Internship project Write-up

By: Georgy Gomon
Studentnumber: s1559370
Date: May 29, 2021

1 Introduction

In high-dimensional regression, where the number of variables p is greater than the sample size n, one often wants to use penalized regression techniques which select only the best variables for prediction. A well-known example of such a technique is Lasso, which leads to sparse results (some model coefficients are set to exactly zero). A disadvantage of Lasso is that in the presence of groups of correlated variables Lasso is not a stable method. Out of each group of strongly correlated variables it chooses a variable at random. Many of the biological high-dimensional datasets, especially those dealing with omics data (genomics, proteomics, metabolomics, etc), have such large groups of highly correlated features.
One of the solutions proposed for correlated groups is the Group Lasso [Yuan and Lin, 2006], which sees a group as a single entity and penalizes groups depending on the group size. A disadvantage with Group Lasso occurs in case of overlap between groups, i.e.: when groups are not disjoint and some variables are present in more than one group. An overlapping variable is selected by Group Lasso if and only if all groups containing that variable are selected. Note that overlapping variables often occur in omics data, where for example an important transcription factor plays a role in multiple cellular pathways for apoptosis and is thus contained in multiple groups.
To handle this kind of overlapping groups the Group Lasso with Overlap (OGL) has been proposed [Jacob et al., 2009]. In this method each group is split into latent subgroups and the problem is reformulated in such a way that a variable is chosen in the model if any of the groups containing that variable are chosen in the model.
Several other methods to deal with overlapping groups in the high-dimensional setting are presented here. We consider the Sparse Overlap Group Lasso as proposed by [Park et al., 2015]. Moreover, the overlap LARS is presented, a method developed during this internship based on the group Least Angle Regression (also known as LARS) algorithm, [Alfons et al., 2016].
The outline of the report is as follows. In the next section the data set used for simulations is introduced. Afterwards the different methods compared within the context of this Internship are discussed. Then a comparison of these methods on the data is shown. Finally, we end with a general discussion regarding the different methods and which technique is best suited for which data-type. All R-code used to perform the simulations can be found at the open github repository `https://github.com/georgygomon/Internship_open` .

2 Data

All simulations presented in this report are conducted on synthetic datasets, since no real-life omics dataset was available during the internship. All synthetic datasets are of the form shown in Figure 1a. In this data-structure there are unique variables belonging to just one group in addition to variables of varying overlap, belonging to 2,3 or 4 groups. An example of such a dataset can be seen in Figure 1a. Here group 2 exists out of the variables 5,6,7 & 8 (unique to this group), the variables 17,18,19 & 20 (shared by 1 other group), the variables 25,26 & 27 (shared by 2 other groups) and the variables 29 & 30 (shared by 4 groups). Note that this data structure can be extended to include many more groups and that not all types of overlap should be present. For example, the representation of the synthetic dataset used by [Jacob et al., 2009] is shown in Figure 1b. In the examples given in Figure 1 the groups are indicated by a dashed square. In further data representations this will not longer be done, and instead the variables acting as support will be indicated by a dashed square. The groups will consist of the unique variables together with the overlapping variables in adjacent structures (squares, triangles or rounds), unless stated otherwise (such as in Figure 2).
Subsets and singletons can be introduced into the model, which are groups of variables that are completely enclosed within another group. An example of a data-structure with subsets and singletons is shown in Figure 2. We see that all elements of group 1 are also present as a subgroup or singleton, with the variables 1,2,3 & 4 forming a subset while the variables 17,20,21 & 22 are present as singletons.
A function in R was created to automatically generate datasets of this general structure. The function can be found on the github repository and works in the following way:

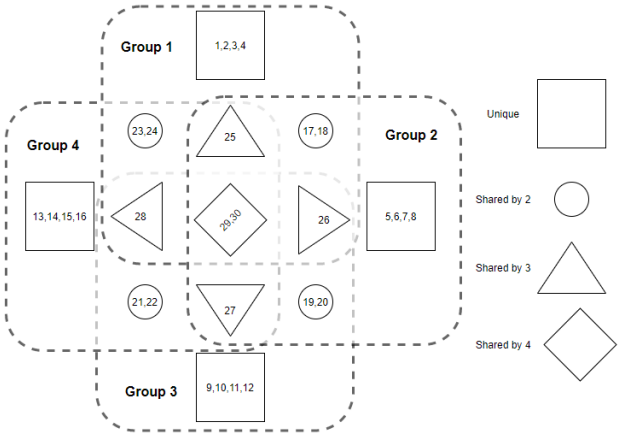
- Variables belonging to just 1 group are created by sampling from the normal distribution $\mathcal{N}(0, 1)$ with

a certain correlation ρ between these variables.

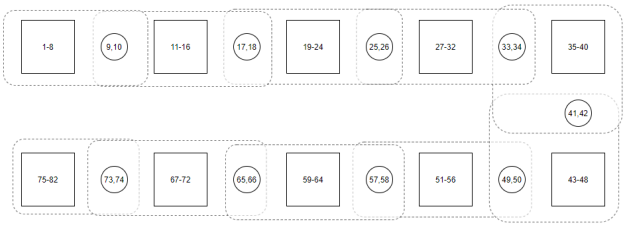
- The overlapping variables are created by taking the average of the variables of the groups they belong to. Then random noise is added equal to $\epsilon_{var} \cdot$ average.
- Lastly, the outcome y is created. For this a support needs to be chosen and the true regression coefficients vector $\vec{\beta}$ needs to be set. Now y is set to $y = \mathcal{N}(\vec{\beta} \cdot support, \epsilon_y \cdot sd(\vec{\beta} \cdot support))$

To make these rules clearer we shall consider an example. Let us have 2 groups $g_1 = (x_1, x_2)$ and $g_2 = (x_2, x_3)$. First x_1 and x_3 are randomly sampled from the $\mathcal{N}(0, 1)$ distribution. Variable x_2 is created as: $x_2 = (x_1 + x_3)/2 + \mathcal{N}(0, \epsilon_{var} \cdot sd[(x_1 + x_3)/2])$. If we take as support group 1 and as true coefficients vector $\beta = (2, 2, 0)$ then y is created as $y = 2x_1 + 2x_2 + \mathcal{N}(0, \epsilon_y \cdot sd(2x_1 + 2x_2))$.
Thus, in the creation of a dataset three free parameters are involved:

- ρ: The correlation between variables in the same group.
- ϵ_{var}, the degree of error for the overlapping variables.
- ϵ_y, the degree of error for the outcome.



(a) General data structure.

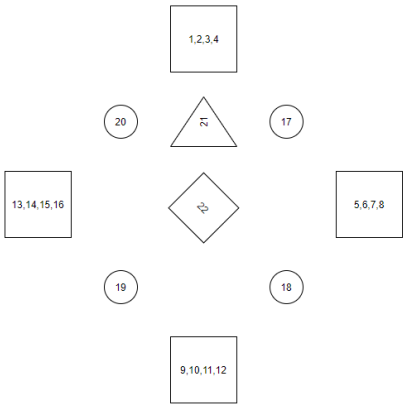


(b) Synthetic data-structure used by [Jacob et al., 2009].

Figure 1: Examples of synthetic datasets.

Group	Variables
1	1:4, 20:22, 17
2	5:8, 17:18, 21:22
3	9:12, 18, 19, 22
4	13:16, 19:22
5	1:4
6	17
7	20
8	21
9	22

(a) The groups with corresponding variables



(b) Graphical representation of the dataset.

Figure 2: Example of a dataset with singletons and a subset.

3 Methods

3.1 Lasso & Group Lasso

Before turning to methods specifically designed for overlapping groups we shall first review Lasso and Group Lasso and introduce some notation. Suppose our data consists of n samples of p features in the data matrix $\mathbf{X}$ and the n response vector $\mathbf{y}$. To account for the intercept a column of 1's is added to the data matrix $\mathbf{X}$.

Furthermore, we have the regression problem:

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}, \quad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\sigma})$$

with $\boldsymbol{\beta}$ the regression parameter. The ordinary least squares method has as loss function

$$L = ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2$$

yielding the ordinary least squares solution $\boldsymbol{\beta} = \left(\mathbf{X}^T\mathbf{X}\right)^{-1}\mathbf{X}^T\mathbf{y}$.
The Lasso minimizes the loss function

$$L = \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda||\boldsymbol{\beta}||_1.$$

Here $\lambda||\boldsymbol{\beta}||_1$ is the penalty, with λ the penalty parameter determining the strength of the penalty and $||\boldsymbol{\beta}||_1$ the l_1-norm, with the general l_k norm defined as: $||\boldsymbol{\beta}||_k = \left(\sum_{i=0}^{p} |\beta_i|^k\right)^{1/k}$. Note that if $\lambda = 0$ Lasso yields the ordinary least squares solution. Because of the geometrical constraints imposed by the Lasso penalty the solution obtained is sparse in the regression coefficients $\boldsymbol{\beta}$, thus effectively dropping variables out of the model. The group Lasso, as proposed by proposed by [Yuan and Lin, 2006], is an extension of the Lasso with the loss function given by:

$$L = \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda \sum_{g \in G} d_g ||\boldsymbol{\beta}_g||_2.$$

Here G is the set of groups with $g \in G$ the individual groups with weights d_g. Note that in this formulation we have a l_1/l_2-norm, with the l_1 norm applied over the groups and the l_2 norm applied over group elements, leading to a solution with sparse groups. The group Lasso is however not fit for overlapping groups, since an overlapping variable is selected if and only if all groups to which it belongs are selected. This is illustrated in Figure 3. Here we see a variable, IGF, with is present in 3 groups. It is furthermore demonstrated that the removal of any group containing the overlapping variable will remove that variable from the model. Thus, the variables selected are always the intersection of the complements of some groups.

(a) Example of an overlapping variable present in 3 groups. (b) By removing groups 1 & 3 only the non-overlapping variables in group 2 are chosen.

Figure 3: Overlapping variables in Group Lasso.

3.2 OGL

The Overlap Group Lasso (OGL) was proposed by [Jacob et al., 2009] to be able to select entire groups of variables in case of overlapping groups, thus to select unions of groups as support. The loss function proposed is of the following form:

$$L = \frac{1}{2}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||^2 + \lambda \sum_{g \in G} d_g ||\boldsymbol{v}^g||_2 \quad \text{such that} \quad \boldsymbol{\beta} = \sum_{g \in G} \boldsymbol{v}^g.$$

Here latent vectors $\boldsymbol{v}^g$ are created such that for each latent vector it's support lies in it's respective group $supp(\boldsymbol{v}^g) \subseteq g$ and the sum of these vectors equals the original coefficient vector: $\boldsymbol{\beta} = \sum_{g \in G} \boldsymbol{v}^g$.
An example of OGL is given in figure Figure 4. In the example we have 4 overlapping groups and the

original regression coefficient vector $\boldsymbol{\beta}$ is split into 4 latent vector γ^i, $i = 1, 2, 3, 4$ such that $supp(\boldsymbol{\gamma}^i) \subseteq g_i$ and $\boldsymbol{\beta} = \sum_{g \in G} \boldsymbol{\gamma}^g$.
In case of no overlap between groups the OGL gives the same result as the Group Lasso, simply set $\boldsymbol{v}^g = \boldsymbol{\beta}_g$ for all $g \in G$, with $\boldsymbol{\beta}_g$ indicating the vector with all coefficients not relating to variables in group g set to 0.
Also note that OGL is equivalent to Group Lasso with all overlapping variables duplicated, since:

$$\mathbf{X}\boldsymbol{\beta} = \mathbf{X} \sum_{g \in G} \boldsymbol{v}^g = \tilde{\mathbf{X}}\tilde{\boldsymbol{v}}$$

where $\tilde{\mathbf{X}}$ and $\tilde{\boldsymbol{v}}$ are the data-matrix and regression vector respectively with all overlapping variables duplicated. This is intuitively explained in Figure 5a. Note that the OGL-implementation in R, supplied by the function overlapgglasso2 in the MLGL package, also uses the variable duplication method, see Figure 5b.
It is important to note the effect of the weights in OGL. A common choice of the weights d_g in Group Lasso is $d_g = \sqrt{|g|}$, with $|g|$ the cardinality of group g. For OGL the weights are more influential than for Group Lasso, especially in case of subgroups. Imagine we have 3 groups with group structure $g_1 = (x_1, x_2), g_2 = (x_1), g_3 = (x_2)$. If we take the weights for all groups to be equal, than the larger group with always be chosen, since it contains most . Thus, for groups g, g' with $g \subset g'$ we need $d_g < d_{g'}$ for group g to not be redundant. Similarly, if we choose the weight of the larger group to be too large, the combination of smaller groups will always be chosen instead of the large group and the large group will be redundant. In general, for no group to be redundant, the weights should be of the form $d_g = |g|^m$ with $m \in (0, 0.5)$.

Group	1	2	3	4
Variables	x_1, x_2	x_2, x_3	x_1, x_3	x_3, x_4

(a) Overlapping groups with their variables.

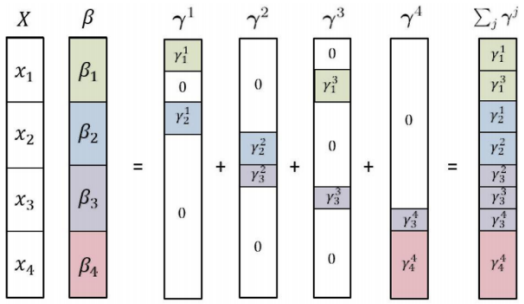
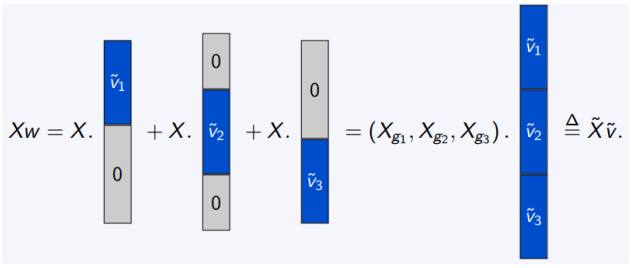


Figure 1. The coefficient decomposition of overlapping group lasso.

(b) Original coefficients vector $\boldsymbol{\beta}$ is split into 4 latent vectors $\gamma_1, ..., \gamma_4$.

Figure 4: Example of OGL with 4 overlapping groups.



(a) Example showing equivalence of OGL and Group Lasso with variable duplication in case of 3 groups.

```
overlapgglasso2<-function (X, y, var, group, ...){
    ord <- order(group)
    groupord <- group[ord]
    varord <- var[ord]
    groupb <- cumsum(!duplicated(groupord))
    Xb <- X[, varord]
    res <- gglasso(Xb, y, groupb, ...)
```

(b) R-code performing OGL uses variable duplication.

Figure 5: The equivalence of OGL and Group Lasso with variable duplication.

3.3 SOGL

The Sparse Overlap Group Lasso (SOGL), as proposed by [Park et al., 2015], is an extension of the OGL to produce a sparse result both on the group and variable levels. The Loss function of SOGL is given by:

$$L = \frac{1}{2}||\mathbf{y} - \tilde{\mathbf{X}}\tilde{\boldsymbol{v}}||^2 + \lambda \left[(1-\alpha) \sum_{g \in G} d_g ||\tilde{\boldsymbol{v}}^g||_2 + \alpha ||\tilde{\boldsymbol{v}}||_1 \right] \quad \text{such that} \quad \boldsymbol{\beta} = \sum_{g \in G} \boldsymbol{v}^g.$$

Here $\tilde{\mathbf{X}}$ and $\tilde{\boldsymbol{v}}$ are the data-matrix and regression vector respectively with all overlapping variables duplicated. SOGL provides sparsity on the group level via the term $(1-\alpha)\sum_{g\in G} d_g||\tilde{\boldsymbol{v}^g}||_2$, while overall sparsity of the variables is provided by $\alpha||\tilde{\boldsymbol{v}}||_1$. The parameter α, which enters the model in addition to the parameters λ and d_g already present in OGL, controls the rate of 'group sparsity' and 'overall sparsity'. Note that with $\alpha = 0$ SOGL produces the solution of OGL while for $\alpha = 1$ the Sparse Group Lasso is obtained as introduced by [Simon et al., 2007]. SOGL was implemented in R by extending the existing SGL function with variable duplication.

3.4 Overlap LARS

During the internship a new penalized regression method was developed based on Least Angle Regression (LARS), an algorithm first proposed by [Fron et al., 2004]. The LARS algorithm sequentially builds the model, at each step adding the 'most informative' variable. The general form of the algorithm is as follows:

- Standardize predictors to mean 0 and SD 1. Initialize residuals $\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}$ and initialize regression coefficients $\boldsymbol{\beta} = 0$.
- Find predictor $\mathbf{x_j}$ most correlated with current residual $\mathbf{r}$.
- Move β_j towards it's least squares coeffcient until some other competitor $\mathbf{x_k}$ has as much correlation with the current residual $\mathbf{r}$ as $\mathbf{x_j}$ does.
- Move β_j and β_k in the direction of their joint least squares coefficient using the current residual $\mathbf{r}$ until some other variable $\mathbf{x_l}$ has as much correlation with the current residual.
- Continue until all p variables have entered the model.

The beauty of the algorithm is that the direction step size of each step can be analytically calculated, and thus the entire algorithm is completed in just p steps. Also, with a slight adjustment this algorithm yields the Lasso profile. The only extra adjustment needed is that when a non-zero coefficient hits zero, this variable should be dropped from the active set. The LARS algorithm thus provides a very elegant way to obtain the Lasso profile.
Note that it is not obvious how to generalize the LARS algorithm to a grouped structure, since one cannot calculate the correlation between a group of variables and the residual. With a slight adaptation, as first proposed by [Yuan and Lin, 2006] and later adjusted by [Alfons et al., 2016], the LARS algorithm can be made to handle groups of variables by using the R^2 between the groups of variables and the current residual. Thus, the LARS group algorithm can be written as:

- Initialize the response $\mathbf{y_0}$ and the matrix $\mathbf{X_j}$ of variables belonging to group $j = 1, ..., m$.
- Let $\hat{\mathbf{y}}_\mathbf{0}^\mathbf{j}$ be the fitted values of regressing $\mathbf{y_0}$ on $\mathbf{X_j}$. Now select the group j with maximum R^2, thus $\max_j R^2(\mathbf{y_0} \sim \mathbf{X_j})/|g_j| = \max_j cor^2(\mathbf{y_0}, \hat{\mathbf{y}}_\mathbf{0}^\mathbf{j})$.
- Move along equiangular direction among predictor groups until a new group with equal R^2 enters the model.
- Repeat until all predictor groups have entered the model.

Within this framework the direction step size can also be calculated analytically. For the exact algorithm we refer to [Alfons et al., 2016]. Note that within this framework we still need to take weights into account, as a large group with a large number of variables will likely have a larger R^2 than a small group with only a few variables. Here as weights the number of elements per group, $|g_j|$ is taken.
Also note that this algorithm is not designed to handle overlapping groups, especially not overlapping groups with subsets and singletons. Imagine having 3 groups with the group structure $g_1 = (x_1, x_2), g_2 = (x_1, x_3), g_3 = (x_1)$, with the support existing of only x_1 and x_2, but with x_1 having a higher correlation with the outcome than x_2. In that case we would expect our algorithm to first choose group g_3, existing out of the singleton x_1. At the next step, however, since x_1 is already chosen, we would not want our algorithm to also base it's choice on an already active variable. Thus, we would prefer that if a singleton or subset group has already been chosen, the larger groups containing that subset or singleton will be chosen based solely on the information of the non-selected items. Note that in this way the overlapping between groups is very intuitively incorporated in the model, since each group is chosen based solely on the elements that are not already active.
To achieve this the group LARS algorithm needs to be slightly adjusted. After a model has been selected all

other models are to be updated to not contain the already active variables. Thus, let A be the set of active variables and let g_j, $j = 1, ..., k$ be the not-yet chosen groups, then every non-chosen group will be updated as $\tilde{g_j} = g_j \setminus A$, with $\setminus$ being the set-difference.
This newly introduced selection method we call overlap group LARS and we implemented it in R based on the group LARS package RobustHD, implemented by [Alfons et al., 2016]. Note that the group LARS in the RobustHD package is written in C++ and it's integration in R uses pre-compiled C++ code. During the Internship, we first tried to change the C++ code to accomodate for the updating of groups. However, because of many interdependencies within the C++ code and no compiling instructions it was not possible to use the C++ code. Therefore, eventually, overlap Group LARS was implemented in R from scratch based on the implementation of Group LARS in C++ in the RobustHD package.

3.5 Other methods

During the Internship a literature search was conducted to search for newly proposed techniques to handle overlapping groups within the context of regression. Also, the literature search was conducted to establish whether the overlap group LARS had already been implemented. During the literature search a review was found, [Li et al., 2020], which covers sparse learning models for feature selection. In this review, published just recently (2020), only OGL and SOGL are mentioned as techniques applicable to overlapping groups in the regression setting.
Articles were found applying the above mentioned methods to real-life problems, an example being [Xie et al., 2017], were SOGL was used to study the proteomics in ovarian cancer.
Many articles were found on Lasso adaptations. One of the disadvantages of Lasso it that the penalty term, apart from providing a sparse solution, also shrinks the coefficients, which is specifically problematic for large 'true' regression coefficients. New techniques proposed to battle this problem, which are SCAD, MCP and Adaptive Lasso were found, see [Breheny,]. These techniques refer to the oracle property, which means that the penalized estimator obtained is asymptotically equivalent to the ideal estimator obtained only with support variables and without penalization. Based on the article by [Leeb and Pötscher, 2008], which criticizes the desirability of the oracle property, it was however decided to not look further into these new Lasso extensions. Lastly, a tree-structured group Lasso method was found [Kim and Xing, 2010] which can be used in case of overlapping groups. However, here the groups and the overlap should have a hierarchical tree structure, which is not applicable in omics data.

4 Results

After performing the literature search for overlapping group penalization, our focus was directed towards four techniques: Lasso, Overlap Group Lasso (as proposed by [Jacob et al., 2009]), Sparse Overlap Group Lasso (as proposed by [Park et al., 2015]) and the Overlap Group LARS developed during the internship (based on the Group LARS as proposed by [Alfons et al., 2016]).
During this internship we would like to get an idea, using simulations, which of these methods works best on what kind of data. We assume that the data is of the form discussed in section 2 and that the data contains groups with some degree of overlap. The variation between data-sets of this type can be of the following form:

- Percentage of data that functioning as support (level of sparsity).
- Percentage of overlapping data.
- Degree of overlap of the data, which is the number of groups overlapping variables share. Using our synthetic data-set we can make overlapping variables shared by 2,3 or 4 groups.
- Presence of subgroups and singletons.

When comparing the 4 methods we are interested in both the prediction accuracy as well as the ability of the methods to select the correct support.
We first examined the data structure as shown in figure Figure 6. Here we have a total of 12 groups and overlapping variables of degree 2 and 4. The support is given by the encircled variables. To run simulations we used a sample size of $N = 100$, a correlation between variables in the same group of $\rho = 0.5$, an $\epsilon_{var} = 0.6$ and an $\epsilon_y = 1$. For the exact interpretation of these parameters we refer to section 2. For the OGL and SOGL we used as weights $d_g = |g|^{0.3}$ while for SOGL we used the mixing parameter $\alpha = 0.5$. We shall be using these parameter settings for all results unless explicitly mentioned otherwise. The support chosen by the different

techniques can be seen in figure Figure 7. Here the probability of selecting a certain variable is shown, averaged over 5 runs. The true support is indicated by the transparent white band. Note that for Lasso, OGL and SOGL on the x-axis the different values of the penalty parameter λ, ranging from $\exp^{-10}$ to $\exp^{3}$, are shown. For overlap LARS, in contrast, the steps of the algorithm are shown on the x-axis. We see from Figure 7 that OGL performs best, in the sense that it first selects the correct support and all other variables are much later together. All other methods also select the correct support first, but for Lasso and overlap LARS the boundary between where the correct support and where non-contributing variables are selected is less clear.
Next we examined the prediction accuracy of these methods, the results of which can be seen in Figure 8. The same model parameters are used as with the support, but now a 5-fold Cross-validation is performed. In Figure 8a we have plotted the RMSE (Root Mean Squared Error) of Lasso, OGL and SOGL against the different values of λ. We see that overall OGL performs best, being however close to Lasso. We see that the prediction accuracy of all methods approaches 0 as λ decreases, but with very low λ the accuracy increases again. This is most likely due to overfitting on the training set. As for the overlap LARS, on the x-axis the algorithm steps are shown. Here we also observe a very quick decrease of the RMSE towards 0.
When considering both the ability of a method to pick the right support while simultaneously maintaining a good prediction, OGL seems to perform best.

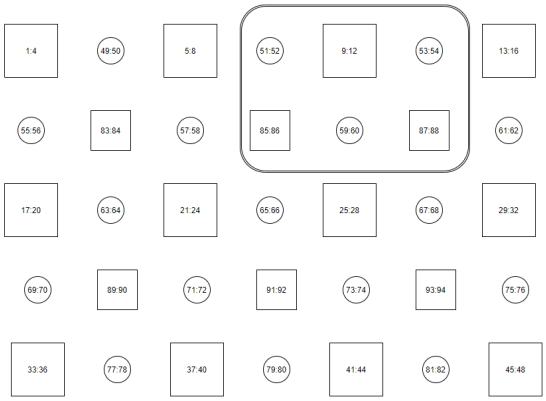
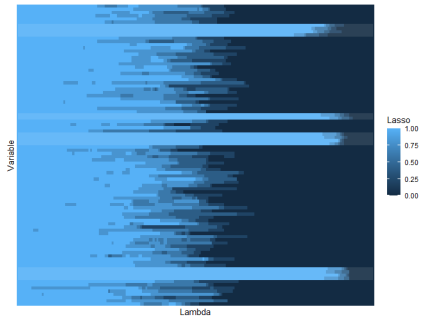
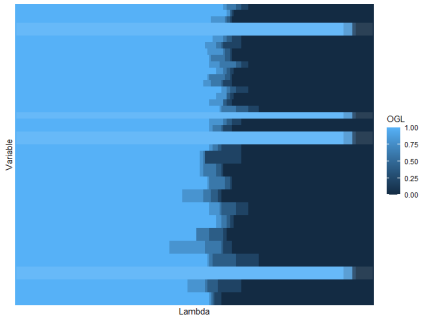


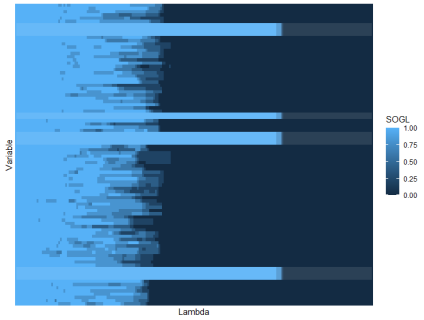
Figure 6: Dataset with a small support



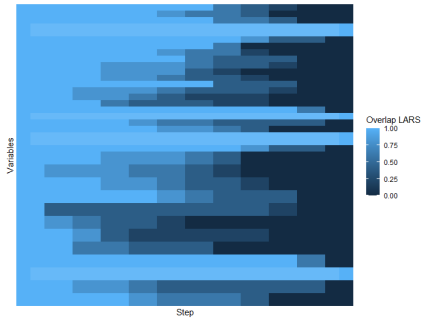
(a) Lasso



(b) OGL



(c) SOGL



(d) Overlap LARS

Figure 7: Performance of the different overlap group penalization methods in choosing the correct support on the dataset as given in Figure 6.

(a) Lasso, OGL and SOGL

(b) Overlap LARS

Figure 8: Prediction performance of the different overlap group penalization methods on the dataset as given in Figure 6.

Next we examined the behaviour of the different methods in case of a large proportion of variables acting as support. For this we have used the dataset that can be seen in Figure 9. Here, as before, the support is encircled. We can observe in Figure 10 that, as before, OGL seems to best select the true support, closely followed by Lasso and SOGL. In this context LARS often selects the wrong variables as support, behaviour we do not observe with the other methods. This could be explained by the inner workings of overlap LARS. Since our support exists out of 3 partly overlapping groups, when one of these groups is selected the overlapping variables no longer contribute to the selection of the other 2 groups. Therefore other groups, with more variables and more information, might be preferable. When looking at the prediction accuracy, given in Figure 11, OGL also seems to perform best.

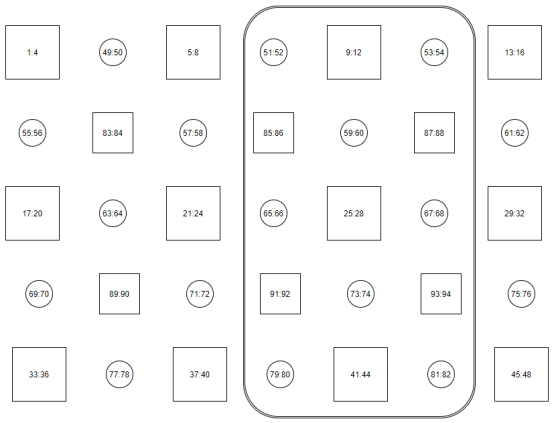
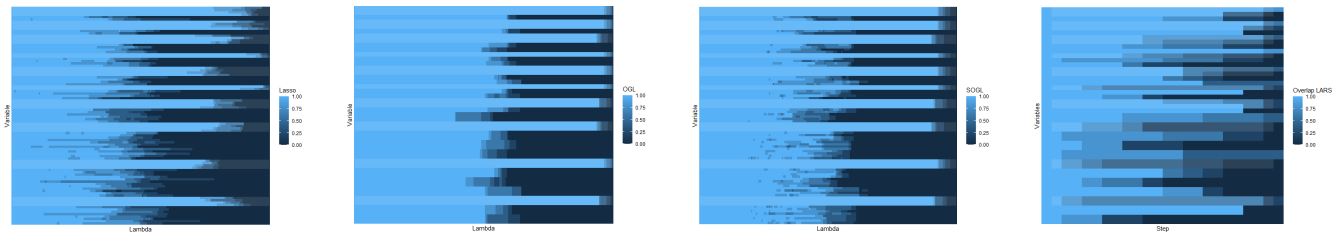


Figure 9: Dataset with a large support



(a) Lasso

(b) OGL

(c) SOGL

(d) Overlap LARS

Figure 10: Support selection on a dataset with a large support, see Figure 9.

(a) Lasso, OGL and SOGL (b) Overlap LARS

Figure 11: Prediction on a dataset with a large support, see Figure 9.

Finally, we examined whether a large percentages of overlapping variables (figures 12, 13, 14) or the presence of subgroups (figures 15, 16, 17) would make OGL perform worse than the other techniques. To achieve a large percentage of overlapping variables (Figure 12), we increased the number of variables that overlap, but did not change the overall group structure. Here we see that OGL still performs best, both in terms of support as well as in terms of Prediction. Overlap LARS seems to perform poorly, possibly because of the fact that many groups contain the overlapping variables and LARS chooses one of the other groups as support.

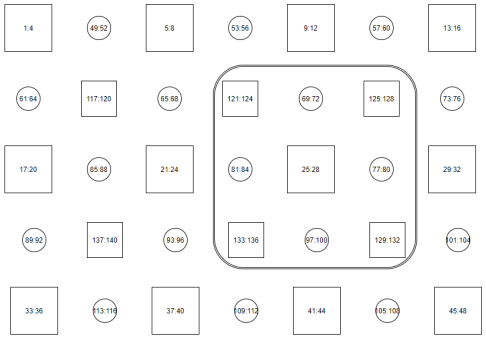
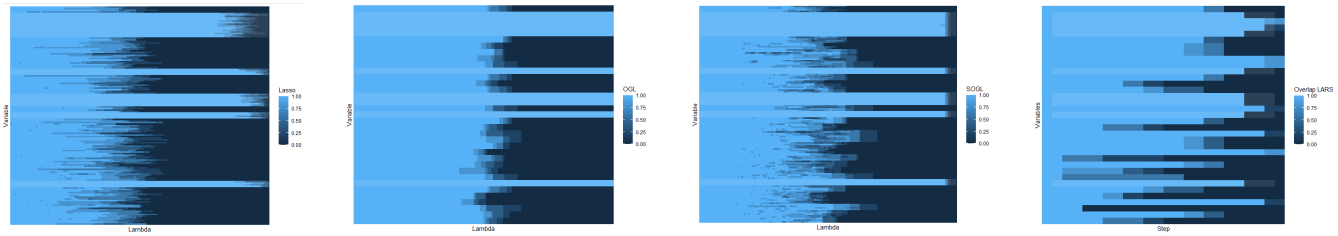
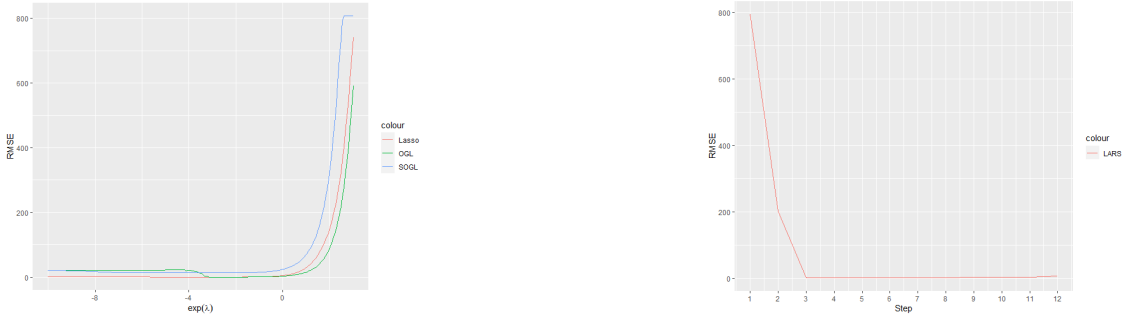


Figure 12: Dataset with a large percentage of overlapping variables.



(a) Lasso (b) OGL (c) SOGL (d) Overlap LARS

Figure 13: Support selected on a dataset which a high percentage of overlapping variables, see Figure 12.



(a) Lasso, OGL and SOGL (b) Overlap LARS

Figure 14: Prediction performance on a dataset which a high percentage of overlapping variables, see Figure 12..

Figure 15 shows a dataset wherein all groups of variables with overlap degree 4 are also included as subgroups. Thus, besides the normal groups, the following 6 subgroups are introduced: $117, ..., 120$; $121, ..., 124$; $125, ..., 128$; $129, ..., 132$; $133, ..., 136$; $137, ..., 140$. On this dataset we see that, judging by the support, OGL performs consistently well, closely followed by SOGL and Lasso. However, here the advantages of overlap LARS seem to manifest themselves. Overlap LARS first selects the correct subgroups and only then selects the remainder of the correct support. Thus, in this way overlap LARS can truly pick out the most important variables and select the other variables not based on their group allegiance but on their contribution to the outcome. When looking at the Prediction we see OGL is a clear winner.

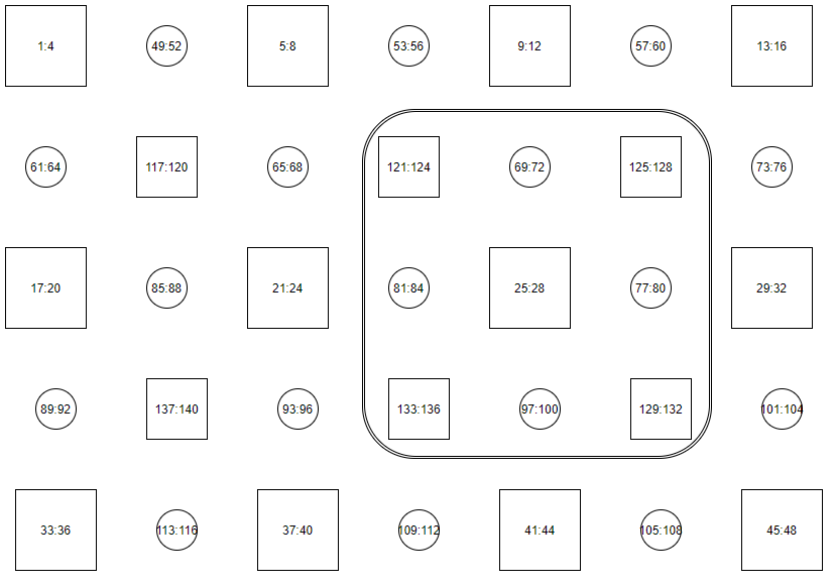
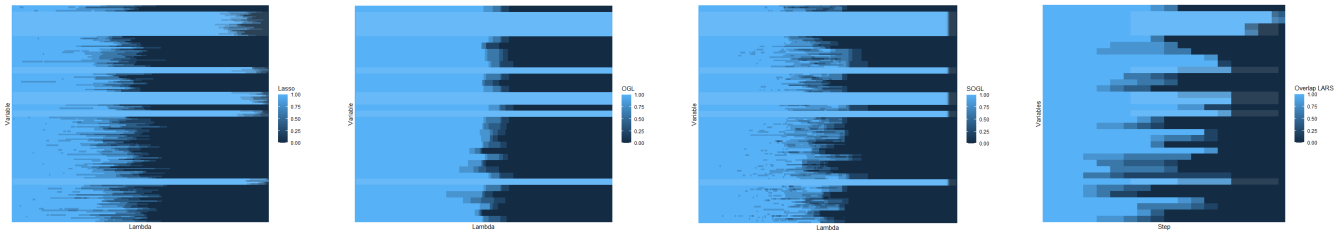
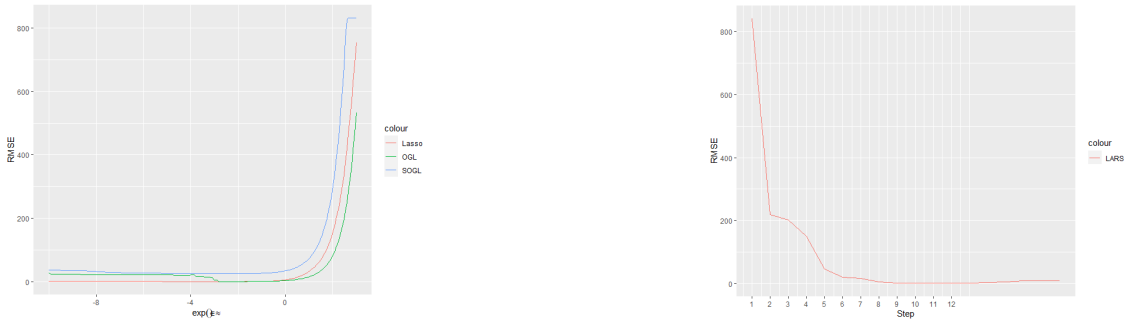


Figure 15: Dataset with all variables belonging to 4 groups also introduced as subsets.



(a) Lasso (b) OGL (c) SOGL (d) Overlap LARS

Figure 16: Performance of the different overlap group penalization methods in choosing the correct support on the dataset as given in figure Figure 12.



(a) Lasso, OGL and SOGL (b) Overlap LARS

Figure 17: Prediction performance of the different overlap group penalization methods on the dataset as given in figure Figure 9.

5 Discussion

During this Internship four methods used for overlapping group regression were compared: Lasso, OGL, SOGL and overlap LARS. Based on simulations we can conclude that OGL is a robust method yielding good prediction and support selection properties as compared to the other methods. We should however be careful in extrapolating these results to real omics data, since the synthetic data used in this report had a very specific structure with grouped variables having a certain degree of correlation, something which is not necessarily true in omics data. Also, the simulations were not really conducted in a high-dimensional setting, with for example Figure 9 having 82 variables with a sample size of 100. Furthermore, all methods discussed here have many parameters that can be tuned, most notably the weights. In the simulations the weights were always set to $d_g = |g|^{0.3}$, but many other options are possible. The newly proposed overlap LARS seems to perform worse than OGL and SOGL, partly because the comparison between the methods is difficult. Within the context of the current Internship the prediction and support selection of overlap LARS was plotted against the step number, making the comparison with OGL and SOGL difficult, as the support selection and prediction of these methods was plotted against λ. If more time would have been available it would be have been possible to compare them on a more similar scale.
Interesting topics for further research would be to compare the different methods on real omics data. Also, it is worth to obtain a more practical point of view of these methods and discover what problems present themselves within the context of omics and what methods would be best fit to solve these problems. It would then become clearer what kind of overlapping group data one should focus on. The first group Lasso was not invented within the context of a biological application, but rather to select groups of factor variables in a regression model. Thus, the methods were not invented with a biological application in mind. Finally, several other implementations of overlap LARS were thought of, one of them involves selecting groups based on an F-test rather than R^2. In this case, however, there would most likely be no analytical solution to the step size and thus Forward Selection would have to be used rather than LARS.
In conclusion we can say that based on the simulations we have conducted Overlap Group Lasso is a robust overlapping group penalization technique which yields good support selection properties as well as prediction properties on a wide variety of data.

References

[Alfons et al., 2016] Alfons, A., Croux, C., and Gelper, S. (2016). Robust groupwise least angle regression. *Computational Statistics and Data Analysis*, 93:421–435.

[Breheny,] Breheny, P. Adaptive lasso , MCP , and SCAD. (Bios 7600):1–34.

[Fron et al., 2004] Fron, B. Y. B. R. E., Astie, T. R. H., Ohnstone, I. A. I. N. J., and R-eb, G. (2004). LEAST ANGLE REGRESSION. *Annals of Statistics*, 32(2):407–499.

[Jacob et al., 2009] Jacob, L., Obozinski, G., and Vert, J. P. (2009). Group lasso with overlap and graph lasso. *Proceedings of the 26th International Conference On Machine Learning, ICML 2009*, pages 433–440.

[Kim and Xing, 2010] Kim, S. and Xing, E. P. (2010). Tree-guided group lasso for multi-task regression with structured sparsity. *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, pages 543–550.

[Leeb and Pötscher, 2008] Leeb, H. and Pötscher, B. M. (2008). Sparse estimators and the oracle property, or the return of Hodges' estimator. *Journal of Econometrics*, 142(1):201–211.

[Li et al., 2020] Li, X., Wang, Y., and Ruiz, R. (2020). A Survey on Sparse Learning Models for Feature Selection. *IEEE Transactions on Cybernetics*, pages 1–19.

[Park et al., 2015] Park, H., Niida, A., Miyano, S., and Imoto, S. (2015). Sparse overlapping group lasso for integrative multi-omics analysis. *Journal of Computational Biology*, 22(2):73–84.

[Simon et al., 2007] Simon, N., Friedman, J., Hastie, T., and Tibshirani, R. O. B. (2007). A sparse-group lasso. pages 1–13.

[Xie et al., 2017] Xie, H., Wang, W., Sun, F., Deng, K., Lu, X., Liu, H., Zhao, W., Zhang, Y., Zhou, X., Li, K., and Hou, Y. (2017). Proteomics analysis to reveal biological pathways and predictive proteins in the survival of high-grade serous ovarian cancer. *Scientific Reports*, 7(1):1–10.

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