CHALMERS

GÖTEBORGS UNIVERSITET

Regulatory Driven Clustering of Single-Cell Data

Clustering of single-cell RNA sequencing from glioblastoma with a novel mathematical method

Master's Thesis in Mathematical Statistics

Kári Kristjánsson

Department of mathematical sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2022

Regulatory Driven Clustering of Single-Cell Data

Clustering of single-cell RNA sequencing from glioblastoma with a novel mathematical method

Master's Thesis in Mathematical Statistics

Kári Kristjánsson

Supervisor: Rebecka Jörnsten
Examiner: Philip Grelee

Department of mathematical sciences
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2022

Abstract

Cancer is a leading cause of death worldwide. Single-cell RNA sequencing has arisen as an important method to explore the gene expression of biological cells, including cancer cells. In this study, we deployed a computational algorithm known as *ScRegClust* to dissect single-cell RNA-sequencing (scRNA-seq) data from brain tumors. This method uncovers modules of co-expressed genes, and identifies corresponding regulators, such as transcription factors and kinases. We sought to discern whether distinct scRNA-seq datasets could mutually inform each other by examining the patterns of gene clustering and regulatory mechanisms. The goal was to leverage this knowledge to guide the algorithm in a subsequent run, thereby enhancing its performance. Although the preliminary findings from simulated data offered promising prospects, transitioning to real-world data consisting of glioblastomas presented considerable hurdles. While our results shed light on the intricacies of reconstructing regulatory programs, the overall performance did not meet our initial projections. These findings underscore the complexity of and challenges associated with scRNA-seq analysis, underscoring the necessity for further exploration and refinement of current methodologies. This research enriches the field of data integration in cancer genomics and lays a foundation for future efforts aimed at refining regulatory-driven clustering of single-cell data.

Contents

Acknowledgements

I wish to express my sincere gratitude to my supervisor Rebecka Jörnsten for sharing your vast knowledge and valuable insights and for providing me with exceptional guidance and counsel during this work. Throughout our engaging discussions your insightful comments have provided invaluable support. I'm also very grateful to Felix Held, for supporting me, teaching me Tetralith and always responding clearly and rapidly to all me questions. Finally I am very thankful to Ida Larsson for generously guiding me through the biological aspects of this work.

1 Introduction

Cancer is a leading cause of death worldwide, accounting for nearly 10 million deaths in 2020 (Bray *et al*, 2021). The most common cancers are breast cancer, lung cancer, prostate cancer, and colorectal cancer (Sung *et al*, 2021). Cancer arises from the transformation of normal cells into tumor cells, which involves a step-wise process from a pre-cancerous lesion to a malignant tumor (Sarkar *et al*, 2013). An interaction between genetic factors and external agents including lifestyle factors, certain viruses, and environmental exposures initiates this process. Conventional cancer therapy is based on different combinations of treatments, mainly surgery, radiation, and chemotherapy (Debela *et al*, 2021). Although cancer survival is improving, mortality is still high (Bray *et al*, 2021), especially for cancers such as glioblastoma, which will be the focus in this thesis. A better understanding of the complex biology of tumor cells is needed to further enable the improvement of outcomes.

The recent development of next-generation sequencing (NGS) offers improved recognition of complex biological systems. NGS-based technologies for transcriptomics are increasingly focused on the characterization of individual cells. Single-cell RNA sequencing (scRNA-seq) has arisen as an important method to explore the gene expression of biological cells. These single-cell analyses have the potential to uncover pathological processes including the occurrence of abnormal cell populations, the detection of adverse regulatory systems, and the recognition of atypical cell lineages. The ability to detect and describe outlier cells may offer a better understanding of therapy-resistant cancers and, thereby, lead to more successful treatment.

Accumulating evidence suggests that gene expression among cancer cells is highly variable. In addition to malignant cells, a tumor also consists of different types of immune and stromal cells which exhibit dynamic and reciprocal interactions. The aggressiveness of the cancer is not only related to the division and growth of cancer cells, but also to immune responses, remodeling of the extracellular matrix, and the formation of new blood vessels. Despite the underlying diversity of cellular gene expression, the majority of transcriptome analyses are still based on the assumption that cells from a given cancer are homogeneous. With this approach, it is likely that important cell-to-cell variability will be neglected. To better understand the stochastic biological processes that regulate gene expression in malignant tumors, a more precise assessment of the transcriptome in individual cells is required.

A study of the transcriptome by scRNA-seq requires the development of powerful and accurate

analytic methods. Existing approaches use clustering, but do not contain regulatory predictions and are only concerned with a grouping of cells or genes. In the present study, we apply the method developed by Ida Larson [1] (I.L) and Felix Held [2] (F.H) (Larsson *et al*, 2023), that swiftly constructs regulatory programs from a large dataset, and allows for the characterization of cellular heterogenicity and description of key regulatory factors. When applied to scRNA-seq, the method identifies modules of co-expressed genes and sets of regulators, such as transcription factors and kinases, that are linked to each module. As compared with conventional techniques, the method is both fast and flexible. The aim of this study was to discern whether distinct scRNA-seq datasets could mutually inform each other by examining the patterns of gene clustering and regulatory mechanisms in glioblastomas. We attempted to leverage this knowledge to guide the algorithm in a subsequent run in order to enhance its performance.

2 Background

2.1 Single-cell RNA sequencing

Single-cell RNA sequencing (scRNA-seq) is a cutting-edge technique used in molecular biology that allows researchers to study gene expression at the level of individual cells. In traditional RNA sequencing (RNA-seq), all the RNA in a sample is extracted and sequenced, resulting in an average measurement of gene expression across all cells in the sample. However, scRNA-seq enables the measurement of gene expression for individual cells, providing a more detailed understanding of the molecular mechanisms underlying cellular behavior.

The scRNA-seq process involves several steps. First, individual cells are isolated and lysed to release their RNA content. The RNA is then reverse transcribed into complementary DNA (cDNA) and amplified. Next, the cDNA is sequenced using high-throughput sequencing technology. Finally, the resulting sequencing reads are aligned to a reference genome or transcriptome, allowing researchers to identify which genes are expressed in each individual cell.

There are several advantages using scRNA-seq. One major advantage is that it allows for the identification of rare cell types or cell subpopulations that may be missed using traditional RNA-seq methods. Additionally, scRNA-seq enables the characterization of cellular heterogeneity within a tissue or population of cells, which can reveal important insights into cellular differentiation, development, and disease.

[1]Department of Immunology, Genetics and Pathology, Uppsala Universitet, SE-751 85 Uppsala, Sweden.
[2]Mathematical Sciences, Chalmers University of Technology, SE-41296 Gothenburg, Sweden.

However, scRNA-seq also has some limitations. One major limitation is that the technique is expensive and time-consuming, making it difficult to scale up to larger studies. Additionally, scRNA-seq data can be noisy and difficult to interpret, especially when dealing with rare or low-abundance transcripts.

Despite these challenges, scRNA-seq is a powerful tool for studying gene expression at the single-cell level. It has already led to numerous discoveries in a wide range of fields, including cancer biology, immunology, and developmental biology, and is likely to continue to revolutionize our understanding of cellular function and disease in the years to come.

2.2 Gene Regulatory Network (GRN) and Regulatory Programs

Gene regulatory networks play a crucial role in allowing cells to respond to environmental changes, such as nutrient availability or chemical signals from other cells (Barbuti *et al*, 2020). A cell's function is orchestrated by the synthesis of specific proteins, each governing a unique aspect of cellular activity.

The synthesis of these proteins is guided by the expression of genes. Genes act as blueprints for the transcription and translation processes that lead to protein production, thereby influencing cell function. Thus, each gene can be conceptualized as being 'active' or 'inactive' depending on whether or not it's being transcribed and translated into a protein at any given moment. The collective configuration of active genes at a certain time provides a snapshot of the cell's functional state. Each specific cellular function corresponds to a unique set of active genes.

The activation or inhibition of genes is mediated by proteins, which are products of other genes. This gives rise to a network of interactions, where each node represents a gene, and the edges symbolize the interactions between them. In this gene regulatory network, the relationships among genes dictate the overall function and behavior of the cell.

The gene regulatory network can be looked at as a static map of possible integration and relations between genes. Gene regulatory programs are, however, the specific use of this map, where different parts of the network are activated temporally and by context-specific causes during different signals and processes. In simpler terms, the gene regulatory network can be looked at as a complete city map given all possible connections between points. The gene regulatory program would then be a specific path between two points A and B that could change depending on weather, time of day, traffic etc.

In this study, genes are categorized into two groups, target genes and regulatory genes:

Target genes are genes whose activity, i.e. their expression level, is controlled by regulatory genes. The influence of regulatory genes on target genes may result in up-regulation or down-regulation of the production of proteins.

Regulatory genes are genes that control the activation of other genes. Regulatory genes control target genes by producing molecules called transcription factors that turn on or turn off the target genes and influence whether they are expressed or not.

The algorithm *Scregclust* (Larsson *et al*, 2023) developed by I.L and F.H is designed to reconstruct cellular regulatory programs and identify essential kinases and transcription factors involved in different tumor cell states. The primary goal of *ScRegClust* is to cluster genes based on their regulatory programs. In this process, each cluster is associated with a specific regulatory program, and target genes are allocated to clusters based on the best fitting regulatory programs.

2.3 Estimation and Selection of Regulators

Scregclust (Larsson *et al*, 2023) utilizes the cooperative lasso method for selecting regulators. Cooperative lasso is a variant of lasso regression, which is commonly used in the analysis of biological data to identify a small subset of factors that are predictive of a given outcome from a large number of potential factors. The cooperative lasso was developed to address the limitations of the group lasso, which in turn was created to overcome the limitations of the regular lasso. The group lasso allows for predefined groups of covariates to be jointly included or excluded from a model. The cooperative lasso makes the additional assumption that groups are sign-coherent, meaning they contain either non-negative, non-positive, or null parameters.

2.3.1 Lasso

Lasso regression (Tibshirani, 1996) is a statistical method that combines variable selection and regularization in order to improve the prediction accuracy and interpretability of the model. Given a dataset X, Y, where $X \in \mathbf{R}^{n \times m}$ is a matrix of features and $Y \in \mathbf{R}^n$ is the response, the linear relationship is modeled as $Y = X\beta$ for some $\beta \in \mathbf{R}^m$. The Lasso coefficients $\hat{\beta}^{lasso}$ are found by minimizing the following quantity:

$$\hat{\beta}^{lasso} := \arg\min_{\beta} \left\{ (1/2)||Y - X\beta||_2^2 + \lambda||\beta|| \right\}$$

where $\lambda \geq 0$ is a tuning parameter that is chosen separately. When $\lambda = 0$, the penalty term has no effect, and the Lasso regression will yield the least square estimate. As λ increases, the Lasso will shrink the coefficients towards zero and eventually results in a null model in which all coefficients are zero. However, for intermediate values of λ, the Lasso can produce models with any number of variables.

2.3.2 Group-Lasso

The group lasso method was developed to address issues with the estimation and inference of parameters when a group structure among predictors and parameters is known (Yuan and Lin, 2006). The group lasso partitions the set of indexes $1, ..., p$ into K groups ${G_k}_{k=1}^{K}$, corresponding to predictors and parameters. The group lasso shrinkage estimator is defined as follows:

$$\hat{\beta}^{group} := \arg\min_{\beta \in \mathbb{R}} \{ \frac{1}{2} ||Y - X\beta||_2^2 + \lambda \sum_{k=1}^{K} w_k ||\beta_{G_k}|| \}$$

where G_k is the subset of indices defining the kth group of variables. The tuning parameter $\lambda \geq 0$ controls the overall amount of penalty, and $w_k \geq 0$ adapts the penalty between groups. The weights are often set according to group size $w = \sqrt{p_k}$.

2.3.3 Cooperative-Lasso

For problems that have a known group structure and where the groups are assumed to be sign-coherent, meaning they contain either non-negative, non-positive, or null parameters, an improvement can be made (Chiquet *et al*, 2012). To address such problems with the added assumption of sign-coherency within groups, a new penalty called the cooperative penalty was developed:

$$\hat{\beta}^{coop} := \arg\min_{\beta \in \mathbb{R}} \{ \frac{1}{2} ||Y - X\beta||_2^2 + \lambda \sum_{k=1}^{K} w_k (||\beta_{G_k}^{+}|| + ||\beta_{G_k}^{-}||) \}$$

where $\lambda \geq 0$ is a tuning parameter and $\beta_{G_k}^{+}$ and $\beta_{G_k}^{-}$ are the defined setting of all negative respectively positive elements to zero in β_{G_k}.

3 Methods

3.1 *ScRegClust* the Clustering Algorithm

We applied a two-step algorithm developed by I.L. and F.H. to analyze our data. This algorithm, which alternates between clustering target genes and linking regulators to each cluster, is designed to efficiently identify key regulatory factors and improve our understanding of the biological processes driving cancer. For a more detailed explanation and the capability of the algorithm see their publication Larsson *et al* (2023)

The algorithm begins by clustering the target genes into K clusters using the k-means algorithm with multiple restarts. Once an initial clustering is established, the algorithm proceeds to the first step, which involves identifying the most predictive regulators for each cluster. In the second step, the optimal regulators are then used to refine the cluster assignments of the target genes. This process is repeated until a stable cluster configuration is reached, or a maximum number of iterations is reached. This allows the algorithm to identify regulatory programs, which can uncover the underlying mechanisms of the disease, including cell-to-cell variability, abnormal cell populations, and key regulatory factors.

3.2 Technical Overview of *ScRegClust*

The technical details presented in this section have been derived and paraphrased from the original work of I.L and F.H (Larsson *et al*, 2023).

The algorithm requires five main inputs to be able to run:

1. The expression matrix, denoted as $\mathbf{Z}$, which serves as the primary dataset for analysis.
2. A vector of gene symbols to enable proper annotation and tracking of specific genes.
3. A regulator gene vector, which identifies potential regulatory genes.
4. A penalization parameter, λ, that helps control the number of regulatoers selected to each cluster.
5. The desired number of clusters, K, which dictates the initial number of clusters and the final clustering.

By providing these inputs, the algorithm can effectively perform gene expression analysis and cluster target genes and link key regulator genes to each cluster.

The expression matrix $\mathbf{Z}$ contains n samples, where each row represents a cell and each column represents a gene. Each entry in the matrix indicates the expression level of a particular gene in a given cell.

To initiate the clustering process, the algorithm first partitions the expression matrix, $\mathbf{Z}$, into two distinct submatrices: $\mathbf{Z}_t$ and $\mathbf{Z}_r$. The former submatrix, $\mathbf{Z}_t$, contains the expression levels of p_t target genes that are to be clustered, while the latter submatrix, $\mathbf{Z}_r$, consists of the expression levels of p_r potential regulator genes that may be linked to clusters of target genes. By segregating the two submatrices, the algorithm can distinguish between the target genes and potential regulatory factors and attempt to accurately identify the relationships between them.

To identify the relationships between the target genes and regulator genes the algorithm aims to determine the following elements (Larsson *et al*, 2023):

1. A $K \times p_t$ matrix of cluster membership $\mathbf{\Pi}$, where $\mathbf{\Pi}^{(i,j)} = 1$ if target gene j is in cluster i, and 0 otherwise.
2. A subset R_i of regulators for each cluster, with a maximum size of $|R_i| \leq N_r$, where N_r denote the total number of potential regulators.
3. Non-negative coefficients $\mathbf{B}_i$ of dimension $|R_i| \times p_t$ for each cluster.
4. A vector of signs $\boldsymbol{s}_i$ of size $|R_i|$ for each cluster.
5. Positive variance parameter σ_{ij}^2.

This is achieved by minimizing the following target function (Larsson *et al*, 2023):

$$\begin{aligned} &\underset{\mathbf{\Pi},R,\mathbf{B},\boldsymbol{s},\sigma}{\arg\min} \frac{1}{2} \sum_{i}^{K} \sum_{j}^{p_t} \Pi^{(i,j)} \left(n \log\left(\sigma_{ij}^2\right) + \frac{1}{\sigma_{ij}^2} \left\| Z_t^{(:,j)} - Z_r^{(:,R_i)} \text{diag}(s_i) B_i^{(:,j)} \right\|_F^2 \right) \\ &\text{such that} \quad |R_i| \leq N_r \quad \text{for all} \quad i = 1, ..., K. \end{aligned} \tag{1}$$

However, it is not practical to solve this optimization problem directly because of the high combinatorial number of regression problems that must be examined. Therefore, the algorithm uses a data-driven approximation to find a solution for equation (1).

3.2.1 Two-Step Data-Driven Approximation

The algorithm employs a data-driven approximation method to tackle the problem of a high combinatorial number of regression problems. This method alternates between two steps (Larsson

et al, 2023):

1. Establishing the structure of clusters, which involves determining R_i, $\boldsymbol{s}_i$, $\mathbf{B}_i$ for each i and σ_{ij} for each i and j.
2. Re-assigning the cluster membership, by picking Π.

The first step of determining R_i and $\boldsymbol{s}_i$ is also of combinatorial complexity and, therefore, further approximations are needed, which will be discussed in detail later.

Before the algorithm begins alternating between these two steps, the data is pre-processed and an initialization process is performed. During the pre-processing phase, the data is split into a training set and a testing set. This allows for unbiased estimation of the model quality during step two when cluster members are re-assigned (Larsson *et al*, 2023). The observations of target genes and regulator genes are randomly split into two equal sets of 50:50. If observations are stratified, such as observations from different patients, the data is split within each subgroup. From here, $\mathbf{Z}_{t,1}$ and $\mathbf{Z}_{r,1}$ denote the first part of the data split containing n_1 observations, whereas $\mathbf{Z}_{t,2}$ and $\mathbf{Z}_{r,2}$ denote the second part of the data split containing n_2 observations.

The data is then centered and scaled. This is done individually for both target genes and regulatory genes. Centering the data allows for more accurate comparison of the expression levels of different genes across different cells, as the data will be normalized to a common reference point. By centering the data clustering can be improved. This is important when working with single-cell RNA sequencing (scRNA-seq) where differences in cell size, RNA content, and other technical factors can introduce variability in the gene expression measurements. The regulator genes are also scaled to standard deviation 1, which helps with regularization since coop-Lasso (Regularization method used in the algorithm, which will be descried later) penalizes by the magnitude of the coefficients. When predictors (regulator genes) are on different scales the regularization may unfairly penalize predictors with larger magnitudes. Scaling regulator genes to standard deviation can help with this issue.

For initialization, the first cluster membership matrix Π_0 is determined. For this, a cross-correlation matrix of $\mathbf{Z}_{t,1}$ and $\mathbf{Z}_{r,1}$ is computed. Given the number K, the algorithm clusters the target genes $\mathbf{Z}_{t,1}$ using the k-means++ algorithm (Arthur and Vassilvitskii, 2007) with multiple restarts into K clusters. The k-means++ algorithm uses the cross-correlation matrix as input. By using the cross-correlation matrix as input, target genes will be clustered based on their degree of linear

correlation with the regulators genes, whereby target genes with similar correlation coefficients are grouped together.

3.2.2 Step 1: Cluster Structure Determination

When the initial cluster membership matrix is determined, the first step of the two-step alternating process begins. The goal of the first step is to identify the regulators that have the strongest linear relationship with the target genes in each cluster. To do this, the optimization problem outlined in Equation (1) is solved for each cluster individually.

The set of target genes in a cluster i is denoted C_i. However, determining the R_i and $\boldsymbol{s}_i$ variables is computationally infeasible due to the high number of possible sets of regulators and their signs, $\sum_{k=0}^{N_r} \binom{p_r}{k} \times 2^k$. To address this, a further approximation is made by using the cooperative Lasso (coop-Lasso) algorithm, first proposed by Chiquet *et al* (2012). The coop-Lasso solves the following modified version of the optimization problem in Equation (1) (Larsson *et al*, 2023):

$$\mathbf{B}_i = \arg\min_B \frac{1}{2} \sum_{j \in C_i} \frac{1}{n_1 \sigma_{ij}^2} \left\| \mathbf{Z}_{t,1}^{(:,j)} - \mathbf{Z}_{r,1} \mathbf{B}^{(:,j)} \right\|_F^2 + \lambda \sum_{k=1}^{p_r} \mathbf{w}_i^{(k)} \left(||\mathbf{B}_+^{(k,:)}||_2 + ||\mathbf{B}_-^{(k,:)}||_2 \right) \tag{2}$$

where $\lambda \geq 0$ is a penalty parameter related to N_r, which controls the number of regulators that will be selected; $\mathbf{w}_i$ is a vector of weights that is specific to each cluster and will be described below; $\mathbf{B}^{(k,:)}$ refers to the k-th row in $\mathbf{B}$; $\mathbf{B}_+^{(k,:)}$ and $\mathbf{B}_-^{(k,:)}$ are defined by setting all negative or all positive elements in $\mathbf{B}^{(k,:)}$ to zero.

The coop-Lasso works similarly to the group-Lasso (Yuan and Lin, 2006) by identifying which groups of coefficients should be included in the model by setting the coefficients of the excluded groups to zero. In this context, the groups are the rows of $\mathbf{B}$, each of which corresponds to a regulator. In addition to promoting sparsity at the group level, the coop-Lasso also promotes sparsity within groups and aims for consistent signs within groups. As a result, the rows of the estimated coefficient matrix $\mathbf{B}_i$ are typically non-negative, non-positive, or zero. However, if coefficients are close to zero or λ is small, it can occur that coefficients within groups have both negative and positive signs. To handle this, the sign of each regulator is assigned $\boldsymbol{s}_i^{(l)} = \text{sgn}\left(\sum_k \mathbf{B}_i^{(l,k)} / |C_i|\right)$. The set of active regulators R_i is then determined by the non-zero rows of B_i. When cluster members are fixed, the coop-Lasso in Equation (2) provides a solution to the optimization problem in Equation (1).

In their publication Larsson *et al* (2023), I.L and F.H used an over-relaxed Altering Direction Method of Multipliers (ADMM) algorithm (Boyd *et al*, 2011) to solve the optimization problem. ADMM decomposes the optimization problem into two sub-problems: one related to the loss and the other to the penalty. To solve the sub-problem corresponding to the penalty, the explicit form of the proximal operator of the coop-Lasso as described in Chiquet *et al* (2012) is used. To accelerate convergence, an adaptive ADMM step-length and over-relaxation parameters, as described in Xu *et al* (2017), are utilized.

Before estimating $\mathbf{B}_i$, the variances σ_{ij}^2 and the weights $\boldsymbol{w}_i$ are determined. To accomplish this, the ordinary least squares (OLS) estimate $\hat{\mathbf{B}}_i$ of the regression coefficients are determined as follows:

$$\hat{\mathbf{B}}_i = \arg\min_{\mathbf{B}} \left\{ \mathbf{Z}_{t,1}^{(:,C_i)} - \mathbf{Z}_{r,1}\mathbf{B} \right\}.$$

The variances are then computed in an unbiased manner using:

$$\hat{\sigma}_{ij}^2 = \frac{1}{n_1 - p_r} \left\| \mathbf{Z}_{t,1}^{(:,j)} - \mathbf{Z}_{r,1}\hat{\mathbf{B}}_i^{(:,j)} \right\|_2^2. \tag{3}$$

The weights are determined using an approach inspired by the adaptive Lasso (Zou, 2006):

$$\boldsymbol{w}_i^{(k)} = \sqrt{\frac{|C_i|}{\left\| \hat{\mathbf{B}}_i^{(k,:)} \right\|_2}}.$$

Generating weights in this way improves the selection of regulators. By employing this approach, regulators with larger magnitudes in the OLS estimate will receive less penalty and have a higher likelihood of being chosen by the cooperative lasso.

In summary, the first step of the algorithm uses a data-driven approximation approach known as the cooperative Lasso (Chiquet *et al*, 2012) to determine the most predictive regulators for each cluster. This is done by solving the optimization problem in equation (2) and utilizing the Alternating Direction Method of Multipliers (ADMM) (Boyd *et al*, 2011) for efficient computation. After step one, the active regulators and signs are determined and are used in the next step of the algorithm.

3.2.3 Step 2: Cluster Assignment

In the second step of the algorithm, the updated cluster structure determined in the first step is used to re-assign cluster membership. This involves estimating non-negative coefficients $\mathbf{B}_i$

for each cluster and residual variances σ_{ij}^2 for each gene and cluster. To estimate these coefficients Larsson *et al* (2023) use a sign-constrained linear regression by performing non-negative least squares (NNLS) regression (Meinshausen, 2013) and taking into account the signs previously determined. The NNLS regression estimator is a constrained least-squared problem where the estimated coefficients are not allowed to be negative. In general the coefficients are estimated by the following:

$$\hat{\beta} := \arg\min_{\beta} ||Y - X\beta||_2^2 \quad \text{s.t.} \quad \min_k \beta_k \geq 0$$

In comparison with least squares (LS) problems, NNLS has no generic formula for solutions. Even though NNLS is a convex optimization problem, multiple iterative algorithms and gradient methods have been used to solve NNLS.

Larsson *et al* (2023) employ a modified version of the approach described by Nguyen and Ho (2017) to compute the NNLS coefficients efficiently.

The modified version can compute responses on a matrix of values rather than just a single vector. Further responses are removed that have met a specific convergence criterion to prevent unnecessary computations. By re-estimating the coefficients in this manner, any bias introduced by the coop-Lasso is removed, and the assigned signs are enforced (Larsson *et al*, 2023). It is crucial to note that coefficients are also estimated for target genes that were not previously included in the cluster, which is essential for re-allocating cluster membership.

To be specific, the optimization problem that NNLS solves is given by:

$$\arg\min_{\mathbf{B}} \frac{1}{2} \left\| \mathbf{Z}_{t,1} - \mathbf{Z}_{r,1}^{(:,R_i)} \text{diag}(\boldsymbol{s}_i)\mathbf{B} \right\|_F^2 \quad \text{subject to} \quad \mathbf{B} \geq 0. \tag{4}$$

Here, $\mathbf{Z}_{t,1}$ is the matrix of gene expression values for the target genes, and $\mathbf{Z}_{r,1}^{(:,R_i)}$ is the matrix of gene expression values for the regulator genes in the i^{th} cluster. The diagonal matrix $\text{diag}(\boldsymbol{s}_i)$ contains the signs of the coefficients estimated in the first step of the algorithm. The objective is to minimize the Frobenius norm of the difference between the predicted and actual expression values, subject to the constraint that the coefficients are non-negative. The signs from the coop-Lasso estimation guide how the coefficients from NNLS are incorporated into the model, effectively enforcing the signs determined by the coop-Lasso. It ensures that the coefficients estimated by NNLS will respect the signs that were obtained during the coop-Lasso stage.

The residual variances needed for the second step are estimated as in Equation (3) in the following way:

$$\hat{\sigma}_{ij}^2 = \frac{1}{n_1 - |R_i|} \left\| \mathbf{Z}_{t,1}^{(:,j)} - \mathbf{Z}_{r,1} \mathbf{B}_i^{(:,j)} \right\|_2^2.$$

Where the OLS estimate $\hat{\mathbf{B}}_i$ are replaced by the NNLS coefficients estimated by Equation (4) and n_p is replaced by $|R_i|$.

To pinpoint target genes that aren't a good fit for any of the clusters, Larsson *et al* (2023) use a technique called Rag Bag clustering. This is achieved by calculating the predictive R^2-value for each target gene and cluster based on the residuals of predicting $Z_{t,2}$ from $Z_{r,2}\text{diag}(\mathbf{s}_i)\mathbf{B}_i$. The highest R^2 value across clusters for each target gene is recorded. If this value falls below an user-specified threshold then that gene was considered as noise or badly predicted within all clusters and placed into its own separate noise cluster (also referred to as a Rag Bag), leaving only remaining target genes being taken into consideration during subsequent steps.

Apart from the main inputs listed in the beginning of section 3.2, the algorithm allows the user to incorporate prior knowledge of biological relationships among target genes using an indicator matrix J. This matrix has a size of $q \times q$, where $J_{(i,j)} = 1$ indicates that genes i and j have a known biological relationship, and $J_{(i,j)} = 0$ indicates that no known biological relationship exists between them.

To use this prior knowledge, for a fixed target gene j, the vector $J^{(j,:)}$ is constructed by setting 1 for the genes that have a biological relationship with gene j, and 0 for the others. Then, the algorithm calculates the fraction $f_{ij} = J^{(i,:)}{\mathbf{\Pi}^{(i,:)}}^{\mathrm{T}} / \sum_g J^{(i,g)}$, where $\mathbf{\Pi}$ is the most updated membership matrix. This fraction represents the biological evidence supporting the assignment of gene j to cluster i. If cluster i is empty, $f_{ij} = 0$.

To normalize the fractions across clusters, the algorithm calculates the denominator $\sum_c f_{cj}$, and then divides each fraction f_{ij} by it to obtain the normalized fraction $p_{ij} = f_{ij} / \sum_c f_{cj} + \alpha$, where $\alpha = 10^{-16}$ is a small baseline parameter added to avoid taking the log of zero later when updating the cluster membership matrix.

To compute the likelihood of a target gene j belonging to a cluster i, given the values of R_i, $\mathbf{s}_i$, $\mathbf{B}_i$, and σ_{ij}^2, the algorithm computes the likelihood for observation l as follows (Larsson *et al*, 2023):

$$\mathbf{L}_j^{(l,i)} = \frac{1}{\sqrt{2\pi\sigma_{ij}^2}} \exp\left(-\frac{1}{2\sigma_{ij}^2}\left(\mathbf{Z}_{t,2}^{(l,j)} - \mathbf{Z}_{r,2}^{(l,R_i)}\text{diag}(\mathbf{s}_i)\mathbf{B}_i^{(:,j)}\right)^2\right),$$

where $\mathbf{Z}_{t,2}^{(l,j)}$ is the expression level of gene j in observation l, and σ_{ij}^2 is the variance of gene j in cluster i.

To update the cluster membership of target gene j, the algorithm normalizes the likelihoods across all clusters c by setting $\mathbf{L}_j^{(i,l)} \leftarrow \mathbf{L}_j^{(i,l)} / \sum_c \mathbf{L}_j(c,l)$. Then, for each observation l, the algorithm computes a vote $\mathbf{v}_j^{(l)} = \arg\max_i (1-\mu)\log \mathbf{L}_j^{(i,l)} + \mu \log p_{ij}$, where $\mu \in [0,1]$ controls the strength of the prior on the allocation process and p_{ij} is the prior probability of gene j belonging to cluster i.

The algorithm assigns gene j to the cluster that receives the majority of votes. In an effort to mitigate bias, target genes undergo a randomized sequence for processing. In each iteration, the prior probabilities for each gene are calculated by considering the updated cluster assignments as well as the previous cluster assignments for genes that have not been updated yet.

In summary, the second step of the algorithm uses non-negative least squares (NNLS) to estimate non-negative coefficients and residual variances to re-assign cluster membership for the updated cluster structure determined in the first step. The modified algorithm can efficiently compute responses on a matrix of values and removes responses that have met a specific convergence criterion, and any bias introduced by the coop-Lasso is removed by re-estimating the coefficients. The algorithm allows the user to incorporate prior knowledge of biological relationships among target genes using an indicator matrix. The likelihood of a target gene belonging to a cluster given the values of R_i, $\mathbf{s}_i$, $\mathbf{B}_i$, and σ_{ij}^2 is computed using a likelihood function. This step allows the algorithm to identify target genes that do not fit well in any cluster by calculating the adjusted R^2-value for each target gene and cluster based on the residuals of predicting $\mathbf{Z}_{t,2}$ from $\mathbf{Z}_{r,2}\text{diag}(\mathbf{s}_i)\mathbf{B}_i$. If the highest R^2 value falls below a user-specified threshold, the gene is considered as noise and is placed in a separate noise cluster.

3.2.4 Establishing Convergence in *ScRegClust*

The algorithm performs in a cycle of the described two steps, keeping track of the cluster membership matrix, denoted as $\mathbf{\Pi}_k$, for each iteration or cycle, k. After completing each cycle, the algorithm then compares the current membership matrix to those from the previous iterations.

If the current membership matrix is identical to the one from the immediate previous cycle, it suggests the algorithm has found a stable solution and has therefore converged, and it stops the process.

On the other hand, if the current matrix matches any other membership matrix from past iterations (not just the immediate one), it indicates that the algorithm has entered into a recurring cycle, meaning the algorithm is jumping between certain states without finding a new or more optimal solution. In this situation, the algorithm halts as well, and all clustering results up to that point are returned.

In simple words, the algorithm stops either when it finds a stable solution or when it starts repeating itself without making further progress

3.3 Providing Guidance to the Algorithm

3.3.1 Mapping the Biological Relationship Matrix Amongst Target Genes

The objective of this study is to investigate the potential presence of shared regulators among various types of cancer cells from glioblasomas. We aim to accomplish this by analyzing different datasets of glioblasomas and examining whether any regulators overlap. Additionally, we plan to use the information gathered in the first analysis to guide a second run of the algorithm, with the goal of increasing the chances of identifying shared regulators. To demonstrate this approach, imagine a scenario in which there are D datasets, each consisting of different types of cancer cells. We will use the *ScRegClust* algorithm to process each dataset individually, and use the resulting clustering as a mean of identifying biological relationships between target genes. This will generate D indicator matrices, $J_1, ..., J_D$, each corresponding to a specific dataset. Each entry in the indicator matrix, $J_d^{(i,j)}$, will be 1 if target i and j are clustered together, and otherwise 0. We will then use a specified procedure to derive D new matrices from these matrices which will be used as input for the second run of the algorithm. The new matrices are generated in the following way:

$$J_{d_{new}} = J_d \cup \bar{J}_{\geq 0.5}$$

$$\text{where} \quad \bar{J} = \frac{\sum_{d=1}^{D} J_d}{D} \quad \text{and} \quad \bar{J}^{(i,j)}_{\geq 0.5} = \begin{cases} 1 & \text{if } \bar{J}^{(i,j)} \geq 0.5 \\ 0 & \text{if } \bar{J}^{(i,j)} < 0.5 \end{cases} .$$

The Idea behind generating the new matrices in this way is to retain the clustering information from the initial run while incorporating biological relationships observed across different datasets in the second run. The aim is to incorporate the shared clustering patterns observed across all datasets as prior information.

3.3.2 Updating Weights for Coop-Lasso

In addition to incorporating prior knowledge about the biological relationship among the target genes, the weights for the coop-lasso in equation (2) are updated using the following method:

$$\mathbf{w}_i^{(k)} = \sqrt{\frac{|C_i|}{\left\|\hat{\mathbf{B}}_i^{(k,:)}\right\|_2 + \alpha \sum_{j=1}^{|C_i|} \sum_{d=1}^{D} \left\|\mathbf{B}_{j,d}^{(k,:)}\right\|_2}} \tag{5}$$

Where $\sum_{j=1}^{|C_i|} \sum_{d=1}^{D} \left\|\mathbf{B}_{j,d}^{(k,:)}\right\|_2$ represents the sum of the L2 norms of the regulator coefficients for all target genes in cluster i across all data sets from the first run and $\alpha \in [0, 1]$ and controls the strength of the prior information. This means that regulators that have the strongest linear relationship to the genes in cluster i in the previous data sets will receive less penalty. The updated weight is designed to prefer regulators that have previously been shown to regulate the target genes within each cluster. In this way, the algorithm uses prior knowledge to refine the selection of regulators. The aim of this approach is to increase the identification of the true relationship, as regulators with previous influence are given priority in the selection process.

3.3.3 Handling of Noise Genes

Since the resulting clustering from the first run was used to identify the biological relationship between genes, a question arose regarding whether genes that ended up in the "Rag Bag" should be considered to have a biological relationship or not. During the analysis of simulated data using the *ScRegClust* algorithm, it became apparent that genes belonging to defined clusters occasionally

ended up in the noise cluster. This phenomenon was also observed when analyzing real data, with a large portion of genes being assigned to the noise cluster. The only commonality among these genes in the "Rag Bag" is that they do not fit into the other defined clusters. However, their functional roles and biological properties can be highly diverse. Therefore, treating this diverse group as a coherent entity may lead to incorrect or misleading conclusions. It is important to note that noise genes were not considered to have a biological relationship among themselves.

3.4 Validation Metrics

To assess the effectiveness of the guidance in improving the algorithm, a simulation study was conducted in three exercises. Several evaluation metrics were employed, including the Rand Index, Sensitivity, Specificity, Predictive R-squared and Regulator Importance.

3.4.1 Rand Index

The Rand Index, introduced by William M. Rand in 1971 (Rand, 1971), is a measure of the similarity between two partitions of a dataset. Given a dataset of n points $S = \{p_1, ..., p_n\}$, two partitions $X = \{X_1, ..., X_s\}$ and $Y = \{Y_1, .., Y_r\}$ can be compared by counting the number of pairs of points that belong to the same cluster according to both X and Y (a), according to X but not Y (b), according to Y but not X (c), and not according to both (d). The Rand index is then calculated as:

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}}$$

where $\binom{n}{2}$ is the number of possible pairs of points in the dataset.

The Rand Index is especially useful when evaluating clustering algorithms applied to simulated data. This is due to the fact that, in simulated data, the ground truth - the actual clusters - are known. In this context, the Rand Index serves as a measure of the similarity between the clusters produced by the algorithm and the actual known clusters. The Rand Index ranges between 0 and 1; a higher value indicates closer alignment between the algorithm's clustering output and the ground truth.

The Rand Index does not take the rag bag cluster in to account (Larsson *et al*, 2023), why a modified version of the rand index was used, which was calculated as follows

$$R' = R \times \frac{\#\text{ Genes not in rag bag}}{\#\text{ Target genes}}.$$

3.4.2 Sensitivity and Specificity

Sensitively measures the proportion of true positive outcomes of an algorithm, while specificity quantifies its true negatives. The sensitivity and specificity are especially beneficial when working with simulated data. Since the ground truth is known, these metrics can therefore be accurately calculated, giving a balanced view of the algorithm performance. Furthermore, sensitivity and specificity can also be used to tune the parameters of the algorithm. For example, the parameters can be adjusted to increase sensitivity, potentially at the cost of decreased specificity, or vice versa.

In this context, sensitivity refers to the accuracy of correctly identifying true regulators. It is calculated as the ratio of true positive results to the total number of true regulators and is expressed as:

$$\text{Sensitivity} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}.$$

Specificity, on the other hand, is the ability to accurately distinguish false regulators. It is calculated as the ratio of true negative results to the total number of false regulators and is expressed as:

$$\text{Specificity} = \frac{\text{True negative}}{\text{True Negative} + \text{False positive}}$$

3.4.3 Predictive R-squared

The algorithm aims to identify the most predictive regulators for each cluster. To evaluate the quality of the clusters, *ScRegClsut* calculates the predictive R^2 for each cluster. The predictive R^2 measures how well the final linear models will predict unseen data. It is calculated on the second data split $\mathbf{Z}_2$ using the coefficients $\mathbf{B}_i$ of the regulators estimated on the first data split $\mathbf{Z}_1$. The formula for calculating the predictive R^2 is as follows (Larsson *et al*, 2023):

$$R_i^2 = \frac{\left\| \mathbf{Z}_{t,2}^{(:,C_i)} - \mathbf{Z}_{r,2}^{(:,R_i)} \text{diag}(\boldsymbol{s}_i) \mathbf{B_i} \right\|_F^2}{\sum_{j \in C_i} \left\| Z_{t,2}^{(:,j)} - \sum_{l=1}^{n_2} \mathbf{Z}_{t,2}^{(l,j)} \right\|_2^2} \tag{6}$$

This measure is used to evaluate the performance of the algorithm for different values of prior weights for the J matrix and different values of α.

3.4.4 Regulator Importance

The importance of a regulator within a cluster i, given that this regulator is donated as j, can be quantified through a calculated metric I_{ij} as specified by Larsson *et al* (2023). This metric is

defined as:

$$I_{ij} = 1 - R^2_{i,-j}/R^2_i$$

Here, $R_{i,-j}$ is computed initially by re-estimating the coefficients, represented as $\mathbf{B}_i$, for the cluster i, while intentionally excluding the regulator j. Subsequently, Equation 6 is computed, maintaining the exclusion of regulator j. Thus, I_{ij} offers a measure of the relative influence of a specific regulator within a defined cluster.

3.5 Computational Resources and Analytical Tools

For the computational component of this thesis, we utilized Tetralith (Centre, 2023), a powerful high-performance computing cluster provided by the National Supercomputer Centre (NSC) at Linköping University. Tetralith is renowned for its efficiency and reliability in handling large-scale and complex computations, making it an ideal platform for our research.

Tetralith played a crucial role in managing the high computational demand required for analyzing large scRNA-seq datasets. The system allowed us to set up specialized commands and runs on compute nodes, enabling efficient processing of data. For this study, we utilized 4 CPU cores per task and a memory of 32GB per CPU, which facilitated fast processing times and seamless handling of large datasets.

The statistical analysis of the data was conducted using the programming language R (R Core Team, 2019). R is an open-source software environment known for its extensive capabilities in statistical computing and graphics. Its flexibility allowed us to perform custom analyses, which were essential for our research. Within the R environment, we relied on the Sureat package (Butler *et al*, 2018) for processing scRNA-seq data.

3.6 Datasets

This study employed three primary datasets, each consisting of cells from glioblastomas.

The first dataset, referred to as "Neftel" comes from the study conducted by Neftel *et al* (2019). This dataset comprises 24131 cells and is accessible for download at Curated Cancer Cell Atlas.

The second dataset, derived from the study by Wang *et al* (2017), will be referred to as "Wang." The dataset can be downloaded from Curated Cancer Cell Atlas and consist 16317 cells.

The third dataset, referred to as "Leblanc," originates from LeBlanc *et al* (2022) and can be downloaded from Gene Expression Omnibus.

4 Results

4.1 Simulation Studies

The study involved three simulation exercises, each consisting of three datasets (d1, d2, and d3) with 500 target genes and 100 potential regulator genes. The simulated data closely resembled real single-cell data, and the datasets were intentionally designed to facilitate the clustering of overlapping target genes between datasets with specific regulators. This enabled us to assess the extent to which the datasets could learn from each other. All three simulation exercises showed similar results. Simulation exercise 3 is shown below whereas simulation exercise 1 and 2 are located in the appendix

Simulation Exercise 3

In the third simulation exercise, all three datasets contained five true clusters, and each cluster had 14 true regulators linked to it. Half of the regulators where shared between the datasets and the rest unique to each dataset. The clusters also overlapped between the datasets. The primary goal of this exercise was to determine whether shared regulators identified in one dataset during the first run of the *ScRegClust* algorithm could be detected in the other datasets during a second run, using information obtained from the first run. The second goal of the study was to determine if the clustering accuracy could be improved.

In Figure 1, we can observe the results from the first and second run. It is noticeable that more true and false regulators were selected in the second run. The presence of additional false regulators in the second run can be attributed to the cluster overlap. However, from the lighter colors of the falsely selected regulators in Figure 1, it seems that these false regulators have lower magnitudes. Table 1 shows an increase in average sensitivity but at the expense of a decrease in average specificity in the second run. Nonetheless, it is important to note that the chosen false regulators appear to be of lower magnitude. Furthermore, Table 1 demonstrates an increase in the Rand index from the first to the second run for all three datasets.

Figure 2 displays the results from the first and second runs when the value of α in equation 5 was reduced from 0.5 to 0.1. It is evident that in the second run more true regulators were selected, while the presence of false regulators was significantly reduced. This observation is further supported by Table 2, which reveals an increase in the average sensitivity with minimal decrease in average specificity. Additionally, there was a slight improvement in the Rand index for all the datasets.

Figure 3 displays the regulator importance of the selected regulators of the second run for each clusters of the dataset d1 for increasing values of α. The Figure also shows the average sensitivity and the specificity for increasing values of α. Whats important to notice is that true regulators appears to be first selected as α increases and then false regulator starts to get selected as α further increases. This is also supported by the graphs in the Figure, which shows that the average sensitivity increases quickly and then plateaus while the average sensitivity decreases slowly in the beginning and then decreases more rapidly as α increases. This shows that with low values of α the algorithm facilitates the identification of shared regulators that are true without selecting false regulators. For large values of α false regulators are picked up, but the false selected regulators are of low importance as reflected in the Figure.

4.1.1 Figure and Tables of Simulation Exercise 3

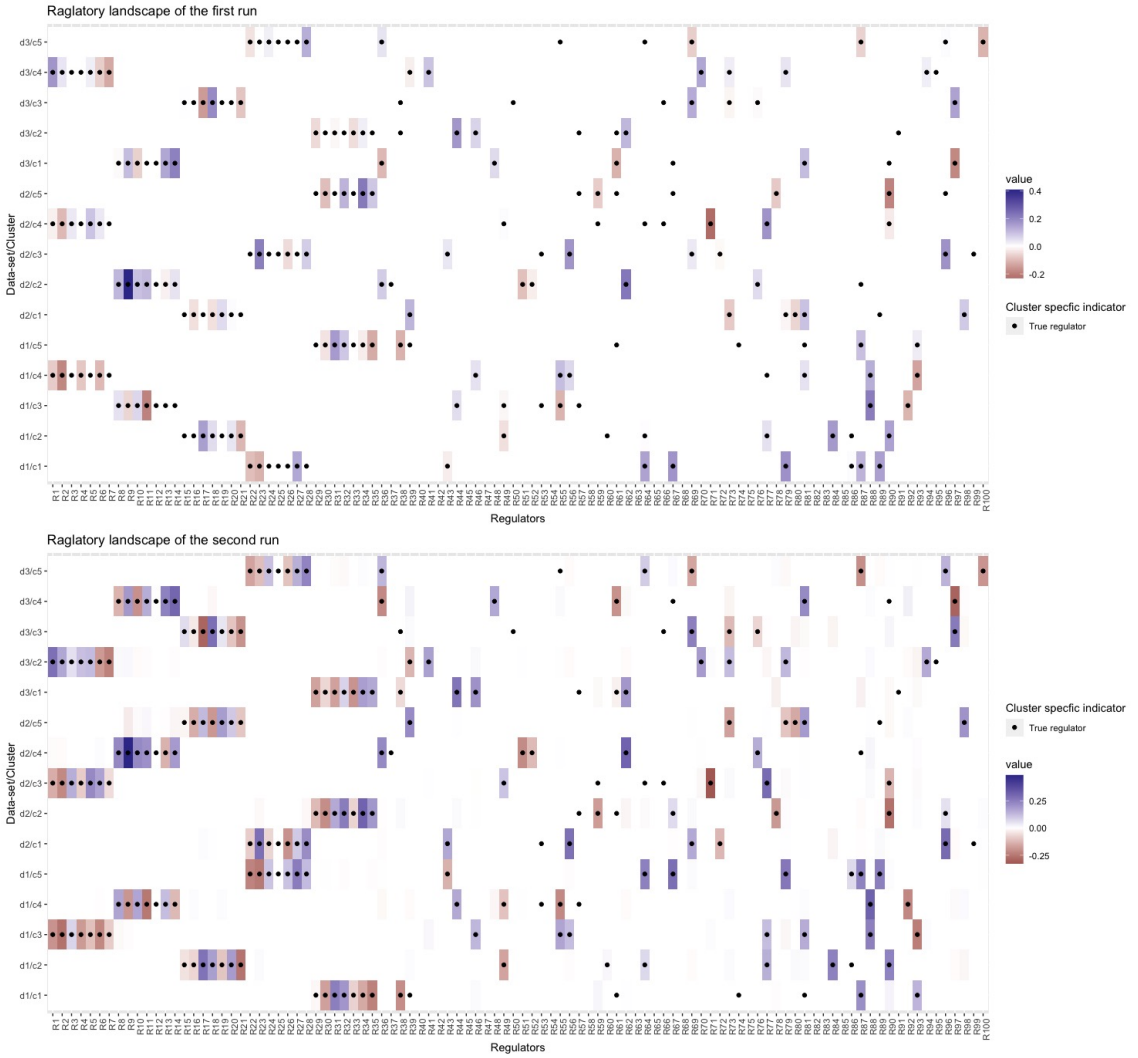

Figure 1: The active regulator and the corresponding cluster they are linked to for three datasets (d1, d2, and d3) for both the first and second run of the *ScRegClust*. The black dots indicate which regulators are true for each cluster to provide an insight into the accuracy of the ScRegClust results.

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.9916	0.6530	1	0.2024	0.1059
d2	1	0.9940	0.6301	1	0.2031	0.1076
d3	1	0.9752	0.6361	0.9997	0.1890	0.1114
d1	2	1	0.8857	0.8906	0.2413	0.0439
d2	2	1	0.8392	0.8523	0.2319	0.0403
d3	2	0.9984	0.8593	0.8715	0.2214	0.0444

Table 1: The results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average Regulator Importance values

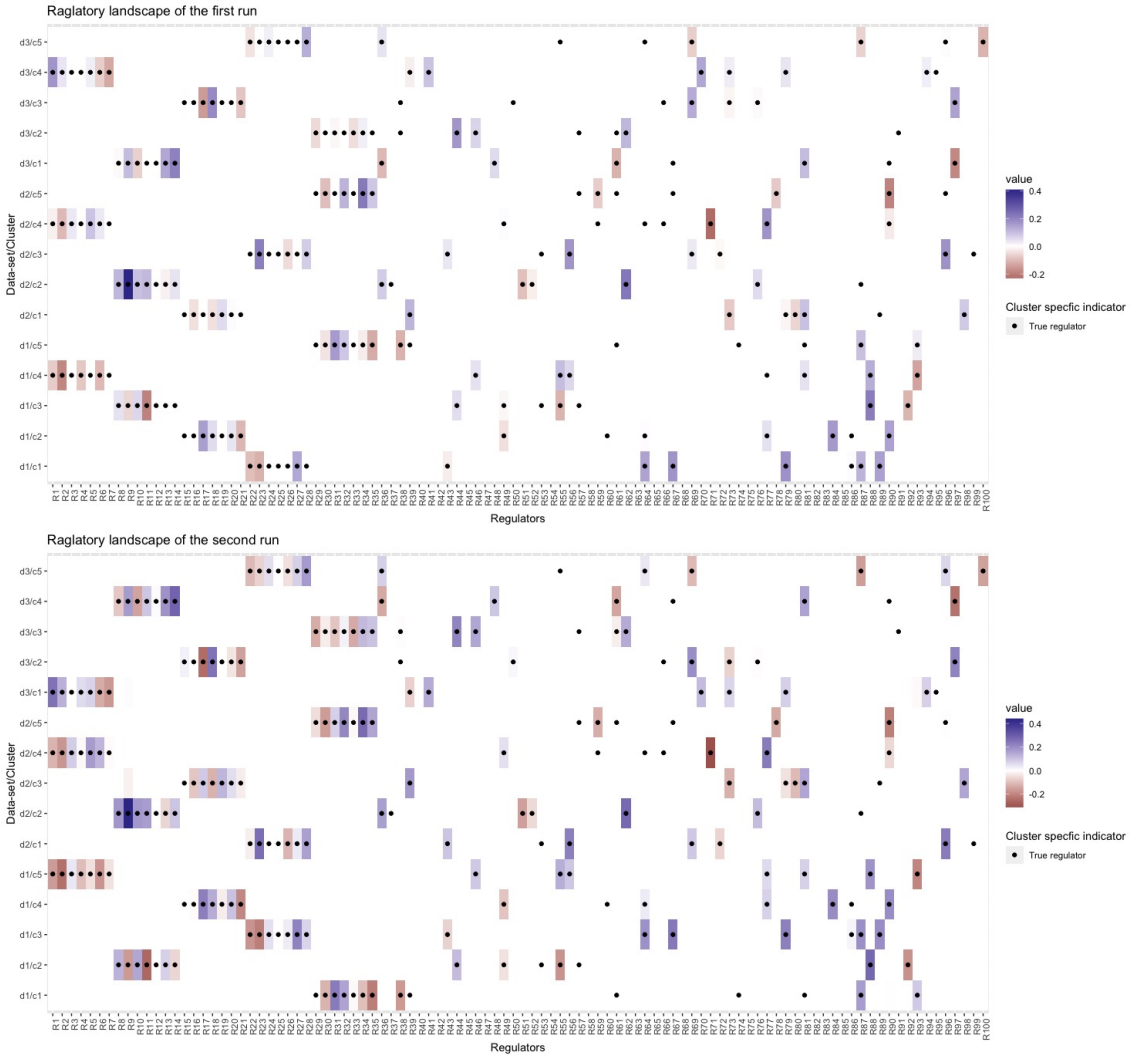

Figure 2: The active regulator and the corresponding cluster they are linked to for three datasets (d1, d2, and d3) for both the first and second run of the *ScRegClust*. The black dots indicate which regulators are true for each cluster to provide an insight into the accuracy of the ScRegClust results.

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.9916	0.6530	1	0.2024	0.1059
d2	1	0.9940	0.6301	1	0.2031	0.1076
d3	1	0.9752	0.6361	0.9997	0.1890	0.1114
d1	2	0.9972	0.8415	0.9976	0.2215	0.0815
d2	2	1	0.7892	0.9971	0.2201	0.0881
d3	2	0.9984	0.8268	0.9938	0.2089	0.0843

Table 2: The results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average Regulator Importance values

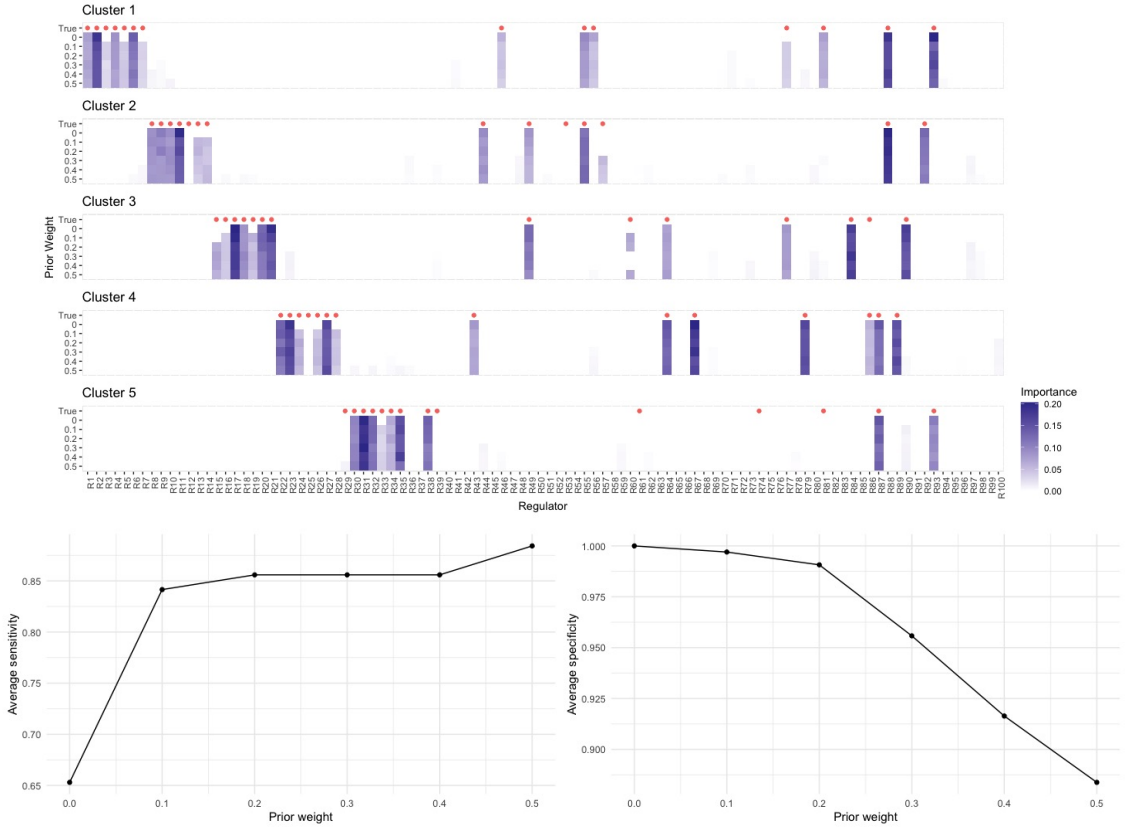

Figure 3: The impact of the prior weight (α) when selecting regulators and their relative importance for each cluster in dataset d1. The top image illustrate the active regulators and their relative importance for each cluster for prior weight ranging from 0 to 0.5.The bottom left graph shows the average sensitivity as prior weight increases. The bottom right graph shows the average specificity as the prior weight increases.

4.2 Transition to real data

4.2.1 First Run of *ScRegClust*

In the first runs of *ScRegClust*, each dataset underwent a grid search, which included various values of the penalization parameter λ and different quantities of clusters, K. The choice of λ was selected based on two factors: First, the predictive R^2 per cluster and second, the regulator importance. The goal was to find a balanced interaction between these two metrics. Ideally, we wanted a high predictive R^2 per cluster, alongside a regulator importance that was neither too high or too low (Larsson *et al*, 2023). Whereas a high value of the regulator importance suggested that too few regulators were selected, a low value indicated that too many regulators were selected. For the choice of K we aimed for a high silhouette score and simultaneously a high average predictive R^2 per cluster.

Figure 4 illustrates the regulator importance and the predictive R^2 per cluster for a range of λ

values from the initial run of *ScRegClust* on the Leblanc dataset. As observed from the Figure 4, the predictive R^2 begins to diminish for penalization values greater than or equal to 0.2. Hence, a penalization value of either 0.14 or 0.16 would be preferable.

Figure 5 shows the average silhouette score and the average predicted R^2 for different numbers of clusters. For $K = 6$ the Figure displays a high average predictive R^2 per cluster and an average silhouette score above 0.9. Figure 6 presents the silhouette score. According to the Figure, initializing with 6 clusters, this resulted in a 4 cluster configuration, whereas starting with either 4 or 5 clusters lead to a 3 cluster outcome. Despite the higher silhouette scores for $K = 3, 4$, and 5, we opted for $K = 6$, which gave a 4 cluster solution, as we prioritized a higher number of clusters over a minor decrease in the silhouette score.

The selection process for the penalization value λ and the number of clusters K was applied similarly across the remaining datasets.

Figure 7 illustrates the active regulators associated with each cluster for all datasets. Only a few active regulators overlapped between datasets. Table 3 displays the average silhouette and the average predicate R^2 per cluster.

Figure 8 provides an overview of gene overlap in the datasets, excluding the genes that fell in to the "Rag bag". Displaying a low overlap between clustering. The interconnectedness of the clusters across different datasets is systematically outlined in Tables 4, 5 and 6. The tables present the size of each cluster, denoted by the total number of genes it contains. Furthermore, they illustrate the overlap of genes between these clusters and those in other datasets.

Table 7 shows the Adjusted Rand Index (ARI) between clustering when running *ScRegClust* only with genes existing in all three dataset. There were 875 genes found to be shared across the three datasets. The results show low similarites between the clusterings of these shared genes among the datasets.

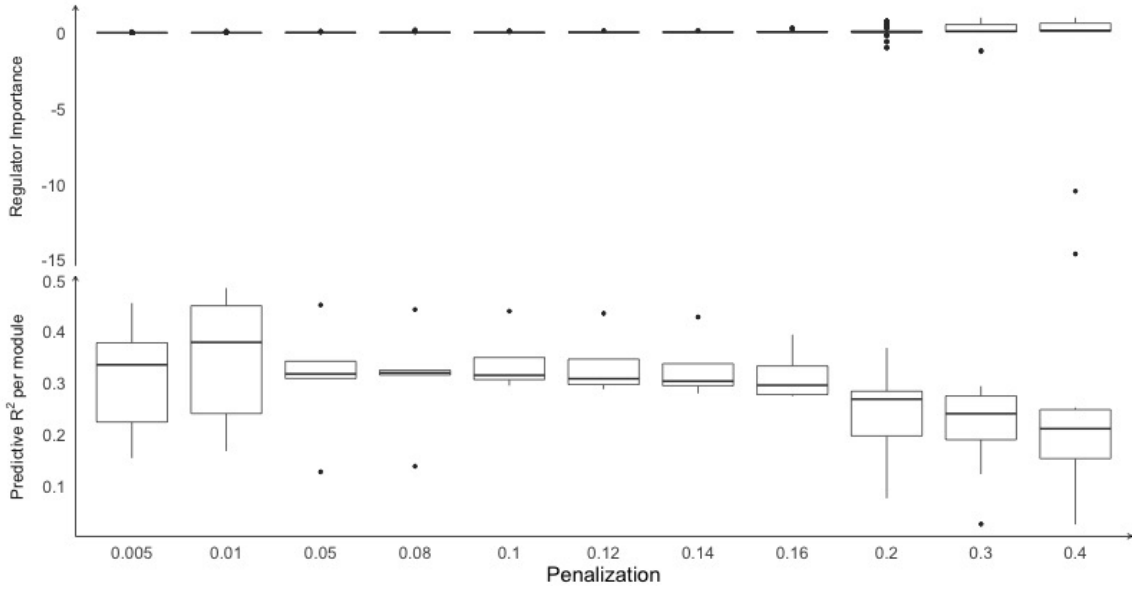

Figure 4: The predictive R^2 per cluster and the regulator importance for various values of λ

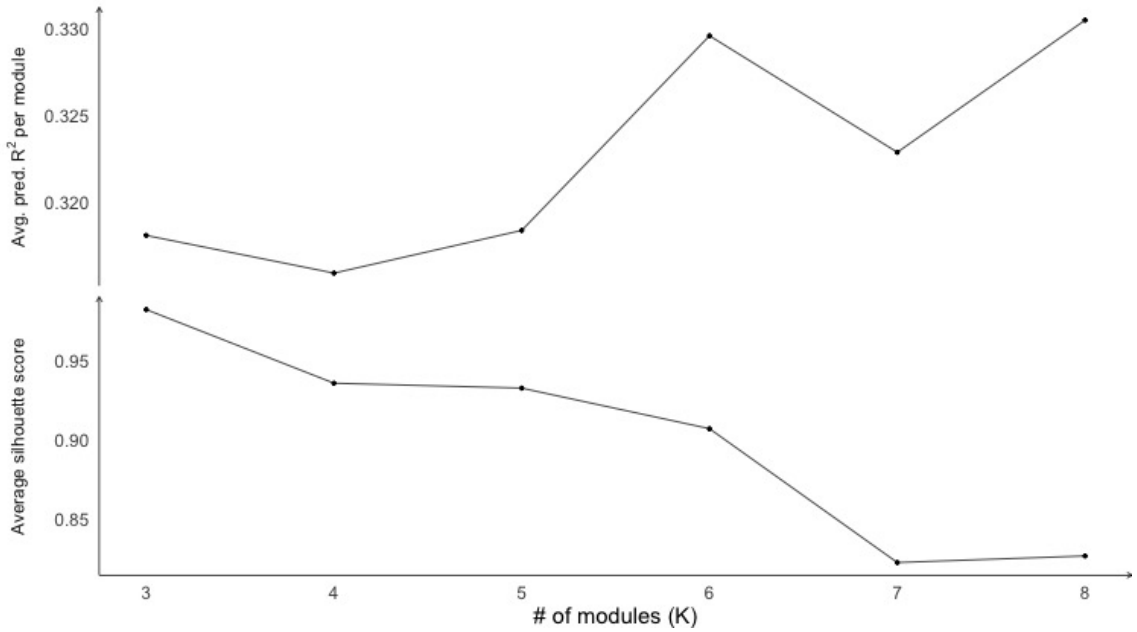

Figure 5: The average predictive R^2 per cluster and the average silhouette score for different numbers of clusters K

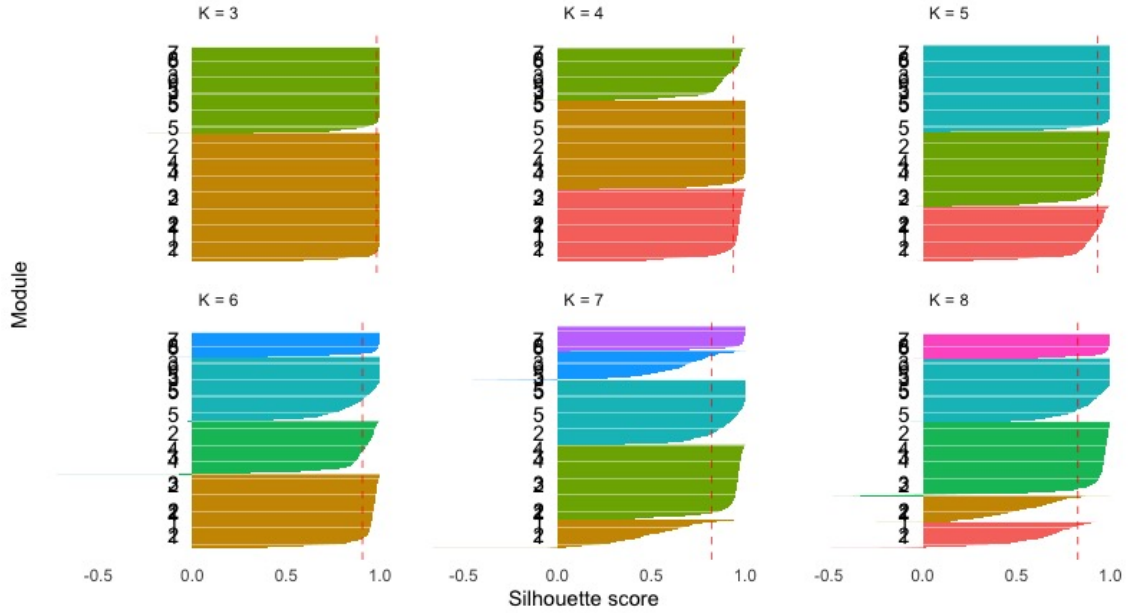

Figure 6: The silhouette score for different values of cluster K

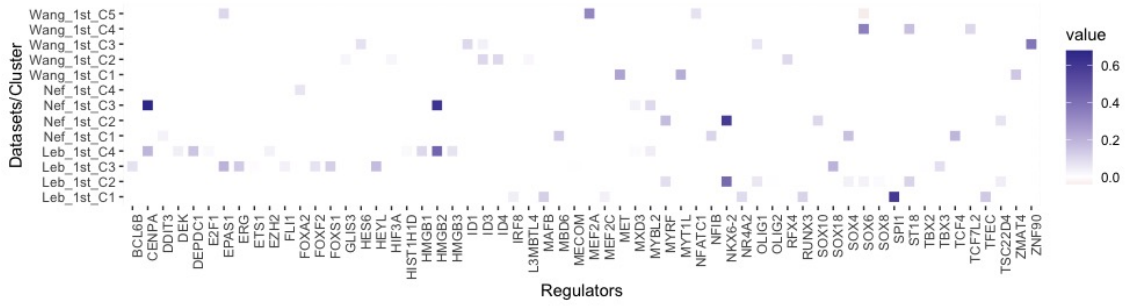

Figure 7: The active regulator and the corresponding cluster they are linked to for the datasets Leblanc, Neftel and Wang for the first run of the *ScRegClust*

Dataset	Avg. Silhouette	Avg. predictive R^2
Leblanc	0.9031	0.3350
Neftel	0.9656	0.2650
Wang	0.8203	0.1921

Table 3: Average silhouette score and average predictive R^2 per cluster from the first run

Neftel/Leblanc	C1	C2	C3	C4	Cluster Size
C1	1	40	0	6	142
C2	3	155	6	1	279
C3	0	0	0	66	71
C4	79	0	1	0	103
Cluster Size	402	363	281	133	

Table 4: Comparison of cluster overlap within clusters between the final clusterings obtained from the first run of *ScReClust* using the Leblanc and Neftel datasets

Wang/Leblanc	C1	C2	C3	C4	Cluster Size
C1	0	15	0	0	117
C2	8	18	17	0	234
C3	6	14	1	6	119
C4	2	28	4	0	106
C5	0	3	3	0	93
Cluster Size	402	363	281	133	

Table 5: Comparison of cluster overlap within clusters between the final clusterings obtained from the first run of *ScReClust* using the Leblanc and Wang datasets

Wang/Neftel	C1	C2	C3	C4	Cluster Size
C1	13	3	0	0	117
C2	1	13	0	0	234
C3	4	2	3	0	119
C4	0	25	0	0	106
C5	1	4	0	0	93
Cluster Size	142	279	71	103	

Table 6: Comparison of cluster overlap within clusters between the final clusterings obtained from the first run of *ScReClust* using the Neftel and Wang datasets

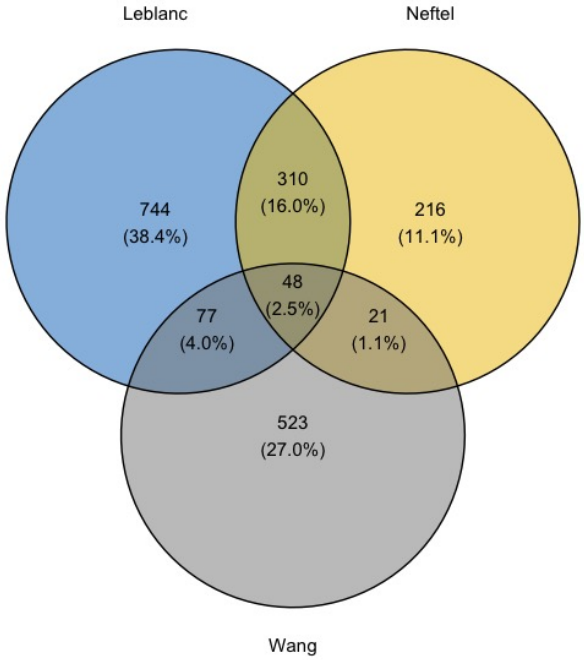

Figure 8: Gene Overlap of the Final Clustering from the First Run of *ScRegClust* Using the Leblanc, Neftal, and Wang Datasets

Clusterings	ARI
Leblanc & Neftel	0.350
Leblanc & Wang	0.035
Neftel & Wang	0.065

Table 7: The Adjusted Rand Index (ARI) between clusterings

4.2.2 Second Run of *ScRegClust*

Figure 9 illustrates the active regulators and the corresponding cluster they are linked to for the second run of *ScRegClust*. In the second run, new Regulators are identified, and the Wang datasets results in 4 cluster outcome instead of 5 cluster outcome.

Table 8 presents the average silhouette score and the average predictive R^2 per cluster for the second run. The average silhouette score remained consistent for both the Leblanc and Wang datasets. However, for the Neftel dataset, there was a decrease in the average silhouette score. This observation is attributed to the addition of approximately 100 new genes to the clusters. It is anticipated that introducing additional genes may lead to a decrease in the silhouette score. On the other hand, the predictive R^2 showed a small increase for the Leblanc and Neftel datasets, while Wang remained consistent with the first run.

Figure 10 shows the gene overlap between datasets, excluding the genes that fell in to the Rag bag, demonstrating that the gene overlap has increased from the first to the second run of *ScRegClust*.

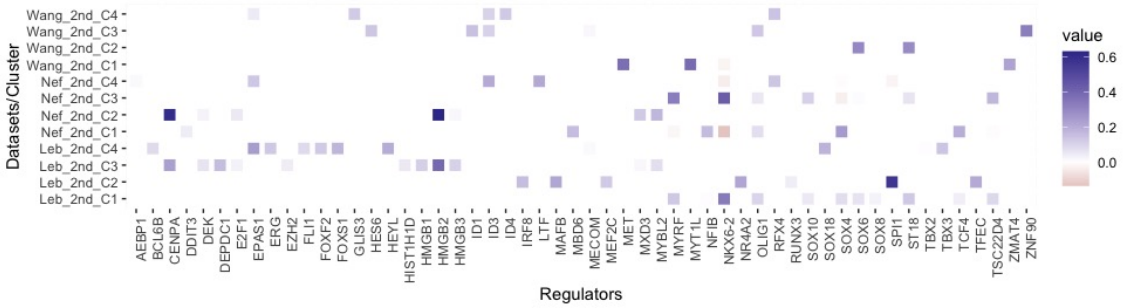

Figure 9: The active regulator and the corresponding cluster they are linked to for the datasets Leblanc, Neftel and Wang for the second run of the *ScRegClust*

Dataset	Avg. Silhouette	Avg. predictive R^2
Leblanc	0.9017	0.3404
Neftel	0.8533	0.2891
Wang	0.8203	0.1940

Table 8: Average silhouette score and average predictive R^2 per cluster from the first run

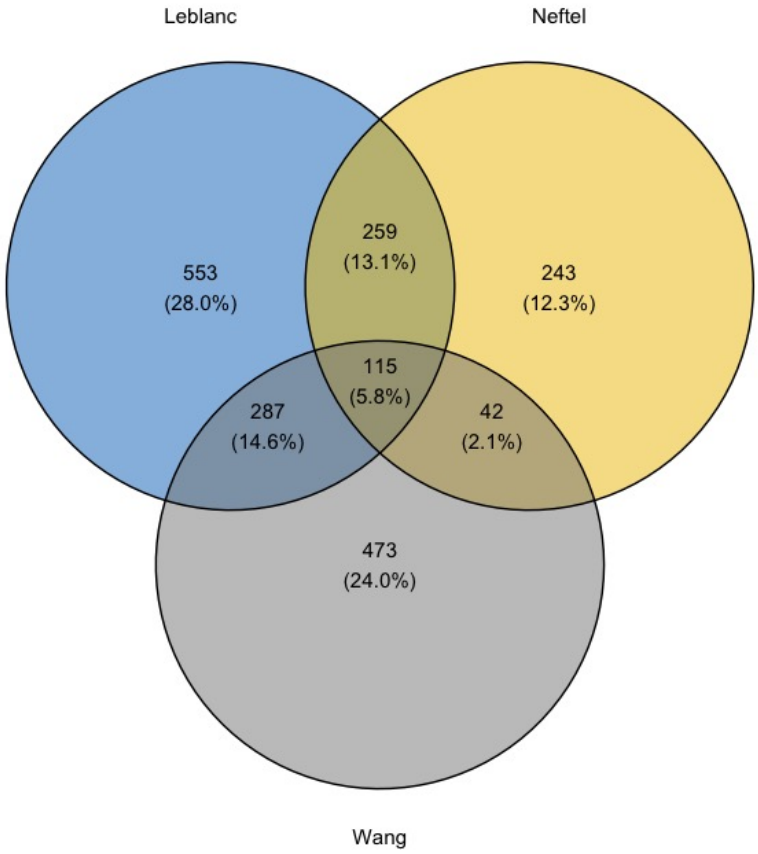

Figure 10: Gene Overlap of the Final Clustering from the Second Run of *ScRegClust* Using the Leblanc, Neftal, and Wang Datasets

5 Discussion

The approach applied in this study demonstrated promising outcomes in a simulated environment where target genes predominantly shared regulators across datasets. Also, the true clusters for each dataset overlapped substantially. However, this simplistic relationship between datasets was not seen in real-world data. In this environment the overlap of genes across datasets was limited, and diminished further after clustering, leading to a low interconnectivity among the clusterings. Thus, comparison with real data revealed that the performance of our results fell short of expectations. While the initial findings from the simulation exercises demonstrated promising results, the transition to real-world application posed significant challenges. Due to this we limited ourselves to three datasets (Leblanc, Neftel, Wang) all from nervous system cancers glioblastoma.

Our method was premised on the assumption that target genes that shared regulators where predominantly clustered together. Yet, this pattern was not seen in real-world data. The method

was intended to guide *ScRegClust* to choose regulators that previously regulated any of the genes in the clusters from the first run. A significant challenge arose when target genes from a single cluster in one of the dataset were dispersed across multiple clusters in another. This dispersion tended to confuse the algorithm rather than guide it, with regulators appearing in multiple clusters in the second run.

A notable observation is that in the first run only 48 genes were commonly identified in the defined clusterings across all three datasets. This implies that the majority of the shared genes ended up in the rag bag. The largest overlap occurred between the Leblanc and Neftel, comprising 358 genes. This was followed by a 125 gene overlap between Leblanc and Wang, and lastly, Neftel and Wang shared an overlap of 69 genes.

Further insight into the data from the first run highlighted an interesting overlap between cluster 4 of Leblanc and cluster 3 of Neftel. A majority of the genes from Neftel´s were also present in Leblanc´s cluster 4, suggesting a significant similarity between these clusters. This is also supported by the fact that cluster 4 of Leblanc and cluster 3 of Neftel share regulators. However, this clear overlap is not observed in the other clusters. For instance, the genes shared between Leblace's cluster 2 and Wang are scattered between all five clusters of Wang, indicating a more complex relationship.

Cluster 3 from Neftal's dataset shared all of its regulators with Cluster 4 in the Leblanc dataset, namely CEPNA, HMGB2, MXD3, and MYBL2. Interestingly, additional regulators – DEK, DEPDC, E2F1, HMGB2, and HMGB3 – are also found as regulators in Leblanc's Cluster 4. Among these, DEK, E2F1, and HMGB3 are present in the Neftel dataset, and were identified in the second run. While the Average predictive R^2 for Neftel increased in the second run the predictive R^2 for cluster 2 (corresponding to cluster 3 in the first run) of Neftel contributed the most to the increase. This indicates the efficacy of the guidance in scenarios where gene overlap is clearly identified and clusters already share certain regulators.

To deepen our understanding of the shared genes across the datasets, we isolated these common genes and ran *ScRegClust* specifically on them. This meant that all datasets consisted of the same genes, thereby enabling a more direct comparison of the clustering. The comparison was done by using the Adjusted Rand Index (ARI), a measurement that quantifies the similarity between clustering. The ARI Score for the clustering comparison of Leblanc and Neftel datasets was 0.35. This suggests a moderate degree of similarity between the two datasets. Since the ARI score is

above 0, this indicates that they are more similar then expected by chance. However the score is still far from 1, suggesting that the clustering are far from identical. On the other hand, the ARI score for the Leblanc and Wang was 0.035 and the ARI score for Neftel and Leblanc was 0.065. Both these scores are close to zero, suggesting that the similarity between these clustering are only by chance. Hence, no patterns can be seen from these clusterings.

The consideration of Transcription Factor (TF) families was an attempt to mitigate cases where target genes were regulated by regulators not present in other datasets. Given that members of TF families share a common functional domain, we hypothesized that if a gene was regulated by a regulator in one dataset but this regulator was absent in another, we would investigate the TF family to identify any members present in this dataset. We would then reduce the penalty for these regulators genes. Unfortunately, this strategy also proved ineffective. This was a naïve simplification, as TFs from the same families were found to regulate target genes in different clusters within a dataset. We learned that it cannot be assumed that TFs from the same family regulate the same genes, even though the likelihood may be higher.

A potential approach for future research could be to specifically target clusters between datasets that demonstrate a clear overlap of genes and that share certain regulators. Thereafter, guiding *ScRegClust* exclusively on these genes, given our model showed improvements in such scenarios. This approach could also help mitigate confusion encountered when target genes from one cluster in a dataset are distributed across multiple clusters in another dataset.

Given the lack of gene overlap across datasets, a strategy could be to seek out genes that show high correlation between datasets. If a cluster in one dataset is regulated by certain genes that are not found in another, we could identify genes that correlate with the absent ones. Subsequently, by lowering the penalty for these correlated genes in the second run could potentially improve performance. While this tactic was tested with the aim to improve clustering, it was shown that genes with high correlation were already co-clustered. However this approach has not been explored in the context of regulator selection, which could be an interesting idea for further research.

Another area for future investigation involves the concept of "proxy-genes". In situations where one gene is difficult to measure or absent, proxy genes can serve as substitutes. Proxy genes are often co-expressed and suggesting a close relationship. By exploring these proxy genes, we can potentially enhance data sharing between datasets, particularly in cases where gene overlap is

limited. Leveraging proxy genes has the potential to bridge the gaps in datasets and facilitate a deeper understanding of the biological processes.

In conclusion our method based on regulatory driven clustering of single-cell data demonstrated well in a simulated environment. However, in real-world data studying brain tumors the performance fell short of expectation. Further studies targeting proxy gens or genes that show high correlation between datasets are required.

References

Arthur D, Vassilvitskii S (2007) K-means++: the advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms* 3.2.1

Barbuti R, Gori R, Milazzo P, Nasti L (2020) A survey of gene regulatory networks modelling methods: from differential equations, to Boolean and qualitative bioinspired models. *Journal of Membrane Computing* **58**: 207–226 2.2

Boyd S, Parikh N, Chu E, Peleato B, Eckstein J (2011) Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers. *Foundations and Trends in Machine Learning* **3**: 1–122 3.2.2, 3.2.2

Bray F, Laversanne M, Weiderpass E, Soerjomataram I (2021) The ever-increasing importance of cancer as a leading cause of premature death worldwide. *Cancer* 1

Butler A, Hoffman P, Smibert P, Papalexi E, Satija R (2018) Integrating single-cell transcriptomic data across different conditions, technologies, and species. *Nature Biotechnology* 3.5

Centre NS (2023) Tetralith. National Supercomputer Centre in Sweden, available from: https://www.nsc.liu.se/systems/tetralith/ 3.5

Chiquet J, Grandvalet Y, Charbonnier C (2012) Sparsity with sign-coherent groups of variables via the cooperative-Lasso. *The Annals of Applied Statistics* **6** 2.3.3, 3.2.2, 3.2.2, 3.2.2

Debela D, Muzazu S, Heraro K, Ndalama M, Mesele B, Haile D, SK. K, Manyazewal T (2021) New approaches and procedures for cancer treatment: Current perspectives. *SAGE Journals* 1

Larsson I, Held F, Popova G, Koc A, Jornsten R, Nelander S (2023) Reconstructing the regulatory programs underlying the phenotypic plasticity of neural cancers. *bioRxiv preprint* 1, 2.2, 2.3, 3.1, 3.2, 3.2, 3.2, 3.2.1, 3.2.1, 3.2.2, 3.2.2, 3.2.3, 3.2.3, 3.4.1, 3.4.3, 3.4.4, 4.2.1

LeBlanc VG, Trinh DL, Aslanpour S, Hughes M, Livingstone D, Jin D, Ahn BY, Blough MD, Cairncross JG, Chan JA, Kelly JJP, Marra MA (2022) Single-cell landscapes of primary glioblastomas and matched explants and cell lines show variable retention of inter- and intratumor heterogeneity. *Cancer Cell* **40**: 379–392 3.6

Meinshausen N (2013) Sign-constrained least squares estimation for high-dimensional regression. *Electronic Journal of Statistics* **7** 3.2.3

Neftel C, Laffy J, Filbin MG, Hara T, Shore ME, Rahme GJ, Richman AR, Silverbush D, Shaw ML, Hebert CM, Dewitt J, Gritsch S, Perez EM, Gonzalez Castro LN, Lan X, Druck N, Rodman

C, Dionne D, Kaplan A, Bertalan MS, *et al* (2019) An Integrative Model of Cellular States, Plasticity, and Genetics for Glioblastoma. *Cell* **178**: 835–849.e21 3.6

Nguyen DK, Ho TB (2017) Accelerated anti-lopsided algorithm for nonnegative least squares. *International Journal of Data Science and Analytics* **3**: 23–34 3.2.3

R Core Team (2019) *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria 3.5

Rand WM (1971) Objective Criteria for the Evaluation of Clustering Methods. *Journal of the American Statistical Association* **66**: 846–850 3.4.1

Sarkar S, Horn G, Moulton K, Oza A, Byler S, Kokolus S, Longacre M (2013) Cancer development, progression, and therapy: an epigenetic overview. *International Journal of Molecular Sciences* 1

Sung H, Ferlay J, Siegel R, Laversanne M, Soerjomataram I. Jemal A, Bray F (2021) Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *A Cancer Journal for Clinicians* 1

Tibshirani R (1996) Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society Series B Methodological* **58**: 267–288 2.3.1

Wang Q, Hu B, Hu X, Kim H, Squatrito M, Scarpace L, deCarvalho AC, Lyu S, Li P, Li Y, Barthel F, Cho HJ, Lin YH, Satani N, Martinez-Ledesma E, Zheng S, Chang E, é CG, Olar A, Lan ZD, *et al* (2017) Tumor Evolution of Glioma-Intrinsic Gene Expression Subtypes Associates with Immunological Changes in the Microenvironment. *Cancer Cell* **32**: 42–56 3.6

Xu Z, Figueiredo MAT, Yuan X, Studer C, Goldstein T (2017) Adaptive Relaxed ADMM: Convergence Theory and Practical Implementation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 3.2.2

Yuan M, Lin Y (2006) Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society Series B Statistical Methodology* **68**: 49–67 2.3.2, 3.2.2

Zou H (2006) The Adaptive Lasso and Its Oracle Properties. *Journal of the American Statistical Association* **101**: 1418–1429 3.2.2

A Appendix

A.1 Simulation Exercise 1 and 2

Simulation Exercise 1

In the first simulation exercises, all three datasets contained five true clusters, and each cluster had 14 true regulators linked to it. In the first simulation exercise, the datasets shared the same set of true regulators. The primary goal of this exercise was to determine whether shared regulators identified in one dataset during the first run of the scregclust algorithm could be detected in the other datasets during a second run, using information obtained from the first run. The second goal of the study was to determine if the clustering accuracy could be improved.

The first study tested four different combinations of datasets, where the signal-to-noise ratio (S/N) differed, as follows:

1. d1, d1 and d3 with S/N = 0.8
2. d1 and d2 with S/N=0.8 and d3 with S/N = 0.6
3. d1 with S/N=0.8 and d2 and d3 with S/N = 0.6
4. d1, d1 and d3 with S/N = 0.6

As the signal-to-noise ratio decreases, the algorithm faces greater difficulty in identifying true regulators, which leads to a decrease in clustering accuracy. A higher signal-to-noise ratio results in less noise relative to the signal and therefore fit the model better. Contrary, A low signal-to-noise ratio introduce more noise and can make it more difficult to recognise patterns and therefor lower the accuracy of the model.

Figure 11 displays the selected regulators and their corresponding strengths for the clusters in each dataset from the first and second run of scregclust. All three datasets (d1, d2, and d3) had a signal-to-noise ratio of 0.8. The figure shows that more true regulators were found in the second round, which is also confirmed by Table 9. In Table 9, we observe that the average sensitivity for the first run is between 0.7611 and 0.9050, while for the second run, it ranges between 0.9920 and 0.9980. Table 9 also shows that the specificity for all three datasets in the first and second rounds was 1. The high sensitivity and specificity score in the second run, indicates that with the help of information from the first runs, the algorithm was able to identify almost all true regulators while avoiding selecting false regulators. Table 9 also indicates an increase in the average predictive

R^2 per cluster and a decrease in the average regulator importance, which further supports our assertion of an improvement in the second round. The Rand index, as shown in Table 9, is high for both the first and second rounds.

Figure 12 is similar to Figure 1, but the signal-to-noise ratio for dataset d3 lowered from 0.8 to 0.6, resulting in the selection of fewer regulators for each cluster in d3 compared to those in Figure 11. However, in the second run, almost all true regulators were identified. Table 10 shows a lower Rand index of 0.8520 for d3 compared to 0.9972 in Table 1 for the first run, which is due to the lower signal-to-noise ratio. Nevertheless, the Rand index increased to 0.9972 in the second run, thanks to the information obtained from the first run. Table 10 shows improvement in Sensitivity especially for d3, where the sensitivity was 0.4902 in the first run and 0,9746 in the second run. The specificity remains 1 in the second run for all three datasets.

In Figure 13, the signal-to-noise ratio for d2 was also set to 0.6, resulting in fewer regulators being selected and only four of the five clusters being found. From Table 11, we can see that changing the signal-to-noise ratio to 0.6 resulted in a decrease in the average sensitivity for d2 from 0.8073 to 0.3672 in the first run. However, in the second run, the average sensitivity improved to 0.9807.

Figure 14 shows the results when the signal-to-noise ratio was set to 0.6 for all three datasets. In the first run, only four out of five clusters were found for d1 and d2, and less than half of the true regulators were identified. However, in the second run, all five clusters were correctly identified for all three datasets, and more true regulators were found. Table 12 reveals a clear improvement in the average sensitivity between the first and second run for all three datasets, while the average specificity remained at 1. These results demonstrate that the algorithm is capable of learning from previous runs and integrating information to improve its performance in subsequent runs.

Simulation Exercise 2

For the second simulation study, we used a similar clustering structure and the same combinations of datasets but with only half of the regulators shared between them and the rest unique to each dataset. The aim of the second study was to see if the unique regulators would show up in the second run since clusters between data set only share half of the true regulators and not the other half. The question was if lowering the penalty for false regulators would make the coop lasso select false regulators for the clusters in the second run. Figure 15 shows the selected regulators for both the first and second runs when all datasets had a signal-to-noise ratio of 0.8. We can see

that shared regulators not found in one of the datasets in the first run, but found by the others, were successfully identified in the second run. This is further confirmed by the increase in average sensitivity from the first to the second run, as shown in Table 13. However, we observed a decrease in average specificity from 1 to 0.9944 for dataset d2, indicating that three false regulators were introduced, but these appear to be of low importance.

Figure 16 shows the result of setting the signal-to-noise ratio to 0.6 for d3, where only four out of five clusters were identified. In the second run, all five clusters were found and all regulators shared with the other dataset were also identified. However, only two of the unique regulators were found, suggesting that the information gained from the first run helped in selecting regulators that were shared between datasets. Table 14 shows that the average sensitivity went up for all three dataset from the first to the second run and especially for d3 where the sensitivity went from 0.4843 to 0.7947. The average sensitivity decreased for both d2 and d3, but of small magnitude.

Figure 17 illustrates the results obtained when the signal-to-noise ratio was reduced to 0.6 for datasets d3 and d2. Lowering the signal-to-noise ratio posed challenges for the algorithm in identifying all the clusters, with d2 only capturing 3 out of the five clusters. Table 15 confirms this difficulty, showing a low rand index of 0.5195 for d2, attributed to the missing clusters, while d3 achieved a rand index of 0.7572. The average sensitivity was initially low for both d3 and d2. However, the high rand index and average sensitivity for d1 provided valuable guidance for the algorithm in the second run. In the second run, all three datasets demonstrated an increase in the rand index, particularly noticeable for d3 and d2. Furthermore, there was an improvement in average sensitivity across all datasets, with d3 and d2 exhibiting substantial gains. The average specificity experienced a minimal decrease. Upon closer examination of d2/c4, d2/c5, and d3/c1 in the second run, which corresponded to d2/c4, d2/c5, and d3/c5 in the first run, it becomes evident that the second run primarily identified shared regulators while the detection of unique regulators was relatively limited.

Figure 18 presents the results obtained when the signal-to-noise ratio was set to 0.6 for all three datasets in the first and second runs. From Figure 18, it is evident that in the first run, d1 could only capture 3 out of the five clusters. However, in the second run, all five clusters were successfully identified for d2 and d3, while d1 could only identify four clusters. Notably, cluster 2 of d1 seemed to be split, with its target genes grouped into clusters 1 and 3. Consequently, shared regulators with d2 and d3 that belonged to cluster one appeared in clusters 1 and 3. Table 16 displays

an increase in both the Rand index and average sensitivity for all three datasets. The average sensitivity experienced a minimal decrease for d2 and d3, but a slightly more noticeable decrease for d1 due to the misclustering of target genes from cluster 2, which ended up in clusters 1 and 3.

A.1.1 Figure and Tables of Simulation Exercise 1

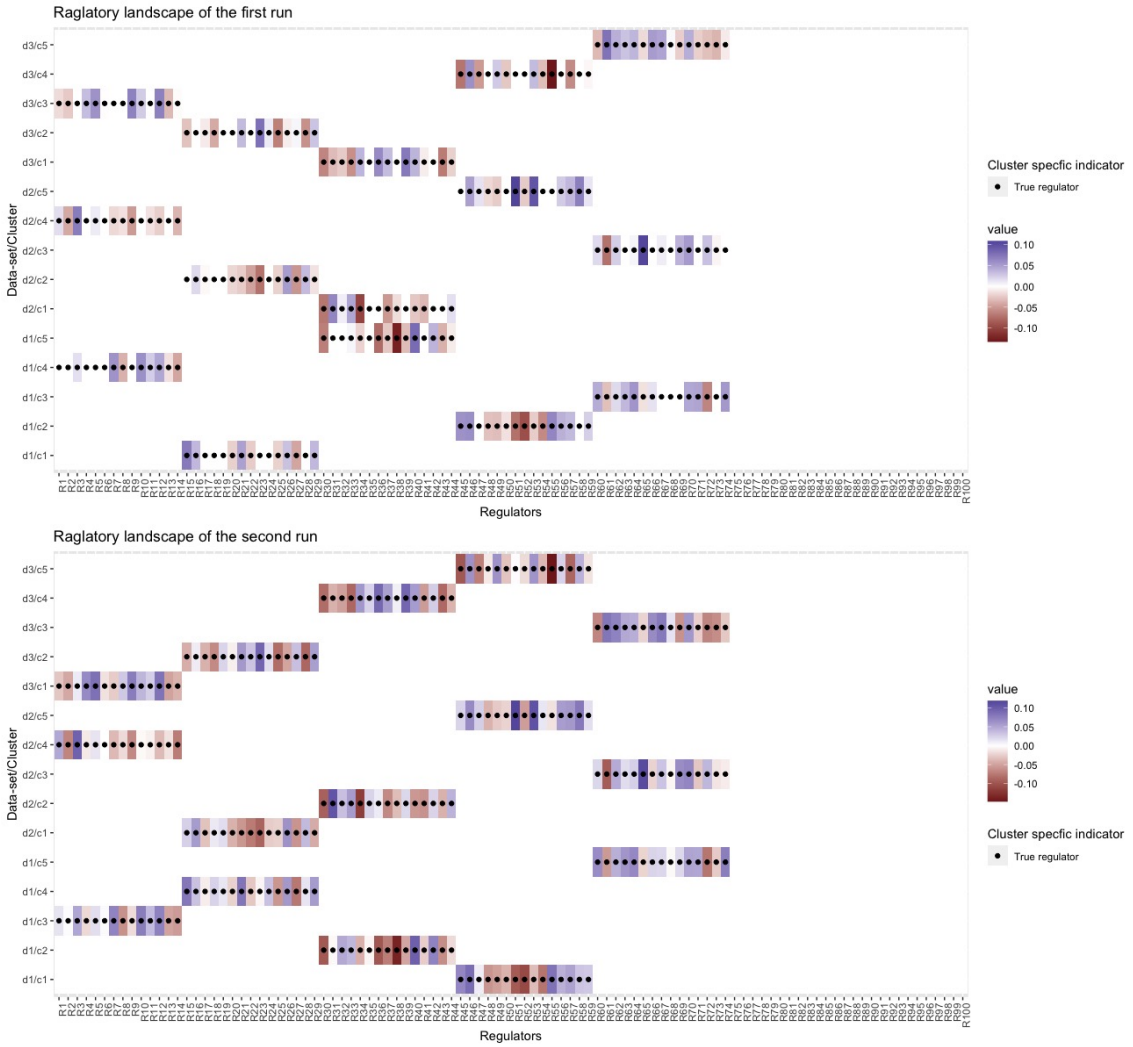

Figure 11: The figure illustrates the active regulator and the corresponding cluster they are linked to for three datasets (d1, d2, and d3) for both the first and second run of the *ScRegClust*. The black dots indicate which regulators are true for each cluster to provide an insight into the accuracy of the ScRegClust results.

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.9833	0.7611	1	0.3124	0.0853
d2	1	0.9944	0.8073	1	0.3121	0.0808
d3	1	0.9972	0.9040	1	0.3310	0.0720
d1	2	0.9972	0.9980	1	0.3440	0.0658
d2	2	0.9944	0.9820	1	0.3286	0.0665
d3	2	0.9972	0.9980	1	0.3394	0.0657

Table 9: The table showcases the results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average regulator importance values

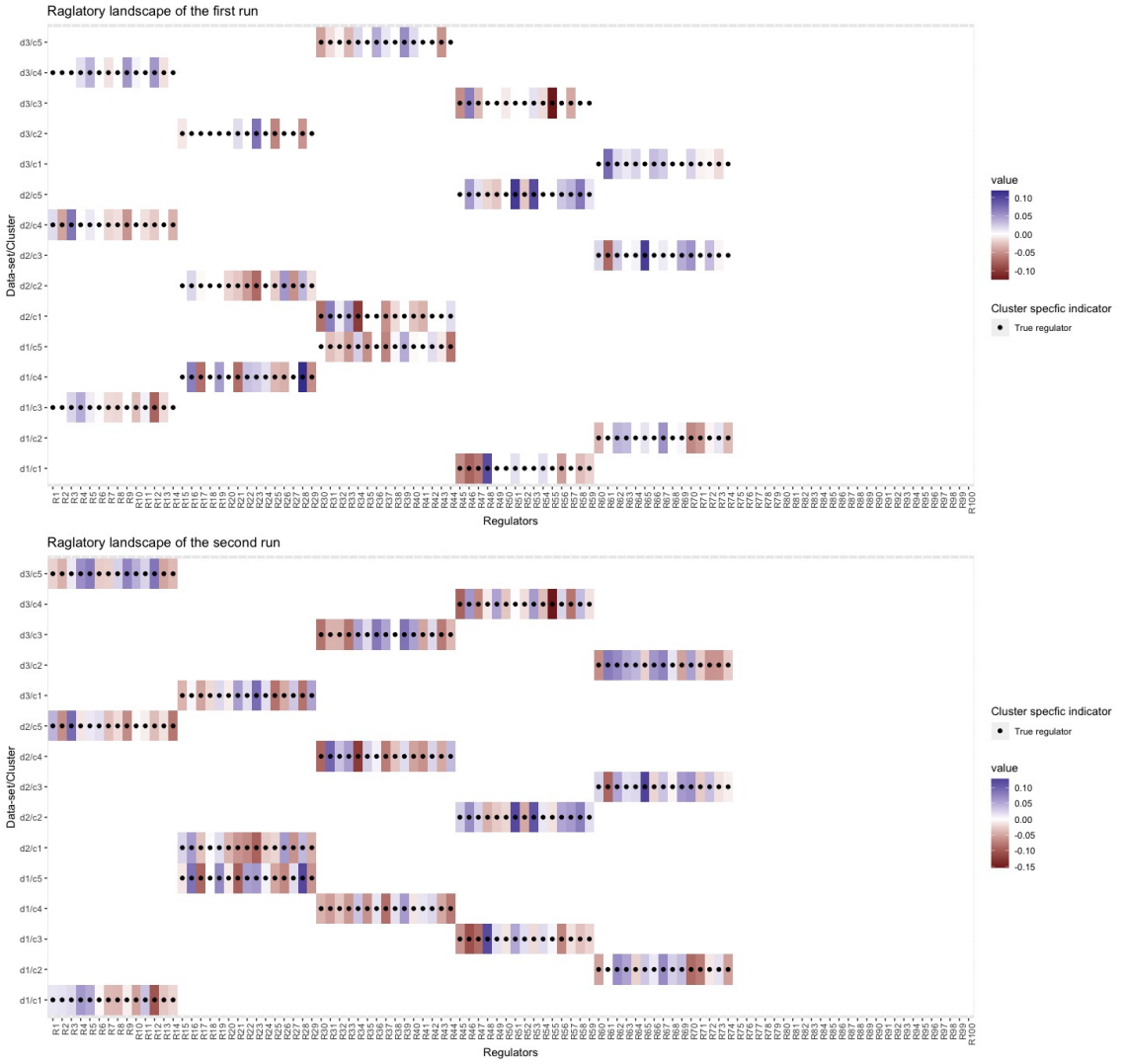

Figure 12: The figure illustrates the active regulator and the corresponding cluster they are linked to for three datasets (d1, d2, and d3) for both the first and second run of the *ScRegClust*. The black dots indicate which regulators are true for each cluster to provide an insight into the accuracy of the ScRegClust results.

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.9833	0.7611	1	0.3124	0.0853
d2	1	0.9944	0.8073	1	0.3121	0.0808
d3	1	0.8520	0.4902	1	0.1827	0.1185
d1	2	0.9972	0.9846	1	0.3440	0.0658
d2	2	0.9944	0.98980	1	0.3286	0.0665
d3	2	0.9972	0.9746	1	0.2316	0.0661

Table 10: The table showcases the results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average regulator importance values

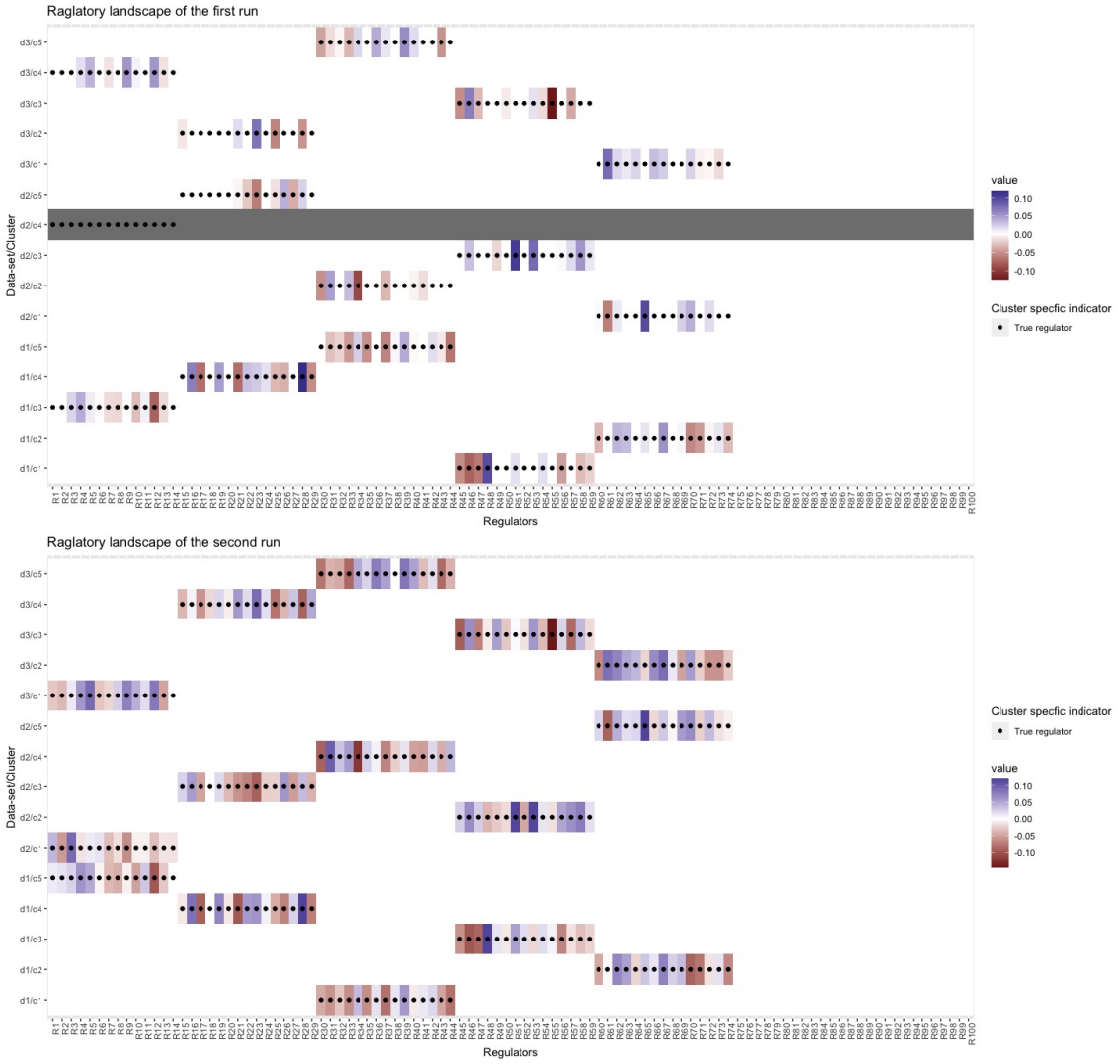

Figure 13: Caption

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.9972	0.8497	1	0.3106	0.0786
d2	1	0.6944	0.3672	1	0.1865	0.1272
d3	1	0.8520	0.4902	1	0.1827	0.1185
d1	2	0.9972	0.9705	1	0.3256	0.0696
d2	2	0.9916	0.9807	1	0.2204	0.0670
d3	2	0.9888	0.9645	1	0.2292	0.0681

Table 11: The table showcases the results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average regulator importance values

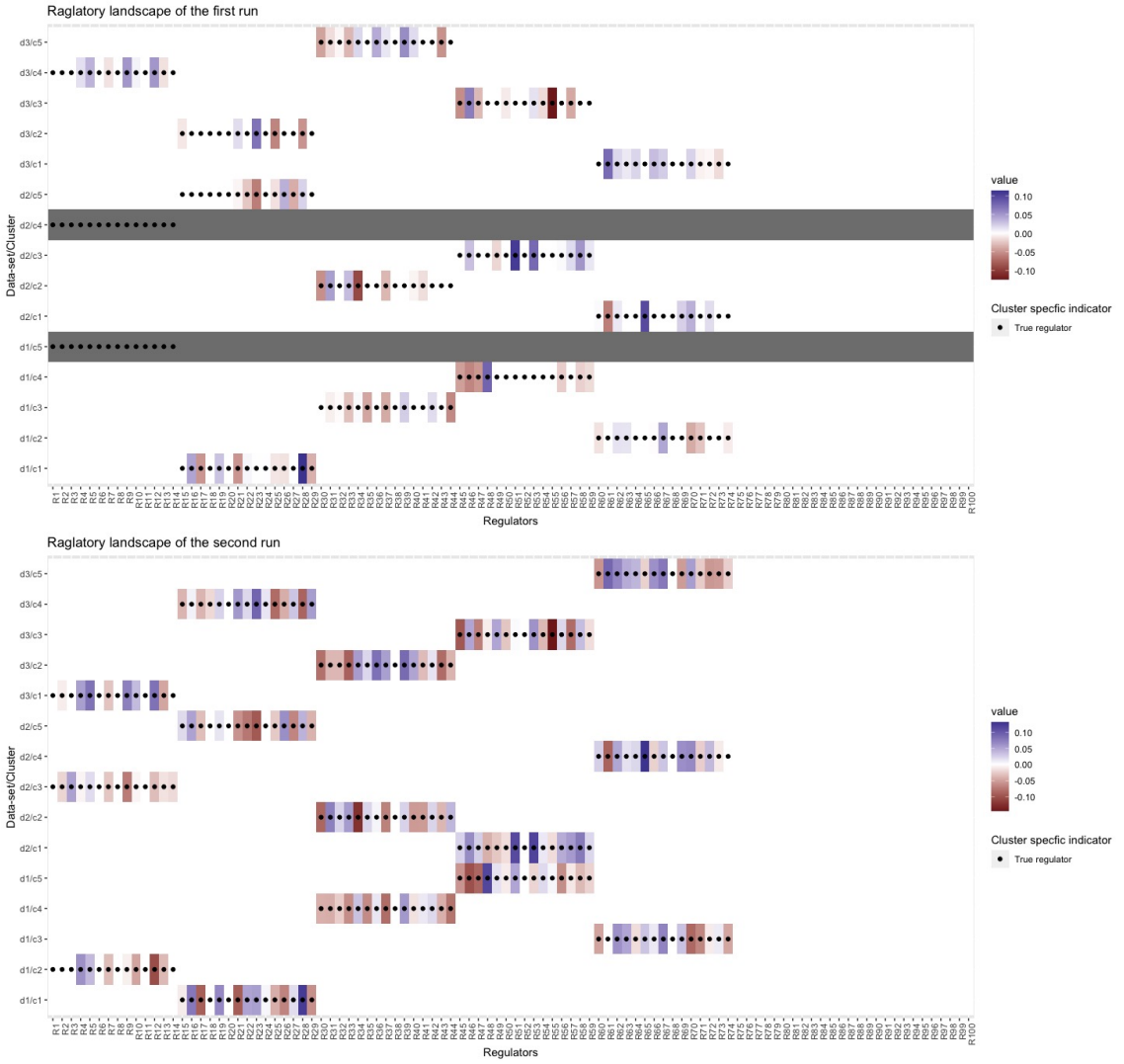

Figure 14: The figure illustrates the active regulator and the corresponding cluster they are linked to for three datasets (d1, d2, and d3) for both the first and second run of the *ScRegClust*. The black dots indicate which regulators are true for each cluster to provide an insight into the accuracy of the ScRegClust results.

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.7165	0.4496	1	0.1929	0.1070
d2	1	0.6944	0.3672	1	0.1865	0.1272
d3	1	0.8520	0.4902	1	0.1827	0.1185
d1	2	0.9497	0.8135	1	0.2052	0.0809
d2	2	0.9696	0.8463	1	0.2056	0.0773
d3	2	0.9724	0.8466	1	0.2157	0.0765

Table 12: The table showcases the results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average regulator importance values

A.1.2 Figure and Tables of Simulation Exercise 2

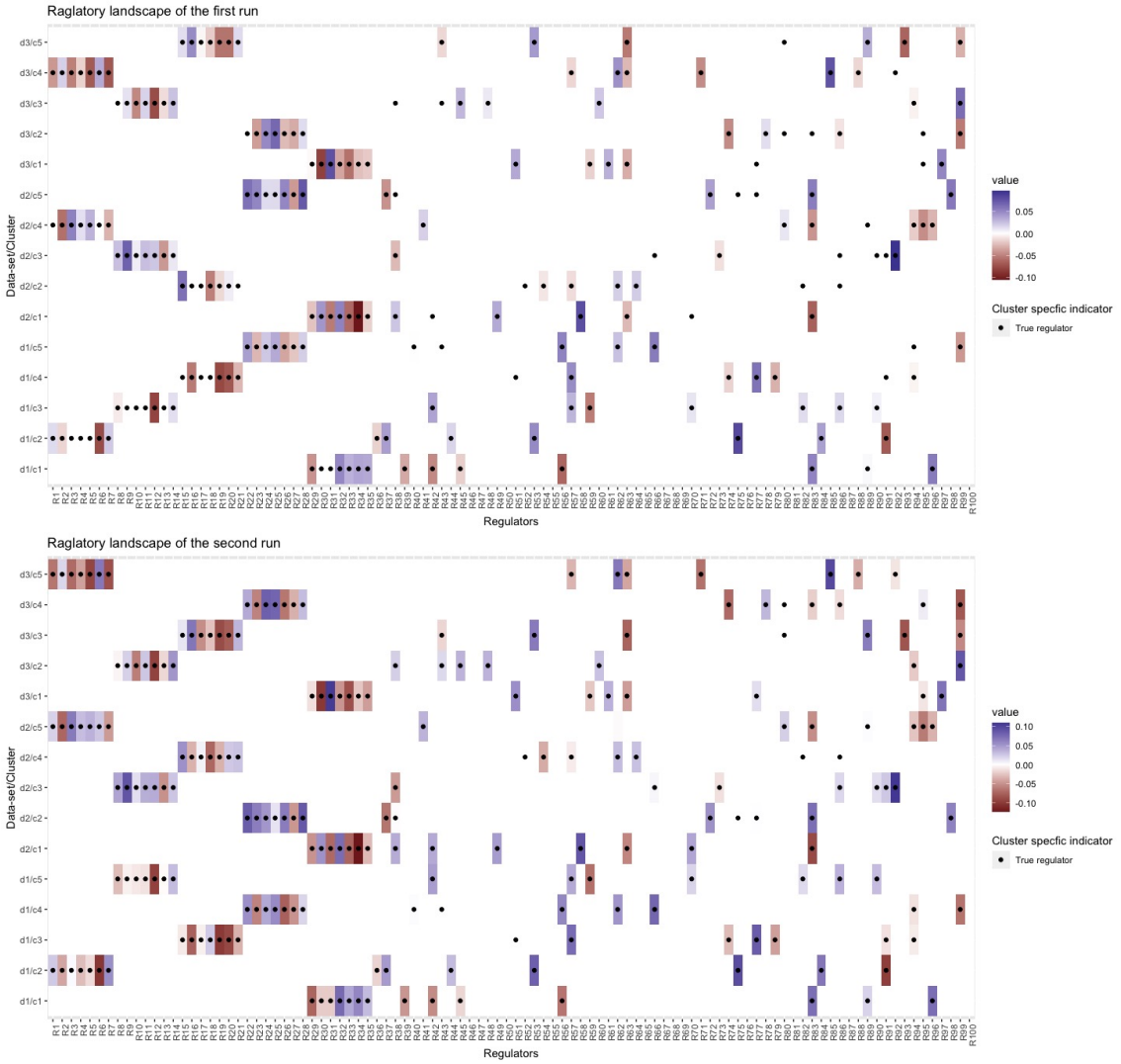

Figure 15: Caption

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.9872	0.8630	0.9997	0.3160	0.0793
d2	1	0.9591	0.8098	1	0.3141	0.0837
d3	1	0.9897	0.9200	0.9992	0.3296	0.0742
d1	2	0.9928	0.9660	0.9997	0.3192	0.0726
d2	2	0.9937	0.9377	0.9940	0.3272	0.0712
d3	2	0.9925	0.9645	0.9992	0.3386	0.0719

Table 13: The table showcases the results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average regulator importance values

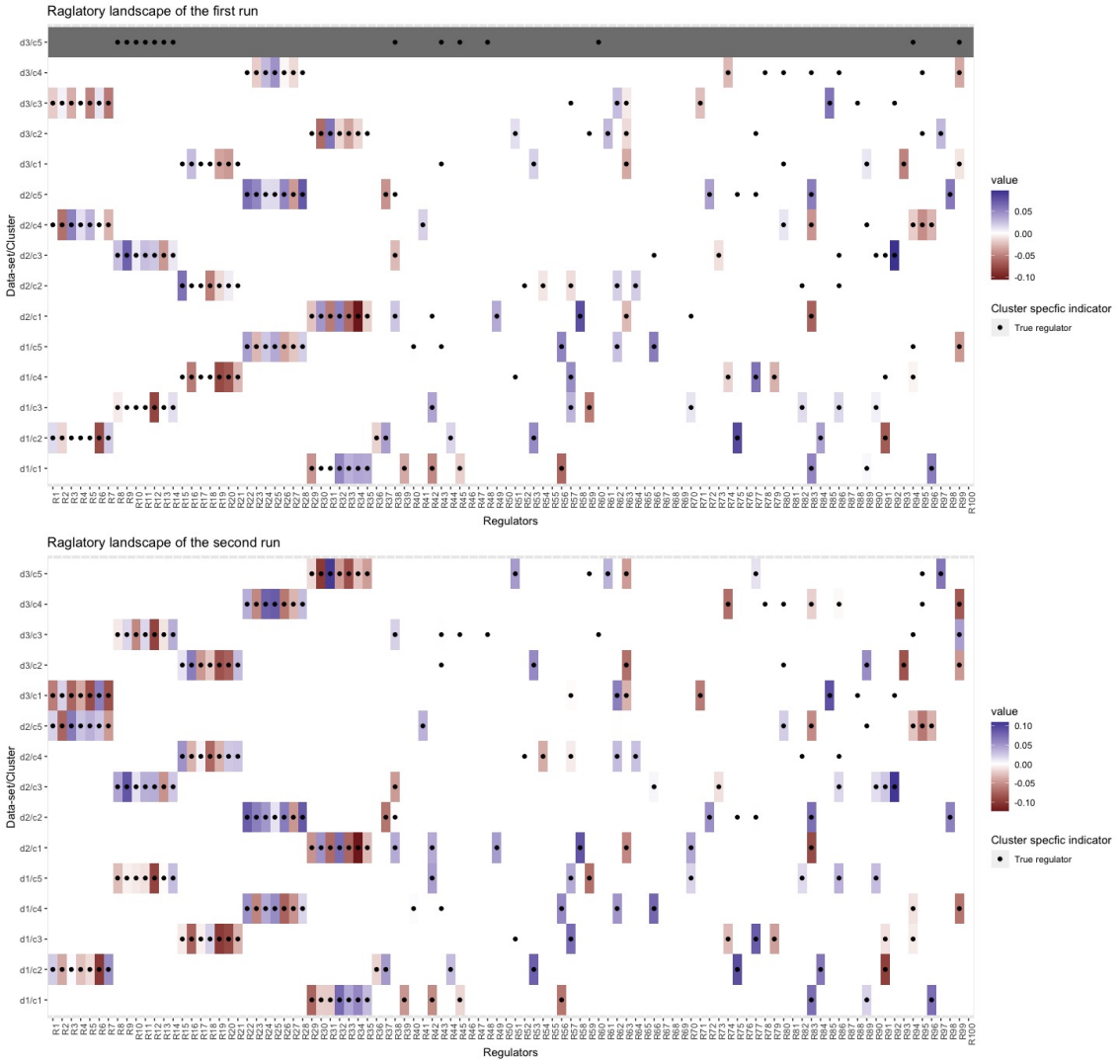

Figure 16: The figure illustrates the active regulator and the corresponding cluster they are linked to for three datasets (d1, d2, and d3) for both the first and second run of the *ScRegClust*. The black dots indicate which regulators are true for each cluster to provide an insight into the accuracy of the ScRegClust results.

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.9872	0.8630	0.9997	0.3160	0.0793
d2	1	0.9591	0.8098	1	0.3141	0.0837
d3	1	0.7572	0.4843	0.9995	0.1920	0.1125
d1	2	0.9928	0.9660	0.9997	0.3192	0.0726
d2	2	0.9913	0.9225	0.9971	0.3265	0.0753
d3	2	0.9676	0.7947	0.9970	0.2057	0.0847

Table 14: The table showcases the results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average regulator importance values

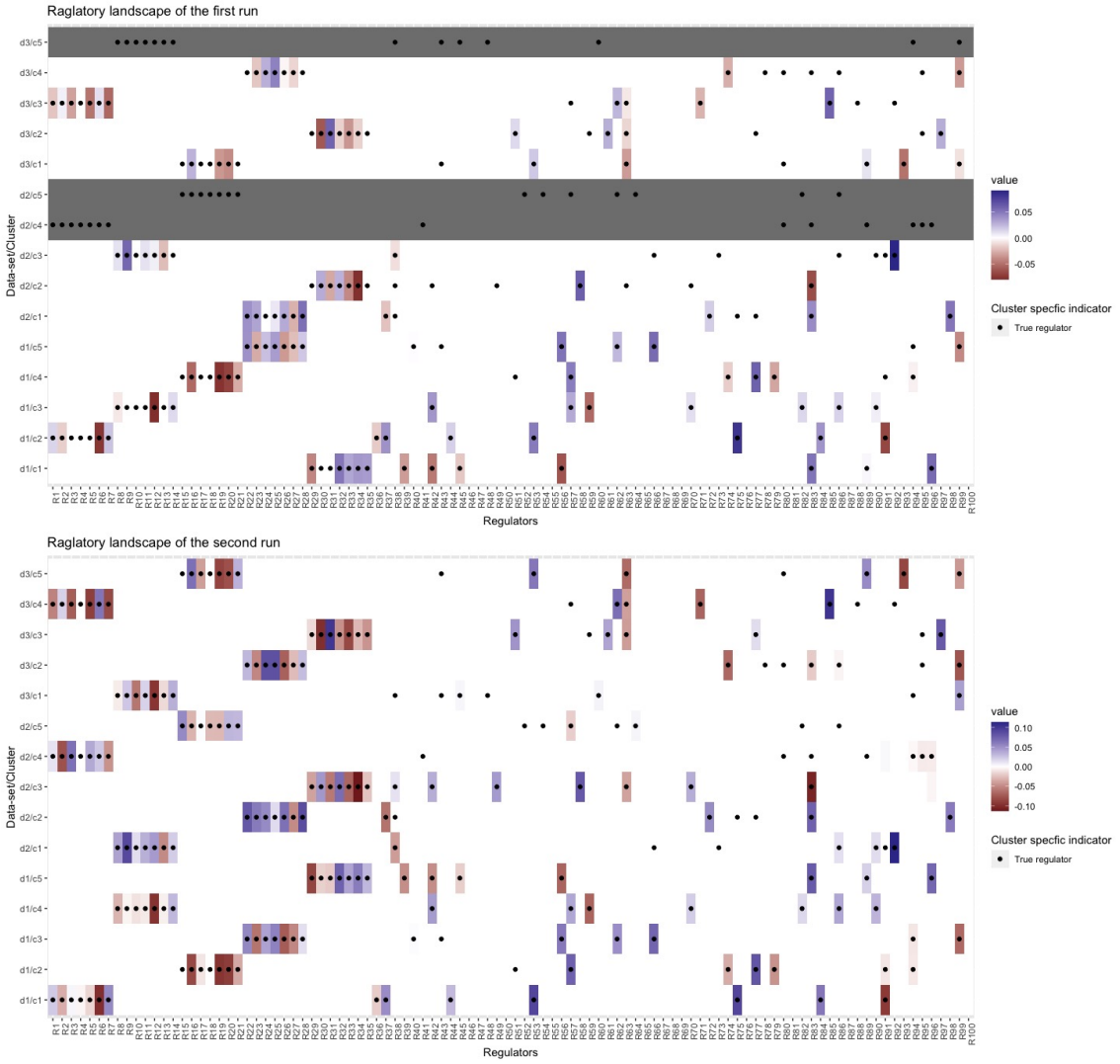

Figure 17: The figure illustrates the active regulator and the corresponding cluster they are linked to for three datasets (d1, d2, and d3) for both the first and second run of the *ScRegClust*. The black dots indicate which regulators are true for each cluster to provide an insight into the accuracy of the ScRegClust results.

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.9872	0.8630	0.9997	0.3160	0.0793
d2	1	0.5195	0.3687	0.9968	0.2136	0.1075
d3	1	0.7572	0.4843	0.9995	0.1920	0.1125
d1	2	0.9900	0.9356	0.9997	0.3188	0.0741
d2	2	0.8950	0.7117	0.9912	0.1989	0.0878
d3	2	0.9591	0.7335	0.9993	0.1952	0.0939

Table 15: The table showcases the results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average regulator importance values

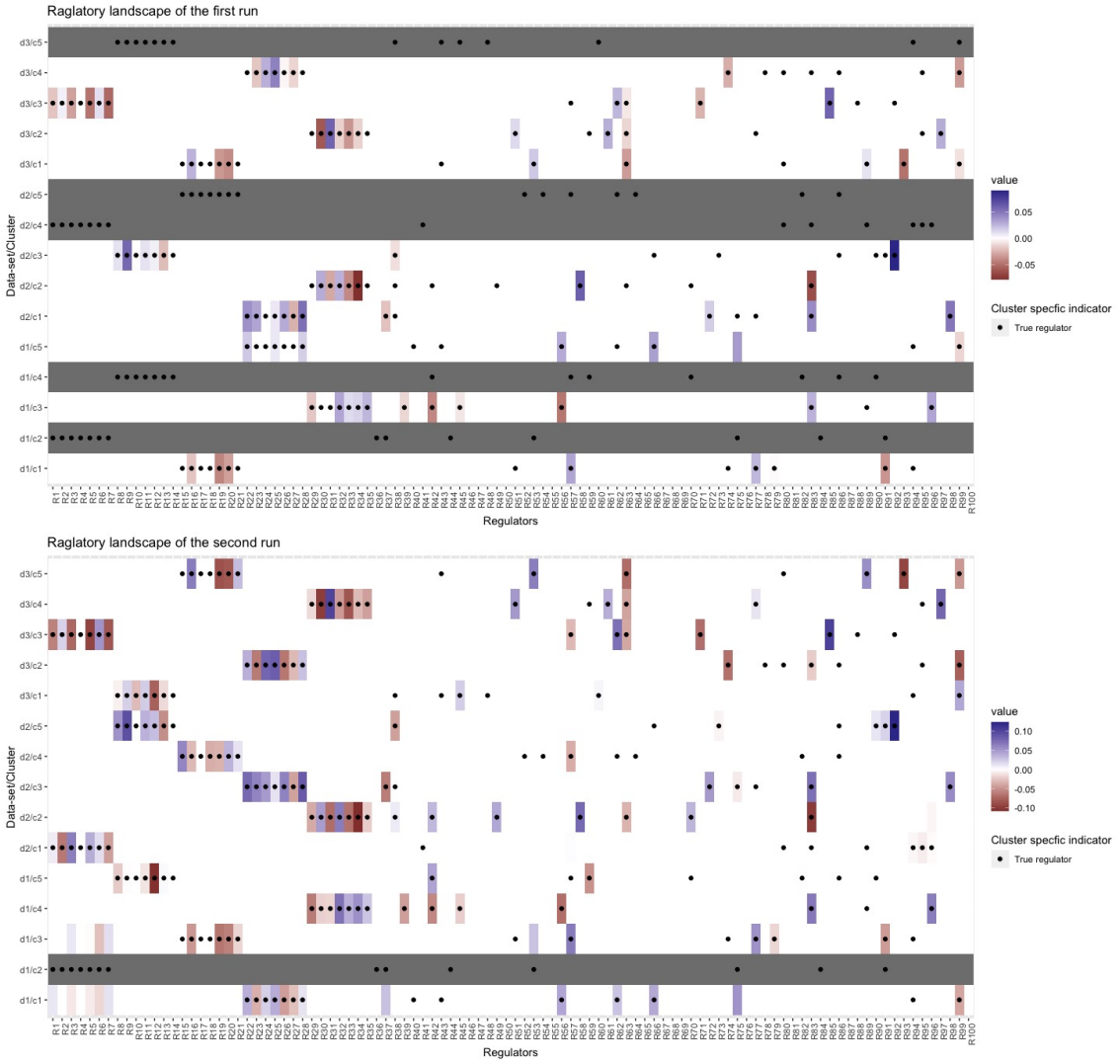

Figure 18: The figure illustrates the active regulator and the corresponding cluster they are linked to for three datasets (d1, d2, and d3) for both the first and second run of the *ScRegClust*. The black dots indicate which regulators are true for each cluster to provide an insight into the accuracy of the ScRegClust results.

Data set	Run	Rand index	Avg. Sens.	Avg. Spec.	Avg. pred. R^2	Avg. Reg. Imp.
d1	1	0.5533	0.3394	0.9886	0.1690	0.1074
d2	1	0.5195	0.3687	0.9968	0.2136	0.1075
d3	1	0.7572	0.4843	0.9995	0.1920	0.1125
d1	2	0.7960	0.6051	0.9473	0.1773	0.0740
d2	2	0.8694	0.6764	0.9906	0.1975	0.0922
d3	2	0.9480	0.7013	0.9994	0.1951	0.0967

Table 16: The table showcases the results of the first and second run of *ScRegClust*, displaying the Rand Index, Average Sensitivity, Average Specificity, Average Predictive R^2, and Average regulator importance values