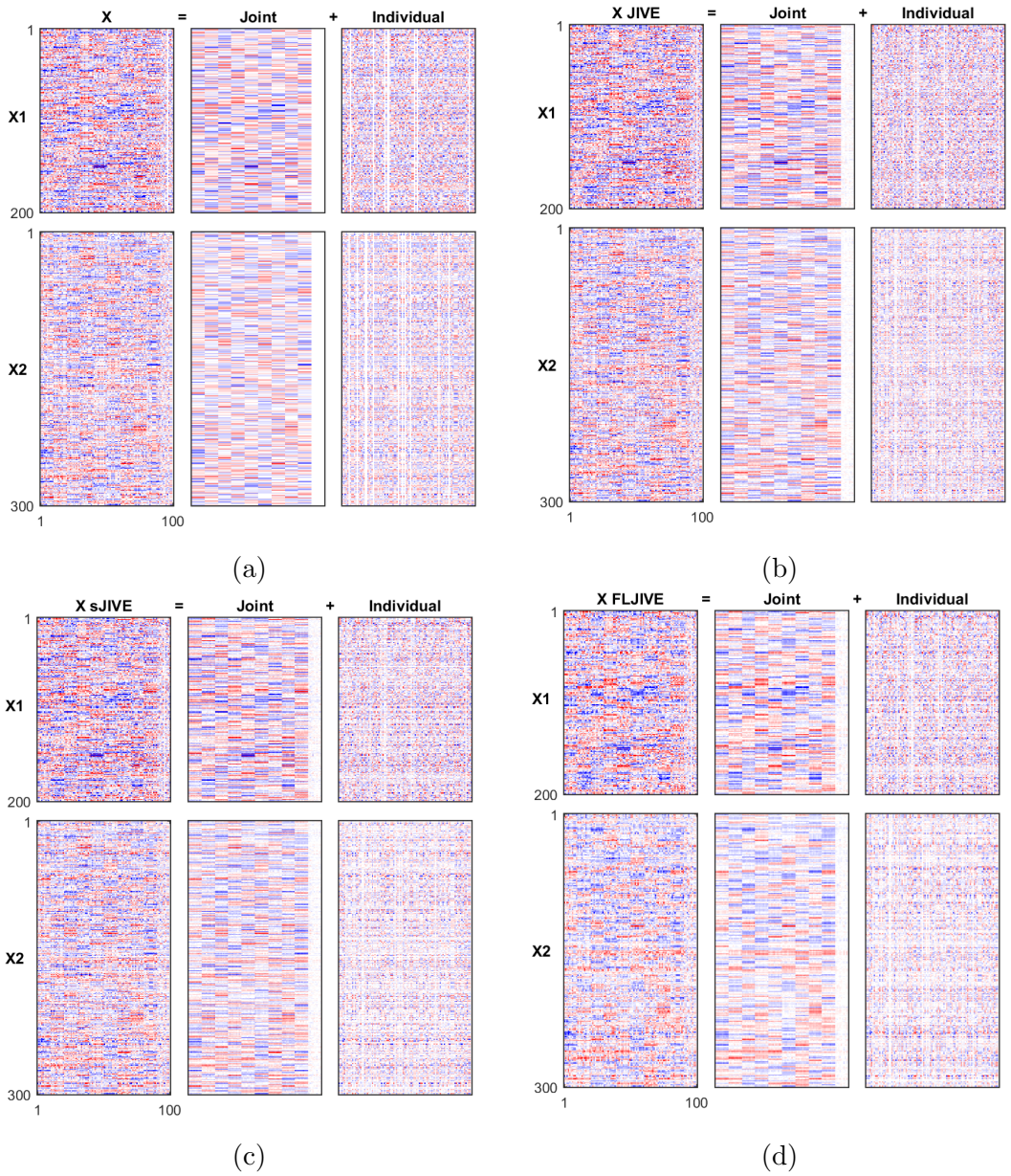


Figure 3.14: Showing the original data set from Figure 3.13 with the noise component being removed from the data (a), the fit from running JIVE on the simulated data (b), the fit from sJIVE (c) and the fit from FLJIVE (d). For each heatmap in the sub-figures, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.

Table 3.2: Showing the squared Frobenious norm of the differences between the simulated data set, with ranks $\boldsymbol{r}_2$, and the fits of the different components for JIVE, sJIVE and FLJIVE.

Component	JIVE	sJIVE	FLJIVE
$\boldsymbol{J}$	43.65	52.53	75.72
$\boldsymbol{A}_1$	22.40	32.65	35.38
$\boldsymbol{A}_2$	23.16	29.19	36.24
Average	29.74	38.13	49.11

3.3.2 Data with underlying fused PC loading

The data set in this subsection has fused joint loadings as described in section 3.1.2. For this data set FLJIVE is expected to perform significantly better than the other two methods. This subsection also explores hybrid versions of the methods where sPCA and FLPCA are used for finding the joint component while regular PCA is used for the individual component (equivalent to setting the penalization parameters $\lambda_1, \lambda_2 = 0$ for the individual components).

Rank setup $\boldsymbol{r} = (r = 3, r_1 = 5, r_2 = 7)$

The simulated data set, with parameters $n = 100, p_1 = 200, p_3 = 300, r = 3, r_1 = 4, r_2 = 5, \sigma = 0.15$, used in this subsection can be seen in Figure 3.15. As seen in the figure, the joint component exhibits both sparse and fused properties. Also, since the joint component is very sparse, the noise component have been increased.

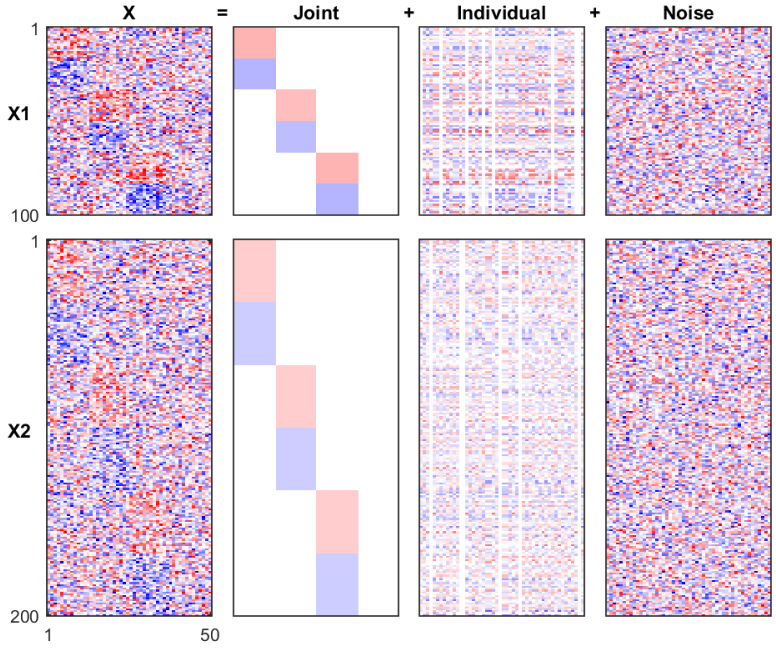


Figure 3.15: Showing the simulated data set used for the estimation study for the rank setup $\boldsymbol{r} = (r = 3, r_1 = 5, r_2 = 7)$. For each heatmap in the figure, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.

In Figure 3.16 one can see the result of fitting JIVE (b), sJIVE (c) and FLJIVE (d) with the correct ranks. Looking at (b) one can still see the fused properties in the joint component for JIVE. However, JIVE also captures a lot of noise in the joint component which is not desirable in real scenario when the true components are not available for comparison. Sparse JIVE (b) managed to reduce the amount of noise in the joint component, but at the same time the signal for the fused parts have been decreased. FLJIVE has on the other hand done a great job in capturing the fused parts without capturing the majority of the noise.

Table 3.3 supports the observations that can be made in Figure 3.16. The squared Frobenious norm between the fitted $\boldsymbol{J}$ and the true $\boldsymbol{J}$ for JIVE and sJIVE is as high 51.08 and 53.14, which is approximately 700% higher than FLJIVE's 7.45. The result of using FLPCA to estimate the joint component and regular PCA to estimate the individual components is around 15% better for the first individual component and 5% better for the second individual component than the other two methods.

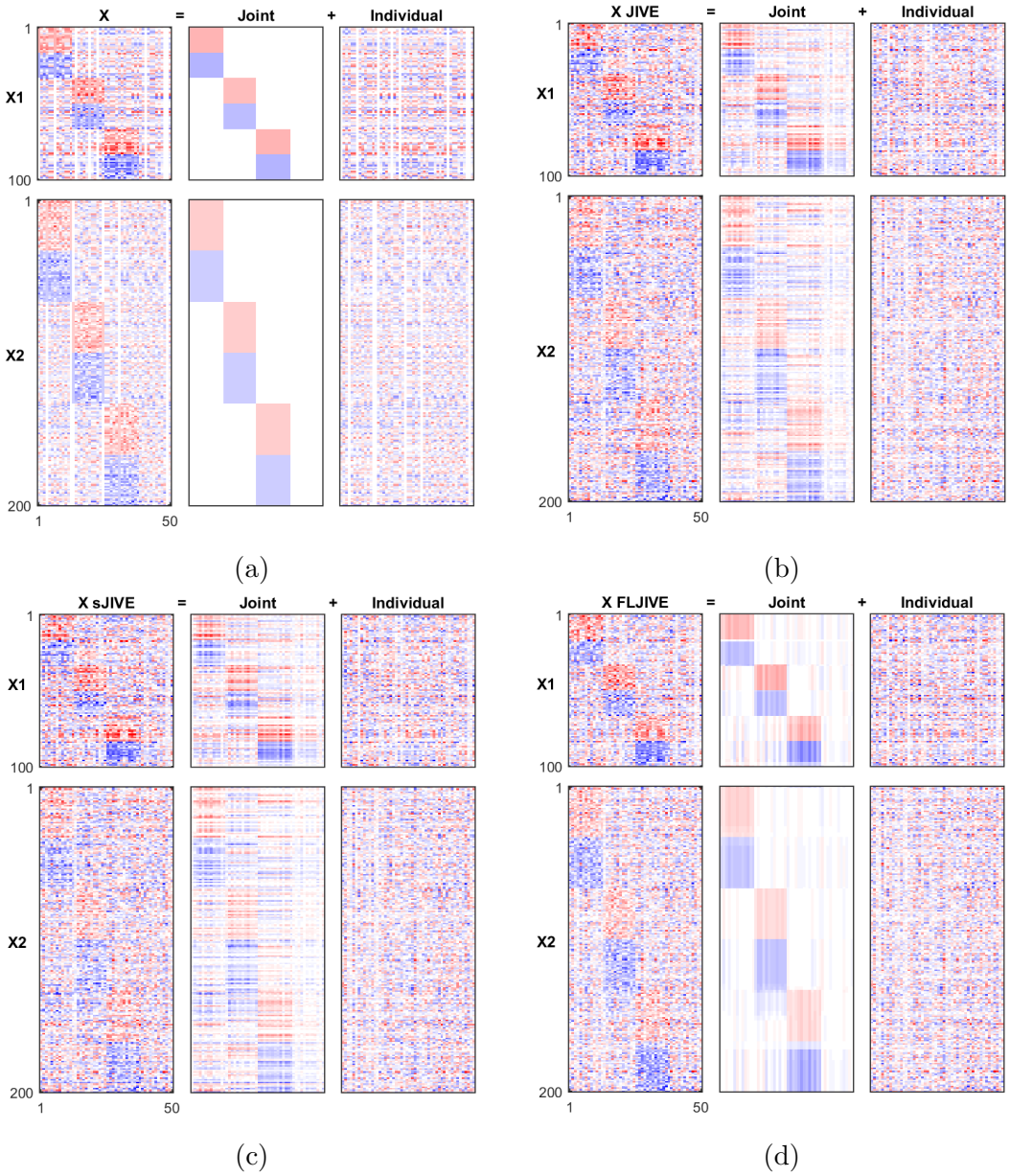
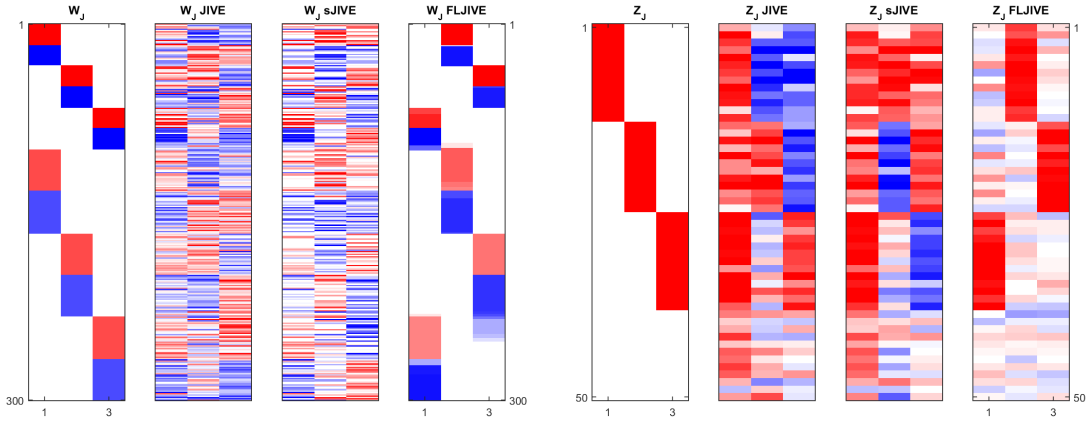


Figure 3.16: Showing the original data set from Figure 3.15 with the noise component being removed from the data (a), the fit from running JIVE on the simulated data (b), the fit from sJIVE (c) and the fit from FLJIVE (d). For each heatmap in the sub-figures, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.

Table 3.3: Showing the squared Frobenious norm of the differences between the simulated data set, with ranks $\boldsymbol{r}_3$, and the fits of the different components for JIVE, sJIVE and FLJIVE.

Component	JIVE	sJIVE	FLJIVE
$\boldsymbol{J}$	51.08	53.14	7.45
$\boldsymbol{A}_1$	35.68	35.38	30.75
$\boldsymbol{A}_2$	63.77	63.17	60.05
Average	50.17	50.56	32.75

In this case it is interesting to look at the underlying loading and scores of the true data and the three methods. Figure 3.17 shows the true loadings together with the estimated loadings (a) and the true scores together with the estimated scores (b) for the three different methods. Since the underlying singular values are all equal, the methods finds the loadings and scores in a random order. As seen in (a) JIVE and sJIVE does not succeed in capturing the true loadings. On the contrary, FLJIVE does surprisingly well when it comes to finding the true loadings. In (b) one can see that the scores for JIVE and sJIVE does not resemble the true underlying scores. The estimated scores for FLJIVE are almost identical to the real scores with the exception of some minor noise.



(a) (b)

Figure 3.17: Showing the underlying joint loadings together with the estimated joint loadings by JIVE, sJIVE and FLJIVE (a) and the underlying joint scores together with the estimated joint scores by JIVE, sJIVE and FLJIVE (b). In the figure the joint loadings (a) and the joint scores (b) are plotted on the x-axis and the corresponding entries of the vectors on the y-axis.

4

TCGA data

The Cancer Genome Atlas (TCGA) provides publicly available data for a variety of cancer types. The cancer type which is the focus of this thesis is Glioblastoma Multiforme (GBM) which is the most aggressive, and common, malignant brain tumor in humans. For each cancer type, TCGA provides data for a relatively large number of objects and multiple data types. The data types provided by TCGA for GBM include copy number aberrations (CNA), gene expression, DNA methylation and a few more. The focus here will lie on the CNA data type which can also be downloaded via the UC Santa Cruz Cancer Browser (Link: *https://genome-cancer.ucsc.edu/proj/site/hgHeatmap/*).

4.1 The data set

The TCGA GMB CNA (gistic2) data originally consists of 577 samples measured over 24174 different gene positions. In order to, in the future, be able to easily extend the analysis in this section, only the samples and genes that intersect with the samples and genes of the gene expression data type is used. The intersection between the CNA (gistic2) and gene expression (AffyU133a) data types consists of 508 samples measure over 11076 gene positions. However, in this analysis the gene expression data type will not be included. Instead, each chromosome of the CNA data will be interpreted as its own data type. In this way JIVE can be used to identify how much of the underlying mutational process is shared between the chromosomes.

4.1.1 CNA

Copy Number Aberrations (CNA or also Copy Number) can be explained as the number of abnormal copies of larger strings of DNA on a certain position on the genome of a cell. A position on the genome of a cell can carry duplications of DNA, referred to as an *amplification*, or parts of the DNA might have been deleted which is referred to as a *deletion*. Humans normally carry two copies of all autosomes (chromosome not related

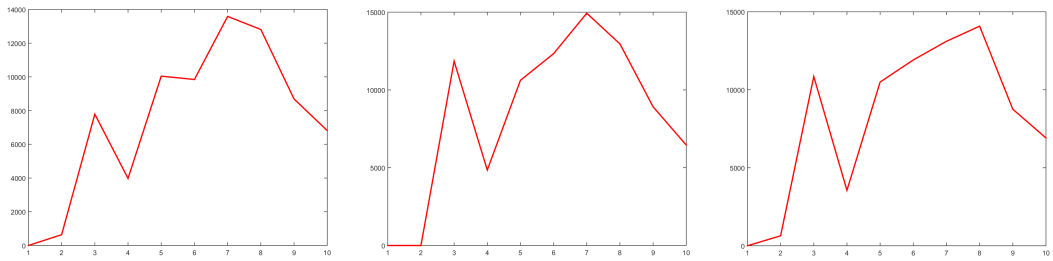
to the person's sex) which makes it favourable to measure CNA in a $\log_2$-scale. In this way deviations above 0 represents amplifications while deviations below zero represents deletions. Also, CNA data is assumed to have regions of equal copy number even in the pre-processing stage [14], and therefore, it is natural to adapt the Fused Lasso approach when working with this kind of data.

4.2 Rank selection

Here the rank selection procedure is applied to three different sets of chromosomes (interpreted as data types). In the first scenario chromosome 7, 9 and 10 will pose as three different data types. These three chromosomes are commonly known for having mutations strongly linked to this cancer type. The aim of this analysis is to discover if and how the underlying mutational process is linked between the three chromosomes. The second analysis will be conducted on chromosome 1, 7, 9 and 10, and the last set of chromosomes that will be analysed is 7, 9, 10 and 15. The aim of adding one chromosome to the original set (7, 9 and 10) is to discover if the added chromosome's underlying process is independent or not of the process behind chromosome 7, 9 and 10.

4.2.1 Chromosome 7,9,10

In this section the consensus rank selection algorithm will be applied to the data with chromosome 7, 9 and 10 being the data types. As described in the methods section for rank selection the CRS algorithm is best applied in a two-step procedure. Figure 4.1 shows the rank selection statistic as a function of the joint rank r with the individual ranks, r_1,r_2,r_3, set to 0. In (a)-(c) one can see that JIVE, sJIVE and FLJIVE all agree that $r = 4$ is the correct rank. The minimum is least clear for JIVE while sJIVE provides the second most clearest minimum and FLJIVE the clearest minimum.



(a) (b) (c)

Figure 4.1: Showing the consensus validation statistic (y-axis) as a function of the joint component rank, r, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c) applied to chromosome 7, 9, and 10 of the TCGA CNA data. The local minima at $r = 4$ is strongly suggested by all three methods.

In Figure 4.2 one can see the rank selection statistic for chromosome 7 ((a)-(c)), chromosome 9 ((d)-(f)) and chromosome 10 ((g)-(i)) given that the joint rank is equal to 4. All three methods agree that $r_1 = 2$ gives a stable clustering for chromosome 1 which can be seen in (a)-(c) of the figure. However, JIVE has a clear local minimum at $r_1 = 4$, sJIVE has a local minimum at $r_1 = 7$ and FLJIVE has a distinct local minimum at $r_1 = 6$. Given that all methods agree on $r_1 = 2$ being stable and that they do not share any other local minima, $r_1 = 2$ is probably the best choice.

In (d)-(f) of Figure 4.2 JIVE, and perhaps also sJIVE, show weak signs of a local minimum at $r_2 = 5$. In (f) one can see that FLJIVE does not support $r_2 = 5$ as being a local minimum. However, all three methods agree that the clustering is stable for $r_2 = 2$. Looking at the plotted statistics in (g) and (h), the almost linearly increasing curves supports the fact that an overfit has already occurred. This means that the individual rank for chromosome 10 is small and probably either 0 or 1. On the contrary, FLJIVE provides evidence for $r_3 = 7$ being an obvious local minimum.

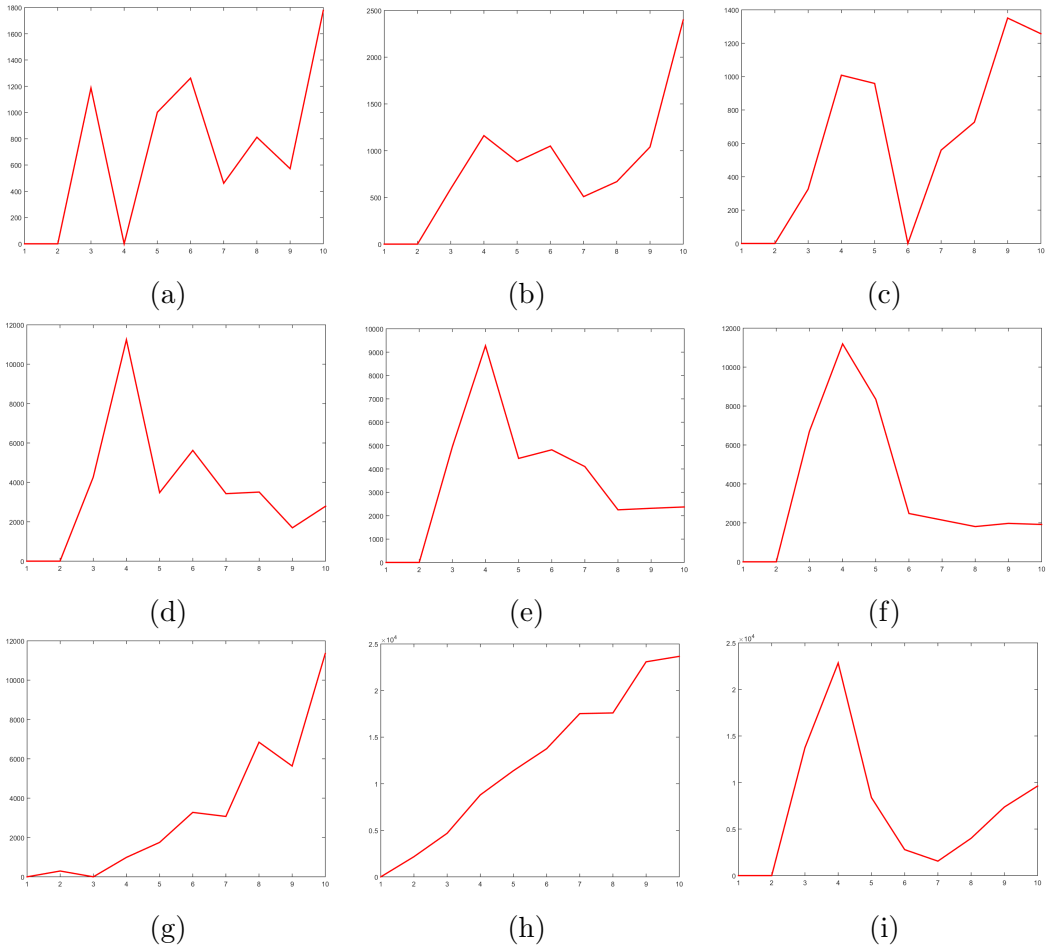
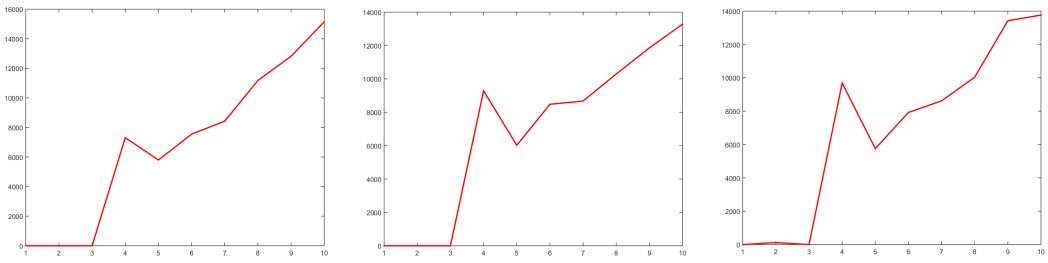


Figure 4.2: Showing the consensus validation statistic (y-axis) as a function of the individual components' ranks (x-axis) for JIVE (left column), sJIVE (middle column) and FLJIVE (right column) applied to chromosome 7 (first row), 9 (second row) and 10 (third row) of the TCGA CNA data. For chromosome 7, in (a)-(c), all three methods disagree on which rank is correct since JIVE, in (a), have a clear local minimum at $r_1 = 4$ and FLJIVE, in (c), have a clear minimum at $r_1 = 6$. However, all three methods agree that $r_1 = 2$ provides a stable clustering for chromosome 7. In (d)-(f) there is a lack of clear local minima larger than 2 which suggest that the correct individual rank for chromosome 9 is $r_2 = 2$. For chromosome 10 FLJIVE, in (i), show a clear local minimum at $r_3 = 7$. In (g) and (h) one can see that JIVE and sJIVE does not provide evidence for the correct rank being $r_3 = 7$ for chromosome 10.

4.2.2 Chromosome 1,7,9,10

The analysis in this subsection is extended to include chromosome 1 in addition to chromosome 7, 9 and 10. In Figure 4.3 one can see the result of the first step in the rank selection process. JIVE (a), sJIVE (b) and FLJIVE all consent to the joint clustering being stable for $r = 3$. In the figure one can also see that the three methods also show weak signs of a local minimum at $r = 5$.

Given that the joint rank is $r = 3$ Figure 4.4 show the rank selection statistic, for chromosome 1 (first row) to chromosome 10 (last row), as a function of the individual ranks r_1,...,r_4. The plots in (a)-(c) suggests that $r_1 = 3$ provides the most stable clustering. In (d)-(f) one can see that there is a lack of local minima for $r_2 > 2$ which suggests $r_2 = 2$ being the best choice. In (g), JIVE shows signs of a local minimum at $r_3 = 6$ while FLJIVE has a local minimum at $r_3 = 5$, and perhaps also $r_3 = 6$, for chromosome 9 which can be seen in (i). Sparse JIVE on the other hand, does in (h) not show any sign of local minima which suggests the correct rank being in the range 0 to 2. For chromosome 10, JIVE has a local minimum at $r_4 = 2$ which can be seen by looking at (j) in the figure. Looking at (k) and (l) one can see that sJIVE and FLJIVE agree on $r_4 = 3$ as being the best choice.



(a) (b) (c)

Figure 4.3: Showing the consensus validation statistic (y-axis) as a function of the joint component rank, r, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c) applied to chromosome 1, 7, 9, and 10 of the TCGA CNA data. All three methods suggest $r = 3$ gives a stable clustering, but all of them also show small local minima at $r = 5$.

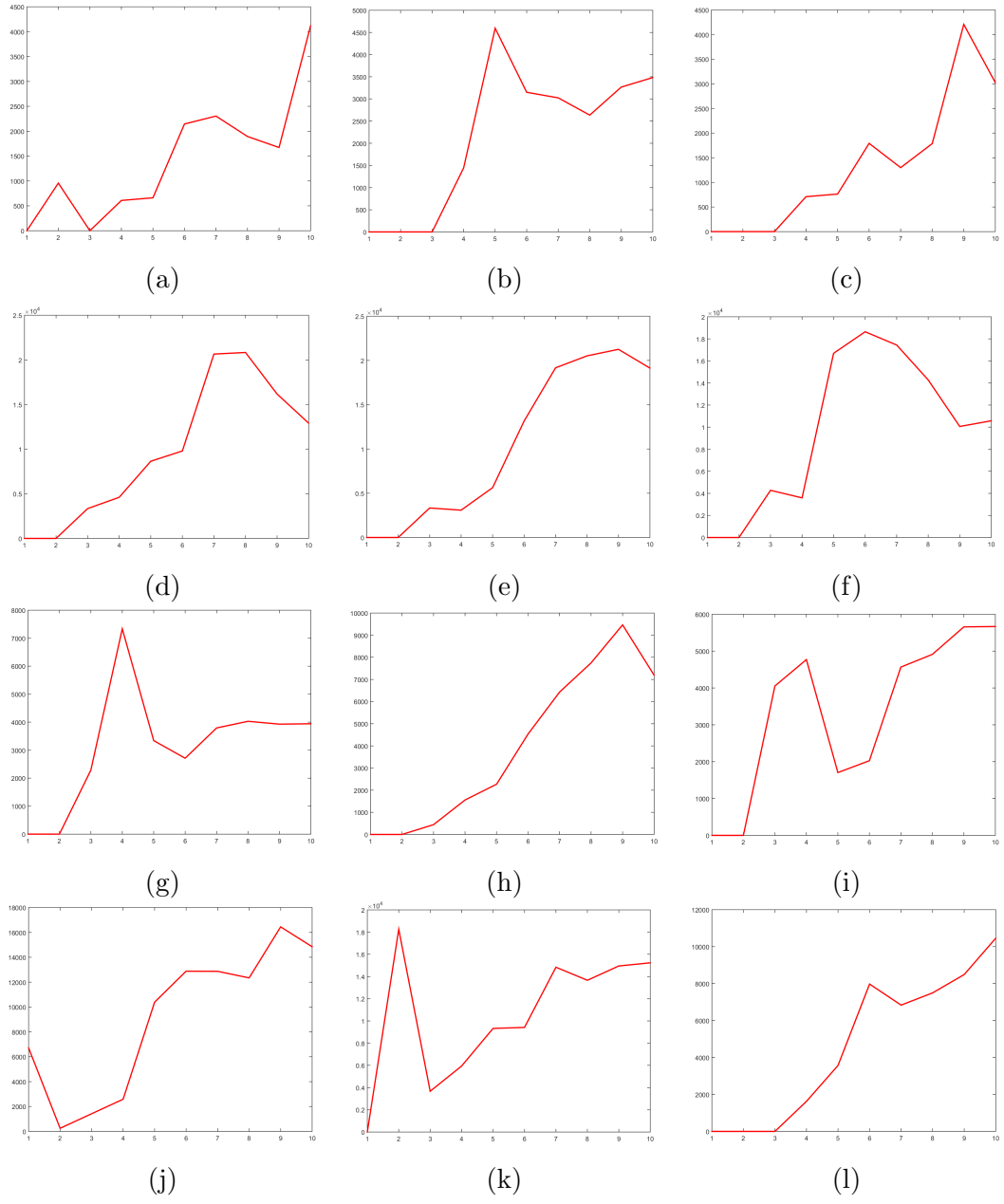


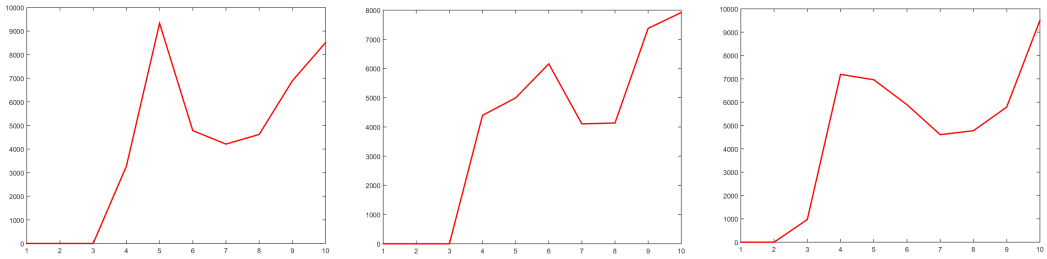
Figure 4.4: Showing the consensus validation statistic (y-axis) as a function of the individual components' ranks (x-axis) for JIVE (left column), sJIVE (middle column) and FLJIVE (right column) applied to chromosome 1 (first row), 7 (second row), 9 (third row) and 10 (fourth row) of the TCGA CNA data. For chromosome 1, in (a)-(c), all three methods seem agree that $r_1 = 3$ is the correct rank. The lack of clear local minima in (d)-(f) suggest that the correct individual rank for chromosome 7 is $r_2 = 1$ or $r_2 = 2$ for which the higher should be favoured. The statistic for JIVE and FLJIVE, in (g) and (i), suggest that the individual rank for chromosome 9 is either $r_3 = 2$, $r_3 = 5$ or $r_3 = 6$, while sJIVE, in (h), favoures $r_3 = 2$. Both sJIVE and FLJIVE, in (k) and (l), seem to agree that $r_4 = 3$ is the correct rank for chromosome 10 while JIVE, in (j), have its local minimum at $r_4 = 2$.

4.2.3 Chromosome 7,9,10,15

In this subsection the analysis includes chromosomes 7, 9, 10 and 15. In Figure 4.5 the rank selection statistic as a function of the joint rank r can be seen. JIVE (a) suggests that either $r = 3$ or $r = 7$ is the correct estimate of the joint rank. This is also supported by sJIVE in (b). FLJIVE also has a local minimum at $r = 7$ but indicates at the same time that the clustering is stable for $r = 2$. It is most probable that the correct joint rank is $r = 7$ since all three methods share a local minimum at that rank.

Figure 4.6 shows the results from the rank selection procedure for the individual components given that the joint rank is $r = 7$. In (a) and (b) it is clear that both JIVE and sJIVE have a local minimum at $r_1 = 5$ for chromosome 7. As seen in (c), FLJIVE does also have a local minimum at $r_1 = 5$, even though it is rather weak, and instead, the clearest local minimum for FLJIVE appears at $r_1 = 8$. For chromosome 9 JIVE (d) and sJIVE (e) show weak signs of local minima at $r_2 = 6$ while FLJIVE (f) have a slight local minimum at $r_2 = 5$. The lack of clear common local minima in all three methods indicates that the rank r_3 should either be set to 0 or 1. In (g)-(i) none of the methods demonstrates the existence of distinct local minima for chromosome 10. Instead, the continuously increasing statistic in all three methods indicates that $r_3 = 0$ or $r_3 = 1$ is the most favourable choice.

The statistics for chromosome 15 differs from the statistics shown so far. JIVE shows, in (j), that all ranks provides a stable clustering. For sJIVE (k) all ranks give s stable clustering except $r_4 = 8$, and all clusterings except $r_4 = 6,8,10$ is stable for FLJIVE as seen in (l). These unusual statistic plots in (j)-(l) appears partly because of the definition of the rank selection statistic and that the clusterings are in fact rather stable. As the rank selection statistic is defined in section 2.6.1 the statistic will be 0 if the median of the values greater than 0.5 is equal to 1. This will occur if 50%, or more, of the values greater than 0.5 are equal to 1. This is what happens for all ranks in (j) and most of the ranks in (k) and (l).



(a) (b) (c)

Figure 4.5: Showing the consensus validation statistic (y-axis) as a function of the joint component rank, r, (x-axis) for JIVE (a), sJIVE (b) and FLJIVE (c) applied to chromosome 7, 9, 10 and 15 of the TCGA CNA data. All three methods suggest $r = 7$ as being the correct rank.

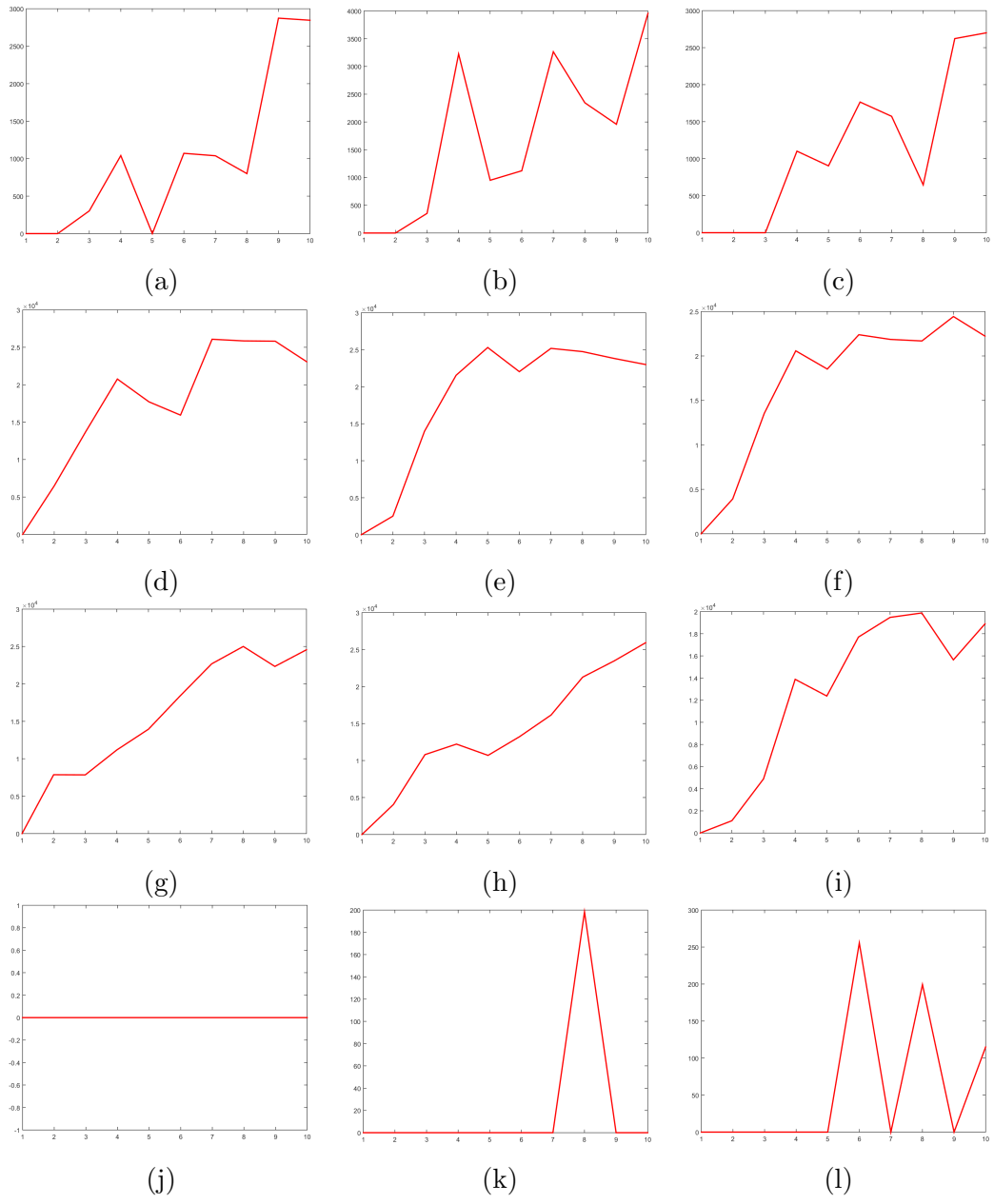


Figure 4.6: Showing the consensus validation statistic (y-axis) as a function of the individual components' ranks (x-axis) for JIVE (left column), sJIVE (middle column) and FLJIVE (right column) applied to chromosome 7 (first row), 9 (second row), 10 (third row) and 15 (fourth row) of the TCGA CNA data. For chromosome 7, in (a)-(c), all three methods seem to have a local minima at $r_1 = 5$ even though the clearest local minimum for FLJIVE is at $r_1 = 8$. The lack of clear local minima in (d)-(f) and (g)-(i) suggest that the correct individual rank for chromosome 9 and 10 is r_2,$r_3 = 0$ or r_2,$r_3 = 1$. The statistic in (j)-(l) suggest that the individual clustering is stable for chromosome 15 independent of the number of clusters.

4.3 Estimation

In this section the focus will lie on the actual fit of JIVE, sJIVE and FLJIVE for the three set of chromosomes. As the true underlying ranks are not known, and that the rank selection procedure in previous section suggested more than one possible rank setup for each set of chromosome, one cannot be entirely sure of which ranks to use for estimation. For each set of chromosomes one fit will be presented using one of the possible rank setups suggested from previous section. Note that fitting the methods with other ranks than presented here might give slightly different results.

4.3.1 Chromosome 7,9,10

The rank selection for chromosome 7, 9 and 10 strongly suggested that $r = 4$ was the most favourable joint rank. However, the three methods were not unanimous in their suggestions about the individual ranks. In Figure 4.7 (a) the CNA data for chromosomes 7, 9 and 10 is shown together with the fit of JIVE (b), sJIVE (c) and FLJIVE (d). In the figure the three different methods are fitted with the ranks $r = 4, r_1 = 2, r_2 = 2, r_3 = 1$.

The difference in the resulting fits can be hard to visually distinguish between the three methods. The differences are most obvious in the individual components. Comparing (b) and (c) in Figure 4.7 one can see that some of the weak signals in (b) have been removed by sJIVE in (c). In the individual component for chromosome 10, in (c), one can actually see that the lack of weak signals have given room for some of the originally strong signals to appear even stronger. In (d) one can see that FLJIVE have instead increased the strength in some of the weak signals that have fused properties.

Looking at the joint components in Figure 4.7 one can see that there are in fact 5 distinct joint subgroups (remember: $r + 1 = 5$ clusters). The first subgroup (green) have amplifications throughout entire chromosome 7, deletions on the beginning of chromosome 9 and that entire chromosome 10 carries deletions. The fifth subgroup (red) is similar to subgroup 1 except for chromosome 9 where instead the entire chromosome carries deletions. Other interesting subgroups are subgroup 4 (yellow) which have amplification on almost entire chromosome 9 and subgroup 2 (blue) which have less activity than the other subgroups. One can also see that the five subgroups differ only slightly between the three methods.

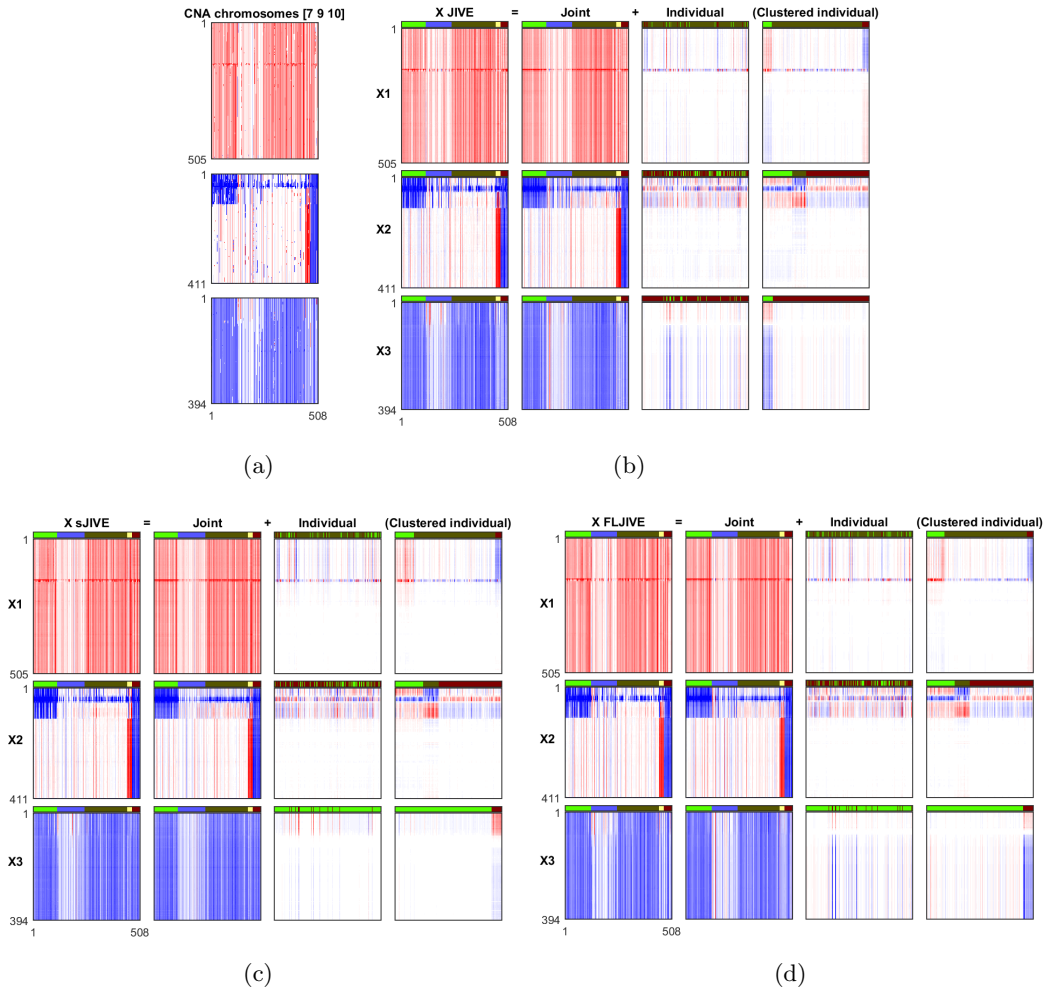


Figure 4.7: Showing chromosomes 7, 9 and 10 for the TCGA CNA data set (a) and the result of fitting JIVE (b), sJIVE (c) and FLJIVE (d) with ranks $r = 4, r_1 = 2, r_2 = 2, r_3 = 1$. The observations have been ordered according to their corresponding k-means clustering shown by the color coding on top of each heatmap. For each heatmap in the sub-figures, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.

Figure 4.8 shows the fitted joint loadings for the three different methods. In (a) one can see small differences in the first loading between the three different methods. One difference is that between $x \approx 600$ and $x \approx 900$ sJIVE and FLJIVE have successfully shrunken the loading to 0 while JIVE fluctuates around 0. Generally, in (a)-(d) one can see that sJIVE has decreased weak signals which in turn gives more room for the stronger signals. One can also see that the loadings for FLJIVE are much smoother and step-function-like than the loadings for sJIVE and JIVE.

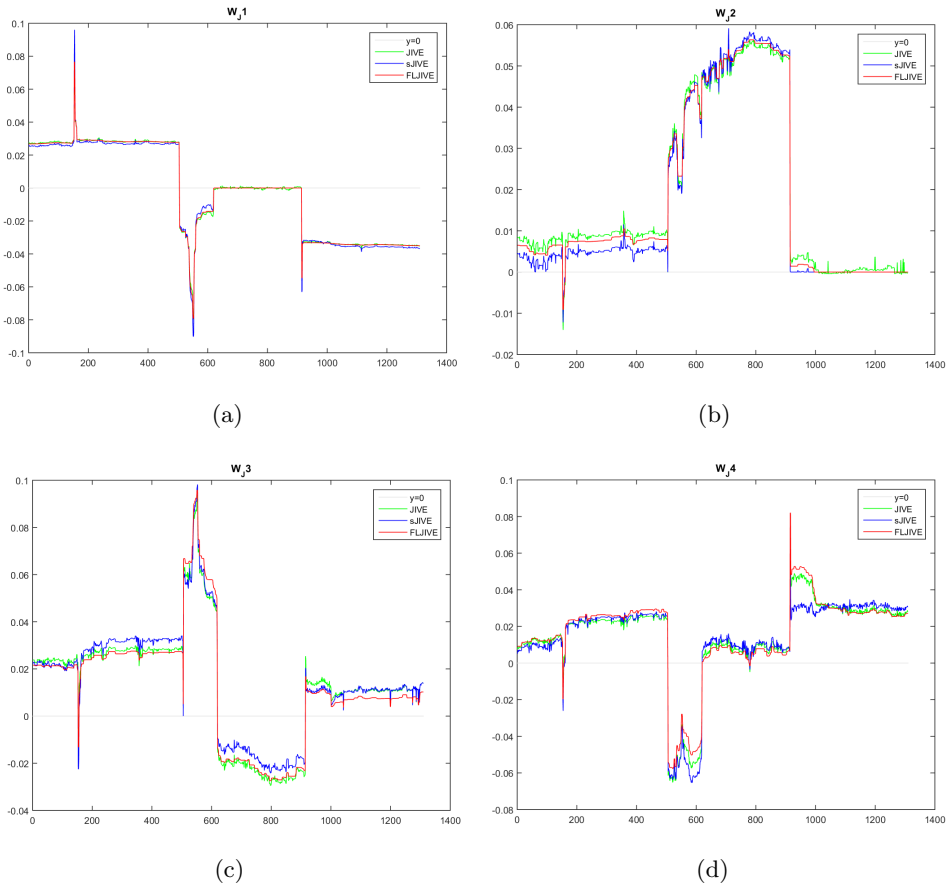


Figure 4.8: Showing the value of the 4 joint loadings (y-axis) as a function of genomic position (x-axis) in (a)-(d) for JIVE, sJIVE and FLJIVE.

4.3.2 Chromosome 1,7,9,10

For chromosomes 1, 7, 9 and 10 the rank selection suggested that $r = 3$ gave the most stable joint clustering and that another candidate rank was $r = 5$. As for the previous set of chromosomes, the results of the rank selection was not as clear when it came to the individual ranks. For the estimation of chromosomes 1, 7, 9 and 10 the ranks $r = 3, r_1 = 3, r_2 = 2, r_3 = 5, r_4 = 3$ was chosen.

Figure 4.9 shows the data for the chromosomes (a) and the corresponding fits for JIVE (b), sJIVE (c) and FLJIVE (d). The differences between the fits are again minor but can best be seen by looking at the individual components. In (b) one can see that JIVE captures alot of structure in the individual component for chromosome 9. Both sJIVE (c) and FLJIVE (d) suggest that this structure is in fact noise. If one looks carefully at the joint component for chromosome 9 one can see that also structure in this component have been reduced, even though the sum of the joint and individual component is

more or less the same. This is an indication that JIVE has captured noise in both the joint and individual components which, when added together, cancel each other out. In chromosome 1 both sJIVE and FLJIVE reduce some of the signals (comparing to JIVE) while FLJIVE also make the signals more smooth. The difference between sJIVE and FLJIVE is visually hard to see without zooming.

For this set of chromosomes the four joint clusters differs significantly between FLJIVE and the other two methods. Independent of which clustering one looks at, one can see that the clusters are not as distinct as when only chromosome 7, 9 and 10 was used. However, one can still identify jointly defined clusters between the chromosomes. There is also more activity and more clearly defined clusters in the individual components compared to the previous chromosome set.

The differences between the methods are more clear when looking at the underlying loadings. Figure 4.10 shows the loadings for the joint components of JIVE, sJIVE and FLJIVE. As seen in (b) sJIVE reduces, and sometimes removes, some of the weak signals present in JIVE. This is most easily seen for $x > 2100$ in (b) where the loading for sJIVE is significantly more sparse than the loading for JIVE. The difference between sJIVE and FLJIVE is much more clear when looking at the loadings. Looking at the three loadings one can see that FLJIVE is generally more sparse than JIVE but not as sparse as sJIVE. Another obvious difference is that FLJIVE has also fused subsequent values together which results in more smooth loadings.

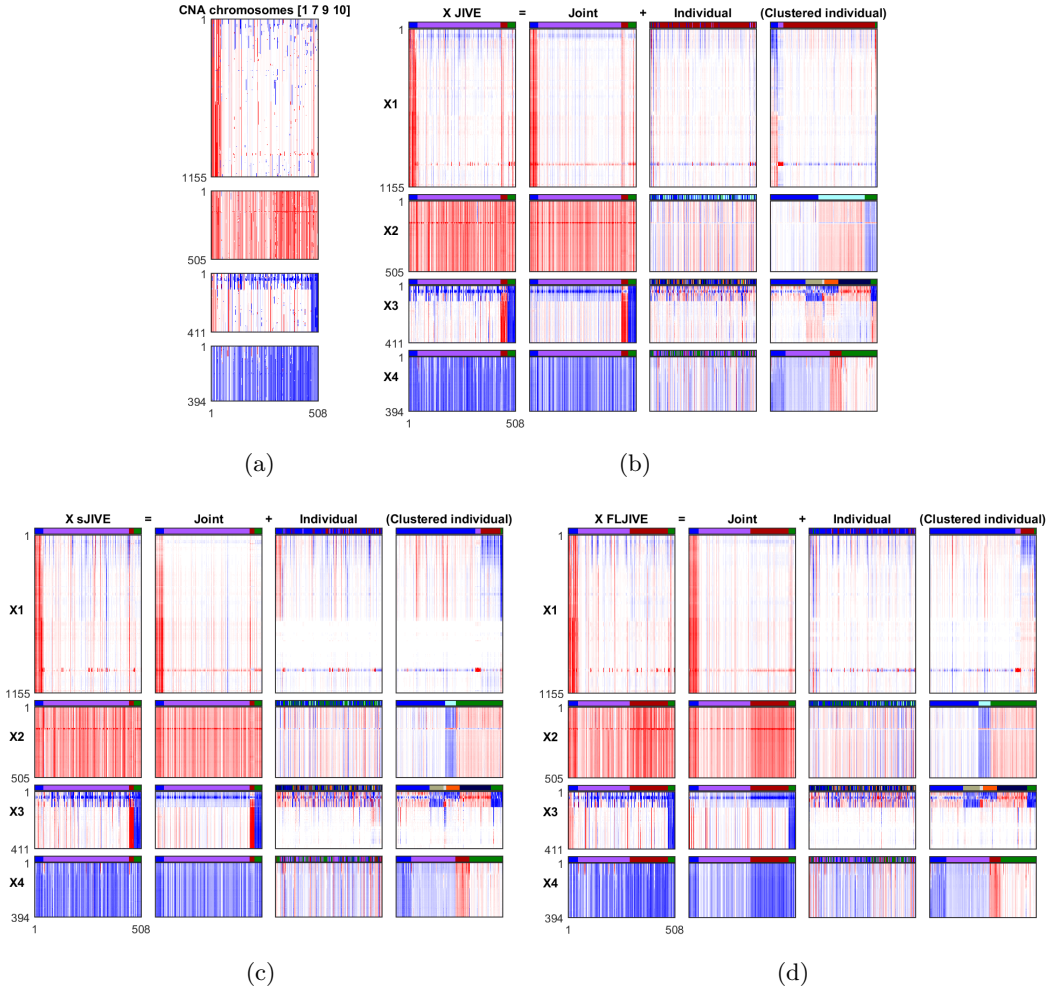
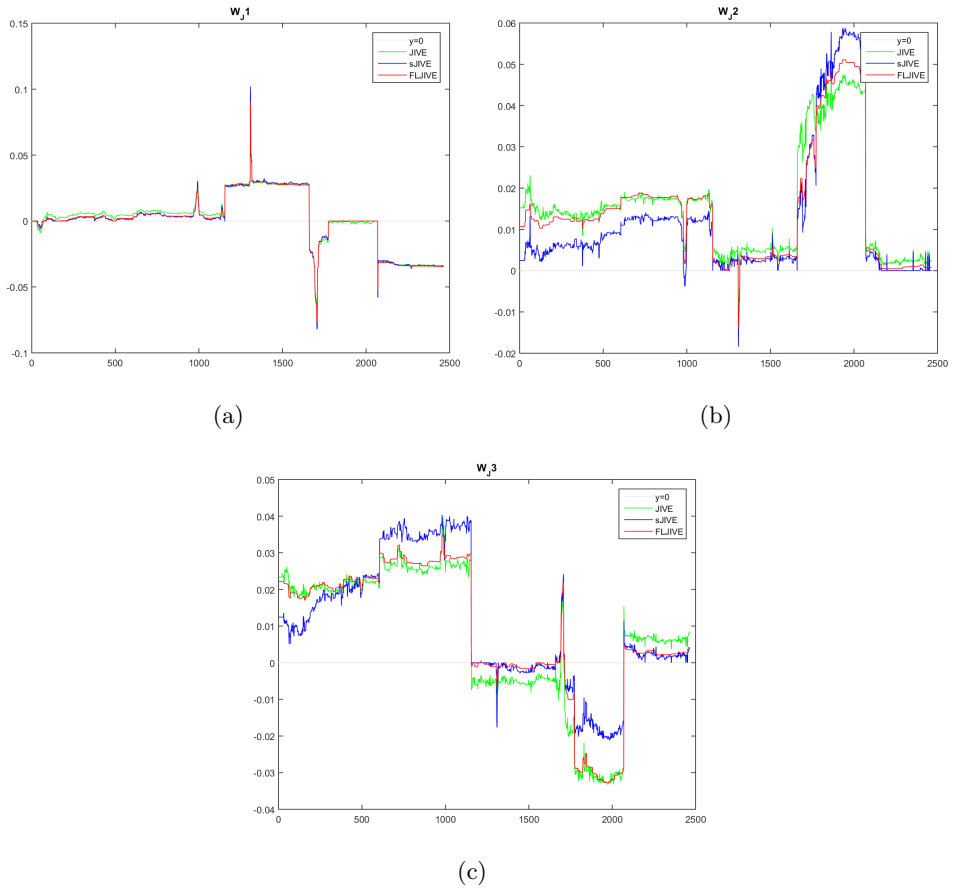


Figure 4.9: Showing chromosomes 1, 7, 9 and 10 for the TCGA CNA data set (a) and the result of fitting JIVE (b), sJIVE (c) and FLJIVE (d) with ranks $r = 3, r_1 = 3, r_2 = 2, r_3 = 5, r_4 = 3$. The observations have been ordered according to their corresponding k-means clustering shown by the color coding on top of each heatmap. For each heatmap in the sub-figures, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.



(a) (b)

(c)

Figure 4.10: Showing the value of the 3 joint loadings (y-axis) as a function of genomic position (x-axis) in (a)-(c) for JIVE, sJIVE and FLJIVE.

4.3.3 Chromosome 7,9,10,15

For the last set of chromosomes, 7, 9, 10 and 15, the joint rank selection for the three methods was not quite as consensual as for the other sets of chromosomes. All three methods did however provide evidence for the joint rank being $r = 7$. As for the individual ranks, the three methods seemed to agree that the most favourable ranks were $r_1 = 5, r_2 = 1, r_3 = 1$. On the other hand, the rank selection statistic for chromosome 15 behaved very unexpectedly. The ranks used for the estimation of chromosomes 7, 9, 10 and 15 was chosen to be $r = 7, r_1 = 5, r_2 = 1, r_3 = 1, r_4 = 1$.

In Figure 4.11 one can see the data for chromosomes 7, 9, 10 and 15 (a) and the three different fits for JIVE (b), sJIVE (c) and FLJIVE (d). As seen in (b)-(d) a lot less structure is captured in the individual components compared to the estimations of the previous sets of chromosomes. This is due to the joint rank being higher, and therefore capturing more of the data, but also due to the fact that the individual ranks are, on

average, lower. The joint rank being as high as 7 makes the differences between the fits of the three methods even smaller since differences in the first few loadings can be compensated for in the last loadings. The most significant difference between the methods is that FLJIVE has moved the signal in the top of chromosome 10 from the individual component to the joint component. By fusing, and thereby increasing, the signal in the top of the joint component, FLJIVE has to compensate by decreasing the signal in the top of the individual component. The differences between the methods are again most clear by looking at the joint loadings.

Looking at the actual clusterings of the joint component it is hard to identify joint clusters that are truly defined for all chromosomes. Most of the clusters make sense in only a few of the chromosomes at the same time. This, and the fact that as much as 8 clusters were needed for a stable clustering, suggests that chromosome 15 share little joint structure with chromosome 7, 9 and 10.

In Figure 4.12 one can see the fitted joint loadings (a)-(g) for the three methods. Generally, sJIVE and FLJIVE encourage more sparsity in the loadings. One can also see that FLJIVE finds loadings with both sparse and fused properties. Even though the loadings have differences between the methods, the resulting fits when adding the components together are very similar. This is mostly due to the fact that the joint component is estimated using as many as 7 components. Even though sJIVE and FLJIVE estimates the first components such that they are sparse, and fused in the case for FLJIVE, the subsequent components compensate for this change in structure, which in the end make the resulting fits almost the same for all three methods.

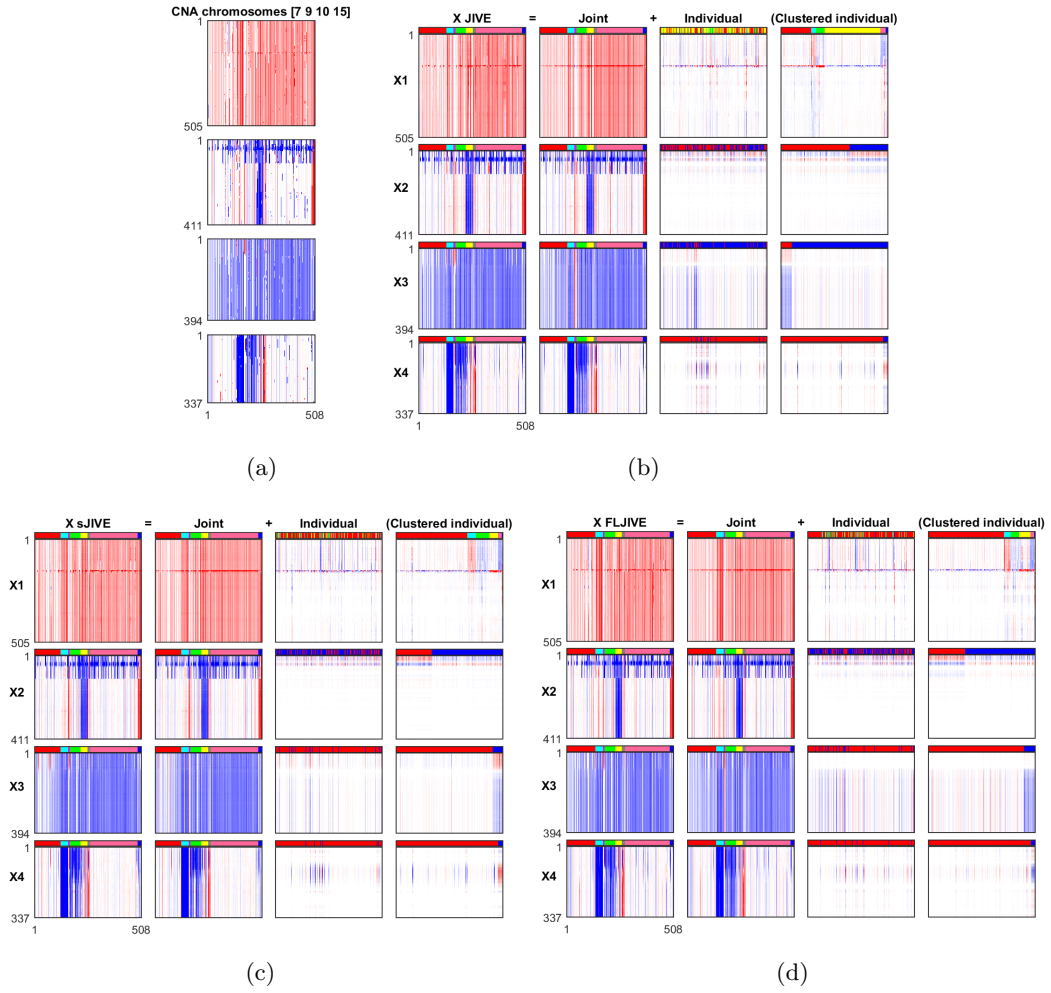
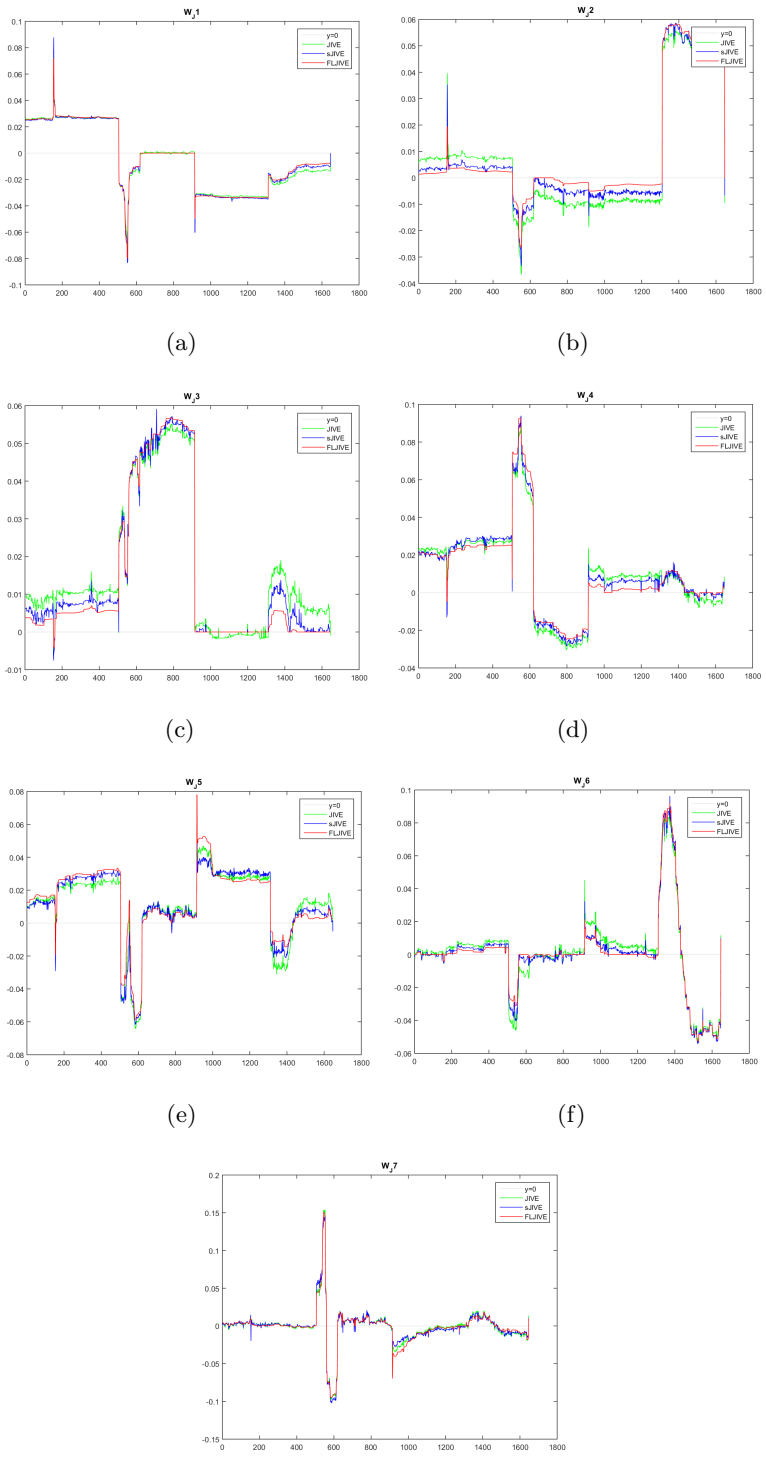


Figure 4.11: Showing chromosomes 7, 9, 10 and 15 for the TCGA CNA data set (a) and the result of fitting JIVE (b), sJIVE (c) and FLJIVE (d) with ranks $r = 7, r_1 = 5, r_2 = 1, r_3 = 1, r_4 = 1$. The observations have been ordered according to their corresponding k-means clustering shown by the color coding on top of each heatmap. For each heatmap in the sub-figures, objects are plotted along the x-axis and the corresponding features are plotted along the y-axis.



(a) (b) (c) (d) (e) (f) (g)

Figure 4.12: Showing the value of the 7 joint loadings (y-axis) as a function of genomic position (x-axis) in (a)-(g) for JIVE, sJIVE and FLJIVE.

5

Discussion

This discussion will mainly focus on three parts: Firstly, the discussion will cover the results of the simulation study and how the proposed consensus rank selection algorithm and FLJIVE, as well as JIVE and sJIVE, performed in a setting where the underlying structures were known. Secondly, the results of applying these methods to the TCGA data set will be discussed. Lastly, a discussion is held about possible future directions to the work presented in this thesis.

5.1 Simulation study

The results in the Simulation Study section provided evidence for that the proposed Consensus Rank Selection method, together with the proposed consensus statistic, is successful in finding the correct underlying ranks. The results also indicated that estimating the individual ranks is a harder problem than finding the joint rank. This conclusion can be made based on the demonstrated difference in degree of distinction between the local minima for the joint and individual components.

The rank selection algorithm was also more successful for the simulated data sets with lower ranks. This is not surprising since having a more complicated underlying structure should also make the rank selection problem harder. When the simulated data did not have underlying sparse or fused structure, the rank selection algorithm with sJIVE and FLJIVE did not perform significantly worse than rank selection with JIVE. However, when the data had underlying fused joint components JIVE and sJIVE failed to find the correct estimate for the joint rank. Only FLJIVE managed to correctly estimate the true joint rank which demonstrates the power of the method when the Fused Lasso assumption of the underlying components are being fulfilled.

The proposed consensus statistic worked rather well for the simulated data sets, although

it was sometimes hard to distinguish the correct rank without having pre-knowledge about the true underlying rank. Therefore, for future analyses a change to the statistic that would highlight local minima more significantly is desired. The statistic also demonstrated some other drawbacks, which were most obvious when being applied to the TCGA data. The most obvious drawback was that if more than 50% of the values greater than 0.5 was equal to 1, the statistic would be 0. This is more probable to happen for lower ranks since with a fewer number of clusters objects are more likely to be clustered together. That is also the reason to why the statistic was 0 for $r = 1$ and $r = 2$ in many of the rank selection plots. The problem could be reduced by switching the 0.5th quantile (the median) in the definition of the statistic for a much lower quantile. This would not solve the problem completely, but it would make it more unlikely to occur.

When taking a closer look at the actual estimates of the different methods FLJIVE did significantly worse than the other two methods when trying to estimate the simulated data sets without fused properties. This is not surprising since FLJIVE will try to fuse subsequent values in the loadings even though they are not close to each other. By chance some of the values in the underlying loadings will be rather close to 0, and in these cases sJIVE will shrink them to be exactly 0. This is the reason to why sJIVE performed slightly worse than JIVE in estimating the components of the data.

However, when the loadings of the data had clear fused properties FLJIVE was superior the other two methods. JIVE captured a significant amount of the noise in the components, and while sJIVE did reduce this level of noise it also reduced the actual underlying signal. FLJIVE was the only method that managed to find the true underlying loadings despite the high level of noise. From this one can learn that using sJIVE and FLJIVE for estimation should be done with caution if there are no evidence of sparse or fused properties in the data and/or there is reason to believe that, or there is prior knowledge of, the data having underlying sparse or fused structure in the loadings.

5.2 TCGA Data

The rank selection section for the TCGA data provided evidence for 5 jointly defined clusters between chromosome 7, 9 and 10. This is still not a large number of clusters needed in order to summarize the joint structure shared between the data types. The plots for the rank selection statistic showed that the local minimum at $r = 4$ was least clear for JIVE and most distinct for FLJIVE which was slightly more clear than the local minimum of sJIVE. This observation, together with the results from the estimation study of the fused data set, suggests, without visually looking at the data, that the TCGA GMB CNA data set has underlying fused structures. This is a confirmation that the assumption of underlying fused loadings, that was made on the both the model in FLJIVE and the data itself, was sound. From the figure of the fitted joint loadings one could also draw the conclusion that the underlying loadings of the data showed characteristics of both sparsity and fusion.

The results of the rank selection for the individual components were not as clear as for the joint component. The estimation study suggested that the individual ranks are harder to estimate, and this results strengthens this hypothesis. Some of the plots did however show clear local minima, but since the methods did not agree on a local minimum it is hard to determine if there is a commonly correct individual rank for the three methods. Possibly, since the methods provide different fits, depending on the penalization parameters, the most favourable individual ranks might also be dependent on the method itself and it's parameters which affect the fit. This is a possible explanation to why two or three methods show clear local individual minima, but for different ranks.

In the estimation section of chromosome 7, 9 and 10 the results from fitting JIVE, sJIVE and FLJIVE was presented. The actual fits differed only moderately between the three methods, and instead, the most interesting observations could be made by looking at the actual clusters. In the figure for the three different fits one could clearly see that there existed 5 distinct clusters and that they were in fact jointly represented. This suggests that there is reason to believe that chromosome 7, 9 and 10 share some underlying mutational process. Although there were not as much activity in the individual components, there were still clusters defined which can be interpreted as deviations within the jointly defined structures. An example of this can be seen by looking back at Figure 4.7 which shows the estimation results of the three methods, together with the original data. In the figure the first cluster (green) for chromosome 9 (middle row) represents a subgroup which have deletions on approximately the first third/fourth of the chromosome. The individual component corresponding to these objects can then be interpreted as variations in both strength of the signal, but also start- and end-points of the deletion.

Rank selection statistic plots for joint and individual ranks, throughout the rank selection of the TCGA data set, highlighted one of the previously mentioned disadvantages with the current consensus statistic. If majority of the values greater than 0.5, in the consensus matrix, is equal to 1, the statistic will be exactly 0. One example where this was highlighted was for the joint rank selection of chromosomes 1, 7, 9 and 10 for where there were signs of a local minimum at $r = 5$. However, all three methods also agreed on $r = 1$,$r = 2$ and $r = 3$ providing stable clusterings since the statistic was 0 for those values of r. If the statistic was defined to work with a much lower quantile than the median, which is the 0.5 quantile, the statistic plot for $r = 1$ to $r = 3$ might have looked different. There is a chance that there was a local minimum in the range $r = 1$ to $r = 3$ which was not visible due to the current definition of the consensus statistic. Another possible scenario is that the statistic for $r = 1$ to $r = 3$ should in fact be separated from 0 and continuously increasing which would make $r = 5$ the only true local minimum. Future work for the consensus statistic will provide more insight into this problem.

The actual fits of the methods to chromosome 1, 7, 9 and 10 was different from the fits to the original set of chromosomes. Most of the joint clusters made sense in only two or

three of the chromosomes at the same time. Although the rank was set to $r = 3$, which was only one less than for chromosome 7, 9 and 10, significantly more structure was put into the individual components. This is a strong indication that the individual components are compensating for structure that could not by captured jointly. The suggestion of these results are that adding chromosome 1 to the original set of chromosomes disturbs the joint structure of chromosome 7, 9 and 10. This would mean that chromosome 1 most likely shares some, but not all, parts of the underlying mutational process with chromosome 7, 9 and 10.

The joint rank selection for chromosome 7, 9, 10 and 15 was not as unanimous across the methods as for the other sets of chromosomes. Another obvious difference is that the rank selection suggested a joint rank as large as $r = 7$. This means that it would take three more clusters to summarize the joint component when adding chromosome 15 to the analysis. This alone is an indicator that chromosome 15 does not have structures that are shared jointly with all, or most of, the other chromosomes at the same time.

The estimation results for JIVE, sJIVE and FLJIVE supports the hypothesis that the mutational process for chromosome 15 is weakly linked to the process shared by the other chromosomes. Many of the clusters seen in chromosome 15 are not distinct clusters in the other data types. This most likely means that JIVE used the three extra clusters just for chromosome 15. However, there are a few clusters shared between chromosome 15 and one, or at most two, other data types at the same time. This could mean that there are weak links between chromosome 15 and some of the other chromosomes. It could also have happened by chance. Another support for chromosome 15 being independent of the other chromosomes would be if the individual components contained lots of the information, which is not the case. Still, since the join rank is large, the activity in the individual components is expected to be small. In summary, the majority of evidence points toward chromosome 15 sharing very limited portions of the mutational process with chromosome 7, 9 and 10.

5.3 Future work

For the future it would be natural to conduct a more in depth analysis of the actual subgroups found in this thesis and their relation to survival data. The goal of that analysis would be to discover how the different groups correlate with how long the patients survived after being diagnosed. It would also be of great importance if one can link therapies that are more successful to each of the subgroups. Extending the analysis to cell lines, where there are opportunities to test novel drugs, is also a possible, and very important, future direction. This future direction would benefit from extending the current JIVE model to also incorporate multiple data sets, and not just multiple data types. In this way one can gain insight into the relation between the cell lines and data from late stage cancer patients. A strong link between cell lines and late stage patients would mean that one could more easily test and specify efficient group or possibly patient

specific treatments on cell lines.

As briefly mentioned in section 4 the next step would be to also include gene expression as a data type into the analysis. This was also the motivation to why the intersection between the objects and gene positions of the CNA and gene expression data types was used for the analysis in this thesis. However, the assumption of fused loadings does not apply to gene expression data as it does to CNA data. Instead, gene expression data can be divided into groups referred to as *pathways*. For the gene expression data type a penalization method referred to as Group Lasso [15] could be applied instead of the Fused Lasso. For that to be possible JIVE must allow different penalization methods for each data type. However, allowing different penalization methods for each data type is a very straight forward implementation.

The consensus rank selection algorithm was proven to be a successful tool for finding the joint and individual ranks of the JIVE decomposition. However, the proposed consensus statistic had some obvious drawbacks. Future work includes improving the current statistic and also to compare it to already existing statistics such as the one proposed by S. Minto et al. [11]. The consensus rank selection algorithm itself is not a tool for selecting the penalization parameters λ_1 and λ_2. Instead, other model selection methodologies should be included into the analysis in order to more efficiently select λ_1 and λ_2 since one can argue that maybe the parameters should have been set even higher throughout section 4. However, the model selection problem for λ_1 and λ_2 is future work, and it is still an open problem in the literature.

The last possible future direction, that will be discussed in this thesis, is related to the analysis of the results in section 4. In the current state the JIVE model assumes that all data types contribute equally to the joint component. However, this assumption may be too strict since in reality it might be the case that not all data types share an equal amount of joint structure with each other. Therefore, it might be more reasonable for the data types to contribute unequally to the joint component. This kind of extension of JIVE would allow for an easier analysis of which data types that share joint structure with each other. It would also serve as an automatic model selection tool for the data types. Since if one data type does not share joint structures with the other data types, it would not be allowed to contribute to the joint component. After fitting this extended JIVE model one should be able to identify the proportion to which the data types contribute to the joint component, and one can in this way make easier decisions about which data types to include in further analyses.

Bibliography

[1] E. Lock et al. Joint and individual variation explained (JIVE) for integrated analysis of multiple datatypes. *Ann. Appl. Stat.*, vol. 7, no. 1, pp. 523-542, 2013.

[2] K. Hellton, M. Thoresen. Integrative clustering of high-dimensional data with joint and individual clusters, with an application to the Metabric study. arXiv:1410.8679, 2014.

[3] R. Shen, A. Olshen, M. Ladanyi, Integrative clustering of multiple genomic data types using a joint latent variable model with application to breast and lung cancer subtype analysis. *Bioinformatics*, vol. 25, pp. 2906–2912, 2009.

[4] Q. Mo, et al. Pattern discovery and cancer gene identification in integrated cancer genomic data. *Proc. Natl Acad. Sci. USA*, vol. 110, pp. 4245–4250, 2013.

[5] M. Andrecut. Parallel GPU Implementation of Iterative PCA Algorithms. *Journal of Computational Biology*, vol 16 Part 11, pp. 1593-1599, 2009.

[6] C. Ding, X. He (2004). K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, pp. 29. ACM.

[7] Zha, H. et al. Spectral relaxation for k-means clustering. In *NIPS*, vol 1, pp. 1057–1064, 2001.

[8] H. Shen, J.Z. Huang. Sparse principal component analysis via regularized low rank matrix approximation. *Journal of Multivariate Analysis*, vol 99 Part 1, pp. 1015-1034, 2008.

[9] R. Tibshirani et al. Sparsity and smoothness via the fused lasso. *J. R. Statist. Soc. B*, vol 67 Part 1, pp. 91-108, 2005.

[10] G.B. Ye, X. Xie. Split Bregman method for large scale fused Lasso. *Computational Statistics & Data Analysis*, vol 55 issue 4, pp. 1552–1569, 2011.

[11] S. Monti et al. Consensus Clustering: A Resampling-Based Method for Class Discovery and Visualization of Gene Expression Microarray Data. *Machine Learning* vol 52, pp. 91–118, 2003.

[12] R.T. Rockafellar. Convex Analysis. In: Princeton Landmarks in Mathematics, Princeton University Press, Princeton, NJ, ISBN: 0-691-01586-4, Reprint of the 1970 original, Princeton Paperbacks, 1997.

[13] B. Wahlberg et al. An ADMM algorithm for a class of total variation regularized estimation problems. *Proceedings of the 16th IFAC Symposium on System Identification* pp. 1–6, 2012.

[14] A. Olshen et al. Circular binary segmentation for the analysis of array-based DNA copy number data. *Biostatistics*, vol 5 issue 4, pp. 557–572, 2004.

[15] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *ournal of the Royal Statistical Society: Series B (Statistical Methodology)* vol 68 issue 1, pp. 49-67, 2006.

A

Mathematical derivations

Lemma A.1 *Given that* $\boldsymbol{z}^T\boldsymbol{z} = 1$*:*

$$\min_{\boldsymbol{w}} \frac{1}{2}\|\boldsymbol{X} - \boldsymbol{z}\boldsymbol{w}^T\|_F^2 + \lambda_1\|\boldsymbol{w}\| \Leftrightarrow \min_{\boldsymbol{w}} \frac{1}{2}(\mathrm{Tr}(\boldsymbol{X}^T\boldsymbol{X}) - 2\boldsymbol{w}\boldsymbol{X}^T\boldsymbol{z} + \boldsymbol{z}^T\boldsymbol{z}\boldsymbol{w}^T\boldsymbol{w}) + \lambda_1\|\boldsymbol{w}\|$$
$$\Leftrightarrow \min_{\boldsymbol{w}} -\boldsymbol{w}\boldsymbol{X}^T\boldsymbol{z} + \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \lambda_1\|\boldsymbol{w}\|.$$

By setting the derivative, with respect to $\boldsymbol{w}$*, to 0 and assuming that* $\boldsymbol{w} > 0$ *then one gets:*

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{w}}\left(-\boldsymbol{w}\boldsymbol{X}^T\boldsymbol{z} + \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \lambda_1\boldsymbol{w}\right) = 0 \\
&\Leftrightarrow -\boldsymbol{X}^T\boldsymbol{z} + \boldsymbol{w} + \lambda_1 = 0 \\
&\Leftrightarrow \boldsymbol{w} = \max\left(\boldsymbol{X}^T\boldsymbol{z} - \lambda_1, 0\right).
\end{aligned}$$

If one instead assume that $\boldsymbol{w} < 0$ *one gets that:*

$$\boldsymbol{w} = \min\left(\boldsymbol{X}^T\boldsymbol{z} + \lambda_1, 0\right).$$

Combining these two cases one gets that the minimizing $\boldsymbol{w}$ *is:*

$$\boldsymbol{w} = T_{\lambda_1}^{soft}\left(\boldsymbol{X}^T\boldsymbol{z}\right)$$

Lemma A.2

$$\min_{\boldsymbol{x}} \lambda\|\boldsymbol{x}\| + \boldsymbol{y}^T(\boldsymbol{z} - \boldsymbol{x}) + \frac{\mu}{2}\|\boldsymbol{z} - \boldsymbol{x}\|_2^2 \Leftrightarrow$$
$$\min_{\boldsymbol{x}} \lambda\|\boldsymbol{x}\| + \boldsymbol{y}^T\boldsymbol{z} - \boldsymbol{y}^T\boldsymbol{x} + \frac{\mu}{2}(\boldsymbol{z}^T\boldsymbol{z} - 2\boldsymbol{z}^T\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{x})$$

By setting the derivative, with respect to $\boldsymbol{x}$*, to 0 and assuming that* $\boldsymbol{x} > 0$ *then one gets:*

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{x}} \Big(\lambda \boldsymbol{x} + \boldsymbol{y}^T \boldsymbol{z} - \boldsymbol{y}^T \boldsymbol{x} + \frac{\mu}{2} (\boldsymbol{z}^T \boldsymbol{z} - 2\boldsymbol{z}^T \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{x}) \Big) = 0 \\
&\Leftrightarrow \lambda - \boldsymbol{y} + \mu(\boldsymbol{x} - \boldsymbol{z}) = 0 \\
&\Leftrightarrow \mu \boldsymbol{x} = \mu \boldsymbol{z} + \boldsymbol{y} - \lambda \\
&\Leftrightarrow \boldsymbol{x} = \max \left(\boldsymbol{z} + \frac{\boldsymbol{y}}{\mu} - \frac{\lambda}{\mu}, 0 \right)
\end{aligned}$$

Instead, by assuming that $\boldsymbol{x} < 0$ *then:*

$$\boldsymbol{x} = \min \left(\boldsymbol{z} + \frac{\boldsymbol{y}}{\mu} + \frac{\lambda}{\mu}, 0 \right)$$

Combining the two cases one gets that the minimizing $\boldsymbol{x}$ *is:*

$$\boldsymbol{x} = T^{soft}_{\lambda/\mu} \left(\boldsymbol{z} + \frac{\boldsymbol{y}}{\mu} \right)$$

Lemma A.3 *With* $\boldsymbol{z}^T \boldsymbol{z} = 1$*, the minimizing* $\boldsymbol{w}$ *for*

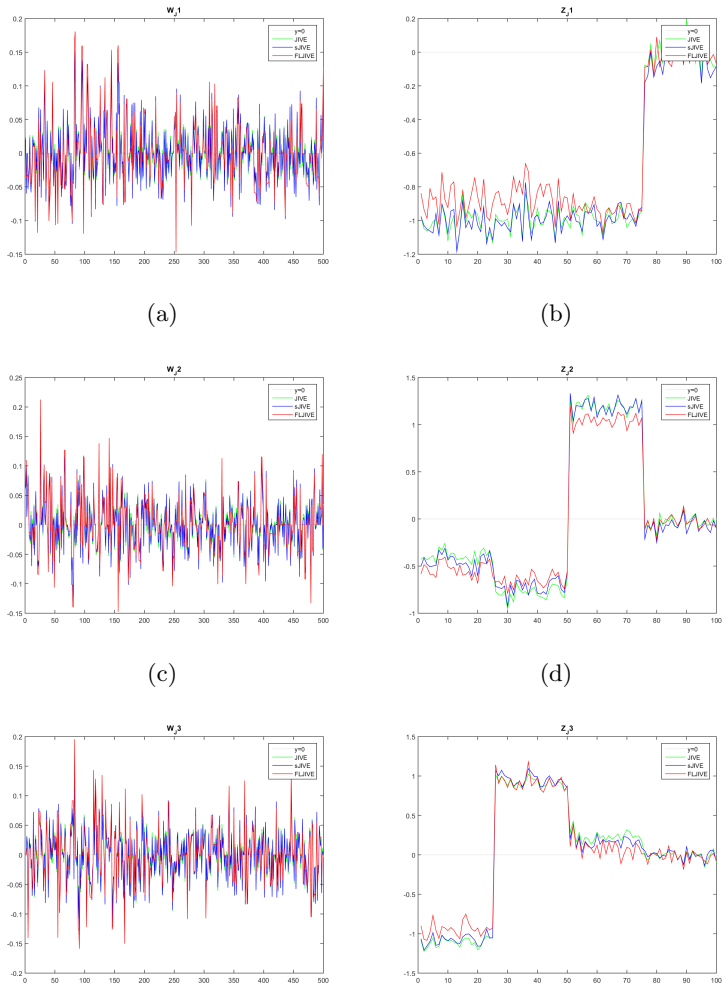
$$\min_{\boldsymbol{w}} \frac{1}{2} \|\boldsymbol{X} - \boldsymbol{z}\boldsymbol{w}^T\|_F^2 + \boldsymbol{u}^{(t)T}(\boldsymbol{w} - \mathbf{a}^{(t)}) + \mathbf{v}^{(t)T}(\mathbf{L}\boldsymbol{w} - \mathbf{b}^{(t)}) + \frac{\mu_1}{2} \|\boldsymbol{w} - \mathbf{a}^{(t)}\|_2^2 + \frac{\mu_2}{2} \|\mathbf{L}\boldsymbol{w} - \mathbf{b}^{(t)}\|_2^2$$

is found by setting the derivative, with respect to $\boldsymbol{w}$*, to 0:*

$$\begin{aligned}
&\frac{\partial}{\partial \boldsymbol{w}} \Big(\frac{1}{2} \|\boldsymbol{X} - \boldsymbol{z}\boldsymbol{w}^T\|_F^2 + \mathbf{u}^{(t)T}(\boldsymbol{w} - \mathbf{a}^{(t)}) + \mathbf{v}^{(t)T}(\mathbf{L}\boldsymbol{w} - \mathbf{b}^{(t)}) + \\
&\frac{\mu_1}{2} \|\boldsymbol{w} - \mathbf{a}^{(t)}\|_2^2 + \frac{\mu_2}{2} \|\mathbf{L}\boldsymbol{w} - \mathbf{b}^{(t)}\|_2^2 \Big) = 0 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Leftrightarrow \\
&\frac{\partial}{\partial \boldsymbol{w}} \Big(\frac{1}{2} (\mathrm{Tr}(\boldsymbol{X}^T \boldsymbol{X}) - 2\boldsymbol{w}^T \boldsymbol{X}^T \boldsymbol{z} + \boldsymbol{w}^T \boldsymbol{w}) + \mathbf{u}^{(t)T}(\boldsymbol{w} - \mathbf{a}^{(t)}) + \mathbf{v}^{(t)T}(\mathbf{L}\boldsymbol{w} - \mathbf{b}^{(t)}) + \\
&\frac{\mu_1}{2} (\boldsymbol{w}^T \boldsymbol{w} - 2\boldsymbol{w}^T \mathbf{a}^{(t)} + \mathbf{a}^{(t)T} \mathbf{a}^{(t)}) + \frac{\mu_2}{2} (\boldsymbol{w}^T \boldsymbol{L}^T \mathbf{L}\boldsymbol{w} - 2\boldsymbol{w}^T \boldsymbol{L}^T \mathbf{b}^{(t)} + \mathbf{b}^{(t)T} \mathbf{b}^{(t)}) \Big) = 0 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Leftrightarrow \\
&- \boldsymbol{X}^T \boldsymbol{z} + \boldsymbol{w} + \mathbf{u}^{(t)} + \mathbf{L}^T \mathbf{v}^{(t)} + \mu_1 (\boldsymbol{w} - \mathbf{a}^{(t)}) + \mu_2 (\boldsymbol{L}^T \mathbf{L}\boldsymbol{w} - \boldsymbol{L}^T \mathbf{b}^{(t)}) = 0 \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Leftrightarrow \\
&\boldsymbol{w} + \mu_1 \boldsymbol{w} + \mu_2 \boldsymbol{L}^T \mathbf{L}\boldsymbol{w} = \boldsymbol{X}^T \boldsymbol{z} - \mathbf{u}^{(t)} - \mathbf{L}^T \mathbf{v}^{(t)} + \mu_1 \mathbf{a}^{(t)} + \mu_2 \boldsymbol{L}^T \mathbf{b}^{(t)} \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \Leftrightarrow \\
&((1 + \mu_1)\mathbf{I} + \mu_2 \mathbf{L}^T \mathbf{L})\boldsymbol{w} = \mathbf{X}^{\mathbf{T}} \boldsymbol{z} + (\mu_1 \mathbf{a}^{(t)} - \mathbf{u}^{(t)}) + \mathbf{L}^T (\mu_2 \mathbf{b}^{(t)} - \mathbf{v}^{(t)}).
\end{aligned}$$

B

Supplementary figures

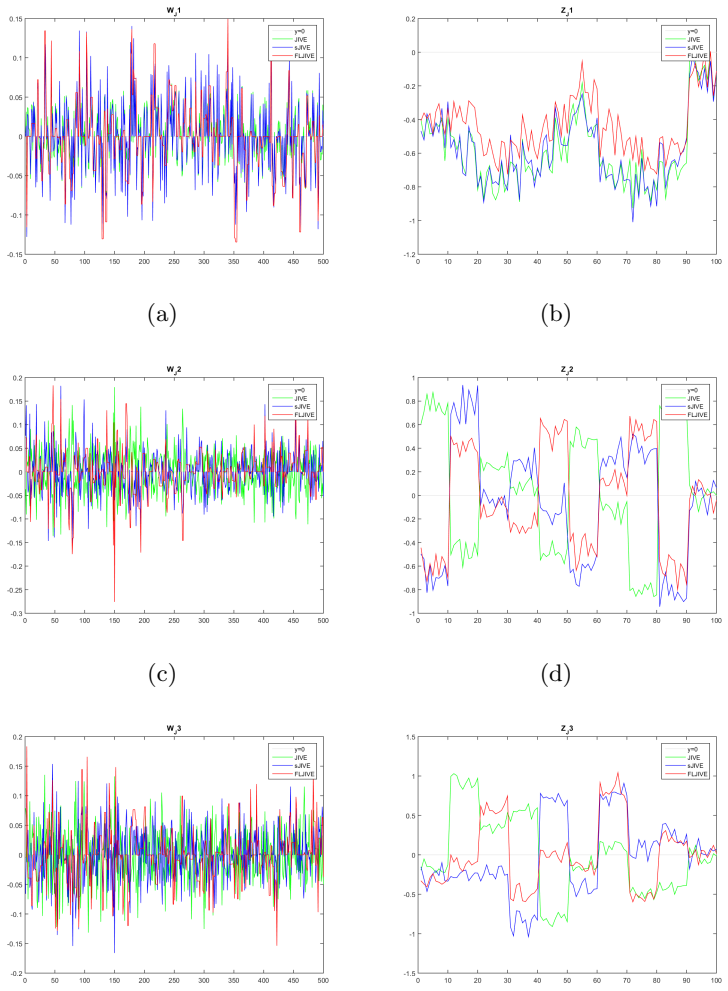


(a) (b)

(c) (d)

(e) (f)

Figure B.1: Showing the joint loadings (a),(c),(e) and scores (b),(d),(f) for JIVE (green), sJIVE (blue) and FLJIVE (red) for the estimation study in section 3.2.2 for the rank setup $\boldsymbol{r}_1 = (r = 3, r_1 = 4, r_2 = 5)$.



(a) (b)

(c) (d)

(e) (f)

Figure B.2: Showing the first three joint loadings (a),(c),(e) and scores (b),(d),(f) for JIVE (green), sJIVE (blue) and FLJIVE (red) for the estimation study in section 3.2.2 for the rank setup $\boldsymbol{r}_2 = (r = 9, r_1 = 11, r_2 = 7)$.