VARIABLE SELECTION BY **SU**BSAMPLING **R**ANKING **F**ORWARD SELECTION (SURF)

by

Lihui Liu

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
November 2023

This thesis is dedicated to my children Tony, Dylan and Grace.

Table of Contents

Abstract

Traditional statistical methods face lots of challenges in model fitting, variable selection and model diagnosis when analysing high-dimensional data. LASSO is one of the most popular regularised approaches for high dimensional data such as gene expression in microbiome research. However, it often selects a large number of noise variables and it does not provide a direct quantitative assessment of the significance of each variable selected. We present a new variable selection method **Su**bsampling **R**anking **F**orward selection (SuRF) based on penalised regression, subsampling and forward-selection methods. We apply our method to classification problems from microbiome data, using a novel agglomeration approach to deal with the special tree-like correlation structure of the variables. Existing methods arbitrarily choose a taxonomic level *a priori* before performing the analysis, whereas by combining SuRF with these aggregated variables, we are able to identify the key biomarkers at the appropriate taxonomic level, as suggested by the data.

The default standardisation used in LASSO regression is effective for the normal predictors, but not for predictors from heavy-tailed distributions. We presented a large scale of simulations showing that heavy-tailed predictors have a large impact on variable selection and prediction in Binomial and Poisson regression, and a less pronounced effect in Gaussian regression. This can cause the model to underselect the true predictors from heavy-tailed distributions such as log-normal and *Pareto* distributions, and to overselect those variables in Poisson regression. SuRF is less influenced by the distribution of the predictors. A Box-Cox transformation generally improves the selection rate of the heavy-tailed predictors for both SuRF and Stability Selection in Binomial regression, but it can cause a diverse effect in Poisson regression.

Generalised additive models (GAMs), a type of non-parametric additive model, are a natural choice to extend SuRF to select predictors with a non-linear relation to the response. Replacing GLMs with GAMs is necessary in both the ranking and the forward-selection steps of SuRF. SuRFgam demonstrates a superior performance in both nonlinear variable selection and the prediction accuracy. It is particularly effective in reducing the noise variables, making it a better choice in various modelling scenarios.

Acknowledgements

I want to extend my profound gratitude to my supervisors Dr. Hong Gu and Dr. Toby Kenney, for their guidance, encouragement, and support throughout my study. This long journey has been both challenging and rewarding, your mentorship has not only enriched my academic experience but also inspired me to be resilient and strong. I am deeply thankful for all that you have done to make this achievement possible.

My heartfelt appreciation also goes out to the members of my thesis committee and the dedicated staff within our department. Your invaluable feedback, unwavering support, and precious time contributed significantly to my accomplishments.

Furthermore, I would like to take a moment to express my deep gratitude to my dear friend, Yun Cai. The memories we created together, whether in the office, during conferences, or during any of our shared moments, will forever hold a special place in my heart.

Finally, I wish to extend my gratefulness to my family. Your love has been the driving force behind my accomplishments, and I am deeply thankful for your understanding and unconditional support.

Chapter 1

Introduction

As technology continues to advance and data collection and storage solutions become increasingly accessible, the volume of information generated in almost every field has grown exponentially. While the process of collecting data has become more straightforward and more economical nowadays, extracting meaningful insights from complex data poses a series of more challenging issues than ever. Effectively harnessing the enriched information and identifying key indicators for predicting the outcomes is a common and pressing problem across numerous research fields and domains, including finance, marketing, environmental science, medicine, microbiome studies, and more.

One of the fundamental challenges in modern data analysis is that the number of predictors, denoted by p, is often larger than the number of observations, denoted by N. When this occurs, the results of many traditional statistical methods become very unreliable.

There are several approaches to deal with this issue. One approach is to develop new methods which can provide reasonable predictions from data with more predictors than observations. Another is to reduce the dimension of the data by creating summary predictors, which summarise a large number of predictor variables. Another important approach, and the topic of this thesis, is variable selection, where we select a small number of the predictors, which contain most of the information needed to predict the response variable.

Variable selection has a number of advantages over other methods. Identifying the important predictors can be important for improving our scientific understanding. Predicting the response from a small number of key predictors also reduces the cost of future data collection, by allowing researchers to only collect a small number of key variables.

Variable selection methods are generally classified into four main types of methods. Filter methods mostly assess predictors based on general features of the predictors, such as correlation with the response. Often, filter methods consider each predictor in isolation, making them very fast, but unable to account for the correlation between predictors. Wrapper

methods and embedded methods select the variables based on a criterion. The difference between wrapper methods and embedded methods is in the search strategy. Wrapper methods fully fit the model on the selected variables, before changing the selection, usually by either adding a variable to the selected set (forward selection) or by removing a variable from the selected set (backward selection). By contrast, embedded methods simultaneously fit the model and select the variables, usually by optimising an objective function that includes a penalty term which is not differentiable with respect to a coefficient when the coefficient is zero. The final class of methods is ensemble methods, which aggregate the results of a large number of variable selection methods to form a consensus variable selection.

Many of the most popular variable selection methods are embedded methods, based on the Least Absolute Shrinkage and Selection Operator (Lasso), which was introduced by Robert Tibshirani in the 1990s in the context of linear regression, and has subsequently been extended to generalised linear models (GLMs). The idea is to minimise the negative log-likelihood, plus a penalty term which is proportional to the L_1 norm of the fitted vector of linear coefficients.

$$L(\beta) = -l(\beta; X, y) + \lambda \sum_{i=1}^{p} |\beta_i|$$

where λ is a tuning parameter that controls the sparsity of the fitted model. The effect of this loss function is to both shrink the coefficients towards zero, and also to select variables by shrinking the coefficients of noise variables to actually equal zero.

There have been a number of variations of the Lasso method developed based on using a modified penalty function. These are designed to address a variety of variable selection situations. A few key Lasso based methods include FUSED Lasso (2005, [60]), Elastic net (2005, [80]), Group Lasso (2006, [73]), Adaptive Lasso (2006, [79]), Dantzing selector (2007, [7]), Smoothly clipped absolute deviation (SCAD) (2008, [64]) and Scaled Lasso (2012, [55]).

In the Lasso and related approaches, the variable selection or the model selection heavily relies on the choice of the turning parameter(s). Common techniques for choosing the parameter include Cross-validation (CV) and AIC/BIC scores. The selection of variables can vary significantly in size depending on these parameter choices, resulting in inconsistent prediction performance. Additionally, the variable selection can be also influenced by some observations or outliers.

Many of these issues are addressed by Stability Selection [39], which is an ensemble method based on subsampling and Lasso. The method takes a large number of subsamples of the original dataset, without replacement, of size half the size of the original data, and applies Lasso to each subsample over a range of tuning parameter values. The final model chosen includes all variables that are selected (non-zero coefficients) for above a chosen proportion of subsamples over some tuning parameter value. The authors of Stability Selection [39]suggest the proportion should be in the range 0.6–0.9, with larger values selecting a more parsimonious model. By controlling the Lasso tuning parameter and the cut-off, it is possible to control the Per Family Error Rate (PFER), resulting in a very sparse model.

Stability Selection has been widely used in a range of regression and classification problems [28, 48, 53, 33, 4]. However, it does have certain limitations:

- The cut-off value is chosen arbitrarily, and does not have an intuitive interpretation. This means that it is hard to control the parsimony of the selected model. Simulation studies performed by [31] have demonstrated that in order to get acceptable variable selection, the cut-off Π_{thre} may need to be set lower than 0.5 in cases where $p \gg N$, and conversely, it may need to exceed 0.9 when $p < N$. The sensitivity of variable selection to the choice of Π_{thre} is a concern and determining an appropriate threshold beforehand can be challenging.

- In datasets with multicollinearity issues, each subsample may select one variable from a set of surrogate variables, with different surrogates selected in different subsamples, resulting in none of them being selected in the final model.

- The selection can become inconsistent when the assumption of similar distributions of sub-samples is violated.

- A 50% sub-sampling proportion is often overly restrictive, especially when the original sample size is small. Using a subsample with a very limited number of observations can reduce the power of the selection procedure.

This thesis focuses on variable selection for Microbiome research problems. Microbiome data consists of counts of particular microbes in an environment. There is substantial evidence [12, 58, 10, 54, 23] that the microbiome has a large influence on a number of areas

ranging from health to environment to agriculture. However, the sheer numbers of microbes make interpretation of microbiome data a challenge. Furthermore, identification of biomarkers that serve as indicators of specific health conditions, diseases, or environmental states, such as the presence of algae bloom in a lake, can improve our understanding of the underlying mechanisms, and give rise to more effective monitoring of these conditions.

It is therefore crucial to develop variable selection in the context of microbiome research. However, microbiome data has a number of features which cause existing variable selection methods to perform badly. Firstly, microbiome data are very high dimensional, with thousands of predictors, and the expense and difficulty of collecting samples means that datasets often contain only several hundred samples or fewer. Secondly, the predictors are highly correlated. Approximate collinearity of predictors is well known to make variable selection challenging. Thirdly, the predictors are structured in a taxonomic tree structure, representing the evolutionary relationships between the microbes. Since evolutionary close microbes are relatively similar, there is a prior belief that the relations between closely related predictors and the response are more likely to be similar. Fourthly, most microbial relative abundances are very sparse and heavy-tailed. The effect of the distribution of predictors on variable selection methods is a seriously understudied problem, to which an entire chapter of this thesis is devoted.

1.1 Organisation of the Thesis

In Chapter 2, we develop a new variable selection method, called **Su**bsampling **R**anking **F**orward selection (SuRF), for regression and classification purposes, particularly for microbiome analysis. Our method is based on Lasso penalised regression, subsampling and forward-selection methods. We apply our method to classification problems from microbiome data, using a novel agglomeration approach to deal with the special tree-like correlation structure of the variables. Existing methods arbitrarily choose a taxonomic level *a priori* before performing the analysis, whereas by combining SuRF with these aggregated variables, we are able to identify the key biomarkers at the appropriate taxonomic level, as suggested by the data. We also present simulations in multiple sparse settings to demonstrate that our approach performs better than several other popularly used existing approaches in recovering the true variables. We apply SuRF to two microbiome data sets: one about prediction of pouchitis and another for identifying samples from two healthy individuals.

We find that SuRF can provide a better or comparable prediction with other methods while controlling the false positive rate of variable selection.

Chapter 3 explores the impact of the distribution of predictors on variable selection from Lasso-based methods in Gaussian, Binomial and Poisson GLM models. Standardisation of the predictors for Lasso is recommended as a default to ensure Lasso is scale-invariant. While standardisation in terms of standard deviation is appropriate for normal predictors, the standard deviation is not always such a good measure of scale for heavy-tailed distributions. The lack of a more appropriate standardisation method for heavy-tailed predictors leads to worse performance of Lasso-based methods. The simulation results confirm that the predictors' distributions usually have limited effect on variable selection for the Gaussian regression models. In contrast, the heavy-tailed predictors are usually under-selected in the Binomial logistic regression, and over-selected in Poisson regression models. Furthermore, this bias in variable selection reduces prediction accuracy in these cases. Box-Cox transformation of the predictors can improve variable selection of some methods, even when it results in misspecified models, but does not completely remove the impact of predictor distribution.

In Chapter 4, we adapt our SuRF method to a new method called SuRFgam (**Su**bsampling **R**anking **F**oward selection for *generalised additive models*) for variable selection in generalised additive models (GAMs). GAMs are a type of non-parametric additive model, that is able to model non-linear relations between predictors and the response. In particular, GAMs model the conditional expectation of the response through a link function as a sum of smooth functions of each predictor, in order to capture non-linear effects of the predictors. Replacing GLMs with GAMs is necessary in both the ranking and the forward-selection steps of SuRF. In the forward-selection step, this can be routinely done by replacing the GLM by a GAM. For the ranking step, we use *Gamsel* [11] (Generalised Additive Model Selection), which is a variable selection method for GAMs, based on group Lasso. We conduct a comprehensive simulation study to compare SuRFgam with various state-of-the-art methods for variable selection in GAMs, including linear methods and non-linear methods. We compare performance on Gaussian and Binomial regression model settings across a range of data dimensions and signal strengths. SuRFgam demonstrates a superior performance in both nonlinear variable selection and prediction accuracy. It is particularly effective in reducing the noise variables, making it a better choice in various

modelling scenarios.

Chapter 2

SuRF: A new method for sparse variable selection, with application in microbiome data analysis

2.1 Introduction

Traditional statistical methods face lots of challenges when analysing high-dimensional data. These challenges occur in model fitting, variable selection and model diagnosis. A series of regularised models have become popular inference approaches for high dimensional data such as gene expression. The most well-known methods include Lasso regression [59], the elastic-net regression model [80] and various variations such as group Lasso [73], Bayesian Lasso [42], etc. Lasso is based on penalising the model by the sum of the absolute values of coefficients of all variables and hence it is a soft thresholding method so that some variables are eliminated due to a resulting zero coefficient. This has the advantage of selecting sparse models. In addition, it is a suitable method to use for tree-structured data, such as microbiome data, as we discuss in Section 2.4.

There are a few issues with the use of Lasso for microbiome data. The first is inference — Lasso can select a parsimonious model, but it does not provide a direct quantitative assessment of the significance of each variable selected. For scientific and clinical research, it is vital to include these assessments of the significance of variables (p-values). There is a method related to this matter [37] but, in practice, high dimensional data rarely satisfies the weak collinearity assumption needed. The more robust approach of Tibshirani *et al.* [61] is only available for Gaussian response variables. Secondly, Lasso provides only a list of variables, with coefficients, but in many cases very strong correlation exists between some variables, either of which might be selected with no indication that the other variables might have an almost equally strong association with the response variable. The choice of which variables are selected can be very unstable.

We introduce a variable selection method, SuRF, based on regularised regression and subsampling of observations in the generalised linear model setting in this chapter. This

method provides a p-value for each variable. The p-values are for forward selection, so should be interpreted for the null hypothesis "All true variables have already been selected". In cases where variables are correlated, the p-value assigned to a given variable measures the extent to which that variable improves prediction compared to the model already selected. A variable might have a high p-value if a surrogate variable is already included, and a variable with a low p-value could be a surrogate of the true predictor.

SuRF also gives information on the stability of the selected variables. There has been previous work on dealing with the lack of stability in Lasso, such as Zakharov and Dupont [75] and Grave *et al.* [24]. A promising recent approach to this issue which has some similarity to our SuRF method is Stability selection [39]. Stability selection has proven to be a very popular and effective method for variable selection.

Another good variable selection method is best subset selection. As the name implies, this method chooses the set of variables of a given size that optimises some suitable criterion. While this is intuitively appealing, actually finding this optimal subset is an NP-hard problem. However, the recent work of Bertsimas *et al.* [2] has provided an efficient method for approximately finding an optimal subset of a given size. Choosing the number of variables is still a challenge in best subset selection. Recently Zhang and Cavenaugh [76] provide a method to use bootstraps in combination with a corrected AIC to solve this issue.

Another popular variable selection approach that we consider is VSURF [21]. This uses the variable ranking produced by the Random Forest method to select variables using a stepwise approach. This has many conceptual similarities with SuRF, in that it is a ranking procedure followed by a stepwise procedure. However, it differs from SuRF, not only in the use of Random Forest for ranking and selecting the variables. Random Forest is based on subsamples, but it does not apply variable selection to the subsamples. Instead, the variable importance ranking is based on the difference in cross-validated prediction accuracy with and without the selected variable.

We are particularly interested in applications to microbiome research. The microbiome is the collection of all bacteria present in a location, e.g. a person's gut, and one of the main questions of microbiome research is the relationship between phenotypes (e.g., healthy versus disease groups) and the microbiome. The data consist of counts of various types of microbes, classified into Operational Taxonomic Units (OTUs). These counts of OTUs are generally normalised to calculate the relative abundances for each microbe in each sample

and the abundance data are used as predictors for certain phenotypes. Through such models, we hope to find the major correlations between microbes and the phenotype under study. Due to the vast number of OTUs and the cost of samples, the sample size is always much smaller than the number of OTUs. In this modelling process, variable selection and the interpretation of the models are the most important results.

We show in our simulations that SuRF gives comparable performance to that of Stability selection [39] in the usual settings when the predictors follow multivariate Normal distributions, and outperforms Lasso and best subset selection. For the simulations where predictor variables are based on real microbiome data, SuRF shows much better performance than Stability selection, best subset selection and VSURF in variable selection and prediction .

The remainder of the chapter is structured as follows: Section 2.2 describes the general SuRF method. Section 2.3 sketches how to prove consistency of SuRF. Section 2.4 describes how we have adapted SuRF for the microbiome data in our example. In Section 2.5, we compare the performance of SuRF with other variable selection and machine-learning methods on simulated data, both with and without the microbiome tree structure. In Section 2.6, we apply SuRF to two real microbiome data sets. In Section 2.7, we study the reliability of the p-values given by SuRF. In Section 2.8, we discuss the extension of SuRF to the Cox-proportional hazards model through a joint work to handle time-to-event outcomes. In Section 2.9, we conclude the chapter with a brief discussion of the advantages of SuRF. In Section 2.10, more detailed tables are presented.

2.2 *Su*bsampling *R*anking *F*orward selection (SuRF)

The framework of SuRF has two main steps. The first step creates a list of ordered predictors that have been selected most frequently by Lasso variable selection on subsamples of the observations. The second step applies ANOVA with forward selection to this ordered list of variables to eliminate or alleviate the issue of surrogate variables. The significance of each variable in the forward selection test is calculated from the likelihood ratio statistic via sequential permutation tests. A sketch of the algorithm is provided in Figure 2.1, with more detailed pseudocode in Section 2.2.3.

2.2.1 Variable Ranking

The subsampling approach plays an important role in formulating the list of top predictors. This technique is widely used in many recent methods for variable selection and the details were summarised well by Dezeure *et al.* [15]. Each subsample is used to perform a variable selection procedure and the results from all subsamples are used to rank the importance of variables according to the frequency of variables being selected.

In principle, any good variable selection methods can be implemented here, but we focus on Lasso for the linear predictors in this chapter because of its empirically better screening performance than other regularised models [6] and its computational speed. In Chapter 4, we will use another method *Gamsel* to deal with non-linear predictors in the ranking. In addition, we find that Lasso is particularly suitable for the variables from the microbiome data (see Section 2.4). For each subsample, we record the variables selected by Lasso, with the tuning parameter selected by cross-validation over the subsample. We rank variables by the frequency they are selected (Ties are broken by reduction of deviance residuals from models containing all higher-ranked variables). The order of variables can be interpreted as measuring the strength of the association with the response variable.

We recommend about 90% of the data for this purpose when the sample size is extremely limited but otherwise the proportion appears to make a minimal difference in results for values in the range 50–90% in the simulations (see Table 2.9). In classification problems, we also recommend taking stratified subsamples (subsamples having the same proportion in each class as the true data) since if the subsamples are not balanced, this can affect prior probabilities of each class, resulting in worse classification. The subsampling procedure is repeated in a large number of times. Some literature suggests 50 or 100 times can produce decent accuracy [15], but as the computation this step is small compared to other parts of the method, there is little cost to performing this step in a much larger number of times (e.g. 1000 times in this chapter) to ensure higher accuracy.

2.2.2 Sequential permutation tests with ANOVA forward selection

The forward selection involves sequentially testing the null hypothesis that no variables beyond the currently selected variables are good predictors for the response variable, given the currently selected variables. In forward selection, the order in which variables are added to the model can be important. At each stage, if multiple predictors are significant,

we add the one ranked highest by our variable ranking procedure. Because we are testing multiple predictors at each stage, Wilks' theorem [66] (that the log-likelihood ratio statistic follows a χ^2 distribution) does not apply. The predictors are not independent, so Bonferroni correction [5] cannot be used. Instead we calculate the critical value empirically using permutations [44].

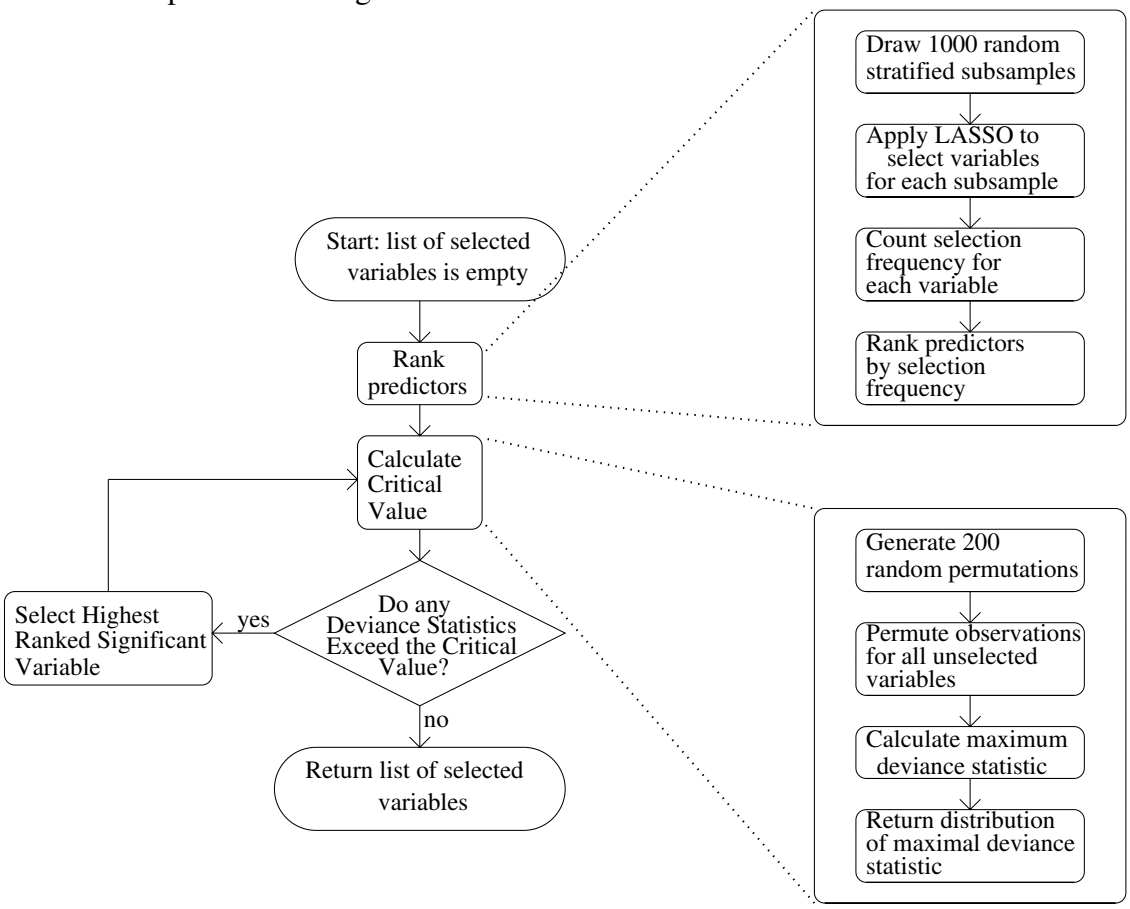
More specifically, we start with a list of candidate variables, containing all variables in the order found using the ranking method from Section 2.2.1, and a list of selected variables being initially empty. At each step, we generate a random permutation for all observations, and apply it only to the variables in the candidate variable list. This breaks the relationship between the candidate variables and the response variable while preserving the correlation structures, both between selected variables and the response variable, and among candidate variables. Now for each candidate variable, we compute the log-likelihood ratio statistic between the current model and the model with this variable added. We record the largest log-likelihood ratio statistic. We repeat this process for many more permutations (we usually use 200 permutations as a compromise between accuracy and speed), to obtain the null distribution of the maximum log-likelihood ratio statistic. We use the $100(1-\alpha)$th percentile ($\alpha = 0.05$) of this null distribution as our critical value, denoted as $D^i_{1-\alpha}$ for the ith variable in this forward sequential variable selection procedure. We now return to the original unpermuted data, and for each candidate variable in the ranking order, we calculate the log-likelihood ratio statistic between the current model, and the model with this candidate variable added. We select the first candidate variable for which this statistic exceeds $D^i_{1-\alpha}$, and add this variable to the model (and remove it from the candidate variable list). We then generate a new distribution, with new permutations and repeat the same procedure. When the log-likelihood ratio statistic for each candidate variable is no greater than the critical value, we terminate the algorithm and output the current model as the final model selected.

For each variable added to the model, SuRF has computed a p-value based on the comparison of the likelihood ratio statistic with the empirically calculated null distribution. This p-value is based on the increase in training fit over the variables that were already included in the model. That is, the p-value is for the null hypothesis that all true variables are in the current model. A variable which is a surrogate for a variable that has been already selected may not have a significant p-value if there is not significant evidence that this

variable improves prediction over the surrogate already selected.

Figure 2.1: Overview of the SuRF method

More detailed pseudocode is given in Section 2.2.3.



2.2.3 SuRF Algorithm: single cut-off method

2.2.4 Revised algorithm: variable selection path

In the revised SuRF package, we have modified the algorithm to accommodate the need for refitting the data with a different significance level α. This feature proves particularly useful for visualising the selection path along a predetermined significance level range [α_l, α_u] (e.g., [0.01,0.2]) in a single model fitting. The idea of the new algorithm is to store the variable selections at each step as a decision tree, based on the significance level range. Each node on the tree contains a set of selected variables and a significance level range. The

Algorithm 1 Variable ranking

Require: X matrix of predictors (e.g. proportions for OTU data)

function VRANK($X_{N\times P}, y_N, q, B$)

Step1: Create a zero matrix M_{BP}

for $i \leftarrow 1$ to B **do**

Randomly select the i^{th} stratified subsample of size qN;

Fit Lasso with this subsample;

$M[i, T_i] = 1$; ▷ T_i selected variable set chosen from i^{th} subsampling

end for

Step2: Rank variables by their frequencies $F_q = \sum_b M[b, q]$;

Step3: In case of a tie, re-rank variables according to the contribution of how each of them decreases the LR statistics by adding each variable to a model already including the variables listed above those tied variables;

Step4: Further reduce the above list by removing the variables listed at the bottom with a low frequency.

return $R = \{R_1, R_2, \ldots, R_{P'}\}$

▷ R represents the ranked variable set (from most frequently selected to the least frequently selected)

end function

Algorithm 2 Variable selection (main)

function VSELECT($X_{N\times P}$, y_N, R, T) ▷ T: the number of permutations

$C \leftarrow R$ ▷ C: Candidate variable set

$S \leftarrow \emptyset$ ▷ S: Selected variable set

repeat

newcutoff=DeriveSampDist($X_{N\times P}$, y_N, C, S, T);

$x^\star$=SelnNewvar($X_{N\times P}$, y_N, C, S, newcutoff)

$S \leftarrow \{S, x^\star\}$

$C \leftarrow \{C \backslash x^\star\}$

until

All observations have been perfectly classified or $x^\star == \emptyset$

return

end function

Algorithm 3 Variable selection step 1: derive the sampling distribution

function DeriveSampDist($X_{N\times P}, y_N, C, S, T, \alpha = 0.05$) ▷ T: the number of permutations

for $i \leftarrow 1$ to T **do**

Select a random permutation π_i

Permute all rows of variables listed in C with π_i

for $j \leftarrow 1$ to w **do** ▷ $w = |C|$: the size of the set

$P_j \leftarrow \{S, C_j\}$

$D_j \leftarrow -2 \ln \frac{L_S}{L_{P_j}}$

end for

$D_i^{max} \leftarrow \max_j D_j$

end for

$D_{1-\alpha}^{max} \leftarrow \{D_1^{max}, \ldots, D_T^{max}\}_{1-\alpha}$ ▷ $D_{1-\alpha}^{max}$: the new cut-off value

return $D_{1-\alpha}^{max}$

end function

D_j Likelihood ratio statistics between model using variables in S and in L_{P_j}

L_S: Likelihood of the model including variables in the current selected variable set S

L_{P_j}: Likelihood of the model including variables in the current selected variable set S plus the j^{th} candidate variable in candidate set C

Algorithm 4 Variable selection step 2: select a new variable

function SELNEWVAR($X_{N\times P}, y_N, C, S, newC$)

for $k \leftarrow 1$ to w **do** ▷ $w = |C|$: the size of the set

$P_k \leftarrow \{S, C_k\}$

$G_k \leftarrow -2 \ln \frac{L_S}{L_{P_k}}$

if $G_k > newC$ **then**

return $xnew = C_k$;break;

end if

end for

return $\emptyset$

G_k Likelihood ratio statistics between model using variables in S and in L_{P_k}

L_S: Likelihood of the model including variables in the current selected variable set S

L_{P_k}: Likelihood of the model including variables in the current selected variable set S plus the k^{th} candidate variable in candidate set C

end function

top node contains the empty set of variables and the full significance level range $[\alpha_l, \alpha_u]$. At a given node, we perform the usual procedure to form the ranked list of predictors and the null distribution of the deviance. We then descend down the list until the significance level needed to select one of the variables X_i is less than α_u. If the necessary significance level is less than α_l, then X_i is selected, and there is a single branch below the current node. If the significance level α_i is between α_l and α_u, then our tree will branch with one node having range $[\alpha_i, \alpha_u]$, and variable X_i selected (in addition to the variables selected in the current node). We will then continue to descend further down the list, to see if any more variables are selected for significance level $\alpha < \alpha_i$. If no more variables are selected, then the second branch has range $[\alpha_l, \alpha_i]$, and selects no additional variables, thus terminating the algorithm for that branch. If X_j is selected at significance level $\alpha_j < \alpha_i$, then a new branch with range $[\alpha_j, \alpha_i]$ (or $[\alpha_l, \alpha_i]$ if $\alpha_j < \alpha_l$) is created, and X_j is selected on this branch. This process is repeated until a variable is selected at significance $\alpha_k \leqslant \alpha_l$, or until no variable is selected in the lowest range. The process is then repeated for all lower nodes for which a variable is selected. The permutation test only needs to be performed once for each node of this tree, regardless of the actual significance level used. In theory, the permutation test may apply to multiple nodes with the same set of variables selected, but the benefit of this is usually limited, as SuRF is parsimonious, so it is fairly rare for the same set of variables to be selected in different orders at different significance levels.

2.3 Theory

SuRF is designed to combine the best parts of three methods: Stability selection, Lasso and forward selection. The advantages and disadvantages of these methods are as follows. Forward Selection provides clear p-values at each stage, but heavily depends on the order of variables entering the model. Lasso does well at identifying the correct variables, but does not provide p-values and often selects too many variables. Stability selection is robust to outliers, but can "fall between two stools" with surrogates, and offers only limited p-values.

Because the variable selection in SuRF is mainly controlled by the forward selection part, SuRF retains the advantages of forward selection — namely the clear p-values and sparsity. However, by choosing the order of variables using Lasso, SuRF is able to avoid the pitfall of choosing variables that do not work well with other predictors. Because the final

selection in SuRF is made using forward selection, it is able to consider variables which are not selected often in the subsampling. This can be important in cases with surrogate variables, where the subsamples could be closer to evenly split. In such cases, Stability selection is likely to select both or neither, depending on its cut-off value. SuRF is able to consider variables that are chosen in fewer subsamples, relying on forward selection to avoid selecting too many variables.

We will base our theory on forward selection, since forward selection is responsible for the final selection decisions made by SuRF. We want to show that asymptotically, SuRF will select the true variables. This occurs in two stages. Firstly, the true variables must be ranked highly by the subsampling procedure. Otherwise, a surrogate variable may be tested and selected before the true variable. At best, this will be a false positive. In worse cases, this could prevent the true variable from being selected. The true variables being ranked highly relies on the performance of Lasso at identifying the true variables. It is known that Lasso is consistent provided the irrepresentability condition holds [78]. In this case, the subsampling is asymptotically guaranteed to select the true variables before other variables, which overcomes the danger with forward selection that the surrogate will be selected first, preventing the true variables from entering the model. In addition, the subsampling approach should offer some robustness, allowing the top variables to be highly ranked even if some outliers might make them less highly ranked in some subsamples. Secondly, assuming the true variables have been ranked above other variables, the hypothesis test must reject the null model when true variables haven't all entered the model. We can prove that, assuming:

1. the true variables are ranked before all other variables by the ranking procedure,
2. $\log(p_n) = o(n)$, where p_n is the number of predictors included in the dataset with n observations and
3. $B\alpha \geqslant 1$, where B is the number of permutations used in calculating the null distribution and α is the significance level for the forward selection test.

SuRF will almost surely select all true variables for large enough n.

We establish the following notation:

- The true model is $\mathbb{E}(Y|\mathbf{X}) = g^{-1}(\mathbf{X}\beta)$, where $Y|X$ follows an exponential family distribution and g is a link function. $\mathbf{X}$ is an $n \times p_n$ matrix.

- V_{true} denotes the set of indices of "true" predictors, i.e. $\beta_i = 0$ for all $i \notin V_{\text{true}}$ and $\beta_i \neq 0$ for any $i \in V_{\text{true}}$.
- $p_{\text{true}} = |V_{\text{true}}|$ is the number of true predictors, and is fixed as $n \to \infty$.
- For a set V of selected indices, $\widehat{\underset{\sim}{\beta}_V} = \arg\max_{\{\underset{\sim}{\beta} | (\forall i \notin V) \underset{\sim}{\beta}_i = 0\}} l(\underset{\sim}{\beta}; Y, \mathbf{X})$ is the estimate for $\underset{\sim}{\beta}$ restricted to elements of V and $l_V = \sup_{\{\underset{\sim}{\beta} | (\forall i \notin V) \underset{\sim}{\beta}_i = 0\}} l(\underset{\sim}{\beta}; Y, \mathbf{X})$ is the maximum log-likelihood for the selected variables.
- For a set V of selected indices, $\beta^*_V = \arg\max_{\{\underset{\sim}{\beta} | (\forall i \notin V) \underset{\sim}{\beta}_i = 0\}} \mathbb{E}(l(\underset{\sim}{\beta}; Y, \mathbf{X}))$ is value of the coefficient vector, $\underset{\sim}{\beta}$ restricted to elements of V, which maximises the expected log-likelihood, or equivalently, minimises the Kullback-Leibler divergence from the true distribution.
- Let $D(V, k) = 2(l_{V \cup \{k\}} - l_V)$.

We first prove a lower bound on the log-likelihood ratio statistic $D(V, k)$ when $\underset{\sim}{\beta}_k \neq 0$. We start by proving a lower bound on the expected log-likelihood.

Lemma 1. *If $V \subsetneq V_{true}$, then there is some $j \in V_{true}$ for which $\mathbb{E}(l(\beta^*_{V \cup \{j\}}; Y, \mathbf{X})) > \mathbb{E}(l(\beta^*_V; Y, \mathbf{X}))$*

Proof. We know that for an exponential family model, the likelihood function is concave. This means that if β^*_V is not the global maximum for $\mathbb{E}(l(\underset{\sim}{\beta}; Y, \mathbf{X}))$, then it must have a non-zero derivative in the direction $\beta - \beta^*_V$. Since it has a non-zero derivative, in this direction, it must have non-zero derivative in one of the component directions. Let k be one of the component directions in which there is a non-zero derivative. Then we must have $\mathbb{E}(l(\beta^*_{V \cup \{k\}}; Y, \mathbf{X})) > \mathbb{E}(l(\beta^*_V; Y, \mathbf{X}))$, which completes the proof. □

Next we use standard asymptotic theory to show that this bound holds for the finite sample values with large probability.

Theorem 3.1: *For $V \subsetneq V_{true}$, there is some $\epsilon > 0$ such that*

$$P\left(\sup_{j \in V_{true} \setminus V} D(V, j) > n\epsilon\right) \to 1$$

Furthermore, if the log-likelihood ratio is almost surely $o(n)$ — that is if

$$\frac{l(\widehat{\underset{\sim}{\beta}_V}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} \to 0$$

a.s., then $\sup_{j \in V_{true} \setminus V} D(V, j) > n\epsilon$ *a.s.*

Proof. We have that

$$\begin{aligned}
\frac{D(V,k)}{2} &= l(\widehat{\beta_{V\cup\{k\}}}; Y, \mathbf{X}) - l(\widehat{\beta_V}; Y, \mathbf{X}) \\
&= l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X}) + \left(l(\widehat{\beta_{V\cup\{k\}}}; Y, \mathbf{X}) - l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X}) \right) \\
&\quad - \left(l(\widehat{\beta_V}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X}) \right) \\
&\geqslant l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X}) - \left(l(\widehat{\beta_V}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X}) \right)
\end{aligned}$$

By Lemma 1, there is some $j \in V_{\text{true}}$ such that $\mathbb{E}(l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X})) - \mathbb{E}(l(\beta^*_V; Y, \mathbf{X})) > \epsilon$. By the strong law of large numbers, for any k,

$$\frac{l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} \to \mathbb{E}(l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X})) - \mathbb{E}(l(\beta^*_V; Y, \mathbf{X}))$$

almost surely, so with probability 1, for large enough n,

$$\left| \frac{l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} - \left(\mathbb{E}_X(l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X})) - \mathbb{E}_X(l(\beta^*_V; Y, \mathbf{X})) \right) \right| < \frac{\epsilon}{6}$$

Therefore, $\frac{l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} > \frac{5}{6}\epsilon$. Therefore, if $\frac{l(\widehat{\beta_V}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} < \frac{\epsilon}{3}$, then $D(V, k) > n\epsilon$.

By Wilks' theorem, we have that $2(l(\widehat{\beta_V}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X}))$ converges in distribution to a χ^2 distribution. Therefore $\frac{l(\widehat{\beta_V}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n}$ converges in probability to 0. Thus

$$\begin{aligned}
&P\left(\sup_{j \in V_{\text{true}} \setminus V} D(V, j) > n\epsilon \right) \\
\geqslant &1 - P\left(\frac{l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} < \frac{5}{6}\epsilon \right) - P\left(\frac{l(\widehat{\beta_V}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} > \frac{\epsilon}{3} \right) \to 1
\end{aligned}$$

Furthermore, if $\frac{l(\widehat{\beta_V}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} \to 0$ a.s., then with probability 1, there is some N such that for all $n > N$, and some $k \in V_{\text{true}}$, $\frac{l(\beta^*_{V\cup\{k\}}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} > \frac{5}{6}\epsilon$ and $\frac{l(\widehat{\beta_V}; Y, \mathbf{X}) - l(\beta^*_V; Y, \mathbf{X})}{n} < \frac{\epsilon}{3}$, which means $\sup_{j \in V_{\text{true}} \setminus V} D(V, j) > n\epsilon$.

□

On the other hand, we can study how the null distribution changes with n and p.

Theorem 3.2: *Suppose X is an $n \times p_n$ matrix and Y is a random vector (independent of X) from an exponential family distribution, then for any $V \subseteq \{1, \ldots, p_n\}$, the maximum deviance $Q = \sup_{j \notin V} D(V, j)$ of a single column of X as a predictor of Y has survival function asymptotically bounded by $S_Q(x) \leqslant \frac{2p_n e^{-\frac{x}{2}}}{\sqrt{2\pi x}}$, where $S_Q(x) = P(Q > x)$.*

Proof. For a single column j of X, the deviance $D(V, j)$ asymptotically follows a chi-square distribution with one degree of freedom. Its survival function is therefore $S(x) = 2\Phi(-\sqrt{x})$. The survival function of the maximum of p_n such distributions is therefore bounded by $S_Q(x) \leqslant 2p_n\Phi(-\sqrt{x})$ (with no assumptions about the joint distribution). Recall for $u < 0$, we have

$$\begin{aligned}
\Phi(u) &= \int_{-\infty}^{u} \frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi}}\, dt \\
&\leqslant \int_{-\infty}^{u} \frac{t}{u} \frac{e^{-\frac{t^2}{2}}}{\sqrt{2\pi}}\, dt \\
&= \frac{1}{\sqrt{2\pi}u} \left[-e^{-\frac{t^2}{2}} \right]_{-\infty}^{u} \\
&= \frac{-e^{-\frac{u^2}{2}}}{\sqrt{2\pi}u}
\end{aligned}$$

In particular, setting $u = -\sqrt{x}$ gives us $S_Q(x) \leqslant 2p_n\Phi(-\sqrt{x}) \leqslant 2p_n \frac{e^{-\frac{x}{2}}}{\sqrt{2\pi x}}$ This means that the survival function is bounded by $\frac{2p_n e^{-\frac{x}{2}}}{\sqrt{2\pi x}}$. □

For the hypothesis test, we sample B permutations of the data, and for each permutation, compute the values $D(V, j)$ for each $j \notin V$, and take the maximum $\sup_{j \notin V} D(V, j)$. This gives B simulated deviances. The critical value is the $100(1 - \alpha)$th percentile of these simulated deviances. Since the deviance for the true variable is larger than $n\epsilon$, for some $\epsilon > 0$, we will reject the null hypothesis if fewer than αB simulated deviances are larger than $n\epsilon$. If $S_Q(x)$ is the survival function of the simulated deviance distribution, then the probability that more than αB simulated deviances are larger than $n\epsilon$ is asymptotically bounded by

$$\binom{B}{B\alpha} S_Q(n\epsilon)^{B\alpha} \leqslant B^{B\alpha} \left(\frac{2p_n e^{-\frac{n\epsilon}{2}}}{\sqrt{2\pi n\epsilon}} \right)^{B\alpha} = \frac{(2B)^{B\alpha}}{\sqrt{2\pi}^{B\alpha}} \left(\frac{p_n}{\sqrt{n\epsilon}} e^{-\frac{\epsilon}{2}n} \right)^{B\alpha}$$

This is an upper bound on the probability that SuRF stops before selecting the next variable. The probability that SuRF stops before selecting all true variables is therefore bounded by

$$p_{\text{true}} \frac{(2B)^{B\alpha}}{\sqrt{2\pi}^{B\alpha}} \left(\frac{p_n}{\sqrt{n\epsilon}} e^{-\frac{\epsilon}{2}n} \right)^{B\alpha}$$

By our assumptions, $p_n e^{-\frac{\epsilon}{4}n} \leqslant 1$ for all sufficiently large n, so $p_n e^{-\frac{\epsilon}{2}n} \leqslant e^{-\frac{\epsilon}{4}n} \to 0$, so the probability of SuRF selecting all true variables converges to 1.

Furthermore, $\sum_{n=1}^{\infty} p_n e^{-\frac{\epsilon}{2}n} \leqslant C + \sum_{n=1}^{\infty} e^{-\frac{\epsilon}{4}n} < \infty$. Given a sequence of datasets, X_n, which are $n \times p_n$ matrices of predictors, with corresponding response variables Y_n, let T_n be an indicator variable for the event that the selected variables under SuRF include all variables in V_{true}. If $\frac{l(X,\widehat{\beta_V})-l(X,\beta_V^*)}{n} \to 0$ almost surely, then with probability 1, for all sufficiently large n, $P(T_n = 0) < p_{\text{true}} \binom{B}{B\alpha} S_Q(n\epsilon)^{B\alpha}$, so since $\sum_{n=1}^{\infty} \binom{B}{B\alpha} S_Q(n\epsilon)^{B\alpha} < \infty$, we have shown that $\mathbb{E}\left(\sum_{n=1}^{\infty}(1 - T_n)\right) = \sum_{n=1}^{\infty}(1 - P(T_n = 1)) < \infty$, so $P\left(\sum_{n=1}^{\infty}(1 - T_n) < \infty\right) = 1$. Thus, SuRF is almost surely consistent.

2.4 Adapting SuRF to Tree Structured Data

SuRF can be applied to any exponential-family GLM variable selection problem. However, our application of interest is microbiome data. In this section we discuss the particular way we have adapted SuRF to deal with such data. Hierarchical clusterings like the taxonomic tree structure of microbiome data are common in many types of data, so this adaptation has wide applicability. We can also use this adaptation with other variable selection methods. The benefits of the aggregation described here depend on the nature of the data being analysed. For microbiome data, we recommend using this aggregation method with all variable selection methods, since the OTU level predictors are already somewhat arbitrary aggregations of strains.

Microbiome data typically consist of proportions of OTUs present in each sample. OTUs are clusters of DNA sequences, usually clustered at 97% similarity, approximately equivalent to species-level resolution. We are working with a GLM $g(\mathbb{E}(Y|X)) = \beta_0 + X\beta$, where X is the column-centralised OTU data matrix with each column representing an OTU variable. The phylogenetic relationships among OTUs provide us with prior knowledge about β. Namely, we expect the β_i to be close for closely-related OTUs because of phenotypic similarity. It is a standard assumption in microbiome research that phylogenetically close organisms have similar functions in the microbiome. We reflect this prior knowledge via the regularisation of the coefficients. We choose to base this on the taxonomic tree, rather

than more detailed phylogenetic trees, because estimation of the phylogenetic tree is subject to a lot of noise, and the taxonomic tree is easily available from the output of most pipelines. However it is trivial to use a phylogenetic tree instead.

A common practice is to aggregate variables at an arbitrarily chosen taxonomic level, usually genus or phylum. That is, to replace the original data matrix X by the aggregated data matrix $\widetilde{X} = XC$, where C is the clustering matrix at the chosen level. For example,

$$C_{ij} = \begin{cases} 1 & \text{if OTU } i \text{ is in phylum } j, \\ 0 & \text{otherwise.} \end{cases}$$

Now, fitting a model $g(\mathbb{E}(Y|X)) = \widetilde{X}\alpha + \beta_0$ is equivalent to fitting $g(\mathbb{E}(Y|X)) = X\beta + \beta_0$, where $\beta = C\alpha$. That is, this regularisation consists of the restriction $\beta = C\alpha$, namely that OTUs from the same phylum have the same coefficients.

While aggregating at a sufficiently high taxonomic level can have the convenient consequence that classical statistical methods can be applied, the aggregated data may lack the resolution to answer the scientific questions, or may lead us to make unsupported or false generalisations. On the other hand, the large noise when analysing at a low taxonomic level may obscure general patterns, and not provide a satisfactory prediction [29].

The trouble with aggregation at a certain taxonomic level is that it converts the soft prior expectation that coefficients for OTUs in the same group should be similar into a hard requirement that the coefficients be equal, even if this is disproved by the data. Instead, we penalise the extent to which the coefficients differ. More formally, instead of setting $\widetilde{X} = XC$, we set $\widetilde{X} = X(C, I)$, where (C, I) is a matrix whose first columns are C, and whose remaining columns are the identity matrix. Now instead of the standard Lasso penalty $\|\beta\|_1$, we apply the modified penalty $\inf_{(C,I)\alpha=\beta}\|\alpha\|_1$. Here we refer to an α with $(C, I)\alpha = \beta$ as an interpretation of β. We are interpreting the coefficients β according to groups. For example, if $G = \{A, B, C\}$ is one group, then the variable selection "A and B" could also be interpreted as "G, except for C". These two interpretations both refer to the same fitted model, and we may choose whichever minimises the penalty term. The penalised regression problem can easily be optimised by fitting a standard Lasso regression to the GLM $g(\mathbb{E}(Y|X)) = \widetilde{X}\alpha$. This GLM is not identifiable because of the linear dependence between columns. There are multiple transformed coefficient vectors α that give rise to the same coefficient vector β. However, this is not a problem, because the coefficient vector β is still identifiable, and it is only the different interpretations of this vector that are not identifiable. Often, for a given

coefficient vector β, there is a unique interpretation α that minimises the penalty. This is the simplest interpretation of the fitted coefficient vector. It can be shown that the penalty term for interpretations of β is minimised by α if for any j at the higher taxonomic level, α_j is the median of $\{0\} \cup \{\beta_i | C_{ij} = 1\}$, and for any i in cluster j (i.e. $C_{ij} = 1$) $\alpha_i = \beta_i - \alpha_j$. We can apply the same method after constructing similar aggregations at every taxonomic level. The resulting penalty for a particular coefficient vector β is the most parsimonious total change of coefficients over the taxonomic tree structure. We clarify this with an example:

Figure 2.2(a) shows a small taxonomic tree containing OTUs $X_1, \ldots, X_6$. We create the combinations $X_7 = X_1 + X_2 + X_3$, $X_8 = X_4 + X_5$, $X_9 = X_4 + X_5 + X_6$ and $X_{11} = X_1 + \cdots + X_6$. We do not consider the combination X_{10}, because it is equal to X_7. For the coefficient vector $\beta = (1, 2, 2.5, -2, -1, -0.5)^T$, i.e. the model $g(\mathbb{E}(Y|X)) = X_1 + 2X_2 + 2.5X_3 - 2X_4 - X_5 - 0.5X_6$, the most parsimonious coefficients in terms of the expanded set of predictors are $\alpha = (0, 1, 1.5, -1, 0, 0, 1, -0.5, -0.5, 0)$ as shown in Figure 2.2(b). That is, $g(\mathbb{E}(Y|X)) = X_2 + 1.5X_3 - X_4 + X_7 - 0.5X_8 - 0.5X_9$ is equivalent to the original estimate, but is given a lower penalty by Lasso. Similarly, for the coefficient vector $\beta' = (2, 2, 2.5, -2, -1, -0.5)^T$, in Figure 2.2(c), the most parsimonious coefficients in the aggregated model are $\alpha' = (0, 0, 0.5, -1, 0, 0, 2, -0.5, -0.5, 0)$. In the original model, the penalty assigned to β' is $\lambda(|2| + |2| + |2.5| + |-2| + |-1| + |-0.5|) = 10\lambda$, which is larger than the penalty $\lambda(|1| + |2| + |2.5| + |-2| + |-1| + |-0.5|) = 9\lambda$ assigned to β, whereas, for the aggregated model, the penalty assigned to α is

$$\lambda(|0| + |1| + |1.5| + |-1| + |0| + |0| + |1| + |-0.5| + |-0.5| + |0|) = 5.5\lambda$$

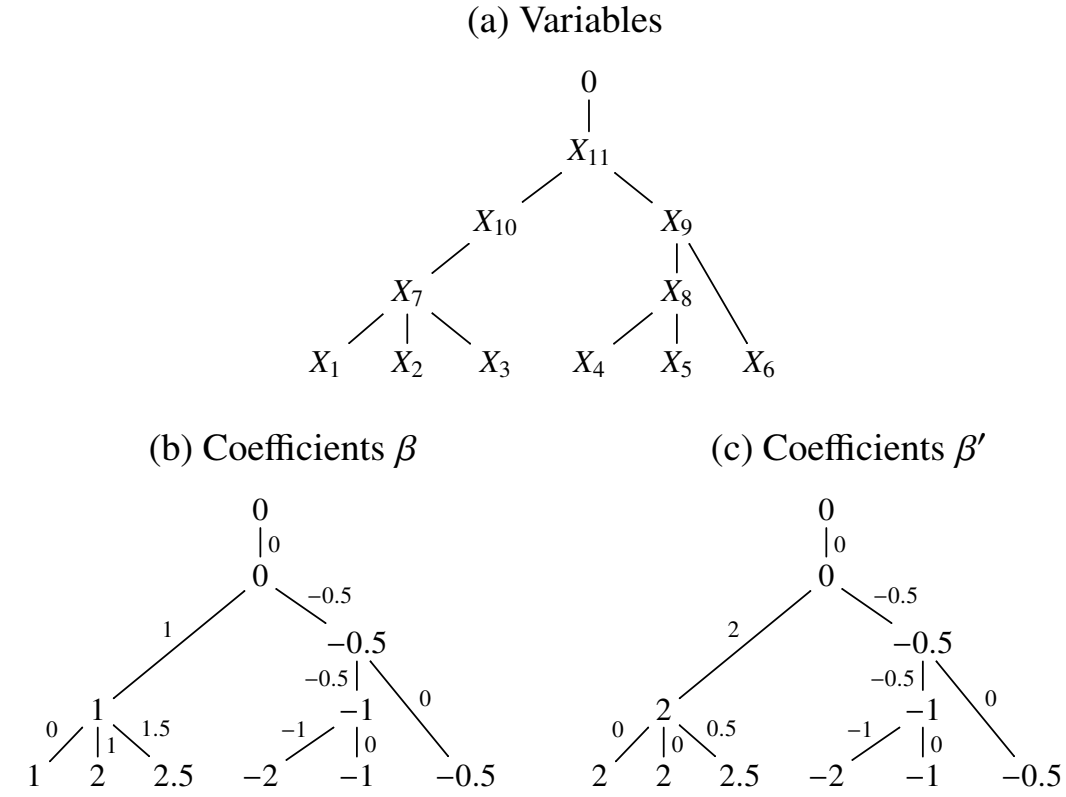
and the penalty for α' is

$$\lambda(|0| + |0| + |0.5| + |-1| + |0| + |0| + |2| + |-0.5| + |-0.5| + |0|) = 4.5\lambda$$

so the penalty for α is larger. Thus, the aggregation approach uses the same space of models, but a different regularisation, which can affect the selected model. In this aggregated setting, we will often say something like "We select the higher level variable X_7" as a shorthand for "We select a model in which variables X_1, X_2 and X_3 are included, but constrained to have equal coefficients."

It is usual to standardise variables prior to applying Lasso. This is to balance the penalty terms between different variables. For microbiome data, the motivation for using

Figure 2.2: Example tree



(a) shows a taxonomic tree relating the OTU variables $X_1, \ldots, X_6$. We create additional variables $X_7, \ldots, X_{10}$ by aggregating the variables below. Let X be the original OTU data matrix $(X_1 \cdots X_6)$. In (b), we consider the estimate $Y = X\beta + \beta_0$ where $\beta = (1, 2, 2.5, -2, -1, -0.5)^T$ and in (c), we consider the estimate $Y = X\beta' + \beta'_0$ where $\beta' = (2, 2, 2.5, -2, -1, -0.5)^T$.

The coefficients in the expanded model are shown on the branches of the trees, and the values shown at internal nodes are cumulative sums of the branches above. For leaf nodes, these are the coefficient β in the original model.

the aggregation method is that phylogenetically similar microbes are often interchangeable, so we expect the coefficients to be similar. This is on the original abundance scale. Therefore, to achieve the correct regularisation, we need to aggregate the abundances prior to standardisation.

Yan and Bien [70] independently develop a similar method involving adding aggregated variables to alter the regularisation. Their approach is tailored to text mining problems, and consequently differs from ours in a couple of respects: Firstly, they include an additional penalty for the coefficients at leaf nodes. This does not make sense for OTU data, since leaf nodes are clusters of lower level strains, so should not be treated differently. Furthermore, additional penalty for coefficients at leaf nodes creates a new optimisation problem. Secondly, their method does not scale the variables before regularisation. For Lasso, standardisation makes predictors more comparable, so that penalties are equivalent. For their count data, the counts are already equivalent. For our tree-based Lasso, it is less clear what standardisation means. Further work on fine-tuning the procedure to produce a better penalty that more accurately reflects this is outside the scope of the current thesis, which focuses on the SuRF procedure, but is a topic the author plans to address in future work.

The theory for this augmented version of Lasso has not been developed. It is not possible to apply the standard theory for the augmented set of predictors, because there are many representations of the true model using the augmented predictors. Described in terms of the augmented predictors, even the notion of consistency is challenging to define — there are multiple correct sets of selected augmented variables, and when we convert back to the original variables, it can be challenging to even determine whether or not a given original variable has been selected. Developing a new theory about the consistency of augmented Lasso is beyond the scope of this thesis, but is an interesting area for future work.

There are several ad-hoc methods in the literature to incorporate tree structure into the Lasso model, for example Xiao et al. [69]. However, an advantage of our aggregated Lasso method is that it is trivial to also incorporate covariates which do not fit into the tree structure, simply by not creating aggregated variables for them. This approach can also be applied to multiple hierarchical clustering structures on the same set of variables e.g. all clades from all gene trees for a given data set. We can do this by simply adding a set of aggregated variables for each clustering.

2.5 Simulation

2.5.1 Study 1: Linear and Logistic Regression with Multivariate Normal Predictors

In this simulation, we simulate the predictors $X_1, \ldots, X_{2000}$ from a multivariate normal distribution, with $\text{Cov}(X_i, X_j) = 0.8^{|j-i|}$ (like an AR1 covariance structure). We perform two studies. In the first study, the response Y follows a normal distribution with mean a linear function of X. In the second study, the response, Y, is binary, with probability an inverse logistic transformation of a linear function of X. We simulate 100 training observations and 100 test samples in each dataset.

In both studies, we consider four cases for true predictors — one true predictor; three true predictors including two consecutive (highly correlated) variables; three true predictors not including two consecutive variables; and eight true predictors with two consecutive pairs. We also consider three different signal-noise ratios: low — 0.7, medium — 1 and high — 3. We simulate 100 replicates in each scenario. We compare SuRF with the significance level set to each of the values 0.05, 0.1, 0.15 and 0.2, Stability with FMER set to 1 and to 0.0526 (this is the expected number of false positives if the significance level α of SuRF is set to 0.05) and proportion set to 0.6, 0.7, 0.8, 0.9. We also compare Lasso, and best subset selection using MIO for optimisation [2] and using APS [76] to choose the number of variables. For each method, we assess results using both the number of true predictors identified, the number of false positives and the predictive performance on test data. Except for Lasso, we fit a GLM using only the selected variables. Using the standard GLM fitting provides a common assessment of the selected variables under each method. However, for Lasso, fitting a GLM on the selected variables performs significantly worse than using Lasso directly on the training data, so we used the results from Lasso directly without refitting the GLM. The test prediction is assessed using R^2 for continuous response and misclassification rate for binary response. Because the average MSE can be heavily influenced by outliers with large MSE, we use the median, rather than the mean.

The test R^2 values for the case where Y is Gaussian are in Table 2.1. The true and false positive rates are given in Table 2.2 and shown in Table 2.17 in Section 2.10 with more details. Average results across all scenarios are very similar between SuRF and Stability selection with FMER upper bound set to 1 and cut-off proportion set to 0.6. Other settings for Stability selection and other methods produce much worse overall results. The poor

results for other settings of Stability selection contradict the claim from Meinshausen and Bühlmann [39] that results are not sensitive to the choice of cut-off. The results are close in the $p = 1$ and $p = 8$ cases. In the $p = 3$ cases, when the true predictors are weakly correlated, particularly when signal-to-noise ratio is not so high, SuRF performs better than other methods. When two of the true predictors are highly correlated, SuRF performs much worse than other methods in terms of the true and false positive rates, and slightly worse in predictive accuracy. This is because SuRF tends to select only one from highly correlated variables, whereas Stability selection is more likely to select both. In the case where two highly correlated variables are both true predictors, this results in a lower true positive rate for SuRF. Because the variables are highly correlated, missing one of the true positive variables does not have a very large effect on prediction using SuRF. In cases where only one of the correlated variables is a true predictor, this results in a lower false positive rate for SuRF, and therefore better prediction.

The average test misclassification rates for the simulation where Y is binary are shown in Table 2.3. The average numbers of true positive and false positive rates are shown in Table 2.4 and Table 2.18 in Section 2.10 with more details. The results are similar to the Gaussian case. Misclassification rates are similar in the $p = 1$ and $p = 8$ cases, and in the $p = 3$ case, Stability selects better predictors than SuRF in the case where two true predictors are highly correlated, and selects worse predictors than SuRF in cases where the true predictors are more weakly correlated. As in the Gaussian case, this is because SuRF is less likely to select two highly correlated variables.

As in the Gaussian case, while stability selection often performs better in individual scenarios, SuRF performs best across all scenarios. The performance of Stability selection is very sensitive to the choice of cut-off value, with the best choice varying between scenarios.

Looking at the variable selection results in more detail, when the three true variables are not highly correlated, SuRF performs better than other methods, achieving the same number of true positives with fewer false positives. In the three true predictor case with two highly correlated predictors, as for the continuous case, SuRF often selects just one of the predictors, resulting in the selection of fewer true variables. In the 8-variable case, SuRF often selects more false positives than Stability for settings where they select a similar number of true predictors. However, from Table 2.3, we see that even in this case, SuRF is able to achieve comparable or better misclassification error than other methods. This

Table 2.1: Average test R^2 for linear regression fitted on selected variables under different methods for a normal response. Results are over 100 simulations.

p	SNR	Measure	SURF				Stability, FMER=1				Stability, FMER=0.0526				Best	Lasso
			0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	Subset	
1	High	Mean	0.755	0.751	0.749	0.747	0.754	0.755	0.756	0.758	0.757	0.757	0.757	0.757	0.734	0.735
		SD	0.017	0.026	0.028	0.029	0.016	0.016	0.012	0.01	0.011	0.011	0.011	0.011	0.025	0.021
	Fair	Mean	0.609	0.607	0.602	0.597	0.605	0.609	0.612	0.612	0.61	0.611	0.612	0.613	0.598	0.461
		SD	0.029	0.031	0.036	0.038	0.025	0.022	0.019	0.019	0.019	0.019	0.019	0.018	0.039	0.073
	Low	Mean	0.467	0.463	0.456	0.444	0.46	0.466	0.468	0.462	0.466	0.467	0.468	0.462	0.458	0.329
		SD	0.0234	0.041	0.047	0.067	0.039	0.033	0.031	0.072	0.033	0.032	0.03	0.072	0.04	0.069
3(a)	High	Mean	0.751	0.755	0.756	0.754	0.769	0.767	0.76	0.739	0.765	0.762	0.754	0.743	0.617	0.71
		SD	0.049	0.041	0.039	0.038	0.026	0.036	0.052	0.075	0.038	0.046	0.059	0.071	0.041	0.04
	Fair	Mean	0.536	0.541	0.54	0.533	0.585	0.569	0.516	0.384	0.576	0.56	0.524	0.434	0.505	0.453
		SD	0.058	0.054	0.06	0.064	0.033	0.055	0.107	0.225	0.046	0.074	0.108	0.184	0.046	0.07
	Low	Mean	0.322	0.32	0.313	0.313	0.334	0.318	0.266	0.151	0.326	0.309	0.274	0.204	0.31	0.265
		SD	0.036	0.036	0.044	0.048	0.06	0.075	0.128	0.16	0.077	0.091	0.124	0.158	0.053	0.072
3(b)	High	Mean	0.79	0.784	0.782	0.782	0.783	0.775	0.744	0.626	0.635	0.673	0.64	0.586	0.726	0.66
		SD	0.016	0.028	0.027	0.026	0.031	0.049	0.083	0.136	0.104	0.108	0.141	0.14	0.034	0.056
	Fair	Mean	0.334	0.334	0.353	0.364	0.275	0.234	0.159	0.084	0.168	0.141	0.104	0.062	0.315	0.187
		SD	0.131	0.131	0.118	0.115	0.107	0.11	0.125	0.115	0.111	0.124	0.12	0.105	0.107	0.117
	Low	Mean	0.196	0.209	0.228	0.227	0.15	0.118	0.064	0.028	0.068	0.049	0.037	0.014	0.162	0.084
		SD	0.121	0.115	0.114	0.11	0.104	0.111	0.105	0.08	0.099	0.092	0.085	0.066	0.132	0.1
8	High	Mean	0.585	0.596	0.602	0.61	0.592	0.583	0.544	0.398	0.579	0.573	0.543	0.475	0.468	0.567
		SD	0.069	0.066	0.069	0.067	0.038	0.045	0.1	0.211	0.043	0.049	0.082	0.158	0.039	0.073
	Fair	Mean	0.385	0.389	0.387	0.387	0. 418	0.379	0.225	0.085	0.389	0.369	0.284	0.116	0.359	0.279
		SD	0.064	0.066	0.07	0.074	0.062	0.112	0.195	0.175	0.104	0.122	0.178	0.187	0.052	0.087
	Low	Mean	0.282	0.281	0.277	0.279	0.28	0.205	0.1	0.033	0.236	0.192	0.134	0.055	0.277	0.177
		SD	0.045	0.056	0.063	0.064	0.105	0.155	0.151	0.113	0.137	0.153	0.158	0.13	0.050	0.096
Average of Mean			0.501	0.503	0.504	0.503	0.508	0.482	0.435	0.363	0.465	0.455	0.428	0.377	0.461	0.409

Table 2.2: Average Numbers of true and false positives for variable selection methods under Gaussian error model over 100 simulations.

(a) **True positive results**

p	SNR	SURF				Stability, FMER=1				Stability, FMER=0.0526				Best	Lasso
		0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	Subset	
	High	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	Fair	0.99	0.99	0.99	0.99	1	1	1	1	1	1	1	1	0.99	1
	Low	0.99	0.99	0.99	0.98	1	0.99	0.99	0.97	0.99	0.99	0.99	0.97	0.99	1
	High	2.09	2.2	2.26	2.28	2.95	2.88	2.81	2.57	2.89	2.86	2.76	2.62	1.09	3
3(a)	Fair	1.3	1.39	1.42	1.48	2.31	1.94	1.38	0.88	1.99	1.8	1.41	0.97	0.31	2.7
	Low	1.12	1.1	1.12	1.13	1.77	1.46	1.03	0.5	1.48	1.27	1.00	0.66	0.2	2.45
	High	3	2.97	2.98	3	2.97	2.9	2.73	2.1	2.02	2.28	2.13	1.82	2.88	3
3(b)	Fair	1.64	1.76	1.81	1.89	1.37	1.11	0.74	0.39	0.65	0.58	0.46	0.29	2.01	2.64
	Low	1.04	1.17	1.35	1.38	0.86	0.67	0.39	0.19	0.32	0.27	0.21	0.12	1.45	2.29
	High	2.03	2.21	2.33	2.49	2.59	2.24	1.78	1.00	2.04	1.88	1.57	1.22	0.83	4.92
8	Fair	0.96	1.01	1.04	1.07	1.43	1.13	0.59	0.22	1.16	1	0.67	0.29	0.69	3.8
	Low	0.8	0.82	0.85	0.88	1.14	0.77	0.34	0.12	0.77	0.59	0.41	0.19	0.69	3

(b) **False positive results** [1]

p	SNR	SURF				Stability, FMER=1				Stability, FMER=0.0526				Best	Lasso
		0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	Subset	
	High	0.05	0.14	0.18	0.24	0.13	0.08	0.04	0	0.05	0.04	0.03	0.01	0.69	8.59
1	Fair	0.06	0.1	0.18	0.27	0.23	0.11	0.03	0.02	0.08	0.05	0.03	0.01	0.3	8.29
	Low	0.06	0.11	0.22	0.36	0.19	0.09	0.06	0.02	0.09	0.07	0.05	0.02	0.26	11.28
	High	0.1	0.15	0.16	0.27	0.4	0.16	0.04	0	0.3	0.15	0.08	0.02	0.14	12.34
3(a)	Fair	0.19	0.26	0.32	0.44	0.5	0.22	0.06	0.01	0.38	0.21	0.15	0.04	1.1	12.09
	Low	0.14	0.18	0.29	0.37	0.44	0.19	0.06	0.01	0.27	0.19	0.11	0.02	1.05	12.96
	High	0.03	0.13	0.2	0.21	0.48	0.19	0.07	0	0.25	0.18	0.07	0.01	2.12	25.65
3(b)	Fair	0.36	0.53	0.61	0.75	0.34	0.15	0.07	0.02	0.2	0.11	0.05	0.02	2.99	19.61
	Low	0.4	0.54	0.59	0.72	0.29	0.09	0.01	0	0.1	0.06	0.03	0.01	3.55	17.22
	High	0.25	0.37	0.49	0.57	0.92	0.51	0.14	0.05	0.82	0.59	0.26	0.08	0.31	16.33
8	Fair	0.35	0.47	0.57	0.65	0.67	0.35	0.15	0.07	0.47	0.31	0.2	0.08	0.72	24.24
	Low	0.33	0.42	0.47	0.54	0.36	0.18	0.07	0.05	0.27	0.18	0.11	0.05	0.61	24.46

indicates that the “false positives” selected by SuRF are still good predictors, i.e. they are surrogates of the true predictors. While SuRF is not able to distinguish which predictor is the true predictor, it is able to select good predictors.

Table 2.3: Average test misclassification error for logistic regression fitted on selected variables under different methods for a binary response. Results are over 100 simulations.

p	SNR	Measure	SURF				Stability, FMER=1				Stability, FMER=0.0526				Lasso
			0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	
1	High	Mean	0.084	0.085	0.088	0.089	0.086	0.084	0.083	0.083	0.167	0.166	0.165	0.083	0.112
		SD	0.014	0.016	0.02	0.02	0.017	0.014	0.014	0.014	0.022	0.021	0.02	0.014	0.042
	Fair	Mean	0.165	0.168	0.17	0.173	0.17	0.167	0.163	0.166	0.167	0.166	0.165	0.17	0.182
		SD	0.023	0.028	0.03	0.032	0.023	0.022	0.018	0.038	0.022	0.021	0.02	0.052	0.036
	Low	Mean	0.204	0.207	0.21	0.211	0.205	0.204	0.211	0.229	0.201	0.201	0.208	0.219	0.245
		SD	0.025	0.029	0.031	0.032	0.029	0.029	0.056	0.088	0.024	0.024	0.048	0.075	0.0045
3(a)	High	Mean	0.158	0.156	0.152	0.153	0.14	0.144	0.161	0.211	0.146	0.148	0.156	0.191	0.15
		SD	0.029	0.029	0.027	0.029	0.023	0.025	0.035	0.099	0.024	0.027	0.032	0.08	0.027
	Fair	Mean	0.254	0.253	0.254	0.254	0.229	0.245	0.277	0.375	0.247	0.251	0.271	0.34	0.252
		SD	0.036	0.036	0.037	0.036	0.041	0.048	0.083	0.121	0.054	0.054	0.073	0.116	0.038
	Low	Mean	0.23	0.23	0.231	0.234	0.227	0.259	0.31	0.412	0.241	0.258	0.314	0.381	0.241
		SD	0.032	0.032	0.034	0.037	0.056	0.097	0.13	0.128	0.081	0.101	0.133	0.139	0.055
3 (b)	High	Mean	0.138	0.137	0.131	0.128	0.224	0.246	0.302	0.383	0.295	0.323	0.368	0.417	0.162
		SD	0.048	0.043	0.036	0.03	0.053	0.049	0.084	0.115	0.072	0.095	0.113	0.112	0.051
	Fair	Mean	0.297	0.285	0.281	0.273	0.333	0.375	0.42	0.471	0.413	0.431	0.451	0.478	0.287
		SD	0.065	0.07	0.068	0.066	0.061	0.083	0.085	0.064	0.086	0.082	0.075	0.057	0.078
	Low	Mean	0.361	0.345	0.337	0.325	0.381	0.423	0.454	0.488	0.442	0.459	0.475	0.489	0.315
		SD	0.074	0.075	0.076	0.075	0.079	0.081	0.068	0.037	0.076	0.068	0.053	0.036	0.093
8	High	Mean	0.207	0.197	0.193	0.188	0.164	0.187	0.261	0.382	0.183	0.195	0.251	0.342	0.178
		SD	0.046	0.048	0.047	0.048	0.032	0.053	0.11	0.138	0.05	0.068	0.11	0.143	0.036
	Fair	Mean	0.292	0.288	0.286	0.285	0.277	0.312	0.38	0.463	0.299	0.326	0.377	0.4425	0.278
		SD	0.024	0.024	0.029	0.03	0.049	0.088	0.11	0.08	0.075	0.092	0.11	0.096	0.048
	Low	Mean	0.337	0.336	0.336	0.334	0.343	0.39	0.452	0.485	0.373	0.398	0.444	0.473	0.332
		SD	0.032	0.032	0.032	0.03	0.054	0.084	0.077	0.049	0.074	0.084	0.08	0.063	0.04
Average of Mean			0.227	0.224	0.222	0.221	0.232	0.253	0.290	0.346	0.265	0.277	0.304	0.335	0.228

Table 2.4: Average Numbers of true and false positives for variable selection methods under logistic model over 100 simulations.

(a) **True positive results**

p	SNR	SURF				Stability, FMER=1				Stability, FMER=0.0526				Lasso
		0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	
	High	1	1	1	1	1	1	1	1	1	1	1	1	1
1	Fair	0.99	0.98	0.97	0.99	1	1	1	0.98	1	1	1	0.97	1
	Low	0.98	0.98	0.98	0.98	1	0.99	0.97	0.91	0.99	0.99	0.96	0.94	1
	High	1.25	1.34	1.4	1.42	2.67	2.45	1.97	1.16	2.34	2.24	1.9	1.26	2.82
3(a)	Fair	1.03	1.05	1.07	1.07	1.79	1.48	1.1	0.57	1.53	1.37	1.04	0.67	2.15
	Low	0.91	0.94	0.94	0.94	1.39	1.02	0.74	0.33	1.11	0.98	0.69	0.43	1.81
	High	2.68	2.73	2.79	2.85	1.86	1.6	1	0.54	0.89	0.79	0.58	0.35	2.96
3(b)	Fair	1.24	1.38	1.43	1.54	1.02	0.77	0.44	0.16	0.42	0.32	0.24	0.11	2.56
	Low	0.71	0.86	0.9	0.99	0.75	0.51	0.33	0.1	0.31	0.27	0.17	0.09	1.82
	High	1.05	1.18	1.24	1.25	1.87	1.47	0.84	0.41	1.55	1.36	0.9	0.55	3.1
8	Fair	0.72	0.72	0.78	0.79	1.15	0.86	0.46	0.14	0.9	0.68	0.46	0.22	2.2
	Low	0.65	0.68	0.68	0.69	0.88	0.54	0.22	0.07	0.59	0.47	0.25	0.13	1.71

(a) **False positive results**

p	SNR	SURF				Stability, FMER=1				Stability, FMER=0.0526				Lasso
		0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	
	High	0.04	0.08	0.17	0.22	0.15	0.02	0	0	0.03	0.01	0	0	2.53
1	Fair	0.06	0.12	0.15	0.27	0.18	0.09	0.02	0.01	0.09	0.06	0.04	0.01	4.38
	Low	0.08	0.13	0.2	0.22	0.17	0.08	0.02	0	0.07	0.04	0.02	0	4.64
	High	0.14	0.16	0.16	0.31	0.37	0.18	0.07	0.01	0.3	0.18	0.07	0.02	4.86
3(a)	Fair	0.15	0.21	0.24	0.28	0.39	0.13	0.04	0.01	0.2	0.14	0.08	0.04	7.62
	Low	0.17	0.21	0.31	0.42	0.26	0.11	0.02	0	0.12	0.08	0.03	0	2.73
	High	0.24	0.33	0.33	0.36	0.28	0.12	0.03	0.01	0.17	0.1	0.05	0.01	17.58
3(b)	Fair	0.43	0.58	0.66	0.76	0.38	0.19	0.07	0.01	0.15	0.1	0.06	0.02	18.17
	Low	0.47	0.52	0.57	0.71	0.33	0.09	0.03	0	0.12	0.04	0.02	0	17.2
	High	0.43	0.47	0.53	0.72	0.81	0.42	0.2	0.06	0.68	0.47	0.27	0.09	10.42
8	Fair	0.36	0.42	0.52	0.53	0.43	0.29	0.14	0.04	0.37	0.27	0.15	0.05	11.31
	Low	0.36	0.46	0.46	0.52	0.44	0.15	0.08	0.02	0.31	0.17	0.08	0.03	9.36

2.5.2 Study 2: Simulation using Pouchitis Data Set

In the remaining simulation studies, we use real microbiome data as the predictors and simulate the response following a generalised linear model. We use the aggregation from Section 2.4 to analyse the data for all variable selection methods.

Our first microbiome simulation is based on the original microbiome data matrix X from the pouch data (afferent limb site) in Tyler *et al.* [62] (see details of the data in Section 4.1). This dataset includes 71 samples with approximately 2000 species level OTUs. We examine our method under the null case (no variable is significantly associated with the outcome variable) and under various sparse settings using variables from higher taxonomic levels: phylum or class. These settings were chosen to be similar to the results from the real data analysis on that dataset. For each simulated dataset we compare the performance of SuRF with several existing popular variable selection methods: Lasso, VSURF [21] and Stability selection. VSURF uses the variable importance from the random forest method to select variables. Stability selection performs Lasso variable selection on a large number of subsamples of the data, and selects the variables that are selected by Lasso for a large proportion of these subsamples.

Because we do not know the underlying distribution of the microbiome data (it is heavy-tailed and skewed) we cannot simulate additional predictors, and due to the sample size, we cannot afford to hold out a test sample, so only in-sample prediction (same predictor matrices for the training and test data, but new values are simulated for the response variable) results are available. The penalty parameter λ for Lasso is obtained by a 5-fold cross validation procedure and we use the largest λ which gives an error within 1 standard error of the best model to select a simpler model. For Stability selection, we adopt a range of threshold probabilities recommended in Meinshausen and Bühlmann [39], between 0.6 and 0.9, and use the default family error rate upper bound parameter of 1 and a value of 0.0526 for this parameter, which is the theoretical number of false positives selected by SuRF at the 5% significance level, so should result in a comparable false positive rate to SuRF. VSURF offers variable selection for different objectives: interpretation and prediction. We compare only the variables selected for prediction, which are always a subset of the variables selected for interpretation.

For assessment of results, we look at both the variables selected and the in-sample predictive accuracy. Variable selection can be used either for interpretation or for prediction.

For microbiome data, the interpretation can be challenging because of the large number of surrogate variables. We therefore view predictive accuracy as the primary objective of our variable selection, with interpretation a secondary goal. However, selection of the true variables is important for both prediction and interpretation. Therefore, we have included it in the results of our simulations.

We simulate a binary response variable, under 5 cases for the true predictors. In the Null case, the response variable Y follows a Bernoulli distribution with probability 0.3, independent of X. In the other cases the logistic transformed probability $\log\left(\frac{P(Y=1)}{P(Y=0)}\right)$ is a linear function of one or two predictor variables. We simulate four scenarios. In Scenario $S1$, the logistic-transformed probability is a function of the abundance of the phylum Bacteroidetes. Nearly all Bacteroidetes in the data set are from the class Bacteroidia, so this forms a very strong surrogate. In Scenario $S2$, the only true predictor is the phylum Firmicutes. This phylum is divided into several classes, so the closest surrogate variable in the dataset has approximately 70% correlation with the true predictor. In Scenario S3, we simulate two weakly correlated true predictors, the phyla Bacteroidetes and Firmicutes, with equal signal strength. In the final Scenario S4, we simulate two predictors at class level, Bacilli and Clostridia. These classes make up the majority of Firmicutes in the dataset.

We simulate 200 replicates for the null scenario and 100 replicates for each non-null scenario. We simulate three levels of signal-noise ratio in each non-null scenario: high — 3.0, fair — 1.0 and low — 0.7. The coefficients of each predictor in each scenario are shown in Table 2.5. These coefficients are calculated to achieve the desired total signal strength using the approximation $\text{SNR} \approx \frac{\text{Var}(\mathbb{E}(Y|X))}{\mathbb{E}(\text{Var}(Y|X))} = \frac{\text{Var}(P(Y=1|X))}{\mathbb{E}(P(Y=1|X)(1-P(Y=1|X)))}$. The coefficients may seem large for logistic regression. This is because of the skewed and heavy-tailed nature of the predictors, meaning that a large proportion of the observations are close to the median, and so contribute little signal. Because the underlying distributions are different for different taxonomic groups, the coefficients that produce the same signal strength also differ between predictors. We use negative coefficients for Bacteroidetes and positive coefficients for Firmicutes, based on the estimated coefficients from the real data analysis.

The results of the variable selection methods are shown in Table 2.6. (For Stability, we only present results for cutoff 0.6 and 0.9 in Tables 2.6–2.8, with more complete results for cutoffs 0.7 and 0.8 in Tables 2.19–2.21). Table 2.6(a) gives the in-sample misclassification error for each method. The prediction results in these simulations are obtained by fitting a

Table 2.5: Coefficients for four different simulation scenarios

Case	SNR	Coefficients of variables (β)			
		Bacteroidetes	Firmicutes	Bacilli	Clostridia
Single variable with one strong surrogate (Case 1)	High	-4.58			
	Fair	-2.84			
	Low	-2.40			
Single variable with no extreme surrogate (Case 2)	High		5.00		
	Fair		2.32		
	Low		1.85		
Two variables with equal strength (Case 3)	High	-4.87	4.39		
	Fair	-1.82	1.64		
	Low	-1.42	1.28		
Two variables equivalent to one variable (Case 4)	High			4.76	4.76
	Fair			2.21	2.21
	Low			1.75	1.75

logistic regression model on the selected variables for each method, except for Lasso, where the model fitted by Lasso was used directly. Tables 2.6(b) and 2.6(c) give the true positive and false positive results for each method. This is not always completely clear-cut because of surrogacy between variables. For Scenarios S1 and S3, we found no method was able to distinguish reliably between Bacteroidetes and Bacteroidia, so we deemed either was correct. However, we deemed the selection of both variables as the inclusion of a noise variable, since once the first variable is included, the second variable does not give additional information. For Scenarios S2 and S3, we did not deem any surrogate of Firmicutes to be acceptable, since the correlation is not so high, and the methods are generally able to distinguish between surrogates and select the true predictor. For Scenario S4, we found that no method was able to identify the true predictors Bacilli and Clostridia. Instead, most methods identified the phylum Firmicutes as a predictor. The phylum Firmicutes is almost entirely comprised of the three classes Bacilli, Clostridia and Erysipelotrichi. Therefore, the combination of Bacilli and Clostridia can also be approximately described as "Firmicutes, with the exception of Erysipelotrichi". We therefore deemed the selection of Firmicutes to be a correct variable, and the selection of Firmicutes and Erysipelotrichi (or an *incertae sedis* genus from within Erysipelotrichi) to be two correct variables.

Across all four non-null scenarios, the in-sample prediction results from SuRF and Stability selection with FMER=1 and cut-off 0.6 are very similar, and better than the other methods compared. (Stability with FMER=1 and cut-off 0.9 shows similar performance in

Table 2.6: Simulation study 2

Scenario	SNR	**SuRF**	**Stability, FMER=1**		**Stability, FMER=0.0526**		**VSURF**	**Lasso**
			0.6	0.9	0.6	0.9		
	(a) **In-sample average misclassification error rate (SD)**							
S1	High	0.095 (0.011)	0.103 (0.044)	0.252 (0.194)	0.119 (0.069)	0.348 (0.191)	0.108 (0.031)	0.126 (0.062)
	Fair	0.190 (0.019)	0.219 (0.080)	0.416 (0.141)	0.243 (0.122)	0.421 (0.138)	0.240 (0.048)	0.365 (0.142)
	Low	0.240 (0.082)	0.274 (0.101)	0.454 (0.107)	0.318 (0.141)	0.454 (0.107)	0.276 (0.027)	0.418 (0.120)
S2	High	0.093 (0.010)	0.095 (0.011)	0.092 (0.008)	0.214 (0.037)	0.275 (0.125)	0.122 (0.018)	0.224 (0.058)
	Fair	0.173 (0.016)	0.178 (0.017)	0.187 (0.074)	0.296 (0.098)	0.408 (0.125)	0.222 (0.023)	0.294 (0.104)
	Low	0.210 (0.020)	0.210 (0.020)	0.282 (0.142)	0.349 (0.110)	0.446 (0.099)	0.266 (0.024)	0.368 (0.144)
S3	High	0.102 (0.010)	0.115 (0.037)	0.196 (0.056)	0.191 (0.082)	0.316 (0.116)	0.124 (0.015)	0.228 (0.063)
	Fair	0.204 (0.080)	0.192 (0.026)	0.316 (0.133)	0.346 (0.108)	0.447 (0.094)	0.232(0.021)	0.311 (0.100)
	Low	0.262 (0.129)	0.232 (0.072)	0.365(0.139)	0.413 (0.116)	0.487 (0.059)	0.265 (0.026)	0.342 (0.127)
S4	High	0.136 (0.030)	0.139 (0.032)	0.152 (0.045)	0.233 (0.036)	0.352 (0.136)	0.117 (0.018)	0.204 (0.059)
	Fair	0.204 (0.016)	0.207 (0.012)	0.318 (0.147)	0.338 (0.116)	0.467 (0.086)	0.220 (0.024)	0.356 (0.160)
	Low	0.245 (0.077)	0.231 (0.055)	0.408 (0.129)	0.407 (0.112)	0.481 (0.064)	0.254 (0.025)	0.403 (0.152)
	(b) Frequency of selecting all true variables (frequency of selecting at least one true variable) [1]							
S1	High	100	98	58	96	38	82	100
	Fair	98	88	26	80	25	79	100
	Low	95	81	16	63	16	83	95
S2	High	100	100	100	99	77	100	100
	Fair	100	100	93	77	32	93	95
	Low	97	99	71	54	19	83	87
S3	High	100 (100)	90 (100)	24 (100)	57 (98)	9 (73)	86 (100)	100 (100)
	Fair	66 (100)	72 (99)	3 (63)	1 (60)	0 (22)	63 (94)	88 (99)
	Low	35 (96)	43 (92)	1 (46)	0 (33)	0 (4)	49 (93)	70 (98)
S4	High	19 (100)	22 (100)	1 (100)	0 (99)	0(54)	8 (100)	57 (99)
	Fair	9 (100)	14 (100)	2 (62)	0 (62)	0 (11)	16 (98)	84 (84)
	Low	8 (92)	8 (98)	0 (32)	0 (36)	0 (7)	5 (94)	20 (66)
	(c) **False positive results:** average number of noise variables per simulation (SD)							
Null[2]		0.03 (0.18)	13.10 (0.67)	0.01 (0.07)	0.035 (0.210)	0.005 (0.071)	3.96 (2.64)	0.92 (5.15)
S1	High	0.02 (0.14)	4.06 (2.18)	0.46 (0.63)	0.79 (0.409)	0.22 (0.416)	4.50 (3.11)	22.45(21.62)
	Fair	0.11 (0.35)	1.78 (1.51)	0.15 (0.46)	0.43 (0.498)	0.06 (0.239)	4.76 (3.13)	31.46(43.69)
	Low	0.09 (0.32)	1.24 (1.20)	0.04 (0.20)	0.23 (0.423)	0.02 (0.141)	4.58 (3.30)	42.96 (55.03)
S2	High	0.06 (0.24)	0.56 (1.09)	0.00 (0.00)	0.13 (0.338)	0.02 (0.141)	5.77 (2.97)	24.04 (25.64)
	Fair	0.11 (0.31)	0.89 (1.16)	0.08 (0.34)	0.23 (0.489)	0.08 (0.273)	5.26 (2.87)	33.92 (37.04)
	Low	0.07 (0.26)	0.93 (1.37)	0.02 (0.14)	0.23 (0.566)	0.05 (0.261)	5.00 (2.82)	29.52 (42.46)
S3	High	0.05 (0.22)	0.54 (0.81)	0.01 (0.10)	0.29 (0.46)	0.03 (0.171)	2.49 (2.27)	18.79 (32.51)
	Fair	0.06 (0.24)	0.81 (1.04)	0.04 (0.20)	0.34 (0.536)	0.07 (0.256)	4.15 (2.76)	31.57 (43.18)
	Low	0.16 (0.40)	1.12 (1.23)	0.04 (0.20)	0.21 (0.498)	0.05 (0.261)	4.12 (2.98)	27.61 (37.27)
S4	High	0.08 (0.31)	0.84 (1.14)	0.03 (0.22)	0.17 (0.378)	0.03 (0.171)	5.60 (2.85)	26.24 (24.94)
	Fair	0.11 (0.31)	1.04 (1.45)	0.14 (1.51)	0.18 (0.411)	0.03 (0.171)	4.30 (2.52)	19.56 (29.58)
	Low	0.09 (0.29)	0.82 (1.26)	0.02 (0.14)	0.09 (0.038)	0.02 (0.2)	4.33 (2.49)	19.21 (33.25)

[1] In Scenarios S1 and S2, the table gives the total number of times the true single variable/surrogate variable is selected. In Scenario S3, the table gives the total number of two true variables selected and the number of times at least one of two true variables selected in the bracket. In Scenario S4, the table gives the number of times two true/surrogate variables are selected (perfect selection) and the number of times the phylum Firmicutes is selected in brackets.

[2] The null Simulation is over 200 batches; all other scenarios are over 100 batches.

the high SNR cases, but clearly drops off as SNR decreases.) There are only four simulations where there is a significant difference in the in-sample prediction between SuRF and Stability with these settings: in Scenario S1 with fair and low SNR and Scenario S3 with high SNR, SuRF performs significantly better, while in Scenario S3 with low SNR, Stability performs significantly better. These differences are borne out in the variable selection results with SuRF selecting both more true positives and fewer false positives for the cases where its in-sample prediction is significantly better. For the case where Stability has lower in-sample misclassification error, Stability selection more frequently chooses both true predictors than SuRF. It also chooses more noise variables; however it is possible that these noise variables are partial surrogates of the true predictor, and might actually enhance predictive accuracy. Selection of surrogate variables can be more advantageous for in-sample prediction because the level of surrogacy is fixed, so the usefulness of the surrogate variable is retained for the prediction. When we want to generalise results to new test data, it is possible that the surrogacy occurs only by random chance and does not hold for new test data, in which case the predictive ability of a surrogate variable will decline for new test data. We will look at this effect in more detail in Simulation 3 where there are sufficient data to hold out a test data set.

Looking at the false positives, for a significance level $\alpha = 0.05$, the number of false positives selected after all true variables have been selected follows a geometric distribution with probability $1 - \alpha$, so the expected number of false positives is 0.0526. The number of false positives observed in Table 2.6(c) are mostly in line with this assumption, except for the low SNR case of Scenario S3. In cases with high false negative results, it is possible to select surrogates instead of the true variables. These would be recorded as false positives. This could explain the number of false positives in Scenario S3 for low SNR. In terms of true positives, only VSURF and Lasso, which select many more variables, select significantly more true variables than SuRF.

2.5.3 Study 3: Simulation using variables from lower taxonomic levels from the Moving Picture dataset

In this simulation, we study the performance of SuRF on a different microbiome dataset with more observations. We also simulate three true predictors at species level to observe the effect of having more predictors. Given that SuRF tends to select very sparse models,

we want to see whether sensitivity decreases as the number of true variables increases. We also simulate predictors at a lower taxonomic level. For the microbiome tree structure, it is conceivable that there could be some bias towards higher or lower taxonomic levels, so it is important to check that SuRF is able to identify the true variables in both cases. In real-world microbiome studies, classification using predictors from lower taxonomic levels such as genus or species has been acknowledged as more challenging [29]. The moving picture gut dataset[8] includes a total of 467 observations, with approximately 2000 OTUs. We divide the data into training and test sets, with the training data set containing the first 2/3 of the time points for each individual.

We perform two simulation studies, one with a binary response under a logistic linear model, and one with a continuous response following a Gaussian linear model with conditional variance 1. We set the coefficients in each simulation to achieve the desired total SNR which we set to one of three levels: High — 5, Fair — 3, and Low — 1. For the Gaussian response case, we set the irreducible error to 1 for all scenarios, so that MSE is comparable for different scenarios, and adjust signal strength to achieve the desired SNR. We compare MSE and R^2 on test data based on linear regression on the selected variables for each method. For the binary case, we compare misclassification rates.

We compare the same methods as in Simulation 2, namely SuRF with critical value $\alpha = 0.05$; Stability selection with FMER=1 and cut-off 0.6, 0.7, 0.8 and 0.9; Stability selection with FMER=0.0526 and cut-off 0.6, 0.7, 0.8, 0.9; VSURF; and Lasso. For the prediction, we also compare Random Forest and Support Vector Machine. These are popular machine learning methods that are able to achieve good predictive accuracy without performing variable selection. We compare the misclassification rates using both the test data, and the in-sample error on the training samples.

The results are shown in Table 2.7 (binary response) and Table 2.8 (continuous response).

For the binary simulation, SuRF has much lower misclassification error rate than other methods. This is achieved by selecting many more true predictors than other methods and fewer false positives than most methods. However, Stability selection achieves comparable in-sample misclassification error despite selecting fewer true predictors and more false positives. This suggests that the false positives include surrogate variables that contain most of the signal. However, we see that these surrogate variables do not generalise well, and the test misclassification error rate is much higher. This also suggests that the

Table 2.7: Simulation study 3 (Binary outcome)

SNR		**SuRF**	**Stability, FMER=1**		**Stability, FMER=0.0526**		**VSURF**	**Lasso**	**RF**	**SVM**
			0.6	0.9	0.6	0.9				
	(a) Mis-classfication error rate in test samples									
High	mean	0.102	0.207	0.295	0.383	0.412	0.288	0.290	0.292	0.197
	SD	(0.020)	(0.042)	(0.021)	(0.033)	(0.035)	(0.034)	(0.041)	(0.026)	(0.046)
Fair	mean	0.191	0.306	0.349	0.366	0.377	0.301	0.323	0.291	0.372
	SD	(0.028)	(0.030)	(0.013)	(0.034)	(0.030)	(0.041)	(0.038)	(0.032)	(0.080)
Low	mean	0.228	0.345	0.371	0.361	0.366	0.315	0.333	0.296	0.390
	SD	(0.037)	(0.030)	(0.013)	(0.032)	(0.035)	(0.047)	(0.033)	(0.032)	(0.084)
	(b) In-sample mis-classification error rate									
High	mean	0.100	0.176	0.259	0.228	0.295	0.126	0.169	0.126	0.128
	SD	(0.009)	(0.044)	(0.024)	(0.038)	(0.033)	(0.011)	(0.034)	(0.011)	(0.014)
Fair	mean	0.220	0.207	0.259	0.309	0.350	0.259	0.295	0.259	0.277
	SD	(0.010)	(0.037)	(0.021)	(0.030)	(0.010)	(0.017)	(0.029)	(0.017)	(0.024)
Low	mean	0.259	0.240	0.265	0.348	0.371	0.285	0.331	0.285	0.317
	SD	(0.017)	(0.032)	(0.019)	(0.031)	(0.012)	(0.020)	(0.034)	(0.020)	(0.032)
	No of true variables	**(c) Frequency of number of true variables selected**								
High	3	99	0	0	0	0	20	30		
	2	1	97	80	4	14	80	70		
	1	0	3	20	96	86	0	0		
	0	0	0	0	0	0	0	0		
Fair	3	82	0	0	0	0	21	5		
	2	18	52	23	3	3	78	90		
	1	0	48	77	97	97	1	5	N/A	
	0	0	0	0	0	0	0	0		
Low	3	71	0	0	0	0	9	4		
	2	24	27	9	2	1	76	76		
	1	5	73	91	97	97	15	20		
	0	0	0	0	1	2	0	0		
	(d) False positive results: number of noise variables per simulation									
High	mean	0.120	1.65	0.07	0.76	0.120	8.71	68.93		
	SD	(0.356)	(1.266)	(0.293)	(0.452)	(0.327)	(3.036)	(40.59)		
Fair	mean	0.690	1.61	0.07	0.340	0.02	7.080	62.45	N/A	
	SD	(0.895)	(1.325)	(0.256)	(0.476)	(0.141)	(3.183)	(46.98)		
Low	mean	0.610	0.160	0.79	0.01	0.010	6.590	61.92		
	SD	(0.764)	(0.935)	(0.1)	(0.368)	(0.327)	(0.100)	(55.24)		

Table 2.8: Simulation study 3 (Continuous outcome)

SNR		**SuRF**	**Stability, FMER=1**		**Stability, FMER=0.0526**		**VSURF**	**Best Subset**	**Lasso**	**RF**
			0.6	0.9	0.6	0.9				
	Oracle MSE	**(a) Median MSE (IQR) in test samples**								
High	1	1.009	3.943	4.589	4.433	4.555	2.781	1.955	3.470	2.977
		(0.131)	(0.708)	(0.671)	(0.672)	(0.678)	(0.379)	(1.671)	(0.794)	(0.416)
Fair	1	1.004	2.919	3.127	2.870	3.117	2.083	1.785	2.535	2.236
		(0.143)	(0.515)	(0.382)	(0.430)	(0.357)	(0.258)	(1.292)	(0.460)	(0.272)
Low	1	1.011	1.698	1.715	1.691	1.711	1.428	1.399	1.644	1.431
		(0.176)	(0.286)	(0.239)	(0.267)	(0.237)	(0.232)	(0.523)	(0.499)	(0.224)
	Oracle R^2	**(b) Average R^2 in test samples**								
High	0.833	0.801	0.28	0.126	0.367	0.356	0.460	0.524	0.324	0.417
		(0.098)	(0.035)	(0.018)	(0.044)	(0.046)	(0.048)	(0.189)	(0.109)	(0.038)
Fair	0.75	0.706	0.189	0.112	0.389	0.336	0.391	0.440	0.267	0.359
		(0.075)	(0.031)	(0.030)	(0.072)	(0.050)	(0.045)	(0.172)	(0.100)	(0.037)
Low	0.5	0.439	0.079	0.068	0.274	0.253	0.215	0.245	0.127	0.211
		(0.043)	(0.030)	(0.051)	(0.070)	(0.054)	(0.062)	(0.124)	(0.071)	(0.053)
	No of true variables	**(c) Frequency of Number of true variables selected**								
High	3	99	0	0	0	0	35	25	81	
	2	1	37	31	0	0	19	42	65	
	1	0	63	69	100	100	0	14	0	
	0	0	0	0	0	0	0	19	0	
Fair	3	97	0	0	0	0	71	31	12	
	2	3	32	23	0	2	19	31	81	N/A
	1	0	68	77	100	98	0	11	7	
	0	0	0	0	0	0	0	27	0	
Low	3	80	0	0	0	0	40	20	1	
	2	19	9	2	4	2	60	24	52	
	1	1	91	98	96	98	0	24	47	
	0	0	0	0	0	0	0	32	0	
	(d) False positive results: number of noise variables per simulation									
High	mean	0.040	2.7	0.08	0.360	0.080	15.16	3.27	52.540	
	SD	(0.197)	(1.592)	(0.273)	(0.503)	(0.273)	(4.334)	(1.043)	(29.186)	
Fair	mean	0.140	2.16	0.13	0.790	0.100	4.436	3.34	46.370	N/A
	SD	(0.377)	(1.475)	(0.367)	(0.686)	(0.302)	(2.106)	(1.193)	(30.833)	
Low	mean	0.540	0.86	0.04	0.700	0.010	14.330	3.65	28.700	
	SD	(0.784)	(1.092)	(0.197)	(0.916)	(0.100)	(5.650)	(1.167)	(16.407)	

comparable performance of SuRF and Stability selection in Simulation 2, in terms of in-sample misclassification error might not generalise to test data. While SuRF is often able to select the true predictors, the false positive rate is higher than the desired significance level, particularly in cases where the SNR is lower, causing SuRF to not select all true variables. The high false positive rate could include partial surrogates of true variables, which would explain why the average number of false positives is higher when the number of true positives is lower. However, it appears that the number of false positives is higher than the significance level, even in cases where SuRF selects all true variables. This is possible because false positives can enter the model before all true predictors have been selected.

For the Gaussian case, SuRF outperforms the other methods in terms of both MSE and R^2. This is achieved by selecting more true predictors, while selecting fewer false positives than most methods. Thus SuRF's variable selection performs well at selecting species level variables. The MSE and R^2 are close to the oracle values (i.e. the theoretical values under the true predictors and coefficients used to simulate the data). As for the binary outcome, the average number of false positives is larger than the significance level, particularly in cases with low SNR. This is partly caused by the selection of surrogate variables, and partly by the ranking part of the method not putting all true predictors first.

Within this section, we also look into assessing how the proportion of subsamples utilized in the SuRF algorithm influences both variable selection and prediction performance. Our recommendation leans toward employing roughly 90% of the data, particularly when dealing with significantly constrained sample sizes. However, our findings (See Table 2.9) suggest that, within the range of 50% to 90%, the selected proportion minimally impacts the outcomes in this study.

Table 2.9: Simulation results for changing the proportion of subsampling in Study 3 (fair SNR scenario for binary response — see Section 2.5.2 for details)

Proportion	Frequency of no. of true variables selected (sd)			Average no. of noise variables (sd)	Mean MCER (sd)
	3	**2**	**1**		
50%	80	19	1	0.77 (0.920)	0.191 (0.029)
60%	81	18	1	0.72 (0.889)	0.191 (0.030)
70%	81	18	1	0.74 (0.883)	0.191 (0.029)
80%	82	17	1	0.71 (0.913)	0.191 (0.028)
90%	82	18	0	0.69 (0.895)	0.191 (0.028)

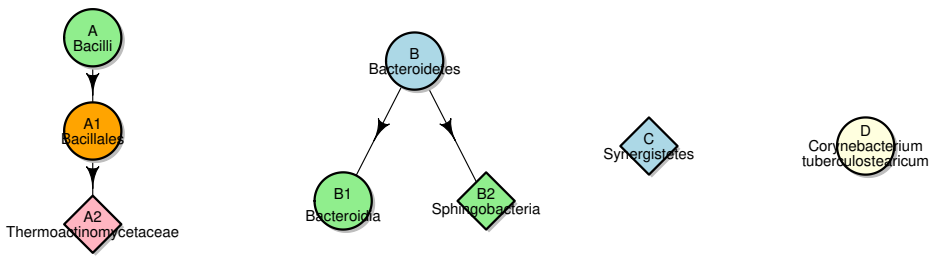
2.5.4 Study 4: Simulation with more true predictors

We also performed a more challenging simulation with eight true predictors, covering a range of taxonomic levels and rarities of taxa, also with different signal strengths for different taxa.

Design

This simulation provides a larger scale simulation to determine the performance of SuRF under different conditions. We base this simulation on the OTU counts from the Left Palm data of the moving picture data set. This data set includes over 12,000 OTUs, many more than in the previous simulations. Furthermore, in this simulation, we include 8 true variables, to assess the performance of SuRF in less sparse situations. We choose the true predictors to assess the influence of various factors on the ability of SuRF to select key predictors. In particular, we choose 8 predictors across a range of taxonomic levels, with some rarer taxa, and several different nesting patterns between the taxa. The nesting patterns and rareness are shown in Figure 2.3. We choose a chain of abundant variables from the class Bacilli; the phylum Bacteroidetes and two classes within it; the phylum Synergistetes; and the species *Corynebacterium tuberculostearicum*. All the variables used have at least one strong surrogate in the dataset (correlation at least 0.9), with the exception of A2 and B. Full details of the simulation are presented in Table 2.10 and Figure 2.3.

Figure 2.3: Variables used in the simulation. Circles represent abundant taxa, while diamonds represent rare taxa. Colours represent taxonomic level: blue — phylum; green — class; orange — order; pink — family; light yellow — species



We also study the effect of the coefficients on the taxa selected, simulating two different

sets of coefficients, one with larger coefficients for abundant taxa, and one with larger coefficients for rare taxa. For each set of coefficients, we simulated a high, medium and low signal-noise ratio. For each scenario, we simulated 100 datasets.

Table 2.10: Coefficients used in Simulation 4. Each coefficient is the product of the relative variable coefficient shown at the top for each scenario, and the factor for signal-noise ratio.

			variable coefficient							
	SNR	factor	A	A1	A2	B	B1	B2	C	D
			1	1.5	−0.5	1	2	−0.5	1	1
S1	High	1.72	1.72	2.58	−0.86	1.72	3.44	−0.86	1.72	1.72
	Fair	0.75	0.75	1.125	−0.375	0.75	1.5	−0.375	0.75	0.75
	Low	0.63	0.63	0.945	−0.315	0.63	1.26	−0.315	0.63	0.63
			1	1	2	1	1	2	2	1
S2	High	1.9	1.9	3.8	1.9	1.9	3.8	3.8	1.9	1.9
	Fair	0.794	0.794	1.588	0.794	0.794	1.588	1.588	0.794	0.794
	Low	0.613	0.613	1.226	0.613	0.613	1.226	1.226	0.613	0.613

Results

Table 2.11 and Table 2.12 summarise the results of SuRF, Stability selection and Lasso on these simulations.

This is a very challenging problem with slightly over 300 observations and over 12,000 variables. This explains why the results are generally worse than Simulation 2. We see that SuRF significantly outperforms the other methods in all scenarios in terms of misclassification error rate, and in terms of number of noise variables selected. SuRF does not achieve the target 0.0526 noise variables, but selects sparse models. SuRF also selects more true variables than Stability in all scenarios, and more than Lasso in many situations. Lasso selects a lot of variables, so would be expected to select more true variables by chance. As expected, the number of true variables selected increases as signal-noise ratio increases. With even higher SNR, or more data, we would expect SuRF to select all the true variables.

We now look at patterns among which true variables are selected. Table 2.11 shows the number of times each of the true variables was selected by each method in each scenario.

In the first scenario (S1), we see that as expected, variables with larger coefficients are selected more often. Even at high signal-noise ratio, Sphingobacteria was never selected, and *Thermoactinomycetaceae* was rarely selected. However, we also see that among variables

with the same coefficients, *Corynebacterium tuberculostearicum* was selected a reasonable proportion of times, while Synergistetes and Bacteroidetes were never selected. This indicates that it may be easier to select abundant predictors. In the case of Bacteroidetes, the effect is absorbed by the class Bacteroidia. In the class Bacilli, the class-level variable is often selected instead of the order Bacillales. The high correlation between the order-level and class-level variables makes distinguishing between them a challenging problem, and among the three methods, only SuRF is able to detect that both have a separate effect, when the signal noise ratio is high.

Lasso also shows similar patterns of favouring more abundant variables from among variables with the same coefficients. Stability selection mostly selected only Bacteroidia, so it is not possible to determine the extent to which it might favour more abundant taxa from the results on this scenario.

In Scenario 2, we see that even when coefficients are larger for rare taxa, the methods still have difficulty selecting these variables. This is an issue with penalised logistic regression, where the underlying distribution of the predictor variable can have a large effect on that variable's ability to be selected. This issue is studied in more detail in Chapter 3. This issue is even more significant for Lasso, which never selects Synergistetes, in spite of the large coefficient. Stability selection also shows some ability to select the abundant predictors Bacteroidetes and Bacteroidia, but cannot select the rare predictors Sphingobacteria or Synergistetes. It is able to select the rare predictor *Corynebacterium tuberculostearicum*. This is presumably related to the problem of selecting between correlated variables. When there are closely correlated variables, Stability selection will often select neither, while SuRF will usually select one, and can select both if there is evidence for separate effects from the two variables.

It is also interesting to look at the selection between surrogates. Many of the true variables in the simulation have strong surrogates in the data. Looking more carefully, we see that for this data set, SuRF is often able to distinguish between surrogates with correlation less than 0.95, whereas the other methods, when they select any variables, tend to have more difficulty determining which surrogate variable should be selected.

As expected, variables with larger coefficients are more easily selected. However, rarer taxa are selected less often, even when they have relatively high coefficients. These patterns are common to all variable selection methods. Across the range of signal-noise ratios

and coefficients, SuRF outperforms Stability selection (we only compare FMER=1 in this simulation, since it produced better results than FMER=0.0526 in previous simulations) in terms of both false positive and false negative rate. SuRF hugely outperforms Lasso in terms of false positive rate, and outperforms Lasso in false negative rate at high SNR, with comparable performance at lower SNR. In terms of misclassification error rate, summarised in Table 2.12, SuRF is clearly the best method. We did not compare VSURF or best subset selection in this simulation because of their slow running time.

Table 2.11: Frequency of selection for each variable over 100 simulations. The relative coefficients for each predictor are shown at the top of each scenario.

Scenario	SNR	Method	A	A1	A2	B	B1	B2	C	D
			1	**1.5**	**-0.5**	**1**	**2**	**-0.5**	**1**	**1**
S1	High	SURF	79	57	17	0	100	0	0	42
		Stability (0.6)	0	8	0	0	100	0	0	0
		Lasso	1	97	1	0	100	0	0	12
	Fair	SURF	68	32	5	0	100	0	0	17
		Stability (0.6)	8	1	0	0	100	0	0	2
		Lasso	28	77	1	0	100	0	0	22
	Low	SURF	68	26	1	0	100	0	0	4
		Stability (0.6)	7	3	0	0	100	0	0	0
		Lasso	26	71	0	0	100	1	0	16
			1	1	**2**	**1**	**1**	**2**	**2**	**1**
S2	High	SURF	14	100	100	100	1	38	46	38
		Stability (0.6)	0	0	100	9	14	0	0	0
		Lasso	0	0	100	88	46	93	0	2
	Fair	SURF	8	79	99	100	0	10	6	6
		Stability(0.6)	0	0	99	7	9	0	0	0
		Lasso	0	3	100	79	49	63	0	2
	Low	SURF	10	63	98	100	0	10	3	3
		Stability (0.6)	0	0	89	1	9	0	0	0
		Lasso	0	1	100	65	43	42	0	0

Table 2.12: Misclassification error rate(SD), average number of true variables selected (SD) and average number of noise variables selected (SD) from 8-variable simulations

Scenario	SNR	SuRF	Stability (0.6)	Lasso
(a) Misclassification Error Rate				
S1	High	0.174 (0.028)	0.235 (0.046)	0.218 (0.035)
	Fair	0.246 (0.028)	0.296 (0.039)	0.274 (0.036)
	Low	0.235 (0.023)	0.311 (0.041)	0.280 (0.036)
S2	High	0.143 (0.030)	0.255 (0.038)	0.258 (0.051)
	Fair	0.239 (0.027)	0.315 (0.028)	0.317 (0.047)
	Low	0.308 (0.027)	0.369 (0.040)	0.318 (0.037)
(b)Average number of true variables selected (SD)				
S1	High	2.95 (0.796)	1.08 (0.273)	2.11 (0.373)
	Fair	2.22 (0.645)	1.11 (0.345)	2.28 (0.697)
	Low	1.99(0.438)	1.10 (0.302)	2.14 (0.725)
S2	High	4.37 (1.390)	1.23 (0.446)	3.29 (0.518)
	Fair	3.08 (0.598)	1.15 (0.386)	2.96 (0.737)
	Low	2.86 (0.551)	0.99 (0.389)	2.51 (0.689)
(c) Average number of noise variables selected (SD)				
S1	High	0.83 (0.792)	5.46 (1.314)	34.28 (14.600)
	Fair	0.42 (0.699)	3.87 (1.353)	19.33 (10.440)
	Low	0.39 (0.618)	2.89 (1.214)	17.20 (8.837)
S2	High	1.40 (0.791)	2.86 (1.092)	32.17 (12.254)
	Fair	0.53 (0.688)	1.75 (0.845)	24.18 (11.832)
	Low	0.43 (0.607)	1.25 (0.857)	18.14 (11.889)

2.6 Application: the pouchitis and moving picture data

We analyse two published microbiome datasets using SuRF. The first dataset is from a pouchitis study [62]. The second dataset includes samples from four body sites of two individuals over a long time period [8].

2.6.1 Pouchitis study

Colectomy with ileal pouch anal anastomosis (IPAA), also referred to as "J-pouch surgery", is a common surgery for patients who have ulcerative colitis (UC) and those with familial adenomatous polyposis syndrome (FAP) [52]. Pouchitis is a common complication of J-pouch surgery involving inflammation of the ileal pouch. It is unclear what triggers pouchitis in some patients but not others: pouchitis occurs almost exclusively in patients with inflammatory bowel disease and not in patients with FAP.

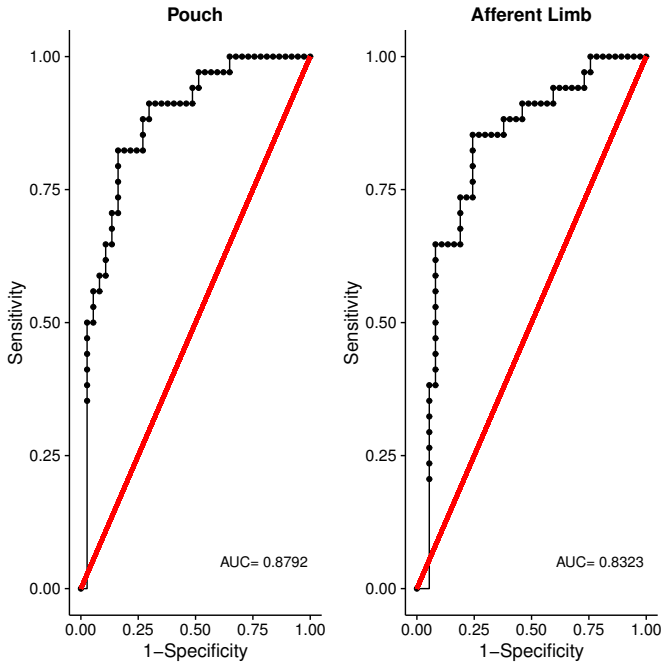
Our data come from the study Tyler *et al.* [62] which includes microbiome samples from biopsies of 71 patients following a J-pouch surgery. Our objective is to classify individuals between the healthy and inflammation group. The inflammation group is composed of the 34 subjects from the "pouchitis" and "CD (Crohn's disease)-like" groups in the original paper. It includes inflammation in either the pouch or the pre-pouch ileum; and the inflammation may or may not be active at time of biopsy. The healthy group is composed of the 37 subjects in the "FAP" and "no pouchitis" groups from the original study.

Some patients received one or two antibiotic treatments before the biopsy. We include two variables describing antibiotics usage in addition to the proportions of OTUs at each taxonomic rank, making a total of 1781 predictors. The same information was measured at both pouch and afferent limb for each patient.

The mean classification error rate is estimated by averaging the cross-validated classification error across a thousand subsamples. It is about 0.2 and 0.35 for pouch and afferent limb, respectively. At both biopsy sites, the phylum Bacteroidetes is the only variable significant at level 0.05. The agreement on the importance of Bacteroidetes at both biopsy sites suggests this phylum is significantly associated with inflammation. The single Bacteroidetes phylum gives a 0.88 and 0.83 AUC (area under the ROC curve shown in Figure 2.4) in the pouch and afferent limb respectively. The ROC curves suggest that Bacteroidetes is an effective discriminant variable for differentiating the inflammation condition at both biopsy

sites, especially for the pouch data. This is consistent with the literature where decreased abundance and diversity of Bacteroidetes in CD samples has been found, both in other datasets [13] and the same dataset using different methods [62].

Figure 2.4: ROC curve for Bacteroidetes as a predictor of inflammation



Even the non-significant highly ranked variables, see Table 2.13, are potentially interesting variables for future studies. Most of these organisms have been found to be associated with conditions related to pouchitis, such as IBD: for example, Fusobacteriaceae [46], Turicibacter [47], Bacilli and Erysipelotrichi [40], and *Subdoligranulum* [57].

Stability selection with FMER=1 also selects Bacteroidetes for cut-off probability 0.6 for both pouch and afferent limb, but selects no variables at higher cut-off probabilities 0.8 and 0.9. At cut-off probability 0.7, it selects Bacteroidetes for the pouch data, but selects no variables for the afferent limb data. In Table 2.15, we compare the predictive accuracy of the logistic regression model using the selected variable Bacteroidetes, with other commonly used classification methods for microbiome data, namely Random Forest (RF) and Support Vector Machine (SVM) with a linear kernel (we obtained similar results for other kernels and omitted them from the table). These predictive accuracies are computed using leave-one-out cross-validation with their corresponding tuning parameters chosen by cross-validation within the training data. The predictive accuracy from RF and SVM are comparable to the

Table 2.13: Top 10 variables selected by SuRF

(a) From the Pouch variables in the Pouchitis data set

Variable Name	**Taxonomy Level**	**Phylum**	**Frequency**	**LR Statistic**	***p*-value**	**Critical Value**
Bacteroidetes	Phylum	Bacteroidetes	923	**32.45**	0.000	**13.36**
Fusobacteriaceae	Family	Fusobacteria	232	5.96	0.995	13.76
unclassified	Order	Proteobacteria	225	2.46	1.000	
Turicibacter	Genus	Firmicutes	220	6.53	0.965	
Subdoligranulum	Genus	Firmicutes	179	5.61	0.995	
Bacteroidia	Class	Bacteroidetes	170	0.03	1.000	
Erysipelotrichi	Class	Firmicutes	151	3.92	1.000	
Bacilli	Class	Firmicutes	150	2.85	1.000	
Dialister	Genus	Firmicutes	141	11.03	0.315	
Granulicatella	Genus	Firmicutes	131	2.90	1.000	

(b) From the Afferent Limb variables in the Pouchitis data set

Variable Name	**Taxonomy Level**	**Phylum**	**Frequency**	**LR Statistic**	***p*-value**	**Critical Value**
Bacteroidetes	Phylum	Bacteroidetes	858	**24.53**	0.000	**12.75**
Bacteroidia	Class	Bacteroidetes	322	0.03	1.000	14.29
Erysipelotrichi	Class	Firmicutes	188	5.21	1.000	
Pasteurellales	Order	Proteobacteria	163	9.60	0.415	
Bacilli	Class	Firmicutes	149	11.18	0.180	
unclassified	Genus	Firmicutes	144	10.80	0.230	
Epsilonproteobacteria	Class	Proteobacteria	139	6.05	0.980	
Deinococcus-Thermus	Phylum	Deinococcus-Thermus	138	4.51	1.000	
Leuconostocaceae	Family	Firmicutes	138	5.56	0.995	
unclassified	Genus	Bacteroidetes	134	5.62	0.995	

results using SuRF since the mean test errors are all within one standard deviation.

2.6.2 Moving picture data

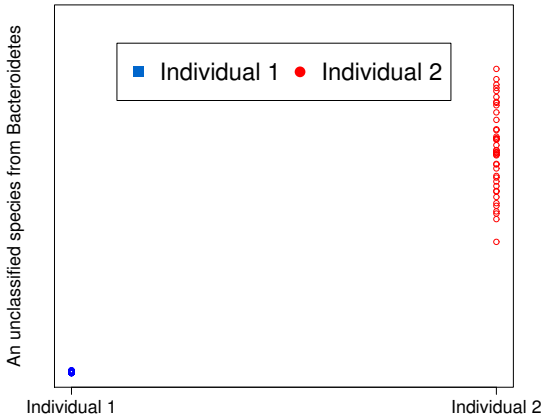
The moving picture data set [8] recorded a long period of repeated observations from multiple body sites (gut, tongue, left and right palms) of two individuals. This data set has a larger sample size for each body site than the pouchitis data. The number of observations for the gut, tongue, left palm and right palm are respectively 131, 135, 134 and 134 for the first individual, and 336, 373, 365 and 359 for the second individual. We split the dataset for each site into a training and a test sample set with a ratio of 2:1. At each body site, the observations from each individual are ordered by time. The earlier 2/3 of time points from each individual are used as training samples and the rest as test samples. Other than for division into training and test data, we make no use of the time metadata in this dataset.

We train SuRF to classify samples from each body site between the two individuals using the training data. The selected variables are summarised in Table 2.14, and the joint distributions are displayed in Figure 2.5. Table 2.15(b) shows the misclassification error rate for SuRF and other methods. Between one and four variables are selected at each body site and the prediction errors for the test samples are very low at all sites. SuRF has found a small set of variables that can distinguish two individuals' microbial environments. For most methods the test error tends to be lowest in the gut and highest for palms. This can be well explained by the fact that the microbiome community is most stable in the gut [63] and least stable for palms because, in contrast to the human gut, the composition of microbial communities from hands, though in the long run relatively stable [41] and personalised [19], can change dramatically even from washing hands with some disinfectant cleaning products. Identifying individuals using the palm microbiomes is feasible but more variable than using a more closed environment such as the gut.

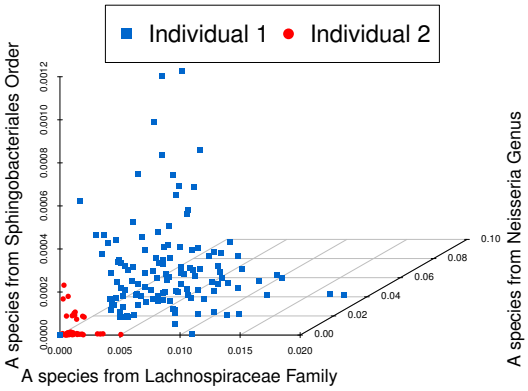
Table 2.14: Variables selected by SuRF for 4 body sites in the Moving pictures dataset

Site	Selected variable	Last identified level	Phylum	Critical value	LR	p-value
Gut	unclassified species	*Bacteroides* (Genus)	Bacteroidetes	17.77	371.21	0.00
Tongue	unclassified species	Lachnospiraceae (Family)	Firmicutes	14.90	309.45	0.00
	unclassified species	*Neisseria* (Genus)	Proteobacteria	18.13	43.18	0.00
	unclassified species	Sphingobacteriales (Order)	Firmicutes	26.12	28.67	0.01
Left Palm	unclassified species	*Deinococcus* (Genus)	Thermi	20.29	327.79	0.00
	Propionibacterium (Genus)	*Propionibacterium* (Genus)	Actinobacteria	17.28	56.97	0.00
Right Palm	unclassified species	*Corynebacterium* (Genus)	Actinobacteria	20.82	169.26	0.00
	unclassified species	*Deinococcus* (Genus)	Thermi	19.85	187.77	0.00

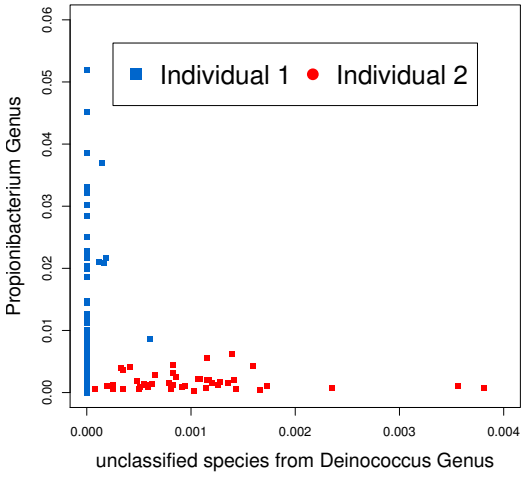
Figure 2.5: Prediction of test samples from gut, tongue, left and right palm



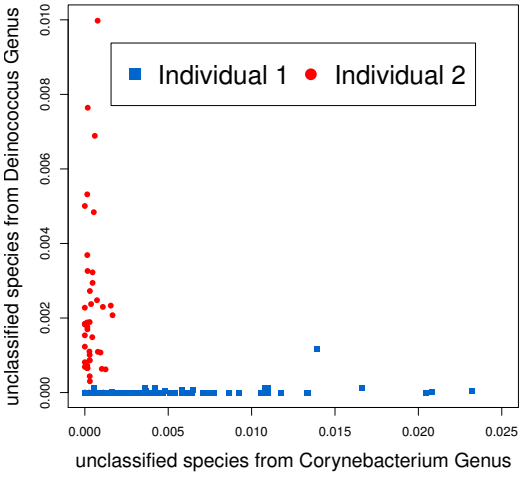
(a) Gut



(b) Tongue



(c) Left palm



(d) Right palm

Table 2.15: Results comparison among SuRF, Stability selection, VSURF, Lasso, Random Forest (RF) and SVM (Linear Kernel) for the pouchitis study and moving picture data

Data	SuRF	Stability Selection	VSURF	Lasso	RF	SVM
(a) Mean Misclassification error (sd)						
Pouch[1]	0.197 (0.047)	0.197 (0.047)	0.268 (0.053)	0.282 (0.053)	0.169 (0.044)	0.211 (0.048)
Afferent limb[1]	0.254 (0.052)	0.254 (0.052)	0.254 (0.052)	0.324 (0.056)	0.225 (0.050)	0.211 (0.048)
Gut	0.000	0.000	0.000	0.000	0.000	0.000
Tongue	0.053 (0.017)	0.000	0.018 (0.010)	0.000	0.006 (0.006)	0.024 (0.012)
Left Palm	0.024 (0.012)	0.030 (0.013)	0.061 (0.05319)	0.079 (0.021)	0.079 (0.021)	0.224 (0.032)
Right Palm	0.025 (0.012)	0.067 (0.020)	0.025 (0.012)	0.129 (0.026)	0.037 (0.015)	0.288 (0.035)
Left predict Right Palm	0.020 (0.006)	0.020 (0.006)	0.014 (0.005)	0.049 (0.010)	0.152 (0.016)	0.148 (0.016)
(b) The total number of variables selected						
Pouch[1]	**1 (0)**	**1 (0)**	4.282 (0.701)	1.254 (1.795)		
Afferent limb[1]	**1 (0)**	**1 (0)**	6.592 (0.729)	2.676 (12.033)		
Gut	**1**	3	**1**	18	N/A	N/A
Tongue	**3**	8	**3**	9		
Left Palm	**2**	**2**	3	67		
Right Palm	**2**	**2**	4	45		

[1] Misclassification error and mean number of variables selected with standard deviation are calculated based on leave-one-out for pouch and afferent limb.

We also tested cross-predictions — using models fitted on one body part to predict the owner of samples from another body part. The prediction model trained on one palm could identify samples from the other palm with low prediction error. The two bacteria selected in the two palm models include the same species-level variable from genus *Deinococcus* and two different unspecified species-level variables from genus *Corynebacterium*. This suggests a similarity between the microbiomes on two palms from a single individual. No other cross-predictions performed significantly better than random guessing.

SuRF and Stability selection (using cutoff probability 0.9 and default family error upper bound) were on average comparable in predictive accuracy to Random Forest and significantly better than SVM (see Table 2.15(b)). Compared to Stability selection, we found that SuRF seemed to achieve a lower prediction error, and consistently selected fewer variables.

In the gut data, SuRF chooses one unspecified species from the genus *Bacteroides*, which is one of three variables selected by Stability selection with cut-off probability 0.9. With one variable we obtain exactly the same prediction training and test errors as with three variables selected by Stability selection. The other two variables selected by Stability selection (another unclassified species from the genus *Bacteroides* and the family Porphyromonadaceae) didn't provide additional predictive accuracy for recognising individuals.

In the tongue data, even using cut-off probability 0.9, Stability selection still selects

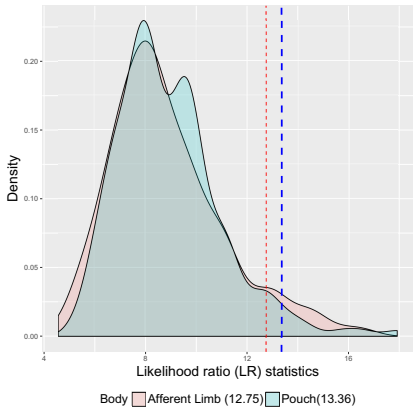
eight variables. SuRF selects only three variables: the most important variable is one species from genus *Neisseria* and the remaining two are unspecified species from the family Lachnospiraceae, and order Sphingobacteriales. There are no common variables selected by both SuRF and Stability selection. SuRF's misclassification error on this dataset is higher than other methods, so it is natural to ask whether SuRF might have selected too few variables in this case. However, using only the first two variables chosen by SuRF reduces the test error to 0.03, so the poor performance here is not entirely explained by excessive sparsity.

For the left palm data, both Stability selection with the highest cut-off probability and SuRF choose the same set of variables (one unspecified species from the genus *Corynebacterium* and another unspecified species from *Deinococcus*).

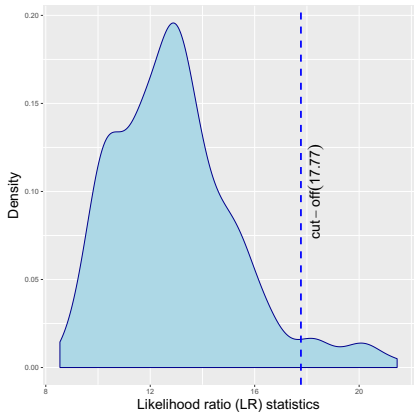
For the right palm data, SuRF selects the same species from the genus *Deinococcus* and a different unspecified species from the genus *Corynebacterium*. The former is also selected by Stability selection for the right palm model, but the second variable is replaced by the kingdom Bacteria. Both methods choose two variables (using cut-off 0.9 for Stability selection), however, SuRF not only provides a smaller prediction error for both training and test data, but also indicates a similarity between two palms within the individual which is not reflected by the variables selected by Stability selection.

These two real datasets exemplify the ability of SuRF to select discriminant OTUs at the appropriate taxonomic level. For the pouchitis data, with large within-class variation at lower levels, SuRF identifies a phylum-level variable. For comparing two healthy individuals, the higher-level structure is more similar, so SuRF selects species-level variables.

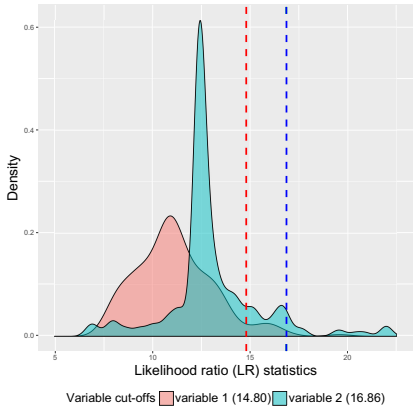
Figure 2.6: Permutation distributions of Likelihood Ratio (LR) in training samples from real data sets.



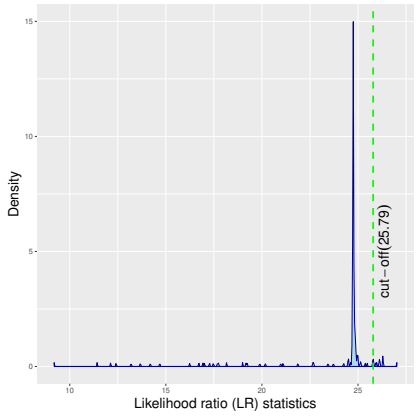
(a) Pouch and Afferent Limb



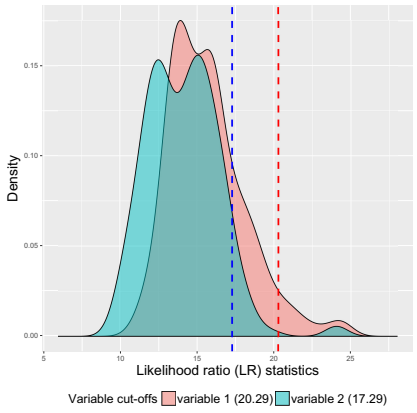
(b) Gut



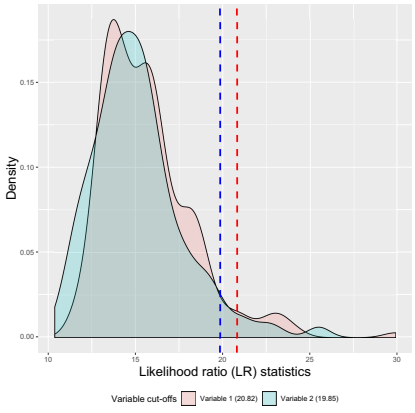
(c) Tongue: variables 1 and 2



(d) Tongue: third variable



(e) Left palm



(f) Right palm

2.7 Discussion on p-values

In this section, we study the reliability of the p-values given by SuRF. The p-value calculated by SuRF represents the p-value for the null hypothesis that all true predictors have already been included in the model. Because of the presence of surrogate variables, if there are true predictors that have not been included in the model, this null hypothesis may be rejected even if the next variable selected is not the "true predictor". Therefore, we can only assess the reliability of this p-value in the simulations by examining cases where all true variables have already been selected. Recall that SuRF is a forward selection method, so there is an order in which variables enter the model. If a noise variable enters the model before all true variables have been selected, then it is technically correct to reject the null hypothesis that all true variables have been selected.

If the probability of rejecting the null hypothesis is p, then the number of variables selected *after* all true variables have already been selected should follow a geometric distribution with parameter $1 - p$. The total number of noise variables selected across all simulations therefore should follow a negative binomial distribution with r the number of simulations in which all true variables are selected, and p the probability of incorrectly rejecting the null hypothesis. Thus if N is the number of cases in which we select all true predictors and X is the number of noise variables selected after all true variables, then the MLE estimate for p is $\frac{X}{X+N}$. If our p-values are well controlled, then we should have $p = 0.05$. We can test the significance of the number of noise variables selected with the null hypothesis $p = 0.05$ and the alternative hypothesis $p > 0.05$. The significance is given by

$$\sum_{x=X}^{\infty} \binom{x+N}{N} 0.05^x 0.95^N$$

We calculate the number of cases where all true variables are selected, and the number of noise variables that are selected after all true variables for Simulations 2 and 3 in Table 2.16. We cannot assess reliability for Simulation 4 because there were no cases where all true predictors were selected.

We see that the results are consistent with the p-values being correct. Only in one case, (Simulation 2, Scenario S2, fair SNR) is the number of noise variables selected after all true variables significantly more than would be expected for a reliable p-value. When accounting for the multiple testing from the 18 different scenarios, this is not significant.

Table 2.16: Testing the p-values

	Scenario	SNR	all true selected	noise variables	p-value	significant
Simulation 2	S1	High	100	2	0.020	0.964
		Fair	98	6	0.058	0.411
		Low	95	3	0.031	0.869
	S2	High	100	6	0.057	0.429
		Fair	100	11	0.099	0.022
		Low	97	4	0.040	0.742
	S3	High	100	5	0.048	0.599
		Fair	66	2	0.029	0.854
		Low	35	0	0	1.000
	S4	High	19	1	0.05	0.623
		Fair	9	1	0.1	0.370
		Low	8	0	0	1.000
Simulation 3	Binary response	High	71	1	0.014	0.974
		Fair	82	3	0.035	0.797
		Low	99	3	0.029	0.886
	Continuous response	High	99	1	0.01	0.994
		Fair	97	6	0.058	0.402
		Low	80	4	0.048	0.601

2.8 Extension of SuRF to Survival model applications: SuRFCox

SuRF is directly applicable to exponential family GLM models. In this chapter, the simulation and data analysis primarily focused on Gaussian and Binomial models, while the performance of the Poisson model will be discussed in Chapter 3.

In a collaborative research paper ([3]) not included in this thesis, the author has developed an extension, SuRFCox, which applies to the Cox-proportional hazards model for survival analysis. This work has been applied for predicting the microbial community transitions for environmental DNA data. In the ranking step, Lasso with family 'cox' is obtained using the `R` package `glmnet`. Due to the presence of censored data, it is challenging to maximise the full log-likelihood function in the CoxPh model. Instead, the partial likelihood estimation is used for this type of data. The ANOVA implemented for the permutation test in SuRFCox algorithm is through the log of partial likelihood as well.

The results from SuRFCox were compared with three other methods including Best Subset Selection [65] (BESS), Survival and Stationary Distribution in a Sure Independence Screening [17, 50] (SIS) and BeSS+SuRFBCox. SuRFCox has shown a great performance

in recovering the true variables and achieved a lower mean squared error (MSE) of the differences between estimated and true survival probabilities across various SNRs in the simulation.

2.9 Concluding Remarks

We have developed a very useful variable selection method for GLMs, SuRF, which involves a subsampling based approach to rank variables that may be highly associated with the response variable followed by variable selection with forward ANOVA. This method takes advantage of the sparseness of the model selected by Lasso and chooses variables that appear more frequently and contribute significantly to reducing residual deviances in the forward ANOVA procedure. Due to its high sparseness and stability, SuRF can be particularly useful for microbiome data or any data that is high dimensional and contains many surrogate variables. The method provides a conservative but stable selection of variables that can predict and classify the outcomes. SuRF can also provide a reasonable way to compute p-values for all variables according to sequentially calculated empirical distributions, whereas Lasso does not provide p-values directly. The forward selection procedure helps to alleviate the phenomenon of including surrogate variables and leads to a highly sparse model. Due to its short list of selected variables, SuRF is particularly suitable for identifying biomarkers.

In our simulation studies we saw that in comparison to many competing methods, SuRF is very competitive at selecting predictors, and typically selects sparser models than other methods, without a loss in predictive ability. A more in-depth simulation study of the performance of SuRF in the classical GLM case is warranted to fully determine the advantages and disadvantages of SuRF, compared to other methods. However, the focus of the current chapter is on the application of SuRF to microbiome data. For the simulations based on microbiome data, the results of SuRF were significantly better than other methods. It is unclear why SuRF shows a clear particular advantage for microbiome data. It could be because of the high correlations between variables, particularly with the aggregation approach that we used with all variable selection methods for the microbiome data. Another possible reason for SuRF's performance in this case might be the marginal distributions of the predictors. Microbiome data tends to be long-tailed and skewed. The distribution of the predictors can influence variable selection methods, and it is possible that SuRF is less affected by this than other methods. The question of how the marginal

distribution of predictors affects variable selection is studied in more detail in Chapter 3.

In the two real data analyses, we found that no other methods significantly outperformed SuRF in terms of prediction error, but SuRF selects fewer variables than other methods. For identifying biomarkers, selecting a smaller set of biomarkers with the same predictive power is valuable, because it allows the development of cheaper tests. We also observe that SuRF was able to adjust the taxonomic levels of the variables selected to suit the individual datasets.

There are many promising avenues for future research into extending the SuRF framework. In this chapter, we have presented SuRF based on penalised regression followed by generalised linear models, because that seemed most appropriate to the structure of the microbiome data. However, the core idea is to use subsampling with a simple variable selection method, then use the ensuing ranking in a forward selection method. This core idea could be applied with any combination of a variable selection method and a family of nested models to be used in forward selection. For example, we could develop a ranking based on Random Forest, and then perform the forward selection based on neural networks. The use of the permutation test for evaluating a variable automatically adjusts to our choice of method. Further research is needed into what combinations of methods work well in this framework.

2.10 Appendix

Simulation and analysis results with more detailed are provided in this section:

Table 2.17: Average Numbers of true and false positives for variable selection methods under Gaussian error model (complete results).

(a) **True positive results**

p	SNR	Measure	SURF				Stability, FMER=1				Stability, FMER=0.0526				Best	Lasso
			0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	Subset	
1	High	Mean	1	1	1	1	1	1	1	1	1	1	1	1	1	1
		SD	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	Fair	Mean	0.99	0.99	0.99	0.99	1	1	1	1	1	1	1	1	0.99	1
		SD	0.1	0.1	0.1	0.1	0	0	0	0	0	0	0	0	0.1	0
	Low	Mean	0.99	0.99	0.99	0.98	1	0.99	0.99	0.97	0.99	0.99	0.99	0.97	0.99	1
		SD	0.1	0.1	0.1	0.141	0	0.1	0.1	0.171	0.1	0.1	0.1	0.171	0.1	0
3(a)	High	Mean	2.09	2.2	2.26	2.28	2.95	2.88	2.81	2.57	2.89	2.86	2.76	2.62	1.09	3
		SD	0.588	0.569	0.562	0.587	0.219	0.327	0.443	0.590	0.314	0.349	0.495	0.582	0.321	0
	Fair	Mean	1.3	1.39	1.42	1.48	2.31	1.94	1.38	0.88	1.99	1.8	1.41	0.97	0.31	2.7
		SD	0.560	0.567	0.535	0.522	0.662	0.750	0.599	0.608	0.785	0.778	0.637	0.540	0.465	0.461
	Low	Mean	1.12	1.1	1.12	1.13	1.77	1.46	1.03	0.5	1.48	1.27	1.00	0.66	0.2	2.45
		SD	0.383	0.414	0.456	0.442	0.679	0.642	0.611	0.522	0.717	0.679	0.586	0.536	0.402	0.609
3(b)	High	Mean	3	2.97	2.98	3	2.97	2.9	2.73	2.1	2.02	2.28	2.13	1.82	2.88	3
		SD	0	0.223	0.2	0	0.171	0.302	0.468	0.704	0.710	0.683	0.747	0.744	0.383	0
	Fair	Mean	1.64	1.76	1.81	1.89	1.37	1.11	0.74	0.39	0.65	0.58	0.46	0.29	2.01	2.64
		SD	1.106	1.016	1.012	1.004	0.761	0.695	0.645	0.530	0.592	0.638	0.576	0.478	0.959	0.560
	Low	Mean	1.04	1.17	1.35	1.38	0.86	0.67	0.39	0.19	0.32	0.27	0.21	0.12	1.45	2.29
		SD	0.921	0.965	0.936	0.929	0.603	0.604	0.567	0.419	0.490	0.468	0.433	0.356	1.048	0.782
8	High	Mean	2.03	2.21	2.33	2.49	2.59	2.24	1.78	1.00	2.04	1.88	1.57	1.22	0.83	4.92
		SD	0.926	1.018	1.035	1.133	0.767	0.780	0.737	0.651	0.650	0.608	0.590	0.613	0.493	1.548
	Fair	Mean	0.96	1.01	1.04	1.07	1.43	1.13	0.59	0.22	1.16	1	0.67	0.29	0.69	3.8
		SD	0.665	0.703	0.737	0.756	0.782	0.747	0.605	0.462	0.762	0.667	0.604	0.498	0.506	1.378
	Low	Mean	0.8	0.82	0.85	0.88	1.14	0.77	0.34	0.12	0.77	0.59	0.41	0.19	0.69	3
		SD	0.471	0.520	0.539	0.590	0.711	0.694	0.536	0.327	0.649	0.588	0.534	0.394	0.526	1.407

(b) **False positive results** [1]

p	SNR	Measure	SURF				Stability, FMER=1				Stability, FMER=0.0526				MIO	Lasso
			0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9		
1	High	Mean	0.05	0.14	0.18	0.24	0.13	0.08	0.04	0	0.05	0.04	0.03	0.01	0.69	8.59
		SD	0.219	0.493	0.520	0.589	0.418	0.307	0.197	0	0.219	0.197	0.171	0.1	0.465	12.452
	Fair	Mean	0.06	0.1	0.18	0.27	0.23	0.11	0.03	0.02	0.08	0.05	0.03	0.01	0.3	8.29
		SD	0.278	0.362	0.458	0.529	0.468	0.345	0.171	0.141	0.273	0.219	0.171	0.1	0.503	10.909
	Low	Mean	0.06	0.11	0.22	0.36	0.19	0.09	0.06	0.02	0.09	0.07	0.05	0.02	0.26	11.28
		SD	0.239	0.345	0.504	0.732	0.465	0.288	0.239	0.141	0.288	0.256	0.219	0.141	0.463	13.141
3(a)	High	Mean	0.1	0.15	0.16	0.27	0.4	0.16	0.04	0	0.3	0.15	0.08	0.02	0.14	12.34
		SD	0.302	0.411	0.420	0.584	0.620	0.368	0.197	0	0.522	0.359	0.273	0.141	0.493	15.205
	Fair	Mean	0.19	0.26	0.32	0.44	0.5	0.22	0.06	0.01	0.38	0.21	0.15	0.04	1.1	12.09
		SD	0.443	0.485	0.530	0.656	0.643	0.440	0.239	0.1	0.582	0.409	0.359	0.197	1.010	12.63
	Low	Mean	0.14	0.18	0.29	0.37	0.44	0.19	0.06	0.01	0.27	0.19	0.11	0.02	1.05	12.96
		SD	0.377	0.411	0.574	0.661	0.671	0.419	0.239	0.1	0.529	0.394	0.314	0.141	0.796	14.427
3(b)	High	Mean	0.03	0.13	0.2	0.21	0.48	0.19	0.07	0	0.25	0.18	0.07	0.01	2.12	25.65
		SD	0.171	0.367	0.492	0.478	0.627	0.443	0.293	0	0.458	0.386	0.256	0.1	0.383	16.402
	Fair	Mean	0.36	0.53	0.61	0.75	0.34	0.15	0.07	0.02	0.2	0.11	0.05	0.02	2.99	19.61
		SD	0.560	0.703	0.737	0.833	0.536	0.359	0.256	0.141	0.426	0.314	0.219	0.141	0.959	17.466
	Low	Mean	0.4	0.54	0.59	0.72	0.29	0.09	0.01	0	0.1	0.06	0.03	0.01	3.55	17.22
		SD	0.586	0.673	0.753	0.805	0.478	0.288	0.1	0	0.302	0.239	0.171	0.1	1.048	18.387
8	High	Mean	0.25	0.37	0.49	0.57	0.92	0.51	0.14	0.05	0.82	0.59	0.26	0.08	0.31	16.33
		SD	0.5	0.580	0.703	0.820	0.706	0.541	0.349	0.219	0.672	0.605	0.441	0.273	0.748	16.703
	Fair	Mean	0.35	0.47	0.57	0.65	0.67	0.35	0.15	0.07	0.47	0.31	0.2	0.08	0.72	24.24
		SD	0.539	0.594	0.655	0.702	0.711	0.575	0.359	0.256	0.577	0.506	0.402	0.273	1.443	22.83
	Low	Mean	0.33	0.42	0.47	0.54	0.36	0.18	0.07	0.05	0.27	0.18	0.11	0.05	0.61	24.46
		SD	0.514	0.554	0.594	0.642	0.482	0.386	0.256	0.219	0.468	0.386	0.314	0.219	1.127	22.25

Table 2.18: Average Numbers and standard deviations of true and false positives for variable selection methods under logistic model (complete results).

(a) **True positive results**

p	SNR	Measure	SURF				Stability, FMER=1				Stability, FMER=0.0526				Lasso
			0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	
	High	Mean	1	1	1	1	1	1	1	1	1	1	1	1	1
1		SD	0	0	0	0	0	0	0	0	0	0	0	0	0
	Fair	Mean	0.99	0.98	0.97	0.99	1	1	1	0.98	1	1	1	0.97	1
		SD	0.1	0.141	0.171	0.1	0	0	0	0.141	0	0	0	0.171	0
	Low	Mean	0.98	0.98	0.98	0.98	1	0.99	0.97	0.91	0.99	0.99	0.96	0.94	1
		SD	0.141	0.141	0.141	0.141	0	0.1	0.171	0.288	0.1	0.1	0.167	0.239	0
	High	Mean	1.25	1.34	1.4	1.42	2.67	2.45	1.97	1.16	2.34	2.24	1.9	1.26	2.82
3(a)		SD	0.479	0.517	0.512	0.516	0.514	0.592	0.674	0.609	0.623	0.622	0.628	0.597	0.435
	Fair	Mean	1.03	1.05	1.07	1.07	1.79	1.48	1.1	0.57	1.53	1.37	1.04	0.67	2.15
		SD	0.437	0.458	0.477	0.477	0.671	0.627	0.541	0.573	0.658	0.614	0.511	0.533	0.744
	Low	Mean	0.91	0.94	0.94	0.94	1.39	1.02	0.74	0.33	1.11	0.98	0.69	0.43	1.81
		SD	0.379	0.397	0.397	0.397	0.695	0.635	0.543	0.473	0.618	0.586	0.526	0.498	0.692
	High	Mean	2.68	2.73	2.79	2.85	1.86	1.6	1	0.54	0.89	0.79	0.58	0.35	2.96
3(b)		SD	0.790	0.723	0.656	0.557	0.682	0.636	0.512	0.540	0.424	0.498	0.554	0.479	0.243
	Fair	Mean	1.24	1.38	1.43	1.54	1.02	0.77	0.44	0.16	0.42	0.32	0.24	0.11	2.56
		SD	1.016	1.08	1.047	1.077	0.635	0.617	0.556	0.368	0.496	0.469	0.429	0.314	0.641
	Low	Mean	0.71	0.86	0.9	0.99	0.75	0.51	0.33	0.1	0.31	0.27	0.17	0.09	1.82
		SD	0.782	0.841	0.870	0.916	0.642	0.611	0.514	0.302	0.506	0.489	0.378	0.288	0.903
	High	Mean	1.05	1.18	1.24	1.25	1.87	1.47	0.84	0.41	1.55	1.36	0.9	0.55	3.1
8		SD	0.833	0.845	0.866	0.892	0.597	0.643	0.662	0.534	0.672	0.689	0.704	0.609	1.259
	Fair	Mean	0.72	0.72	0.78	0.79	1.15	0.86	0.46	0.14	0.9	0.68	0.46	0.22	2.2
		SD	0.494	0.514	0.561	0.591	0.687	0.697	0.576	0.349	0.704	0.634	0.576	0.416	1.146
	Low	Mean	0.65	0.68	0.68	0.69	0.88	0.54	0.22	0.07	0.59	0.47	0.25	0.13	1.71
		SD	0.479	0.510	0.510	0.526	0.671	0.576	0.440	0.256	0.570	0.540	0.435	0.338	0.956

(a) **False positive results**

p	SNR	Measure	SURF				Stability, FMER=1				Stability, FMER=0.0526				Lasso
			0.05	0.1	0.15	0.2	0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9	
	High	Mean	0.04	0.08	0.17	0.22	0.15	0.02	0	0	0.03	0.01	0	0	2.53
1		SD	0.197	0.307	0.428	0.462	0.386	0.141	0	0	0.171	0.1	0	0	6.106
	Fair	Mean	0.06	0.12	0.15	0.27	0.18	0.09	0.02	0.01	0.09	0.06	0.04	0.01	4.38
		SD	0.239	0.356	0.386	0.529	0.386	0.288	0.141	0.1	0.288	0.239	0.197	0.1	8.702
	Low	Mean	0.08	0.13	0.2	0.22	0.17	0.08	0.02	0	0.07	0.04	0.02	0	4.64
		SD	0.307	0.393	0.449	0.462	0.428	0.273	0.141	0	0.293	0.197	0.141	0	9.070
	High	Mean	0.14	0.16	0.16	0.31	0.37	0.18	0.07	0.01	0.3	0.18	0.07	0.02	4.86
3(a)		SD	0.349	0.395	0.420	0.563	0.544	0.386	0.256	0.1	0.503	0.386	0.256	0.141	7.32
	Fair	Mean	0.15	0.21	0.24	0.28	0.39	0.13	0.04	0.01	0.2	0.14	0.08	0.04	7.62
		SD	0.359	0.433	0.474	0.494	0.584	0.338	0.197	0.1	0.402	0.349	0.273	0.197	13.373
	Low	Mean	0.17	0.21	0.31	0.42	0.26	0.11	0.02	0	0.12	0.08	0.03	0	2.73
		SD	0.378	0.456	0.581	0.713	0.505	0.314	0.141	0	0.327	0.273	0.171	0	4.144
	High	Mean	0.24	0.33	0.33	0.36	0.28	0.12	0.03	0.01	0.17	0.1	0.05	0.01	17.58
3(b)		SD	0.515	0.604	0.604	0.644	0.494	0.327	0.171	0.1	0.378	0.302	0.219	0.1	13.867
	Fair	Mean	0.43	0.58	0.66	0.76	0.38	0.19	0.07	0.01	0.15	0.1	0.06	0.02	18.17
		SD	0.573	0.669	0.685	0.726	0.546	0.394	0.256	0.1	0.359	0.302	0.239	0.141	17.38
	Low	Mean	0.47	0.52	0.57	0.71	0.33	0.09	0.03	0	0.12	0.04	0.02	0	17.2
		SD	0.627	0.627	0.624	0.729	0.570	0.321	0.171	0	0.327	0.197	0.141	0	20.424
	High	Mean	0.43	0.47	0.53	0.72	0.81	0.42	0.2	0.06	0.68	0.47	0.27	0.09	10.42
8		SD	0.573	0.627	0.643	0.780	0.631	0.572	0.426	0.239	0.566	0.559	0.468	0.288	13.566
	Fair	Mean	0.36	0.42	0.52	0.53	0.43	0.29	0.14	0.04	0.37	0.27	0.15	0.05	11.31
		SD	0.503	0.554	0.643	0.643	0.573	0.498	0.349	0.197	0.506	0.468	0.359	0.219	16.954
	Low	Mean	0.36	0.46	0.46	0.52	0.44	0.15	0.08	0.02	0.31	0.17	0.08	0.03	9.36
		SD	0.482	0.576	0.576	0.611	0.592	0.359	0.273	0.141	0.506	0.403	0.273	0.171	15.405

Table 2.19: Simulation study 2 complete results

Scenario	SNR	SuRF	Stability, FMER=1				Stability, FMER=0.0526				VSURF	Lasso
			0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9		
(a) **In-sample mean misclassification error rate (SD)** over 100 simulations												
S1	High	0.095 (0.011)	0.103 (0.044)	0.115 (0.076)	0.152 (0.130)	0.252 (0.194)	0.119 (0.069)	0.167 (0.141)	0.214 (0.175)	0.348 (0.191)	0.108 (0.031)	0.126 (0.062)
	Fair	0.190 (0.019)	0.219 (0.080)	0.264 (0.130)	0.339 (0.157)	0.416 (0.141)	0.243 (0.122)	0.284 (0.146)	0.343 (0.158)	0.421 (0.138)	0.240 (0.048)	0.365 (0.142)
	Low	0.240 (0.082)	0.274 (0.101)	0.311 (0.127)	0.381 (0.139)	0.454 (0.107)	0.318 (0.141)	0.337 (0.143)	0.393 (0.141)	0.454 (0.107)	0.276 (0.027)	0.418 (0.120)
S2	High	0.093 (0.010)	0.095 (0.011)	0.094 (0.010)	0.092 (0.009)	0.092 (0.008)	0.214 (0.037)	0.213 (0.037)	0.229 (0.077)	0.275 (0.125)	0.122 (0.018)	0.224 (0.058)
	Fair	0.173 (0.016)	0.178 (0.017)	0.175 (0.016)	0.172 (0.014)	0.187 (0.074)	0.296 (0.098)	0.311 (0.115)	0.351 (0.123)	0.408 (0.125)	0.222 (0.023)	0.294 (0.104)
	Low	0.210 (0.020)	0.210 (0.020)	0.214 (0.069)	0.223 (0.088)	0.282 (0.142)	0.349 (0.110)	0.369 (0.115)	0.396 (0.117)	0.446 (0.099)	0.266 (0.024)	0.368 (0.144)
S3	High	0.102 (0.010)	0.115 (0.037)	0.127 (0.048)	0.151 (0.062)	0.196 (0.056)	0.191 (0.082)	0.187 (0.071)	0.238 (0.087)	0.316 (0.116)	0.124 (0.015)	0.228 (0.063)
	Fair	0.204 (0.080)	0.192 (0.026)	0.207 (0.029)	0.231 (0.068)	0.316 (0.133)	0.346 (0.108)	0.356 (0.113)	0.387 (0.114)	0.447 (0.094)	0.232(0.021)	0.311 (0.100)
	Low	0.262 (0.129)	0.232 (0.072)	0.251 (0.097)	0.282 (0.116)	0.365(0.139)	0.413 (0.116)	0.422 (0.119)	0.455 (0.094)	0.487 (0.059)	0.265 (0.026)	0.342 (0.127)
S4	High	0.136 (0.030)	0.139 (0.032)	0.145 (0.031)	0.147 (0.029)	0.152 (0.045)	0.233 (0.036)	0.240 (0.051)	0.246 (0.063)	0.352 (0.136)	0.117 (0.018)	0.204 (0.059)
	Fair	0.204 (0.016)	0.207 (0.012)	0.210 (0.032)	0.231 (0.080)	0.318 (0.147)	0.338 (0.116)	0.358 (0.122)	0.409 (0.120)	0.467 (0.086)	0.220 (0.024)	0.356 (0.160)
	Low	0.245 (0.077)	0.231 (0.055)	0.242 (0.076)	0.280 (0.114)	0.408 (0.129)	0.407 (0.112)	0.412 (0.111)	0.454 (0.092)	0.481 (0.064)	0.254 (0.025)	0.403 (0.152)
(b) **True positive results** over 100 simulations [2]												
S1	High	100	98	92	80	58	96	84	72	38	82	100
	Fair	98	88	69	48	26	80	67	48	25	79	100
	Low	95	81	65	39	16	63	56	37	16	83	95
S2	High	100	100	100	100	100	99	98	92	77	100	100
	Fair	100	100	100	100	93	77	71	53	32	93	95
	Low	97	99	96	91	71	54	49	33	19	83	87
S3	High	100 (100)	90 (100)	80 (100)	60 (100)	24 (100)	57 (98)	58 (99)	30 (95)	9 (73)	86 (100)	100 (100)
	Fair	66 (100)	72 (99)	40 (98)	19 (90)	3 (63)	1 (60)	2 (59)	1 (44)	0 (22)	63 (94)	88 (99)
	Low	35 (96)	43 (92)	25 (84)	9 (74)	1 (46)	0 (33)	1 (31)	0 (15)	0 (4)	49 (93)	70 (98)
S4	High	19 (100)	22 (100)	10 (100)	3 (100)	1 (100)	0 (99)	0 (97)	0 (95)	0 (54)	8 (100)	57 (99)
	Fair	9 (100)	14 (100)	11 (99)	8 (92)	2 (62)	0 (62)	0 (53)	0 (34)	0 (11)	16 (98)	84 (84)
	Low	8 (92)	8 (98)	4 (94)	0 (80)	0 (32)	0 (36)	0 (32)	0 (17)	0 (7)	5 (94)	20 (66)
(c) **Null Case: False positives** [1]												
Null		5	200	200	200	1	6	2	2	1	200	13
	mean	0.03	13.10	8.06	3.01	0.01	0.035	0.015	0.01	0.005	3.96	0.92
	(SD)	(0.18)	(0.67)	(0.56)	(0.27)	(0.07)	(0.210)	(0.158)	(0.100)	(0.071)	(2.64)	(5.15)
(d) **False positive results:** average number of noise variables (SD) over 100 simulations												
S1	High	0.02 (0.14)	4.06 (2.18)	2.88 (1.94)	1.66 (1.39)	0.46 (0.63)	0.79 (0.409)	0.71 (0.478)	0.48 (0.522)	0.22 (0.416)	4.50 (3.11)	22.45(21.62)
	Fair	0.11 (0.35)	1.78 (1.51)	1.17 (1.20)	0.56 (0.88)	0.15 (0.46)	0.43 (0.498)	0.38 (0.488)	0.21 (0.409)	0.06 (0.239)	4.76 (3.13)	31.46(43.69)
	Low	0.09 (0.32)	1.24 (1.20)	0.76 (0.87)	0.38 (0.60)	0.04 (0.20)	0.23 (0.423)	0.2 (0.402)	0.08 (0.273)	0.02 (0.141)	4.58 (3.30)	42.96 (55.03)
S2	High	0.06 (0.24)	0.56 (1.09)	0.24 (0.71)	0.06 (0.28)	0.00 (0.00)	0.13 (0.338)	0.2 (0.402)	0.12 (0.327)	0.02 (0.141)	5.77 (2.97)	24.04 (25.64)
	Fair	0.11 (0.31)	0.89 (1.16)	0.89 (1.16)	0.89 (1.16)	0.08 (0.34)	0.23 (0.489)	0.25 (0.539)	0.14 (0.377)	0.08 (0.273)	5.26 (2.87)	33.92 (37.04)
	Low	0.07 (0.26)	0.93 (1.37)	0.61 (1.05)	0.18 (0.48)	0.02 (0.14)	0.23 (0.566)	0.2 (0.512)	0.17 (0.451)	0.05 (0.261)	5.00 (2.82)	29.52 (42.46)
S3	High	0.05 (0.22)	0.54 (0.81)	0.31 (0.66)	0.06 (0.24)	0.01 (0.10)	0.29 (0.46)	0.32 (0.469)	0.18(0.386)	0.03 (0.171)	2.49 (2.27)	18.79 (32.51)
	Fair	0.06 (0.24)	0.81 (1.04)	0.45 (0.77)	0.24 (0.26)	0.04 (0.20)	0.34 (0.536)	0.32 (0.529)	0.19 (0.394)	0.07 (0.256)	4.15 (2.76)	31.57 (43.18)
	Low	0.16 (0.40)	1.12 (1.23)	1.23 (0.74)	0.29 (0.62)	0.04 (0.20)	0.21 (0.498)	0.18 (0.458)	0.1 (0.333)	0.05 (0.261)	4.12 (2.98)	27.61 (37.27)
S4	High	0.08 (0.31)	0.84 (1.14)	0.37 (0.80)	0.11 (0.51)	0.03 (0.22)	0.17 (0.378)	0.15 (0.359)	0.07 (0.256)	0.03 (0.171)	5.60 (2.85)	26.24 (24.94)
	Fair	0.11 (0.31)	1.04 (1.45)	1.09 (1.11)	0.33 (0.90)	0.14 (1.51)	0.18 (0.411)	0.21 (0.518)	0.07 (0.256)	0.03 (0.171)	4.30 (2.52)	19.56 (29.58)
	Low	0.09 (0.29)	0.82 (1.26)	0.41 (0.95)	0.15 (0.64)	0.02 (0.14)	0.09 (0.038)	0.11 (0.399)	0.06 (0.343)	0.02 (0.2)	4.33 (2.49)	19.21 (33.25)

[1] Simulation under Null case: the number of batches that any noise variables are selected, together with mean and standard deviation of number of noise variables over 200 batches.

[2] In scenarios S1 and S2, the table gives the total number of times the true single variable/surrogate variable is selected. In scenario 3, the table gives the total number of two true variables selected and the number of times at least one of two true variables selected in the bracket. In scenario 4, the tables gives the number of times two true/surrogate variables selected (perfect selection) and the number of times variables selected deemed correct selection in bracket.

Table 2.20: Simulation study 3 (Binary outcome) complete results

SNR	No of true variables selected	SuRF	Stability, FMER=1				Stability, FMER=0.0526				VSURF	Lasso	RF	SVM
			0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9				
(a) Mean misclassification error rate in test samples (sd)														
High	mean	0.102	0.207	0.247	0.274	0.295	0.383	0.382	0.399	0.412	0.288	0.290	0.292	0.197
	SD	(0.020)	(0.042)	(0.053)	(0.044)	(0.021)	(0.033)	(0.043)	(0.041)	(0.035)	(0.034)	(0.041)	(0.026)	(0.046)
Fair	mean	0.191	0.306	0.324	0.338	0.349	0.366	0.369	0.373	0.377	0.301	0.323	0.291	0.372
	SD	(0.028)	(0.030)	(0.032)	(0.024)	(0.013)	(0.034)	(0.033)	(0.033)	(0.030)	(0.041)	(0.038)	(0.032)	(0.080)
Low	mean	0.228	0.345	0.357	0.364	0.371	0.361	0.359	0.361	0.366	0.315	0.333	0.296	0.390
	SD	(0.038)	(0.030)	(0.028)	(0.021)	(0.013)	(0.037)	(0.032)	(0.031)	(0.035)	(0.047)	(0.033)	(0.032)	(0.084)
(b) In-sample mean misclassification error rate (sd)														
High	mean	0.100	0.176	0.214	0.239	0.259	0.228	0.249	0.276	0.295	0.126	0.169	0.126	0.128
	SD	(0.009)	(0.044)	(0.053)	(0.045)	(0.024)	(0.038)	(0.053)	(0.049)	(0.033)	(0.011)	(0.034)	(0.011)	(0.014)
Fair	mean	0.220	0.207	0.229	0.245	0.259	0.309	0.324	0.340	0.350	0.259	0.295	0.259	0.277
	SD	(0.010)	(0.037)	(0.040)	(0.033)	(0.021)	(0.030)	(0.031)	(0.024)	(0.010)	(0.017)	(0.029)	(0.017)	(0.024)
Low	mean	0.259	0.240	0.251	0.258	0.265	0.348	0.357	0.364	0.371	0.285	0.331	0.285	0.317
	SD	(0.017)	(0.032)	(0.030)	(0.027)	(0.019)	(0.031)	(0.029)	(0.022)	(0.012)	(0.020)	(0.034)	(0.020)	(0.032)
(c) Frequency of number of true variables selected over 100 simulations														
High	3	99	0	0	0	0	0	0	0	0	20	30		
	2	1	97	95	89	80	4	12	13	14	80	70		
	1	0	3	5	11	20	96	88	87	86	0	0		
	0	0	0	0	0	0	0	0	0	0	0	0		
Fair	3	82	0	0	0	0	0	0	0	0	21	5		
	2	18	52	50	39	23	3	7	5	3	78	90		
	1	0	48	50	61	77	97	93	95	97	1	5	N/A	
	0	0	0	0	0	0	0	0	0	0	0	0		
Low	3	71	0	0	0	0	0	0	0	0	9	4		
	2	24	27	24	19	9	2	5	4	1	76	76		
	1	5	73	76	81	91	97	95	96	97	15	20		
	0	0	0	0	0	0	1	0	0	2	0	0		
(d) Mean number of noise variables selected (sd)														
High	mean	0.120	1.65	0.8	0.3	0.07	0.76	0.71	0.37	0.120	8.71	68.93		
	SD	(0.356)	(1.266)	(0.943)	(0.541)	(0.293)	(0.452)	(0.556)	(0.544)	(0.327)	(3.036)	(40.59)		
Fair	mean	0.690	1.61	0.82	0.38	0.07	0.340	0.250	0.120	0.02	7.080	62.45	N/A	
	SD	(0.895)	(1.325)	(0.978)	(0.678)	(0.256)	(0.476)	(0.458)	(0.327)	(0.141)	(3.183)	(46.98)		
Low	mean	0.610	0.79	0.44	0.17	0.01	0.160	0.140	0.050	0.010	6.590	61.92		
	SD	(0.764)	(0.935)	(0.743)	(0.378)	(0.1)	(0.368)	(0.377)	(0.219)	(0.327)	(0.100)	(55.24)		

Table 2.21: Simulation study 3 (Continuous outcome)

SNR	No of true variables	SuRF	Stability, FMER=1				Stability, FMER=0.0526				VSURF	Best Subset	Lasso	RF
			0.6	0.7	0.8	0.9	0.6	0.7	0.8	0.9				
	Oracle MSE		**(a) Median MSE (IQR) in test samples**											
High	1	1.009	3.943	4.130	4.500	4.589	4.433	4.380	4.475	4.555	2.781	1.955	3.470	2.977
		(0.131)	(0.708)	(0.964)	(0.806)	(0.671)	(0.672)	(0.758)	(0.774)	(0.678)	(0.379)	(1.671)	(0.794)	(0.416)
Fair	1	1.004	2.919	3.029	3.130	3.127	2.870	2.917	3.069	3.117	2.083	1.785	2.535	2.236
		(0.143)	(0.515)	(0.499)	(0.368)	(0.382)	(0.430)	(0.454)	(0.496)	(0.357)	(0.258)	(1.292)	(0.460)	(0.272)
Low	1	1.011	1.698	1.698	1.708	1.715	1.691	1.695	1.708	1.711	1.428	1.399	1.644	1.431
		(0.176)	(0.286)	(0.255)	(0.228)	(0.239)	(0.267)	(0.251)	(0.255)	(0.237)	(0.232)	(0.523)	(0.499)	(0.224)
	Oracle R^2		**(b) Mean R^2 (sd) in test samples**											
High	0.833	0.801	0.28	0.222	0.148	0.126	0.367	0.373	0.365	0.356	0.460	0.524	0.324	0.417
		(0.010)	(0.098)	(0.104)	(0.067)	(0.035)	(0.044)	(0.058)	(0.057)	(0.046)	(0.048)	(0.189)	(0.109)	(0.038)
Fair	0.75	0.706	0.189	0.154	0.120	0.112	0.389	0.381	0.357	0.336	0.391	0.440	0.267	0.359
		(0.030)	(0.075)	(0.068)	(0.046)	(0.031)	(0.072)	(0.073)	(0.066)	(0.050)	(0.045)	(0.172)	(0.100)	(0.037)
Low	0.5	0.439	0.079	0.075	0.069	0.068	0.274	0.262	0.257	0.253	0.215	0.245	0.127	0.211
		(0.051)	(0.043)	(0.039)	(0.033)	(0.030)	(0.070)	(0.064)	(0.062)	(0.054)	(0.062)	(0.124)	(0.071)	(0.053)
	Oracle MSE		**(c) In-sample Median MSE (IQR)**											
High	1	1.043	3.555	3.791	4.965	4.971	4.964	4.966	4.968	4.971	1.655	2.683	1.723	1.698
		(0.010)	(0.364)	(0.522)	(0.788)	(0.018)	(0.816)	(0.216)	(0.016)	(0.015)	(0.092)	(1.255)	(0.100)	(0.099)
Fair	1	1.025	2.673	2.818	3.336	3.338	2.796	2.810	3.338	3.340	1.498	1.893	1.487	1.535
		(0.010)	(0.337)	(0.652)	(0.446)	(0.010)	(0.646)	(0.648)	(0.559)	(0.010)	(0.103)	(0.752)	(0.079)	(0.101)
Low	1	1.071	1.978	1.996	1.999	2.00	1.993	1.997	1.999	2.000	1.555	1.664	1.312	1.570
		(0.018)	(0.159)	(0.048)	(0.012)	(0.011)	(0.155)	(0.013)	(0.011)	(0.011)	(0.089)	(0.364)	(0.035)	(0.100)
	Oracle R^2		**(d) In-sample mean R^2 (sd)**											
High	0.833	0.841	0.460	0.404	0.314	0.272	0.553	0.547	0.536	0.525	0.759	0.525	0.773	0.755
		(0.002)	(0.053)	(0.081)	(0.075)	(0.034)	(0.054)	(0.058)	(0.052)	(0.039)	(0.012)	(0.119)	(0.026)	(0.012)
Fair	0.75	0.756	0.378	0.316	0.264	0.234	0.553	0.543	0.511	0.482	0.651	0.482	0.718	0.644
		(0.004)	(0.066)	(0.083)	(0.069)	(0.041)	(0.065)	(0.070)	(0.064)	(0.040)	(0.018)	(0.117)	(0.024)	(0.019)
Low	0.5	0.511	0.189	0.171	0.161	0.157	0.428	0.410	0.400	0.396	0.358	0.287	0.548	0.349
		(0.018)	(0.037)	(0.028)	(0.017)	(0.007)	(0.041)	(0.030)	(0.018)	(0.007)	(0.025)	(0.079)	(0.019)	(0.025)
			(e) Number of true variables selected											
High	3	99	0	0	0	0	0	0	0	0	81	25	35	
	2	1	37	48	47	31	0	0	0	0	19	42	65	
	1	0	63	52	53	69	100	100	100	100	0	14	0	
	0	0	0	0	0	0	0	0	0	0	0	19	0	
Fair	3	97	0	0	0	0	0	0	0	0	71	31	12	
	2	3	32	39	31	23	0	0	0	2	19	31	81	
	1	0	68	61	69	77	100	100	100	98	0	11	7	N/A
	0	0	0	0	0	0	0	0	0	0	0	27	0	
Low	3	80	0	0	0	0	0	0	0	0	40	20	1	
	2	19	9	8	5	2	4	4	4	2	60	24	52	
	1	1	91	92	95	98	96	96	96	98	0	24	47	
	0	0	0	0	0	0	0	0	0	0	0	32	0	
			(f)Mean number of noise variables selected (sd)											
High	mean	0.040	2.70	1.48	0.47	0.08	0.360	0.260	0.160	0.080	15.16	3.27	52.540	
	SD	(0.197)	(1.592)	(1.123)	(0.688)	(0.273)	(0.503)	(0.463)	(0.368)	(0.273)	(4.334)	(1.043)	(29.186)	
Fair	mean	0.140	2.16	1.04	0.42	0.13	0.790	0.680	0.370	0.100	4.436	3.34	46.370	N/A
	SD	(0.377)	(1.475)	(1.053)	(0.684)	(0.367)	(0.686)	(0.709)	(0.614)	(0.302)	(2.106)	(1.193)	(30.833)	
Low	mean	0.540	0.86	0.42	0.13	0.04	0.700	0.290	0.110	0.010	14.330	3.65	28.700	
	SD	(0.784)	(1.092)	(0.794)	(0.393)	(0.197)	(0.916)	(0.574)	(0.373)	(0.100)	(5.650)	(1.167)	(16.407)	

Table 2.22: Full results comparison among SuRF, Stability selection with FMER=1, VSURF, Lasso, Random Forest (RF) and SVM (Linear Kernel) for the pouchitis study and moving picture data

a) Pouchitis study (Leave-one-out prediction mean test error (sd))

Site	SuRF	Stability Selection	VSURF	Lasso	RF	SVM
Pouch	0.197 (0.047)	0.197 (0.047)	0.268 (0.053)	0.282 (0.053)	0.169 (0.044)	0.211 (0.048)
Afferent limb	0.254 (0.052)	0.254 (0.052)	0.254 (0.052)	0.324 (0.056)	0.225 (0.050)	0.211 (0.048)

b) Moving picture

Site	SuRF		Stability Selection			VSURF		Lasso		RF	SVM
	no. var	Test Error mean(sd)	Cut-off Probability	no. var	Test Error	no. var	Test Error mean(sd)	no. var	Test Error mean(sd)	Test Error mean(sd)	Test Error mean(sd)
Gut	*1*	0.000	0.6	*10*	0.000	*1*	0.000	*18*	0.000	0.000	0.000
			0.7	*10*	0.006 (0.006)						
			0.8	*6*	0.000						
			0.9	*3*	0.000						
Tongue	*3*	0.053 (0.017)	0.6	*14*	0.030 (0.013)	*3*	0.018 (0.010)	*9*	0.000	0.006 (0.006)	0.024 (0.012)
			0.7	*13*	0.053 (0.017)						
			0.8	*10*	0.018 (0.010)						
			0.9	*8*	0.000						
Left Palm	*2*	0.024 (0.012)	0.6	*5*	0.030 (0.013)	*3*	0.061 (0.019)	*67*	0.079 (0.021)	0.079 (0.021)	0.224 (0.032)
			0.7	*3*	0.042 (0.016)						
			0.8	*3*	0.042 (0.016)						
			0.9	*2*	0.030 (0.013)						
Right Palm	*2*	0.025 (0.012)	0.6	*7*	0.080 (0.021)	*4*	0.025 (0.012)	*45*	0.129 (0.026)	0.037 (0.015)	0.288 (0.035)
			0.7	*3*	0.061 (0.019)						
				3							
			0.8	*3*	0.061 (0.019)						
			0.9	*2*	0.067 (0.020)						
Left palm predicts right palm		0.020 (0.006)	0.6		0.034 (0.008)		0.014 (0.005)		0.049 (0.010)	0.152 (0.016)	0.148 (0.016)
			0.7		0.028 (0.007)						
			0.8		0.028 (0.007)						
			0.9		0.020 (0.006)						

Chapter 3

Effect of predictors' distributions on Lasso-based variable selection

3.1 Introduction

Lasso [59] has become one of the most popular variable selection methods, particularly in high-dimensional settings and for generalised linear models (GLMs). There has been a large amount of research on the performance of Lasso in the literature, both in terms of its theoretical properties and its practical performance. However, there has been almost no research into the effect of the marginal distributions of the predictor variables on the performance of Lasso-based variable selection methods. For some classification methods, such as quadratic discriminant analysis (QDA), the method works under the assumption that the predictors follow a multivariate Gaussian distribution. It was well known that the QDA classification results are sensitive to violations of this assumption [68]. In other studies about robustness to heavy-tailed distributions, the research has focused on heavy-tailed response variables (Fan et. al, 2014,[16]), and the approaches developed in that literature are not appropriate for handling heavy-tailed predictors. For very high-dimensional problems, it is often appropriate to precede the Lasso variable selection by a screening process such as correlation learning (Lv and Fan, 2008 and 2010, [17, 18]) or feature ranking. Delaigle and Hall (2012) [14] have shown that heavy-tailed predictors can adversely influence these screening methods and that correlation ranking based on student's t scores of predictors and a robust transformation using the median and interquartile range before the correlation learning can alleviate some impact of heavy-tailed predictors on the variable selection. The latter performs better when the dimension of predictors, p, is very large and distributions are extremely heavy-tailed. However, there is no literature about how much impact heavy-tailed predictors have on the variable selection based on the Lasso shrinkage method.

GLMs are widely used for regression and classification in real world problems, which often include a large number of predictors with heavy-tailed distributions. For example, in microbiome data where the predictors are the abundance of microbes in a sample, the marginal distributions of the majority of these abundances are heavy-tailed, resembling

a log-normal distribution. Variable selection, or biomarker identification, in microbiome data is very important. Not only can it improve the prediction accuracy for some problems, but it can also focus our attention on some key microbes, leading to better interpretation and enabling further study into the functions of these microbes. Lasso shrinkage based variable selection has been routinely used in both regression and classification problems in microbiome research, for example, identifying biomarkers for particular diseases [22], or for toxic Cyanobacterial blooms in lakes [51, 9]. Variable selection for these classification problems can be performed using logistic regression with a Lasso penalty. Lasso GLMs have also been applied to other real-world variable selection problems from areas such as genetics [20], medicine [72], ecology [27], biology [32] and finance [1], all of which can involve a wide variety of marginal distributions of predictor variables.

Typically, Lasso GLM minimises a loss function consisting of the negative log-likelihood of the model with an L^1 penalty on the coefficients. The relative penalties on different coefficients depend on the scale of the predictors. To make the method scale-invariant, it is a common practice to standardise all the predictors prior to applying Lasso, so that the variance of each predictor is one. While standardisation in terms of standard deviation is appropriate for normal predictors, the standard deviation is not always such a good measure of scale for heavy-tailed distributions. The lack of a more appropriate standardisation method for heavy-tailed predictors leads to worse performance of Lasso-based methods.

In this chapter, we perform extensive simulations to study the effect of different distributions of predictors on the performance of Lasso-based variable selection in GLMs, focusing on three different commonly-used GLMs: Gaussian linear regression with identity link function; Binomial logistic regression; and Poisson GLM with log link function; and a variety of predictor distributions. Four Lasso based variable selection methods: Lasso [59], Adaptive-Lasso [79], Stability Selection [39], and SuRF [36] are included for the comparisons. Furthermore, since the variable selection is well-established for Gaussian predictors, we also apply each of these methods after performing a Box-Cox transformation on each predictor. Since the data are simulated from the original predictors, this causes the model to be misspecified.

We find the marginal distributions of predictors have a limited but statistically significant effect on variable selection performance for Gaussian linear regression but can have a large effect on both logistic regression and Poisson regression. The biggest difference in

performance is between light-tailed and heavy-tailed predictors with heavy-tailed predictors selected less often for logistic regression, and selected more often than light-tailed predictors for Poisson regression. These effects are common for Lasso and for methods based on Lasso, such as Stability selection. For methods which combine Lasso with other variable selection approaches, such as SuRF, the effect of predictors' distributions is reduced, so that SuRF performs comparably with Stability when the true predictors are light-tailed, and better than Stability selection for heavy-tailed predictors.

This chapter is organised as follows. Section 3.2 presents the simulation designs for three different GLM models. A detailed comparison among different methods including Lasso, Adaptive-Lasso, Stability Selection (with per-family error rate $PERF = 1$ and $PERF = 0.0526$, called STAB1, STAB2, respectively), and SuRF for Gaussian linear regression, logistic regression and Poisson regression with log link models are given in Section 3.3. We also apply SuRF and Stability Selection to the Box-Cox transformed predictor variables to examine whether such transformation can help to improve the identification of the true variables, especially the heavy-tailed variables. Finally, we discuss all simulation results in Section 3.4.

3.2 Simulation design

Our objective is to study how the distributions of predictors influence the variable selection in three frequently used generalised linear models (GLMs), Binomial, Gaussian, and Poisson regression. For each model, we include two scenarios — only one true predictor and three true predictors. We are interested in the variable selection in cases where the number of predictor variables is much larger than the number of observations and where the predictor variables follow a variety of distributions, including both light-tailed and heavy-tailed distributions. In every scenario, we simulate 400 predictors from each of 11 distributions (listed in Table 3.1). For each data set, we simulate 100 observations, thus all data sets are of dimension 100×4400. For each scenario, 100 replicate data sets are simulated.

We use the following procedure to simulate 100 replicates of the predictor matrix X with correlated predictors following 11 different marginal distributions. We first simulate 100 replicate matrices, each of dimension 100×4400, from a multivariate standard normal distribution with a covariance matrix whose $(i, j)^{th}$ element is given by $\rho^{|i-j|}$ ($\rho = 0.8$), so that there is a high correlation among predictors in the adjacent columns and the correlation

diminishes as the separation between the two variables increases. Secondly, we randomly permute the columns of the data matrix, so that the correlation pattern varies between simulations. Finally, we perform a univariate transformation on each predictor so that it has the desired marginal distribution. The transformation is given by $\psi(X_{ij}) = F_j^{-1}(\Phi(X_{ij}))$ where F_j is the distribution function of the target distribution and Φ is the distribution function of the standard normal distribution. These 100 predictor matrices are fixed across different scenarios.

3.2.1 Simulations for different GLM models

In order to fairly compare the effects of variable selection on different distributions, we need to ensure that within each simulation scenario, true predictors with different distributions have comparable signal strengths. This is somewhat challenging, because there is not a clear definition of signal strength. The commonly used signal-noise-ratio (SNR) defined as the variance of the conditional mean over the mean of the conditional variance is a good measure of signal strength for Gaussian regression, but it is inappropriate for logistic regression. We use mutual information as a measure of signal strength for logistic regression. For Poisson regression, the signal-noise-ratio is an acceptable measure of signal strength. The details of how to control the signal strength in each type of model are given below.

Gaussian linear regression model

In linear regression models, the signal strength is most often expressed by the SNR and the coefficients can be determined explicitly. Generally for model $Y = f(X) + \epsilon$, the signal-noise-ratio is defined as $SNR = \frac{\text{Var}(f(X))}{\text{Var}(\epsilon)}$. We fix $\epsilon \sim N(0, 1)$. It is easy to calculate the regression coefficients so that the linear model $Y = X\beta + \epsilon$ has the target SNR level. When there is only one true predictor with variance 1, $\beta = \sqrt{SNR}$. In the multivariate cases where there are 3 true predictors, since the correlations between these three predictors are different for different data sets, we simply assign $\sqrt{\text{SNR}/3}$ to each non-zero coefficient after each predictor is standardised based on its empirical distribution so that approximately the same strength of signal comes from each predictor.

Binomial logistic regression model

Unlike linear regression, the variance of a Bernoulli variable in the logistic regression is a function of the mean, and there is no separate parameter for the variance. Furthermore, this variance is too small for observations with small or large probability of belonging to one class, so SNR is not a good measure of signal strength for logistic regression. For this reason, we use mutual information as the measure of signal strength. Mutual information measures the dependence between two random variables. It is defined as the Kullback Leibler divergence of the product of the marginal distributions $p(X)p(Y)$ from the joint distribution $p(X,Y)$, i.e. $MI(X,Y) = D_{KL}(p_{(X,Y)}||p_X p_Y) = E_{(X,Y)} \log \frac{p(X,Y)}{p(X)p(Y)}$. In the GLM setting, it is convenient to note that $\frac{p(X,Y)}{p(X)}$ is the conditional probability of Y given X.

In the logistic regression model with one predictor variable X, $Y|X$ has a Bernoulli distribution $Bern(p(X))$ where $p(X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$. If we let $P = E(p(X))$, then the marginal distribution of Y is $Bern(P)$. The mutual information between the random variables Y and X is

$$
\begin{aligned}
MI(X,Y) &= E_X E_{Y|X} \log\left(\frac{f_{Y|X}(Y)}{f_Y(Y)}\right) \\
&= E_X\left[p(X)\log\frac{p(X)}{P} + (1-p(X))\log\frac{1-p(X)}{1-P}\right] \\
&= E_X\left[p(X)\left(\log(p(X)) - \log(P)\right) + (1-p(X))\left(\log(1-p(X)) - \log(1-P)\right)\right] \\
&= E_X\left[\log(1-p(X)) - \log(1-P) + p(X)\left(\log\frac{p(X)}{1-p(X)} - \log\frac{P}{1-P}\right)\right] \\
&= E_X\left[-\log(1+e^{\beta_1 X+\beta_0}) - \log(1-P) + p(X)\left(\beta_1 X + \beta_0 - \log\frac{P}{1-P}\right)\right]
\end{aligned}
\tag{3.1}
$$

The mutual information can be very small if the quantity $P = E(p(X))$ is relatively close to 0 or 1. For our simulations, we fix $P = 1/2$. This is achieved by the following parameter values in the simulations with a single true predictor:

1. For the distributions that are symmetric about μ, setting $\beta_0 = -\beta_1\mu$ gives $P = 1/2$. Thus for the normal and t distributions, we set $\beta_0 = 0$. For Beta(α,α) distributions, we set $\beta_0 = -\beta_1/2$;

2. For the remaining distributions: log-normal, Pareto, Gamma and Poisson, the intercept β_0 needs to be solved numerically for each value of the coefficient β_1.

For the scenarios with one true predictor, we perform simulations at four signal strength levels: $MI = 0.05, 0.1, 0.2$ and 0.3. For each distribution of the true predictor, we use a grid search to find the coefficient β_1 that achieves the desired mutual information. These coefficients are summarised in Table 3.1.

For the scenarios in which there are three true predictor variables, it is difficult to set the coefficients to achieve a target level of mutual information, with each predictor contributing equally, because the mutual information between one predictor and the response depends on the coefficients of other predictors, and the marginal mutual information between each predictor and the response will be different from the conditional mutual information given the other predictors. We therefore fix the coefficients calculated from the single true predictor scenarios (in Table 3.1), and use these coefficients for the simulations. We drop the 0.05 mutual information level because selecting multiple variables with such weak signal is too challenging for all methods. For the coefficients from each mutual information level, $MI = 0.1, 0.2$ and 0.3, we simulate one scenario (100 replicates) for each set of three different distributions from 8 of the 11 distributions studied (excluding the gamma distribution with shape 2, the t distribution with 4 degrees of freedom, and the Pareto distribution with shape 3).

Although it is not possible to set the coefficients so that every true predictor has the desired mutual information with the response variable, it is possible to use Monte Carlo methods to calculate the total mutual information between the predictors and the response for a given set of coefficients. We do this by simulating a large number of samples from the joint distribution of the predictors, then calculating the conditional probability of the response variable, and thus the conditional entropy. This allows us to check that the simulation scenarios have comparable total signal strength. The mutual information values are shown in Figure 3.1.

The joint mutual information $\mathrm{MI}(Y, X)$ of each case is higher than the mutual information in the single true predictor case with the same coefficient. The cases in which the true variables are not long-tailed (Case 1–Case 10) generally have a slightly lower mutual information level, especially Case 7, highlighted in Figure 3.1. However, the values are fairly similar for all scenarios.

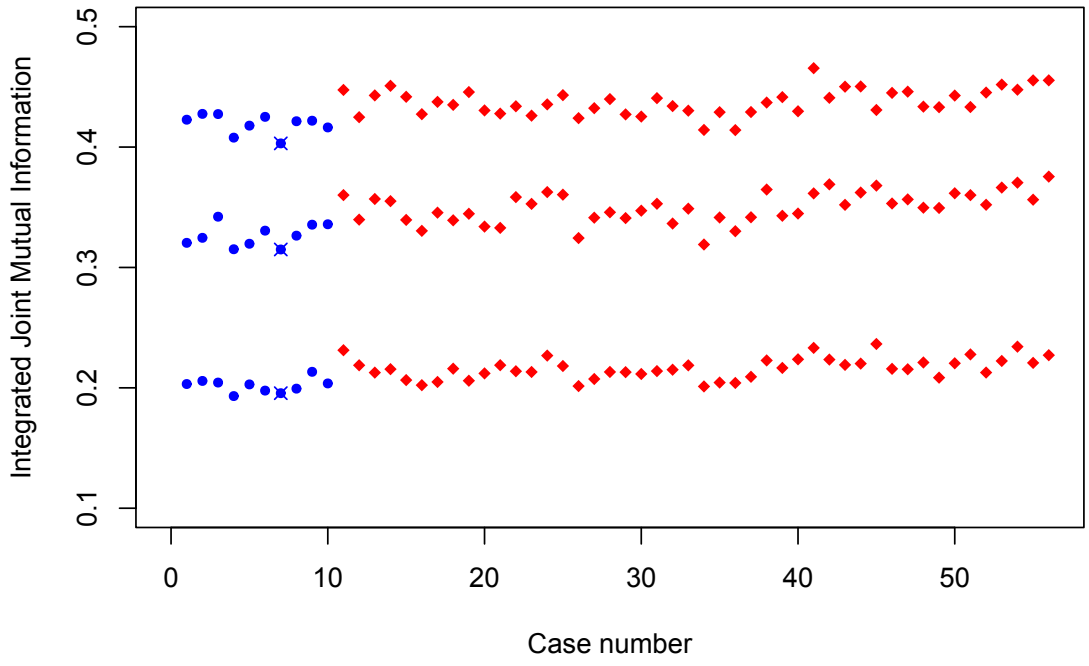


Figure 3.1: Integrated joint mutual information between response variable and predictors for Binomial model simulations
Circle for Cases 1-10 (no heavy-tailed true predictors); diamond for Cases 11-56 (at least one heavy-tailed true predictor). Case 7 (highlighted with a cross) shows a consistently lower integrated mutual information at all MI levels.

Table 3.1: Coefficients for predictors with different distributions in the single true predictor case for the logistic regression model for the given levels of mutual information

	MI=0.3		MI=0.2		MI=0.1	
Distribution	β_1	β_0	β_1	β_0	β_1	β_0
Normal(0, 1)	2.64	0	1.75	0	1.04	0
Poisson λ=2	1.88	−3.53	1.26	−2.40	0.75	−1.46
Log Normal (0, 2)*	3.12	−2.13	1.60	−1.25	0.67	−0.56
Beta (0.5, 0.5)	5.93	−2.97	4.28	−2.14	2.74	−1.37
Uniform (0, 1)	7.75	−3.88	5.44	−2.72	3.42	−1.71
t-dist (df=2)*	2.04	0	1.25	0	0.66	0
t-dist (df=4)*	2.32	0	1.49	0	0.84	0
$\text{Gamma}_{1,2}$ (shape=1, scale=2)	1.75	−2.74	1.11	−1.85	0.61	−1.10
$\text{Gamma}_{2,2}$ (shape=2, scale=2)	1.09	−3.86	0.71	−2.59	0.41	−1.55
Pareto_2(scale=1, shape=2.01)*	4.70	−2.38	2.76	−1.57	1.30	−0.87
Pareto_3 (scale=1, shape=3)*	8.00	−2.48	4.83	−1.65	2.41	−0.94

*a long-tailed distribution

Poisson count model

For the Poisson count model $Y_i \sim Poisson(\lambda_i)$ where $log(\lambda_i) = X_i^T\beta$, there is no good measure of signal strength, with SNR and mutual information both potentially influenced by outliers due to the log link function. We therefore standardise all the predictors to have variance 1 and fix all the coefficients for true predictors at one of the values $0.2, 0.3$, or 0.4 for low, medium or high signal respectively.

An important issue about Poisson regression on one hand is that when λ_i's are small, the vast majority of observed values are zero, which gives little information about the Poisson means, making variable selection particularly challenging. On the other hand, when λ_i is large, $\log(Y_i)$ can be reasonably well approximated by a normal distribution, making this simulation too similar to the Gaussian case. To avoid these extremes, we set the intercept β_0 such that the median of λ_i's (MRates) takes one of the values 0.8, 1 or 2, by using a grid search to find this coefficient. For these values, the discrete nature of the Poisson response still plays a role, but the response is not too sparse to invalidate all variable selection procedures.

3.2.2 Methods Compared

A few Lasso-based methods are selected to compare the impact of the predictors' distributions on variable selection using the simulated datasets. The methods are: Lasso [59], Adaptive-Lasso [79], Stability Selection [39], and SuRF [36].

Lasso minimises the GLM negative log-likelihood function plus an L^1 penalty on the coefficients. While this method has been widely used in various cases including very high dimensional predictors, it is also well known to produce biased estimates for large coefficients, and hence inconsistent selection (Fan and Lv, 2001,[18]).

Adaptive-Lasso is a two-step method which introduces a weight to each coefficient as given in (3.2) for Gaussian linear regression, in order to achieve stronger asymptotic oracle properties than Lasso. These oracle properties also hold true for GLM models in general.

$$L(\beta|X,y) = \frac{1}{2n}\|y - X\beta\|^2 + \lambda \sum_j w_j|\beta_j| \tag{3.2}$$

In more detail, the first step of Adaptive-Lasso is to obtain an initial estimate of the coefficients, $\hat{\beta}_j^{\text{init}}$, using another method, such as ordinary least squares, ridge regression, Lasso, etc.. Then the weight w_j for the jth coefficient β_j is adaptively selected to ensure the shrinkage is inversely proportional to the size of each initial estimate, i.e. $w_j = \frac{1}{|\hat{\beta}_j^{\text{init}}|}$. Compared to a constant Lasso penalty on all coefficients of different sizes, this secondary selection adjusts the penalty based on the initial estimate of the parameter to reduce the bias and improve the prediction. Huang *et al.* [30] and Lin, *et al.* [35] have proved the oracle properties of Adaptive-Lasso in the linear regression case in the sparse and/or high dimensional settings. Huang, *et al.* [30] assumes the error in the regression to have a Gaussian tail and Lin, *et al.* [35] relaxes the condition to any errors that have the finite $2k^{th}$ moments for an integer $k > 0$, such as a t distribution with sufficient degrees-of-freedom. Ridge regression is recommended by Zou [79] to obtain the initial estimate for Adaptive-Lasso for high dimensional data. We therefore use this approach for Adaptive-Lasso in our simulations.

Both Stability selection and SuRF are Lasso-based sub-sampling approaches. Stability selection performs Lasso variable selection on a large number of subsamples of the data set, and selects variables which are selected in a sufficiently large proportion of these subsamples. The cutoff for this selection is usually set in the range of $0.6 - 0.9$. Another tuning parameter for Stability selection is the per-family error rate (PERF), which is set

by controlling the tuning parameter λ in Lasso. In our simulations, we compare both the default setting for the per-family error control rate (PFER) of 1 and a more conservative setting at approximately 0.05. The theory underpinning Stability selection is based on a multivariate normal distribution of the predictors.

In the Binomial and Poisson regression models, we also compare the variable selection performance with an additional method where we perform a variant of the Box-Cox transformation on each predictor prior to applying Stability selection. We adopted the Yeo-Johnson transformation [71] for this purpose, because of its ability to handle zeros or negative values. This transformation destroys the linear relation between the predictors and the response, but makes heavy-tailed predictors closer to a normal distribution. The transformation is defined as

$$\phi_{y,\lambda} = \begin{cases} \frac{(y+1)^{\lambda}-1}{\lambda} & \text{if } y \geq 0, \lambda \neq 0 \\ \log(y+1) & \text{if } y \geq 0, \lambda = 0 \\ \frac{(-y+1)^{2-\lambda}-1}{2-\lambda} & \text{if } y < 0, \lambda \neq 2 \\ \log(-y+1) & \text{if } y < 0, \lambda = 2 \end{cases} \tag{3.3}$$

where the constant shift +1 in the formula keeps zero as a fixed point for the transformation. The parameter λ is estimated via maximum likelihood, under the assumption that the transformed distribution is normal.

SuRF is a two-stage method. The first step is to create a ranked list of all variables by performing Lasso variable selection on a large number of random subsamples of data points, similar to Stability selection. The critical difference between Stability selection and SuRF is at the final variable selection decision. SuRF uses this ranked list as the basis for a test-based forward selection procedure. We compare SuRF selection results with the significance level α ranging between 0.002 and 0.2 for the sequential tests. We also apply SuRF on the aforementioned Yeo-Johnson transformed predictors in the Binomial and Poisson regression models for variable selection as a comparison.

3.3 Simulation results

Our objective is to study how the distributions of predictors influence the variable selection in three frequently used generalised linear models (GLMs), Binomial, Gaussian, and Poisson

regression. For each model, we include two scenarios — only one true predictor and three true predictors. We are interested in the variable selection in cases where the number of predictor variables is much larger than the number of observations and where the predictor variables follow a variety of distributions, including both light-tailed and heavy-tailed distributions. In every scenario, we simulate 400 predictors from each of 11 distributions (listed in Table 3.1). For each data set, we simulate 100 observations, thus all data sets are of dimension 100×4400. For each scenario, 100 replicate data sets are simulated.

We use the following procedure to simulate 100 replicates of the predictor matrix X with correlated predictors following 11 different marginal distributions. We first simulate 100 replicate matrices, each of dimension 100×4400, from a multivariate standard normal distribution with a covariance matrix whose $(i, j)^{th}$ element is given by $\rho^{|i-j|}$ ($\rho = 0.8$), so that there is a high correlation among predictors in the adjacent columns and the correlation diminishes as the separation between the two variables increases. Secondly, we randomly permute the columns of the data matrix, so that the correlation pattern varies between simulations. Finally, we perform a univariate transformation on each predictor so that it has the desired marginal distribution. The transformation is given by $\psi(X_{ij}) = F_j^{-1}(\Phi(X_{ij}))$ where F_j is the distribution function of the target distribution and Φ is the distribution function of the standard normal distribution. These 100 predictor matrices are fixed across different scenarios.

We present the results in two scenarios for Gaussian, Binomial and Poisson models. In the first scenario, the underlying model is based on only one true variable generated from one of 11 distributions. Among all listed distributions, the log-normal, Pareto_2 (scale=1 and shape=2.01), t_2, t_4, and Pareto_3 (scale=1 and shape=3) are heavy-tailed. The tail densities of these distributions are shown in Figure 3.2. We will present the results of standard normal, $\text{Gamma}_{1,2}$ (scale=1, shape=2) , t_2, t_4 and Pareto_2 (scale=1 and shape=2.01) in this section, and all other scenarios in Sections A.1 and A.2.

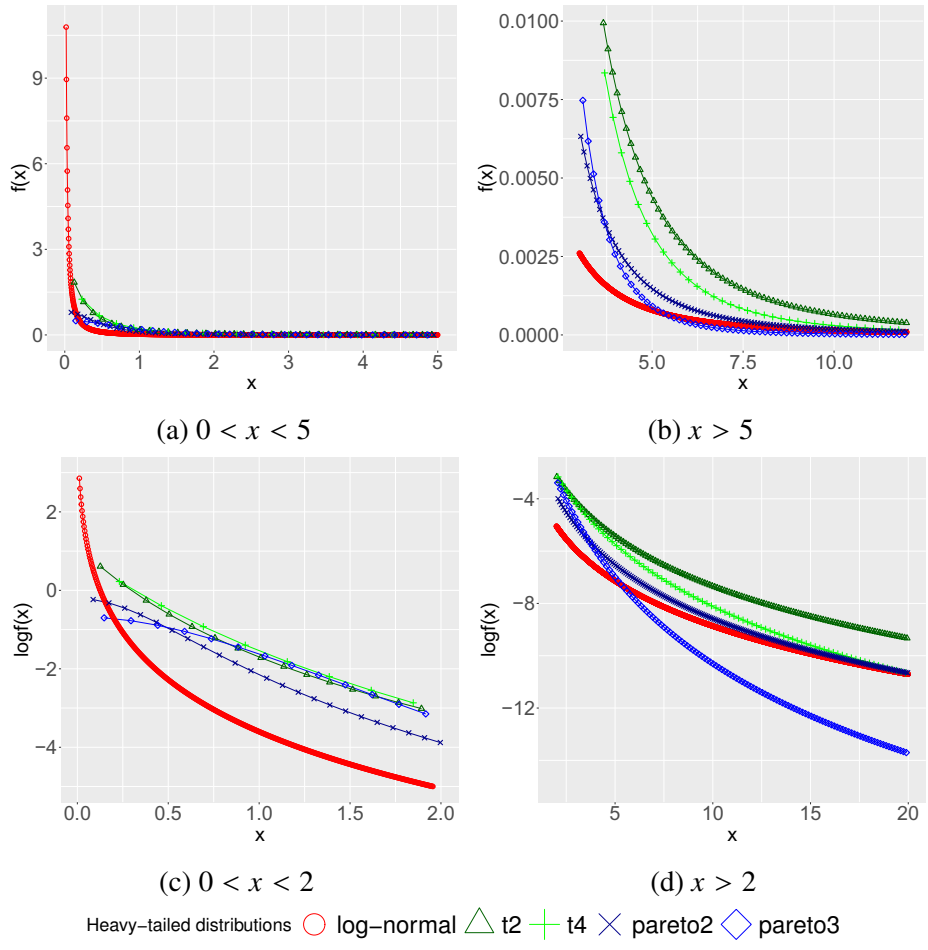
In the second scenario, the underlying model is based on three true variables with different distributions. We consider separately the cases in which the three true variables all follow light-tailed distributions (where we include the Gamma distributions as light-tailed) and the cases where one or more are from heavy-tailed distributions. Due to the scale of the simulations in the second scenario, we present only some selected representative cases, including six cases where all three true predictors are light-tailed; two cases where

one of the true predictors is heavy-tailed; three case where two of the true predictors are heavy-tailed; and one case where all three true predictors are heavy-tailed. Results for all other cases are given in Sections A.1 and A.2.

In both scenarios, the average number of true variables and false variables selected are used for comparing the variable selection performance for all models. We exclude Lasso and Adaptive-Lasso results from some figures, as these methods have very high false positive rates. To make the comparisons clearer, we have linked the points for different tuning parameters of each method at the same SNR level in all the plots showing the true and false variable selections, i.e. for the cutoff values of 0.6, 0.7, 0.8 and 0.9 for the Stability selection results for the per-family error rate of 1 (termed STAB1) and the per-family error rate of 0.0526 (termed STAB2); and for the significance levels α in the range of $0.002 - 0.2$ for SuRF.

For predictive performance, we simulate independent test data with the same number of replicates (100 replicates) and the same sample sizes ($N = 100$) and use the prediction mean squared errors (PMSE), misclassification error rate (MCER) and the mean squares of the log Poisson mean (MSElogLambda) to compare the prediction results for Gaussian, Binomial and Poisson regression models, respectively. The MSElogLambda is defined as mean squared differences between the predicted values for the log Poisson means and the true log Poisson means in the test data set. To avoid the results being unduly influenced by several outstanding outliers, we adopted a robust version of the measure in Gaussian model (trimmed PMSE) and Poisson model (trimmed MSElogLambda) with the largest 5% of the errors (corresponding to five test datasets with the largest mean errors) removed. Lasso and Adaptive-Lasso both estimate regularised coefficients for the selected models, and we use these regularised coefficients for prediction. Stability selection and SuRF, do not estimate regularised coefficients for the selected predictors, so we use models fitted on the training data using the selected variables without further shrinkage for prediction. For Stability selection and SuRF applied on the Box-Cox transformed predictors, we calculated the predictive accuracies using both a misspecified GLM model fitted on the transformed version of the selected predictors (with the test data transformed in the same way as the training data), and a GLM model fitted on the selected predictors using the original untransformed predictors. The former method resulted in worse prediction due to the misspecification. Thus we only present the prediction results from the latter approach.

Prediction accuracies for different methods under the same SNR level are also linked in all the plots to make the comparison clearer. We also include the model prediction on test data based on the model fitted on the training data using the known true variable(s) (”the true model”). We only include the prediction accuracy for Stability selection with cutoff 0.6 in the Binomial and Poisson scenarios, since the prediction accuracies at other cutoffs are generally similar for high SNRs and less good for low SNRs.

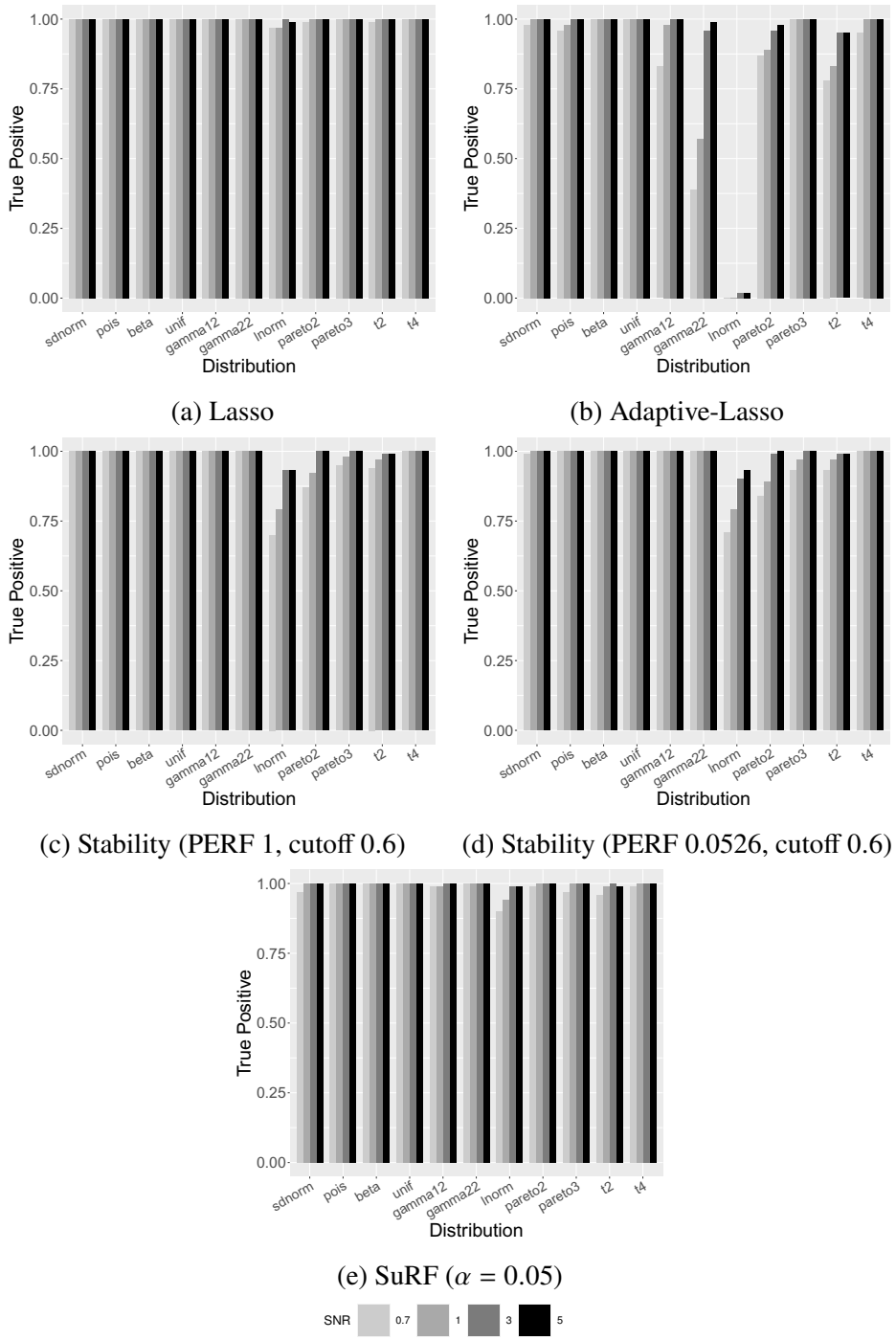


(a) $0 < x < 5$ (b) $x > 5$

(c) $0 < x < 2$ (d) $x > 2$

Figure 3.2: Comparison of tail behaviour of heavy-tailed distributions: original scale and log scale.

Distributions have been centred and rescaled to have variance 1.



(a) Lasso

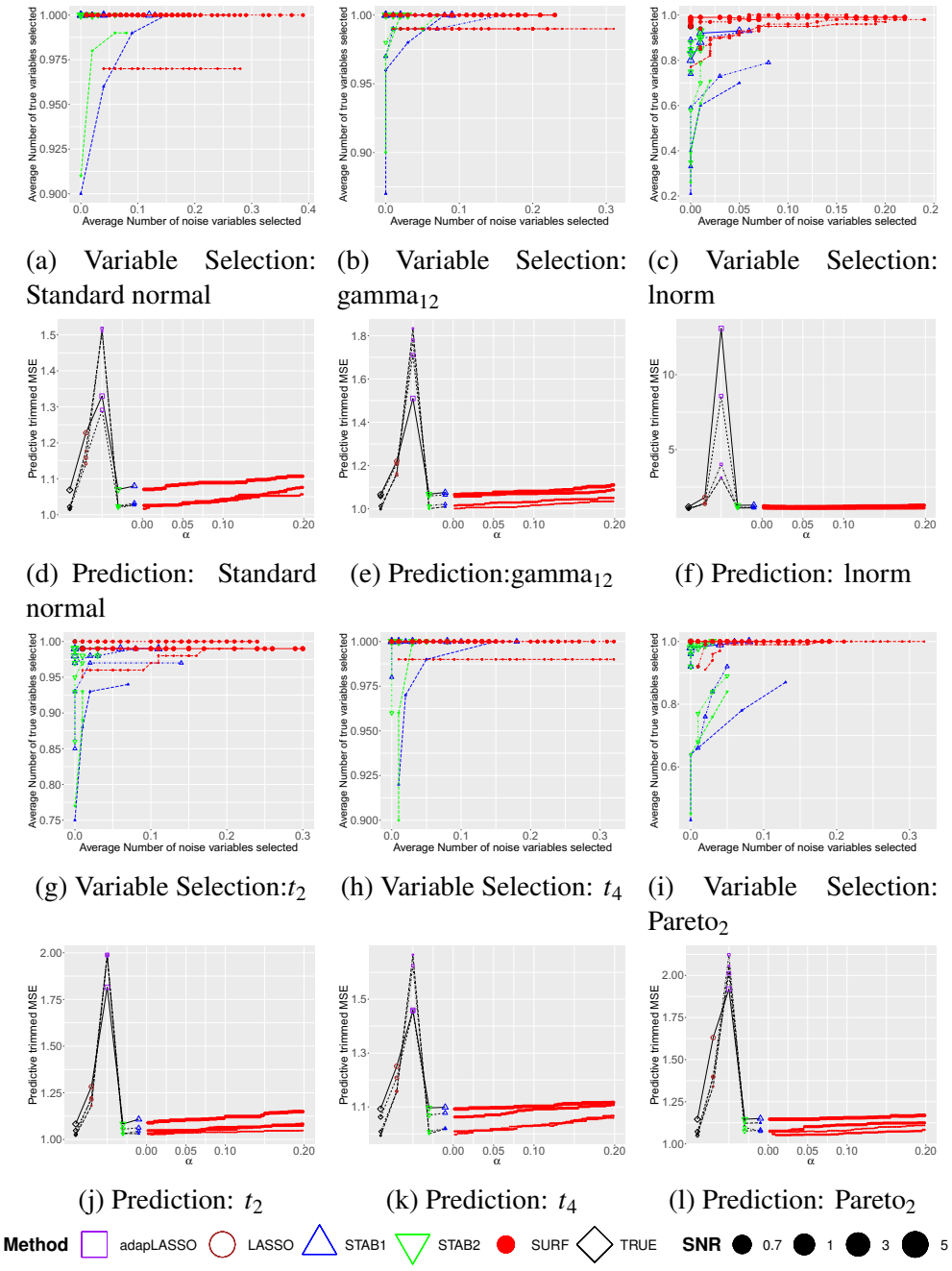
(b) Adaptive-Lasso

(c) Stability (PERF 1, cutoff 0.6)

(d) Stability (PERF 0.0526, cutoff 0.6)

(e) SuRF ($\alpha = 0.05$)

Figure 3.3: Comparison of true positive selection rates for Gaussian regression models with one true predictor under different distributions of the true predictor, at four SNR levels.



(a) Variable Selection: Standard normal (b) Variable Selection: gamma$_{12}$ (c) Variable Selection: lnorm

(d) Prediction: Standard normal (e) Prediction:gamma$_{12}$ (f) Prediction: lnorm

(g) Variable Selection:t_2 (h) Variable Selection: t_4 (i) Variable Selection: Pareto$_2$

(j) Prediction: t_2 (k) Prediction: t_4 (l) Prediction: Pareto$_2$

Figure 3.4: Comparison of variable selection and prediction for Gaussian regression with one true predictor

Panels in the 1st and 3rd rows show the true positive versus false positive rates for variable selection: results for different tuning parameters for each method at each SNR level are linked. Panels in the 2nd and 4th rows show the corresponding predictive trimmed MSEs on test data sets: results at each SNR level are linked. All results are averaged over 100 data sets. The circled SuRF results in the variable selection panels correspond to the cases where the prediction MSEs are the same for SuRF and STAB1 (cutoff 0.6) in the prediction panels.

3.3.1 Results for Gaussian Regression Model

Single true variable scenario

Figure 3.3 shows the frequency of the true positive selections when the true variable follows different distributions under four SNR levels. We see that all methods reliably select the four light-tailed distributions: normal, Poisson, beta(0.5,0.5) and uniform. The Gamma distributions were reliably selected by all methods except Adaptive-Lasso. The heavy-tailed distributions: log-normal, Pareto and t, were more difficult to select. Lasso is still able to reliably select these predictors, and SuRF is able to reliably select the Pareto and t predictors. Stability and Adaptive-Lasso frequently fail to select these predictors at low SNR. Scenarios where the true predictor is log-normal are most challenging, with Lasso, then SuRF, able to more reliably select the true predictor at low SNR, Stability sometimes failing to select the true predictor even at high SNR, and Adaptive-Lasso rarely selecting the true predictor even at high SNR. Adaptive-Lasso is clearly most affected by the distribution of the true predictor, followed by Stability, with SuRF and Lasso able to select the true predictor in almost all scenarios. The per-family error rate bound does not greatly affect the true positive selection of Stability.

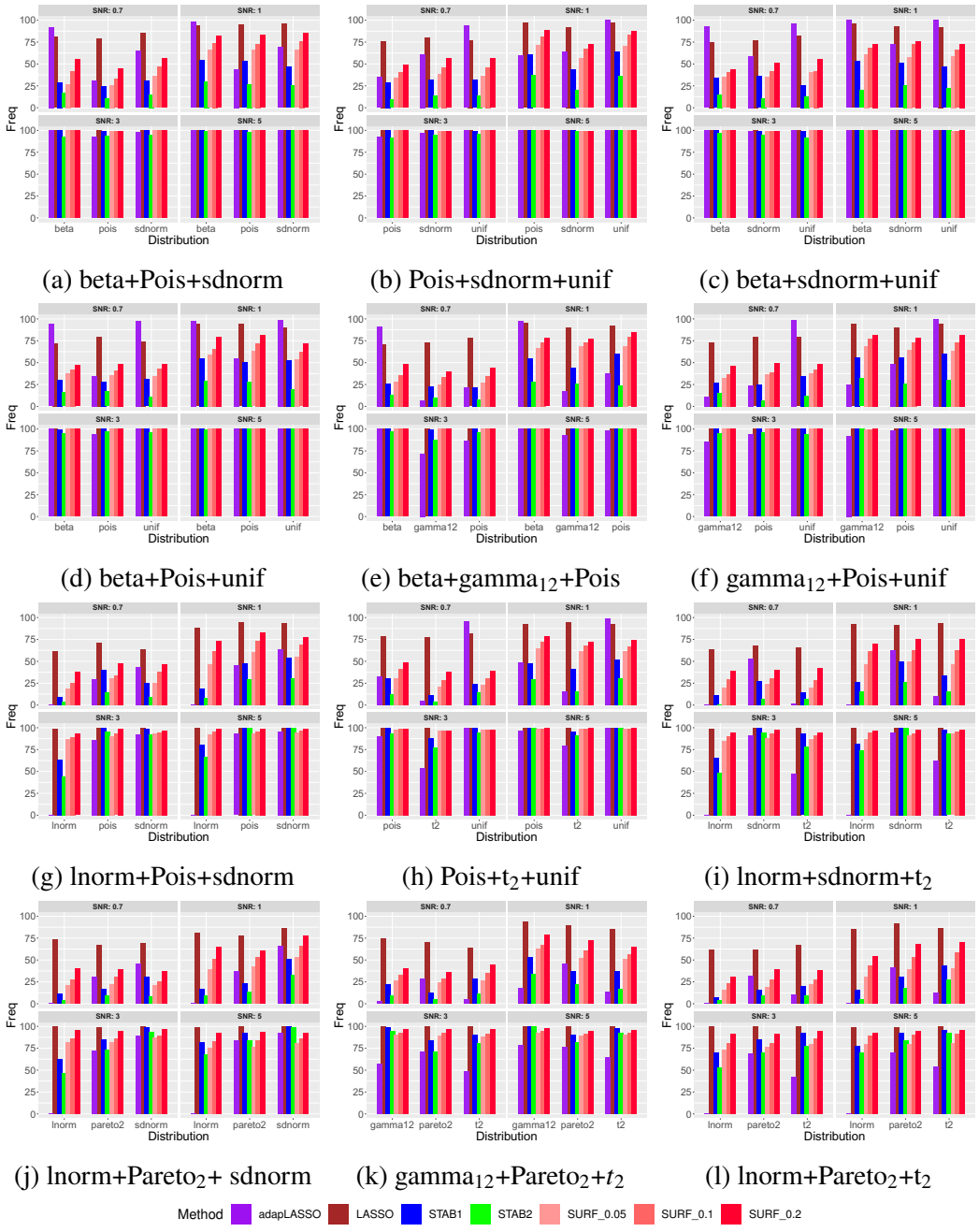
We also compared the number of false variables selected by each method (not shown in the figure). Adaptive-Lasso selects by far the most false positives, with the number of false positives higher for the low SNR scenarios. Lasso also selects a large number of false positives, but much less than Adaptive-Lasso, even at high SNR. The number of false positives selected by Lasso is comparable for all SNR levels. The false positive rates for Stability and SuRF in some scenarios are shown in Figure 3.4. The other scenarios are similar, with the plots included in Section A.1. In all scenarios, Stability and SuRF select many fewer false positives than Lasso. For the light-tailed (including gamma) scenarios, Stability and SuRF both reliably select the true predictor, with comparable false positive rates, which vary as we change the cut-off for Stability and the significance level for SuRF. For the log-normal distribution, SuRF is able to achieve a larger true positive rate with the same false positive rate as Stability at all SNR levels. We see that this results in better predictive accuracy. For the t and Pareto distributions, SuRF outperforms Stability at lower SNR; at higher SNR, both methods are able to reliably select the true predictor without selecting noise predictors.

Figure 3.4 also shows the trimmed Prediction mean squared errors (PMSEs) on the test datasets for each method. We used the trimmed PMSEs because the PMSEs based on all simulations were influenced by outliers in the heavy-tailed predictor case, even for the true model. Since Adaptive-Lasso has higher false positive rate and lower true positive rate than the other methods, it is not surprising that it has the highest trimmed PMSE in all scenarios. Lasso has the highest true positive rate, but also a very high false positive rate, which leads to larger trimmed PMSE than Stability and SuRF, which are much more conservative in variable selection. Furthermore, because of the shrinkage, even if Lasso selects the true variables, its estimate is biased, leading to inferior prediction. For the two values of per-family error bound in Stability, the prediction errors are similar, with the more conservative bound of 0.0526 performing slightly better. For low significance level, α, SuRF outperforms Stability in most scenarios, particularly when the true predictor is heavy-tailed.

Multiple true variables scenario

Figure 3.5 shows the frequency of each of the three true variables being selected in Gaussian regression models for some of the multiple true variables scenarios. As in the single true variable scenarios, Lasso and Adaptive-Lasso select a large number of noise variables. Being less conservative than Stability and SuRF, they also tend to select the true variable more often, particularly in the low SNR cases. Indeed, Lasso selects all true variables more frequently than Stability and SuRF, at the cost of selecting many more false positives. We note, however, that in the low SNR cases, Lasso selects heavy-tailed true predictors less frequently than light-tailed true predictors. Adaptive-Lasso is more volatile, selecting some true predictors more frequently than Lasso, but others less frequently, and almost never selecting log-normal predictors.

When all true predictors are light-tailed (e.g. Figure 3.5(a)–(f)), all methods reliably select the true predictors at high SNR. Selecting heavy-tailed predictors is much more difficult for all methods, and even at the highest SNR, Stability and SuRF sometimes fail to select these predictors. In particular, all methods struggle to select log-normal predictors. These results are similar to the single true predictor scenarios. We also note that when a heavy-tailed true predictor is included in the scenario, the ability of the methods to select light-tailed predictors also decreases. This makes sense, as the predictors are correlated, so



(a) beta+Pois+sdnorm (b) Pois+sdnorm+unif (c) beta+sdnorm+unif

(d) beta+Pois+unif (e) beta+gamma$_{12}$+Pois (f) gamma$_{12}$+Pois+unif

(g) lnorm+Pois+sdnorm (h) Pois+t$_2$+unif (i) lnorm+sdnorm+t$_2$

(j) lnorm+Pareto$_2$+ sdnorm (k) gamma$_{12}$+Pareto$_2$+t_2 (l) lnorm+Pareto$_2$+t$_2$

Figure 3.5: Frequency of each of three true variables being selected in the Gaussian regression model

The frequency bars for each variable are arranged from left to right by methods Adaptive-Lasso, Lasso, STAB1 (cutoff 0.6), STAB2 (cutoff 0.6), SuRF at significance level 0.05, 0.1 and 0.2. Four SNR levels 0.7, 1, 3 and 5 are shown in each panel.

if variable selection for one predictor becomes worse, it will impact variable selection for the other variables.

Since Stability and SuRF have different settings to control how conservative the variable selection is, we need to compare the performance in terms of both average number of true positives and average number of false positives.

Figure 3.6 shows the true positive and false positive rates for Stability and SuRF in 6 scenarios where all three true predictors are light-tailed, and the corresponding trimmed MPSEs. We see that Stability and SuRF trace very similar curves. At its most conservative, Stability is more conservative than SuRF, even at the lowest significance level considered. On the other hand, at higher values of the significance level, SuRF is able to be much less conservative than Stability. In the low SNR case, this allows it to achieve much better trimmed PMSE, as missing a true predictor usually has more effect on PMSE than including a false positive. Even at this level, SuRF selects a sparse model, with an average of much less than 1 false positive. Generally, the significance level where SuRF achieves the same trimmed PMSE as Stability is the one where the number of true predictors selected is equal. Sometimes, SuRF will have lower trimmed PMSE at a level where Stability has higher true positive rate and lower false positive rate. This can be explained by surrogate variables. When SuRF selects a surrogate of the true predictor, this increases the "false positive rate" without increasing the "true positive rate", but it also usually reduces the trimmed PMSE.

Figure 3.7 shows the variable selection performance and trimmed PMSE for Stability and SuRF on a number of scenarios where the three true predictors include at least one long-tailed variable. We see that at a given SNR level, the true positive versus false positive curve is much lower for both methods. False positives are at a similar level to the light-tailed cases, but the number of true positives is greatly reduced for both methods. As a consequence, the trimmed PMSE is also increased compared with the light-tailed cases. As in the previous cases, the reduction in true positives is most noticeable when one of the predictors is log-normal. Again both methods produce similar true positive versus false positive curves, but over different ranges. Similarly, for the different per-family error bounds in Stability, the trade-off between true positive and false positive rates is not changed much, but the lower per-family error bound selects variables more conservatively, with lower true positive and false positive rates. As in the light-tailed cases, prediction is most affected by missing a true predictor, so less conservative settings for SuRF and Stability tend to

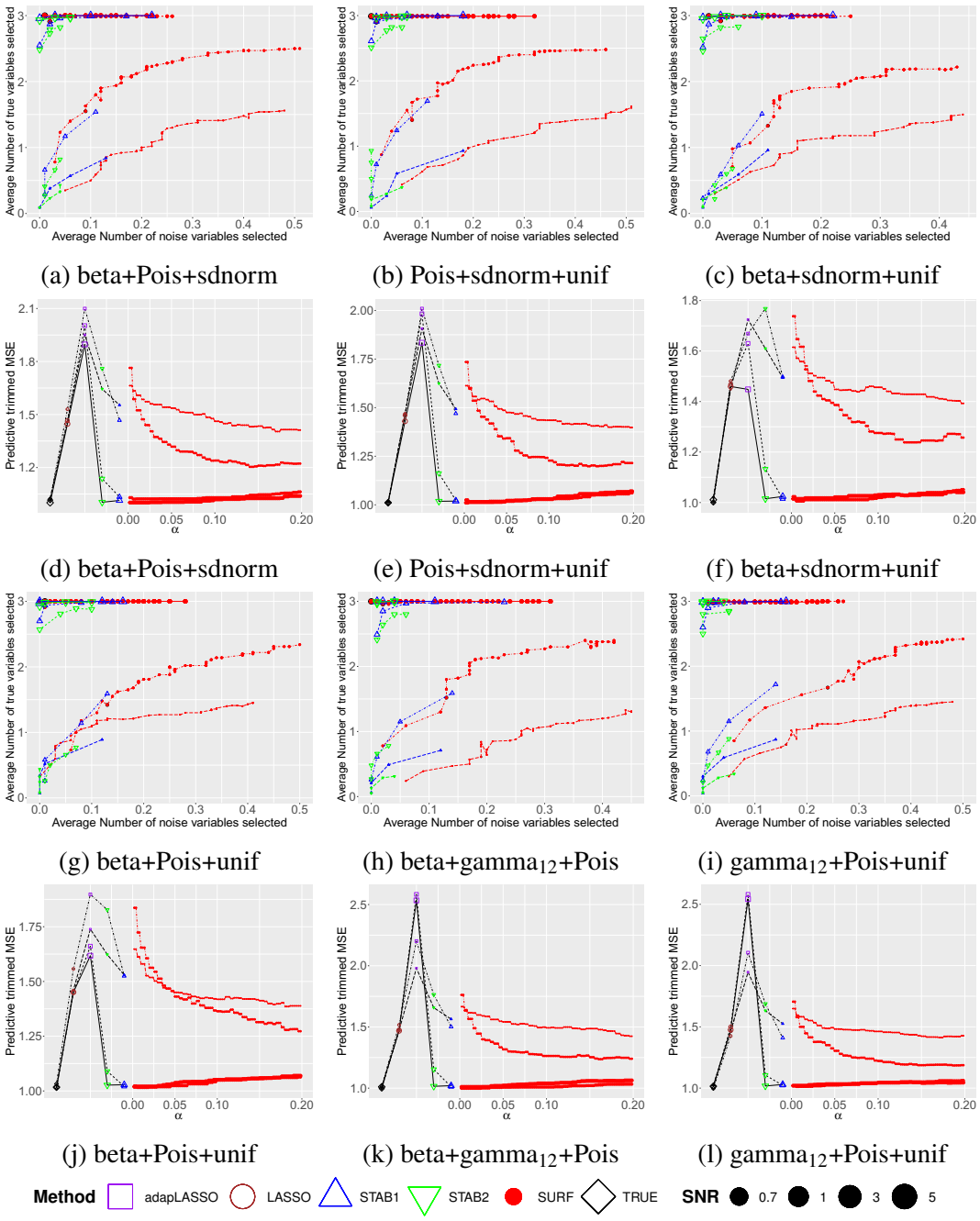


Figure 3.6: Gaussian regression models with three true light-tailed distributed predictors

Gaussian models with three true predictors, all with light-tailed distributions: panels in the 1st and 3rd rows show the true positive versus false positive rates for variable selection, the same method with different tuning parameters at the same SNR level are linked; panels in the 2nd and 4th rows show the corresponding predictive MSEs on test data sets, different methods on the same SNR level are linked. All results are averaged over 100 data sets. The circled SuRF results in the variable selection panels correspond to the cases that the prediction MSEs are the same for SuRF and STAB1 (cutoff 0.6) in the prediction panels.

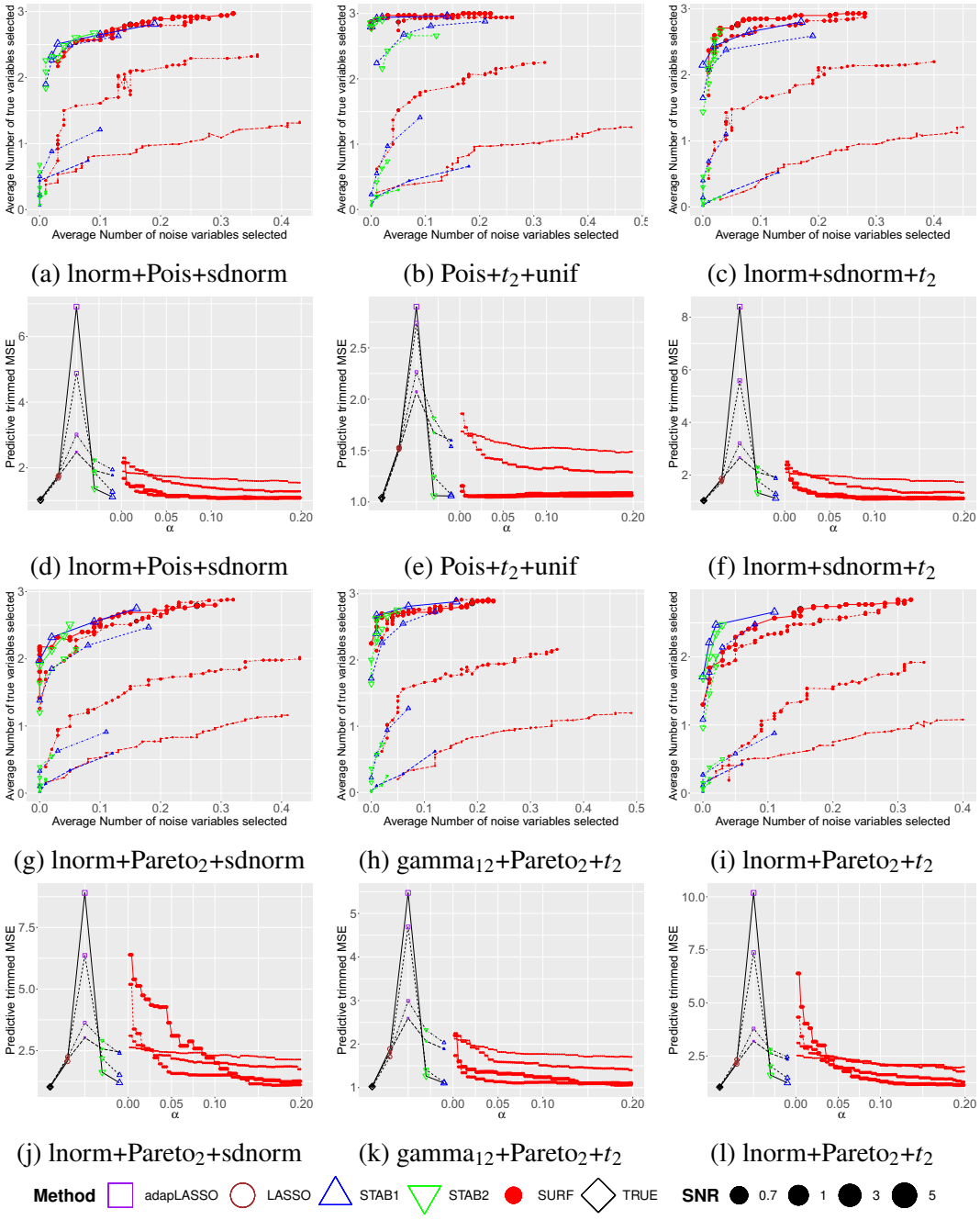


Figure 3.7: Gaussian models with three true predictors: cases with at least one heavy-tailed predictor

Gaussian models with three true predictors: Panels (a) (b) and (d) (e) are for variable selection and prediction with two light-tailed and one heavy-tailed predictors; Panels (c)(g)(h) and (f)(j)(k) are for variable selection and prediction with one light-tailed and two heavy-tailed predictors; Panels (i) and (l) are for variable selection and prediction with three heavy-tailed predictors.

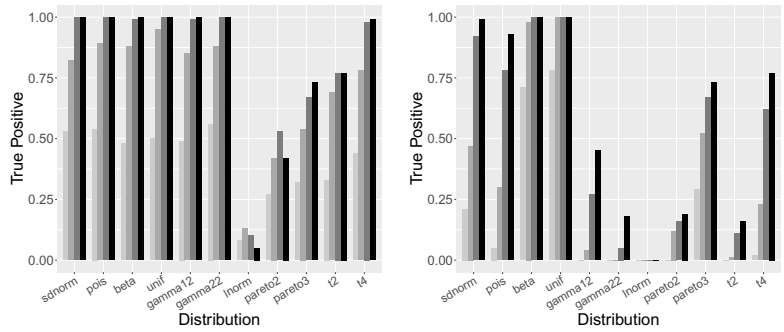
have lower trimmed PMSE. However, Lasso selects a lot of false positives, and shrinks the coefficients of the true positives, so it has higher trimmed PMSE, compared with Stability and SuRF in the high SNR case. In the low SNR case, it often outperforms Stability and SuRF, which often fail to select the true predictors.

3.3.2 Results for Binomial Regression Model

Single True Variable Scenarios

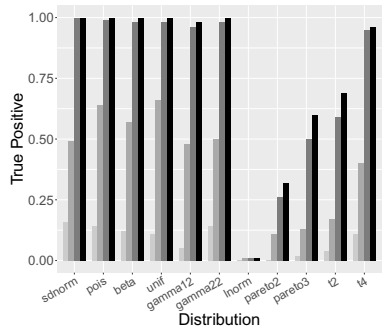
Figure 3.8 shows the proportion of simulations in which the true predictor is selected in each scenario for each method. We see that for all methods, the light-tailed true predictors are easier to select than the heavy-tailed true predictors. Most methods select the gamma predictors at about the same frequency as the light-tailed predictors. However, Adaptive-Lasso has difficulty selecting the true gamma predictors, selecting them with lower frequency than some of the heavy-tailed distributions. Among the heavy-tailed distributions, the log-normal is hardest to select for all untransformed methods, with many methods almost never selecting the log-normal true predictors, even at the highest SNR. Among the methods, Lasso has the highest frequency of selecting all light-tailed true predictors. On the other hand, Lasso has much higher false positive rate than Stability and SuRF, so it is not surprising that it is able to achieve a higher true positive rate. Adaptive-Lasso performs worse than Lasso in terms of both true positive and false positive rate. For the heavy-tailed true predictors, Stability selects the true predictor less often than Lasso, meaning that it amplifies the bias in Lasso. SuRF is much more able to select heavy-tailed true predictors, particularly the log-normal true predictors, where it has higher true positive rate than Lasso. For the other heavy-tailed predictors, it has higher true positive rate than Lasso at high SNR, and lower true positive rate at low SNR. Overall, SuRF is far less influenced by the distribution of the true predictor. This makes sense, since the final selection in SuRF is by forward selection based on hypothesis tests instead of Lasso shrinkage, and forward selection is more robust to the distribution of the predictors.

The Box-Cox transformation improves the ability of SuRF and Stability to select the true log-normal and Pareto distributed predictors. For the t-distributed true predictor with 2 degrees-of-freedom, the Box-Cox transformation improves performance of Stability, but not SuRF, and for the t-distributed true predictor with 4 degrees-of-freedom, the Box-Cox transformation makes it a bit more difficult to select the true predictor. Note that

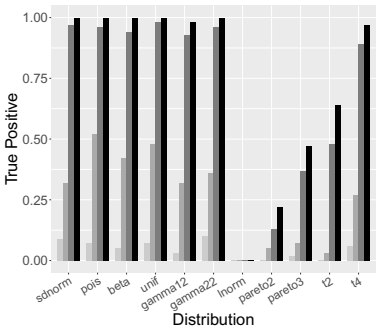


(a) Lasso

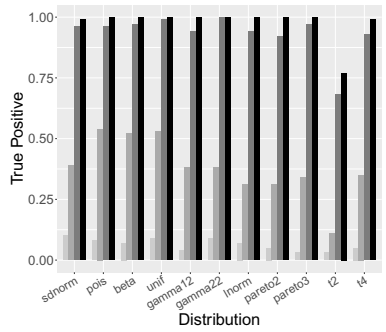
(b) Adaptive Lasso



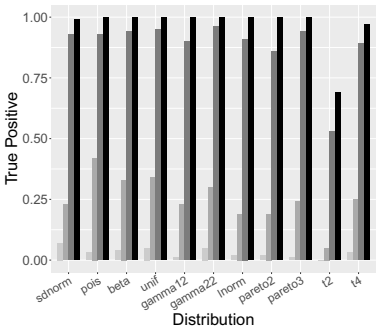
(c) Stability (PERF, cutoff 0.6)



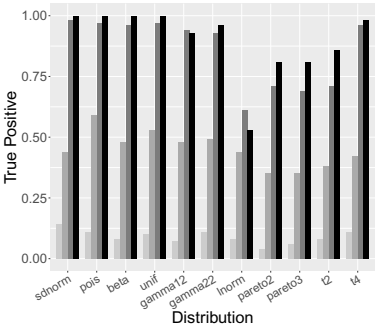
(d) Stability (PERF 0.0526, cutoff 0.6)



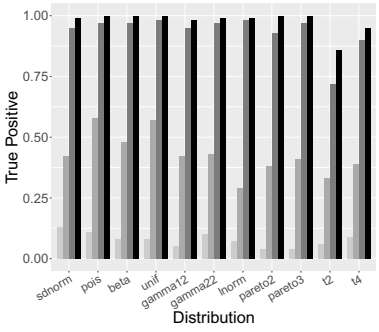
(e) Stability (PERF 1, cutoff 0.6) after Box-Cox



(f) Stability (PERF 0.0526, cutoff 0.6) after Box-Cox

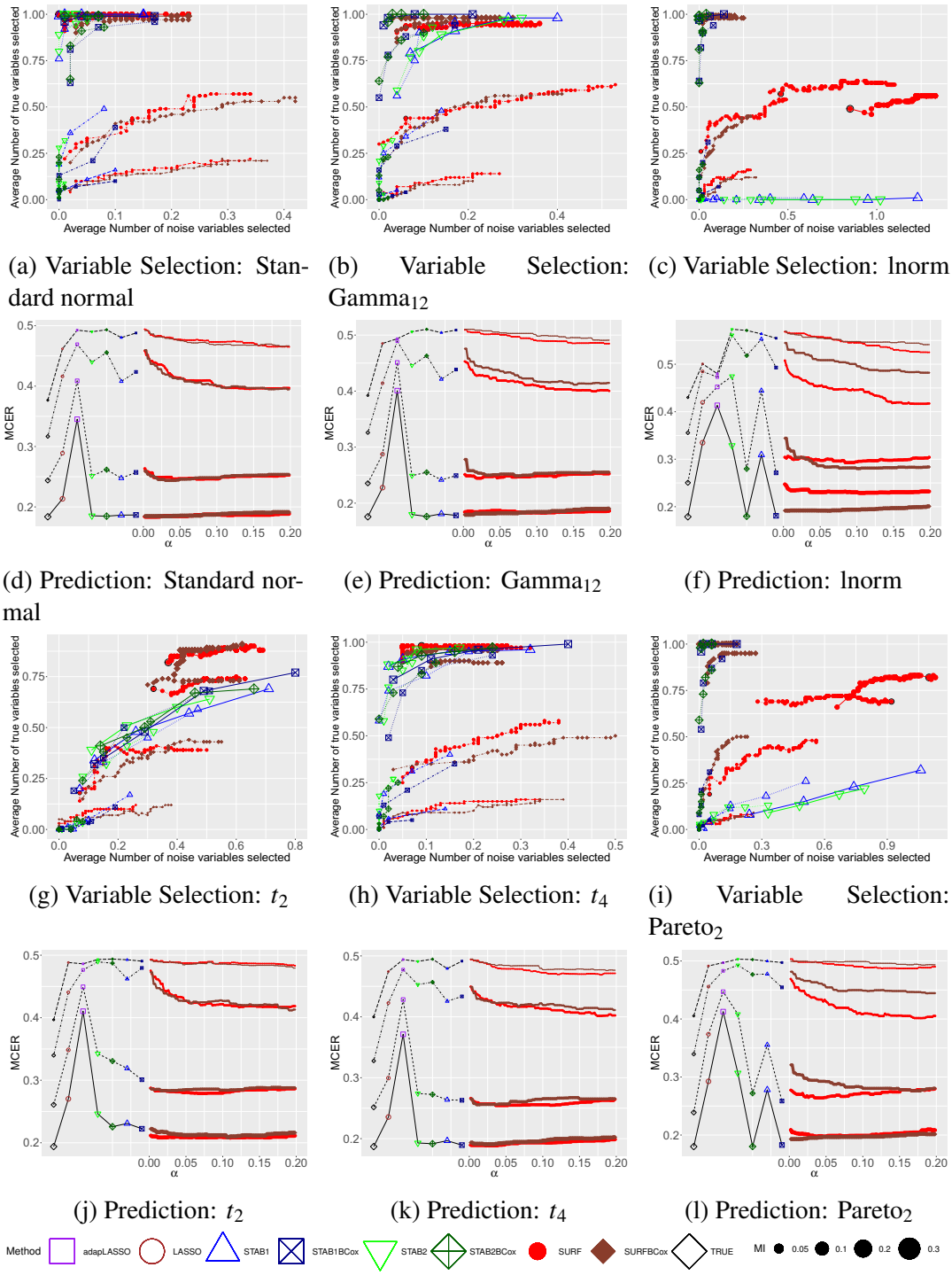


(g) SuRF ($\alpha = 0.05$)



(h) SuRF ($\alpha = 0.05$) after Box-Cox transformation

Figure 3.8: Comparison of true positive rate for Binomial logistic regression by distribution of the true predictor for various methods.



(a) Variable Selection: Standard normal

(b) Variable Selection: Gamma_{12}

(c) Variable Selection: lnorm

(d) Prediction: Standard normal

(e) Prediction: Gamma_{12}

(f) Prediction: lnorm

(g) Variable Selection: t_2

(h) Variable Selection: t_4

(i) Variable Selection: Pareto_2

(j) Prediction: t_2

(k) Prediction: t_4

(l) Prediction: Pareto_2

Figure 3.9: Comparison of variable selection and prediction for Binomial regression with one true predictor

Binomial models with one true predictor: panels in the 1st and 3rd rows show the true positive versus false positive rates for variable selection, the same method with different tuning parameters at the same mutual information (MI) level are linked; panels in the 2nd and 4th rows show the corresponding misclassification error rate (MCER) on test data sets, different methods on the same mutual information level are linked. All results are averaged over 100 data sets. The circled SuRF results in the variable selection panels correspond to the cases that the prediction MCERs are the same for SuRF (or SuRFBCox) and STAB1 (cutoff 0.6) in the prediction panels.

the data were simulated following logistic regression on the untransformed data, so after the Box-Cox transformation, the model is misspecified. However, the bias caused by the model being misspecified is less than the bias caused by the heavy-tailed predictor. It is not surprising that the Box-Cox transformation has more effect on the log-normal and Pareto distributions, because these distributions are skewed, so the Box-Cox transformation will select a transformation which makes the distribution less heavy-tailed; whereas the t-distribution is symmetric, so a Box-Cox transformation which makes the distribution less heavy-tailed would also make it more skewed.

Figure 3.9 shows true positives versus false positives for different methods and their predictive accuracy on test data for several cases. Lasso and Adaptive-Lasso are excluded from the comparison of true and false positives due to their high false positives. For light-tailed true predictors, SuRF and Stability selection both perform extremely well at high mutual information, reliably selecting the true predictor, and selecting very few noise predictors. When the SNR is smaller, Stability performs slightly better than SuRF, selecting fewer false positive variables for the same number of true positives. For the heavy-tailed true predictors, the performance of both methods drops, but the drop in performance is much larger for Stability. This drop in performance is most pronounced for the log-normal true predictor, followed by the Pareto true predictor. For the t-distributed true predictor with 4 degrees-of-freedom, the performance of SuRF is similar to the light-tailed true predictor cases, both in terms of true positives versus false positives and in terms of prediction misclassification error rate, though the performance for Stability is slightly worse. The better performance of Stability for the light-tailed true variable scenarios might be explained by its worse performance on the heavy-tailed predictors. Since many of the candidate predictors are heavy-tailed, in the case where the heavy-tailed predictors are all noise variables, a method which has difficulty selecting heavy-tailed predictors would be expected to select fewer noise variables.

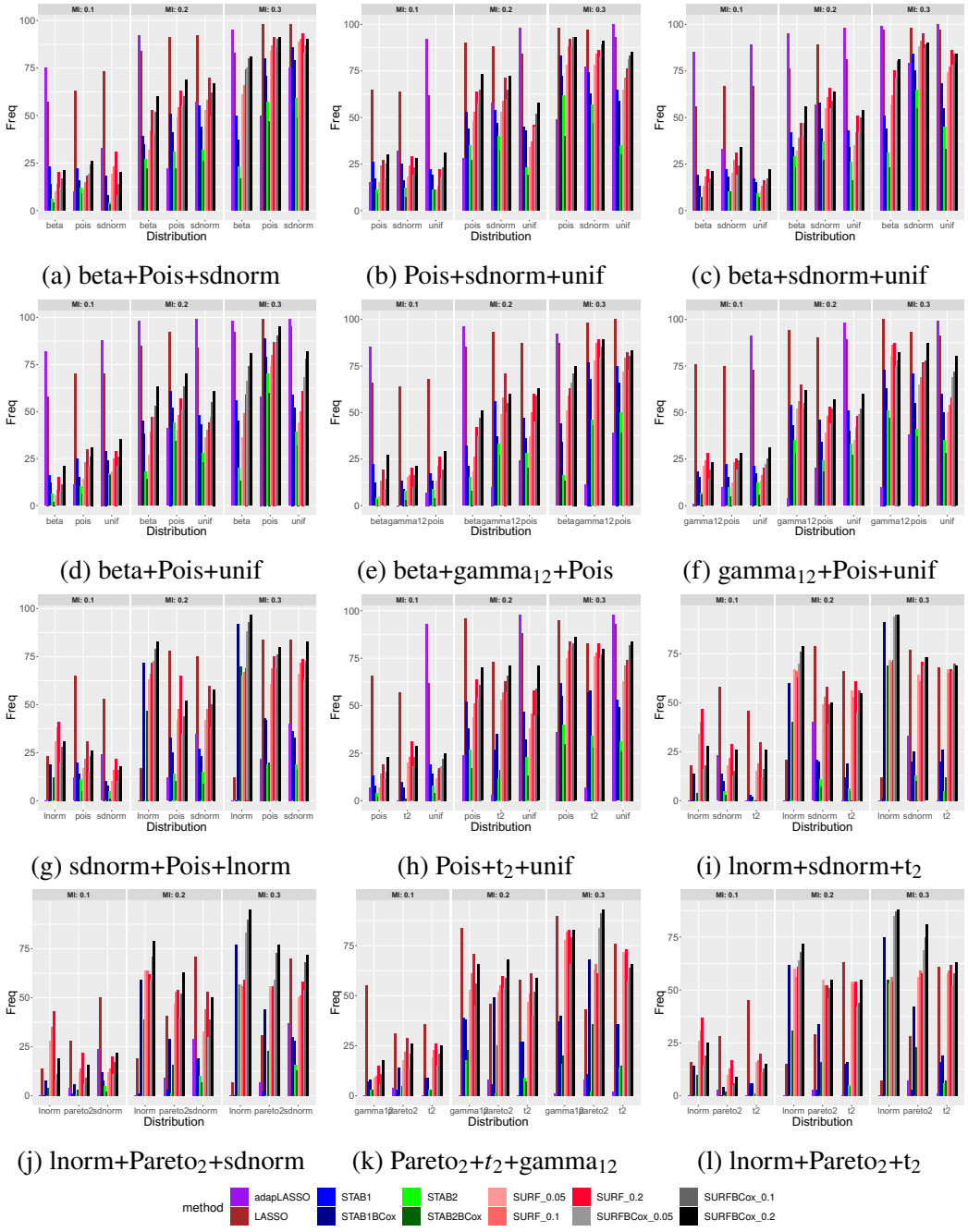
It can be observed that the Box-Cox transformation greatly improves the variable selection and prediction for both SuRF and Stability when the true predictor follows a skewed and long-tailed distribution such as log-normal or Pareto distribution in high SNR scenarios; however this effect is not obvious or even slightly worse at lower SNRs. For light-tailed true predictors and t distributions, Box-Cox transformation has little effect on both the variable selection and prediction.

Multiple True Predictor Scenarios

Figure 3.10 shows the frequency with which each true predictor is selected by each method, over 100 simulations for each scenario, over a range of cases, including cases with zero, one, two and three heavy-tailed true predictors. As in the single-variable case, Lasso consistently selects light-tailed variables more than other methods (with the exception of Adaptive-Lasso, which is extremely variable, sometimes frequently selecting the true variable, and sometimes selecting it less often than other methods). Lasso and Adaptive-Lasso also select a large number of noise variables. Stability and SuRF are more conservative, often struggling to select the true predictors when the MI is low. All methods are less likely to select heavy-tailed predictors, particularly log-normally distributed predictors. SuRF is more robust to long-tailed predictors than Stability and Lasso. Stability almost never selects the log-normal distributed true predictors, even in the high MI case.

Also as in the single-variable case, the Box-Cox transformation greatly improves the ability of Stability to select log-normal and Pareto true predictors, but decreases the ability to select other predictors. Even after the Box-Cox transformation, Stability selects the log-normal and Pareto distributed true predictors less often than SuRF without Box-Cox transformation. The Box-Cox transformation also increases the ability of SuRF to select true log-normal, uniform and beta predictors at high MI, with similar performance as without Box-Cox transformation for selecting true predictors for other cases.

Figures 3.11 and 3.12 compare the true positive versus false positive curves for Stability and SuRF with and without the Box-Cox transformation, and the predictive misclassification error rate for all methods. For cases without long-tailed predictors, the methods cover a different range of the true positive versus false positive curves, but show similar trade-offs between true positives and false positives with some methods performing better in some cases and other methods performing better in other cases. Stability slightly outperforms SuRF overall in the medium and low MI cases, which might be explained by the fact that Stability has difficulty selecting the heavy-tailed predictors, which effectively makes this a lower-dimensional problem for Stability than for SuRF. The prediction MCER are consistent with the true positive versus false positive curves — methods with higher true positive rates predict better, with the exception of Lasso and Adaptive-Lasso, which have very high false positive rates. At low and medium MI, Lasso usually gives the best prediction, while at high MI SuRF with a high value of α usually outperforms Lasso. The Box-Cox transformation



(a) beta+Pois+sdnorm (b) Pois+sdnorm+unif (c) beta+sdnorm+unif

(d) beta+Pois+unif (e) beta+gamma$_{12}$+Pois (f) gamma$_{12}$+Pois+unif

(g) sdnorm+Pois+lnorm (h) Pois+t$_2$+unif (i) lnorm+sdnorm+t$_2$

(j) lnorm+Pareto$_2$+sdnorm (k) Pareto$_2$+t_2+gamma$_{12}$ (l) lnorm+Pareto$_2$+t$_2$

Figure 3.10: Frequency of selecting individual true variables in logistic regression with three true predictors

MI level between 0.1 and 0.3 from left to right for each panel. Methods compared include Adaptive-Lasso, Lasso, two Stability selection methods (cutoff 0.6) and their application on the Box-Cox transformed predictors (STAB1BCox and STAB2BCox), SuRF (significant level 0.05, 0.1 and 0.2) and SuRF applied to the Box-Cox transformed predictors (SuRFBCox).

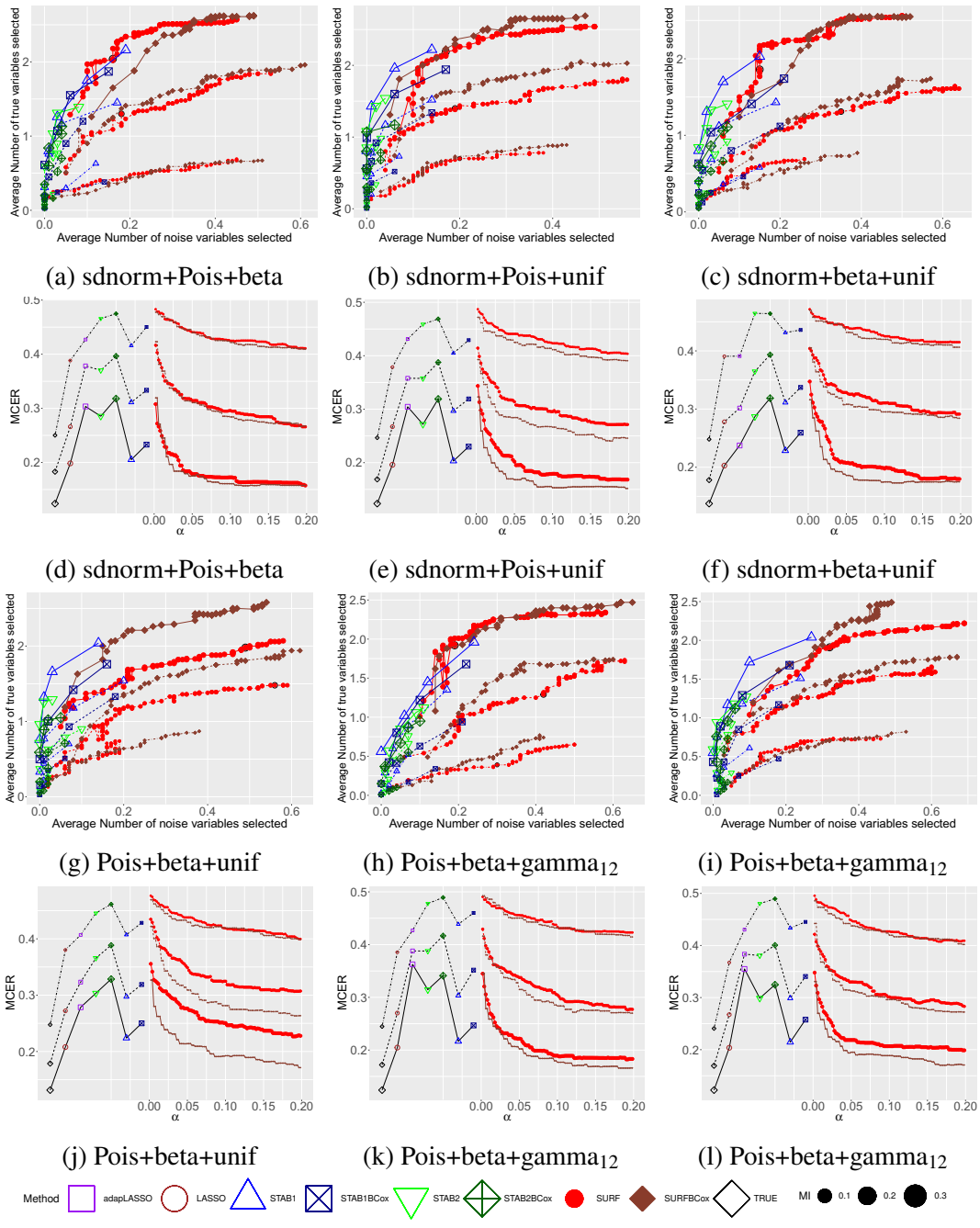


Figure 3.11: Comparison of variable selection and prediction for Binomial regression with three true predictors from light-tailed distributions

Panels in the 1st and 3rd rows show the true positive versus false positive rates for variable selection, the same method with different tuning parameters at the same mutual information (MI) level are linked; panels in the 2nd and 4th rows show the corresponding predictive MCERs on test data sets, different methods on the same MI level are linked. All results are averaged over 100 data sets. The circled SuRF results in the variable selection panels correspond to the cases that the prediction MCERs are the same for SuRF and STAB1 (cutoff 0.6) in the prediction panels.

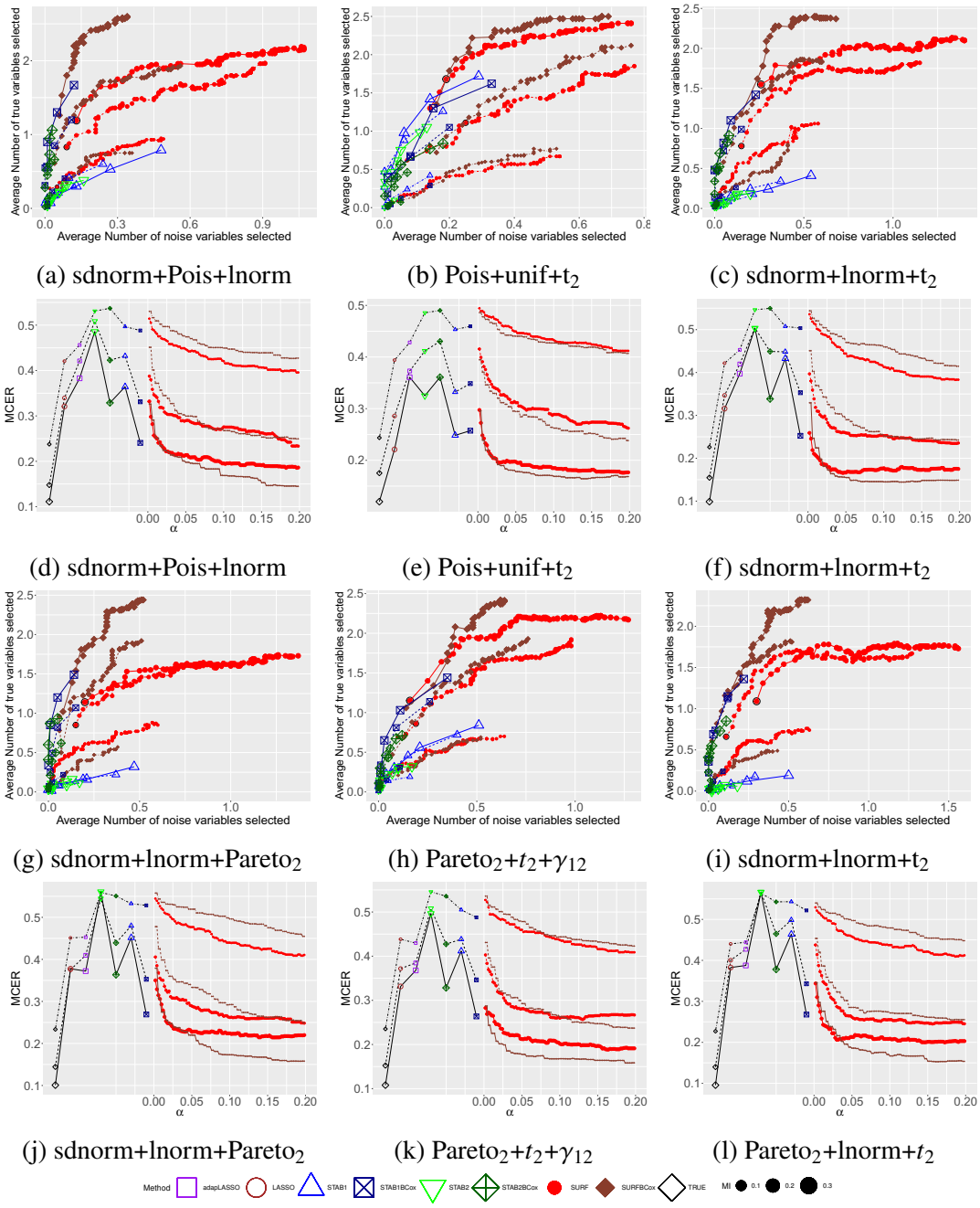


Figure 3.12: Comparison of variable selection and prediction for Binomial regression with three true predictors of which at least one from heavy-tailed distribution

Panels (a) (b) and (d) (e) are for variable selection and prediction with two light-tailed and one heavy-tailed predictors; Panels (c)(g)(h) and (f)(j)(k) are for variable selection and prediction with one light-tailed and two heavy-tailed predictors; Panels (i) and (l) are for variable selection and prediction with three heavy-tailed predictors. Panels in the 1st and 3rd rows show the true positive versus false positive rates for variable selection: results for different tuning parameters for each method at each mutual information (MI) level are linked. Panels in the 2nd and 4th rows show the corresponding predictive MCERs on test data sets: results at each MI level are linked. All results are averaged over 100 data sets. The circled SuRF results in the variable selection panels correspond to the cases where the prediction MCERs are the same for SuRF and STAB1 (cutoff 0.6) in the prediction panels.

in these cases causes Stability methods to perform slightly worse in terms of both true/false positive rate and MCER; It generally has a small positive impact on the variable selection of SuRF and often leads to a similar or lower MCER. Overall, the Box-Cox transformation has little effect for cases without long-tailed true predictors.

For scenarios involving a log-normal or Pareto true predictor, Stability performs much worse, while SuRF performs slightly worse than in the cases without heavy-tailed true predictors. The Box-Cox transformation greatly improves the performance of Stability and SuRF in these cases, by both increasing the true positive rate and decreasing the false positive rate, and also decreasing the prediction MCER. SuRF often shows the best variable selection and prediction results in cases where at least one heavy-tailed true predictor is present. At low MI level or in the scenarios where there are no heavy-tailed true predictors, the Box-Cox transformation shows little effect. The general conclusion is that applying the Box-Cox transformation on long-tailed predictors can be a useful step prior to Stability and SuRF variable selection for the Binomial model.

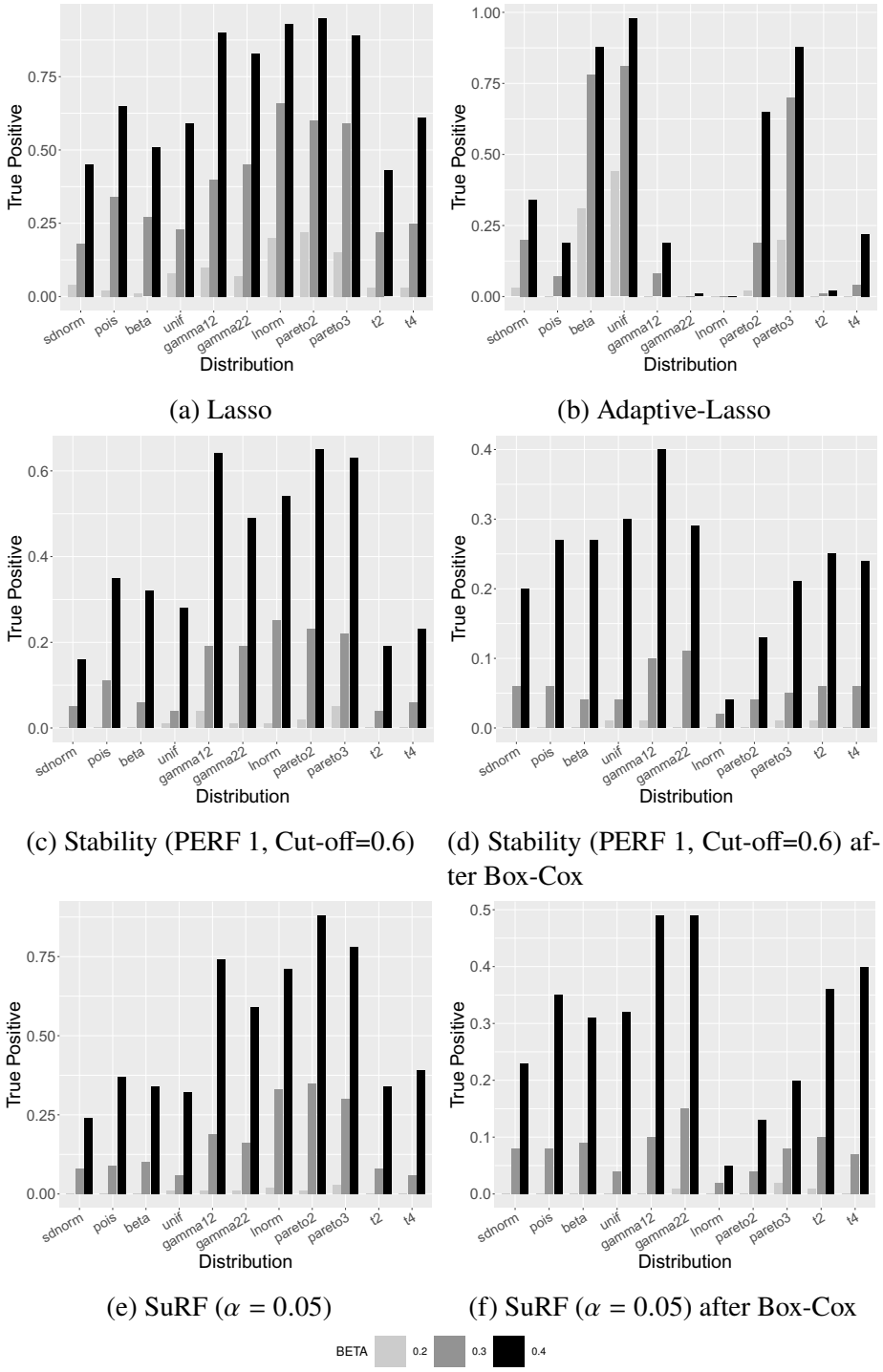
3.3.3 Results for Poisson Regression Model

Single True Variable Scenarios

Figure 3.13 shows the true positive rates for the different variable selection methods for different true variable cases when the median value of the Poisson mean is 1. Results for median value of Poisson mean equal to 0.8 and 2 are similar, except that variable selection is generally easier when the median is 2, and harder when the median is 0.8, and are shown in Section A.1. We see that the skewed medium-to-heavy-tailed true predictors (Gamma, log-normal, and Pareto) are selected more frequently by Lasso, Stability and SuRF. Adaptive Lasso is more likely to select beta and uniform true predictors, and never selects the log-normal true predictors. Unlike the binomial and Gaussian cases, SuRF is affected by the distribution of predictors as much as the other methods.

The Box-Cox transformation has little effect on the frequency with which the light-tailed (excluding Gamma) true predictors are selected, but causes a large decrease in the frequency with which the skewed heavy-tailed (including Gamma) predictors are selected, and causes an increase in the frequency with which the t-distributed true predictors are selected. The decrease in frequency is particularly large for the Pareto and log-normal distributions.

Figure 3.14 shows the true positive and false positive rates and trimmed PMSE of $\log(\lambda)$



(a) Lasso

(b) Adaptive-Lasso

(c) Stability (PERF 1, Cut-off=0.6)

(d) Stability (PERF 1, Cut-off=0.6) after Box-Cox

(e) SuRF ($\alpha = 0.05$)

(f) SuRF ($\alpha = 0.05$) after Box-Cox

Figure 3.13: Comparison of true positive rate for Poisson regression by distribution of the true predictor (Mrate=1).

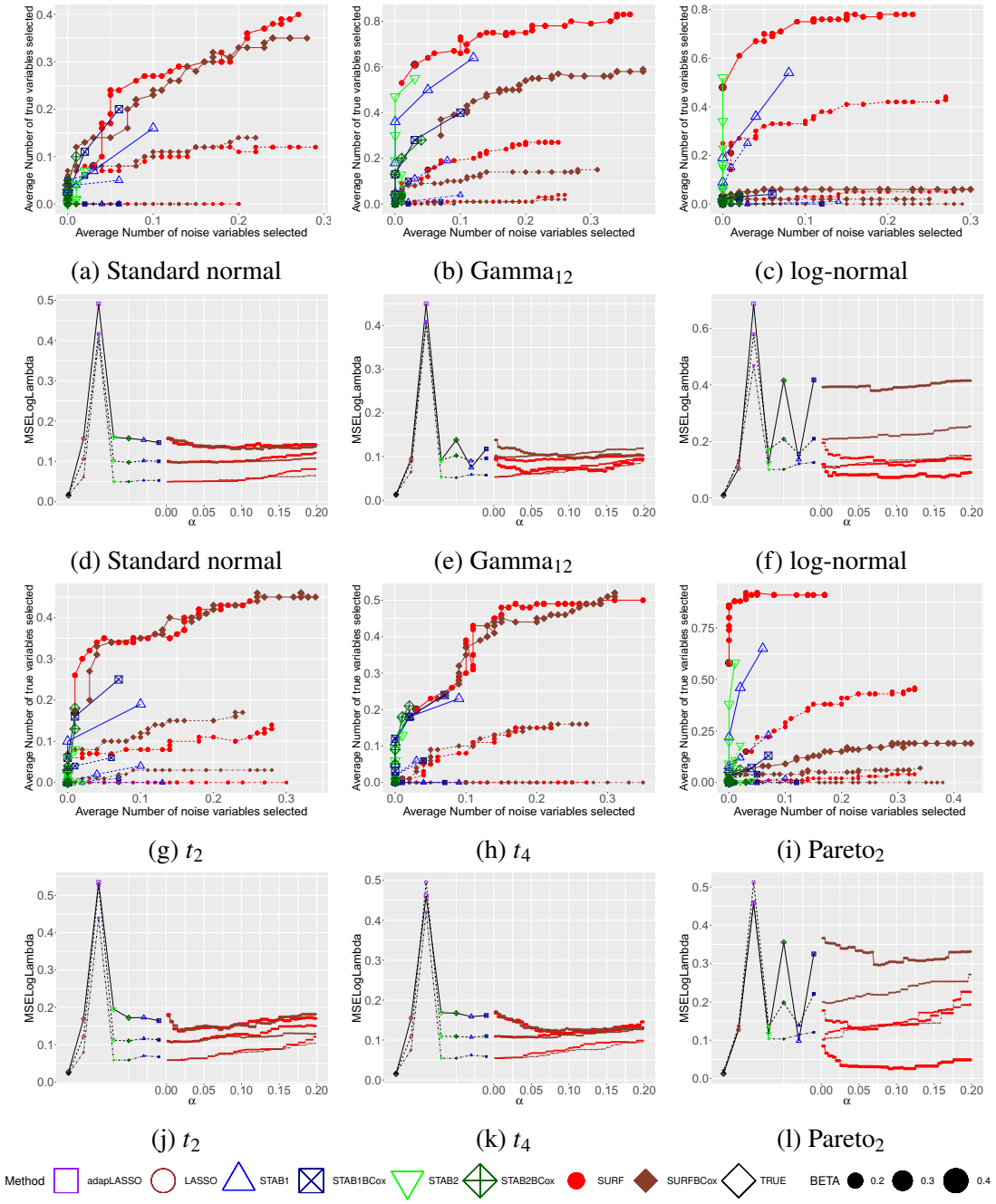


Figure 3.14: Comparison of variable selection and prediction for Poisson regression with one true predictor ((Mrate=1)

Panels in the 1st and 3rd rows show the true positive versus false positive rates for variable selection: results for different tuning parameters for each method at each β level are linked. Panels in the 2nd and 4th rows show the corresponding predictive MCERs on test data sets: results at each MI level are linked. All results are averaged over 100 data sets. The circled SuRF results in the variable selection panels correspond to the cases where the prediction MCERs are the same for SuRF and STAB1 (cutoff 0.6) in the prediction panels.

for each scenario in the cases with median Poisson mean equal to 1. Cases with median Poisson mean equal to 0.8 and 2 are similar, except that the variable selection is easier when the median is 2 and harder when it is 0.8, and shown in Section A.1. We see that the variable selection is much better for both methods when the true predictor has a skewed long-tailed distribution, with higher true positive rates and lower false positive rates, particularly for the high SNR case. This results in lower trimmed PMSE for $\log(\lambda)$. Note that the trimmed PMSE is sometimes higher in the high SNR case. This is because, although the variable selection is better in the high SNR case, the cost of failing to select the true predictor is also much higher.

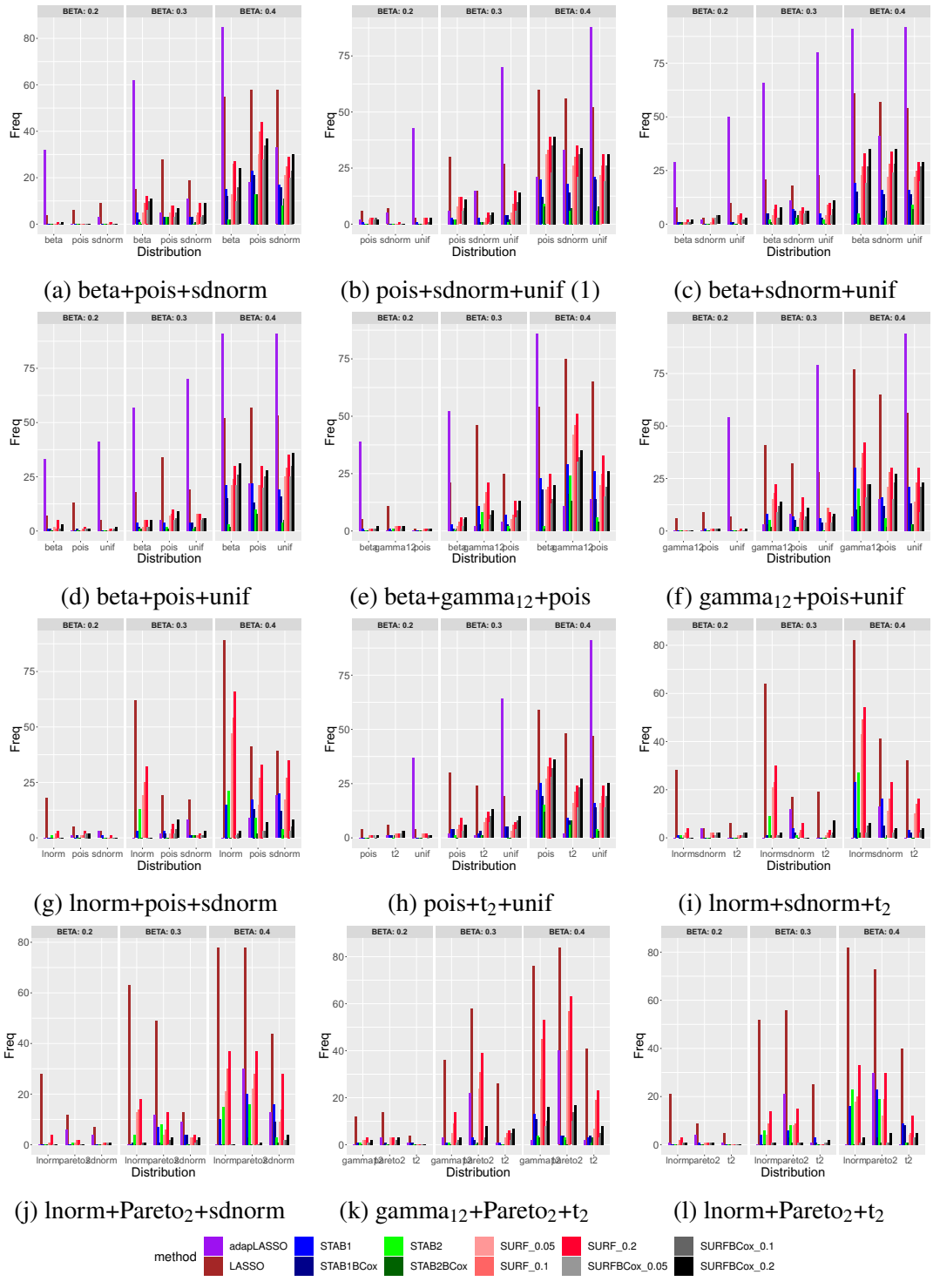
Stability with familywise error bound 0.0526 and SuRF have a similar true positive versus false positive curve in most cases, but cover different parts of the curve, with Stability more conservative. The exception is when the true predictor follows a t-distribution with 2 degrees-of-freedom, in which case SuRF performs better than Stability. Stability with the per-family error bound 1 generally performs worse than SuRF.

The Box-Cox transformation has little effect on the variable selection or prediction when the true predictor is symmetric, and causes variable selection and prediction to get much worse when the predictor is skewed.

Multiple true variable scenario

Figure 3.15 shows the frequency with which each true predictor was selected by each method for a selection of scenarios with median Poisson mean equal to 1. Similar results with median of Poisson mean equal to 0.8 or 2 are shown in Section A.1. Lasso selects more true positives than other methods, except Adaptive-Lasso, which varies a lot, but also selects a large number of false positives. Adaptive-Lasso does very well at selecting the Beta and uniform true predictors, and very badly for other predictors. With the exception of Adaptive-Lasso, all methods select the long-tailed skewed true predictors more often than other predictors. For Adaptive-Lasso, this pattern is reversed, and long-tailed predictors are selected less frequently, as in the Gaussian and binomial cases. As in the single-variable case, the Box-Cox transformation reduces the ability to select skewed heavy-tailed predictors without increasing the ability to select light-tailed predictors.

Figures 3.16 and 3.17 show the true positive rate and false positive rate, and the trimmed PMSE for $\log(\lambda)$ for Stability and SuRF with and without Box-Cox transformations, in a



(a) beta+pois+sdnorm (b) pois+sdnorm+unif (1) (c) beta+sdnorm+unif

(d) beta+pois+unif (e) beta+gamma$_{12}$+pois (f) gamma$_{12}$+pois+unif

(g) lnorm+pois+sdnorm (h) pois+t$_2$+unif (i) lnorm+sdnorm+t$_2$

(j) lnorm+Pareto$_2$+sdnorm (k) gamma$_{12}$+Pareto$_2$+t$_2$ (l) lnorm+Pareto$_2$+t$_2$

Figure 3.15: Frequency of true positive selection on individual variables in Poisson regression with three true predictors (Mrate=1)

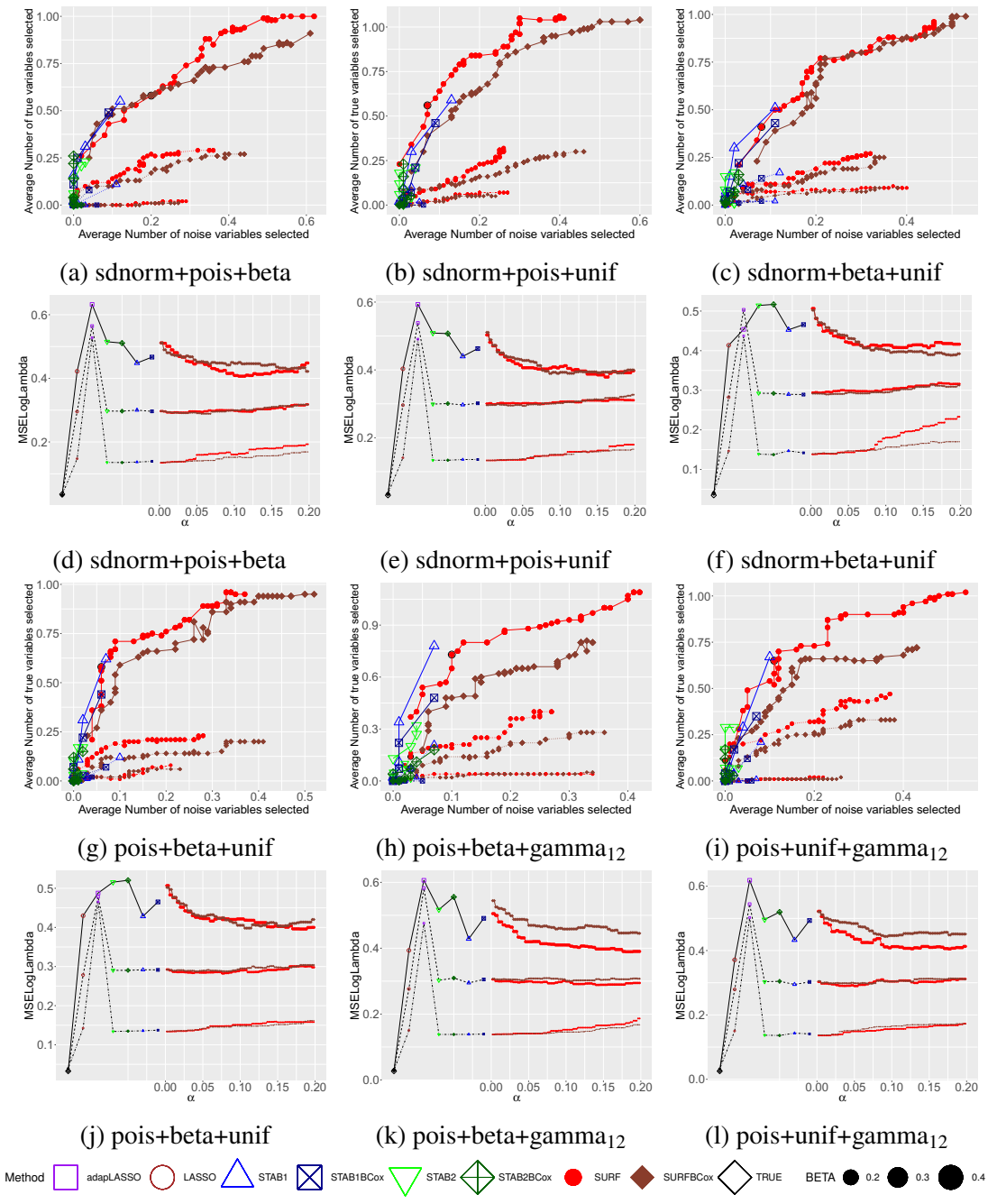


Figure 3.16: Comparison of variable selection and prediction for Binomial regression with three true light-tailed predictors (Mrate=1)

Panels in the 1st and 3rd rows show the true positive versus false positive rates for variable selection: results for different tuning parameters for each method at each β level are linked. Panels in the 2nd and 4th rows show the corresponding predictive MSEs (log rates) on test data sets: results at each β level are linked. All results are averaged over 100 data sets. The circled SuRF results in the variable selection panels correspond to the cases where the prediction MSEs (log rates) are the same for SuRF and STAB1 (cutoff 0.6) in the prediction panels.

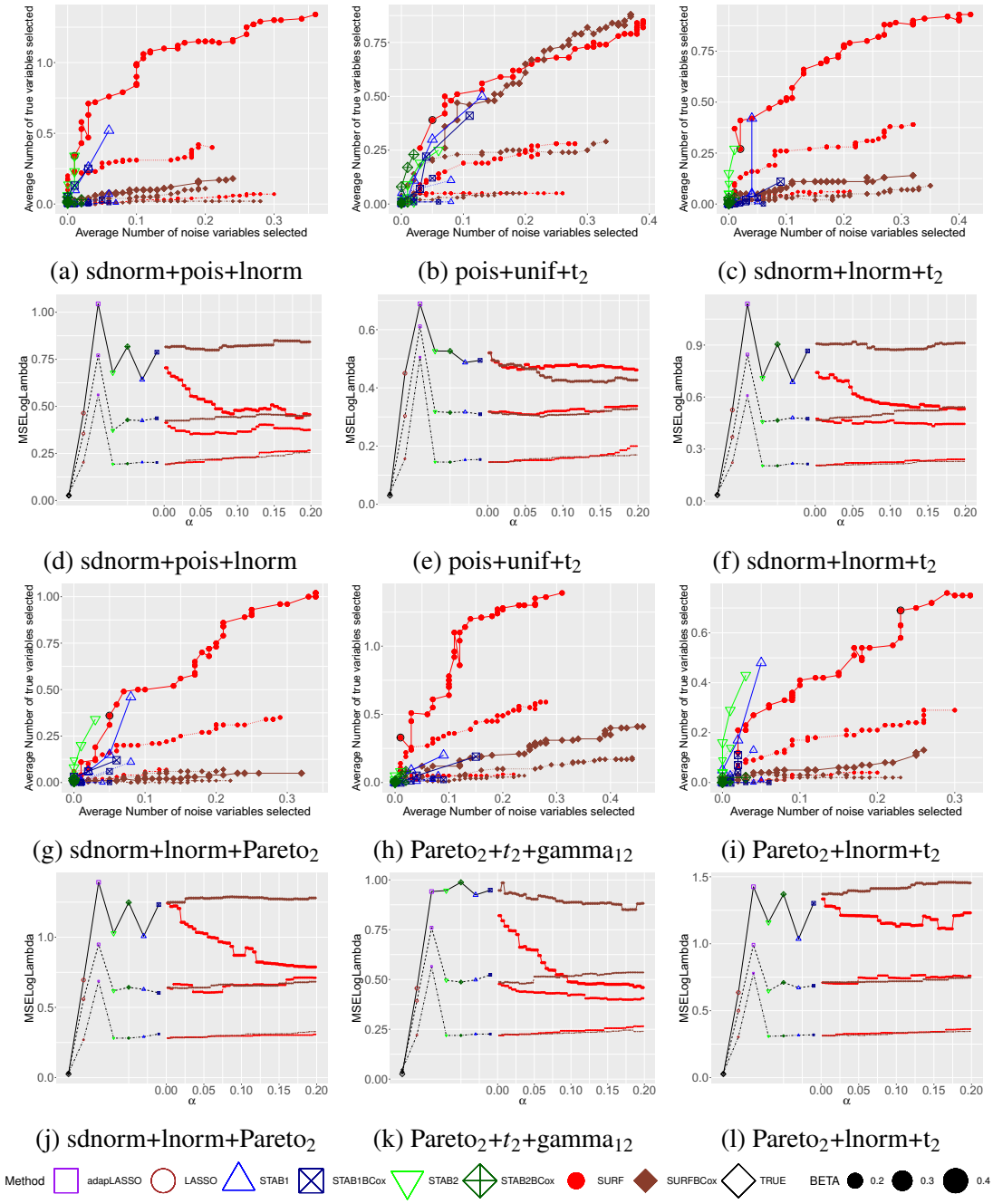


Figure 3.17: Comparison of variable selection and prediction for Poisson regression with three true predictors including at least one heavy-tailed predictor ((Mrate=1)

Panels (a) (b) and (d) (e) are for variable selection and prediction with two light-tailed and one heavy-tailed predictors; Panels (c)(g)(h) and (f)(j)(k) are for variable selection and prediction with one light-tailed and two heavy-tailed predictors; Panels (i) and (l) are for variable selection and prediction with three heavy-tailed predictors. Panels in the 1st and 3rd rows show the true positive versus false positive rates for variable selection: results for different tuning parameters for each method at each β level are linked. Panels in the 2nd and 4th rows show the corresponding predictive MSEs (log rates) on test data sets: results at each β level are linked. All results are averaged over 100 data sets. The circled SuRF results in the variable selection panels correspond to the cases where the prediction MSEs (log rates) are the same for SuRF and STAB1 (cutoff 0.6) in the prediction panels.

variety of scenarios with median of Poisson mean equal to 1. Results with median of Poisson mean equal to 0.8 or 2 are similar and shown in Section A.1. Stability appears to be more affected by the median of Poisson means than SuRF, though this varies significantly between scenarios.

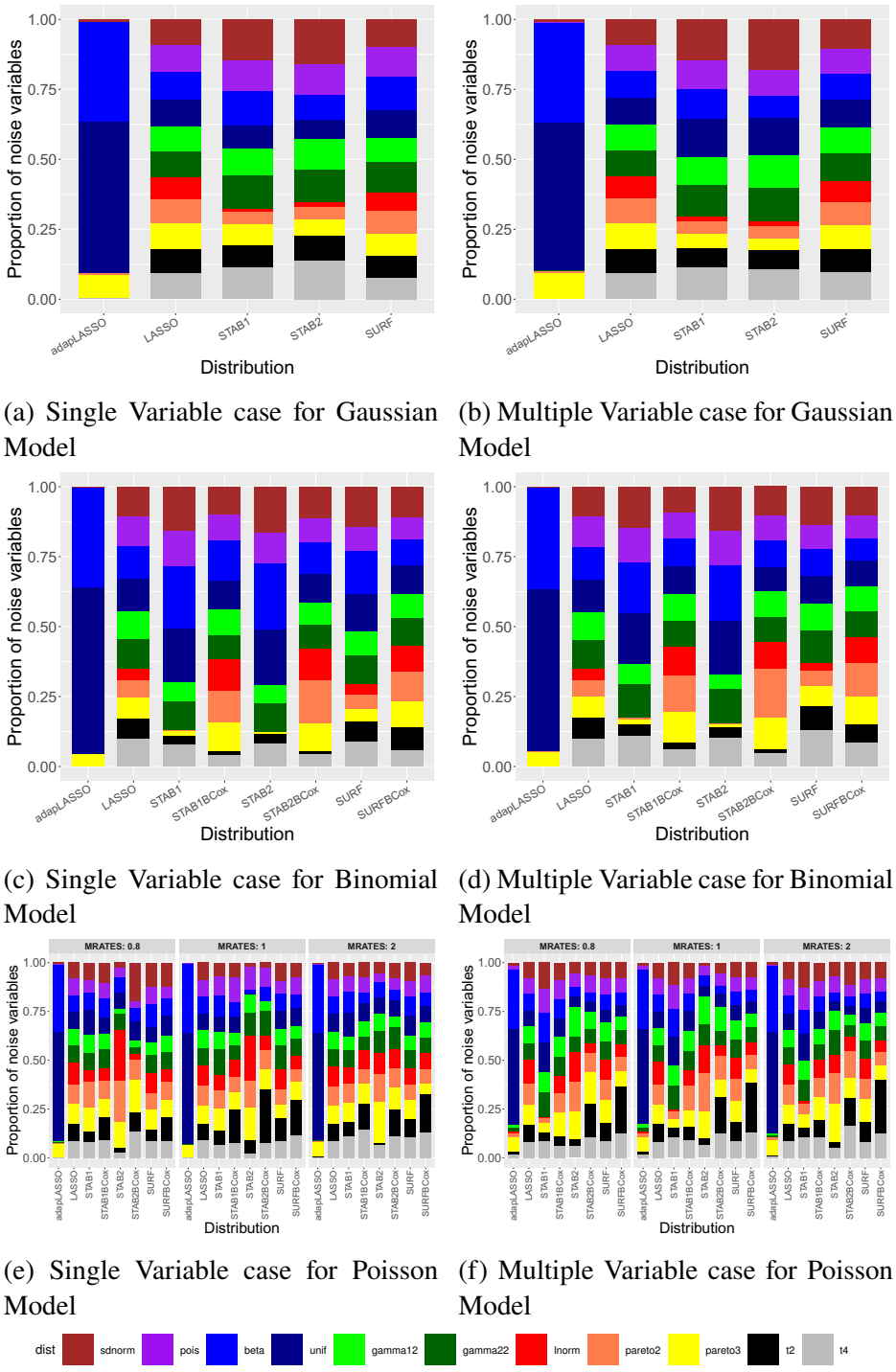
As in the other studies, Stability is limited to the more conservative part of the curve. For this study, the performance of Stability and SuRF is similar for most scenarios, but there are scenarios, such as (Pareto, shape=2.01; t, df=2; Gamma, shape=1), where SuRF outperforms Stability by a substantial margin, and scenarios such as (Pareto, shape=2.01; t, df=2; log-normal) where Stability outperforms SuRF for the largest coefficients.

This is a very challenging problem, and even for the largest coefficients the methods rarely select an average of more than 1 true predictor. As in the single variable case, results are generally better when at least one of the predictors is skewed and heavy-tailed.

Patterns of selection bias

Figure 3.18 shows the relative frequency with which noise variables were selected from each predictor distribution for each method. Recall that in all scenarios, there were 400 predictors with each distribution, so if there were no bias in the selected predictors, we would expect each distribution to represent the same proportion of all noise variables selected. It is worth noting that Stability and SuRF are conservative methods, and select an average of less than 0.2 false positives over all simulations. This means that the total number of false positives selected for the single true predictor cases was about 600 or less for these methods. Therefore, there is substantial variability in the estimated relative frequency. There is less variability in the multiple true variable case, as there were many more scenarios in that case.

We see that the noise variables show a similar pattern to the true predictors, with Lasso and SuRF selecting the noise variables equally for Gaussian regression; underselecting the heavy-tailed predictors in the log-normal, Pareto and t cases for logistic regression; and overselecting the log-normal and Pareto predictors for Poisson regression with log link function. Adaptive Lasso heavily favours beta and uniform predictors for all models, and almost never selects any heavy-tailed noise variables, despite selecting a large number of noise variables overall. Stability underselects the log-normal and Pareto predictors and the t distribution with 2 degrees-of-freedom, even in the Gaussian case.



(a) Single Variable case for Gaussian Model

(b) Multiple Variable case for Gaussian Model

(c) Single Variable case for Binomial Model

(d) Multiple Variable case for Binomial Model

(e) Single Variable case for Poisson Model

(f) Multiple Variable case for Poisson Model

Figure 3.18: Comparison of distributions of noise variables selected for Gaussian, Binomial and Poisson regression.

Results are averaged over all signal strengths and simulation scenarios. Cut-off 0.6 is used for Stability, and significance level $\alpha = 0.05$ is used for SuRF.

For logistic regression, Stability massively underselects the log-normal and Pareto predictors, and also underselects the t predictors with 2 degrees-of-freedom. The Box-Cox transformation makes it easier for Stability to select the log-normal and Pareto distributed predictors, but makes it more difficult to select the t-distributed predictors. These patterns of selection bias shown in the single variable cases are also confirmed in the multiple true predictor cases. The Box-Cox transformation causes SuRF to select more log-normal and Pareto predictors over other types of predictors. It is likely that these variables are surrogates of the true predictors.

For Poisson regression with a log link function, Lasso and SuRF select the noise variables almost equally between the distributions, suggesting that the different results for different true predictor distributions might be due to the signal sizes not being equal. Stability overselects the beta and uniform predictors when the per-family error bound is set to 1, and overselects the Pareto and log-normal predictors when it is set to 0.0526. The Box-Cox transformation causes Stability and SuRF to overselect t-distributed predictors, especially the t predictors with 2 degrees-of-freedom.

3.4 Conclusion

We have demonstrated that the marginal distribution of predictors can have a significant effect on the ability of Lasso and similar methods to select the true predictors. For Gaussian and logistic regression, heavy-tailed variables, particularly log-normal and Pareto, are selected less often than light-tailed predictors. For Poisson regression with log link function, skewed heavy-tailed predictors such as Gamma, Pareto and log-normal are selected more often than other predictors by most methods. Adaptive-Lasso is more heavily influenced by the distribution of the predictors, and always struggles to select long-tailed predictors, even for Poisson regression. By contrast, Adaptive-Lasso is more able to select predictors with finite support, such as uniform or beta.

The effect of marginal distribution of predictors is relatively small for Gaussian regression, but is much larger for logistic regression and Poisson regression with log link. In the case of the Poisson regression with log link, for Lasso, this might be due to the different signal strengths of true variables, rather than a bias in variable selection.

Stability, being based on a consensus estimate from multiple Lasso variable selections, amplifies the differences in variable selection based on marginal distribution of predictors.

Even at the highest MI level considered in this study, Stability is almost never able to select log-normal distributed predictors.

SuRF is less affected by the marginal distribution of predictors than Lasso and Stability. This means that SuRF performs by far the best when at least one true predictor follows a log-normal or Pareto distribution for logistic regression. For Poisson regression with log link, SuRF performs similarly to Stability, though it is easier to select less conservative models.

For the logistic regression, performing a Box-Cox transformation on all predictors before variable selection greatly improves results when the true predictors are log-normal or Pareto distributed. When the true predictors follow a t-distribution, the Box-Cox transformation does not improve results. Note that the true probability was simulated as a logistic function of the predictors, so logistic regression on the Box-Cox transformed predictors is misspecified. Despite the misspecification, variable selection is still far better when performed on the Box-Cox transformed predictors. This means that the marginal distribution of predictors has more impact on variable selection than the correct specification of the model.

For the Poisson regression, the Box-Cox transformation makes the variable selection worse. This is not surprising, as the skewed predictors are easiest to select, and therefore, performing a Box-Cox transformation has the effect of both making the predictor have a more difficult distribution to be selected, and making the model misspecified.

A natural question is why the shape of the predictor distributions affects the variable selection to such an extent. More work is needed to fully understand the cause of this effect. However, the difficulty seems to be based on standardisation. It is standard practice when performing Lasso to standardise the predictors prior to variable selection, in order to make the method scale-invariant. This is done by dividing each predictor by its standard deviation. For Gaussian regression, this makes the coefficient of the standardised variable equal to the covariance between the predictor and the fitted value, which should give reasonable results. For heavy-tailed predictors, the sample variance of the predictor is more variable than for light-tailed predictors. Since, for a fixed coefficient, the sample variance of the predictor determines the total amount of signal in the data, the empirical SNR is more variable for heavy-tailed predictors than light-tailed predictors. This could account for the results in the Gaussian regression simulation.

For logistic regression and Poisson regression with a log link function, the standard

deviation of the predictor is not a good measure of its total influence on the regression, since, in the case of heavy-tailed predictors, the standard deviation is mostly due to a small number of outliers. In logistic regression, the influence of a single observation is limited, so for a heavy-tailed predictor, the total influence is lower than for a light-tailed predictor with the same standard deviation. For the log-normal distribution, a higher proportion of observations are very close to the mean, so the effect is stronger for the log-normal. For the Poisson regression, because of the log link function, large points are extremely influential. This explains why the skewed predictors are easier to select. For the t-distribution, half of the outliers are on the negative side, which means the Poisson mean is very close to zero, so there is little information for these data points, making it more difficult to select the predictor.

Since Stability is based on the frequency with which Lasso selects a predictor, when applied to subsamples, predictors which are underselected by Lasso are much more underselected by Stability. Furthermore, for heavy-tailed predictors, many subsamples do not include the influential observations needed to select these predictors. While it could be argued that a predictor selected because of a single observation is not very reliable, the fact that these outliers are used to standardise predictors means that the signal strength for the other observations is underrepresented.

By contrast, SuRF uses Lasso to identify the relative importance of the predictors, but bases the final variable selection on an hypothesis-test-based forward selection procedure. This forward selection procedure is not so strongly influenced by the distribution of the predictors. Therefore, SuRF is more often able to select heavy-tailed predictors. The danger is that a less heavy-tailed surrogate may be ranked above the true predictor, causing SuRF to select the surrogate instead. We see this in many of the simulation studies, where at high MI levels, SuRF selects a large number of "noise" variables. (See for example, Figure 3.9(c).) These are likely to be surrogates of the true predictor.

Future research is needed into methods to improve the variable selection of Lasso-based methods when some or all of the predictors have heavy-tailed distributions. Applying a Box-Cox transformation prior to variable selection, or using SuRF for variable selection can improve performance for logistic regression when some or all of the predictors follow heavy-tailed distributions. However, even with these methods, the ability to select heavy-tailed predictors still lags behind the ability to select light-tailed predictors with similar signal

strength, and the Box-Cox transformation works well for skewed heavy-tailed distributions, such as the log-normal or Pareto distribution, but is not effective for symmetric heavy-tailed distributions such as the t distribution. Furthermore, neither of these approaches resolves the variable selection bias for Poisson regression with log link. A natural attempt is to choose a different standardisation method that better accounts for the total influence of the observations in non-Gaussian regression. It is unclear exactly what form such a standardisation would take. From our other experiments which are not detailed in this chapter, standard robust measures of scale do not seem to work.

Chapter 4

Sub-sampling Ranking Forward selection for generalised additive models (SuRFgam)

4.1 Introduction

Generalised additive models (GAMs), introduced by Hastie and Tibshirini [25], provide a flexible option for modelling the non-linear relationship between the predictors and the response. Unlike linear models, GAM models don't require specifying a detailed parametric relationship between predictors and the response but use only smooth functions of low complexity as explained in [67]. Consider a dataset with n observations, including the response variable denoted by $y_i \in \mathbb{R}$ and predictors x_{ij} for the i^{th} observation where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, P$. A GAM models the transformed conditional mean of the response as sum of smooth functions of the predictors:

$$\eta\left(\mathbb{E}(y_i|x_{ij})\right) = \sum_{j}^{P} f_j(x_{ij}) = \sum_{j}^{P}\sum_{k=1}^{m} b_{jk}(x_{ij})\beta_{jk} \tag{4.1}$$

where η is a continuous monotone link function, and $f_j(x_{ij})$ is fitted from a semiparametric family of smooth functions, usually cubic splines. Cubic splines are twice differentiable piecewise cubic functions. The points at which the functions are not cubic are called *knots*, and choice of number and placement of knots can influence performance of the method. The functions f_j are chosen to minimise the negative log-likelihood plus a smoothness penalty on each f_j. In theory, the choice of basis functions b_{ij} shouldn't influence the results, but some choices such as B-splines greatly simplify the computation, resulting in better optimisation. For a full discussion of various spline fitting methods, see [67].

In many real world problems, the number of predictors can be significantly larger than the sample size, and it is necessary to select a sparse model including only the most important predictors. This is the problem addressed in this chapter. A common approach to variable selection problems is to maximise penalised log-likelihood or equivalently to minimise a

penalised loss function:

$$\underset{f_1,\ldots,f_P}{\arg\min}\, l(y; f_1, \ldots, f_P) + \sum_{j=1}^{P} J(f_j) \tag{4.2}$$

with a penalty designed so that the fitted effects of noise variables become zero. For example, Lasso uses an L^1 penalty on the coefficients of a GLM to shrink coefficients to zero. This idea extends easily to the GAM case, often using a group-Lasso style penalty, $J(f_j) = \sqrt{\sum_{k=1}^{K} \beta_{jk}^2}$ in addition to the smoothness penalty used by GAM methods without variable selection. Methods based on this idea include Gamsel, RGAM, RGAM_SEL and SPAM.

Research on variable selection for GLMs has shown that penalty-based methods often select a relatively large number of variables, many of which are false positives. This can impair both prediction and interpretation. Furthermore, variable selection results can be unstable, sometimes based on a single outlier. Variable selection methods based on subsampling, such as SuRF and Stability, are much more conservative, often not selecting a single false positive. In sparse cases, this can result in better prediction and interpretation. Both SuRF and Stability apply Lasso to a large number of random subsamples of the data, and use these results to identify the variables most frequently selected for these subsamples. Stability then selects all variables that are selected in more than a fixed proportion of these random subsamples. SuRF incorporates an additional forward-selection step, using the selection frequency to decide the order in which to attempt to add variables to the model. This forward-selection step has the benefit of making SuRF more robust to long-tailed predictors, which are underselected by Lasso-based methods in some cases, and overselected in others (see Chapter 3).

This chapter develops a novel extension of the SuRF methodology to the GAM case. This method is able to select very sparse models with non-linear relations between predictors and response variables. We demonstrate the ability of our new SuRFgam method to select the true variables while strictly controlling the false positives across a range of scenarios.

The remainder of the chapter is structured as follows: in Section 4.2, we conduct a comprehensive review of recent related developments in selecting sparse additive models, with a particular focus on the methods applicable to the high dimensional case. We also review methods designed to control the noise variables in linear models. Section 4.3 provides

a detailed introduction to our proposed method, called Sub-sampling Ranking Forward selection for generalised additive models (SuRFgam). In Section 4.4, we present extensive simulation studies conducted across various data settings. These experiments compare the performance of SuRFgam in variable selection and prediction, in both Gaussian and Binomial models, against several other existing methods. Section 4.6 presents the analysis of a real dataset containing 57 predictors collected from 4061 emails and spam samples, originally used in the illustration in the book "Elements of Statistical Learning" [26]. Finally, the chapter concludes with a discussion of the results and potential future research work in Section 4.7.

4.2 Review of related work

4.2.1 Existing sparse additive models

Numerous methods have emerged for fitting sparse GAM models in the context of $N \ll P$ over the past two decades, based on the introduction of an L_1 penalty, as in Lasso. Different methods include different penalty terms $J(f)$ in the objective function. One pioneering method in sparse GAMs is the Component Selection and Smoothing Operator (COSSO) introduced by Lin[34]. Unlike traditional spline methods that used a squared norm, COSSO adopts a penalty that consists of a sum of L_1 norms on functions. The formulation of $J(f)$ is built based on a reproducing kernel Hilbert space (RKHS) and the objective function is expressed as

$$\underset{f_1,\ldots,f_P \in RKHS\mathcal{F}}{\arg\min} \; l(y; f_1, \ldots, f_P) + \tau_n^2 J(f) \tag{4.3}$$

with $J(f) = \sum_{\alpha=1}^{P} \|\mathcal{P}^\alpha f\|$ and τ_n is the smoothing parameter in this model. $\mathcal{P}^\alpha$ is the orthogonal projection of f onto RKHS $\mathcal{F}^\alpha$ where $\mathcal{F}^\alpha$'s are the main effect spaces. This penalty is similar to a group Lasso penalty and thus in special cases it is equivalent to the Lasso penalty. SpAM (Sparse additive models [45]) instead uses a functional version of the Group Lasso[74] penalty defined as $J(f) = \sum_{j=1}^{P} \sqrt{E\{f_j^2\}}$.

More recently, several methods have been developed to not only select the predictors, but also distinguish between linear and non-linear predictors at the same time. Gamsel

(Generalised additive model selection [11]) minimises the objective function

$$\begin{aligned}\underset{f_1,\ldots,f_p}{\arg\min}\frac{1}{2}\Big\|y-\alpha_0-\sum_{j=1}^{p}\alpha_j x_j-\sum_{j=1}^{p}U_j\beta_j\Big\|_2^2 \\ \underbrace{+\lambda\sum_{j=1}^{p}\Big(\gamma|\alpha_j|+(1-\gamma)\|\beta_j\|_{D_j^*}\Big)}_{\text{selection penalty}}+\underbrace{\sum_{j=1}^{p}\frac{1}{2}\phi_j\beta_j^T D_j\beta_j}_{\text{end-of-path penalty}}\end{aligned} \tag{4.4}$$

where the U_j is the matrix of evaluations of the basis functions b_{jk} at the observed points x_i, i.e. $f_j(x_i)=\alpha_j x_i+b_j(x_i)^T\beta_j$ where b_j is a vector of basis functions. As linear functions are a special case of cubic splines, the parametrisation is redundant. This redundancy has the effect of allowing a separate penalty on the linear coefficients $|\alpha_j|$ (a Lasso penalty) and the nonlinear coefficients $\|\beta_j\|$ (a group Lasso penalty by $\|\beta_j\|_{D_j^*}$), so that the model can select one or both. The end of path penalty is the standard quadratic smoothness penalty. Gamsel classifies each predictor to be in one of the three categories: null ($\alpha=0,\beta=0$), linear ($\alpha\neq 0,\beta=0$) and non-linear ($\beta\neq 0$).

Two alternative methods capable of selecting both linear and nonlinear predictors are SPLAM (Sparse Partially Linear Additive Models [38]) and SPLAT (Sparse partially linear additive trend filtering [43]). SPLAM is a form of hierarchical group Lasso, encompassing both the SPAM model (by setting $\alpha=1$) and Lasso model (by setting α to a small value and configuring a specific λ) as special cases in its objective function:

$$\underset{f1,\ldots,f_P}{\arg\min}\, l(y;f_1,\ldots,f_P)+\underbrace{\lambda\sum_{j=1}^{p}\Big(\alpha\|\beta_j\|_2+(1-\alpha)\|\beta_{j,-1}\|_2\Big)}_{\text{J(f)}} \tag{4.5}$$

where $\beta_j=(\beta_{j1},\beta_{j,-1})$ with β_{j1} denoting the coefficient for the linear basis function and $\beta_{j,-1}$ including all coefficients for the non-linear basis functions. This method excels in scenarios where the true variables comprise a mixture of linear and nonlinear predictors. However, in settings where all predictors are linear or all predictors are non-linear, SPLAM is generally outperformed by Lasso or SpAM [45].

SPLAT takes a trend filtering technique used in FLAM [49]

$$\underset{f1,\ldots,f_P}{\arg\min}\, l(y;f_1,\ldots,f_P)+\underbrace{\alpha\lambda\sum_{j=1}^{p}\|D^{K+1}P_j\gamma_j\|_1+(1-\alpha)\lambda\sum_{j=1}^{p}\|\gamma_j\|_2+\tilde{\lambda}\sum_{j=1}^{p}\|\theta_j\|_2\Big)}_{\text{J(f)}} \tag{4.6}$$

where $\theta_j = x_j\beta_j + \gamma_j$ with γ_j capturing all non-linear effect. The first term in the above penalty is a Lasso penalty on $(K+1)^{th}$ order difference of the permuted (ordered) γ_j which controls the complexity of the nonlinear function γ_j; the second term in $J(f)$ is a group Lasso penalty on the non-linear terms; and the third penalty is a group Lasso penalty on θ_j for variable selection purposes. Compared to SpAM and SPLAM, SPLAT provides the additional advantage of allowing a flexible fitting of the nonlinear function with adaptively chosen knots and fewer degrees of freedom.

RGAM (Reluctant GAM [56]) is a three-step algorithm. In the first step, a linear model of y on x with Lasso penalty is fitted, and the residuals are computed; In the second step, a smoothing spline $\hat{f}_j$ is fitted using residuals obtained from the first step on each variable x_j and the resulting functions $\hat{f}_j's$ are scaled; Finally, y is fitted on X and F combined, where $F_{ij} = \hat{f}_j(x_i)$. RGAM can be modified to RGAM_SEL by a slight adjustment in the second step. This modification restricts the fitting of smoothing splines only to the non-zero linear predictors from the first step.

4.2.2 Existing work in variable selection with a false positive control

Sub-sampling is a widely used approach in variable selection to control the false positive rate, particularly in ultra-high dimensional scenarios. The idea is that important predictors should be selected in a vast majority of sub-samples of the original data. This can avoid selecting the variables due to a few outliers and maintains a set of reliable set of predictors. Stability selection [39] and SuRF (Chapter 2) are both built on the sub-sampling technique using Lasso for the variable selection on subsamples. In the Stability selection procedure, the data is split randomly into two equal samples B times. Each time, all predictors are fit by Lasso at a pre-specified and fixed grid of penalty parameter values. The variable is selected by Stability if the proportion of the B subsamples for which it is selected by Lasso exceeds a pre-specified cut-off π_{thr}, usually in the range $(0.6, 0.9)$. By choosing the penalty parameter in the Lasso to select a fixed number, q, of variables, the per-family error rate (PFER) is controlled by $E(V) \leq \frac{q^2}{(2\pi_{thr}-1)p}$, where V is the number of noise variables selected, and p is the total number of predictors.

SuRF uses a similar sub-sampling procedure to construct a ranked list of predictors, from the most frequently selected to the least. SuRF then uses this list to perform an ANOVA forward selection step to formally test whether each selected variable significantly

improves the fit of the model. The fit is assessed using log-likelihood ratio, and the critical value, accounting for multiple testing, is computed via a permutation test.

4.3 Method

We develop a two-step method, SuRFgam, based on a sub-sampling approach, a regularised generalised additive model and forward selection. A similar framework employed in the SuRF method has been successful in selecting linear predictors, maintaining a low false positive rate and providing a very competitive prediction performance in various Generalised linear models (GLM). To select non-linear predictors in subsamples, SuRFgam uses Gamsel instead of Lasso to perform variable selection on the subsamples.

In the first step, we randomly select B (typically 1,000) subsamples from the data. The size of these subsamples can be changed, with a range 50–90% of the original dataset producing reasonable results. We use larger proportions when the original dataset is small, in order to ensure all subsamples have reasonable size. For each subsample, we use Gamsel, with fixed $\gamma = 0.7$, and $\lambda = \lambda_{\min}$ selected to minimise cross-validated error, to select variables. The value $\gamma = 0.7$ was found to produce good results. This value is larger than the recommendations of the authors of Gamsel, who recommend $\gamma \leqslant 0.5$. However, this recommendation is designed to enable selection of linear predictors, whereas SuRFgam does not distinguish between linear and nonlinear predictors, so this criterion is not so important. Furthermore, larger values of γ tend to result in less sparse models, which is desirable, because the later stage of the SuRFgam method further reduces the set of selected variables, so selecting sparse models on the subsamples can reduce the ability to select the true predictors. We then order the predictors by the number of subsamples in which they are selected ($\alpha \neq 0$ or $\beta \neq 0$) by Gamsel.

In the second step, a sequential forward selection via permutation tests is used to select variables. The null hypothesis H_0 of the permutation test is that no variables beyond the currently selected set are good predictors for the response variable. The active selected variable set starts as an empty set. For each new selection, for each candidate variable, the test statistic is a log-likelihood ratio between the current GAM model and the model having this candidate variable added, using a thin-plate spline basis with a dimension k for each variable in the GAM models. The number of basis functions is fixed at dimension $k = 6$. To find the appropriate critical value, a permutation is applied to all data samples on all

variables, excluding the ones already selected and the response variable. We then compute the log-likelihood ratio for each permuted predictor, and take the largest of these values for each permutation. We perform a large number (usually, $C = 200$) of permutations, obtaining a log-likelihood ratio for each one, then take the $100(1-\alpha)$ percentile, $D_{1-\alpha}$ of these values, where α is the chosen significance level for the test, to get the critical value for the log-likelihood ratio test. Permuting samples in this way breaks the association between the candidate variables and the response variable while reserving the relationship between the candidate variables, so that the null hypothesis holds for the permuted samples. The first variable on the ranked list whose log-likelihood ratio statistic exceeds the critical value is added to the model, and the next forward selection step begins. When no likelihood ratio statistic exceeds the critical value, the method ends, and the current GAM is returned as the final selected model.

4.4 Simulation design

4.4.1 Design matrix

We first simulate a matrix, X_0, of 1000 predictors and $N = 100,000$ observations from a multivariate standard normal distribution with a co-variance matrix whose $(i,j)^{th}$ element is given by $Cov(X_0)_{ij} = \rho^{|i-j|}$ ($\rho = 0.8$), so that there is a high correlation among predictors in the adjacent columns and the correlation diminishes as the separation between the two variables increases.

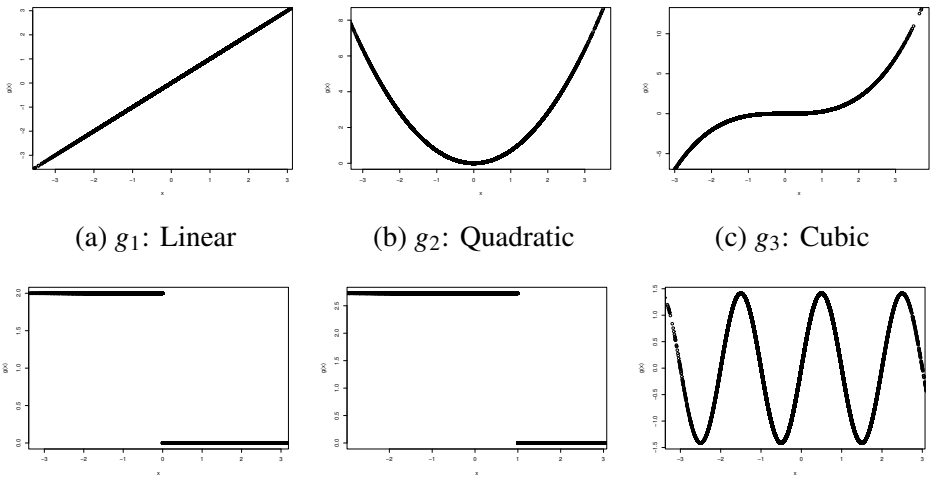
Furthermore, we divide the observations in X_0 into two equal parts: training samples X_{tr} and test samples X_{te}. The training samples are further divided into 100 replicates, X_{tr}^1, X_{tr}^2, ..., X_{tr}^{100}. Each replicate has $N = 500$ observations and $p = 1000$ predictors. The prediction outcomes will be assessed on a large test dataset X_{te}. From each replicate, we randomly select 6 predictors as true predictors so that the correlation within the true predictors and correlation between the true predictors and the remaining predictors are all different. The training response variables are generated according to those true predictors in each replicate and 100 sets of the response variables in the test data are simulated based on the corresponding true variables in the test samples X_{te}.

To demonstrate the impact of the strength of the signal on the variable selection, the standard deviation of the noise term N(0, σ_{tr}) in the Gaussian model is set to 6 levels:

$\sigma_{\text{tr}} = (1, 1.2, 1.5, 1.8, 2, 2.5)$. Similarly, we generate the binary responses at 4 levels: $SNR = (0.7, 1, 3, 5)$ in the logistic regression. Both models share the same training and test X matrices. The choices of the function of each true predictor in the model are as follows:

$$
\begin{aligned}
f_1(x) &= x && \text{Linear} \\
f_2(x) &= x^2 && \text{Nonlinear; quadratic} \\
f_3(x) &= x^3 && \text{Nonlinear; cubic} \\
f_4(x) &= \begin{cases} 0 & \text{if } x > 0 \\ 1 & \text{if } x \leqslant 0 \end{cases} && \text{Nonlinear; step function} \\
f_5(x) &= \begin{cases} 0 & \text{if } x > 1 \\ 1 & \text{if } x \leqslant 1 \end{cases} && \text{Nonlinear; step function} \\
f_6(x) &= \sin(\pi x) && \text{Nonlinear; periodic function}
\end{aligned}
$$

These functions cover a range of different types of linear and non-linear behaviour, shown in Figure 4.1.



(a) g_1: Linear (b) g_2: Quadratic (c) g_3: Cubic

(d) g_4: Step (0) (e) g_5: Step (1) (f) g_6: Periodic $\sin(\pi x)$

Figure 4.1: True functions in the simulations

Functions f_1–f_3 are polynomial, and can therefore be perfectly modelled by cubic splines, while functions f_4 and f_5 are discontinuous, so can only be crudely approximated by a spline. f_6 is smooth, so can be fairly well approximated by a spline. However, because of the periodic nature, the smoothness penalty is relatively high, making it difficult to select predictors with this functional relationship. Also note that since each X_i is simulated from

a standard normal distribution, for f_2 and f_6 there is no correlation between X_i and $f_j(X_i)$, meaning there is no linear signal for these predictors. This should make these predictors particularly challenging to select for linear methods.

The Gaussian model can be written as

$$y^i_{\text{tr}} = \sum_{n=1}^{6} g_i(v_i) + N(0, \sigma_{\text{tr}})$$

where v_i, i=1,2, ...,6, represent the randomly selected true predictors in each replicate. $g_i(v) = f_i(v)/\tau$ is the scaled $f_i(v)$, chosen so that the non-linear terms f_2-f_6 have the same scale (mean and standard deviation). The scale factors τ=$\sigma(f(x))$ for these functions are $\sqrt{2}$, $\sqrt{15}$, $\sqrt{0.5(1-0.5)}$, $\sqrt{0.16(1-0.16)}$ and $\sqrt{\frac{1}{2}(1-e^{-2\pi^2})}$, respectively .

For the logistic regression model, we used the same coefficient β for all six true predictors $v_2 - v_6$ and these predictors are scaled in the same way as in the simulations for the Gaussian model. The coefficients β and β_0 are provided in the Table 4.1 for different SNRs. The proportion of 0's and 1's in the simulated training and test samples are reasonably balanced: roughly 50%.

Table 4.1: Simulation coefficients for the Binomial model at various SNRs

SNR	β_0	β	**percentage of 1's**
0.7	-3.9	0.91	46%
1	-5	1.17	46%
3	-11	2.68	48%
5	-16	4.10	52%

Additionally, we aim to investigate how the data dimension impacts the variable selection of SuRFgam. We therefore consider four subsets of each training data replicate with dimensions summarised in Table 4.2. X^i_1 represents the original training replicate itself. X^i_2, is a subset of X^i_1, for $i = 1, 2, \ldots, 100$ that includes all observations from X^i_1 but only 200 predictors including all six true predictors plus 194 randomly selected additional predictors. The same response data y is used for X_2 as for X_1. Both X^i_3 and X^i_4 are a subset of X^i_2 with reduced observations, 100 and 200, respectively. These observations are randomly drawn from X^i_2. The predictors in X^i_2–X^i_4 are the same. The response variable y^i_{tr} for replicates X^i_3 and X^i_4 are the corresponding subset of the response variables y^i_{tr} from the training data sets X^i_1 and X^i_2.

Case	Scenario	Training X	Training y	N	p	No. of replicates
Design X_1	$p \gg N$	$X_1^i (=X_{\text{tr}}^i)$	y_1^i $(=y_{\text{tr}}^i)$	500	1000	100
Design X_2	$N > p$	X_2^i	y_1^i	500	200	100
Design X_3	$N = p$	X_3^i	subset from y_1^i	200	200	100
Design X_4	$p > N$	X_4^i	subset from y_1^i	100	200	100

Table 4.2: Data matrices with four different dimensions

From these scenarios, X_2 is the easiest, having the joint largest number of observations and the smallest number of predictors. X_1 and X_3 are more challenging than X_2, having more predictors or fewer observations respectively. X_4 is more challenging than X_3, having fewer observations. These scenarios cover a range of cases, with X_1 and X_4 having $p \gg N$, X_2 having $N > p$ and X_3 having $N = p$.

4.4.2 Methods Compared

We compare SuRFgam with a number of variable selection methods, including both linear methods, which have been shown to work well in many non-linear cases [77], and non-linear methods. The linear methods compared are Lasso, SuRF, and Stability selection; the non-linear methods compared are Gamsel, SPAM, RGAM and RGAM_SEL. The overall true positive rate (TPR) and false positive (FPR) are compared to assess variable selection performance. The true positive rate for each true variable $\nu_1, \ldots, \nu_6$ is also compared and discussed.

For Stability Selection, the probability cutoff is usually set to the range 0.6–0.9. The higher the cut-off value, the fewer variables will be selected by the method. We only consider the cutoff 0.6 since previous research has shown that this cut-off is most comparable with SuRF. Higher cut-offs tend to be more conservative, making the methods incomparable in terms of variable selection. For Gamsel models, although $\gamma < 0.5$ is recommended, this recommendation is designed to improve identification of linear predictors, which is not assessed in our simulation study. Furthermore, our simulation settings are more complex. We therefore compared a wider range of γ values, ranging from 0.4 to 0.9. For SuRF, we used the significance levels $\alpha \in \{0.05, 0.1, 0.2\}$ in all settings. For SuRFgam, we used $\alpha \in \{0.05, 0.1, 0.2\}$ for scenarios of type X_1 and X_2; but for some of the scenarios of type X_3 and X_4, the variable selection results at these levels were incomparable with Gamsel and other methods, so we used a wider range for α, even comparing $\alpha = 0.9$ in some cases.

For SPAM, the R package does not provide an automated method to select the tuning parameters. We therefore performed variable selection using a grid of penalty parameters, and in each simulation, for each fixed number of true predictors 1–6, we chose the smallest penalty value which selects that many true values, and recorded the false positive rate. This assessment gives SPAM a huge advantage, selecting the best penalty parameter in each simulation. However, even with this huge advantage, SPAM is unable to outperform SuRFgam in the majority of scenarios.

We use trimmed prediction MSE (the largest 5% of simulation MSEs are removed) and misclassification error rate (MCER) to compare the prediction performance in the Gaussian and Binomial models respectively. Trimmed MSE avoids issues caused by a small number of outliers, where performances of some methods can be really bad, and focuses on the typical performance. For Lasso, Stability Selection, SuRF and SuRFgam, we fit GAM on the selected variables to predict the test samples. For Gamsel, RGAM and RGAM_SEL, we use their regularised models with cross-validations based on the training samples to predict test samples.

4.5 Simulation Results

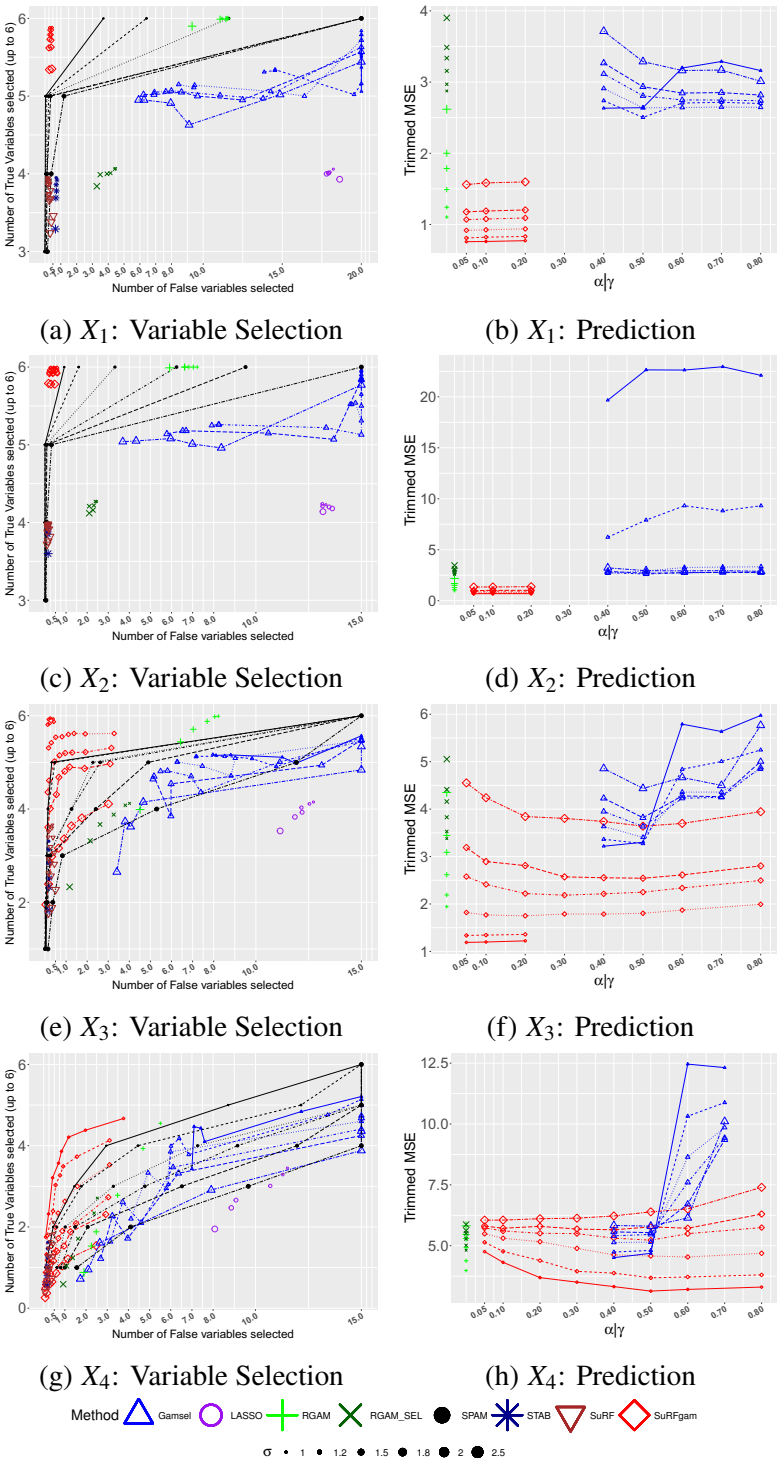
Detailed tables of results for all simulations are presented in Section 4.8. In this section, we provide figures to show the results, and discuss the findings for each scenario.

4.5.1 Gaussian Model

Design X_1 ***(N=500, P=1000)***

Data design in Scenario X_1 present a $p \gg N$ case with a large sample size and a large number of predictors.

Figure 4.2(a) compares the true positive and false positive rates for all methods, while Figure 4.2(b) compares the predictive accuracy. We see that SuRFgam significantly outperforms Gamsel, RGAM_SEL and the linear methods, with higher true positive rate and lower false positive rate. RGAM is able to achieve a higher true positive rate, but at the cost of a much higher false positive rate. The difference in true positive rate is small, so the large difference in false positive rate results in worse prediction. Recall that the SPAM package does not provide appropriate tuning methods, so we



(a) X_1: Variable Selection (b) X_1: Prediction

(c) X_2: Variable Selection (d) X_2: Prediction

(e) X_3: Variable Selection (f) X_3: Prediction

(g) X_4: Variable Selection (h) X_4: Prediction

Figure 4.2: Gaussian models with six true predictors $V_1 - V_6$: Left panels show the true positive versus false positive for variables selection of Lasso, STAB (cutoff 0.6), SuRF ($\alpha = 0.05, 0.1, 0.2$), Gamsel (γ ranges from 0.4 to 0.9), RGAM, RGAM_SEL and SuRFgam (α varies, up to 0.2 in X_1–X_2 and up to 0.8 in X_3–X_4) ; right panels shows the trimmed prediction MSEs on test data sets for non-linear methods only; results for different tuning parameters for each method at each signal strength level (σ =1, 1.2, 1.5, 1.8, 2, and 2.5) are linked. All results are averaged over100 data sets.

are comparing that method in a way that gives it an unfair advantage. Nevertheless, SPAM is unable to outperform SuRFgam at any SNR level, and for the low SNR case, is outperformed by SuRFgam in terms of both true positive and false positive rate.

Figure 4.3 breaks down the true positive rate by functional form of the relation between predictor and response. As predicted, the linear methods and RGAM_SEL are unable to select V_2 and V_6, where there is no correlation between the predictor and response. The other variables are selected fairly reliably even by the linear methods, though the more conservative linear methods (Stability and SuRF) struggle to select V_3 and V_5 in the low SNR cases. The nonlinear methods (excluding RGAM_SEL) have no trouble selecting V_2, which has a fairly simple functional relationship between predictor and response. However, the more fluctuating relationship between V_6 and the response proves more difficult, and only RGAM is able to reliably select this variable in the low SNR cases. We see that SuRFgam and RGAM have similar rates of selecting V_1–V_5, and the difference in their true positive rates is almost entirely due to the difficulty of selecting V_6. Gamsel also selects V_1–V_5 reliably for all γ values at all SNRs, but almost never selects V_6 for $\gamma \leqslant 0.5$, and rarely selects V_6 for $\gamma \leqslant 0.8$ in the medium-to-low SNR case. Increasing γ above this level to enable selection of V_6 results in extremely high false positive rates.

Design X_2 (N=500, P=200)

Data design X_2 is an easier scenario, with fewer noise variables included. From Figure 4.2(c), we see that all methods have slightly higher true positive rates than in design X_1, and Gamsel and RGAM have substantially lower false positive rates. However, the comparison between methods tells a similar story to design X_1, with SuRFgam outperforming Gamsel by a large margin in terms of both true positive and false positive rates, and the linear methods massively underperforming in terms of true positive rate. Again, RGAM is able to achieve a slightly higher true positive rate at the cost of a much higher false positive rate. From Figure 4.2(d), we see that this trade-off results in a substantial increase in prediction trimmed MSE for RGAM compared with SuRFgam. In this scenario, the predictive performance of Gamsel has actually got worse than design X_1. Again, despite the assessment method giving an unfair advantage to SPAM, it is unable to outperform SuRFgam at any SNR level,

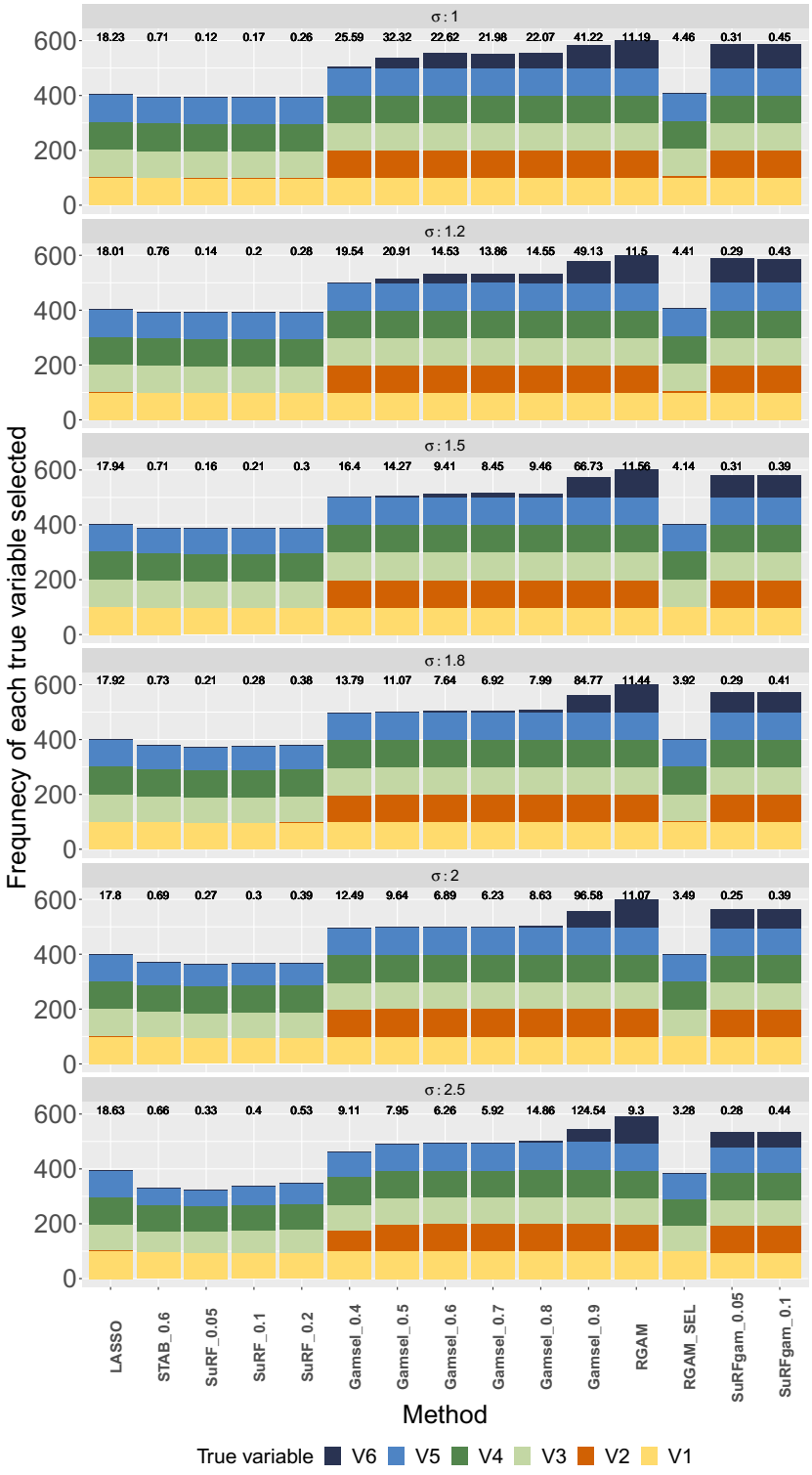


Figure 4.3: Frequency of six true predictors $V_1 - V_6$ being selected for the X_1 scenario (N=500, p=1000) in the Gaussian model. The bar-plot shows the selection frequency of each true predictor and the sum of all true predictors over 100 simulations. The average false positive rate over 100 simulations from the same method is displayed in text at the top of the bar-plot.

and for low SNR, it is outperformed by SuRFgam.

Figure 4.4 shows the frequency with which the true predictors are selected by each method. Results are very similar to the X_1 design, with all methods selecting V_1, V_3, V_4 and V_5 in the vast majority of cases, all nonlinear methods (excluding RGAM_SEL) selecting V_2 reliably, and Gamsel struggling to select V_6. In this design SuRFgam does better at selecting V_6, even at low SNR.

Design X_3 ***(N=200, P=200)***

Design X_3 has a much smaller sample size, which makes variable selection more challenging. In Figure 4.2(e), we see that all methods struggle in this design, except at the highest SNR. In this design, Gamsel and SuRFgam with low α are incomparable, with SuRFgam having lower true positive rate and lower false positive rate. However, by increasing the significance level α, we are able to compare the methods, and we see that SuRFgam outperforms Gamsel in terms of true positive and false positive rate. In the lowest SNR case, SuRFgam with $\alpha = 0.8$ also outperforms RGAM in terms of both true positive and false positive rate. At higher SNR, the methods are incomparable, with RGAM achieving both higher true positive rate and higher false positive rate. From Figure 4.2(f), we see that as in the X_1 and X_2 designs, this trade-off results in better prediction for SuRFgam. In this design, the linear methods are more competitive, having an advantage selecting variables with strong linear correlation, and a disadvantage selecting variables with little-to-no linear correlation. However, the linear methods still lag behind SuRFgam in terms of both true positive rate and false positive rate. In this design, SPAM is directly comparable with SuRFgam, for all SNR levels, and at all SNR levels, SuRFgam performs better, obtaining higher true positive and lower false positive rates than SPAM.

Figure 4.5 breaks down the true positive rate by functional form of the relationship. We see that V_6 is still the most difficult predictor to select, with most methods struggling to select it, particularly at low SNR. However, many of the other predictors are difficult to select. V_1 and V_2 are generally the most frequently selected by the nonlinear methods, with V_3–V_5 being more difficult to select. The linear methods are obviously more able to select V_1, which has a linear relation with the response, than the nonlinear methods. Furthermore, in some cases, the linear methods are more able

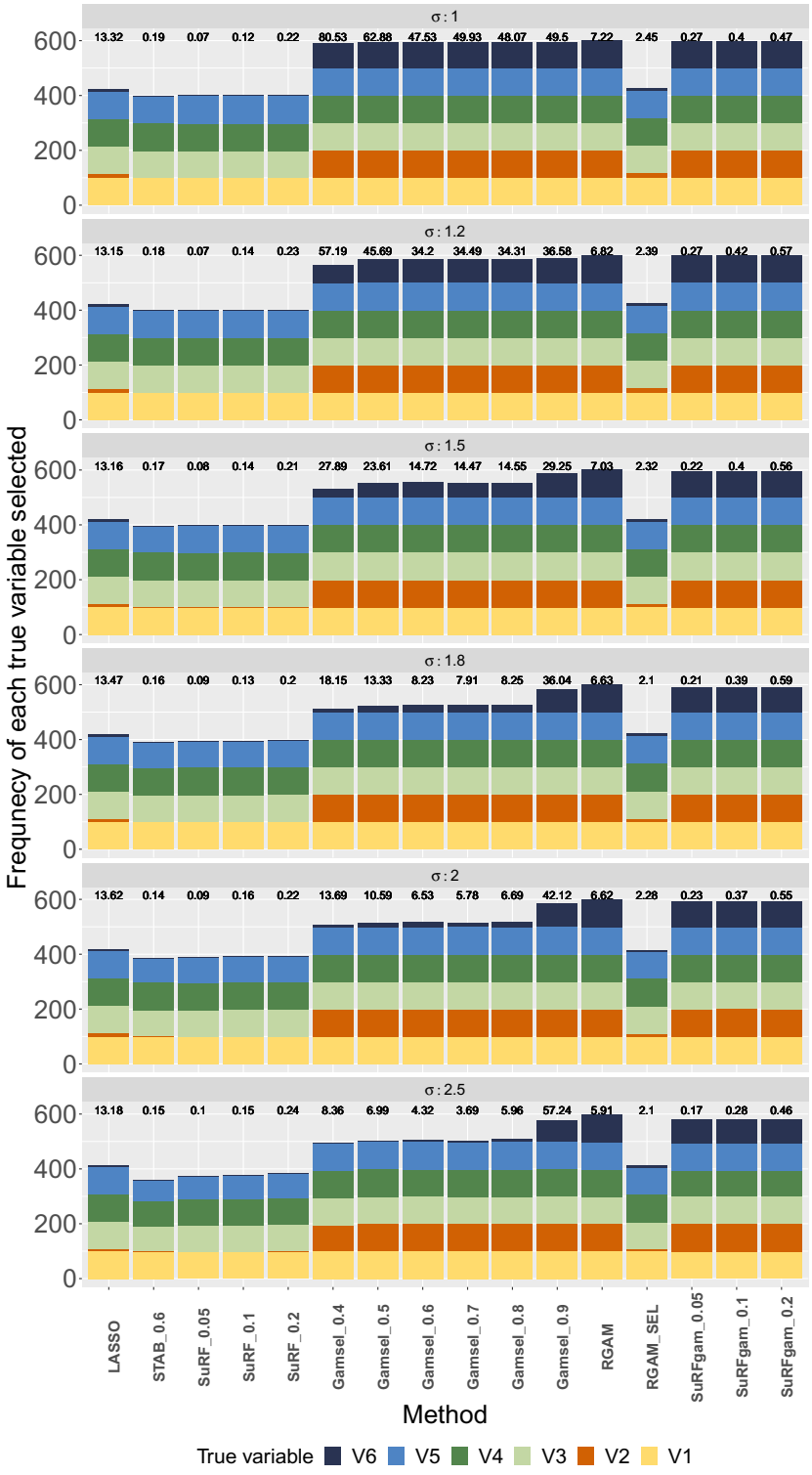


Figure 4.4: Frequency of six true predictors $V_1 - V_6$ being selected for the X_2 scenario (N=500, p=200) in the Gaussian model. The bar-plot shows the selection frequency of each true predictor and the sum of all true predictors over 100 simulations. The average false positive over 100 simulations from the same method is displayed at the top of the bar-plot.

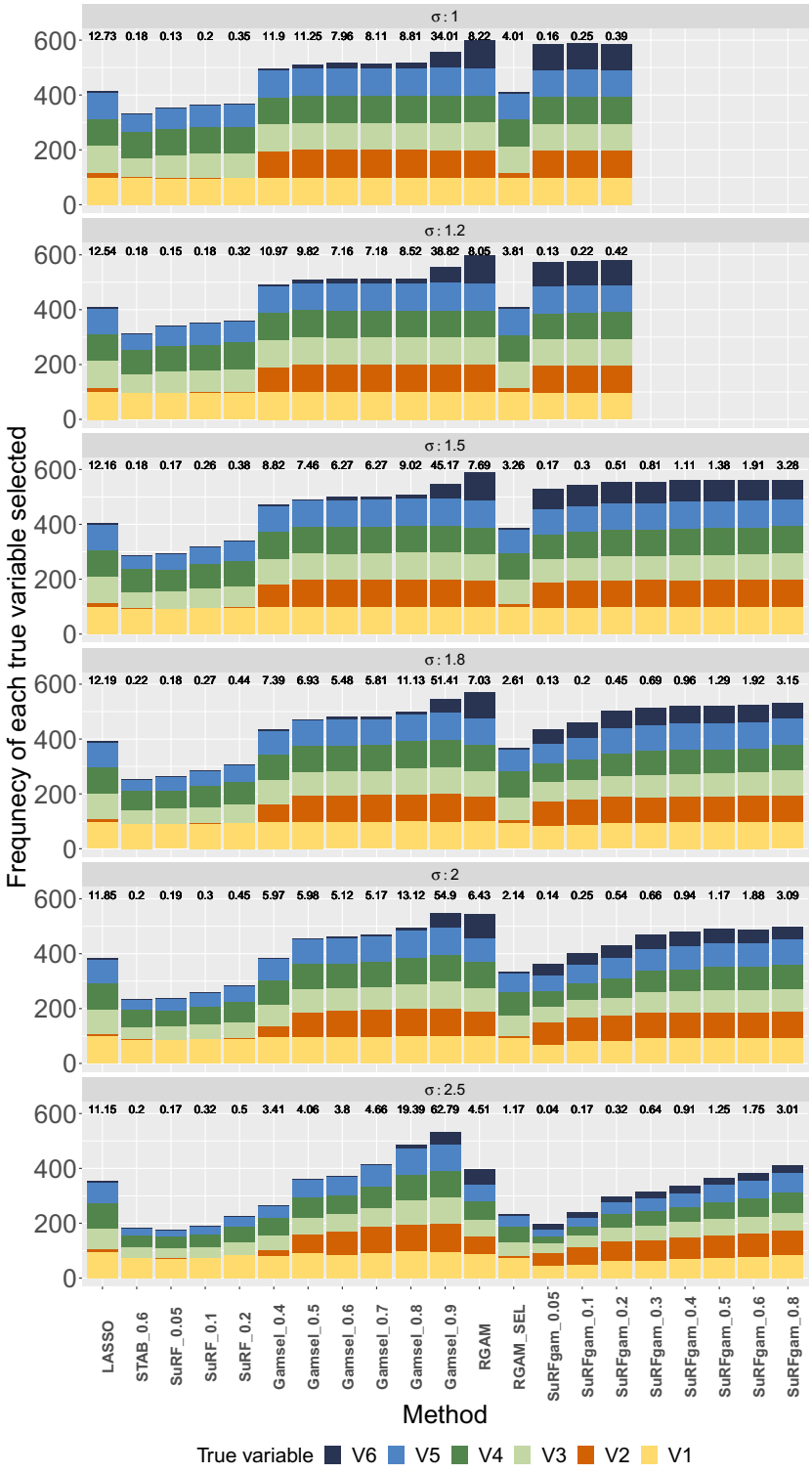


Figure 4.5: Frequency of six true predictors $V_1 - V_6$ being selected for the X_3 scenario (N=200, p=200) in the Gaussian model. The bar-plot shows the selection frequency of each true predictor and the sum of all true predictors over 100 simulations. The average false positive over 100 simulations from the same method is displayed at the top of the bar-plot. SuRFgam is performed at the significance level up to 0.2 when σ is 1 and 1.2; SuRFgam is performed up to 0.8 for the other values of σ.

to select some nonlinear predictors with strong linear correlation with the response than corresponding nonlinear methods. For example, with $\alpha = 0.05$, the linear SuRF selects V_4 significantly more frequently than the nonlinear SuRFgam.

Design X_4 (N=100, P=200)

Design X_4 represents the most challenging scenario among the four data designs due to its limited sample size. With half of the number of samples from X_3, all methods select fewer true variables. Variable selection and prediction results are shown in Figure 4.2(g) and (h) respectively. The parameter used in Gamsel and SuRFgam has a large effect on the true positive-false positive trade-off. For small significance level, SuRFgam is very conservative, and we increase the significance level up to $\alpha = 0.8$ to obtain comparable results to the other methods. At this level of α, SuRFgam outperforms RGAM, SPAM and Gamsel with $\gamma \leqslant 0.5$ in terms of both true positive and false positive rates. The linear methods are very competitive in these scenarios. Interestingly, at low SNR, RGAM is able to outperform SuRFgam in terms of prediction, despite having fewer true positives and more false positives. This discrepancy is because RGAM uses Lasso to shrink the fitted functions. In this case, with small sample size, even methods that select some of the true predictors rarely do much better than the null model, which selects no predictors. Even when RGAM selects a large number of noise variables, the shrinkage means that it fits a model close to the null model. Indeed, looking in more detail at the results, we see that RGAM selects no variables in over 60% of cases, but in other cases selects a large number of variables, many of which are noise variables. Because of the shrinkage, it performs well in these cases. On the other hand, SuRFgam selects more true variables, but the variance of the function estimation means that SuRFgam does not outperform RGAM by a wide margin in these cases, while when SuRFgam selects noise variables, it performs badly, because the final model has no shrinkage.

Figure 4.6 shows the true positive rates broken down by prediction variable. The results are similar to the X_3 design scenarios with V_6 being the most difficult variable to select under all methods. V_2 is the easiest variable to select for the nonlinear methods, and is almost never selected by the linear methods or by RGAM_SEL. V_4 and V_5 are harder to select than V_3 when the parameters α or γ are set to conservative

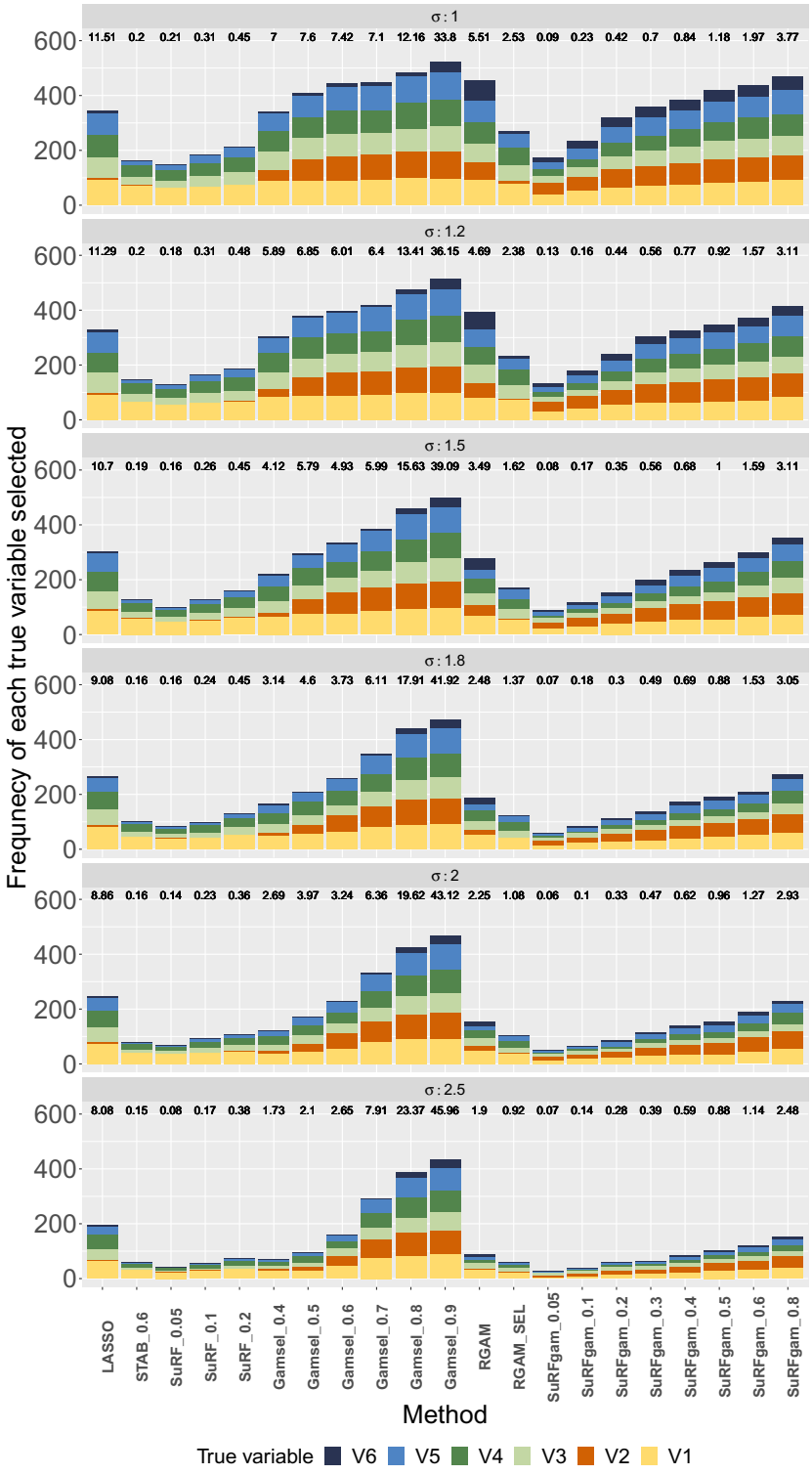


Figure 4.6: Frequency of six true predictors $V_1 - V_6$ being selected for the X_4 scenario (N=100, p=200) in the Gaussian model. The bar-plot shows the selection frequency of each true predictor and the sum of all true predictors over 100 simulations. The average false positive over 100 simulations from the same method is displayed at the top of the bar-plot.

low values, but become easier to select at the less conservative high values, whereas the frequency of selecting V_3 increases much less as α or γ increases. The linear methods are more able to select V_1 and V_4 than the nonlinear methods with comparable parameter settings, which is why these methods are competitive in these scenarios.

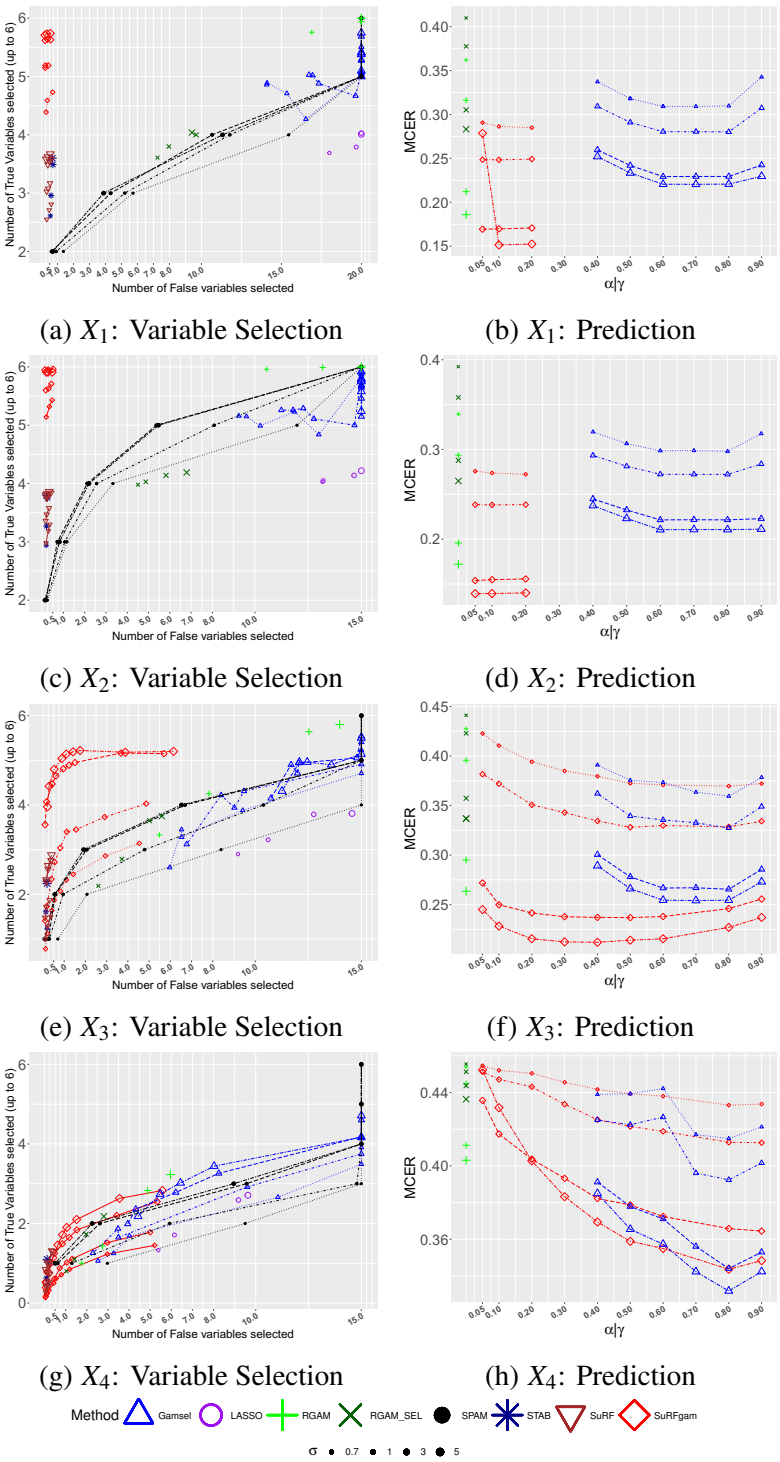
4.5.2 Binomial Model

Figure 4.7 shows the true positive versus false positive rates and misclassification error rates for all methods across all scenarios.

Design X_1 (N=500, P=1000)

As seen in Figure 4.7(a), SuRFgam is able to achieve a comparable true positive rate to Gamsel, but with a much lower false positive rate. RGAM is able to achieve a higher true positive rate, but at the cost of an extremely high false positive rate. From Figure 4.7(b), we see that this results in a higher misclassification error rate for RGAM. SPAM also has a much higher false positive rate for every true positive rate and SNR in this scenario. With this scenario being relatively easy, even in the lowest SNR cases, the nonlinear methods select more than 4 true predictors. As the linear methods never select V_2 or V_6, they are unable to compete in this scenario.

Figure 4.8 shows a breakdown of the true positive rates by variable type. We see that even at the lowest SNR, SuRFgam does well at selecting V_1, V_2 and V_4. V_1 and V_4 are well selected by all methods, even the linear methods. V_2 is well selected by all non-linear methods, with the exception of RGAM_SEL and Gamsel with $\gamma = 0.4$. SuRFgam reliably selects V_5 for SNR $\geqslant 1$, but sometimes fails to select it at SNR=0.7, particularly with $\alpha \leqslant 0.1$. RGAM and Gamsel with $\gamma \geqslant 0.5$ select V_5 more reliably, at the cost of much higher false positive rates. SuRFgam does not reliably select V_3 at SNR $\leqslant 1$, but is able to reliably select it at SNR $\geqslant 3$. RGAM and Gamsel are more able to select V_3 at low SNR. V_6 is even more challenging, with SuRFgam unable to reliably select this variable, even at SNR=5. However, Gamsel with $\gamma \leqslant 0.8$ struggles even more with this variable, selecting it in fewer than half the simulations, even at SNR=5, and rarely selecting it at all for SNR $\leqslant 1$. Gamsel with $\gamma = 0.9$ selects so many false positives, that it is not worth comparing. RGAM does well at selecting V_6, and is therefore a viable, less conservative alternative to SuRFgam that is able



(a) X_1: Variable Selection (b) X_1: Prediction

(c) X_2: Variable Selection (d) X_2: Prediction

(e) X_3: Variable Selection (f) X_3: Prediction

(g) X_4: Variable Selection (h) X_4: Prediction

Figure 4.7: Binomial models with six true predictors $V_1 - V_6$: Left panels show the true positive versus false positive for variables selection of Lasso, STAB (cutoff 0.6), SuRF ($\alpha = 0.05, 0.1, 0.2$), Gamsel (γ ranges from 0.4 to 0.9), RGAM, RGAM_SEL and SuRFgam (α varies, up to 0.2 in $X_1 - X_2$ and up to 0.9 in $X_3 - X_4$). False positive rates are censored at 20 in X_1 and at 15 in other designs, in order to better show most differences. Right panels shows the misclassification error (MCER) on test data sets for non-linear methods only; results for different tuning parameters for each method at each signal strength level (SNR= 0.7, 1, 3 and 5) are linked. All results are averaged over 100 data sets.

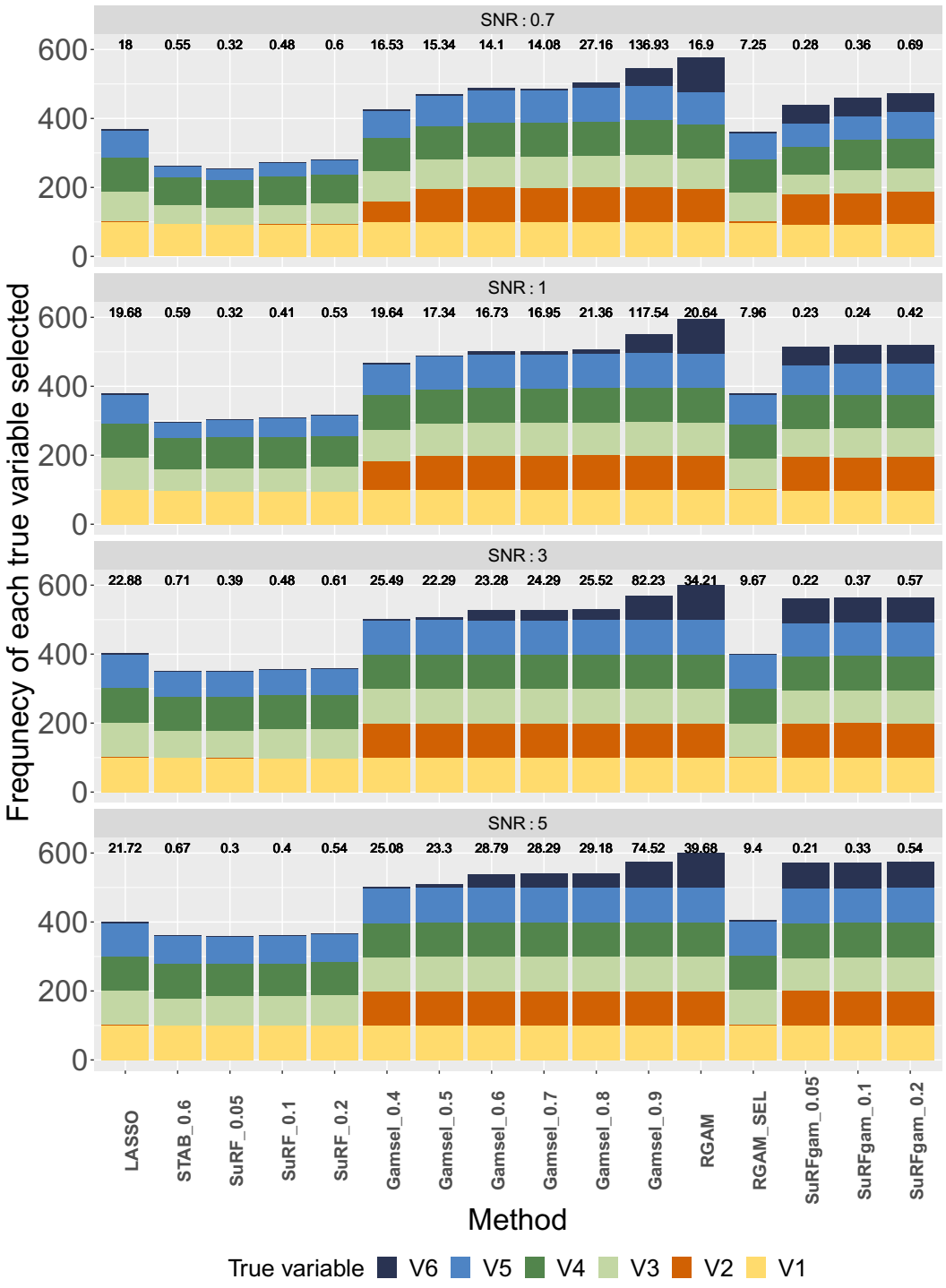


Figure 4.8: Frequency of six true predictors $V_1 - V_6$ being selected for the X_1 scenario (N=500, p=1000) in the Binomial model. The bar-plot shows the selection frequency of each true predictor and the sum of all true predictors over 100 simulations. The average false positive over 100 simulations from the same method is displayed at the top of the bar-plot.

to select predictors whose functional relationship has high smoothness penalty, if prediction is not the main aim of the variable selection.

Design X_2 (N=500, P=200)

Figure 4.7(c) and (d) tell a similar story to the X_1 design case. SuRFgam achieves comparable true positive rate to Gamsel, but with much lower false positive rate. RGAM achieves a higher true positive rate, but at the cost of an extremely high false positive rate, resulting in a higher misclassification error rate for RGAM. SPAM still has much higher false positive rate for every true positive rate.

Figure 4.9 also shows a similar story to the X_1 design, with SuRFgam reliably selecting V_1, V_2, and V_4 even SNR=0.7, and being less able to select the other variables at low SNR. However, in this design, SuRFgam is able to reliably select even V_6 for SNR $\geqslant 3$. Again, RGAM is able to reliably select all true predictors, even at lower SNR, but at the cost of high false positive rate. Interestingly, the false positive rate for RGAM gets larger in the large SNR case, where the problem is, in theory, easier. Gamsel, on the other hand, still struggles to select V_6 at higher SNR.

Design X_3 (N=200, P=200)

As the sample size decreases, making the problem more challenging, the true positive rate falls for all methods. Figure 4.7(e) shows that SuRFgam, even with $\alpha = 0.9$ is still more conservative than RGAM and Gamsel, with the results being incomparable between SuRFgam and RGAM, and SuRFgam outperforming Gamsel with $\gamma = 0.4$, and at higher SNR, also outperforming Gamsel with $\gamma = 0.5$, in terms of both true-positive and false-positive rates. Gamsel with higher values of γ is incomparable with SuRFgam, with higher true-positive and false-positive rates than SuRFgam, but is outperformed by RGAM. SuRFgam still outperforms SPAM in all cases where comparisons can be made. Figure 4.7(f) shows that, in most cases, more conservative variable selection provides better classification, with α in the range 0.4–0.6 providing the lowest MCER. The much less conservative methods, RGAM and Gamsel therefore do substantially worse at prediction. For the cases with SNR $\leqslant 1$, Gamsel actually outperforms RGAM, and even outperforms SuRFgam for one or two values of γ. For these cases, SuRFgam has the lowest false positive and true positive rates, while Gamsel has the highest for both rates, with RGAM falling in the middle. Thus, the

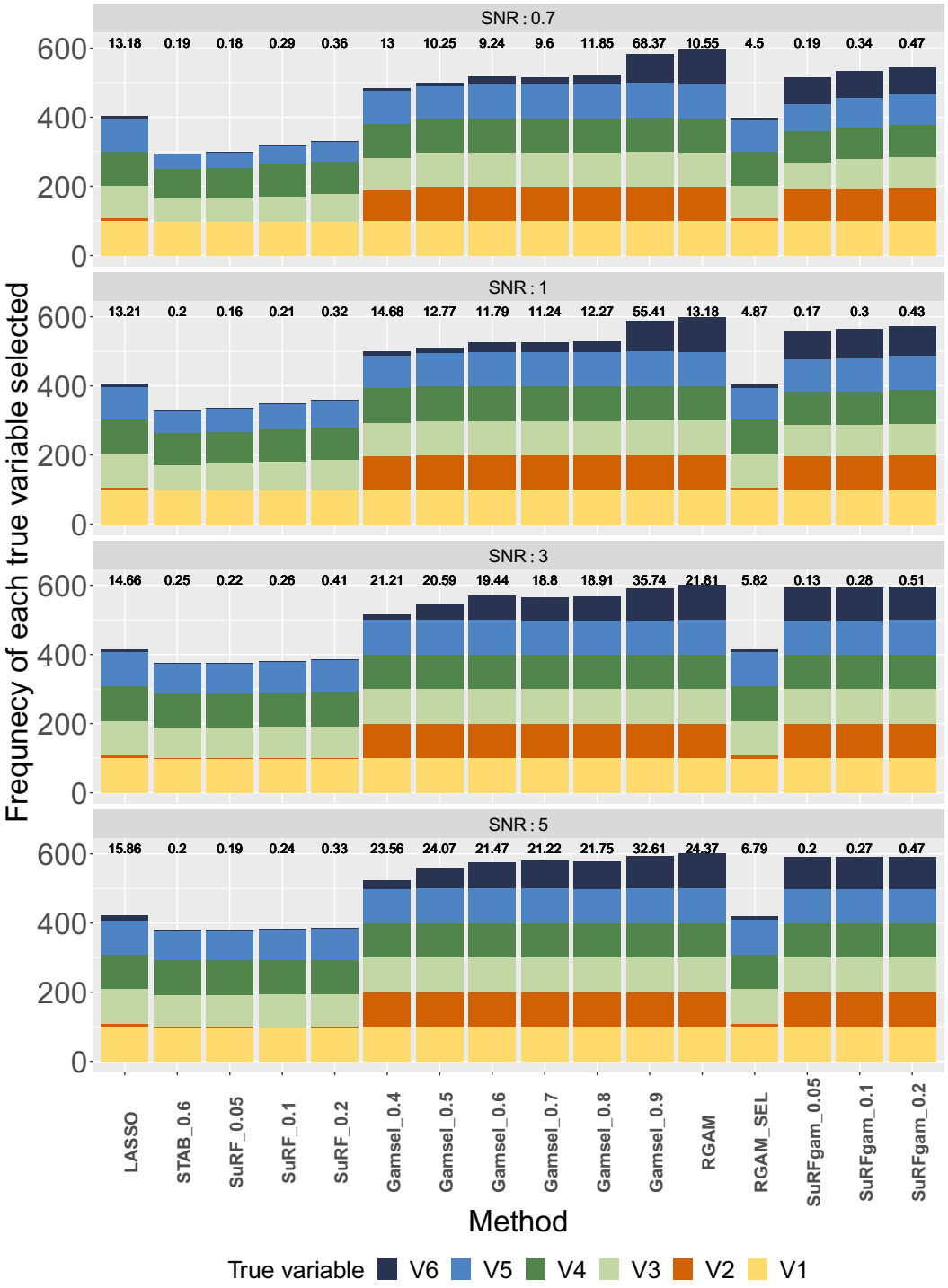


Figure 4.9: Frequency of six true predictors $V_1 - V_6$ being selected for the X_2 scenario (N=500, p=200) in the Binomial model. The bar-plot shows the selection frequency of each true predictor and the sum of all true predictors over 100 simulations. The average false positive over 100 simulations from the same method is displayed at the top of the bar-plot.

effect on prediction of the trade-off between true-positive and false-positive rate is not completely clear-cut. However, even in these cases, the difference in MCER between SuRFgam and Gamsel is very small.

Figure 4.10 shows that all methods struggle to select the true variables at lower SNR, with the same patterns of V_1, V_2 and V_4 being the easiest variables to select for SuRFgam and Gamsel. Interestingly, RGAM struggles to select V_2 at low SNR in this scenario, indicating that different methods are sensitive to different functional forms of the relationship between predictors and response. Again, V_6 is the most challenging variable to select for SuRFgam and Gamsel, but interestingly, RGAM has more difficulty selecting V_2, V_3 and V_5. Figure 4.10 is also interesting, because it shows a wider range of α values for SuRFgam. We see that V_1–V_5 all show a similar pattern of being selected more frequently as α increases. However, V_6 is still difficult to select, even at the highest α value. Indeed the frequency of selection of V_6 actually drops slightly for very high α values. This may be because V_6 is ranked fairly low by Gamsel, so there will often be noise variables ranked above it. At high α values, these noise variables have an increased chance of being selected, and after noise variables have been selected, it becomes harder to select V_6. There is also the risk of saturation of the model — for a binomial model with relatively small sample size, perfect separation is possible, after which no further variables will be selected. This can prevent selection of the variables that appear lower down the ranking list. This problem is actually more acute at higher SNR, since the stronger signal increases the chance of perfect separation.

Design X_4 (N=100, P=200)

From the comparisons in Figure 4.7(g), we see that this scenario is less clear-cut than other scenarios, with more comparisons between the methods possible, but some cases where SuRFgam performs slightly better in terms of true-positive and false-positive rates, and some cases where it performs worse than other methods. The differences are fairly small, indicating that SuRFgam, RGAM and Gamsel perform similarly in this respect. There are no cases where SPAM outperforms SuRFgam, but the comparisons are closer in a lot of cases. Looking at the MCER in Figure 4.7(h), we see that SuRFgam and Gamsel are fairly similar, with Gamsel being able to achieve

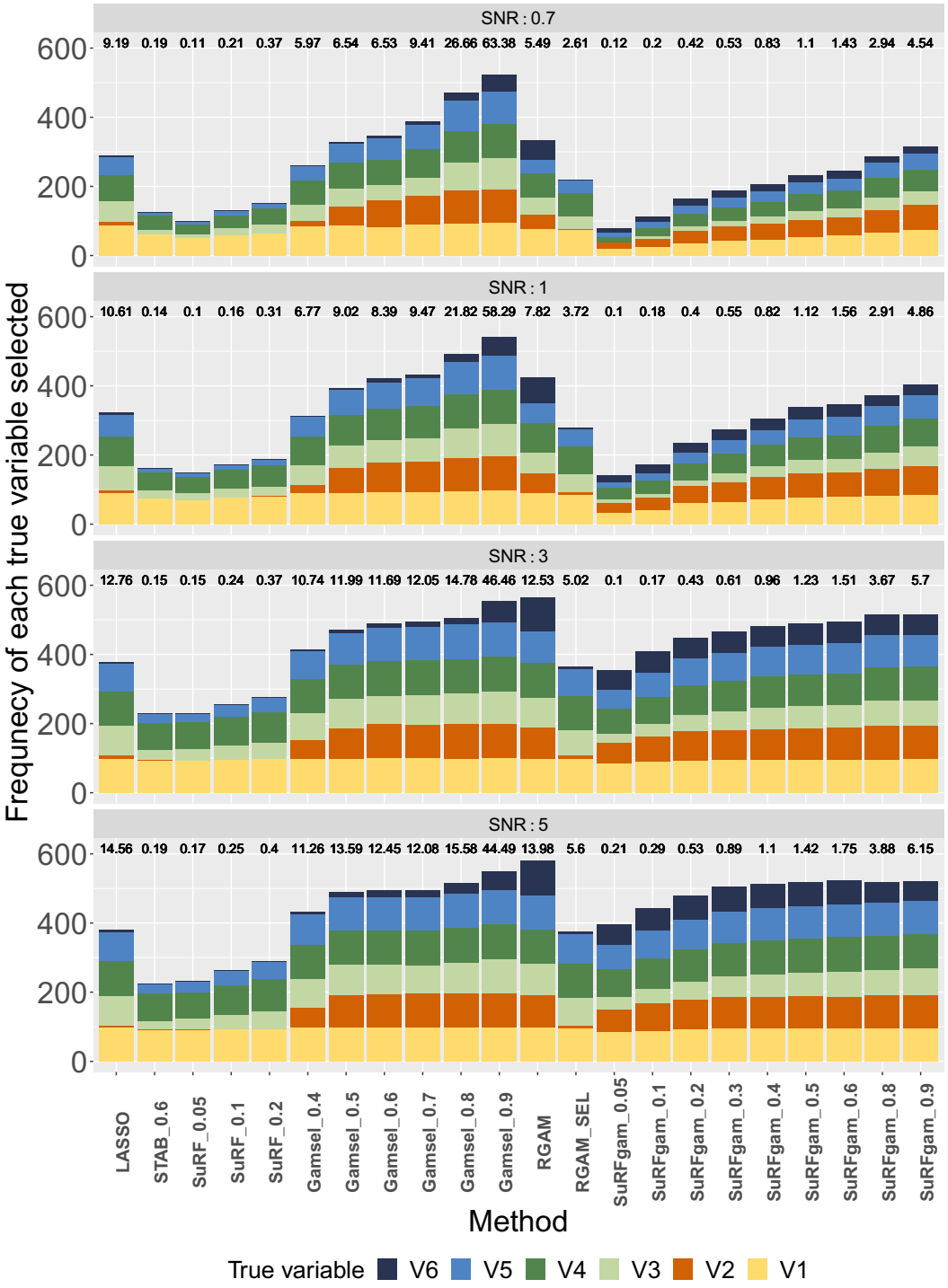


Figure 4.10: Frequency of six true predictors $V_1 - V_6$ being selected for the X_3 scenario (N=200, p=200) in the Binomial model. The bar-plot shows the selection frequency of each true predictor and the sum of all true predictors over 100 simulations. The average false positive over 100 simulations from the same method is displayed at the top of the bar-plot.

a slightly better MCER. Provided α is not chosen too conservatively, SuRFgam and Gamsel both outperform RGAM by a wider margin. This is despite RGAM being the best method in terms of true positive and false positive rate in at least some of the cases. This may be because of surrogate variables selected by SuRFgam or Gamsel, or may be because the shrinkage harms classification in these cases.

From the breakdown of true positives in Figure 4.9, we see that while the overall true positive and false positive rates can be comparable for certain tuning parameter values for all three methods, there are noteworthy differences between which true predictors are selected, indicating that the different methods are more sensitive to different functional relations between predictors and response. In particular, all methods are sensitive to the linear predictor V_1 and the symmetric step function V_4. However, SuRFgam and Gamsel also do well at selecting the quadratic predictor V_2 and the asymmetric step function V_5, while RGAM struggles more with these, particularly the quadratic V_2. On the other hand, SuRFgam and particularly Gamsel have great difficulty selecting the periodic V_6, while RGAM does very well at selecting this predictor. All methods do fairly badly at selecting the cubic predictor V_3.

4.6 Real Data Analysis

We re-analyse the spam email classification dataset from the book 'The Elements of Statistical learning' as an example. This dataset includes information from 4,601 emails with 57 predictors in total. Among these emails, 1813 emails are labelled as 'spam'. Each predictor corresponds to a proportion of specific words or characters within the email. We adopted a 20/80 training-test split for evaluating the performance of the variable selection and prediction. More specifically we divide the dataset into five folds, each containing approximately 920 observations. Each fold is used as training samples and the remaining four folds are test samples.

We compared the performance for Lasso, Stability Selection, SuRF, Gamsel, RGAM, RGAM_SEL, and SuRFgam. The evaluation of prediction performance is based on two metrics: the average misclassification error rate (MCER) and average number of selected variables. For Stability Selection, we explored a range of per-family-error-rate (PERF) from 1 to 15 (close to the maximum allowed for the data dimension). However, using a PERF

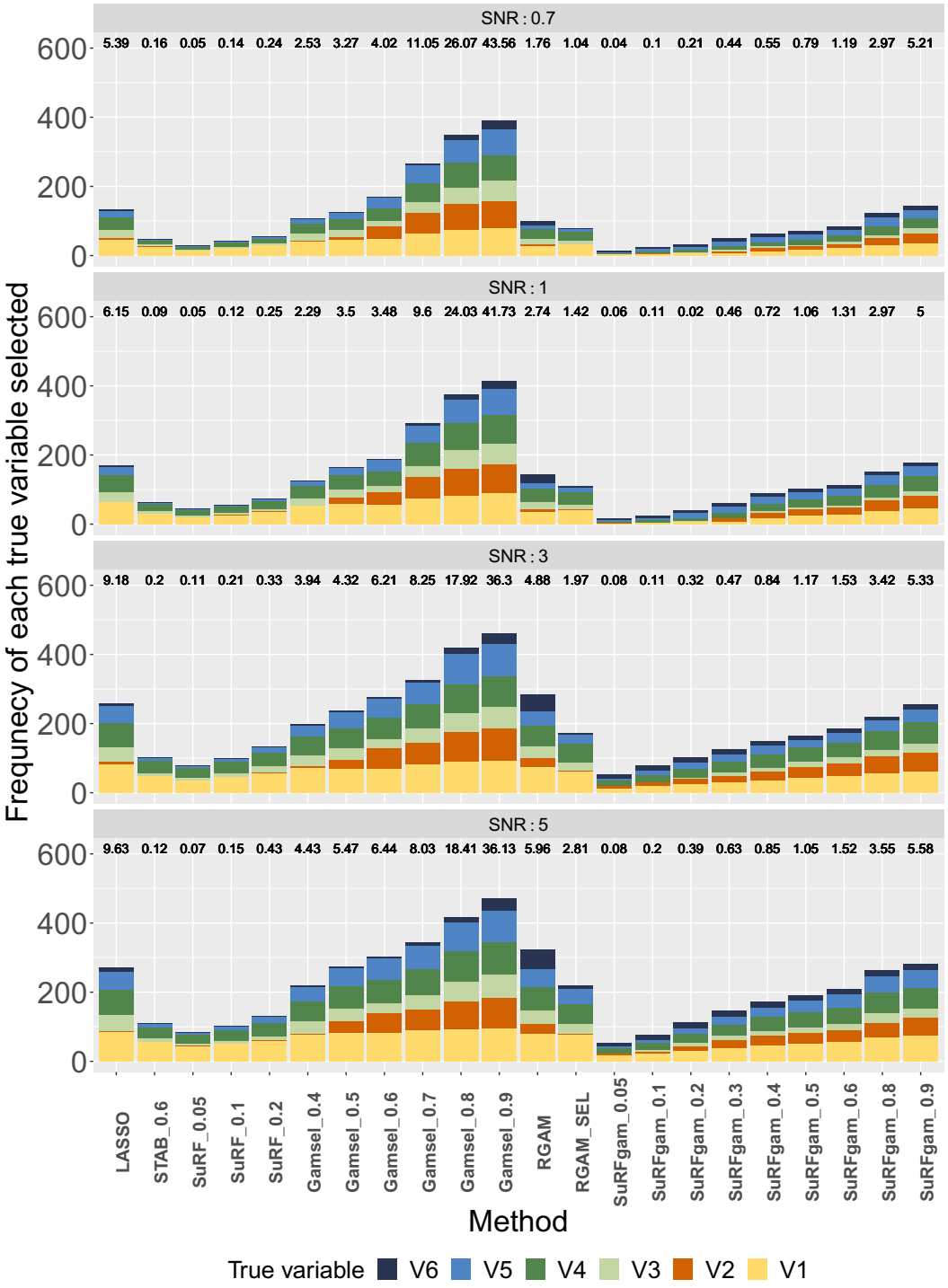


Figure 4.11: Frequency of six true predictors $V_1 - V_6$ being selected for the X_4 scenario (N=100, p=200) in the Binomial model. The bar-plot shows the selection frequency of each true predictor and the sum of all true predictors over 100 simulations. The average false positive over 100 simulations from the same method is displayed at the top of the bar-plot.

value smaller than 5 resulted in the selection of a very small set of variables and significantly higher MCER, making it not comparable with other methods. As a result, we use the PERF 6, 10 and 15 and a probability cutoff 0.6 for Stability Selection. In Gamsel, we consider a range of γ values from 0.4 to 0.9 as we used in the simulation. For SuRF and SuRFgam, the significance level varied from 0.05-0.9. We adopted 5 degrees of freedom and 4 basis functions when fitting the Gamsel and SuRFgam.

As shown in Figure 4.12, although the prediction MCER is low, Lasso consistently includes a majority of variables, between 44 and 54, in each fold and the average number of variables selected across 5 folds amounts to 49 out of 57 available predictors. Except for 'fold3', Gamsel also selects at least 30 variables in each fold and the average number of selected variables across the folds are between 32.8 and 44.2 for γ values of 0.4 and 0.9, respectively. RGAM performs similarly to Gamsel with a value of $\gamma = 0.4$ in terms of number of variables selected but its prediction MCER is less favourable than Gamsel. There is no significant difference between RGAM_SEL and RGAM. RGAM selects slightly more variables than RGAM_SEL with negligible differences in the prediction. In contrast, Stability Selection selects considerably fewer variables than any method, up to 15 approximately. Even with the largest PERF, its average prediction MCER remains the highest among all methods. SuRF and SuRFgam are very competitive in this case. Both methods can achieve similar prediction to those of Gamsel with a γ value of 0.6, but SuRFgam stands out by selecting fewer variables than SuRF. We also observe that increasing the significance level above about 0.5 or 0.6 in SuRF and SuRFgam does not significantly change the prediction MCERs, indicating that additional variables are not necessary. The results show the efficacy of SuRFgam in selecting important features that contribute to a good prediction while effectively controlling the total number of variables.

4.7 Conclusion

In this chapter, we have developed a new variable selection method, SuRFgam, for selecting predictors in a GAM. SuRFgam is based on the method SuRF (Chapter 2) for linear variable selection problems, which was shown to be an extremely competitive conservative variable selection method. SuRFgam is more conservative than other variable selection methods for GAMs. In principle the SuRFgam method can apply to any GAM, but our current implementation depends on an implementation of the Gamsel method that only offers

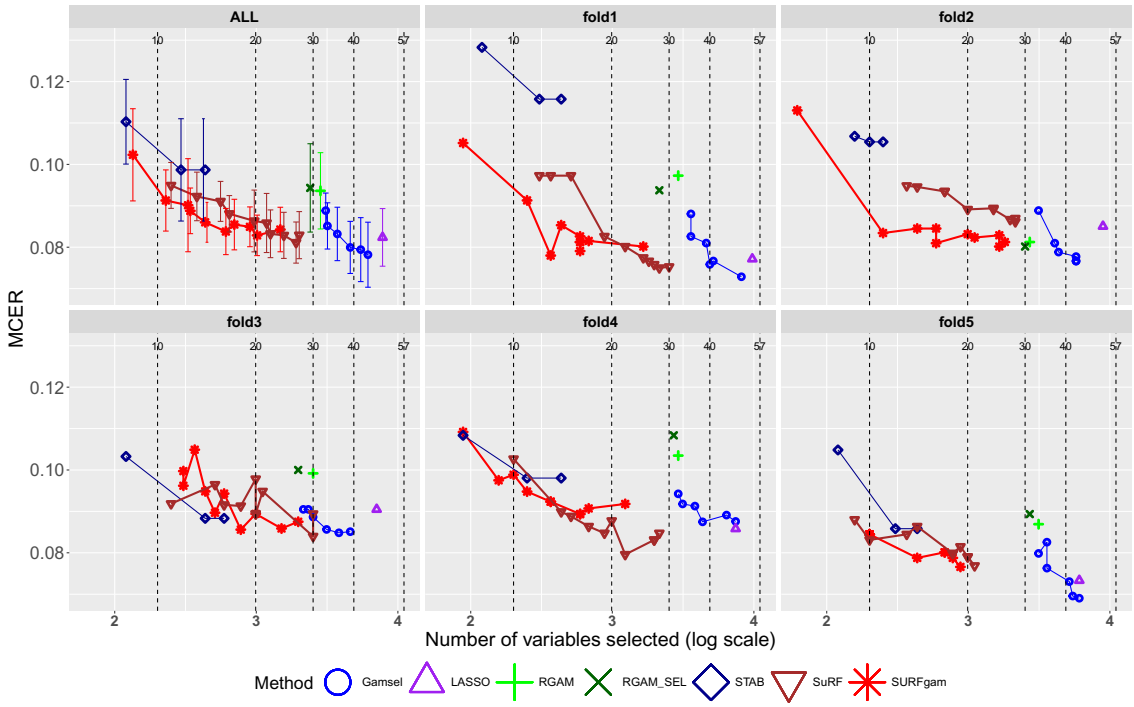


Figure 4.12: (Average) MCER vs the (average) number of selected variables The first subfigure shows average cross-validated MCER with 1 SD vs average number of selected variables across 5-folds for Lasso, STAB (PERF=6,10, and 15), SuRF (α ranging from 0.05 to 0.9), SuRFgam (α ranging from 0.05 to 0.9), RGAM,RGAM_SEL, Gamsel (γ ranging from 0.4 to 0.9). The rest shows test MCER vs average number of selected variables in each of 5 folds of data, respectively. 5 dotted lines from left to right represent the total number of selected variables at 10, 20, 30, 40 and 57 (all available variables).

Gaussian or Binomial models. We have compared the performance of SuRFgam on a large range of simulated scenarios, for both Gaussian and Binomial response, in terms of both true-positive rate and false-positive rate, and found that it performs very well. In many cases SuRFgam outperforms Gamsel in terms of both true-positive rate and false-positive rate, and is incomparable with RGAM. In a large majority of cases, SuRFgam outperforms both other methods in terms of prediction (trimmed MSE for Gaussian simulations, and MCER for binomial).

SuRFgam was far more conservative than Gamsel and RGAM across all simulations, always selecting fewer noise variables. In many cases, SuRFgam is also able to select more true predictors than Gamsel. RGAM nearly always selects more true predictors than SuRFgam, but also selects many more false predictors, and performs worse in terms of prediction in all scenarios. The different methods perform differently for different types of functional relation between predictors and response variables. SuRFgam outperforms Gamsel at selecting the predictor with a periodic relationship, and outperforms RGAM at selecting the predictor with a quadratic relationship in the difficult cases.

We also compared the performance of various linear methods for the GAM variable selection. In many GAM variable selection problems, linear methods can perform very competitively. This depends a lot on the type of functional relationship between the predictor and the response. In our simulations, we included two predictors with no linear correlation with the response. These are extremely challenging for the linear variable selection methods, so the linear methods are unable to compete with the more general methods in any of the scenarios studied. The RGAM_SEL method behaves like the linear methods, because it relies on a linear step, which is unable to select the predictors with no linear correlation with the response.

In terms of computing time, SuRFgam takes significantly longer time than other methods in comparison. This is mainly due to the permutation tests in the forward selection step. However both ranking step and permutation test step can be accelerated by parallel computing.

4.8 Full Tables of Simulation Results

Table 4.3: Simulation results (Gaussian) for data type X_1 (N=500, p=1000)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
1	SuRFgam	0.05	100	100	100	100	100	88	5.88	0.03	0.31	0.06
	SuRFgam	0.10	100	100	100	100	100	88	5.88	0.03	0.45	0.07
	Gamsel	0.4	100	100	100	100	100	6	5.06	0.02	25.59	2.07
	Gamsel	0.5	100	100	100	100	100	37	5.37	0.05	32.32	2.47
	Gamsel	0.6	100	100	100	100	100	54	5.54	0.05	22.62	2.05
	Gamsel	0.7	100	100	100	100	100	51	5.51	0.05	21.98	1.88
	Gamsel	0.8	100	100	100	100	100	54	5.54	0.05	22.07	1.71
	Gamsel	0.9	100	100	100	100	100	84	5.84	0.04	41.22	1.42
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	11.19	0.65
	RGAM_SEL	NA	100	7	100	100	100	0	4.07	0.03	4.46	0.33
	Lasso	N/A	100	5	100	100	100	1	4.06	0.02	18.23	1.01
	STAB	0.6	100	0	99	100	96	0	3.95	0.02	0.71	0.08
	SuRF	0.05	99	0	99	100	97	0	3.95	0.02	0.12	0.04
	SuRF	0.10	99	0	99	100	97	0	3.95	0.02	0.17	0.04
	SuRF	0.20	99	0	99	100	97	0	3.95	0.02	0.26	0.05
1.2	SuRFgam	0.05	100	100	100	100	100	88	5.85	0.04	0.29	0.05
	SuRFgam	0.10	100	100	100	100	100	85	5.85	0.04	0.43	0.07
	Gamsel	0.4	100	100	100	100	100	2	5.02	0.01	19.54	1.48
	Gamsel	0.5	100	100	100	100	100	16	5.16	0.04	20.91	1.75
	Gamsel	0.6	100	100	100	100	100	33	5.33	0.05	14.53	1.51
	Gamsel	0.7	100	100	100	100	100	31	5.31	0.05	13.86	1.27
	Gamsel	0.8	100	100	100	100	100	34	5.34	0.05	14.55	1.31
	Gamsel	0.9	100	100	100	100	100	79	5.79	0.04	49.13	1.66
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	11.50	0.69
	RGAM_SEL	NA	100	6	100	100	100	0	4.06	0.02	4.41	0.32
	Lasso	N/A	100	2	100	100	100	0	4.02	0.01	18.01	1.00
	STAB	0.6	100	0	99	100	93	0	3.92	0.03	0.76	0.08
	SuRF	0.05	99	0	97	100	95	0	3.91	0.03	0.14	0.04
	SuRF	0.10	99	0	97	100	95	0	3.91	0.03	0.20	0.04
	SuRF	0.20	99	0	97	100	95	0	3.91	0.03	0.28	0.06
1.5	SuRFgam	0.05	100	100	100	100	100	79	5.79	0.04	0.31	0.06
	SuRFgam	0.10	100	100	100	100	100	79	5.79	0.04	0.39	0.06
	Gamsel	0.4	100	100	100	100	100	0	5.00	0.00	16.40	1.05
	Gamsel	0.5	100	100	100	100	100	6	5.06	0.02	14.27	1.24
	Gamsel	0.6	100	100	100	100	100	11	5.11	0.03	9.41	0.94
	Gamsel	0.7	100	100	100	100	100	15	5.15	0.04	8.45	0.78
	Gamsel	0.8	100	100	100	100	100	14	5.14	0.03	9.46	0.92
	Gamsel	0.9	100	100	100	100	100	72	5.72	0.05	66.73	1.85
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	11.56	0.72
	RGAM_SEL	NA	100	3	100	100	98	0	4.01	0.02	4.14	0.31
	Lasso	N/A	100	3	100	100	98	0	4.01	0.02	17.94	0.94
	STAB	0.6	100	0	97	100	89	0	3.86	0.03	0.71	0.08
	SuRF	0.05	99	0	97	100	90	0	3.86	0.03	0.16	0.04
	SuRF	0.10	99	0	97	100	90	0	3.86	0.03	0.21	0.05
	SuRF	0.20	99	0	97	100	91	0	3.87	0.03	0.30	0.06
	SuRFgam	0.05	100	100	100	100	100	73	5.73	0.04	0.29	0.05
	SuRFgam	0.10	100	100	100	100	100	73	5.72	0.05	0.41	0.07
	Gamsel	0.4	100	98	100	100	99	0	4.97	0.02	13.79	0.79
	Gamsel	0.5	100	100	99	100	100	2	5.01	0.02	11.07	0.94
	Gamsel	0.6	100	100	99	100	100	7	5.06	0.03	7.64	0.78
	Gamsel	0.7	100	100	99	100	100	6	5.05	0.03	6.92	0.59
	Gamsel	0.8	100	100	99	100	100	8	5.07	0.03	7.99	0.73
	Gamsel	0.9	100	100	100	100	100	63	5.63	0.05	84.77	1.99
	RGAM	NA	100	100	99	100	100	100	5.99	0.01	11.44	0.75

Table 4.3: Simulation results (Gaussian) for data type X_1 (N=500, p=1000)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
1.8			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	RGAM_SEL	NA	100	3	99	100	98	0	4.00	0.02	3.92	0.28
	Lasso	N/A	100	2	100	100	98	1	4.01	0.02	17.92	0.97
	STAB	0.6	100	0	94	100	84	0	3.78	0.04	0.73	0.08
	SuRF	0.05	98	0	93	100	82	0	3.73	0.05	0.21	0.05
	SuRF	0.10	98	0	93	100	83	0	3.74	0.05	0.28	0.05
	SuRF	0.20	98	0	95	100	86	0	3.79	0.04	0.38	0.07
	SuRFgam	0.05	98	100	99	99	98	68	5.62	0.06	0.25	0.04
	SuRFgam	0.10	98	100	99	100	98	68	5.63	0.05	0.39	0.06
	Gamsel	0.4	100	97	99	100	99	0	4.95	0.02	12.49	0.73
	Gamsel	0.5	100	100	99	100	100	1	5.00	0.01	9.64	0.84
	Gamsel	0.6	100	100	99	100	100	3	5.02	0.02	6.89	0.69
	Gamsel	0.7	100	100	99	100	100	3	5.02	0.02	6.23	0.57
	Gamsel	0.8	100	100	99	100	100	6	5.05	0.03	8.63	0.72
2.0	Gamsel	0.9	100	100	99	100	100	58	5.57	0.05	96.58	2.04
	RGAM	NA	100	100	99	100	100	100	5.99	0.01	11.07	0.75
	RGAM_SEL	NA	100	3	98	100	98	0	3.99	0.03	3.49	0.27
	Lasso	N/A	100	2	100	100	98	0	4.00	0.02	17.8	0.98
	STAB	0.6	100	0	90	99	80	0	3.69	0.05	0.69	0.08
	SuRF	0.05	96	0	89	100	80	0	3.65	0.05	0.27	0.05
	SuRF	0.10	96	0	91	100	81	0	3.68	0.05	0.30	0.05
	SuRF	0.20	96	0	91	100	81	0	3.68	0.05	0.39	0.06
	SuRFgam	0.05	94	98	95	98	92	57	5.34	0.07	0.28	0.05
	SuRFgam	0.10	94	98	96	98	92	57	5.35	0.06	0.44	0.07
	Gamsel	0.4	100	75	96	100	92	0	4.63	0.06	9.11	0.69
	Gamsel	0.5	100	98	96	100	97	0	4.91	0.03	7.95	0.70
	Gamsel	0.6	100	100	96	98	100	1	4.95	0.03	6.26	0.60
	Gamsel	0.7	100	100	96	98	100	1	4.95	0.03	5.92	0.61
	Gamsel	0.8	100	100	98	99	100	5	5.02	0.03	14.86	1.11
2.5	Gamsel	0.9	100	100	99	100	100	45	5.44	0.05	124.54	2.30
	RGAM	NA	100	98	96	100	97	99	5.90	0.03	9.30	0.71
	RGAM_SEL	NA	100	2	92	97	93	0	3.84	0.05	3.28	0.29
	Lasso	N/A	100	3	95	100	94	1	3.93	0.04	18.63	1.06
	STAB	0.6	98	0	76	93	62	0	3.29	0.07	0.66	0.08
	SuRF	0.05	93	0	79	92	60	0	3.24	0.07	0.33	0.05
	SuRF	0.10	94	0	83	92	69	0	3.38	0.06	0.40	0.06
	SuRF	0.20	94	0	84	94	74	0	3.46	0.06	0.53	0.07

Table 4.4: Simulation results (Gaussian) for data type X_2 (N=500, p=200)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	SuRFgam	0.05	100	100	100	100	100	99	5.99	0.01	0.27	0.05
	SuRFgam	0.10	100	100	100	100	100	99	5.99	0.01	0.40	0.07
	SuRFgam	0.20	100	100	100	100	100	99	5.99	0.01	0.47	0.07
	Gamsel	0.4	100	100	100	100	100	91	5.91	0.03	80.53	2.70
	Gamsel	0.5	100	100	100	100	100	95	5.95	0.02	62.88	1.99
	Gamsel	0.6	100	100	100	100	100	94	5.94	0.02	47.53	2.14
	Gamsel	0.7	100	100	100	100	100	96	5.96	0.02	49.93	2.04
1	Gamsel	0.8	100	100	100	100	100	96	5.96	0.02	48.07	2.12

Table 4.4: Simulation results (Gaussian) for data type X_2 (N=500, p=200)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	Gamsel	0.9	100	100	100	100	100	95	5.95	0.02	49.50	2.02
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	7.22	0.56
	RGAM_SEL	NA	100	18	100	100	100	9	4.27	0.05	2.45	0.24
	Lasso	N/A	100	15	100	100	100	8	4.23	0.04	13.32	0.84
	STAB	0.6	100	0	99	100	99	0	3.98	0.01	0.19	0.04
	SuRF	0.05	100	0	99	100	100	0	3.99	0.01	0.07	0.03
	SuRF	0.10	100	0	99	100	100	0	3.99	0.01	0.12	0.03
	SuRF	0.20	100	0	99	100	100	0	3.99	0.01	0.22	0.04
	SuRFgam	0.05	100	100	100	100	100	99	5.99	0.01	0.27	0.05
	SuRFgam	0.10	100	100	100	100	100	99	5.99	0.01	0.42	0.07
	SuRFgam	0.20	100	100	100	100	100	99	5.99	0.01	0.57	0.08
	Gamsel	0.4	100	100	100	100	100	65	5.65	0.05	57.19	3.17
	Gamsel	0.5	100	100	100	100	100	85	5.85	0.04	45.69	2.25
	Gamsel	0.6	100	100	100	100	100	85	5.85	0.04	34.20	2.10
	Gamsel	0.7	100	100	100	100	100	85	5.85	0.04	34.49	2.24
1.2	Gamsel	0.8	100	100	100	100	100	84	5.84	0.04	34.31	2.19
	Gamsel	0.9	100	100	100	100	100	90	5.90	0.03	36.58	1.96
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	6.82	0.48
	RGAM_SEL	NA	100	18	100	100	100	9	4.27	0.05	2.39	0.24
	Lasso	N/A	100	14	100	100	100	10	4.24	0.04	13.15	0.81
	STAB	0.6	100	0	99	100	99	0	3.98	0.01	0.18	0.04
	SuRF	0.05	100	0	99	100	100	0	3.99	0.01	0.07	0.03
	SuRF	0.10	100	0	99	100	100	0	3.99	0.01	0.14	0.04
	SuRF	0.20	100	0	99	100	100	0	3.99	0.01	0.23	0.05
	SuRFgam	0.05	100	100	100	100	100	96	5.96	0.02	0.22	0.05
	SuRFgam	0.10	100	100	100	100	100	96	5.96	0.02	0.40	0.07
	SuRFgam	0.20	100	100	100	100	100	96	5.96	0.02	0.56	0.08
	Gamsel	0.4	100	100	100	100	100	31	5.31	0.05	27.89	2.44
	Gamsel	0.5	100	100	100	100	100	50	5.50	0.05	23.61	1.85
	Gamsel	0.6	100	100	100	100	100	54	5.54	0.05	14.72	1.51
	Gamsel	0.7	100	100	100	100	100	52	5.52	0.05	14.47	1.58
1.5	Gamsel	0.8	100	100	100	100	100	52	5.52	0.05	14.55	1.54
	Gamsel	0.9	100	100	100	100	100	86	5.86	0.03	29.25	1.15
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	7.03	0.49
	RGAM_SEL	NA	100	13	100	100	100	9	4.22	0.04	2.32	0.27
	Lasso	N/A	100	12	100	100	100	9	4.21	0.04	13.16	0.78
	STAB	0.6	100	0	99	100	95	0	3.94	0.02	0.17	0.04
	SuRF	0.05	100	0	99	100	98	0	3.97	0.02	0.08	0.03
	SuRF	0.05	100	0	99	100	99	0	3.98	0.01	0.14	0.04
	SuRF	0.05	100	0	99	100	100	0	3.99	0.01	0.21	0.05
	SuRFgam	0.05	100	100	100	100	99	92	5.91	0.02	0.21	0.05
	SuRFgam	0.10	100	100	100	100	100	92	5.92	0.03	0.39	0.07
	SuRFgam	0.20	100	100	100	100	100	92	5.92	0.03	0.59	0.09
	Gamsel	0.4	100	100	100	100	100	13	5.13	0.03	18.15	1.69
	Gamsel	0.5	100	100	100	100	100	22	5.22	0.04	13.33	1.37
	Gamsel	0.6	100	100	100	100	100	26	5.26	0.04	8.23	0.99
	Gamsel	0.7	100	100	100	100	100	25	5.25	0.04	7.91	1.05
1.8	Gamsel	0.8	100	100	100	100	100	26	5.26	0.04	8.25	1.00
	Gamsel	0.9	100	100	100	100	100	83	5.83	0.04	36.04	1.08
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	6.63	0.47
	RGAM_SEL	NA	100	13	100	100	100	8	4.21	0.04	2.10	0.23
	Lasso	N/A	100	12	100	100	100	8	4.2	0.04	13.47	0.80
	STAB	0.6	100	0	96	100	92	0	3.88	0.03	0.16	0.04
	SuRF	0.05	100	0	98	100	96	0	3.94	0.02	0.09	0.03

Table 4.4: Simulation results (Gaussian) for data type X_2 (N=500, p=200)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	SuRF	0.10	100	0	98	100	96	0	3.94	0.02	0.13	0.03
	SuRF	0.20	100	0	99	100	98	0	3.97	0.02	0.20	0.05
2.0	SuRFgam	0.05	100	100	100	100	100	93	5.93	0.03	0.23	0.05
	SuRFgam	0.10	100	100	99	100	100	93	5.92	0.03	0.37	0.07
	SuRFgam	0.20	100	100	100	100	100	93	5.93	0.03	0.55	0.09
	Gamsel	0.4	100	99	100	100	100	8	5.07	0.03	13.69	1.33
	Gamsel	0.5	100	100	100	100	100	15	5.15	0.04	10.59	1.15
	Gamsel	0.6	100	100	100	100	100	18	5.18	0.04	6.53	0.84
	Gamsel	0.7	100	100	100	100	100	14	5.14	0.03	5.78	0.83
	Gamsel	0.8	100	100	100	100	100	18	5.18	0.04	6.69	0.82
	Gamsel	0.9	100	100	100	100	100	84	5.84	0.04	42.12	1.19
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	6.62	0.47
	RGAM_SEL	NA	100	11	100	100	99	6	4.16	0.04	2.28	0.25
	Lasso	N/A	100	12	100	100	99	7	4.18	0.04	13.62	0.83
	STAB	0.6	100	0	96	100	89	0	3.85	0.04	0.14	0.03
	SuRF	0.05	99	0	97	99	93	0	3.88	0.03	0.09	0.03
	SuRF	0.05	99	0	98	99	96	0	3.92	0.03	0.16	0.04
	SuRF	0.05	99	0	98	99	96	0	3.92	0.03	0.22	0.05
2.5	SuRFgam	0.05	99	100	99	96	99	86	5.79	0.04	0.17	0.05
	SuRFgam	0.10	99	100	99	96	98	86	5.78	0.04	0.28	0.06
	SuRFgam	0.20	99	100	99	96	98	86	5.78	0.04	0.46	0.06
	Gamsel	0.4	100	94	100	100	99	3	4.96	0.03	8.36	0.74
	Gamsel	0.5	100	100	99	100	99	3	5.01	0.02	6.99	0.77
	Gamsel	0.6	100	100	99	99	100	7	5.05	0.03	4.32	0.56
	Gamsel	0.7	100	100	98	100	100	6	5.04	0.03	3.69	0.45
	Gamsel	0.8	100	100	99	100	100	9	5.08	0.03	5.96	0.60
	Gamsel	0.9	100	100	100	100	100	77	5.77	0.04	57.24	1.33
	RGAM	NA	100	100	99	100	100	100	5.99	0.01	5.91	0.49
	RGAM_SEL	NA	100	9	97	100	98	8	4.12	0.05	2.10	0.24
	Lasso	N/A	100	8	100	100	99	7	4.14	0.04	13.18	0.80
	STAB	0.6	100	0	90	94	76	0	3.6	0.06	0.15	0.04
	SuRF	0.05	99	0	93	98	82	0	3.72	0.05	0.1	0.03
	SuRF	0.05	99	0	93	99	85	0	3.76	0.05	0.15	0.04
	SuRF	0.05	99	0	96	99	88	0	3.82	0.04	0.24	0.05

Table 4.5: Simulation results (Gaussian) for data type X_3 (N=200, p=200)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
1	SuRFgam	0.05	98	99	99	97	99	94	5.92	0.04	0.16	0.04
	SuRFgam	0.10	98	99	99	98	99	96	5.95	0.02	0.25	0.05
	SuRFgam	0.20	98	99	99	97	99	95	5.93	0.04	0.39	0.07
	Gamsel	0.4	100	94	99	99	99	7	4.98	0.04	11.90	0.95
	Gamsel	0.5	100	100	99	99	100	13	5.11	0.04	11.25	1.20
	Gamsel	0.6	100	100	99	99	100	19	5.17	0.04	7.96	0.85
	Gamsel	0.7	100	100	99	99	100	17	5.15	0.04	8.11	0.71
	Gamsel	0.8	100	100	99	99	100	18	5.16	0.04	8.81	0.89
	RGAM	NA	100	100	100	99	100	100	5.99	0.01	8.22	0.53
	RGAM_SEL	NA	100	15	96	99	96	6	4.12	0.06	4.01	0.34

Table 4.5: Simulation results (Gaussian) for data type X_3 (N=200, p=200)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	Gamsel	0.9	100	100	100	100	100	57	5.57	0.05	34.01	0.87
	Lasso	N/A	100	16	98	99	95	7	4.15	0.06	12.73	0.74
	STAB	0.6	98	2	71	95	65	0	3.31	0.08	0.18	0.04
	SuRF	0.05	97	1	83	95	76	1	3.53	0.07	0.13	0.04
	SuRF	0.1	97	1	89	97	78	1	3.63	0.07	0.2	0.05
	SuRF	0.2	97	2	89	98	79	1	3.66	0.07	0.35	0.06
1.2	SuRFgam	0.05	98	98	96	95	99	89	5.81	0.05	0.13	0.04
	SuRFgam	0.10	98	99	97	96	99	89	5.84	0.05	0.22	0.05
	SuRFgam	0.20	98	99	98	97	99	90	5.87	0.04	0.42	0.07
	Gamsel	0.4	100	90	99	99	98	5	4.91	0.05	10.97	0.96
	Gamsel	0.5	100	100	99	99	99	11	5.08	0.04	9.82	1.12
	Gamsel	0.6	100	100	98	99	100	15	5.12	0.04	7.16	0.78
	Gamsel	0.7	100	100	99	99	100	15	5.13	0.04	7.18	0.69
	Gamsel	0.8	100	100	99	99	100	15	5.13	0.04	8.52	0.91
	Gamsel	0.9	100	100	100	99	100	54	5.53	0.05	38.82	0.87
	RGAM	NA	100	100	99	99	100	100	5.98	0.01	8.05	0.56
	RGAM_SEL	NA	100	15	95	98	94	6	4.08	0.06	3.81	0.32
	Lasso	N/A	100	15	98	99	93	6	4.11	0.05	12.54	0.78
	STAB	0.6	96	2	66	90	59	0	3.13	0.08	0.18	0.05
	SuRF	0.05	98	1	76	94	70	1	3.40	0.08	0.15	0.04
	SuRF	0.10	98	1	81	94	76	1	3.51	0.08	0.18	0.04
	SuRF	0.20	98	1	85	98	76	1	3.59	0.07	0.32	0.06
1.5	SuRFgam	0.05	96	95	82	90	94	74	5.31	0.10	0.17	0.05
	SuRFgam	0.10	97	97	84	94	95	75	5.42	0.09	0.3	0.06
	SuRFgam	0.20	98	98	89	96	97	76	5.54	0.08	0.51	0.08
	SuRFgam	0.30	98	99	89	95	98	76	5.55	0.08	0.81	0.10
	SuRFgam	0.40	99	99	90	97	99	76	5.60	0.07	1.11	0.11
	SuRFgam	0.50	99	99	91	97	99	77	5.62	0.07	1.38	0.12
	SuRFgam	0.60	99	100	93	97	99	73	5.61	0.06	1.91	0.16
	SuRFgam	0.80	99	100	98	98	99	68	5.62	0.06	3.28	0.21
	Gamsel	0.4	100	79	96	98	93	5	4.71	0.06	8.82	0.74
	Gamsel	0.5	100	98	96	98	97	3	4.92	0.04	7.46	0.84
	Gamsel	0.6	100	99	94	98	99	11	5.01	0.05	6.27	0.74
	Gamsel	0.7	100	99	95	98	99	10	5.01	0.05	6.27	0.61
	Gamsel	0.8	100	100	98	98	100	13	5.09	0.04	9.02	0.68
	Gamsel	0.9	100	100	99	98	100	50	5.47	0.05	45.17	0.87
	RGAM	NA	100	95	96	98	99	100	5.88	0.04	7.69	0.58
	RGAM_SEL	NA	98	13	87	96	88	6	3.88	0.08	3.26	0.28
	Lasso	N/A	100	14	95	98	90	6	4.03	0.06	12.16	0.76
	STAB	0.6	94	2	58	84	47	0	2.85	0.08	0.18	0.04
	SuRF	0.05	93	1	62	78	58	1	2.93	0.10	0.17	0.04
	SuRF	0.10	95	1	70	88	64	1	3.19	0.09	0.26	0.05
	SuRF	0.20	97	2	75	94	70	1	3.39	0.08	0.38	0.06
1.8	SuRFgam	0.05	83	91	69	70	72	51	4.36	0.16	0.13	0.04
	SuRFgam	0.10	87	94	72	72	81	55	4.61	0.14	0.2	0.05
	SuRFgam	0.20	96	94	77	82	91	62	5.02	0.11	0.45	0.07
	SuRFgam	0.30	96	93	80	89	94	63	5.15	0.11	0.69	0.09
	SuRFgam	0.40	97	94	81	90	95	63	5.20	0.10	0.96	0.11
	SuRFgam	0.50	97	96	82	89	94	63	5.21	0.10	1.29	0.12
	SuRFgam	0.60	97	96	85	89	94	61	5.22	0.10	1.92	0.16
	SuRFgam	0.80	99	97	91	93	97	54	5.31	0.08	3.15	0.22
	Gamsel	0.4	97	67	86	96	86	4	4.36	0.10	7.39	0.71
	Gamsel	0.5	98	94	89	96	91	2	4.70	0.06	6.93	0.84
	Gamsel	0.6	99	97	86	95	96	8	4.81	0.06	5.48	0.67

Table 4.5: Simulation results (Gaussian) for data type X_3 (N=200, p=200)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	Gamsel	0.7	99	98	86	96	95	8	4.82	0.06	5.81	0.61
	Gamsel	0.8	100	99	95	98	99	10	5.01	0.05	11.13	0.62
	RGAM	NA	100	91	93	97	94	96	5.71	0.06	7.03	0.60
	RGAM_SEL	NA	95	10	84	93	80	5	3.67	0.09	2.61	0.25
	Gamsel	0.9	100	100	99	98	100	49	5.46	0.05	51.41	0.93
	Lasso	N/A	99	10	91	98	88	7	3.93	0.07	12.19	0.81
	STAB	0.60	91	1	49	72	39	0	2.52	0.08	0.22	0.05
	SuRF	0.05	92	1	54	67	47	1	2.62	0.10	0.18	0.04
	SuRF	0.10	93	1	59	77	53	1	2.84	0.10	0.27	0.05
	SuRF	0.20	94	1	66	84	60	1	3.06	0.09	0.44	0.07
2.0	SuRFgam	0.05	70	79	57	59	55	41	3.61	0.16	0.14	0.04
	SuRFgam	0.10	82	87	63	61	66	42	4.01	0.15	0.25	0.05
	SuRFgam	0.20	83	92	67	68	75	46	4.31	0.14	0.54	0.08
	SuRFgam	0.30	92	94	75	77	80	50	4.68	0.12	0.66	0.08
	SuRFgam	0.40	92	94	77	81	85	52	4.81	0.12	0.94	0.11
	SuRFgam	0.50	94	94	79	84	89	50	4.90	0.11	1.17	0.12
	SuRFgam	0.60	93	95	79	85	87	48	4.87	0.11	1.88	0.16
	SuRFgam	0.80	93	98	82	88	91	45	4.97	0.10	3.09	0.22
	Gamsel	0.4	96	42	75	91	79	2	3.85	0.12	5.97	0.67
	Gamsel	0.5	97	91	82	96	87	1	4.54	0.09	5.98	0.70
	Gamsel	0.6	97	95	81	92	93	6	4.64	0.08	5.12	0.66
	Gamsel	0.7	98	98	82	92	95	5	4.70	0.06	5.17	0.55
	Gamsel	0.8	99	99	92	96	99	9	4.94	0.05	13.12	0.67
	Gamsel	0.9	100	100	99	98	100	51	5.48	0.05	54.90	0.94
	RGAM	NA	100	90	85	97	86	86	5.44	0.09	6.43	0.71
	RGAM_SEL	NA	94	8	73	86	67	4	3.32	0.11	2.14	0.23
	Lasso	N/A	99	10	88	98	83	5	3.83	0.07	11.85	0.82
	STAB	0.6	87	1	45	64	34	1	2.32	0.08	0.20	0.04
	SuRF	0.05	87	0	50	58	40	1	2.36	0.10	0.19	0.04
	SuRF	0.10	89	0	53	67	47	1	2.57	0.11	0.30	0.06
	SuRF	0.20	91	1	57	77	55	1	2.82	0.09	0.45	0.07
2.5	SuRFgam	0.05	45	48	36	24	24	18	1.95	0.14	0.04	0.02
	SuRFgam	0.10	51	61	42	35	31	19	2.39	0.15	0.17	0.04
	SuRFgam	0.20	63	74	47	51	42	21	2.98	0.13	0.32	0.05
	SuRFgam	0.30	64	77	50	53	48	24	3.16	0.14	0.64	0.08
	SuRFgam	0.40	70	80	54	56	51	25	3.36	0.14	0.91	0.10
	SuRFgam	0.50	74	83	60	61	63	23	3.64	0.14	1.25	0.14
	SuRFgam	0.60	79	83	62	67	64	26	3.81	0.13	1.75	0.15
	SuRFgam	0.80	85	88	67	72	72	27	4.11	0.12	3.01	0.21
	Gamsel	0.4	82	21	52	64	45	1	2.65	0.15	3.41	0.54
	Gamsel	0.5	92	69	58	77	65	1	3.62	0.13	4.06	0.51
	Gamsel	0.6	85	87	61	67	71	2	3.73	0.12	3.80	0.59
	Gamsel	0.7	94	93	67	81	77	3	4.15	0.09	4.66	0.50
	Gamsel	0.8	98	99	86	95	95	11	4.84	0.06	19.39	0.86
	Gamsel	0.9	98	100	96	96	98	46	5.34	0.06	62.79	0.92
	RGAM	NA	89	63	59	71	60	57	3.99	0.19	4.51	0.71
	RGAM_SEL	NA	75	8	47	57	42	4	2.33	0.14	1.17	0.17
	Lasso	N/A	97	8	76	92	76	4	3.53	0.08	11.15	0.87
	STAB	0.6	75	0	38	44	25	1	1.83	0.09	0.20	0.04
	SuRF	0.05	73	0	35	45	23	1	1.77	0.10	0.17	0.04
	SuRF	0.10	75	0	39	46	27	1	1.88	0.10	0.32	0.05
	SuRF	0.20	85	0	46	59	36	1	2.27	0.10	0.50	0.07

Table 4.6: Simulation results (Gaussian) for data type X_4 (N=100, p=200)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
1	SuRFgam	0.05	41	42	25	24	26	17	1.75	0.17	0.09	0.03
	SuRFgam	0.10	52	53	33	32	37	25	2.32	0.19	0.23	0.05
	SuRFgam	0.20	65	68	47	48	59	34	3.21	0.20	0.42	0.07
	SuRFgam	0.30	70	74	55	55	66	37	3.57	0.19	0.70	0.09
	SuRFgam	0.40	76	79	60	63	69	39	3.86	0.18	0.84	0.11
	SuRFgam	0.50	82	87	66	68	77	41	4.21	0.17	1.18	0.13
	SuRFgam	0.60	86	88	71	76	76	41	4.38	0.15	1.97	0.18
	SuRFgam	0.80	91	92	70	81	86	47	4.67	0.13	3.77	0.27
	Gamsel	0.4	89	38	70	75	64	7	3.43	0.16	7.00	0.79
	Gamsel	0.5	91	76	78	78	79	8	4.10	0.14	7.60	0.87
	Gamsel	0.6	91	90	79	86	86	11	4.43	0.10	7.42	0.69
	Gamsel	0.7	94	93	78	81	89	12	4.47	0.10	7.10	0.61
	Gamsel	0.8	99	96	85	94	96	14	4.84	0.06	12.16	0.58
	Gamsel	0.9	98	98	92	99	98	36	5.21	0.06	33.80	0.66
	RGAM	NA	92	66	67	80	77	73	4.55	0.16	5.51	0.51
	RGAM_SEL	NA	79	11	58	63	52	8	2.71	0.16	2.53	0.42
	Lasso	N/A	92	8	76	81	79	9	3.45	0.11	11.51	0.80
	STAB	0.6	73	0	32	40	17	0	1.62	0.09	0.20	0.04
	SuRF	0.05	64	0	27	38	19	0	1.48	0.10	0.21	0.04
	SuRF	0.10	70	0	36	48	28	0	1.82	0.11	0.31	0.05
	SuRF	0.20	77	1	43	53	38	0	2.12	0.11	0.45	0.07
1.2	SuRFgam	0.05	31	35	19	16	20	11	1.32	0.13	0.13	0.04
	SuRFgam	0.10	43	45	23	24	28	15	1.78	0.16	0.16	0.04
	SuRFgam	0.20	55	56	31	35	38	24	2.39	0.18	0.44	0.07
	SuRFgam	0.30	63	69	43	48	52	28	3.03	0.18	0.56	0.08
	SuRFgam	0.40	63	74	51	52	59	28	3.27	0.17	0.77	0.10
	SuRFgam	0.50	68	79	56	56	62	28	3.49	0.16	0.92	0.11
	SuRFgam	0.60	73	82	59	65	64	30	3.73	0.16	1.57	0.16
	SuRFgam	0.80	83	88	61	75	72	34	4.13	0.15	3.11	0.26
	Gamsel	0.4	85	28	62	70	55	5	3.05	0.17	5.89	0.77
	Gamsel	0.5	89	68	68	76	70	7	3.78	0.15	6.85	0.87
	Gamsel	0.6	88	85	70	72	77	7	3.99	0.13	6.01	0.62
	Gamsel	0.7	91	85	71	77	87	7	4.18	0.10	6.40	0.54
	Gamsel	0.8	99	94	82	91	95	15	4.76	0.07	13.41	0.60
	Gamsel	0.9	98	97	89	97	97	36	5.14	0.07	36.15	0.60
	RGAM	NA	83	54	65	67	61	63	3.93	0.19	4.69	0.56
	RGAM_SEL	NA	73	6	50	54	42	9	2.34	0.16	2.38	0.55
	Lasso	N/A	91	8	73	74	75	8	3.29	0.12	11.29	0.86
	STAB	0.6	68	0	28	38	13	0	1.47	0.09	0.20	0.04
	SuRF	0.05	56	0	24	34	16	0	1.30	0.10	0.18	0.04
	SuRF	0.10	64	0	35	43	22	0	1.64	0.11	0.31	0.05
	SuRF	0.20	69	1	36	49	29	0	1.84	0.11	0.48	0.07
1.5	SuRFgam	0.05	23	23	15	6	14	7	0.88	0.09	0.08	0.03
	SuRFgam	0.10	31	32	18	12	17	8	1.18	0.11	0.17	0.04
	SuRFgam	0.20	40	37	21	17	24	13	1.52	0.14	0.35	0.06
	SuRFgam	0.30	49	49	26	26	30	19	1.99	0.14	0.56	0.08
	SuRFgam	0.40	53	60	29	33	38	21	2.34	0.14	0.68	0.09
	SuRFgam	0.50	56	65	34	41	47	20	2.63	0.15	1.00	0.11
	SuRFgam	0.60	65	71	44	51	50	19	3.00	0.15	1.59	0.14
	SuRFgam	0.80	74	76	56	64	61	22	3.53	0.13	3.11	0.27
	Gamsel	0.4	65	16	42	53	40	4	2.20	0.18	4.12	0.65
	Gamsel	0.5	76	53	52	61	49	4	2.95	0.19	5.79	0.92
	Gamsel	0.6	77	78	53	59	61	5	3.33	0.15	4.93	0.62
	Gamsel	0.7	88	84	62	71	75	5	3.85	0.11	5.99	0.50

Table 4.6: Simulation results (Gaussian) for data type X_4 (N=100, p=200)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	Gamsel	0.8	94	92	78	84	92	20	4.60	0.08	15.63	0.66
	Gamsel	0.9	97	97	85	91	96	34	5.00	0.07	39.09	0.59
	RGAM	NA	70	37	46	50	35	40	2.78	0.21	3.49	0.50
	RGAM_SEL	NA	56	4	33	39	33	6	1.71	0.17	1.62	0.28
	Lasso	N/A	88	7	63	72	66	5	3.01	0.13	10.70	0.92
	STAB	0.6	61	0	24	30	11	0	1.26	0.08	0.19	0.04
	SuRF	0.05	47	0	20	23	10	0	1.00	0.09	0.16	0.04
	SuRF	0.10	54	0	26	33	14	0	1.27	0.10	0.26	0.05
	SuRF	0.20	63	1	34	40	20	0	1.58	0.11	0.45	0.07
	SuRFgam	0.05	16	16	13	4	7	2	0.58	0.07	0.07	0.03
	SuRFgam	0.10	26	18	15	6	15	5	0.85	0.09	0.18	0.04
	SuRFgam	0.20	29	28	19	13	18	6	1.13	0.11	0.30	0.05
	SuRFgam	0.30	34	36	20	18	20	9	1.37	0.11	0.49	0.07
	SuRFgam	0.40	40	45	23	23	28	13	1.72	0.12	0.69	0.09
	SuRFgam	0.50	46	50	25	26	32	11	1.90	0.12	0.88	0.10
	SuRFgam	0.60	52	60	24	31	32	10	2.09	0.12	1.53	0.15
	SuRFgam	0.80	62	66	39	48	43	15	2.73	0.14	3.05	0.23
	Gamsel	0.4	52	11	29	40	28	4	1.64	0.17	3.14	0.60
1.8	Gamsel	0.5	57	34	36	47	33	3	2.10	0.20	4.60	0.89
	Gamsel	0.6	64	60	37	51	46	3	2.61	0.15	3.73	0.56
	Gamsel	0.7	80	78	53	64	68	4	3.47	0.11	6.11	0.50
	Gamsel	0.8	91	91	72	82	86	19	4.41	0.10	17.91	0.67
	Gamsel	0.9	93	94	76	87	92	32	4.74	0.08	41.92	0.55
	RGAM	NA	52	20	32	37	24	23	1.88	0.20	2.48	0.41
	RGAM_SEL	NA	42	2	24	31	22	3	1.24	0.15	1.37	0.29
	Lasso	N/A	82	6	58	64	52	4	2.66	0.13	9.08	0.92
	STAB	0.6	48	0	18	27	9	0	1.02	0.08	0.16	0.04
	SuRF	0.05	42	0	14	19	9	0	0.84	0.08	0.16	0.04
	SuRF	0.10	45	0	18	25	10	0	0.98	0.09	0.24	0.05
	SuRF	0.20	53	1	27	34	13	0	1.28	0.10	0.45	0.07
	SuRFgam	0.05	13	15	11	4	4	2	0.49	0.07	0.06	0.02
	SuRFgam	0.10	19	17	13	6	7	3	0.65	0.08	0.10	0.03
	SuRFgam	0.20	25	19	13	8	15	5	0.85	0.09	0.33	0.06
	SuRFgam	0.30	30	29	18	16	18	5	1.16	0.10	0.47	0.07
	SuRFgam	0.40	33	37	19	21	20	10	1.40	0.11	0.62	0.09
	SuRFgam	0.50	36	40	19	21	25	12	1.53	0.11	0.96	0.11
	SuRFgam	0.60	46	52	24	26	29	11	1.88	0.12	1.27	0.13
	SuRFgam	0.80	55	64	28	40	34	10	2.31	0.13	2.93	0.25
	Gamsel	0.4	40	8	22	32	17	3	1.22	0.15	2.69	0.61
2.0	Gamsel	0.5	45	29	30	38	27	3	1.72	0.18	3.97	0.85
	Gamsel	0.6	58	54	35	41	38	1	2.27	0.14	3.24	0.51
	Gamsel	0.7	80	75	51	60	62	4	3.32	0.11	6.36	0.48
	Gamsel	0.8	91	90	67	75	82	20	4.25	0.10	19.62	0.68
	Gamsel	0.9	93	93	74	86	90	32	4.68	0.09	43.12	0.56
	RGAM	NA	49	17	28	30	14	15	1.53	0.18	2.25	0.43
	RGAM_SEL	NA	38	3	19	24	18	2	1.04	0.14	1.08	0.26
	Lasso	N/A	76	4	54	61	47	5	2.47	0.13	8.86	0.98
	STAB	0.6	42	0	11	20	7	0	0.80	0.07	0.16	0.04
	SuRF	0.05	38	0	10	14	6	0	0.68	0.08	0.14	0.04
	SuRF	0.10	43	0	15	23	10	0	0.91	0.08	0.23	0.05
	SuRF	0.20	47	0	21	28	12	0	1.08	0.10	0.36	0.06
	SuRFgam	0.05	5	9	8	2	1	1	0.26	0.05	0.07	0.03
	SuRFgam	0.10	8	11	10	4	2	2	0.37	0.06	0.14	0.04
	SuRFgam	0.20	15	16	12	8	6	2	0.59	0.08	0.28	0.05

Table 4.6: Simulation results (Gaussian) for data type X_4 (N=100, p=200)

SIGMA	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	SuRFgam	0.30	17	17	13	8	7	3	0.65	0.08	0.39	0.06
	SuRFgam	0.40	22	21	15	10	13	5	0.86	0.09	0.59	0.08
	SuRFgam	0.50	29	27	17	12	14	5	1.04	0.10	0.88	0.10
	SuRFgam	0.60	33	32	18	15	17	6	1.21	0.10	1.14	0.14
	SuRFgam	0.80	40	42	19	21	23	9	1.54	0.10	2.48	0.24
	Gamsel	0.4	31	5	12	13	9	2	0.72	0.13	1.73	0.51
	Gamsel	0.5	31	12	14	24	14	0	0.95	0.14	2.10	0.50
	Gamsel	0.6	47	37	26	27	23	1	1.61	0.12	2.65	0.47
	Gamsel	0.7	75	67	45	54	47	3	2.91	0.11	7.91	0.52
	Gamsel	0.8	85	82	57	72	72	20	3.88	0.10	23.37	0.65
	Gamsel	0.9	90	86	67	79	82	32	4.36	0.10	45.96	0.58
	RGAM	NA	32	7	18	13	9	9	0.88	0.15	1.90	0.57
	RGAM_SEL	NA	23	3	14	11	7	1	0.59	0.12	0.92	0.51
	Lasso	N/A	65	4	39	54	29	4	1.95	0.13	8.08	1.01
	STAB	0.6	34	0	6	15	4	0	0.59	0.06	0.15	0.04
	SuRF	0.05	25	0	5	8	3	0	0.41	0.06	0.08	0.03
	SuRF	0.10	32	0	5	14	6	0	0.57	0.07	0.17	0.04
	SuRF	0.20	38	0	10	17	8	0	0.73	0.07	0.38	0.06

Table 4.7: Simulation results (Binomial) for data type X_1 (N=500, p=1000)

SNR	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	SuRFgam	0.05	91	88	57	83	68	52	4.39	0.12	0.28	0.05
	SuRFgam	0.10	94	91	65	88	69	52	4.59	0.12	0.36	0.06
	SuRFgam	0.20	94	94	69	86	78	52	4.73	0.11	0.69	0.08
	Gamsel	0.4	99	62	86	98	77	5	4.27	0.09	16.53	1.10
	Gamsel	0.5	100	96	85	98	89	3	4.71	0.06	15.34	0.84
	Gamsel	0.6	100	100	90	98	95	6	4.89	0.04	14.10	0.91
	Gamsel	0.7	100	100	89	98	95	4	4.86	0.05	14.08	0.87
0.7	Gamsel	0.8	100	100	93	99	97	14	5.03	0.05	27.16	1.32
	Gamsel	0.9	100	100	96	99	100	50	5.45	0.05	136.93	2.09
	RGAM	NA	100	96	89	99	92	100	5.76	0.05	16.90	1.35
	RGAM_SEL	NA	99	3	83	98	74	4	3.61	0.07	7.25	0.46
	Lasso	N/A	100	2	87	97	79	4	3.69	0.06	18.00	1.10
	STAB	0.6	96	0	53	81	31	0	2.61	0.08	0.55	0.07
	SuRF	0.05	93	0	49	80	33	0	2.55	0.08	0.32	0.05
	SuRF	0.10	94	0	55	84	38	0	2.71	0.08	0.48	0.070
	SuRF	0.20	94	0	61	84	42	0	2.81	0.08	0.6	0.08
	SuRFgam	0.05	98	97	82	97	88	53	5.15	0.09	0.23	0.05
	SuRFgam	0.10	98	96	85	97	90	53	5.19	0.10	0.24	0.05
	SuRFgam	0.20	98	97	85	96	90	53	5.19	0.09	0.42	0.06
	Gamsel	0.4	100	83	92	99	90	3	4.67	0.07	19.64	1.16
	Gamsel	0.5	100	100	92	100	95	1	4.88	0.04	17.34	1.03
	Gamsel	0.6	100	100	95	100	98	10	5.03	0.04	16.73	0.97
	Gamsel	0.7	100	100	95	100	99	8	5.02	0.03	16.95	1.04
1	Gamsel	0.8	100	100	96	100	100	12	5.08	0.03	21.36	1.05
	Gamsel	0.9	100	100	97	100	100	54	5.51	0.05	117.54	1.95
	RGAM	NA	100	98	98	100	98	100	5.94	0.03	20.64	1.74
	RGAM_SEL	NA	100	2	89	99	87	3	3.80	0.05	7.96	0.50

Table 4.7: Simulation results (Binomial) for data type X_1 (N=500, p=1000)

SNR	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
3	Lasso	N/A	100	2	91	99	84	3	3.79	0.05	19.68	1.18
	STAB	0.6	98	0	61	92	45	0	2.96	0.07	0.59	0.07
	SuRF	0.05	95	0	68	90	50	0	3.03	0.08	0.32	0.05
	SuRF	0.10	95	0	68	90	55	0	3.08	0.08	0.41	0.06
	SuRF	0.20	95	0	72	90	60	0	3.17	0.08	0.53	0.07
	SuRFgam	0.05	100	100	95	99	97	71	5.62	0.06	0.22	0.04
	SuRFgam	0.10	100	100	96	100	97	71	5.64	0.06	0.37	0.06
	SuRFgam	0.20	100	100	95	100	97	71	5.63	0.06	0.57	0.07
	Gamsel	0.4	100	100	99	100	99	3	5.01	0.02	25.49	1.22
	Gamsel	0.5	100	100	99	100	100	7	5.06	0.03	22.29	1.34
	Gamsel	0.6	100	100	99	100	100	29	5.28	0.05	23.28	1.33
	Gamsel	0.7	100	100	99	100	100	29	5.28	0.05	24.29	1.42
	Gamsel	0.8	100	100	100	100	100	29	5.29	0.05	25.52	1.39
	Gamsel	0.9	100	100	99	100	100	70	5.69	0.05	82.23	1.70
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	34.21	1.91
	RGAM_SEL	NA	100	3	97	100	98	2	4.00	0.03	9.67	0.61
	Lasso	N/A	100	3	99	100	98	3	4.03	0.03	22.88	1.26
	STAB	0.6	100	0	79	99	71	0	3.49	0.06	0.71	0.09
	SuRF	0.05	99	0	80	98	72	0	3.49	0.06	0.39	0.06
	SuRF	0.10	99	0	84	98	74	0	3.55	0.06	0.48	0.07
	SuRF	0.20	99	0	84	98	77	0	3.58	0.06	0.61	0.07
5	SuRFgam	0.05	100	100	96	100	100	75	5.71	0.05	0.21	0.05
	SuRFgam	0.10	100	100	98	100	100	75	5.73	0.05	0.33	0.06
	SuRFgam	0.20	100	100	99	100	100	75	5.74	0.05	0.54	0.08
	Gamsel	0.4	100	98	100	100	99	3	5.00	0.03	25.08	1.17
	Gamsel	0.5	100	100	100	100	100	10	5.10	0.03	23.30	1.38
	Gamsel	0.6	100	100	100	100	100	38	5.38	0.05	28.79	1.62
	Gamsel	0.7	100	100	100	100	100	41	5.41	0.05	28.29	1.64
	Gamsel	0.8	100	100	100	100	100	41	5.41	0.05	29.18	1.70
	Gamsel	0.9	100	100	100	100	100	75	5.75	0.04	74.52	1.58
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	39.68	2.06
	RGAM_SEL	NA	100	3	100	100	98	3	4.04	0.03	9.40	0.53
	Lasso	N/A	100	3	98	100	97	3	4.01	0.03	21.72	1.22
	STAB	0.6	100	0	80	99	81	0	3.60	0.06	0.67	0.07
	SuRF	0.05	100	0	85	95	78	0	3.58	0.06	0.30	0.05
	SuRF	0.10	100	0	86	95	81	0	3.62	0.05	0.40	0.06
	SuRF	0.20	100	0	89	95	83	0	3.67	0.05	0.54	0.07

Table 4.8: Simulation results (Binomial) for data type X_2(N=500, p=200)

SNR	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
0.7	SuRFgam	0.05	100	94	76	92	76	76	5.14	0.10	0.19	0.04
	SuRFgam	0.10	100	95	85	92	84	76	5.32	0.09	0.34	0.07
	SuRFgam	0.20	100	96	88	95	88	76	5.43	0.08	0.47	0.09
	Gamsel	0.4	100	89	94	98	96	7	4.84	0.06	13.00	0.88
	Gamsel	0.5	100	100	98	98	95	8	4.99	0.04	10.25	0.84
	Gamsel	0.6	100	100	97	100	98	21	5.16	0.05	9.24	0.55
	Gamsel	0.7	100	100	97	100	98	20	5.15	0.05	9.6	0.74

Table 4.8: Simulation results (Binomial) for data type X_2(N=500, p=200)

SNR	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	Gamsel	0.8	100	100	97	100	99	27	5.23	0.05	11.85	0.71
	Gamsel	0.9	100	100	100	100	100	82	5.82	0.04	68.37	1.37
	RGAM	NA	100	100	98	99	99	100	5.96	0.02	10.55	0.73
	RGAM_SEL	NA	100	9	93	98	92	6	3.98	0.06	4.50	0.34
	Lasso	N/A	100	8	95	98	94	8	4.03	0.06	13.18	0.79
	STAB	0.6	99	0	65	88	42	0	2.94	0.07	0.19	0.04
	SuRF	0.05	99	0	67	87	45	0	2.98	0.08	0.18	0.05
	SuRF	0.10	99	0	73	91	56	0	3.19	0.08	0.29	0.06
	SuRF	0.20	99	0	80	93	58	0	3.30	0.07	0.36	0.06
	SuRFgam	0.05	99	99	89	97	93	83	5.60	0.07	0.17	0.04
	SuRFgam	0.10	99	99	90	97	94	84	5.63	0.06	0.30	0.06
	SuRFgam	0.20	99	99	93	99	97	84	5.71	0.06	0.43	0.06
	Gamsel	0.4	100	97	97	100	95	11	5.00	0.05	14.68	0.81
	Gamsel	0.5	100	100	98	100	98	15	5.11	0.04	12.77	0.94
	Gamsel	0.6	100	100	99	100	99	28	5.26	0.05	11.79	0.91
	Gamsel	0.7	100	100	99	100	100	27	5.26	0.05	11.24	0.86
1	Gamsel	0.8	100	100	99	100	100	30	5.29	0.05	12.27	0.93
	Gamsel	0.9	100	100	100	100	100	87	5.87	0.03	55.41	1.26
	RGAM	NA	100	100	100	100	99	100	5.99	0.01	13.18	0.89
	RGAM_SEL	NA	100	7	95	100	93	8	4.03	0.05	4.87	0.38
	Lasso	N/A	100	7	97	100	93	8	4.05	0.05	13.21	0.80
	STAB	0.6	99	0	72	94	62	0	3.27	0.07	0.20	0.04
	SuRF	0.05	98	0	77	93	68	0	3.36	0.07	0.16	0.04
	SuRF	0.10	98	0	83	95	71	0	3.47	0.07	0.21	0.05
	SuRF	0.20	98	0	88	95	76	1	3.58	0.06	0.32	0.06
	SuRFgam	0.05	100	100	100	100	99	96	5.95	0.03	0.13	0.03
	SuRFgam	0.10	100	100	100	100	99	96	5.95	0.03	0.28	0.06
	SuRFgam	0.20	100	100	100	100	100	96	5.96	0.02	0.51	0.08
	Gamsel	0.4	100	100	100	100	100	15	5.15	0.04	21.21	1.22
	Gamsel	0.5	100	100	100	100	100	46	5.46	0.05	20.59	1.25
	Gamsel	0.6	100	100	100	100	100	69	5.69	0.05	19.44	1.20
	Gamsel	0.7	100	100	100	100	100	65	5.65	0.05	18.80	1.12
3	Gamsel	0.8	100	100	100	100	100	68	5.68	0.05	18.91	1.11
	Gamsel	0.9	100	100	100	100	100	91	5.91	0.03	35.74	0.98
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	21.81	1.23
	RGAM_SEL	NA	100	8	100	100	99	7	4.14	0.04	5.82	0.44
	Lasso	N/A	100	9	99	100	99	7	4.14	0.04	14.66	0.95
	STAB	0.6	100	0	88	100	87	0	3.75	0.05	0.25	0.05
	SuRF	0.05	100	0	90	99	86	0	3.75	0.05	0.22	0.05
	SuRF	0.10	100	0	92	99	88	0	3.79	0.04	0.26	0.05
	SuRF	0.20	100	0	93	99	92	0	3.84	0.04	0.41	0.06
	SuRFgam	0.05	100	100	100	100	99	92	5.91	0.03	0.20	0.04
	SuRFgam	0.10	100	100	100	100	99	92	5.91	0.03	0.27	0.05
	SuRFgam	0.20	100	100	100	100	99	92	5.91	0.03	0.47	0.09
	Gamsel	0.4	100	100	100	100	100	24	5.24	0.04	23.56	1.29
	Gamsel	0.5	100	100	100	100	100	58	5.58	0.05	24.07	1.26
	Gamsel	0.6	100	100	100	100	100	75	5.75	0.04	21.47	1.06
	Gamsel	0.7	100	100	100	100	100	80	5.8	0.04	21.22	1.06
5	Gamsel	0.8	100	100	100	100	100	78	5.78	0.04	21.75	1.06
	Gamsel	0.9	100	100	100	100	100	92	5.92	0.03	32.61	0.98
	RGAM	NA	100	100	100	100	100	100	6.00	0.00	24.37	1.22
	RGAM_SEL	NA	100	10	100	100	100	9	4.19	0.04	6.79	0.47
	Lasso	N/A	100	9	100	100	100	13	4.22	0.04	15.86	0.86
	STAB	0.6	100	0	92	100	88	0	3.80	0.05	0.20	0.04

Table 4.8: Simulation results (Binomial) for data type X_2(N=500, p=200)

SNR	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	SuRF	0.05	100	0	93	99	88	0	3.80	0.04	0.19	0.04
	SuRF	0.10	100	0	94	99	89	0	3.82	0.04	0.24	0.05
	SuRF	0.20	100	0	96	99	91	0	3.86	0.04	0.33	0.05

Table 4.9: Simulation results (Binomial) for data type X_3 (N=200, p=200)

SNR	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
0.7	SuRFgam	0.05	21	16	2	16	11	12	0.78	0.08	0.12	0.04
	SuRFgam	0.10	26	23	7	24	18	14	1.12	0.11	0.20	0.04
	SuRFgam	0.20	37	34	15	35	24	19	1.64	0.13	0.42	0.06
	SuRFgam	0.30	43	42	16	39	28	19	1.87	0.13	0.53	0.08
	SuRFgam	0.40	47	46	20	42	32	20	2.07	0.13	0.83	0.12
	SuRFgam	0.50	53	50	26	50	34	19	2.32	0.13	1.10	0.14
	SuRFgam	0.60	58	53	27	51	35	21	2.45	0.13	1.43	0.16
	SuRFgam	0.8	67	64	38	58	42	17	2.86	0.13	2.94	0.27
	SuRFgam	0.9	74	72	41	61	46	20	3.14	0.12	4.54	0.35
	Gamsel	0.4	86	14	47	72	40	1	2.60	0.12	5.97	0.79
	Gamsel	0.5	87	55	51	77	55	3	3.28	0.13	6.54	0.67
	Gamsel	0.6	84	76	45	72	62	6	3.45	0.13	6.53	0.81
	Gamsel	0.7	91	81	54	82	72	8	3.88	0.11	9.41	0.85
	Gamsel	0.8	94	95	80	93	86	23	4.71	0.08	26.66	1.04
	Gamsel	0.9	96	97	89	100	94	47	5.23	0.07	63.38	0.86
	RGAM	NA	78	40	49	72	39	55	3.33	0.17	5.49	0.62
	RGAM_SEL	NA	73	5	37	65	38	1	2.19	0.13	2.61	0.33
	Lasso	N/A	88	9	62	74	53	4	2.9	0.12	9.19	0.78
	STAB	0.6	62	0	14	40	8	0	1.24	0.09	0.19	0.04
	SuRF	0.05	52	0	10	30	8	0	1	0.09	0.11	0.03
	SuRF	0.10	60	0	19	38	12	0	1.29	0.09	0.21	0.05
	SuRF	0.20	64	0	26	48	12	0	1.5	0.09	0.37	0.07
1	SuRFgam	0.05	33	30	8	35	15	20	1.41	0.11	0.10	0.04
	SuRFgam	0.10	40	38	10	39	22	24	1.73	0.13	0.18	0.05
	SuRFgam	0.20	61	49	18	48	31	27	2.34	0.12	0.40	0.08
	SuRFgam	0.30	64	56	28	57	38	29	2.72	0.14	0.55	0.08
	SuRFgam	0.40	73	63	33	61	42	32	3.04	0.14	0.82	0.10
	SuRFgam	0.50	76	72	38	66	52	36	3.40	0.13	1.12	0.12
	SuRFgam	0.60	79	72	37	70	52	35	3.45	0.13	1.56	0.15
	SuRFgam	0.8	82	78	48	77	59	29	3.73	0.12	2.91	0.26
	SuRFgam	0.9	85	85	55	80	67	31	4.03	0.12	4.86	0.36
	Gamsel	0.4	90	25	56	84	55	2	3.12	0.12	6.77	0.70
	Gamsel	0.5	90	72	66	89	72	5	3.94	0.12	9.02	0.69
	Gamsel	0.6	92	88	63	92	76	11	4.22	0.11	8.39	0.80
	Gamsel	0.7	93	89	66	94	80	9	4.31	0.10	9.47	0.87
	Gamsel	0.8	96	96	85	99	93	22	4.91	0.07	21.82	0.96
	Gamsel	0.9	99	99	92	100	97	54	5.41	0.06	58.29	0.89
	RGAM	NA	90	57	60	86	58	74	4.25	0.16	7.82	0.79
	RGAM_SEL	NA	85	7	53	81	49	4	2.79	0.12	3.72	0.40
	Lasso	N/A	91	6	70	88	61	6	3.22	0.10	10.61	0.82
	STAB	0.6	75	0	23	52	11	0	1.61	0.08	0.14	0.03
	SuRF	0.05	70	0	21	46	11	0	1.48	0.09	0.1	0.03

Table 4.9: Simulation results (Binomial) for data type X_3 (N=200, p=200)

SNR	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	SuRF	0.10	77	0	26	55	14	0	1.72	0.08	0.16	0.04
	SuRF	0.20	81	0	28	61	17	0	1.87	0.09	0.31	0.05
3	SuRFgam	0.05	85	59	28	71	56	57	3.56	0.16	0.10	0.03
	SuRFgam	0.10	91	73	36	79	68	60	4.07	0.14	0.17	0.04
	SuRFgam	0.20	94	84	48	86	77	58	4.47	0.12	0.43	0.07
	SuRFgam	0.30	95	87	54	88	81	60	4.65	0.11	0.61	0.08
	SuRFgam	0.40	96	88	62	90	87	58	4.81	0.11	0.96	0.12
	SuRFgam	0.50	96	91	64	91	87	60	4.89	0.10	1.23	0.14
	SuRFgam	0.60	96	93	64	92	88	62	4.95	0.10	1.51	0.14
	SuRFgam	0.8	97	97	73	97	93	59	5.16	0.08	3.67	0.36
	SuRFgam	0.9	98	95	75	97	92	58	5.15	0.09	5.70	0.43
	Gamsel	0.4	99	53	79	99	81	4	4.15	0.09	10.74	0.89
	Gamsel	0.5	99	89	84	99	91	9	4.71	0.07	11.99	0.97
	Gamsel	0.6	100	98	83	100	97	12	4.90	0.06	11.69	0.79
	Gamsel	0.7	100	97	87	100	97	13	4.94	0.06	12.05	0.84
	Gamsel	0.8	100	99	88	100	100	18	5.05	0.05	14.78	0.75
	Gamsel	0.9	100	100	94	100	100	60	5.54	0.05	46.46	0.84
	RGAM	NA	99	90	87	100	91	97	5.64	0.06	12.53	0.88
	RGAM_SEL	NA	99	9	73	98	80	6	3.65	0.09	5.02	0.45
	Lasso	N/A	99	10	85	99	81	5	3.79	0.06	12.76	0.76
	STAB	0.6	94	1	30	78	26	0	2.29	0.08	0.15	0.04
	SuRF	0.05	92	1	34	78	23	0	2.28	0.09	0.15	0.04
	SuRF	0.10	95	1	42	84	34	0	2.56	0.09	0.24	0.05
	SuRF	0.20	98	1	47	87	43	0	2.76	0.09	0.37	0.06
5	SuRFgam	0.05	85	66	34	83	69	59	3.96	0.16	0.21	0.05
	SuRFgam	0.10	89	79	41	90	79	64	4.42	0.14	0.29	0.06
	SuRFgam	0.20	93	86	52	92	88	69	4.8	0.11	0.53	0.08
	SuRFgam	0.30	95	90	62	95	91	71	5.04	0.09	0.89	0.10
	SuRFgam	0.40	95	92	66	97	94	70	5.14	0.09	1.1	0.10
	SuRFgam	0.50	96	92	70	97	94	70	5.19	0.08	1.42	0.14
	SuRFgam	0.60	96	92	73	99	93	69	5.22	0.08	1.75	0.14
	SuRFgam	0.8	96	96	73	99	96	58	5.18	0.08	3.88	0.29
	SuRFgam	0.9	96	96	77	98	96	57	5.20	0.08	6.15	0.47
	Gamsel	0.4	98	56	85	99	87	6	4.31	0.09	11.26	0.87
	Gamsel	0.5	98	93	90	99	97	13	4.90	0.06	13.59	0.88
	Gamsel	0.6	98	96	85	100	97	20	4.96	0.07	12.45	0.89
	Gamsel	0.7	98	98	82	100	98	20	4.96	0.07	12.08	0.83
	Gamsel	0.8	98	99	89	100	99	29	5.14	0.06	15.58	0.86
	Gamsel	0.9	98	100	98	100	100	54	5.50	0.05	44.49	0.90
	RGAM	NA	99	94	89	100	99	99	5.80	0.05	13.98	0.86
	RGAM_SEL	NA	97	8	79	99	86	6	3.75	0.07	5.60	0.46
	Lasso	N/A	98	7	86	99	85	6	3.81	0.06	14.56	0.92
	STAB	0.6	92	0	25	81	26	0	2.24	0.08	0.19	0.04
	SuRF	0.05	92	0	33	74	33	0	2.32	0.10	0.17	0.04
	SuRF	0.10	93	0	42	85	44	0	2.64	0.09	0.25	0.05
	SuRF	0.20	94	0	52	92	50	0	2.88	0.09	0.40	0.06

Table 4.10: Simulation results (Binomial) for data type X_4 (N=100, p=200)

SNR	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	SuRFgam	0.05	3	2	0	2	3	4	0.14	0.03	0.04	0.02
	SuRFgam	0.10	5	2	0	6	6	4	0.23	0.04	0.1	0.03
	SuRFgam	0.20	6	2	2	7	9	6	0.32	0.05	0.21	0.05
	SuRFgam	0.30	8	6	3	11	13	9	0.50	0.06	0.44	0.09
	SuRFgam	0.40	14	8	5	12	14	9	0.62	0.07	0.55	0.09
	SuRFgam	0.50	17	10	5	14	16	9	0.71	0.07	0.79	0.10
	SuRFgam	0.60	22	12	6	19	17	9	0.85	0.08	1.19	0.14
	SuRFgam	0.80	30	21	9	26	26	11	1.23	0.10	2.97	0.31
	SuRFgam	0.90	35	29	15	30	24	12	1.45	0.10	5.21	0.46
0.7	Gamsel	0.4	42	1	21	29	12	1	1.06	0.11	2.53	0.50
	Gamsel	0.5	46	9	19	33	17	1	1.25	0.13	3.27	0.59
	Gamsel	0.6	49	38	13	36	31	2	1.69	0.13	4.02	0.50
	Gamsel	0.7	65	59	31	55	51	5	2.66	0.11	11.05	0.67
	Gamsel	0.8	75	74	49	71	67	13	3.49	0.11	26.07	0.71
	Gamsel	0.9	80	79	58	74	76	24	3.91	0.11	43.56	0.65
	RGAM	NA	29	4	16	27	12	12	1.00	0.13	1.76	0.32
	RGAM_SEL	NA	33	0	9	28	8	2	0.80	0.10	1.04	0.18
	Lasso	N/A	45	5	25	36	19	3	1.33	0.14	5.39	0.75
	STAB	0.6	27	0	5	12	3	0	0.47	0.06	0.16	0.04
	SuRF	0.05	15	0	3	8	2	0	0.28	0.05	0.05	0.02
	SuRF	0.10	24	0	3	11	3	0	0.41	0.06	0.14	0.03
	SuRF	0.20	31	0	4	14	5	0	0.54	0.07	0.24	0.04
	SuRFgam	0.05	3	2	0	3	6	3	0.17	0.04	0.06	0.03
	SuRFgam	0.10	4	2	0	6	7	6	0.25	0.05	0.11	0.04
	SuRFgam	0.20	6	2	1	9	14	7	0.39	0.06	0.02	0.04
	SuRFgam	0.30	8	9	1	16	18	9	0.61	0.07	0.46	0.08
	SuRFgam	0.40	17	17	4	22	20	8	0.88	0.09	0.72	0.09
	SuRFgam	0.50	24	20	4	24	21	9	1.02	0.09	1.06	0.12
	SuRFgam	0.60	28	22	6	26	21	8	1.11	0.09	1.31	0.13
	SuRFgam	0.80	38	30	10	36	28	10	1.52	0.10	2.97	0.28
	SuRFgam	0.90	46	38	11	43	31	9	1.78	0.11	5.00	0.39
1	Gamsel	0.4	54	1	19	38	13	1	1.26	0.10	2.29	0.44
	Gamsel	0.5	59	17	25	42	21	1	1.65	0.14	3.50	0.55
	Gamsel	0.6	58	34	18	43	34	0	1.87	0.12	3.48	0.45
	Gamsel	0.7	76	61	33	65	51	6	2.92	0.10	9.60	0.55
	Gamsel	0.8	83	77	56	76	70	13	3.75	0.10	24.03	0.66
	Gamsel	0.9	89	83	62	82	77	21	4.14	0.11	41.73	0.54
	RGAM	NA	37	8	19	41	13	25	1.43	0.15	2.74	0.48
	RGAM_SEL	NA	42	1	14	37	12	4	1.10	0.12	1.42	0.24
	Lasso	N/A	63	2	28	49	24	5	1.71	0.14	6.15	0.79
	STAB	0.6	31	0	7	20	5	0	0.63	0.07	0.09	0.03
	SuRF	0.05	21	0	5	14	4	0	0.44	0.06	0.05	0.02
	SuRF	0.10	27	0	7	18	4	0	0.56	0.07	0.12	0.04
	SuRF	0.20	37	0	7	23	6	1	0.74	0.08	0.25	0.05
	SuRFgam	0.05	14	6	1	15	6	11	0.53	0.07	0.08	0.03
	SuRFgam	0.10	21	10	1	21	11	14	0.78	0.08	0.11	0.04
	SuRFgam	0.20	26	15	4	26	16	14	1.01	0.09	0.32	0.06
	SuRFgam	0.30	31	19	9	31	22	13	1.25	0.09	0.47	0.07
	SuRFgam	0.40	37	24	12	38	26	12	1.49	0.10	0.84	0.11
	SuRFgam	0.50	44	30	17	40	23	11	1.65	0.11	1.17	0.12
	SuRFgam	0.60	49	36	19	43	26	11	1.84	0.12	1.53	0.13
	SuRFgam	0.80	57	49	18	55	31	10	2.20	0.12	3.42	0.29
	SuRFgam	0.90	63	53	26	62	38	13	2.55	0.11	5.33	0.38
3	Gamsel	0.4	72	5	31	54	33	4	1.99	0.14	3.94	0.50

Table 4.10: Simulation results (Binomial) for data type X_4 (N=100, p=200)

SNR	Method	Param	Selected variables						True Positive		False Positive	
			V1	V2	V3	V4	V5	V6	mean	SE	mean	SE
	Gamsel	0.5	71	26	32	59	45	3	2.36	0.15	4.32	0.47
	Gamsel	0.6	71	58	27	62	55	5	2.78	0.14	6.21	0.63
	Gamsel	0.7	82	64	40	72	61	7	3.26	0.13	8.25	0.60
	Gamsel	0.8	92	84	55	84	87	16	4.18	0.08	17.92	0.69
	Gamsel	0.9	93	93	64	89	91	30	4.60	0.10	36.30	0.65
	RGAM	NA	75	26	34	60	41	47	2.83	0.17	4.88	0.53
	RGAM_SEL	NA	61	4	22	57	26	3	1.73	0.13	1.97	0.28
	Lasso	N/A	82	8	43	68	51	7	2.59	0.13	9.18	0.90
	STAB	0.6	50	0	8	35	9	0	1.02	0.08	0.2	0.05
	SuRF	0.05	36	0	7	30	4	1	0.78	0.08	0.11	0.03
	SuRF	0.10	47	0	10	34	7	1	0.99	0.09	0.21	0.04
	SuRF	0.20	58	0	18	41	14	1	1.32	0.10	0.33	0.06
5	SuRFgam	0.05	18	5	1	14	6	9	0.53	0.08	0.08	0.03
	SuRFgam	0.10	23	6	4	21	8	14	0.76	0.09	0.20	0.05
	SuRFgam	0.20	31	14	9	26	16	18	1.14	0.11	0.39	0.07
	SuRFgam	0.30	39	23	12	33	23	16	1.46	0.11	0.63	0.10
	SuRFgam	0.40	48	26	15	40	27	16	1.72	0.12	0.85	0.11
	SuRFgam	0.50	52	31	14	46	32	15	1.90	0.13	1.05	0.13
	SuRFgam	0.60	57	34	18	47	39	15	2.10	0.13	1.52	0.16
	SuRFgam	0.80	70	43	26	61	45	18	2.63	0.12	3.55	0.30
	SuRFgam	0.90	75	52	27	58	53	18	2.83	0.11	5.58	0.37
	Gamsel	0.4	78	3	35	57	41	4	2.18	0.14	4.43	0.56
	Gamsel	0.5	83	34	37	65	50	5	2.74	0.15	5.47	0.59
	Gamsel	0.6	84	55	30	66	62	5	3.02	0.14	6.44	0.66
	Gamsel	0.7	90	61	40	77	67	9	3.44	0.13	8.03	0.61
	Gamsel	0.8	94	81	55	89	84	14	4.17	0.10	18.41	0.66
	Gamsel	0.9	96	89	67	94	90	35	4.71	0.08	36.13	0.64
	RGAM	NA	81	27	38	70	51	56	3.23	0.18	5.96	0.63
	RGAM_SEL	NA	79	0	30	57	44	8	2.18	0.13	2.81	0.34
	Lasso	N/A	86	2	47	73	52	11	2.71	0.13	9.63	0.90
	STAB	0.6	57	0	10	32	11	0	1.10	0.08	0.12	0.03
	SuRF	0.05	45	0	7	26	6	0	0.84	0.08	0.07	0.03
	SuRF	0.10	52	0	8	31	11	0	1.02	0.09	0.15	0.04
	SuRF	0.20	61	0	12	40	16	0	1.29	0.10	0.43	0.08

Chapter 5

Conclusion

In this thesis, we developed a novel method, **Su**bsampling **R**anking **F**orward selection (SuRF), for a stable and sparse variable selection for high-dimensional data.

In Chapter 2, we introduced the two-step selection procedure: ranking the predictors by applying Lasso from subsamples and selecting most important predictors via a sequential forward ANOVA test, using a permutation to determine the critical value. SuRF offers major advantages over existing variable selection methods in terms of both sparsity of selected models and model inference. We also introduced an aggregation method for selecting significant OTUs across all levels of the taxonomic tree structure, when analysing microbiome data. Existing methods arbitrarily choose a taxonomic level *a priori* before performing the analysis, whereas by combining SuRF with these aggregated variables, we are able to identify the key biomarkers. The forward selection step makes SuRF distinct from an alternative method, Stability Selection. While Stability selection also relies on Lasso to select variables from subsamples, the ultimate selection is solely based on whether the selection frequency of the variables surpasses a arbitrarily predetermined cutoff value.

Through extensive simulation studies, we have clearly established SuRF's applicability in a broad spectrum of variable selection and prediction tasks. It can be particularly useful in identifying significant features for predicting various types of outcomes within the context of generalised linear model (GLM) settings, including Gaussian, Binomial and Poisson regression models. In a collaborative research paper ([3]) not included in this thesis, the author has developed an extension, SuRFCox, which applies to the Cox-proportional hazards model for survival analysis.

In Chapter 3, we conducted a comprehensive simulation study to study the effect of the marginal distribution of predictors on various Lasso-based variable selection methods. The variable scaling in Lasso may yield desired results in the Gaussian model, but may not be as effective in the context of Binomial and Poisson models. The effect of marginal distribution of predictors is relatively small for Gaussian regression, but is much larger for

logistic regression and Poisson regression with log link. Heavy-tailed predictor are selected less frequently in the Binomial GLM with logistic link, and more frequently in the Poisson GLM with log link. This effect was most noticeable for the log-normal predictors, which were almost never selected by Stability selection, even at the highest MI level considered. SuRF is less affected by the marginal distribution of predictors than Lasso and Stability. This means that SuRF performs by far the best when at least one true predictor follows a log-normal or Pareto distribution for logistic regression. For Poisson regression with log link, SuRF performs similarly to Stability, though it is more able to select less conservative models. We discovered that applying a Box-Cox transformation to the predictors in the Binomial model can improve variable selection, even when this results in the linear model becoming misspecified. However, the performance is still worse than for the less heavy-tailed predictors. The Box-Cox transformation doesn't show the same benefits for the Poisson regression model.

In Chapter 4, we extended SuRF to perform variable selection for generalised additive models (GAMs), a type of nonparametric additive model. GAMs model the conditional expectation of the response through a link function as a sum of smooth functions of each predictor, in order to capture non-linear effects of the predictors. Replacing GLMs with GAMs is necessary in both the ranking and the forward-selection steps of SuRF. In the forward-selection step, this can be routinely done by replacing the GLM by a GAM. For the ranking step, we use *Gamsel* [11] (Generalised Additive Model Selection), which is a variable selection method for GAMs, based on group Lasso.

We conducted a comprehensive simulation study to compare SuRFgam with various state-of-the-art methods for variable selection in GAMs, including linear methods, such as Lasso, SuRF, and Stability Selection, and non-linear methods Gamsel, RGAM, RGAM_SEL and SPAM. We compared performance on Gaussian and Binomial regression model settings across a range of data dimensions and signal strengths. SuRFgam demonstrates a superior performance in both nonlinear variable selection and the prediction accuracy. It is particularly effective in reducing the noise variables, making it a better choice in various modelling scenarios.

Finally, we provided an `R` package that can implement our method for generalised linear models, generalised additive models and survival models.

There are many possible directions for future work to improve the SuRF method:

- It should be reasonably straightforward to extend SuRF to multilevel classification through logistic regression.

- SuRFgam performs a variable selection without making a distinction between linear and nonlinear predictors during the model fitting. However, many variable selection methods for GAMs consider it important to distinguish between linear and nonlinear predictors in these models. The Gamsel method used to rank predictors in SuRFgam does distinguish between linear and nonlinear predictors, so incorporating this information into the ranking is relatively straightforward. The forward selection step is more challenging, because the critical value for adding a linear predictor and for adding a nonlinear predictor will be different. We can resolve this by calculating separate null distributions for linear and nonlinear predictors. This could be extended to allow certain predictors to be limited to only linear relations with the response variable. This would be very useful in cases where nonlinear functions cannot feasibly be fit for some predictors, for example if a predictor is categorical, or count data with only a small number of values, then a spline cannot be fitted on that variable, so it can only be considered as linear. This could apply in microbiome data, where many rare OTUs have mainly zero counts. If these predictors could be restricted to linear effects, then SuRFgam could be applied to these datasets to fit linear models on the sparse OTUs and nonlinear models on the more abundant OTUs.

- Further research is needed to reduce the influence of the marginal distribution of predictors on variable selection methods. One approach would be to develop a new standardisation procedure that is more appropriate for heavy-tailed predictors in GLMs.

- The main disadvantage of SuRF and SuRFgam is the heavy computational cost. For SuRFgam, the ranking step by Gamsel alone takes significantly longer than some other variable selection methods. While the computation time is not excessive, and it is usually worth the extra time to get better variable selection, more work could be done on methods to accelerate the variable selection. A straightforward implementation change to improve the speed is to implement parallel computing for ranking predictors and performing the permutations to estimate the null distribution.

Appendix A

Complete figures for Chapter 3

A.1 Appendix1: Complete Figures for Variable selection and prediction

A.1.1 Gaussian single true variable cases

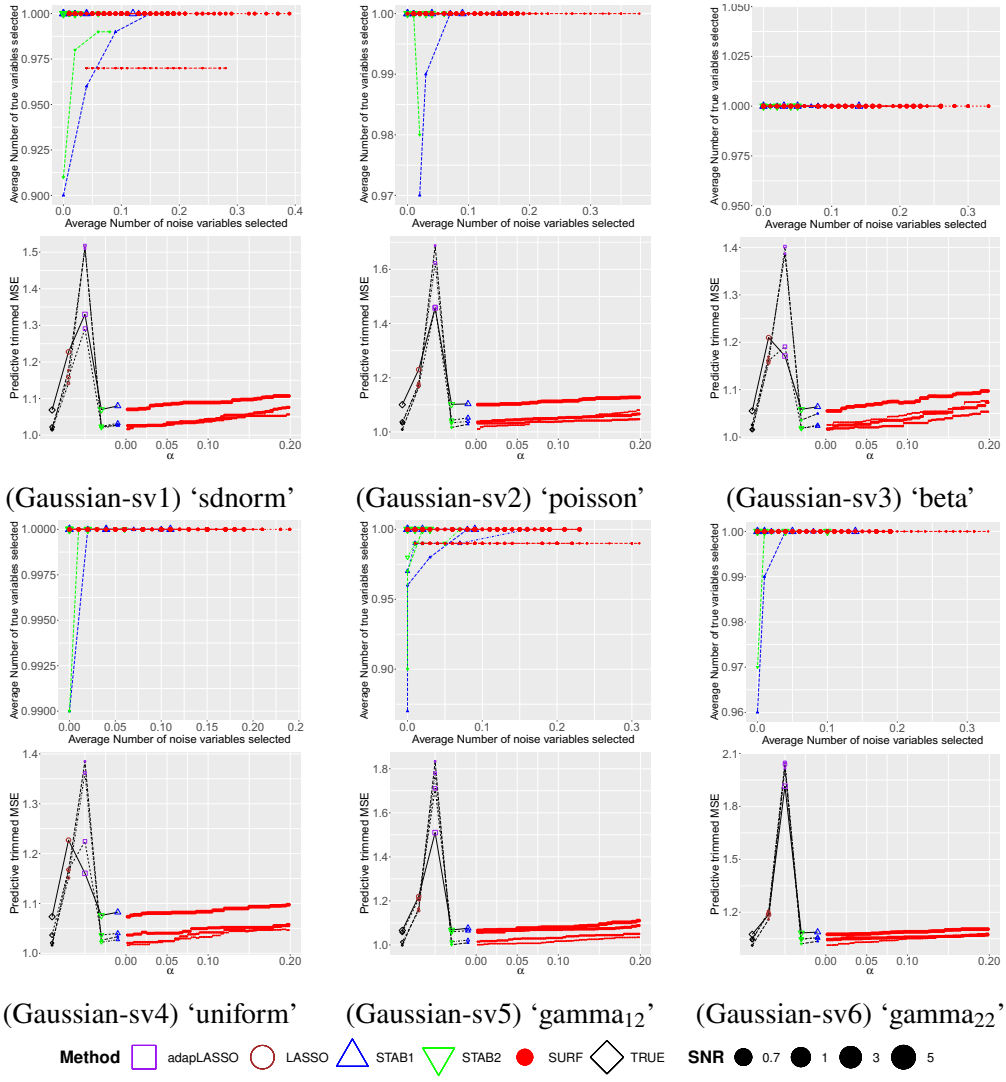


Figure A.1: Variable selection and prediction for a single true variable Gaussian regression

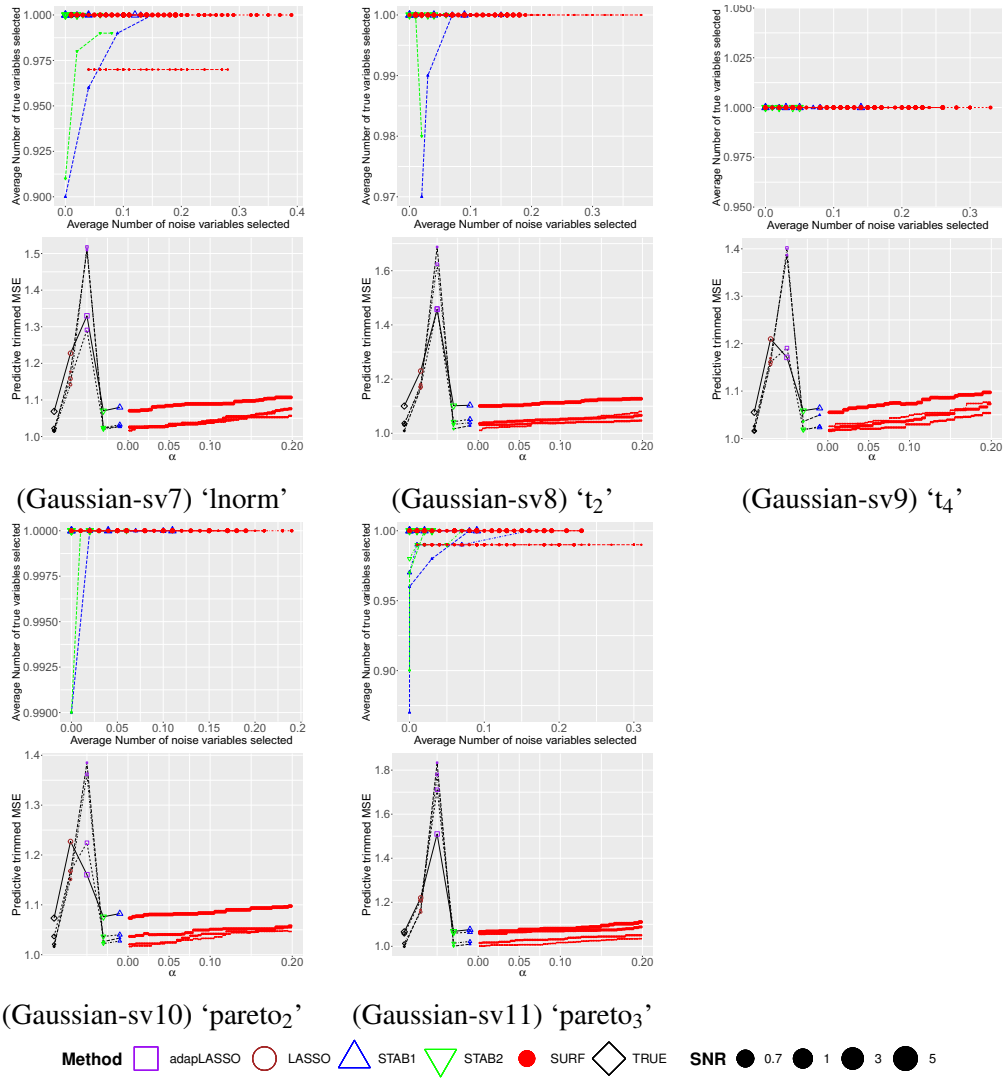


Figure A.1: Variable selection and prediction for a single true variable Gaussian regression

A.1.2 Gaussian multiple true variable cases

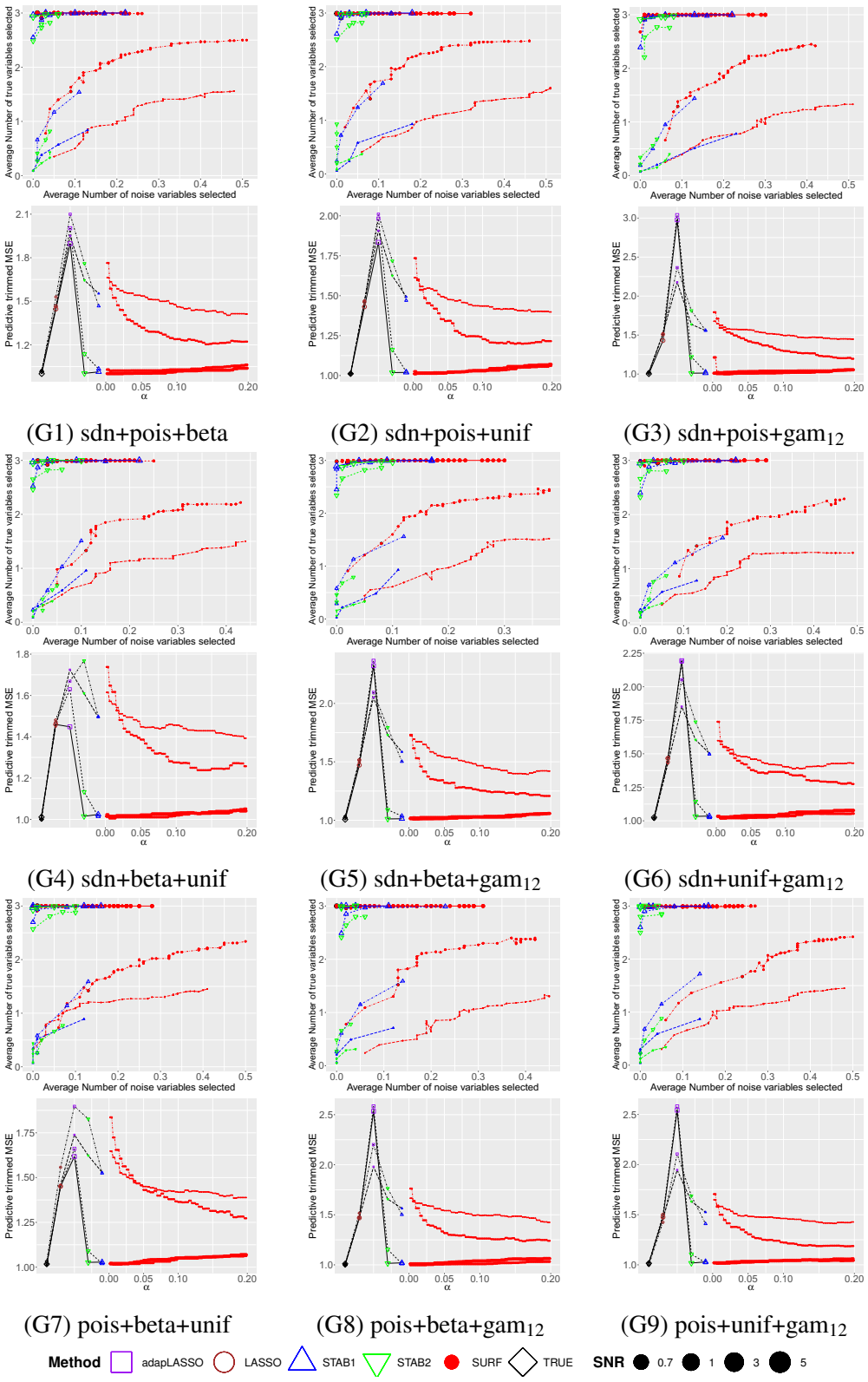


Figure A.2: Variable selection and prediction for multiple true variable Gaussian regression

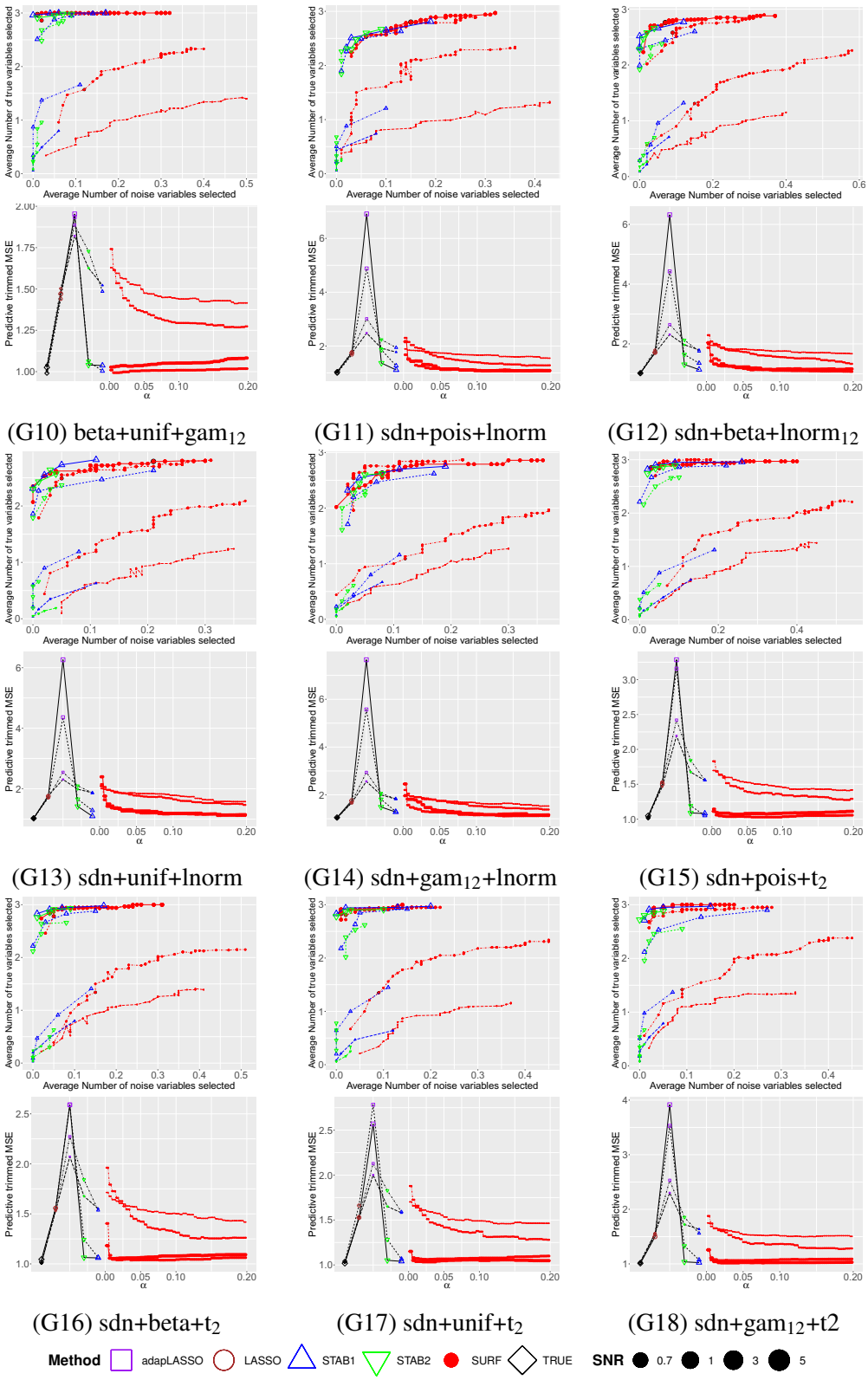


Figure A.2: Variable selection and prediction for multiple true variable Gaussian regression

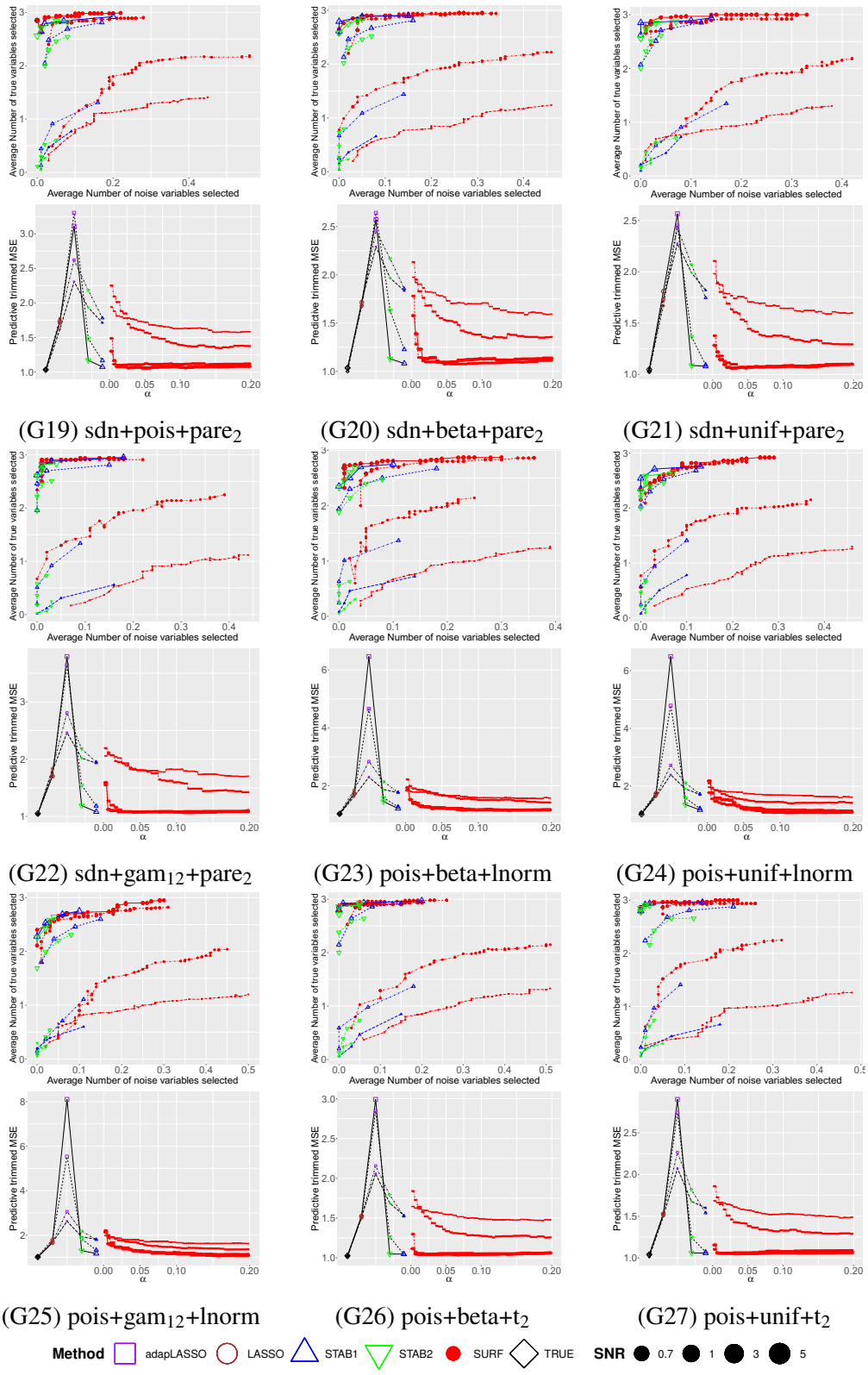


Figure A.2: Variable selection and prediction for multiple true variable Gaussian regression

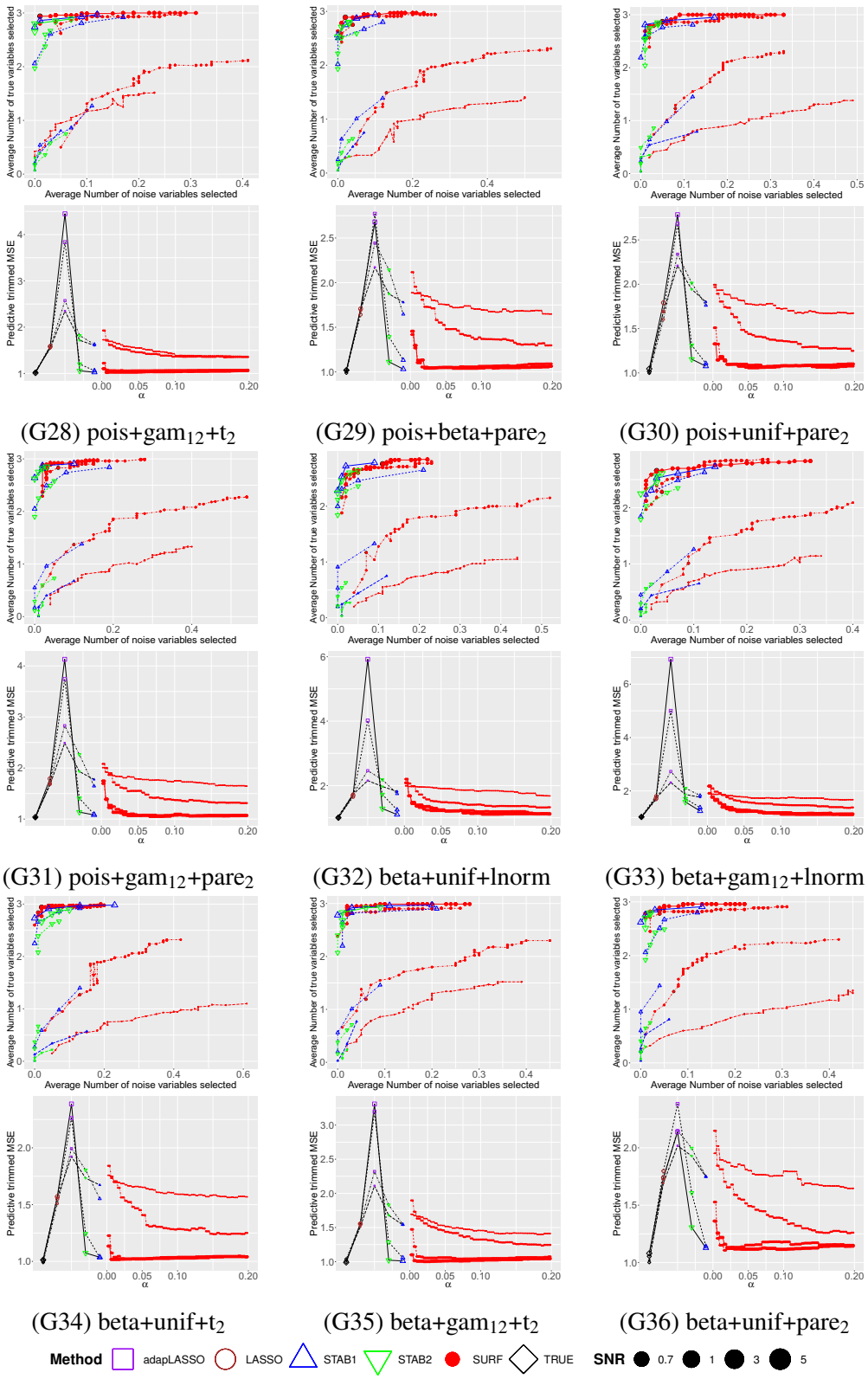


Figure A.2: Variable selection and prediction for multiple true variable Gaussian regression

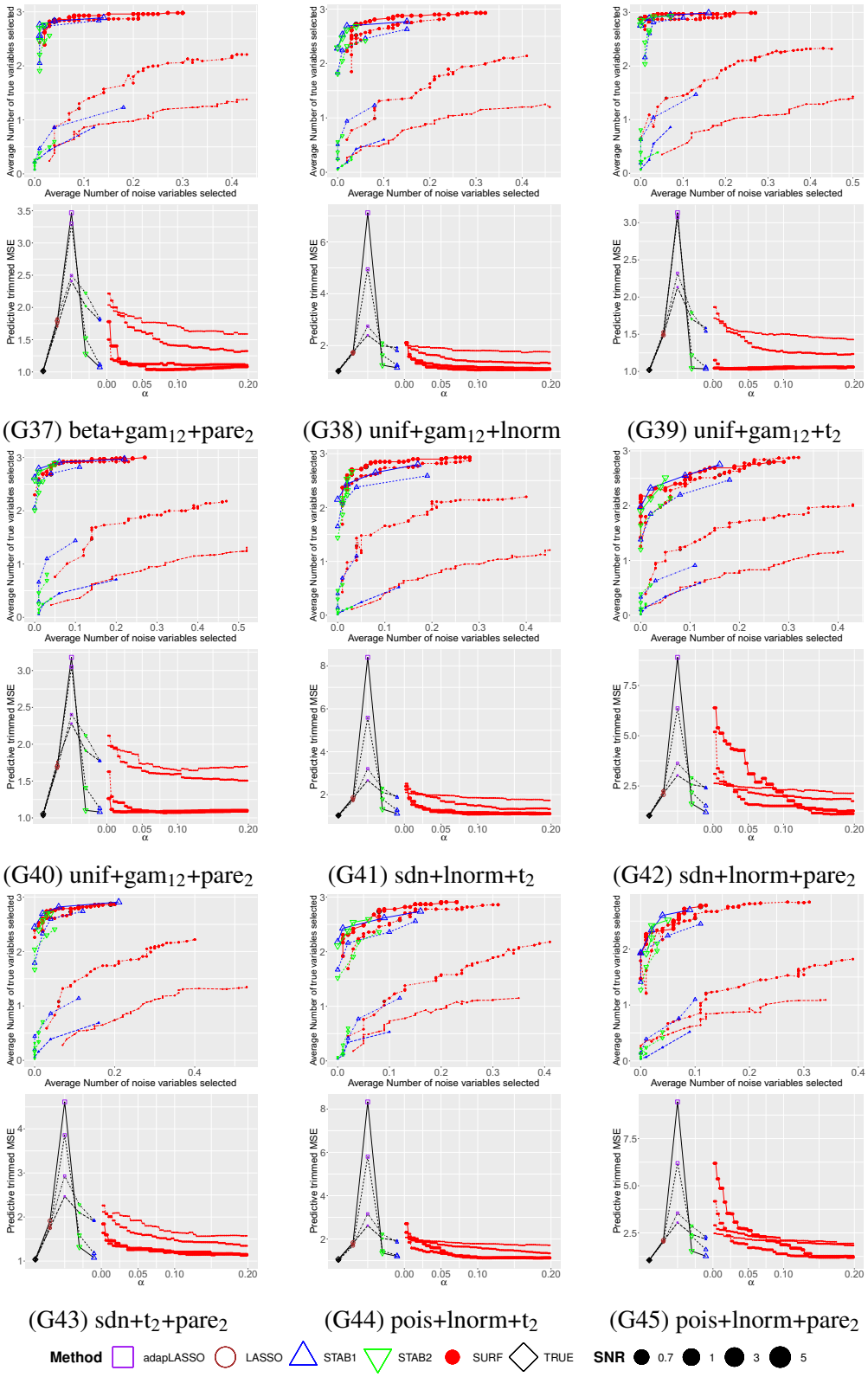


Figure A.2: Variable selection and prediction for multiple true variable Gaussian regression

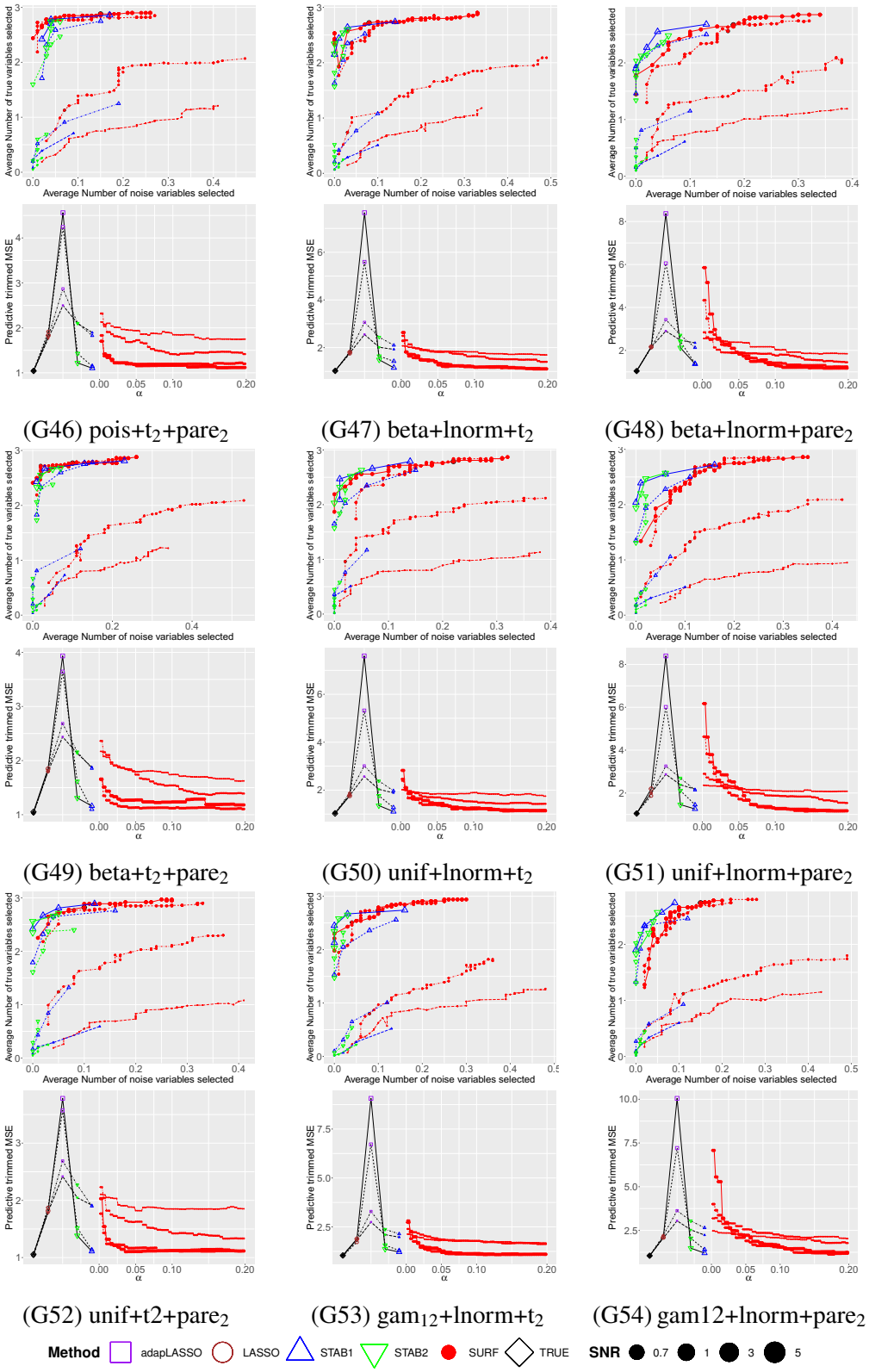


Figure A.2: Variable selection and prediction for multiple true variable Gaussian regression

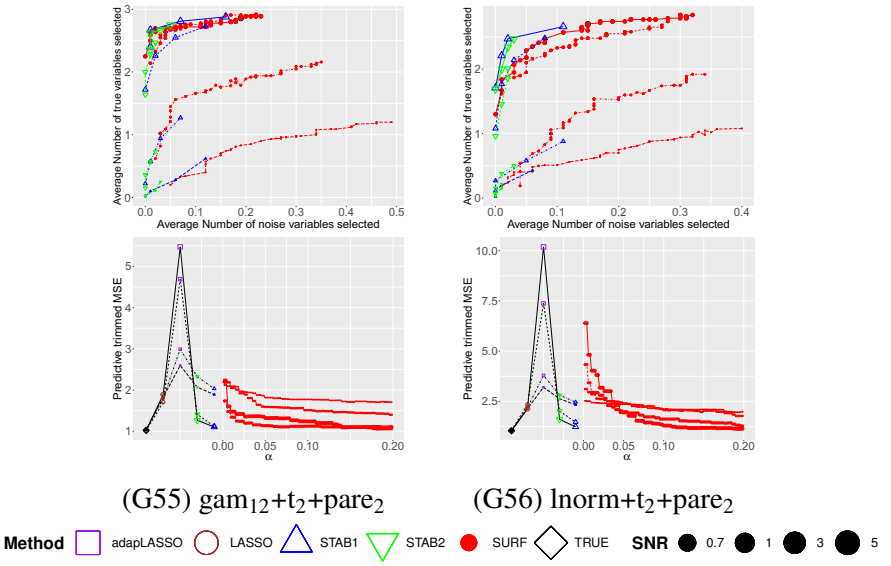


Figure A.2: Variable selection and prediction for multiple true variable Gaussian regression

A.1.3 Binomial single true variable cases

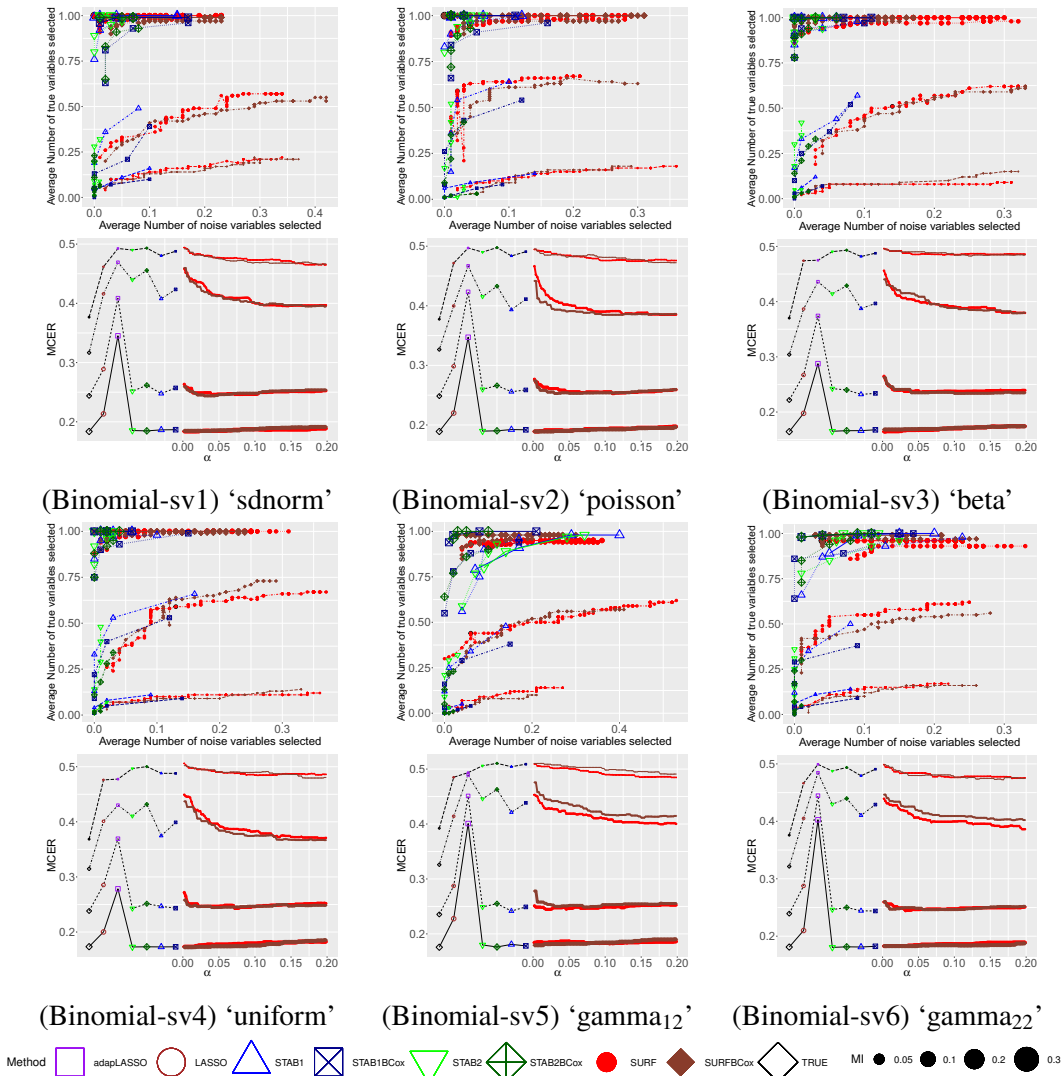


Figure A.3: Variable selection and prediction for a single true variable logistic regression

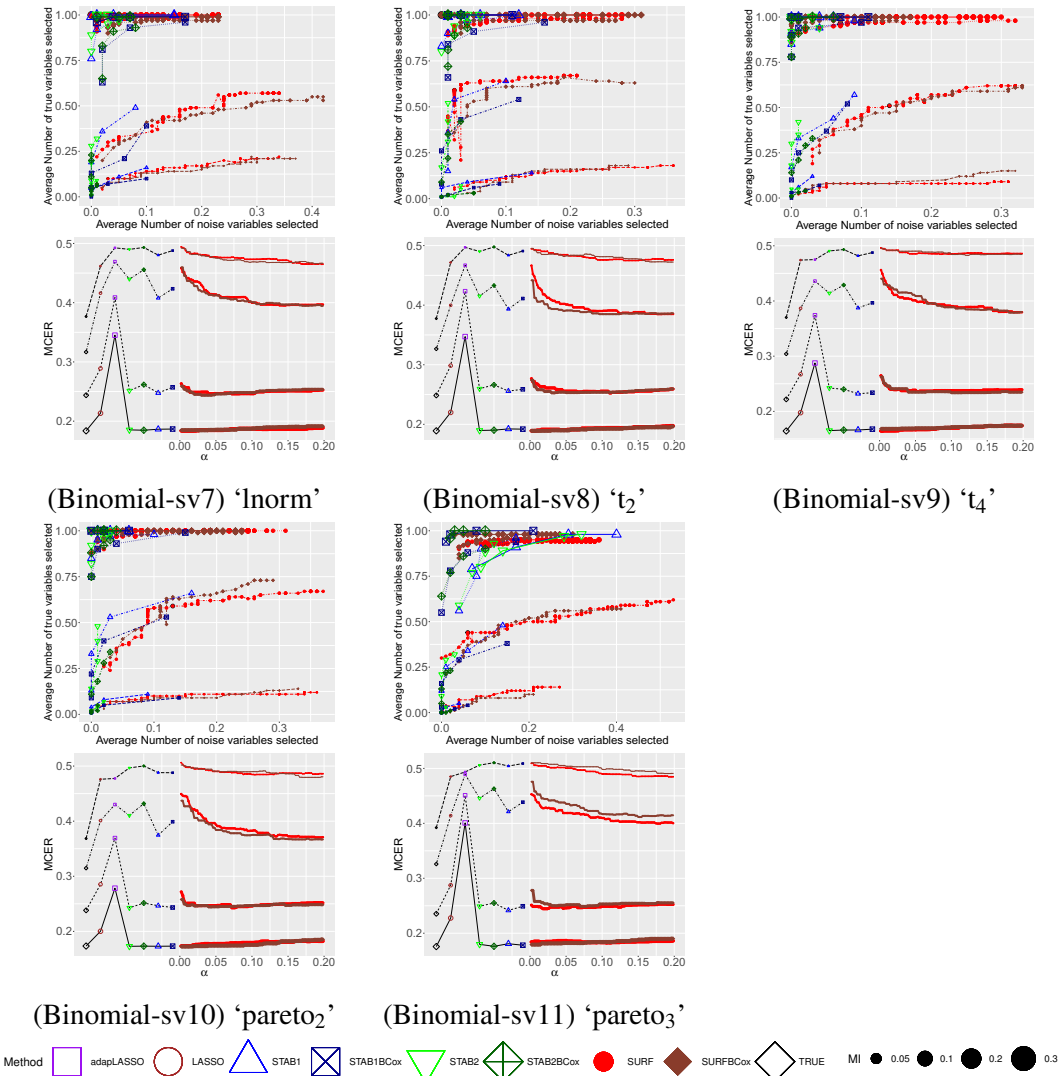


Figure A.3: Variable Selection and Prediction for a single true variable logistic regression

A.1.4 Binomial multiple true variable cases

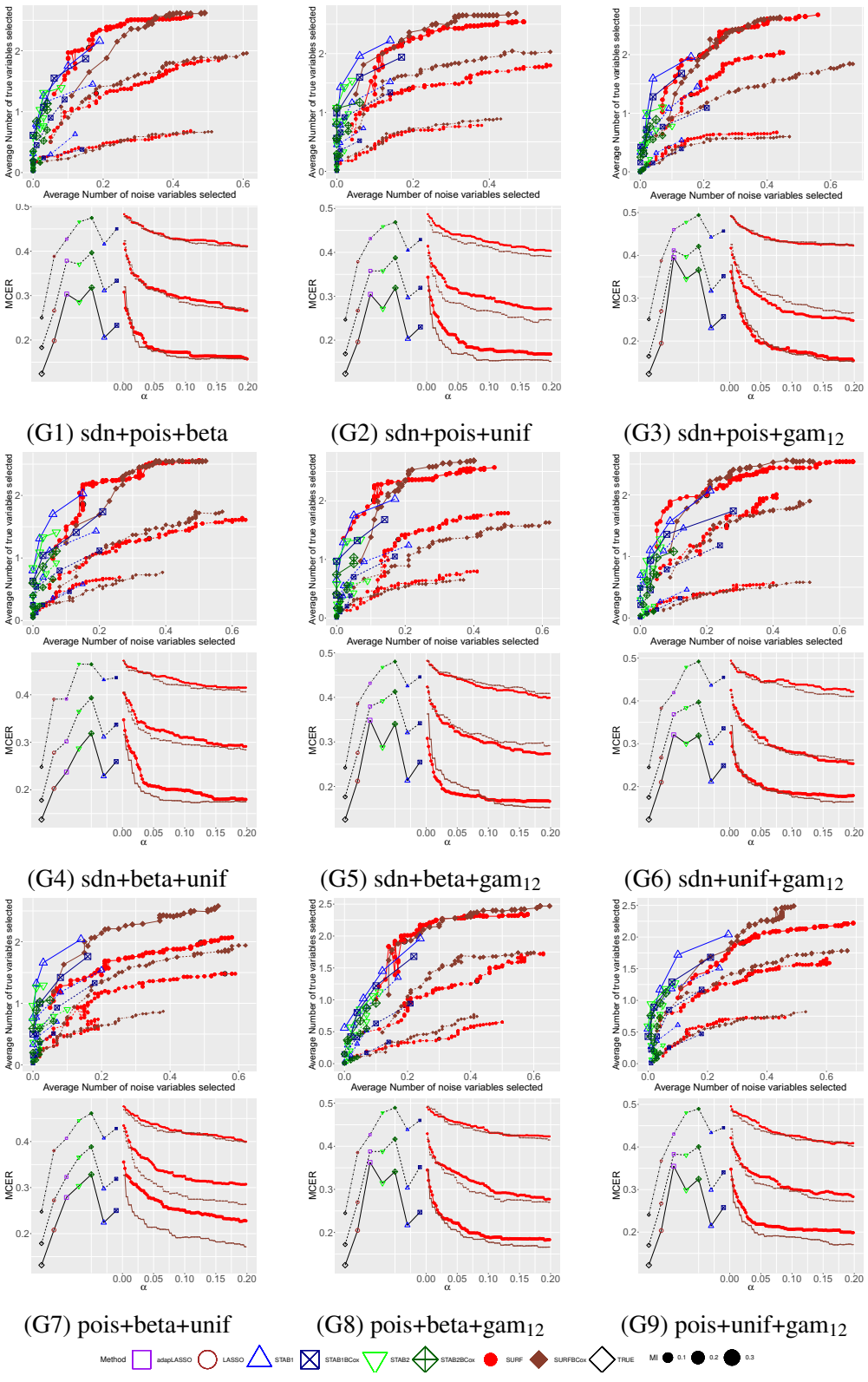


Figure A.4: Variable selection and prediction for multiple true variable logistic regression

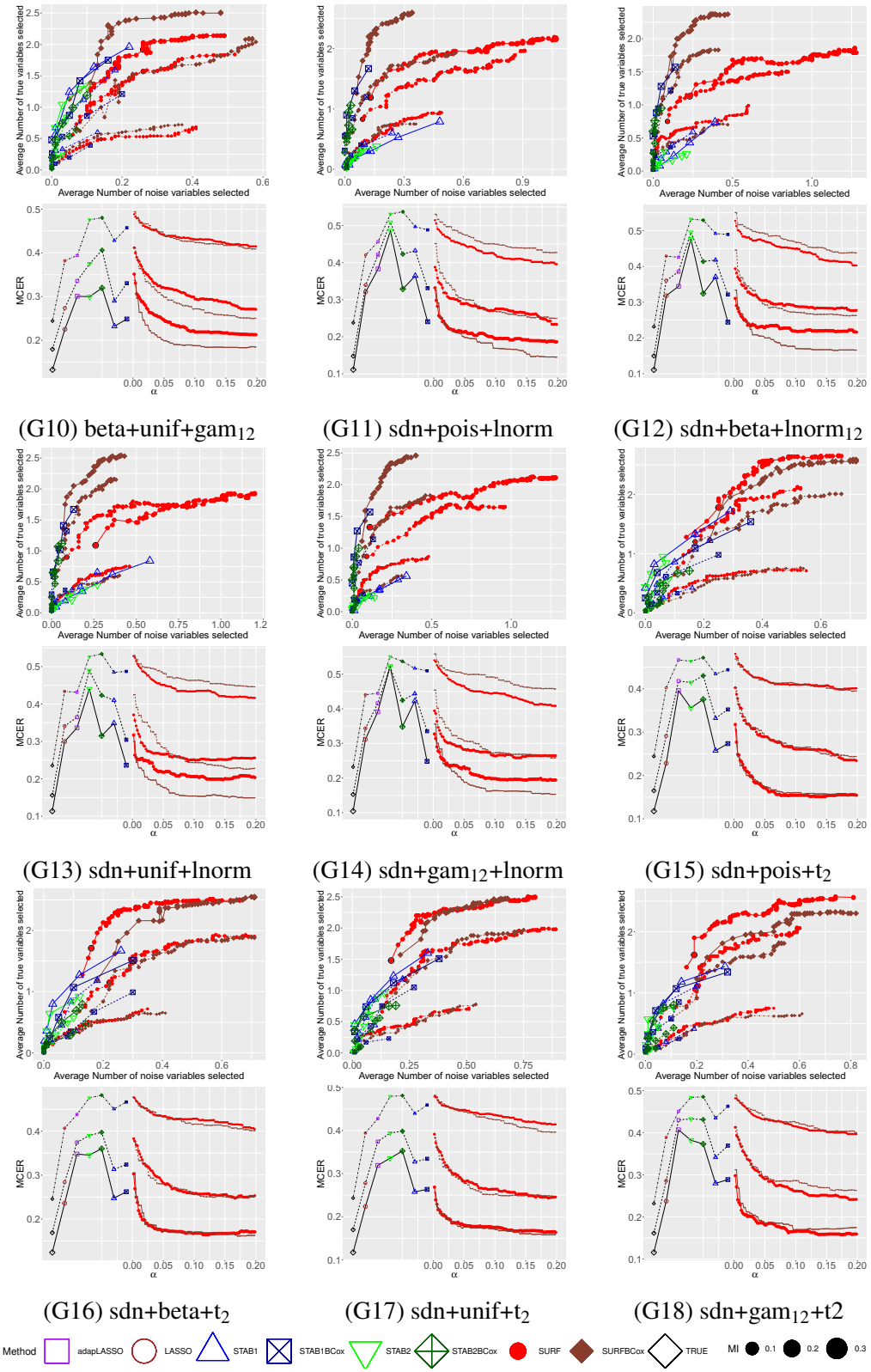


Figure A.4: Variable selection and prediction for multiple true variable logistic regression

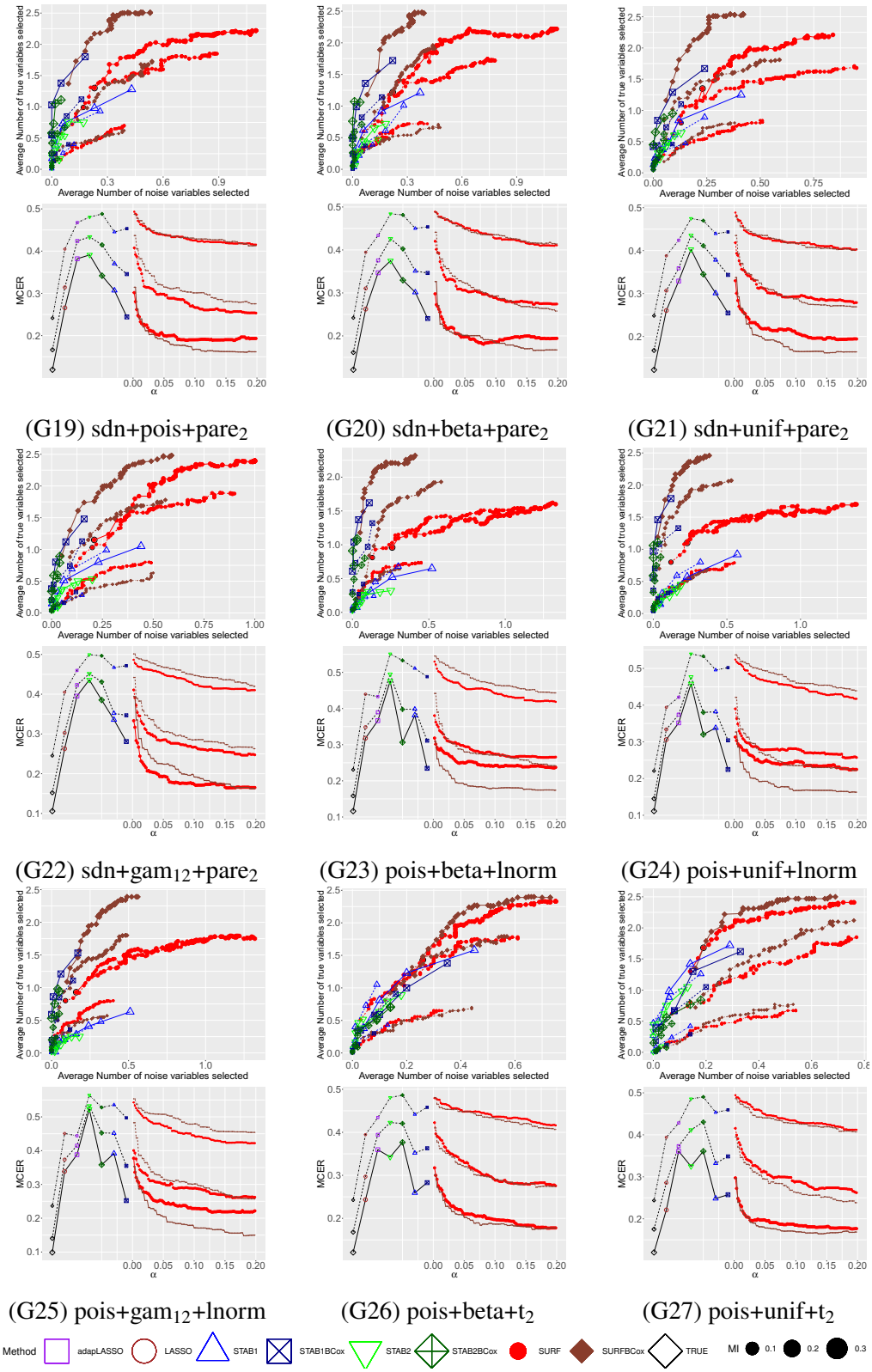


Figure A.4: Variable selection and prediction for multiple true variable logistic regression

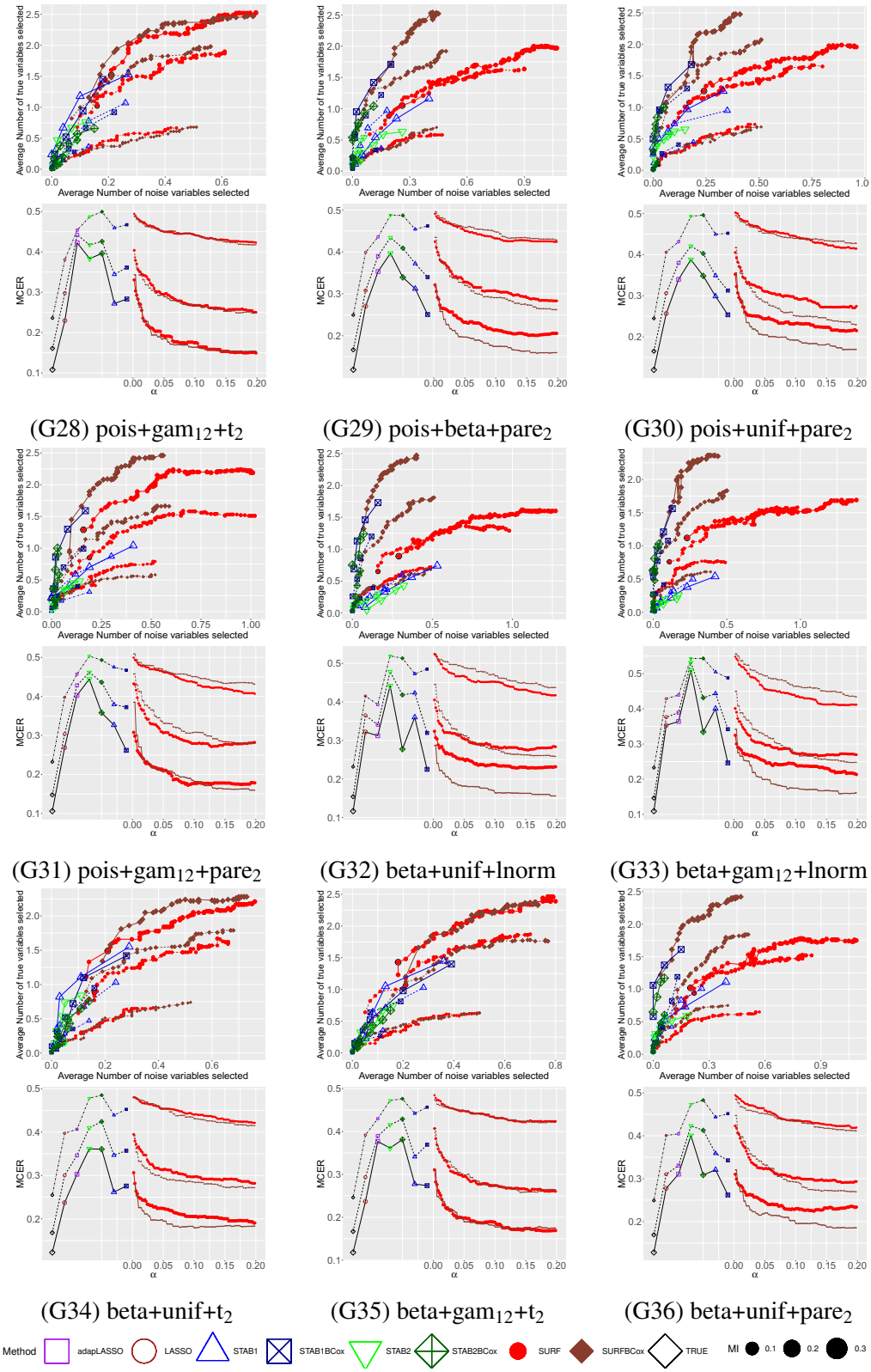


Figure A.4: Variable selection and prediction for multiple true variable logistic regression

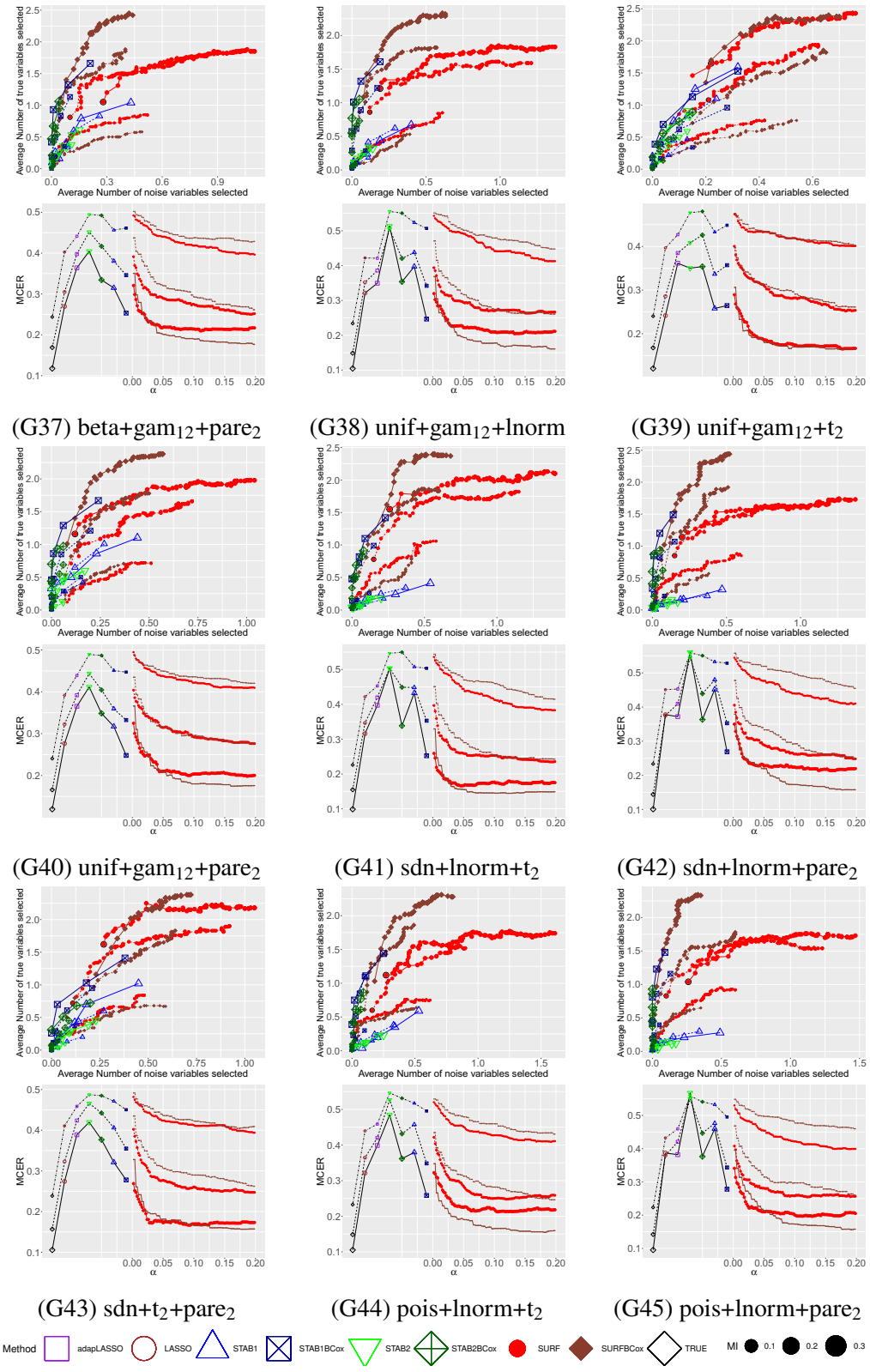


Figure A.4: Variable selection and prediction for multiple true variable logistic regression

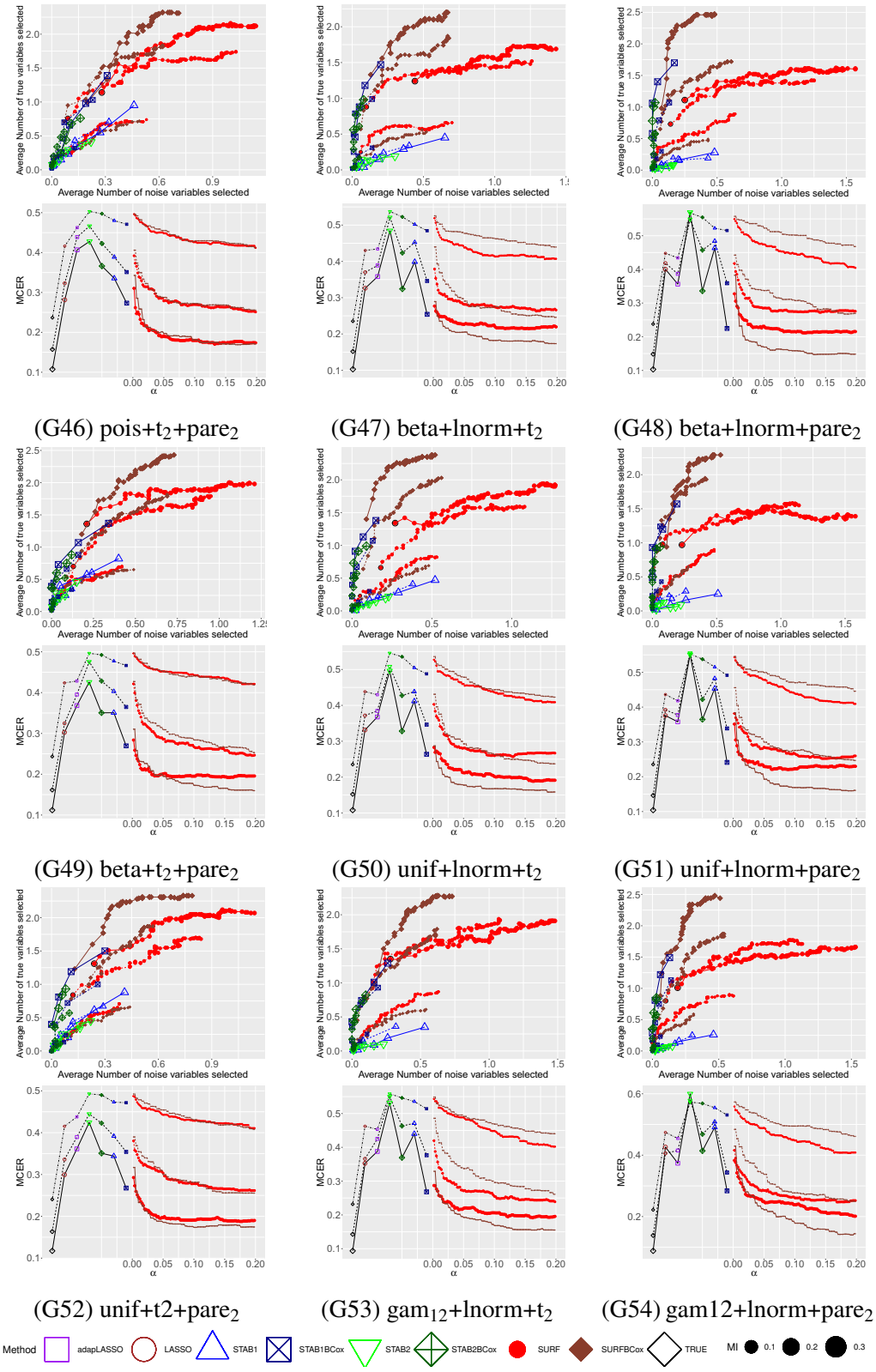


Figure A.4: Variable selection and prediction for multiple true variable logistic regression

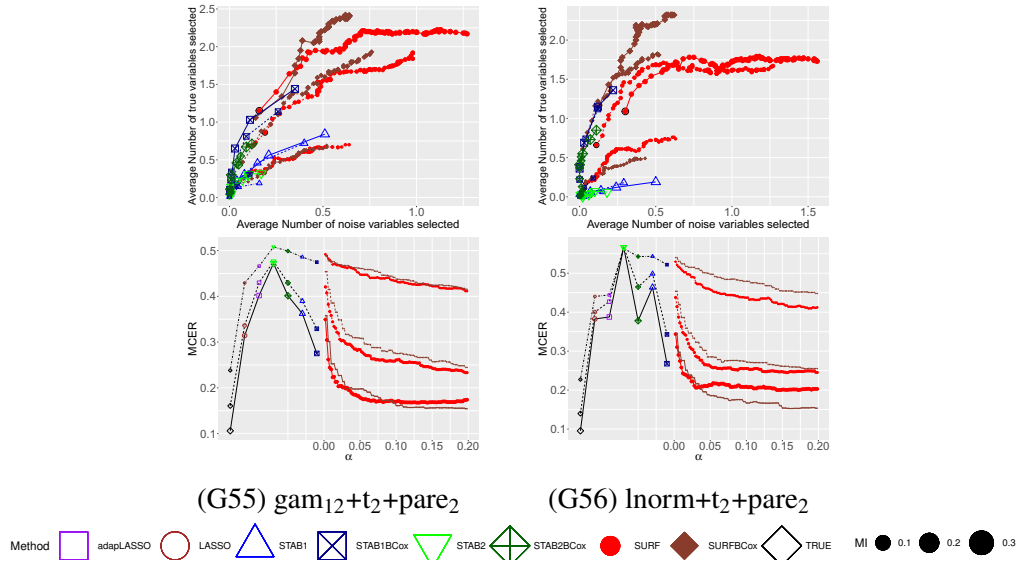
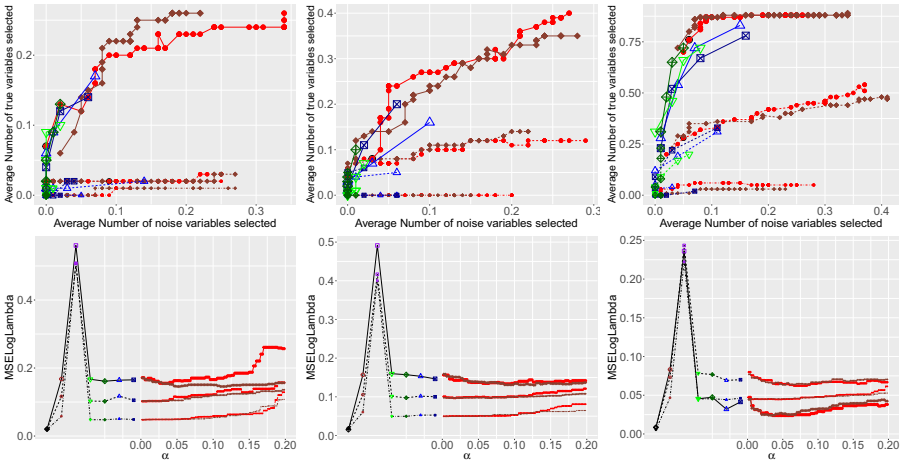
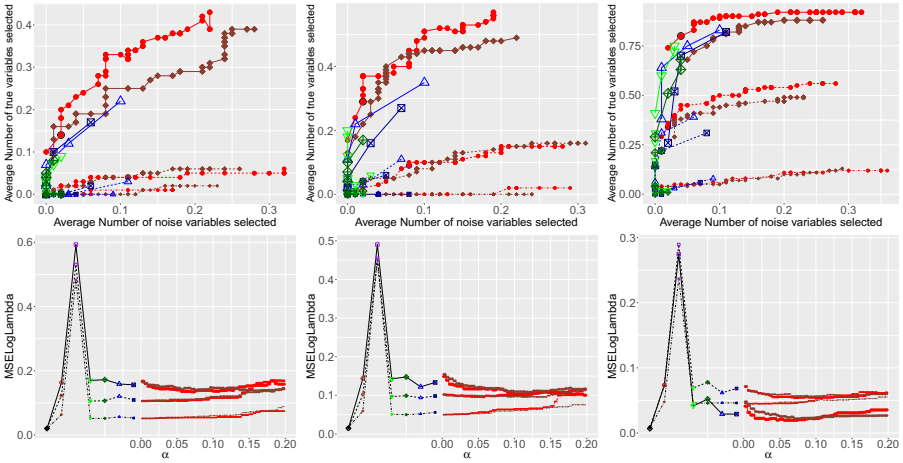


Figure A.4: Variable selection and prediction for multiple true variable logistic regression

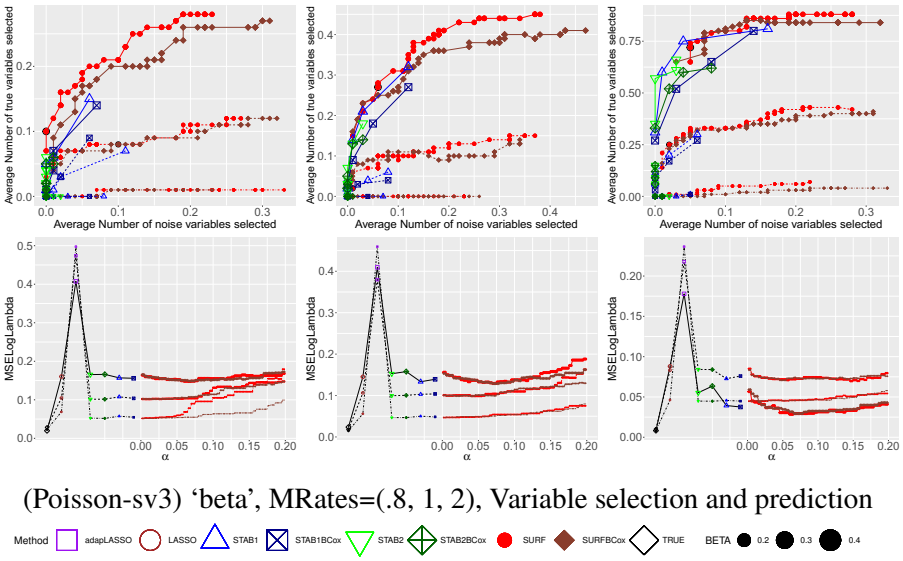
A.1.5 Poisson single true variable cases



(Poisson-sv1) 'sdnorm', MRates=(.8, 1, 2), Variable selection and prediction

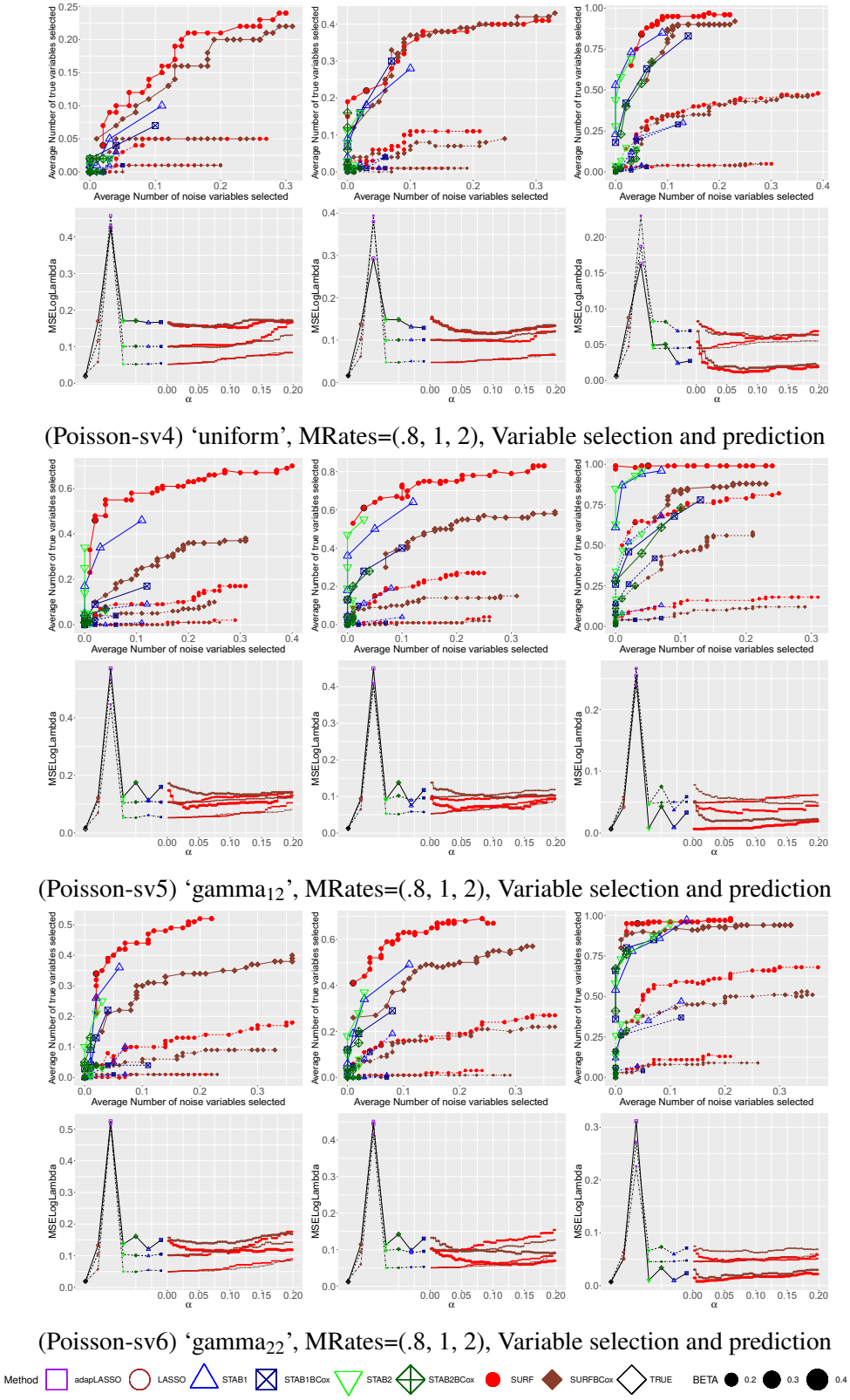


(Poisson-sv2) 'poisson', MRates=(.8, 1, 2), Variable selection and prediction



(Poisson-sv3) 'beta', MRates=(.8, 1, 2), Variable selection and prediction

Figure A.5: Variable selection and prediction for a single true variable Poisson regression

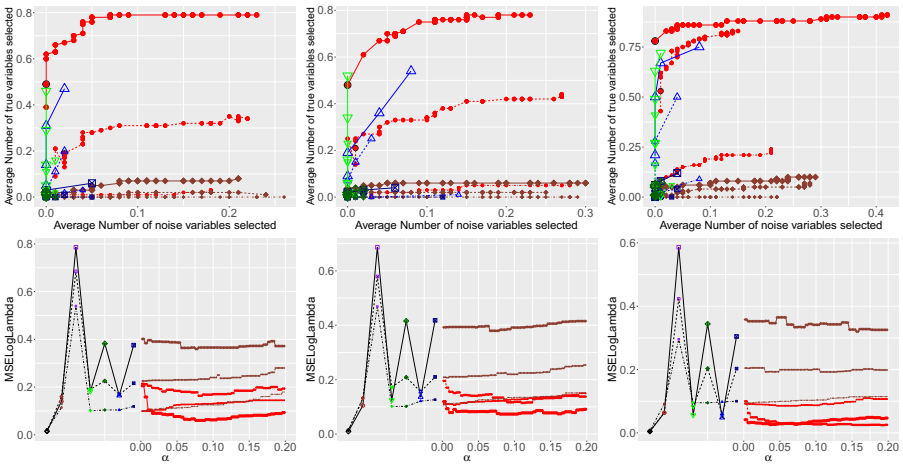


(Poisson-sv4) 'uniform', MRates=(.8, 1, 2), Variable selection and prediction

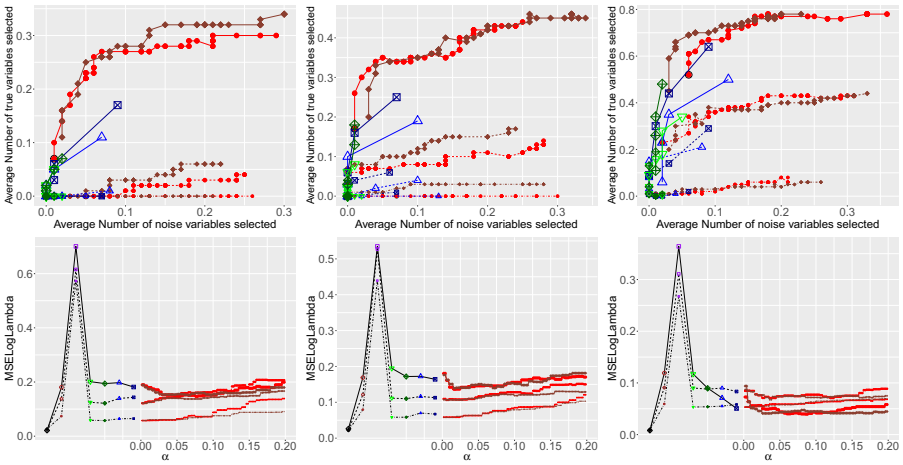
(Poisson-sv5) 'gamma$_{12}$', MRates=(.8, 1, 2), Variable selection and prediction

(Poisson-sv6) 'gamma$_{22}$', MRates=(.8, 1, 2), Variable selection and prediction

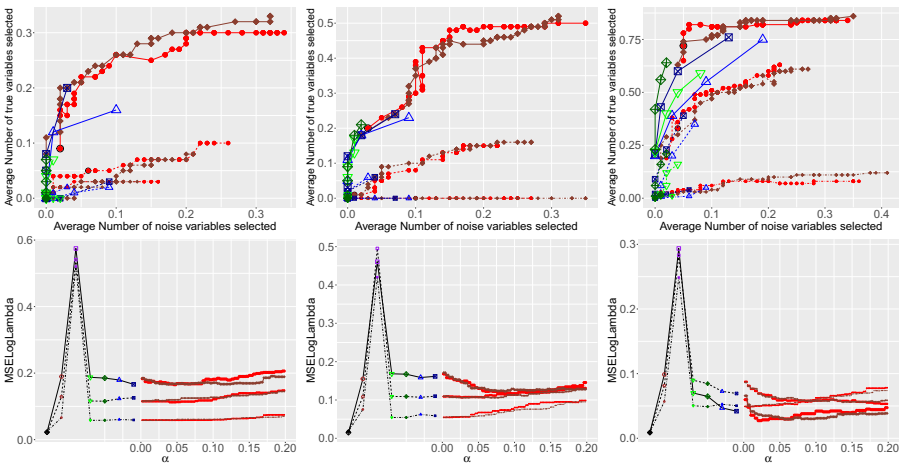
Figure A.5: Variable selection and prediction for a single true variable Poisson regression



(Poisson-sv7) 'lnorm', MRates=(.8, 1, 2), Variable selection and prediction

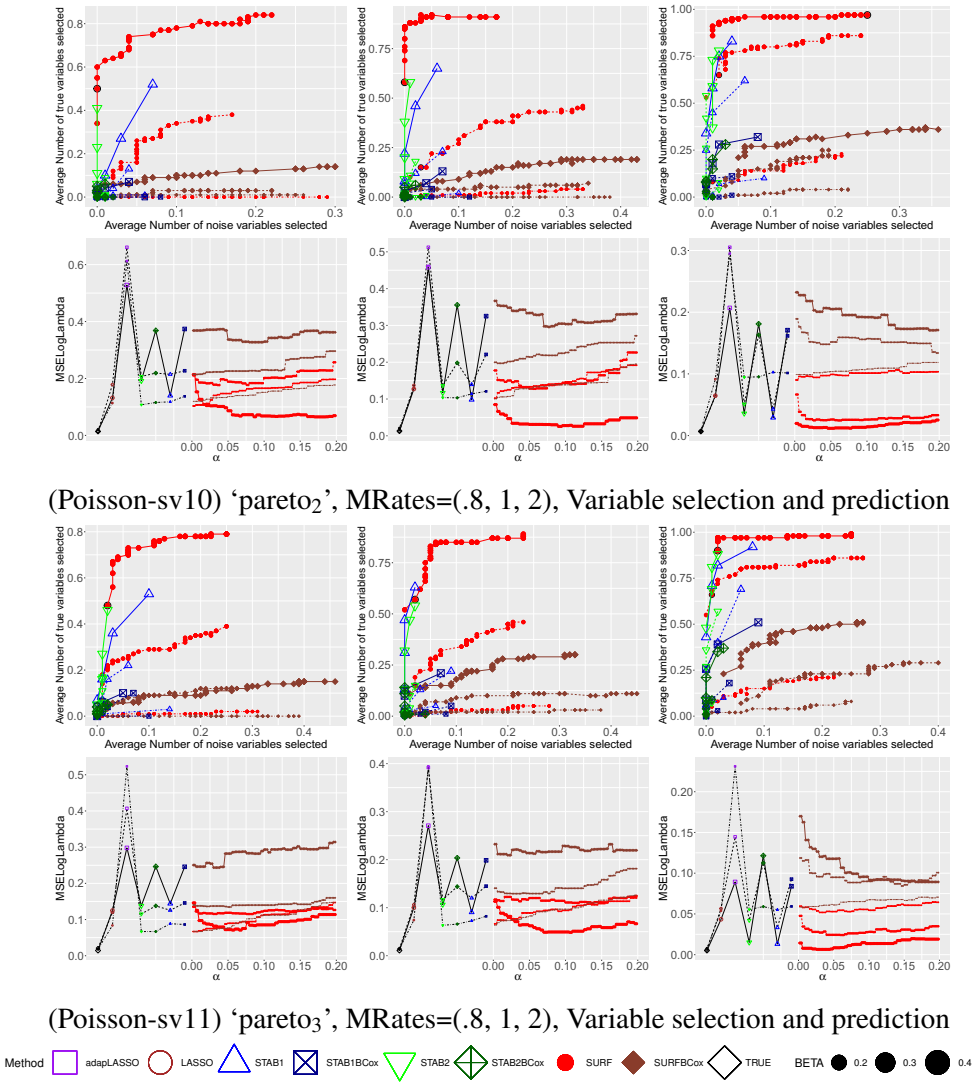


(Poisson-sv8) 't_2', MRates=(.8, 1, 2), Variable selection and prediction



(Poisson-sv9) 't_4', MRates=(.8, 1, 2), Variable selection and prediction

Figure A.5: Variable selection and prediction for a single true variable Poisson regression

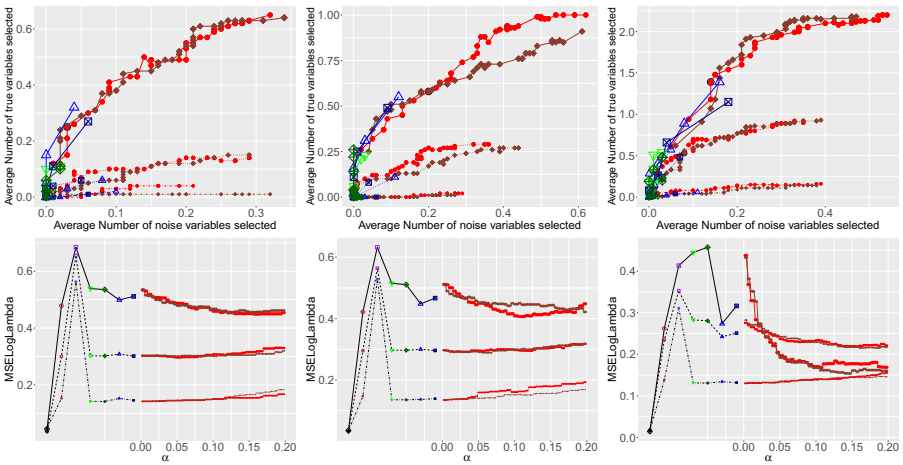


(Poisson-sv10) 'pareto$_2$', MRates=(.8, 1, 2), Variable selection and prediction

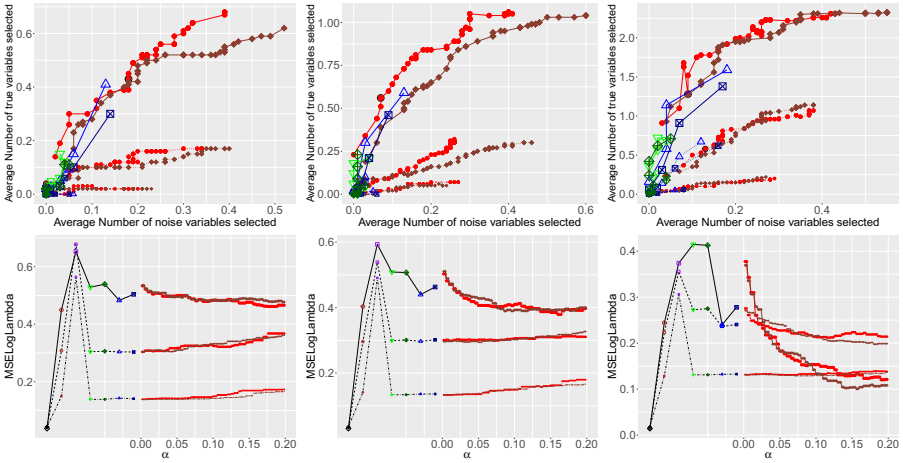
(Poisson-sv11) 'pareto$_3$', MRates=(.8, 1, 2), Variable selection and prediction

Figure A.5: Variable selection and prediction for a single true variable Poisson regression

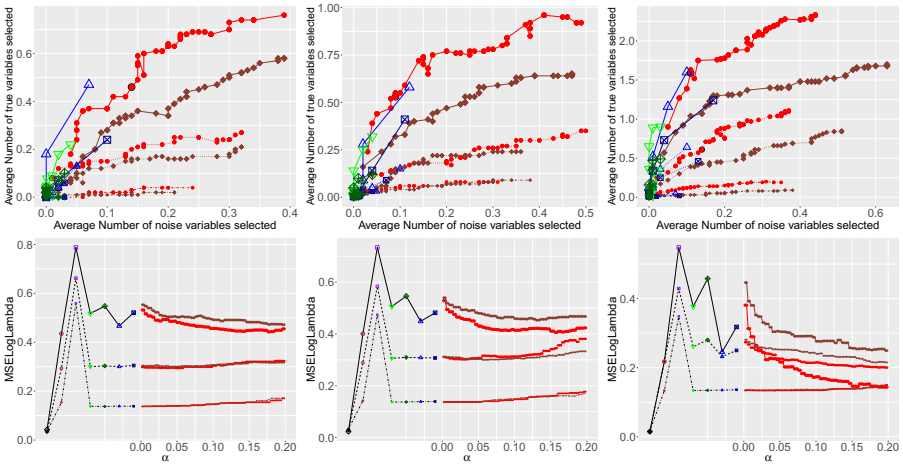
A.1.6 Poisson multiple true variable cases



(P1) sdn+pois+beta, MRates=(.8, 1, 2), Variable selection and prediction

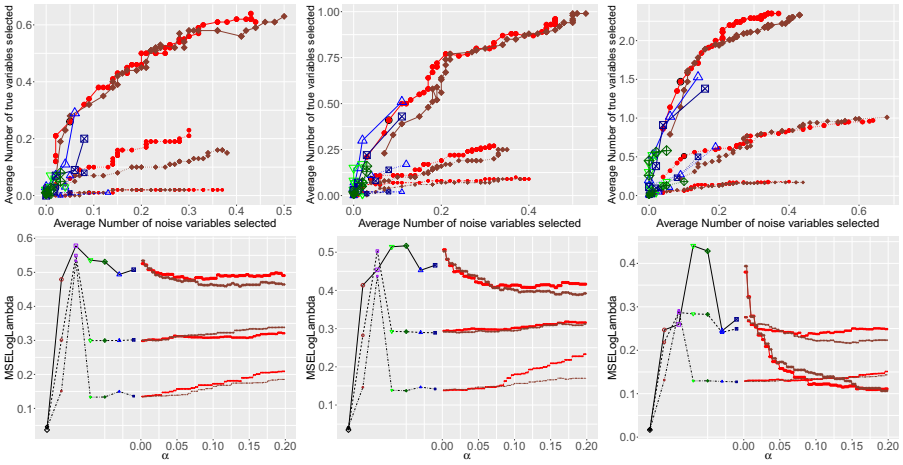


(P2) sdn+pois+unif, MRates=(.8, 1, 2), Variable selection and prediction

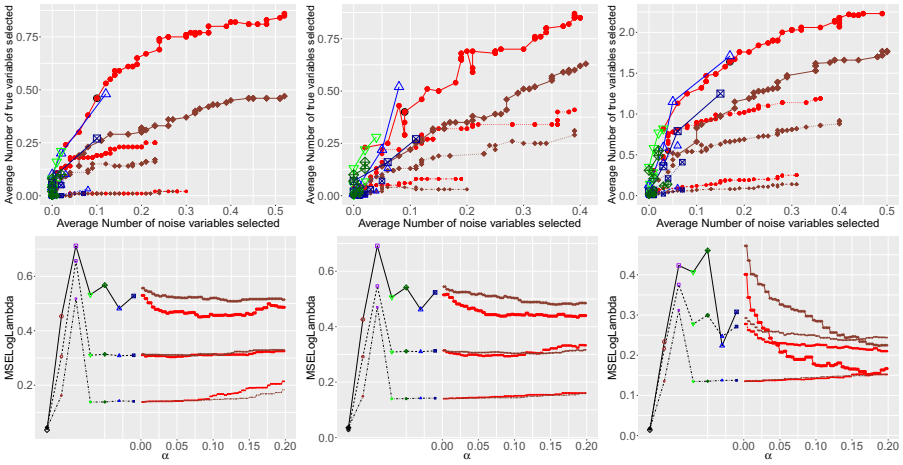


(P3) sdn+pois+gam$_{12}$, MRates=(.8, 1, 2), Variable selection and prediction

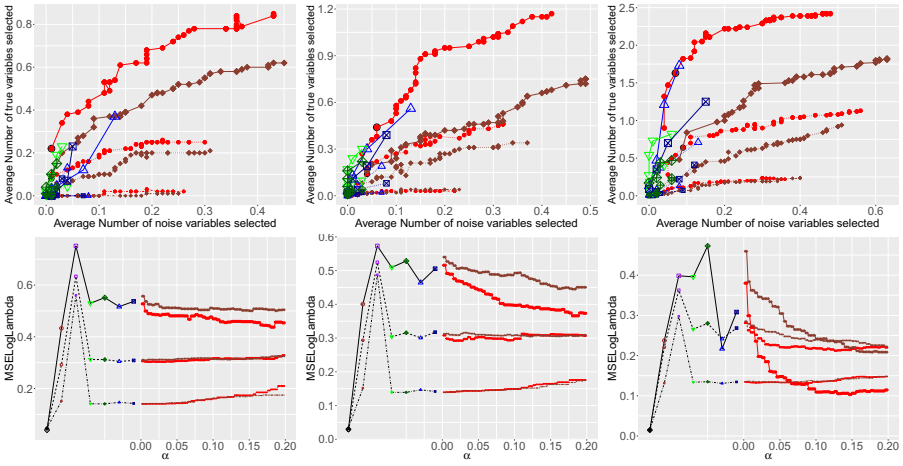
Figure A.6: Variable selection and prediction for multiple true variable Poisson regression



(P4) sdn+beta+unif, MRates=(.8, 1, 2), Variable selection and prediction

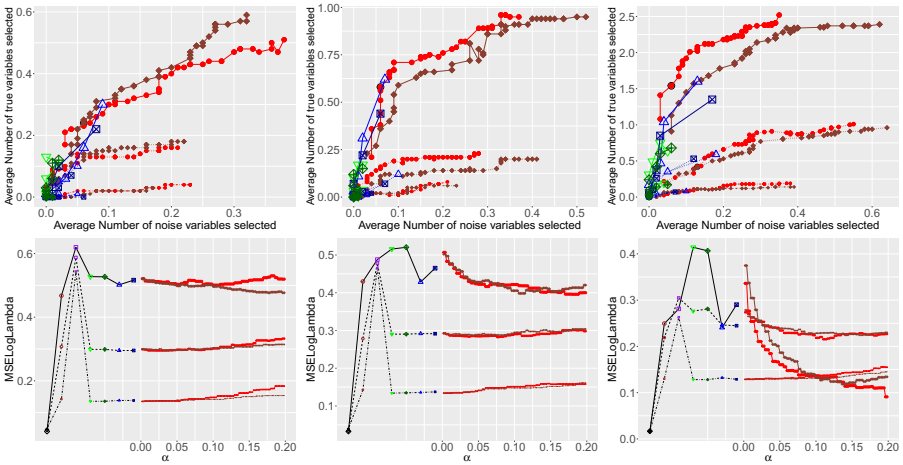


(P5) sdn+beta+gam$_{12}$, MRates=(.8, 1, 2), Variable selection and prediction

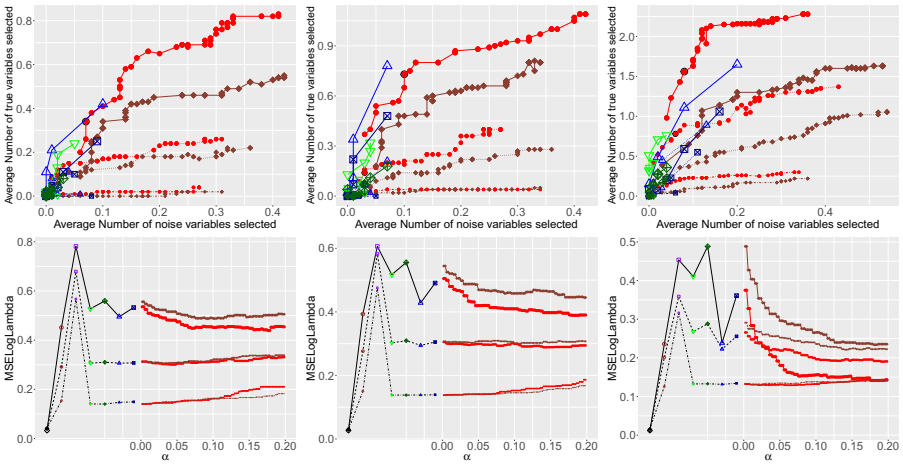


(P6) sdn+unif+gam$_{12}$, MRates=(.8, 1, 2), Variable selection and prediction

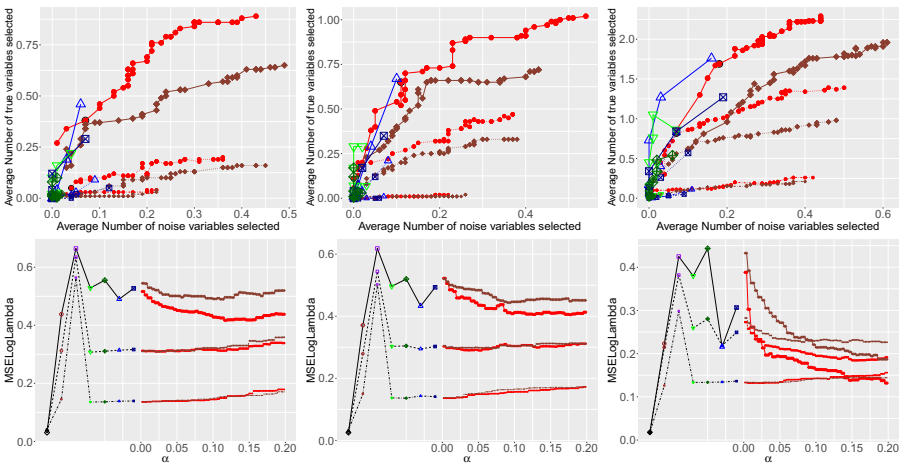
Figure A.6: Variable selection and prediction for multiple true variable Poisson regression



(P7) pois+beta+unif, MRates=(.8, 1, 2), Variable selection and prediction

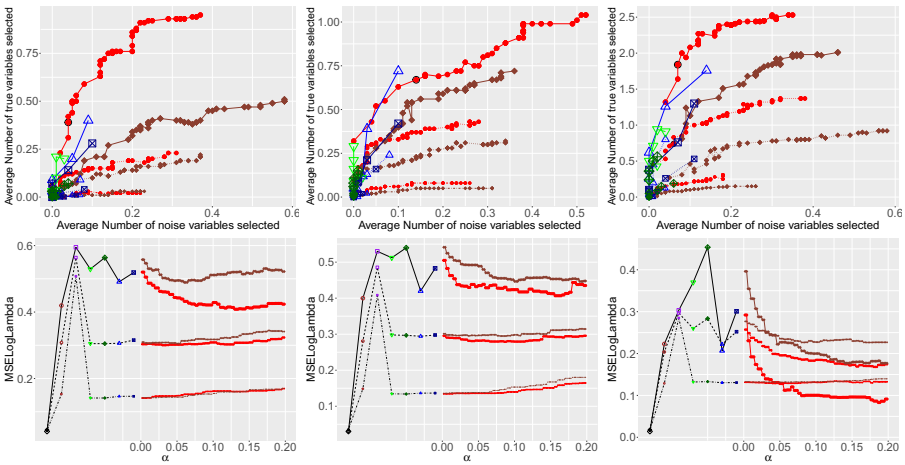


(P8) pois+beta+gam$_{12}$, MRates=(.8, 1, 2), Variable selection and prediction

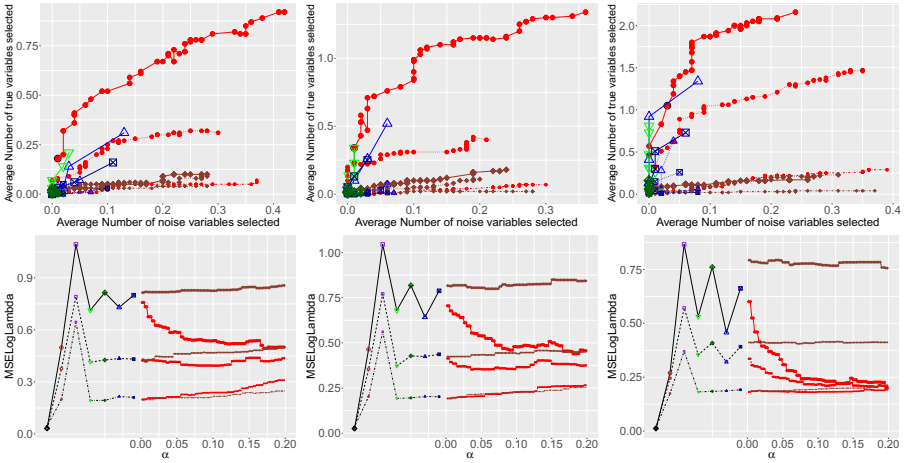


(P9) pois+unif+gam$_{12}$, MRates=(.8, 1, 2), Variable selection and prediction

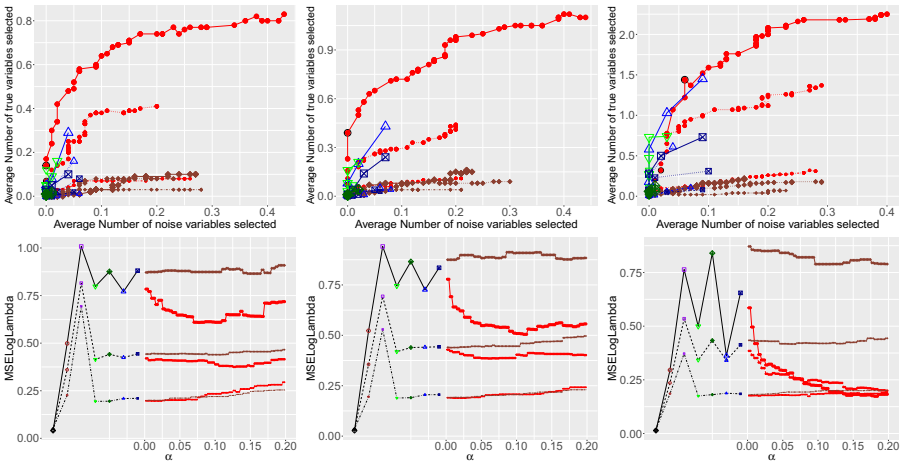
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P10) beta+unif+gam$_{12}$, MRates=(.8, 1, 2), Variable selection and prediction



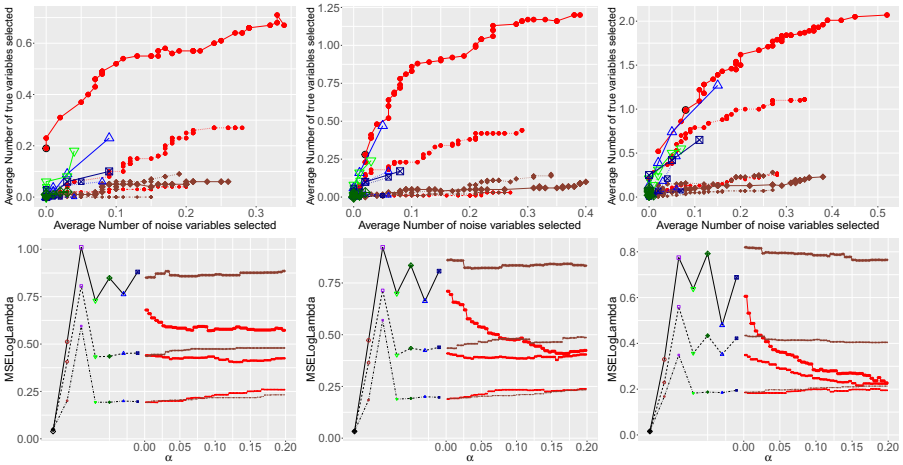
(P11) sdn+pois+lnorm, MRates=(.8, 1, 2), Variable selection and prediction



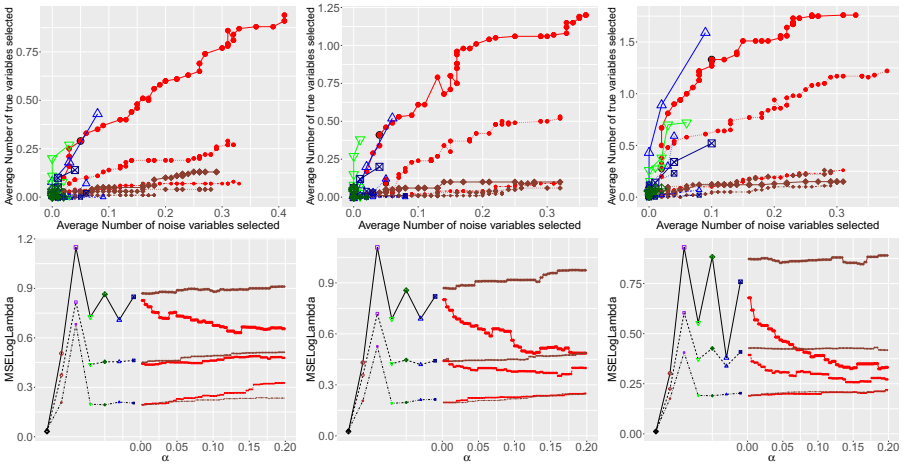
(P12) sdn+beta+lnorm, MRates=(.8, 1, 2), Variable selection and prediction

*

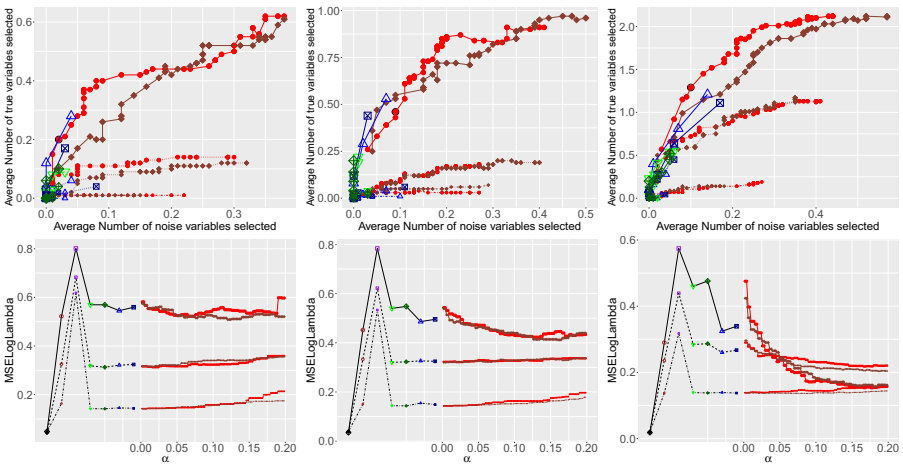
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P13) sdn+unif+lnorm, MRates=(.8, 1, 2), Variable selection and prediction

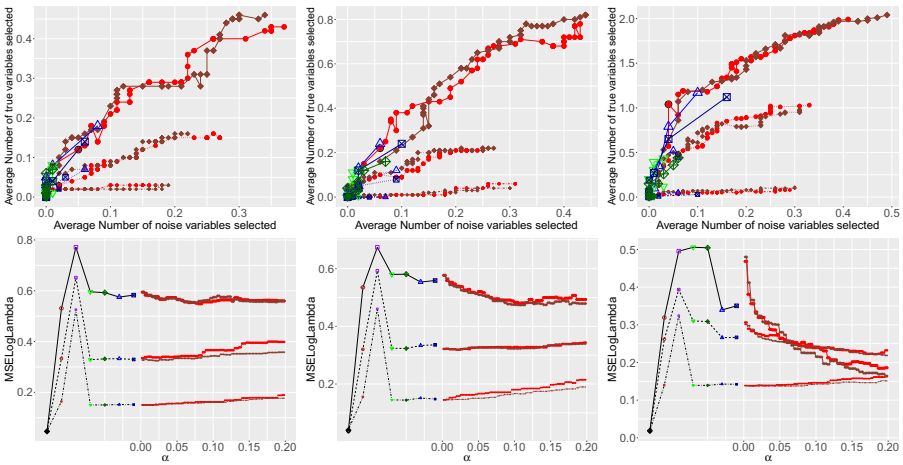


(P14) sdn+gam$_{12}$+lnorm, MRates=(.8, 1, 2), Variable selection and prediction

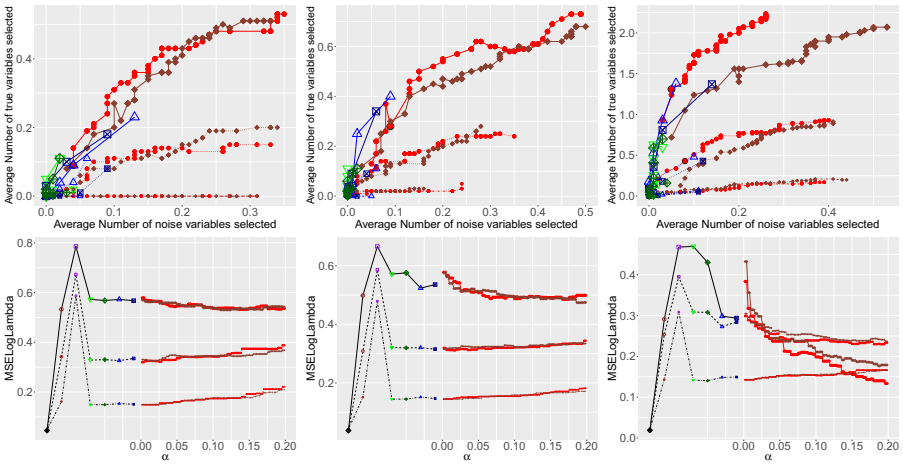


(P15) sdn+pois+t$_2$, MRates=(.8, 1, 2), Variable selection and prediction

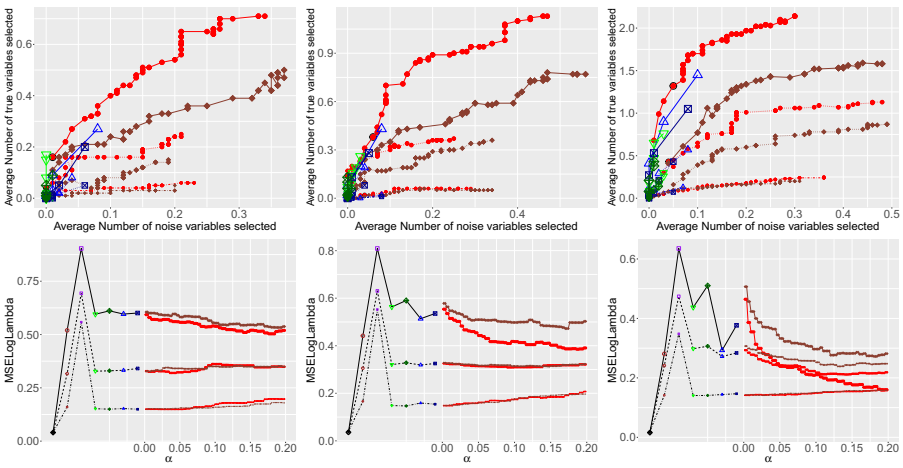
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P16) sdn+beta+t_2, MRates=(.8, 1, 2), Variable selection and prediction

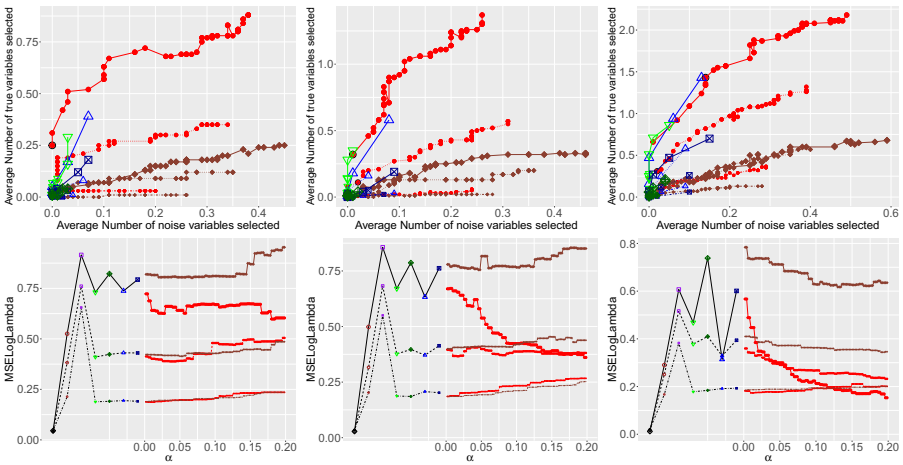


(P17) sdn+unif+t_2, MRates=(.8, 1, 2), Variable selection and prediction

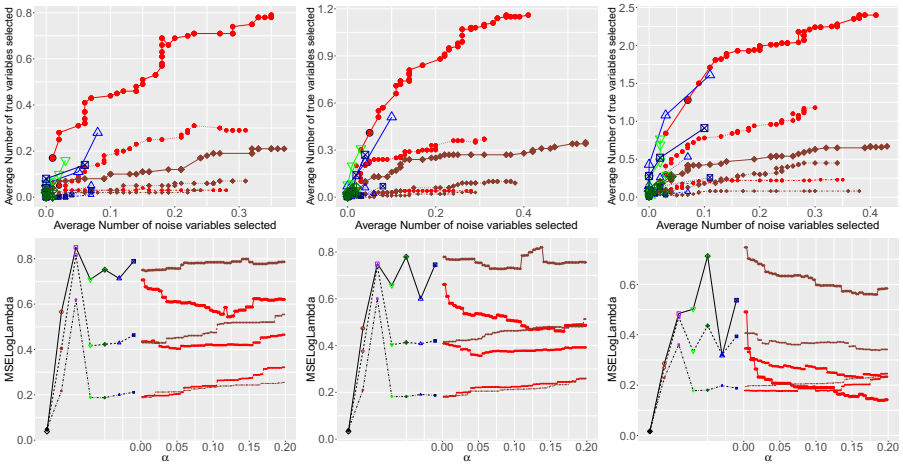


(P18) sdn+gam_{12}+t2, MRates=(.8, 1, 2), Variable selection and prediction

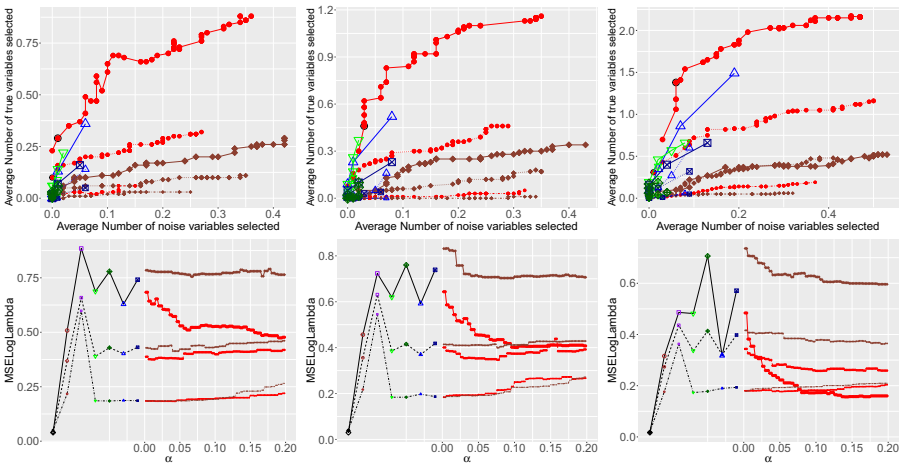
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P19) sdn+pois+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

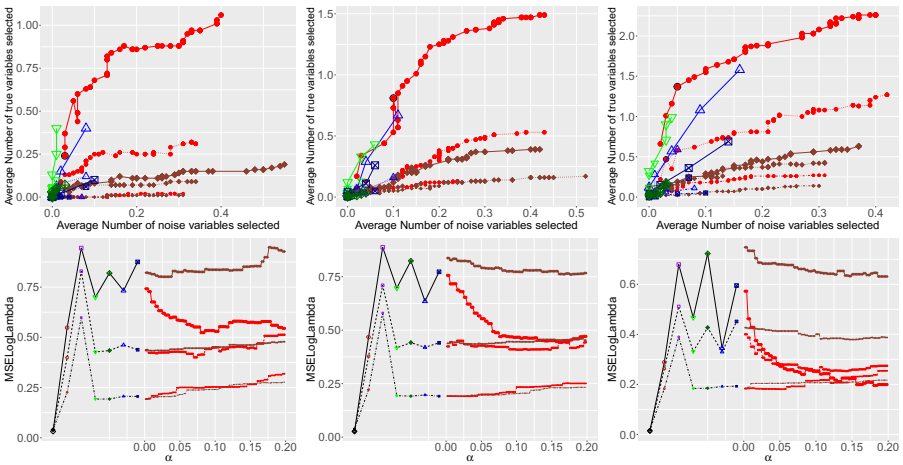


(P20) sdn+beta+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

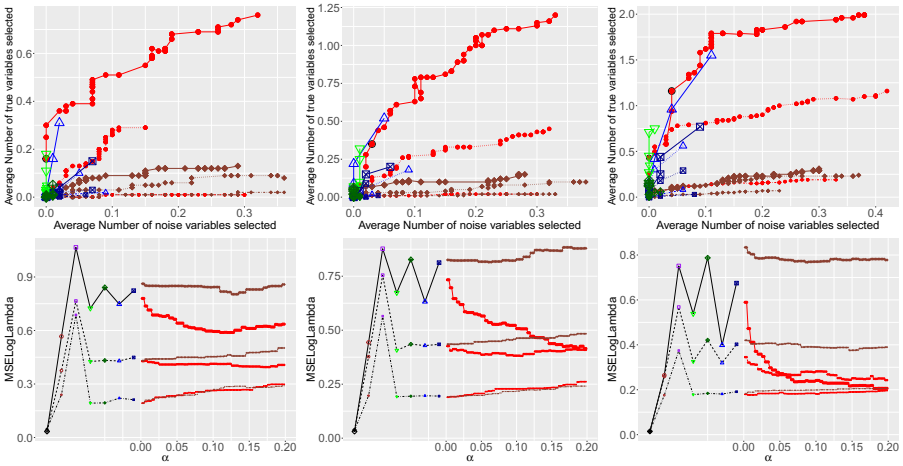


(P21) sdn+unif+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

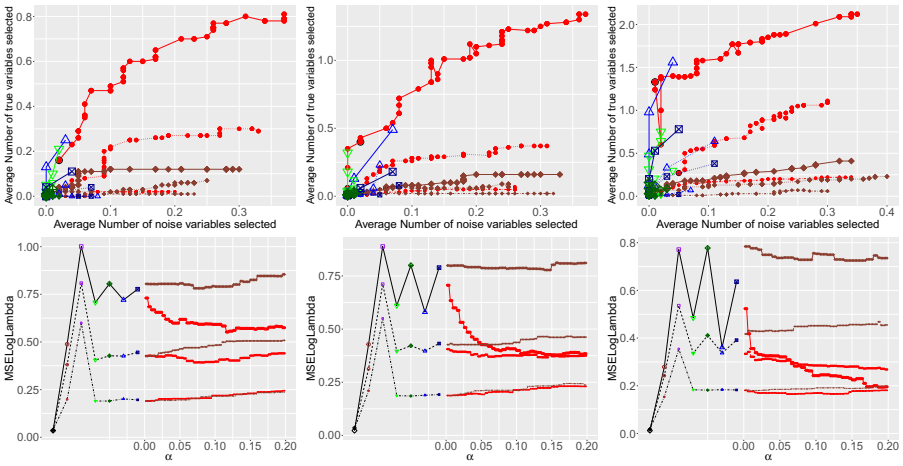
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P22) sdn+gam$_{12}$+pare$_2$ MRates=(.8, 1, 2), Variable selection and prediction

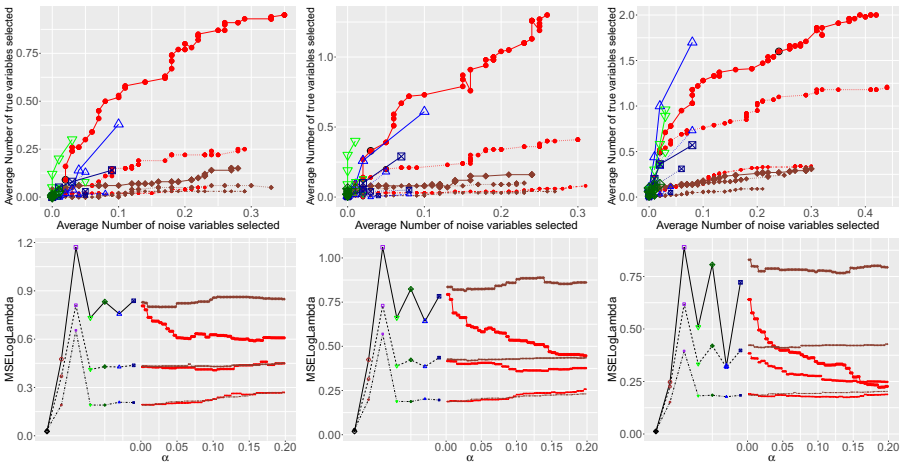


(P23) pois+beta+lnorm, MRates=(.8, 1, 2), Variable selection and prediction

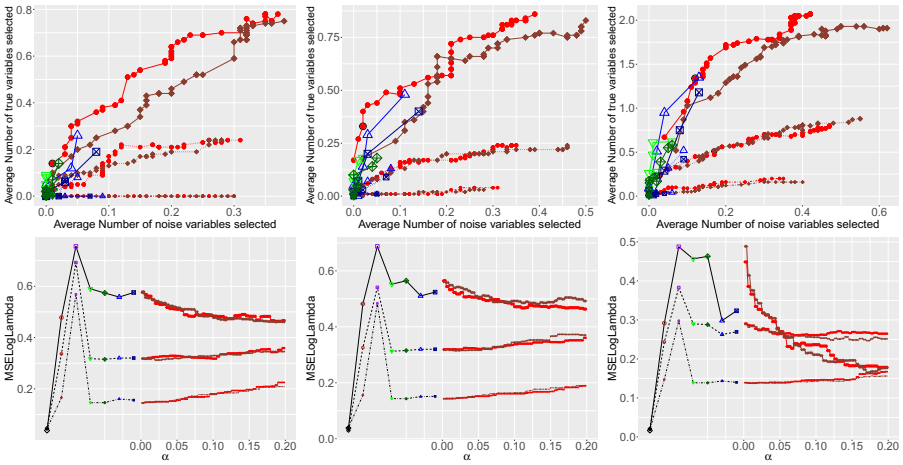


(P24) pois+unif+lnorm, MRates=(.8, 1, 2), Variable selection and prediction

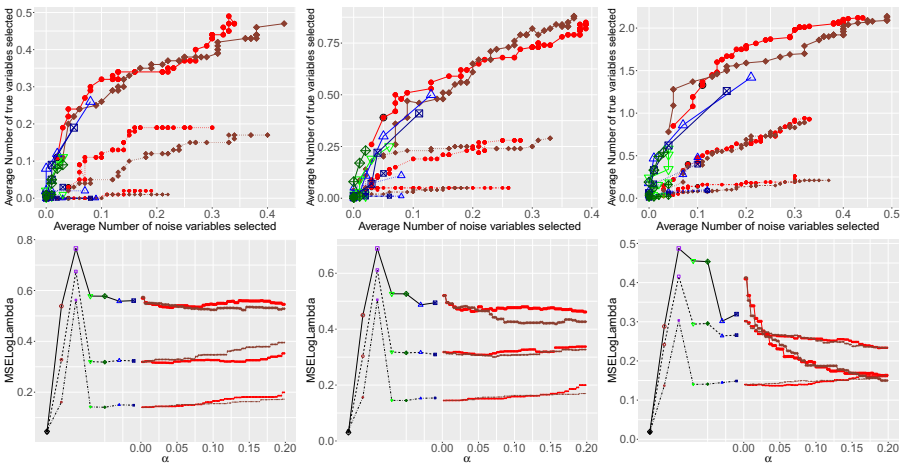
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P25) pois+gam$_{12}$+lnorm, MRates=(.8, 1, 2), Variable selection and prediction

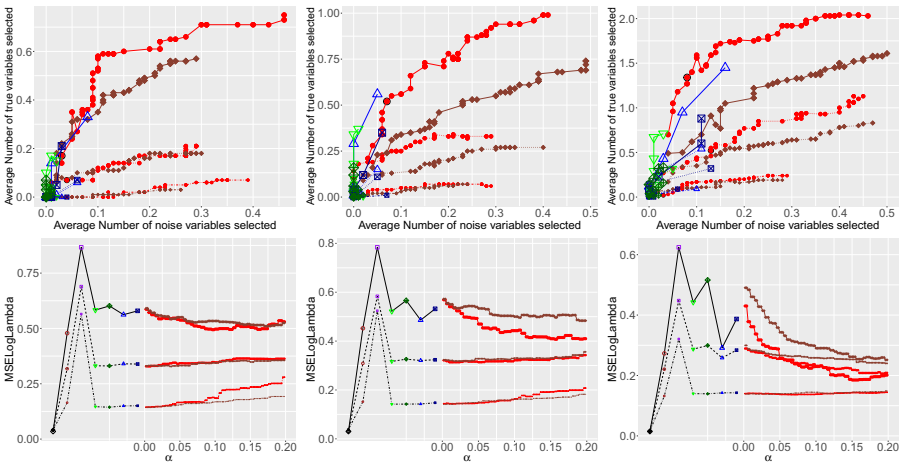


(P26) pois+beta+t$_2$, MRates=(.8, 1, 2), Variable selection and prediction

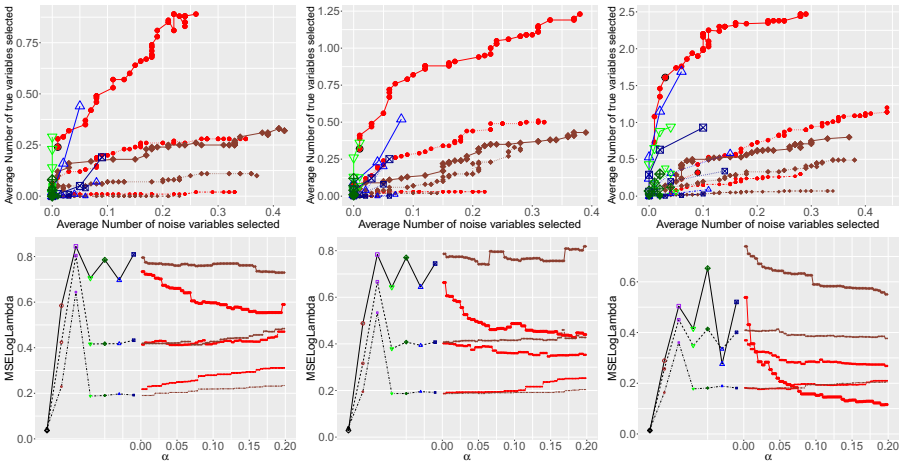


(P27) pois+unif+t$_2$, MRates=(.8, 1, 2), Variable selection and prediction

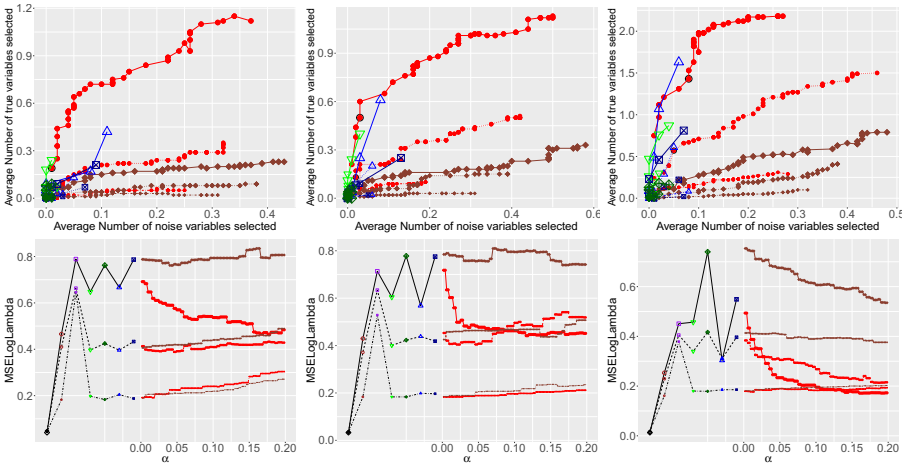
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P28) pois+gam$_{12}$+t$_2$, MRates=(.8, 1, 2), Variable selection and prediction

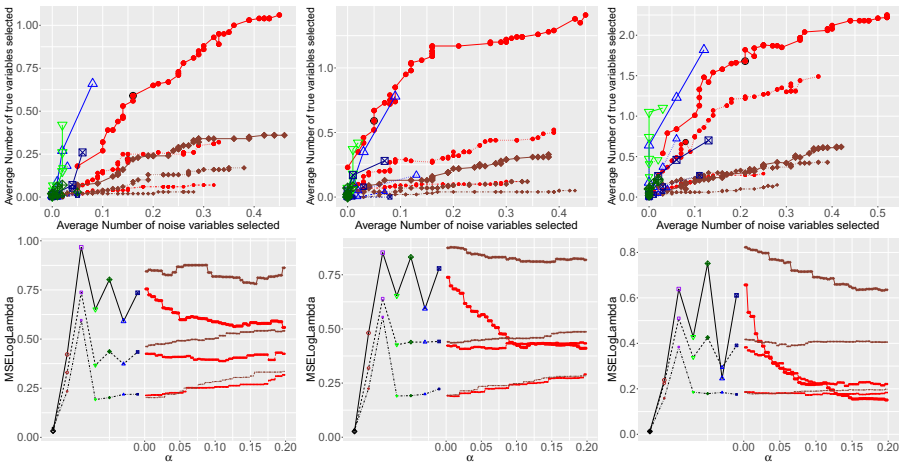


(P29) pois+beta+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

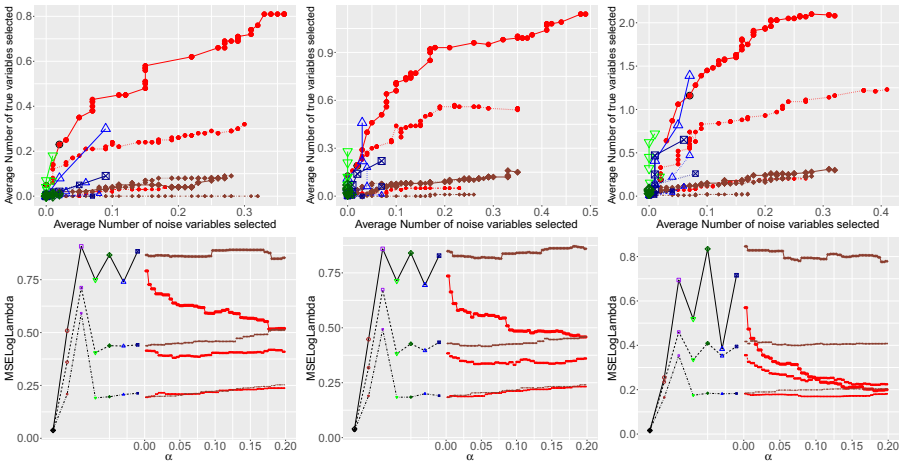


(P30) pois+unif+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

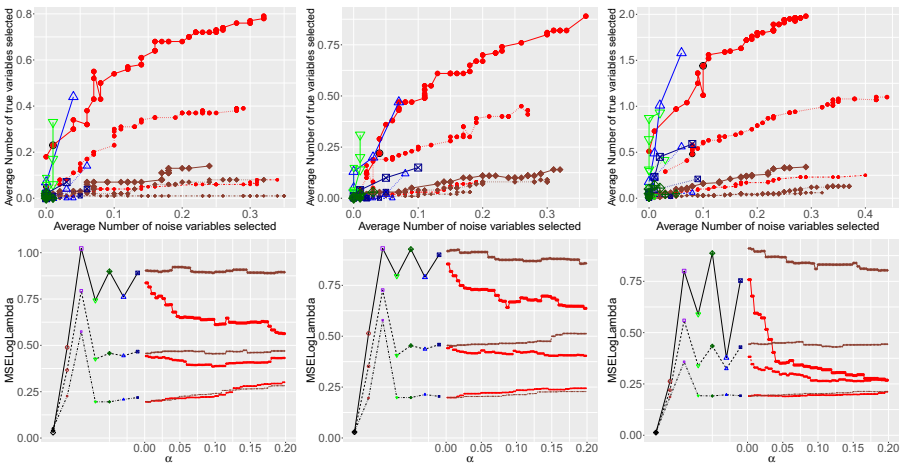
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P31) pois+gam$_{12}$+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

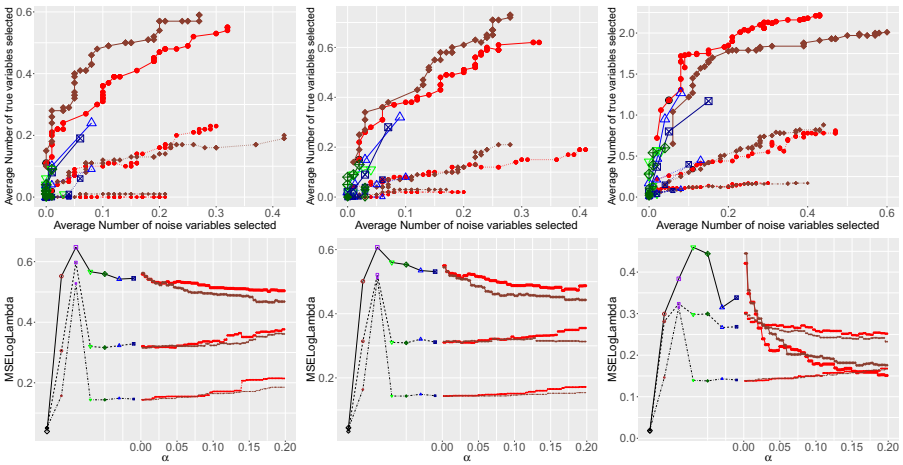


(P32) beta+unif+lnorm, MRates=(.8, 1, 2), Variable selection and prediction

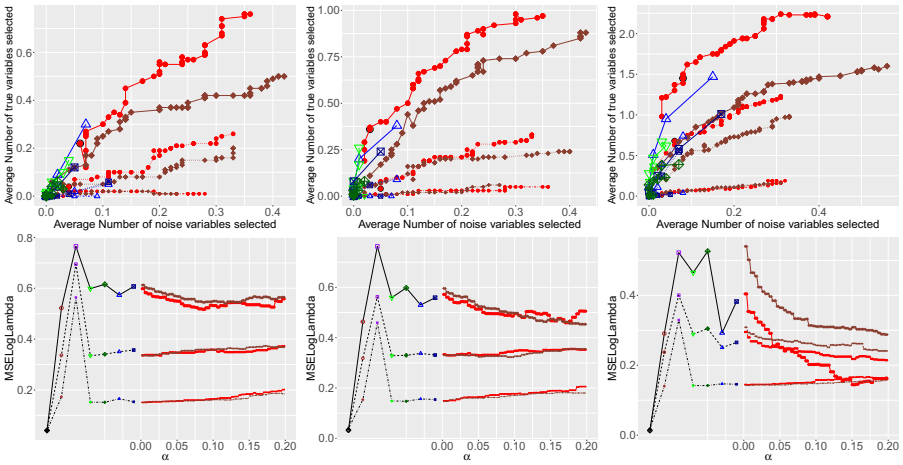


(P33) beta+gam$_{12}$+lnorm, MRates=(.8, 1, 2), Variable selection and prediction

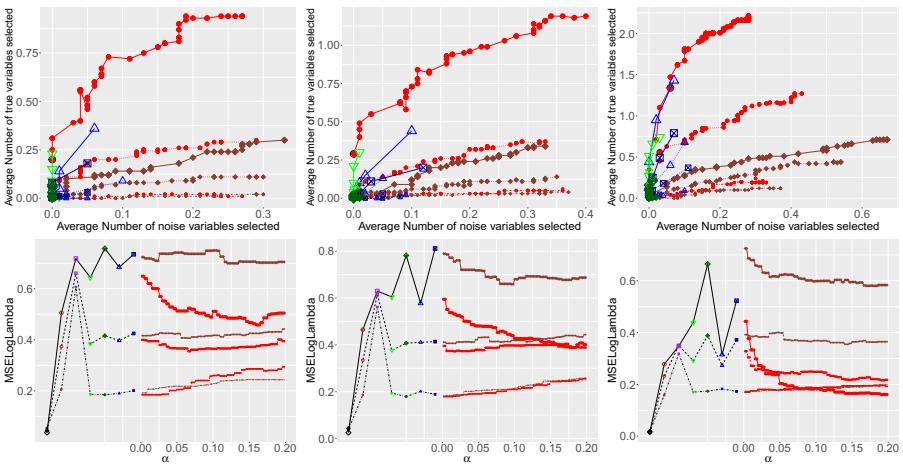
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P34) beta+unif+t_2, MRates=(.8, 1, 2), Variable selection and prediction

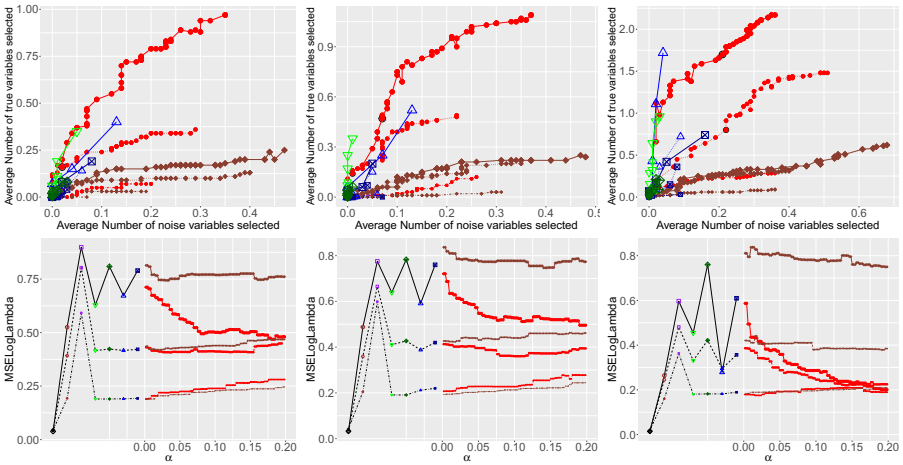


(P35) beta+gam_{12}+t_2, MRates=(.8, 1, 2), Variable selection and prediction

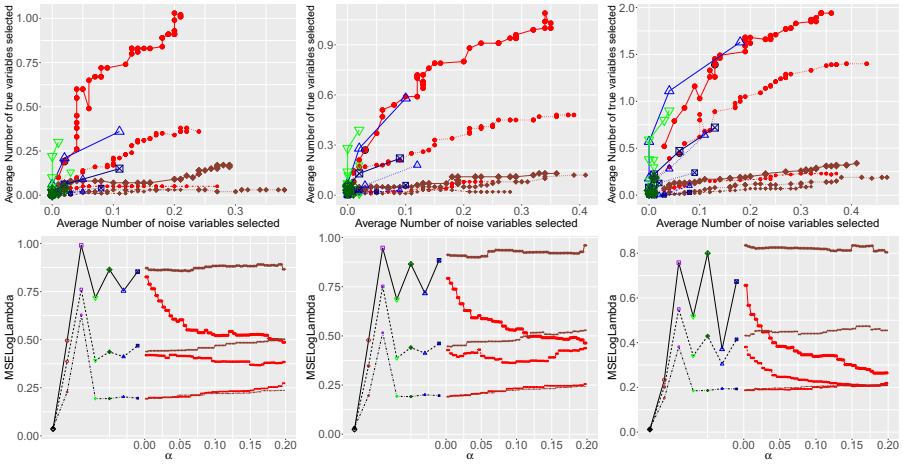


(P36) beta+unif+$pare_2$, MRates=(.8, 1, 2), Variable selection and prediction

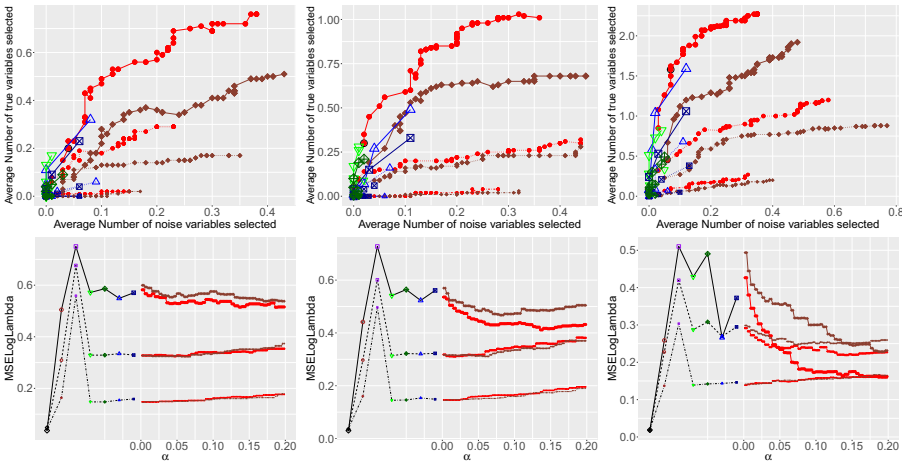
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P37) beta+gam$_{12}$+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

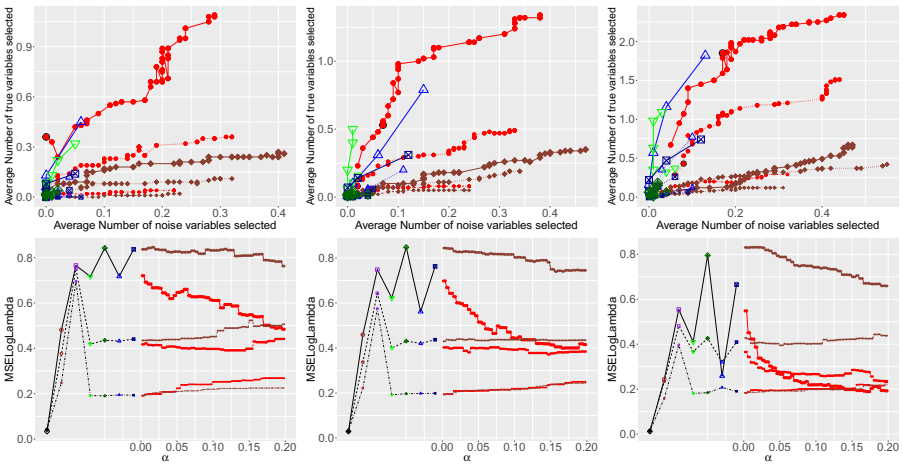


(P38) unif+gam$_{12}$+lnorm, MRates=(.8, 1, 2), Variable selection and prediction

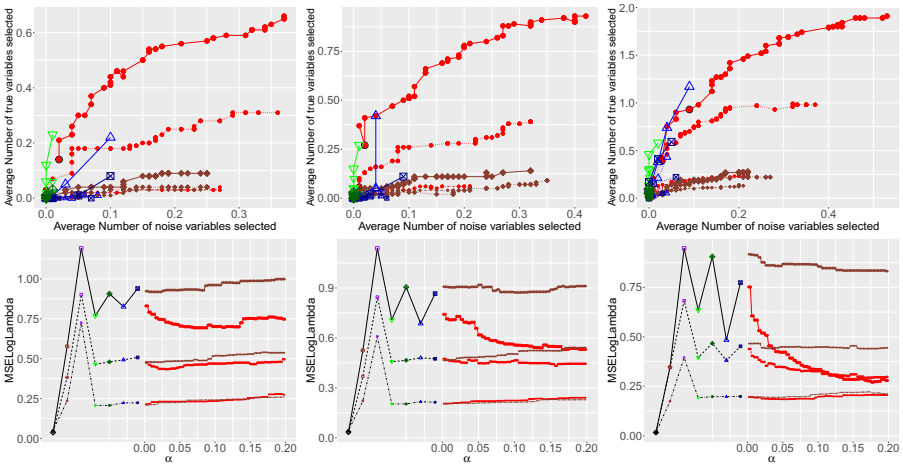


(P39) unif+gam$_{12}$+t$_2$, MRates=(.8, 1, 2), Variable selection and prediction

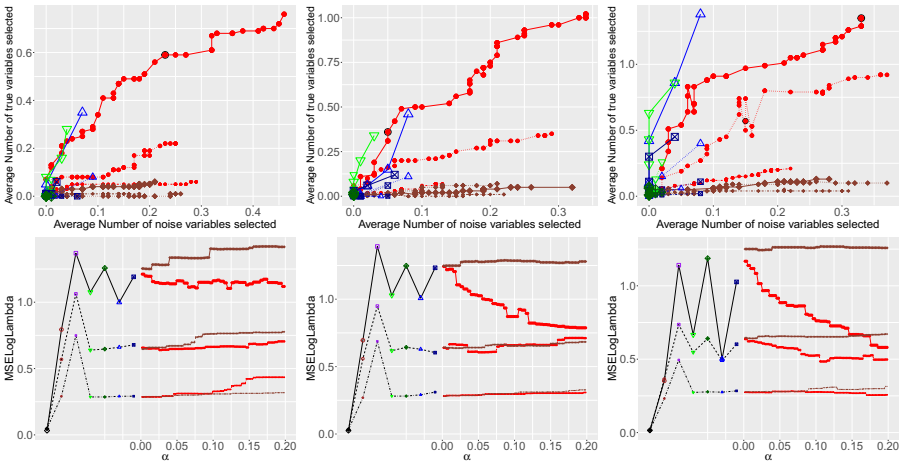
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P40) unif+gam$_{12}$+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

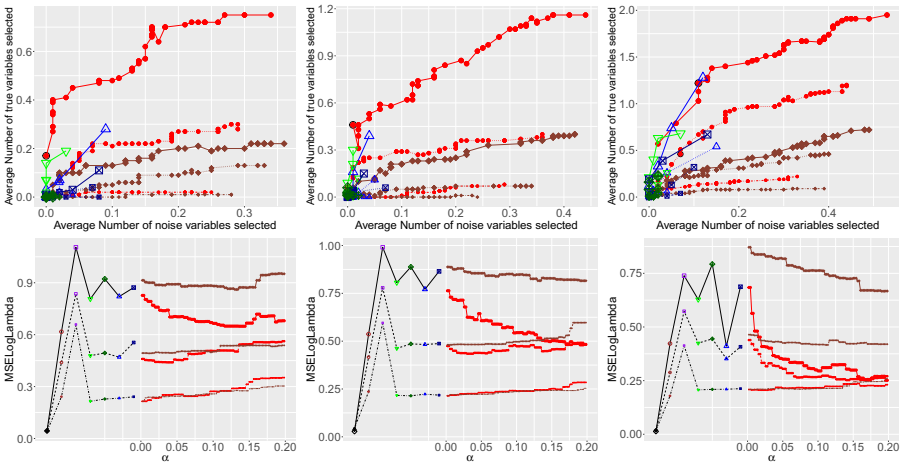


(P41) sdn+lnorm+t$_2$, MRates=(.8, 1, 2), Variable selection and prediction

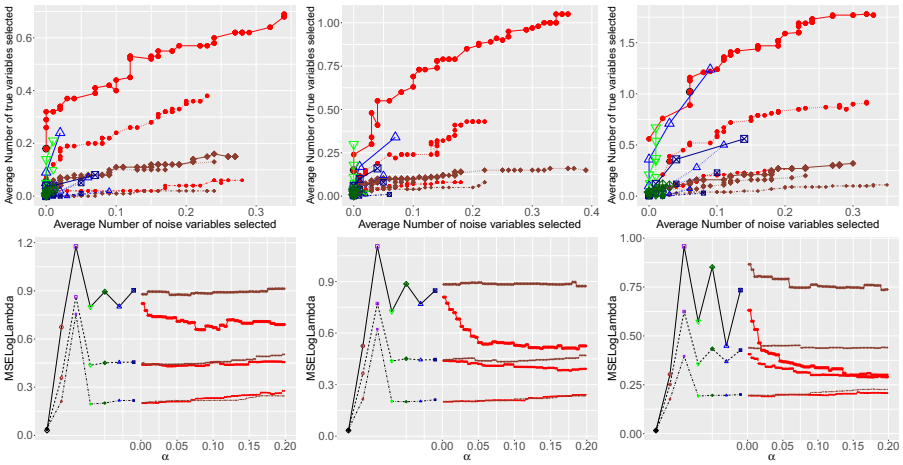


(P42) sdn+lnorm+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

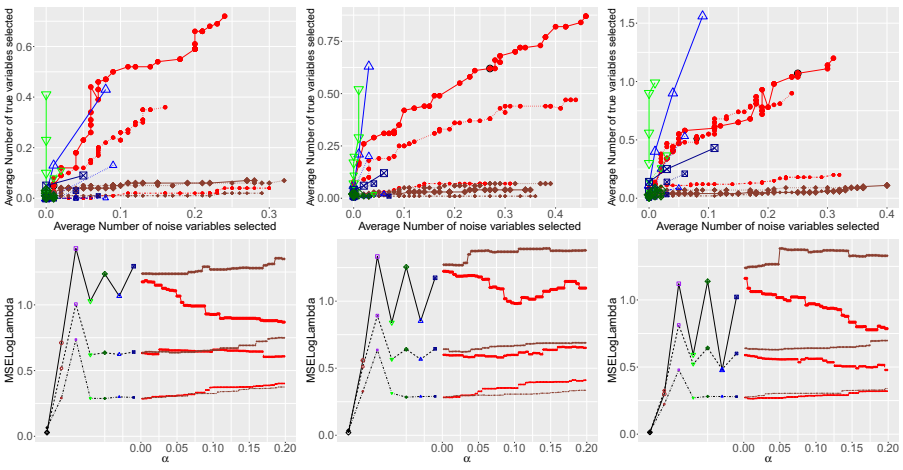
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P43) sdn+t_2+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

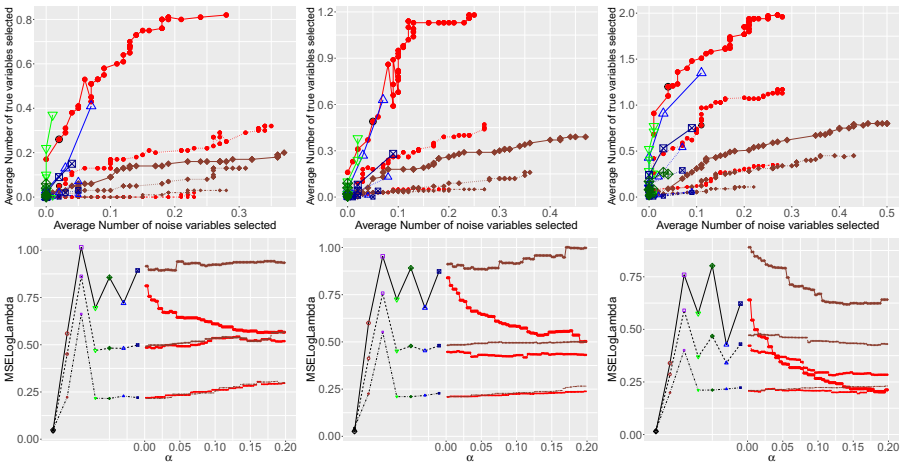


(P44) pois+lnorm+t_2, MRates=(.8, 1, 2), Variable selection and prediction

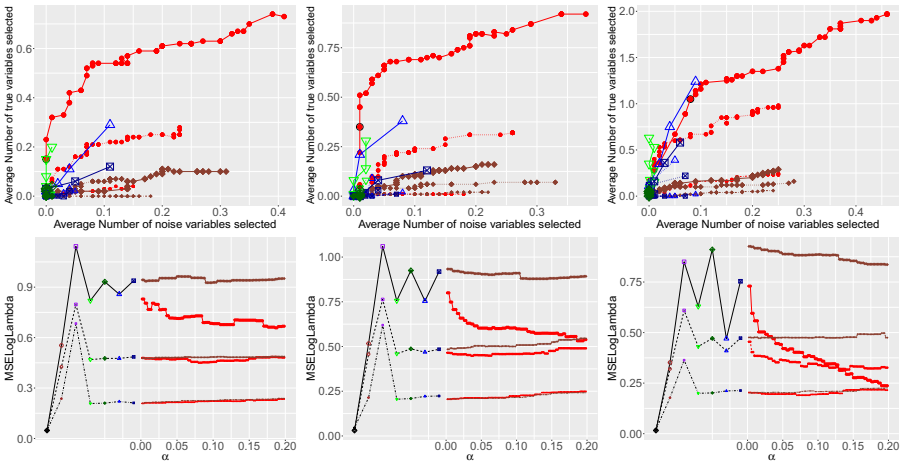


(P45) pois+lnorm+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

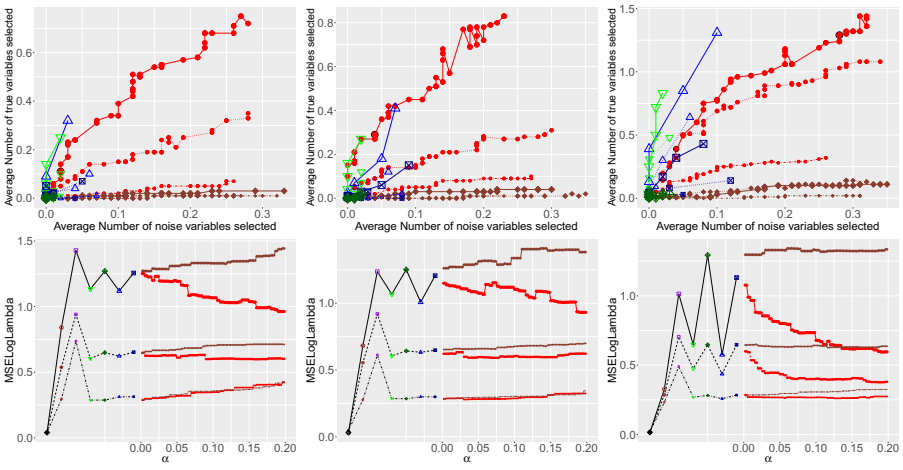
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P46) pois+t_2+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

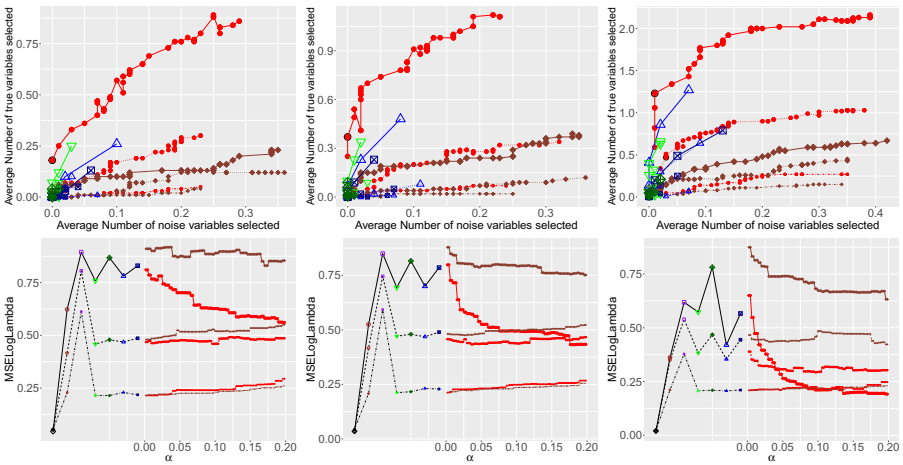


(P47) beta+lnorm+t_2, MRates=(.8, 1, 2), Variable selection and prediction

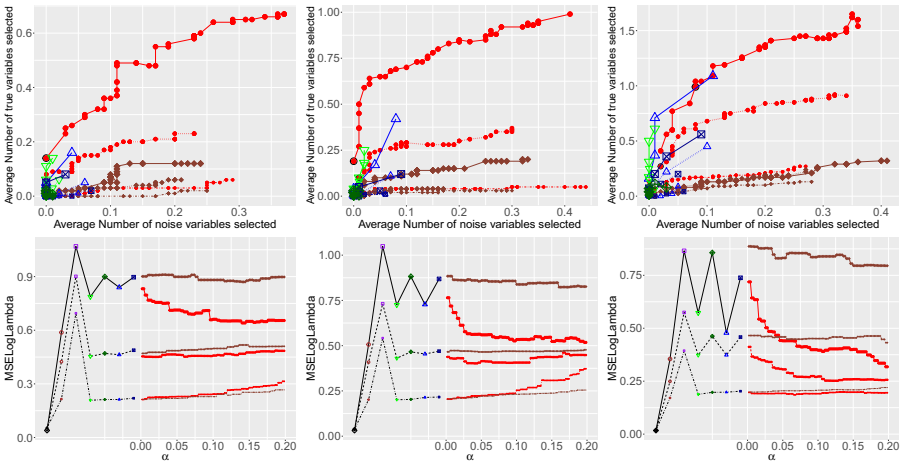


(P48) beta+lnorm+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

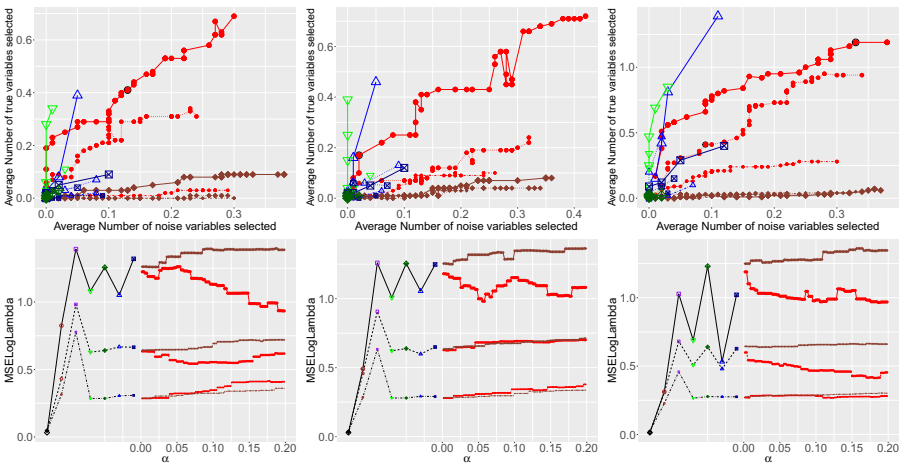
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P49) beta+t_2+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

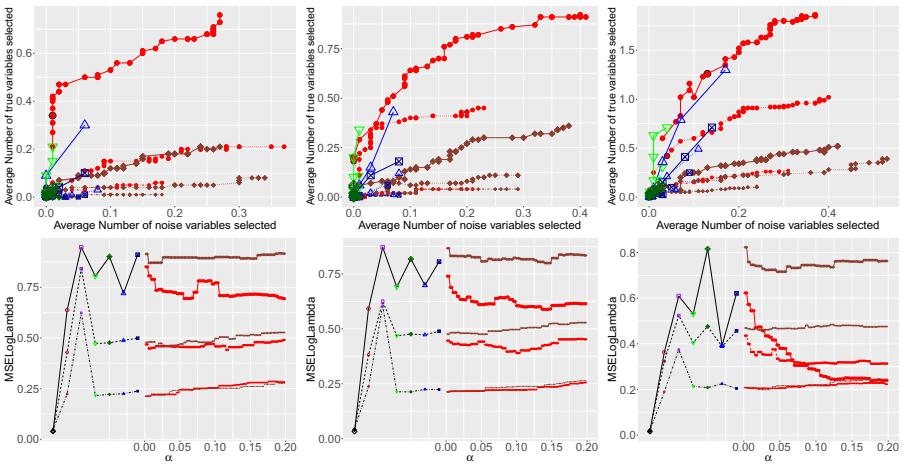


(P50) unif+lnorm+t_2, MRates=(.8, 1, 2), Variable selection and prediction

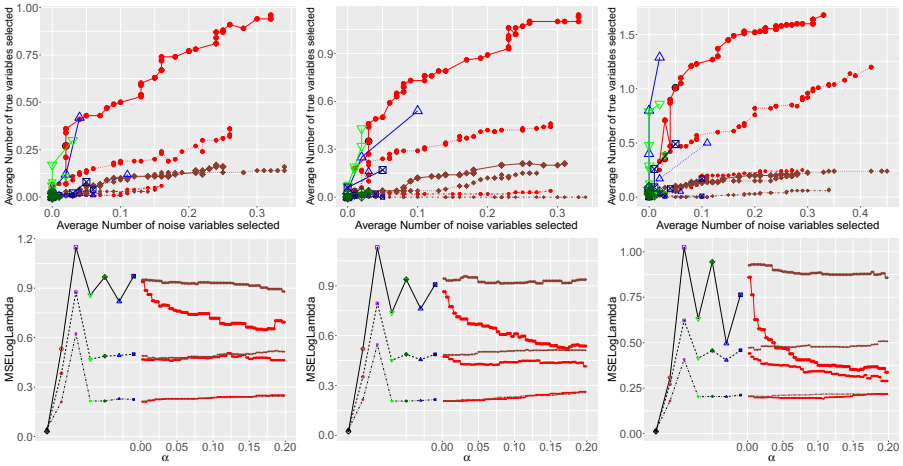


(P51) unif+lnorm+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

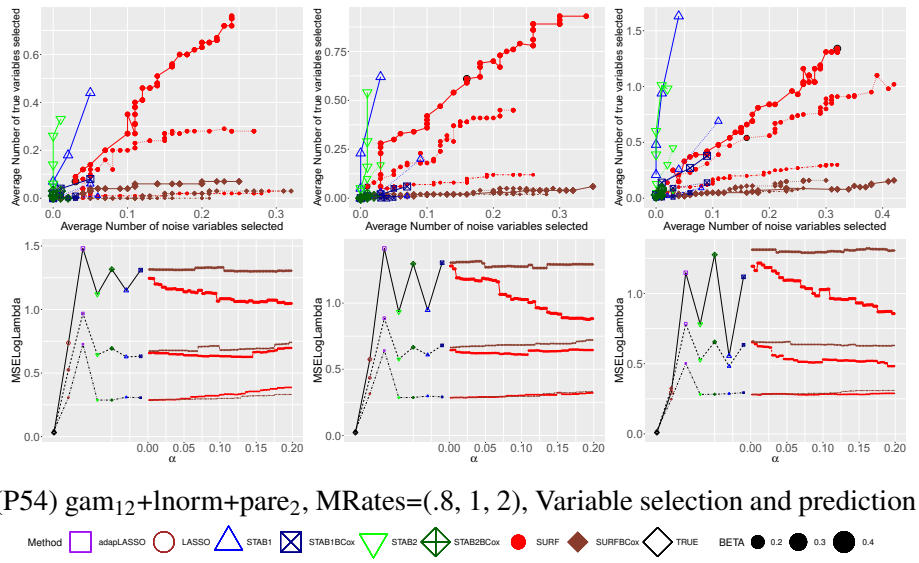
Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P52) unif+t2+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

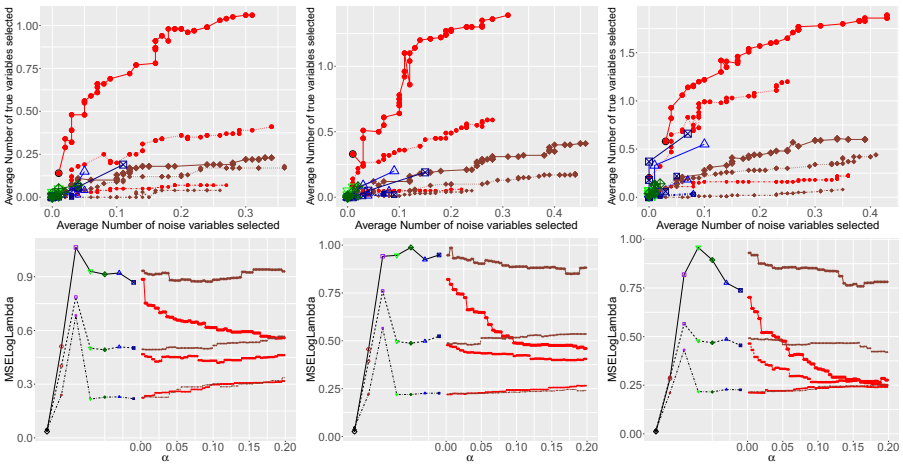


(P53) gam$_{12}$+lnorm+t$_2$, MRates=(.8, 1, 2), Variable selection and prediction

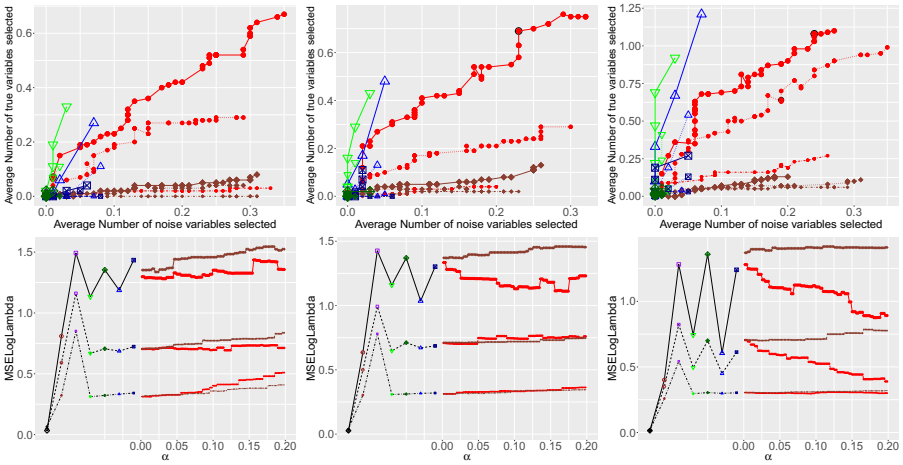


(P54) gam$_{12}$+lnorm+pare$_2$, MRates=(.8, 1, 2), Variable selection and prediction

Figure A.6: Variable Selection and Prediction for multiple true variable case in Poisson Model



(P55) gam_{12}+t_2+$pare_2$, MRates=(.8, 1, 2), Variable selection and prediction

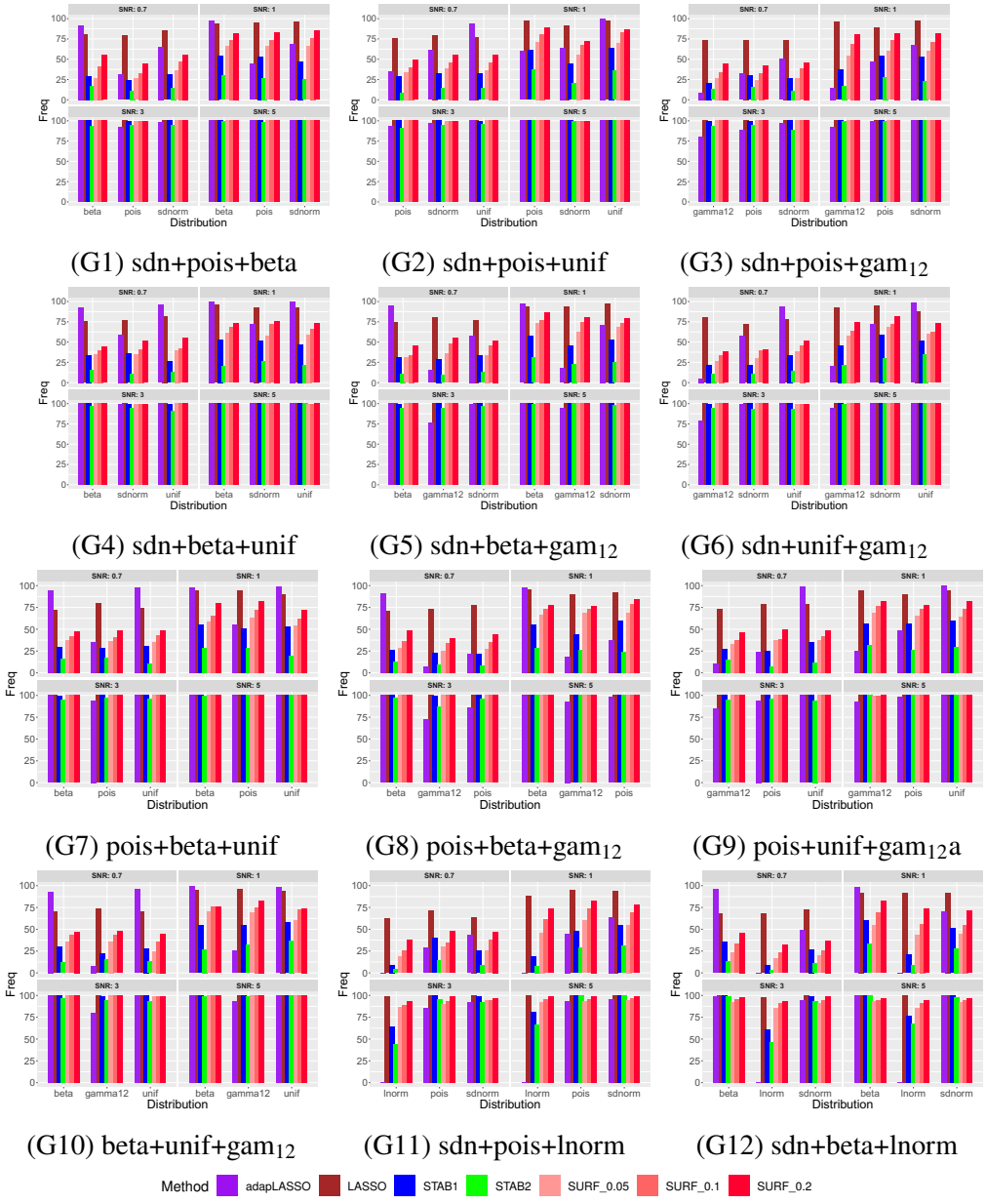


(P56) lnorm+t_2+$pare_2$, MRates=(.8, 1, 2), Variable selection and prediction

Figure A.6: Variable selection and prediction for multiple true variable Poisson regression

A.2 Appendix2: Complete Figures for True positives

A.2.1 Gaussian multiple true variable cases



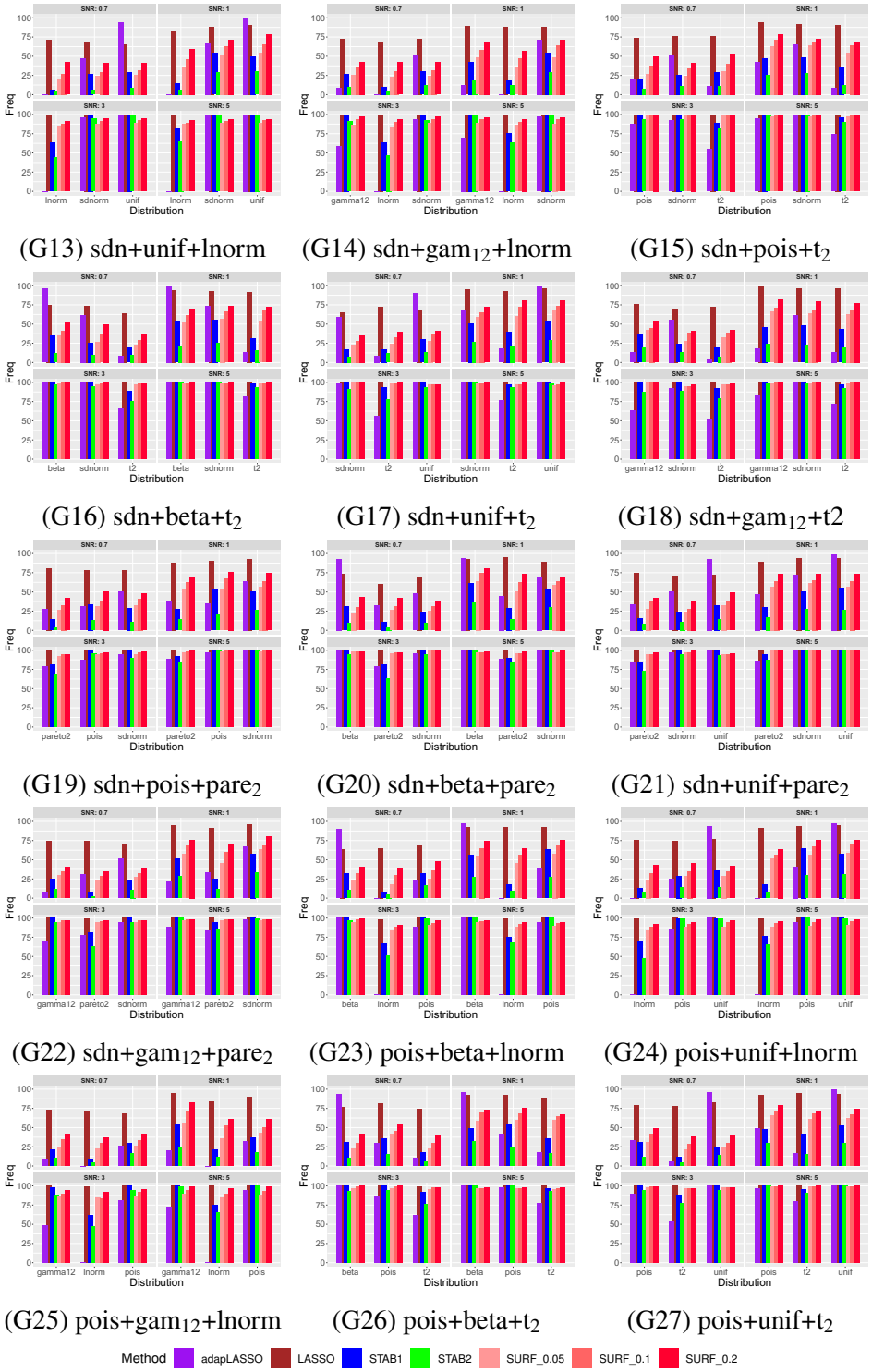
(G1) sdn+pois+beta (G2) sdn+pois+unif (G3) sdn+pois+gam$_{12}$

(G4) sdn+beta+unif (G5) sdn+beta+gam$_{12}$ (G6) sdn+unif+gam$_{12}$

(G7) pois+beta+unif (G8) pois+beta+gam$_{12}$ (G9) pois+unif+gam$_{12}$a

(G10) beta+unif+gam$_{12}$ (G11) sdn+pois+lnorm (G12) sdn+beta+lnorm

Figure A.7: Selection frequency of different distributions for Gaussian regression



(G13) sdn+unif+lnorm (G14) sdn+gam$_{12}$+lnorm (G15) sdn+pois+t$_2$

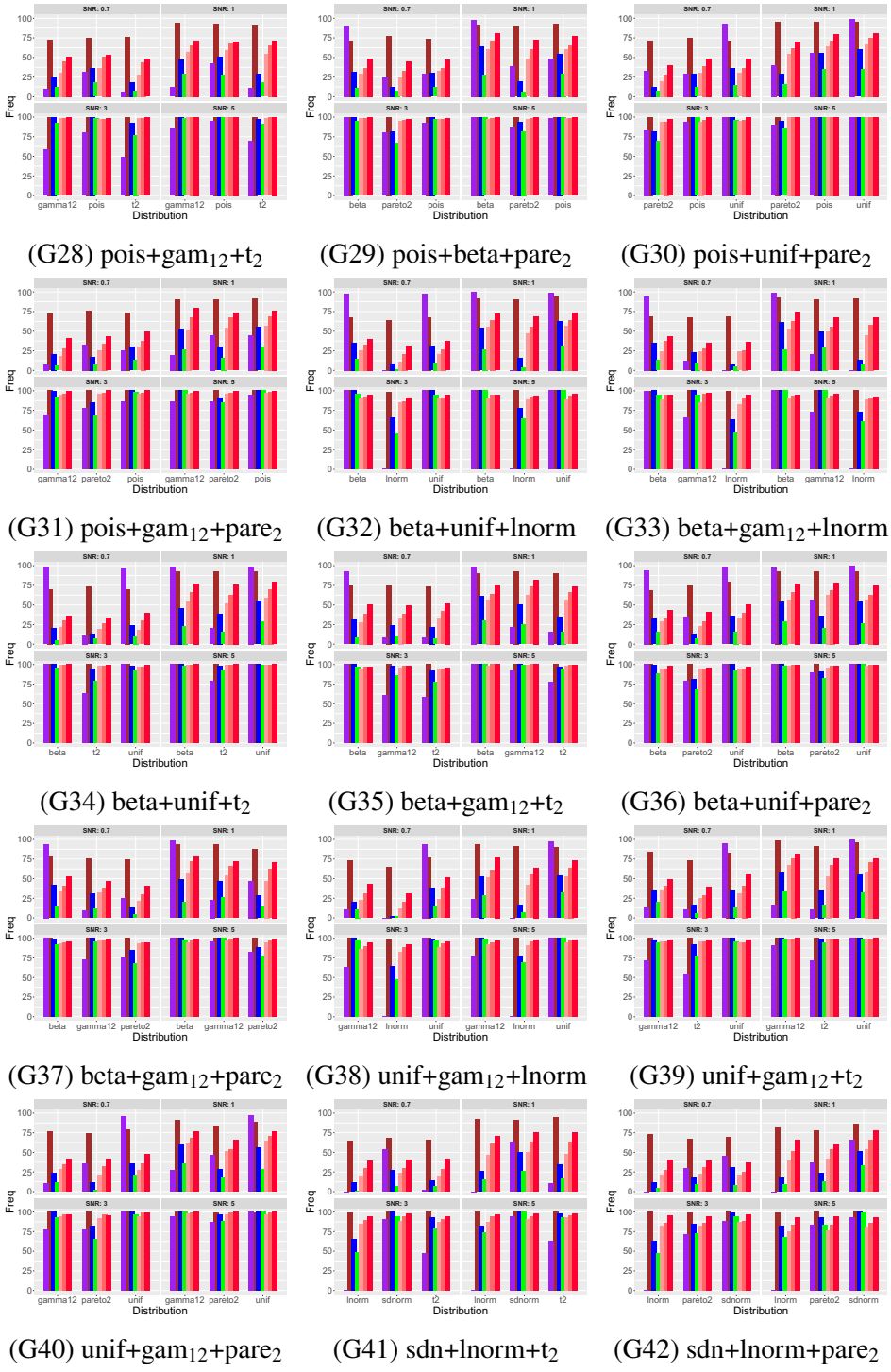
(G16) sdn+beta+t$_2$ (G17) sdn+unif+t$_2$ (G18) sdn+gam$_{12}$+t2

(G19) sdn+pois+pare$_2$ (G20) sdn+beta+pare$_2$ (G21) sdn+unif+pare$_2$

(G22) sdn+gam$_{12}$+pare$_2$ (G23) pois+beta+lnorm (G24) pois+unif+lnorm

(G25) pois+gam$_{12}$+lnorm (G26) pois+beta+t$_2$ (G27) pois+unif+t$_2$

Figure A.7: Selection frequency of different distributions for Gaussian Model



(G28) pois+gam$_{12}$+t$_2$ (G29) pois+beta+pare$_2$ (G30) pois+unif+pare$_2$

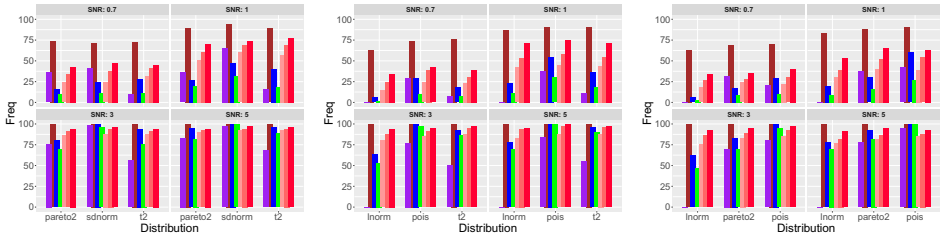
(G31) pois+gam$_{12}$+pare$_2$ (G32) beta+unif+lnorm (G33) beta+gam$_{12}$+lnorm

(G34) beta+unif+t$_2$ (G35) beta+gam$_{12}$+t$_2$ (G36) beta+unif+pare$_2$

(G37) beta+gam$_{12}$+pare$_2$ (G38) unif+gam$_{12}$+lnorm (G39) unif+gam$_{12}$+t$_2$

(G40) unif+gam$_{12}$+pare$_2$ (G41) sdn+lnorm+t$_2$ (G42) sdn+lnorm+pare$_2$

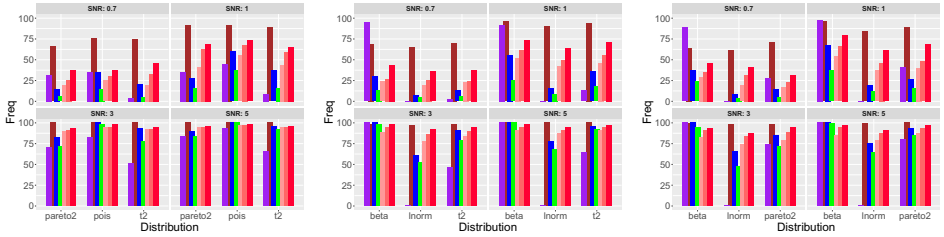
Figure A.7: Selection frequency of different distributions for Gaussian Model



(G43) sdn+t_2+pare$_2$

(G44) pois+lnorm+t_2

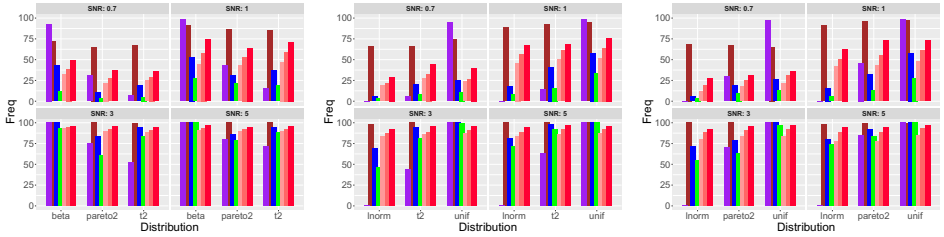
(G45) pois+lnorm+pare$_2$



(G46) pois+t_2+pare$_2$

(G47) beta+lnorm+t_2

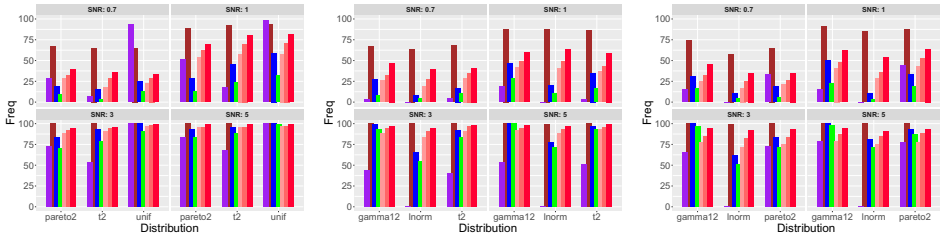
(G48) beta+lnorm+pare$_2$



(G49) beta+t_2+pare$_2$

(G50) unif+lnorm+t_2

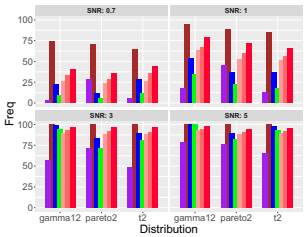
(G51) unif+lnorm+pare$_2$



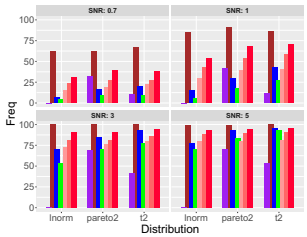
(G52) unif+t2+pare$_2$

(G53) gam$_{12}$+lnorm+t_2

(G54)gam$_{12}$+lnorm+pare$_2$



(G55) gam$_{12}$+t_2+pare$_2$



(G56) lnorm+t_2+pare$_2$

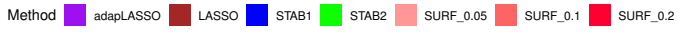


Figure A.7: Selection frequency of different distributions for Gaussian regression

A.2.2 Binomial multiple true variable cases

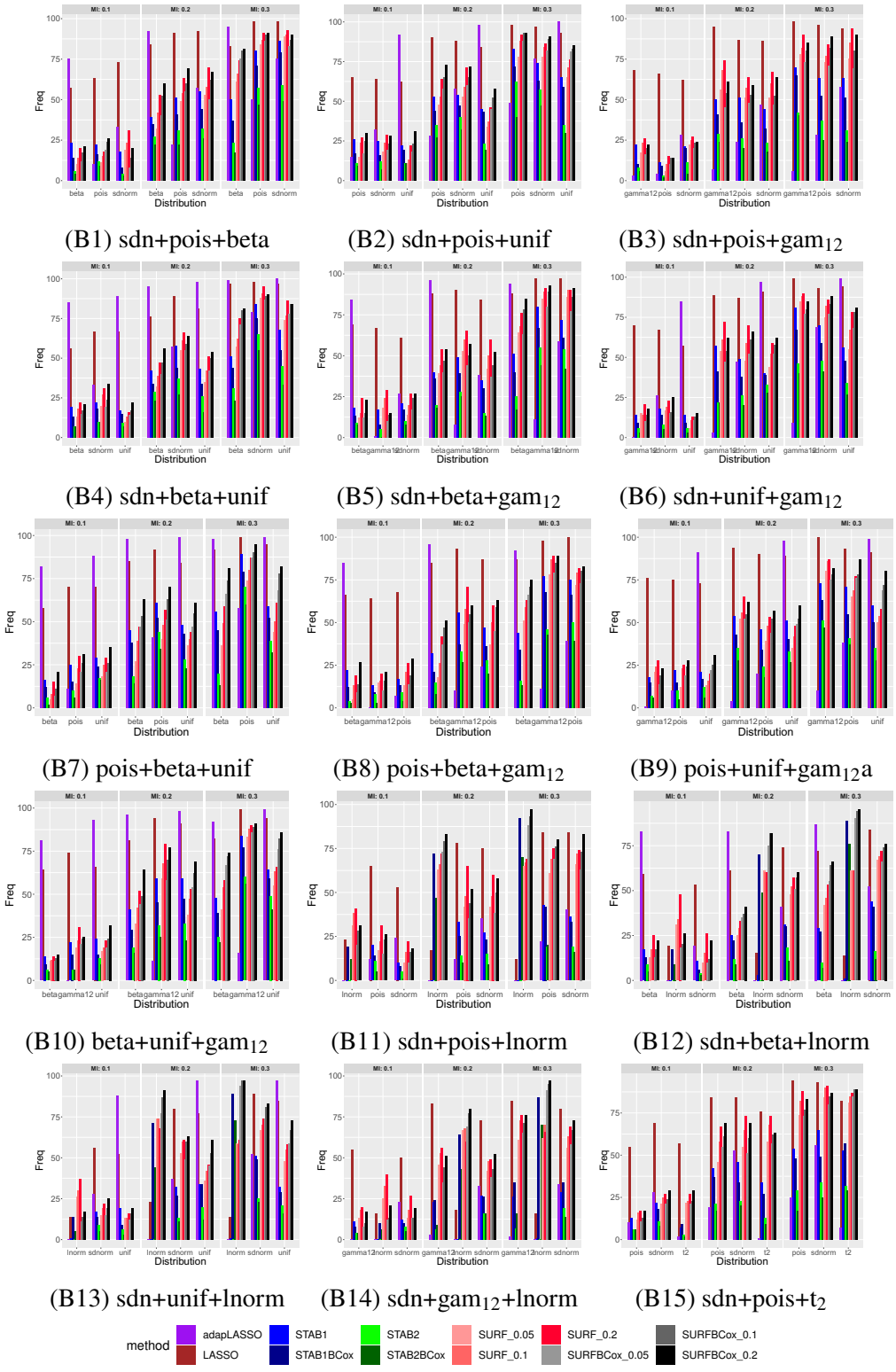


Figure A.8: Selection frequency of different distributions for Binomial Model

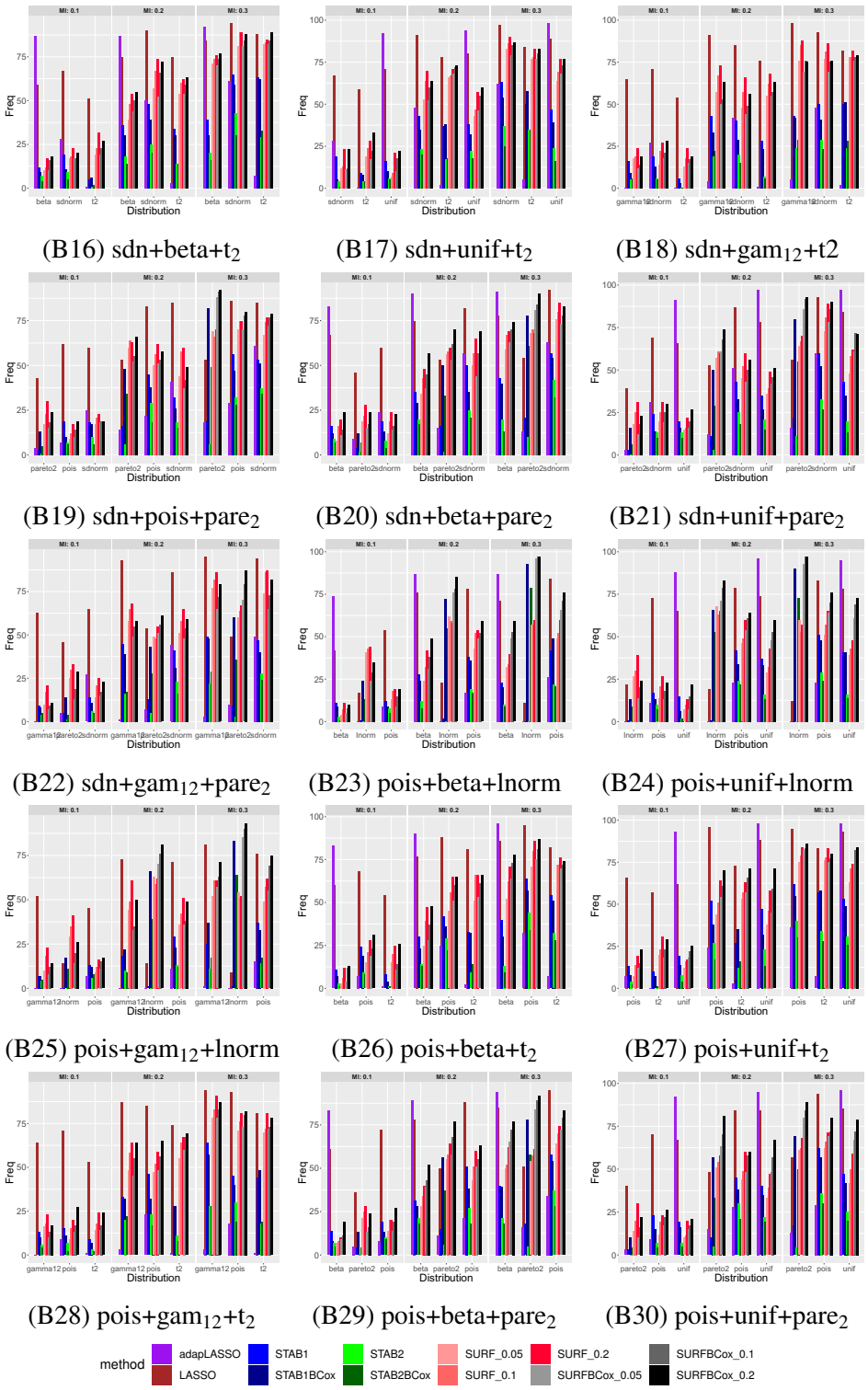


Figure A.8: Selection frequency of different distributions for Binomial Model

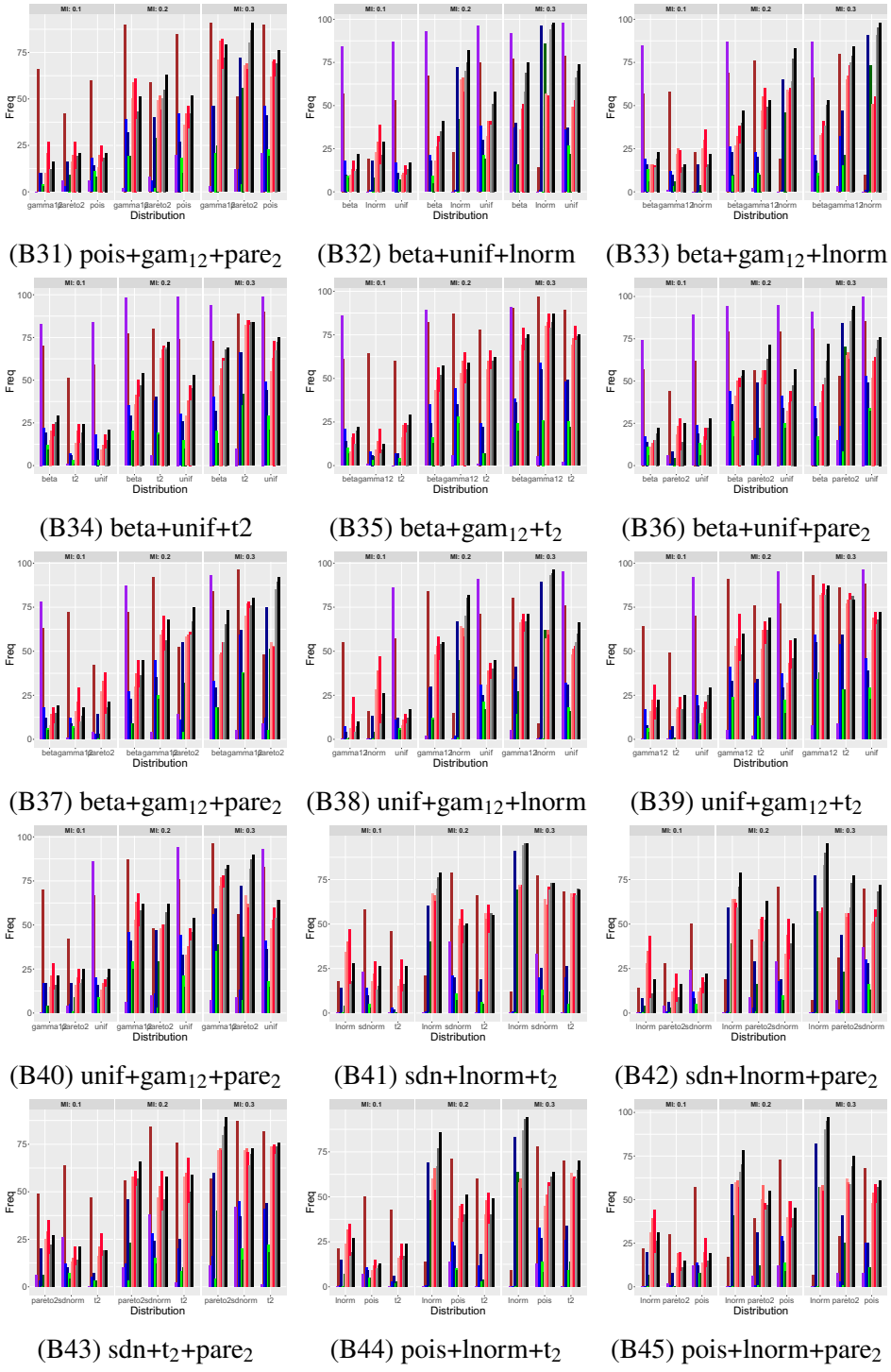
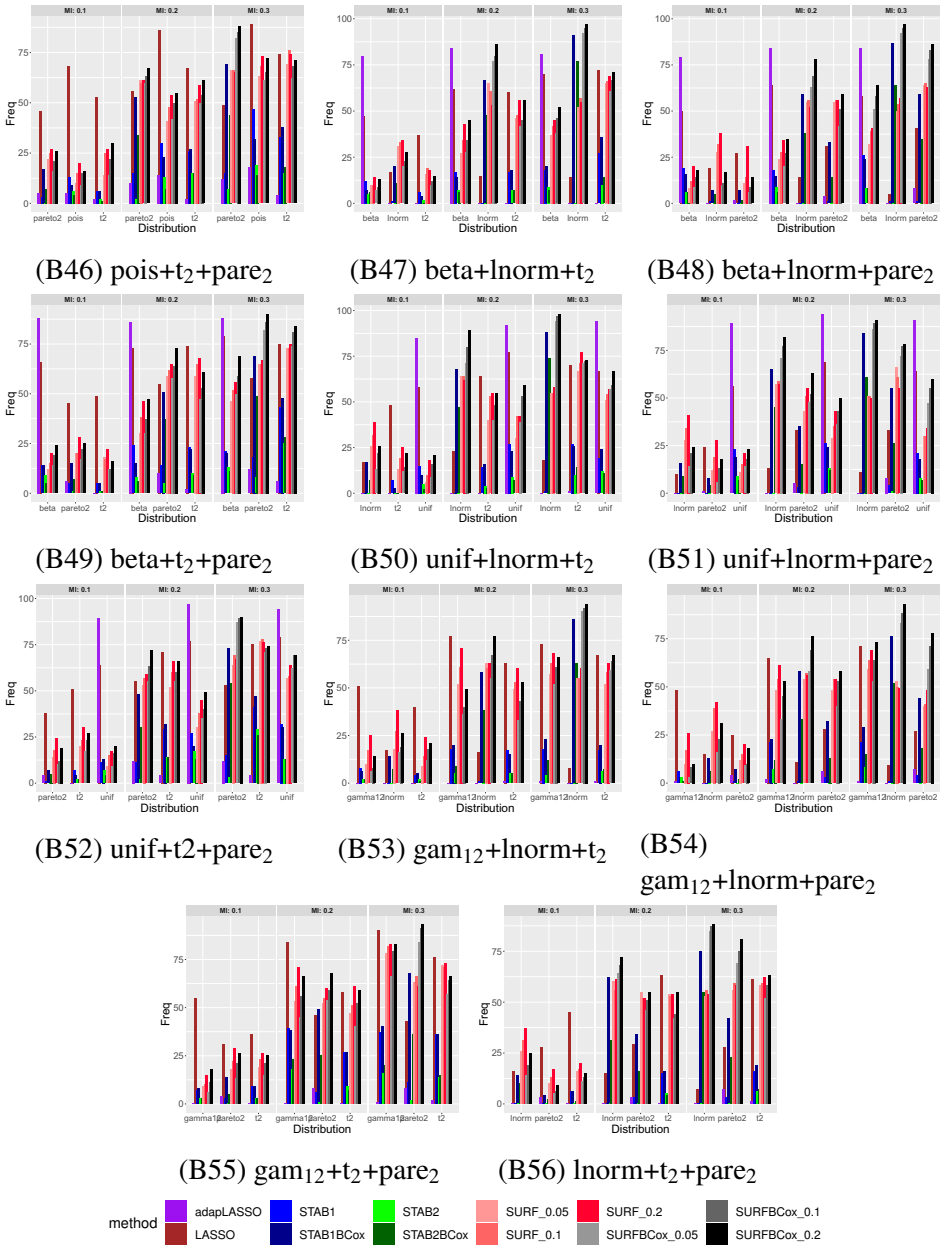


Figure A.8: Selection frequency of different distributions for Binomial Model



(B46) pois+t_2+pare$_2$ (B47) beta+lnorm+t_2 (B48) beta+lnorm+pare$_2$

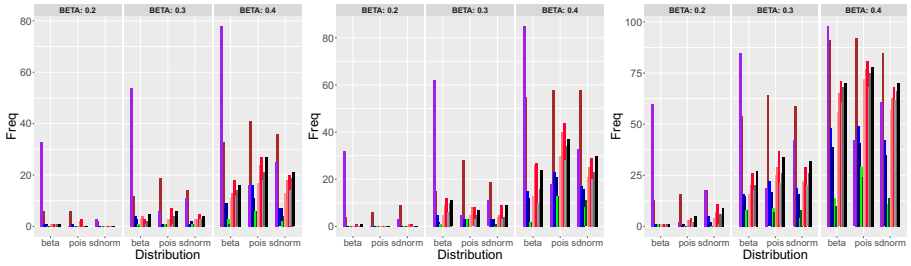
(B49) beta+t_2+pare$_2$ (B50) unif+lnorm+t_2 (B51) unif+lnorm+pare$_2$

(B52) unif+t2+pare$_2$ (B53) gam$_{12}$+lnorm+t_2 (B54) gam$_{12}$+lnorm+pare$_2$

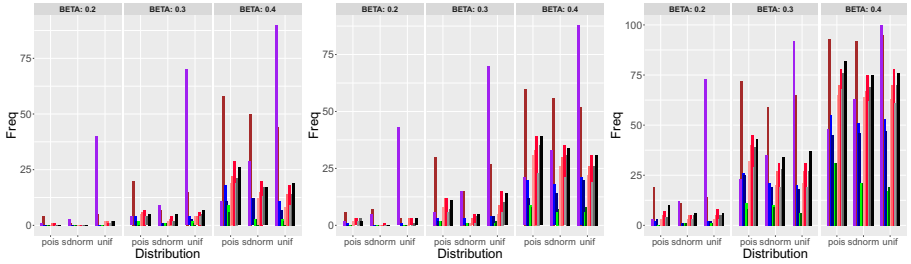
(B55) gam$_{12}$+t_2+pare$_2$ (B56) lnorm+t_2+pare$_2$

Figure A.8: Selection frequency of different distributions for Binomial Model

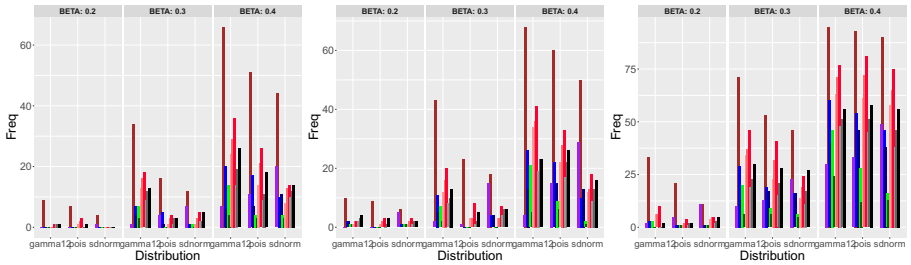
A.2.3 Poisson multiple true variable cases



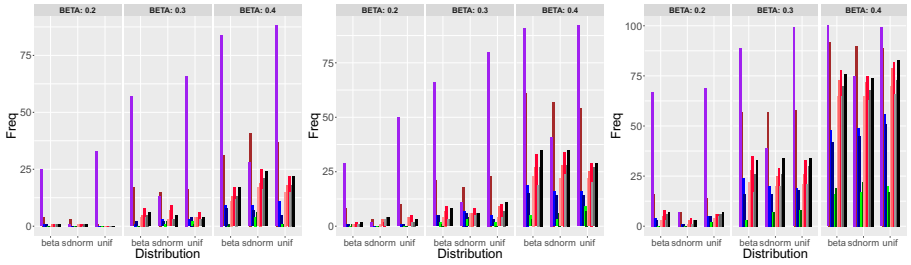
(P1) sdn+pois+beta, MRates=(.8, 1, 2)



(P2) sdn+pois+unif, MRates=(.8, 1, 2)



(P3) sdn+pois+gam$_{12}$, MRates=(.8, 1, 2)



(P4) sdn+beta+unif, MRates=(.8, 1, 2)

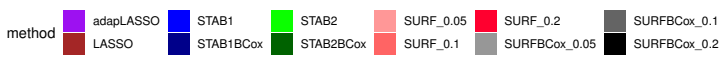
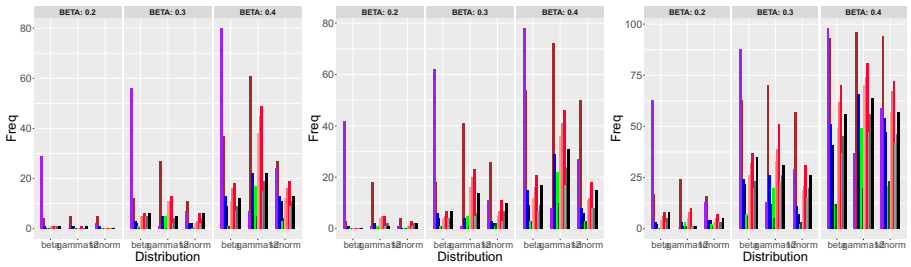
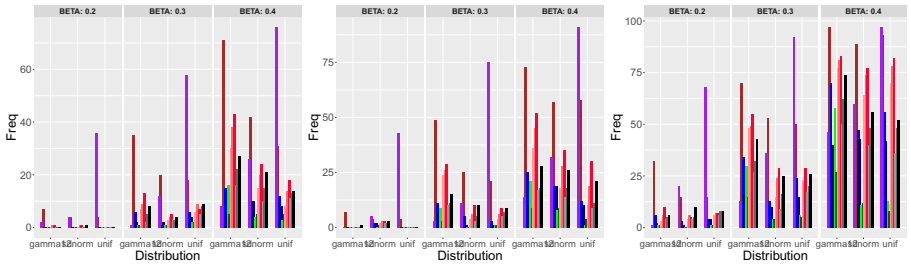


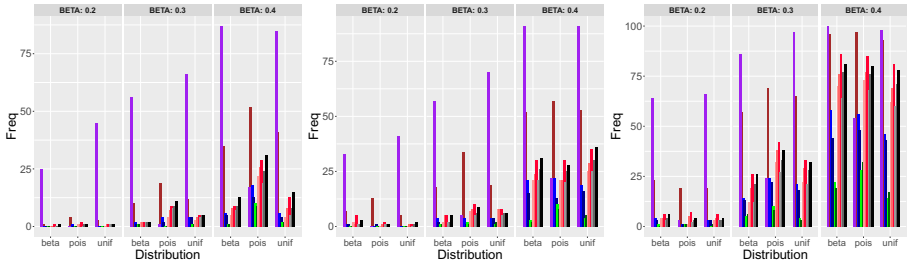
Figure A.9: Selection frequency of different distributions for Poisson regression



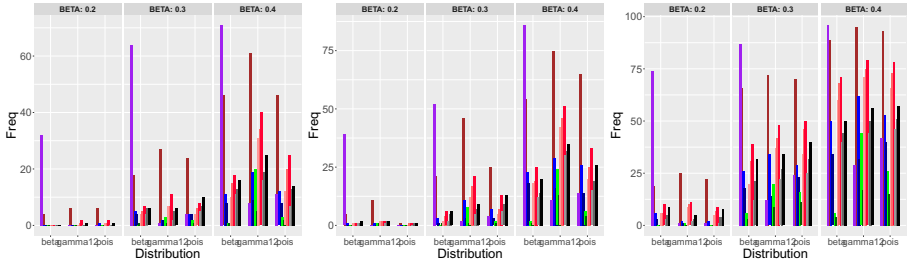
(P5) sdn+beta+gam$_{12}$, MRates=(.8, 1, 2)



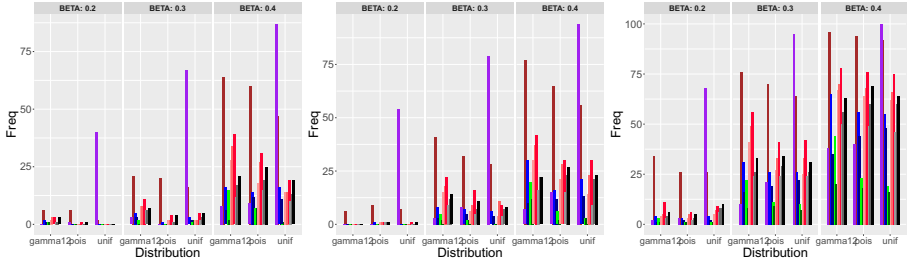
(P6) sdn+unif+gam$_{12}$, MRates=(.8, 1, 2)



(P7) pois+beta+unif, MRates=(.8, 1, 2)



(P8) pois+beta+gam$_{12}$, MRates=(.8, 1, 2)



(P9) pois+unif+gam$_{12}$, MRates=(.8, 1, 2)

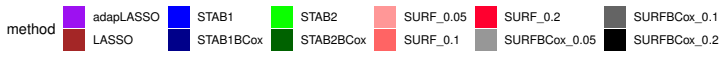
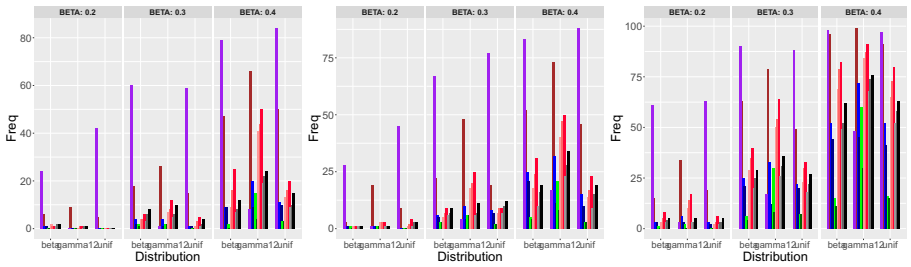
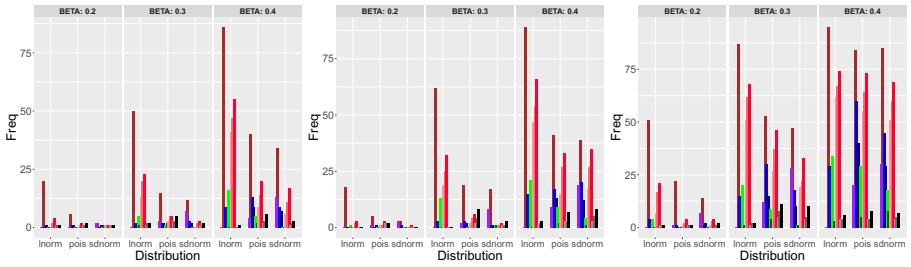


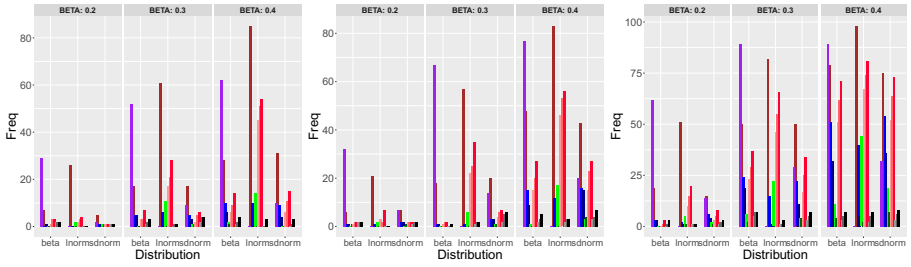
Figure A.9: Selection frequency of different distributions for Poisson Model



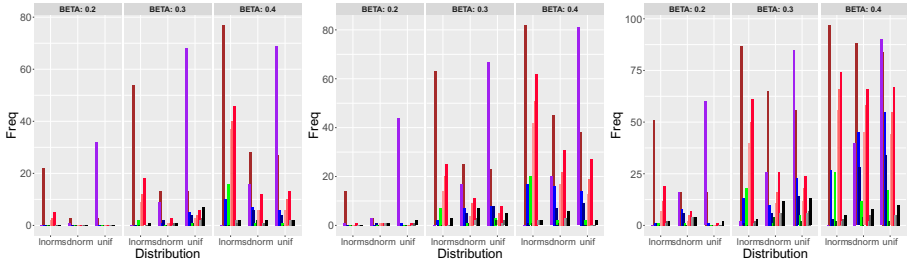
(P10) beta+unif+gam$_{12}$, MRates=(.8, 1, 2)



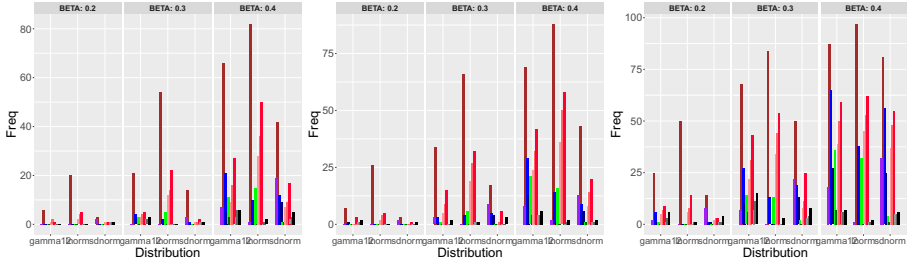
(P11) sdn+pois+lnorm, MRates=(.8, 1, 2)



(P12) sdn+beta+lnorm, MRates=(.8, 1, 2)



(P13) sdn+unif+lnorm, MRates=(.8, 1, 2)



(P14) sdn+gam$_{12}$+lnorm, MRates=(.8, 1, 2)

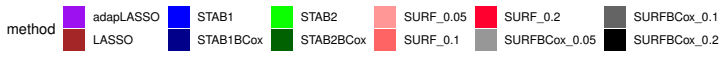
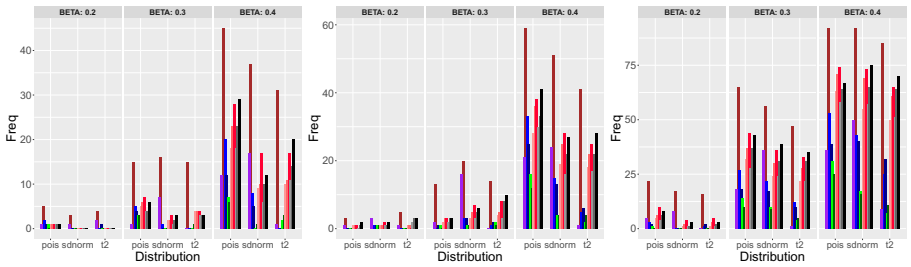
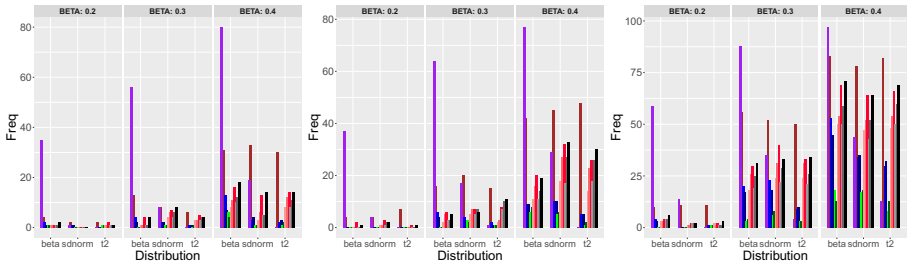


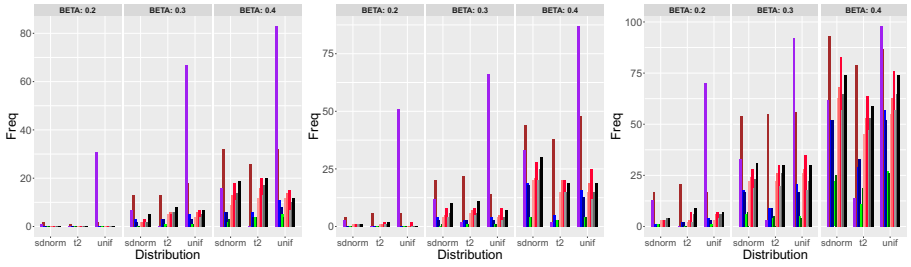
Figure A.9: Selection frequency of different distributions for Poisson Model



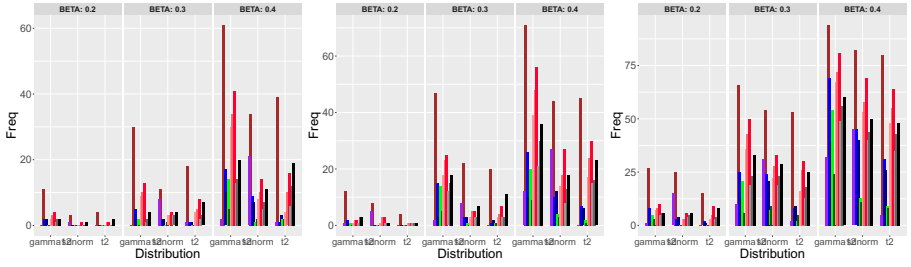
(P15) sdn+pois+t_2, MRates=(.8, 1, 2)



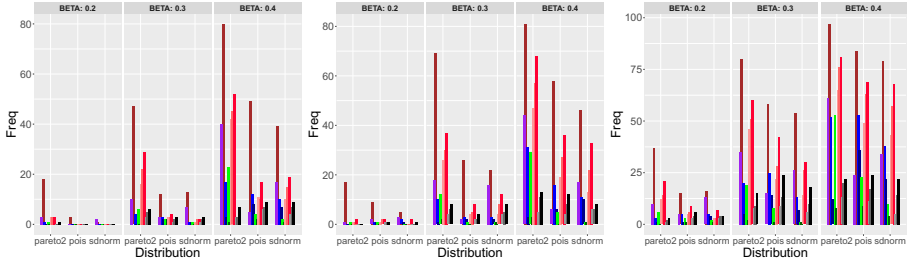
(P16) sdn+beta+t_2, MRates=(.8, 1, 2)



(P17) sdn+unif+t_2, MRates=(.8, 1, 2)



(P18) sdn+gam$_{12}$+t2, MRates=(.8, 1, 2)



(P19) sdn+pois+pare$_2$, MRates=(.8, 1, 2)

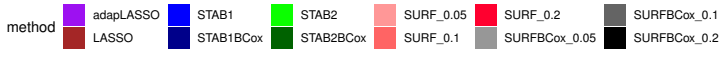
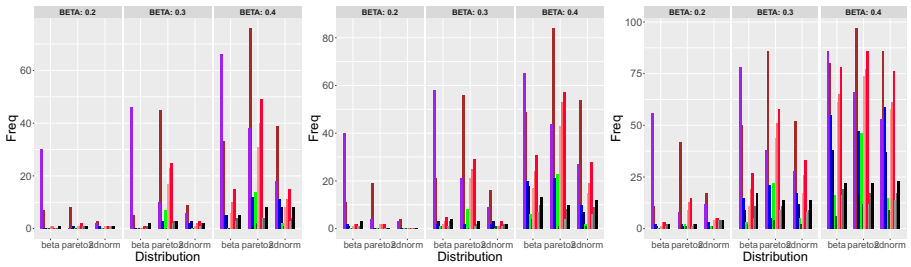
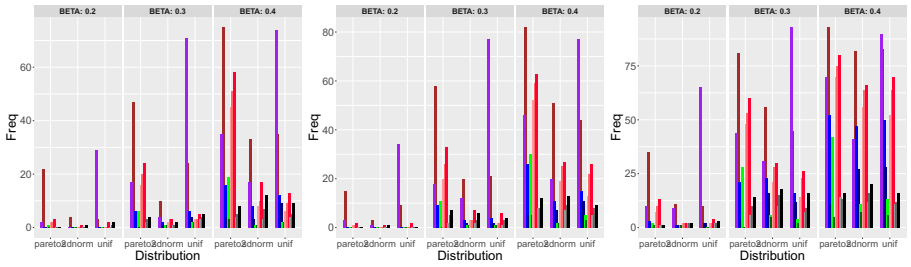


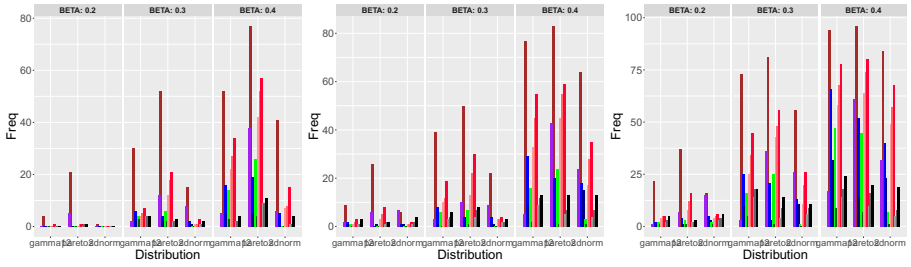
Figure A.9: Selection frequency of different distributions for Poisson Model



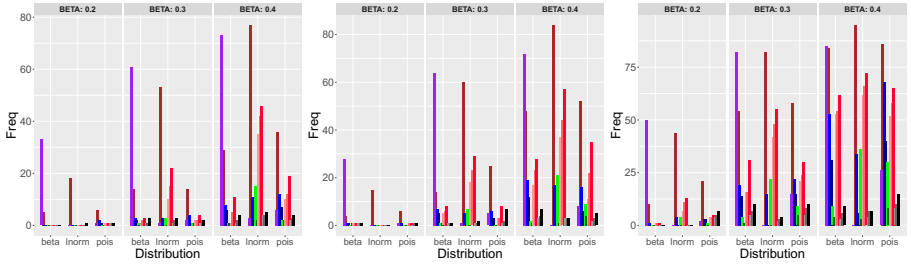
(P20) sdn+beta+pare$_2$, MRates=(.8, 1, 2)



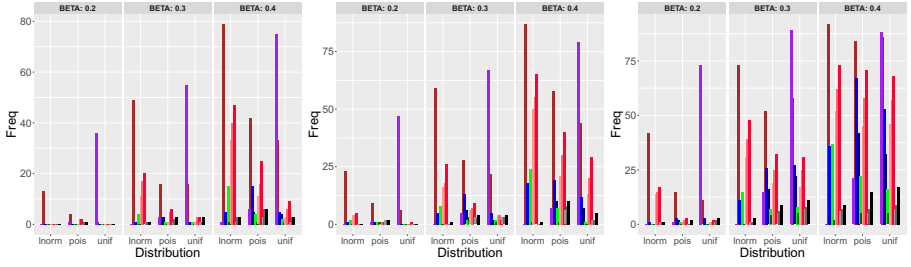
(P21) sdn+unif+pare$_2$, MRates=(.8, 1, 2)



(P22) sdn+gam$_{12}$+pare$_2$, MRates=(.8, 1, 2)



(P23) pois+beta+lnorm, MRates=(.8, 1, 2)



(P24) pois+unif+lnorm, MRates=(.8, 1, 2)

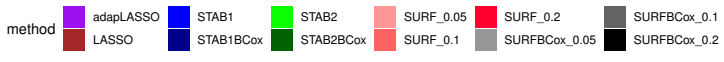
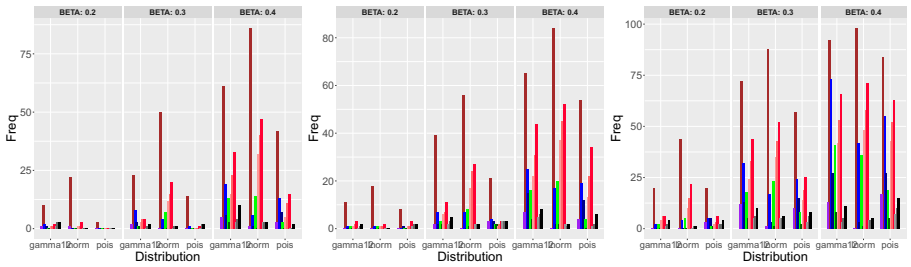
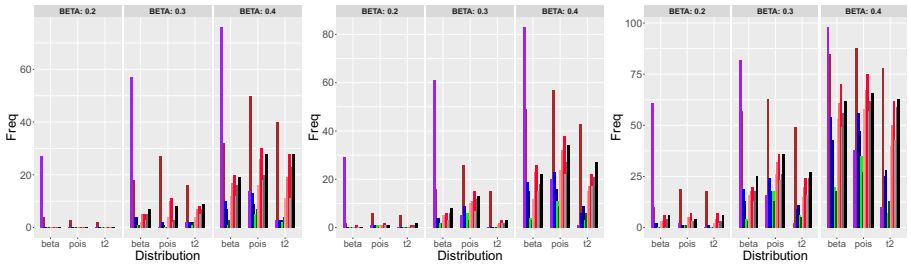


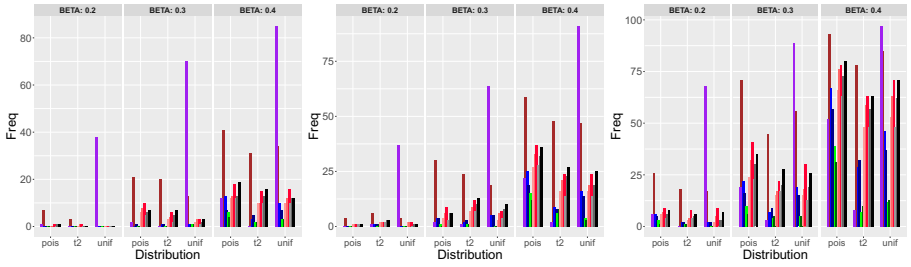
Figure A.9: Selection frequency of different distributions for Poisson Model



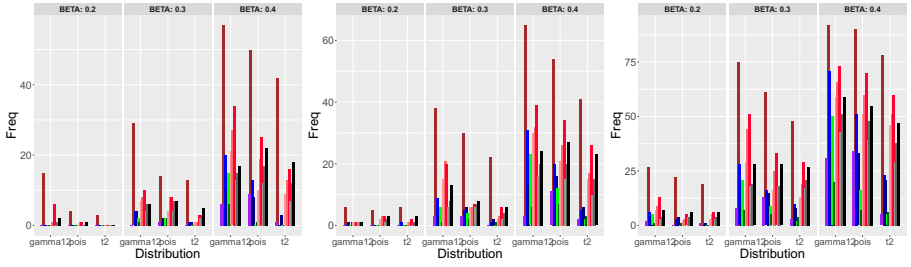
(P25) pois+gam$_{12}$+lnorm, MRates=(.8, 1, 2)



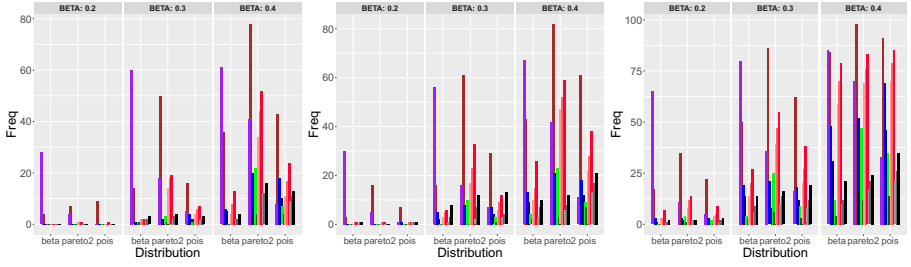
(P26) pois+beta+t$_2$, MRates=(.8, 1, 2)



(P27) pois+unif+t$_2$, MRates=(.8, 1, 2)



(P28) pois+gam$_{12}$+t$_2$, MRates=(.8, 1, 2)



(P29) pois+beta+pare$_2$, MRates=(.8, 1, 2)

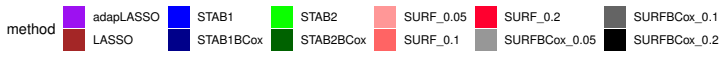
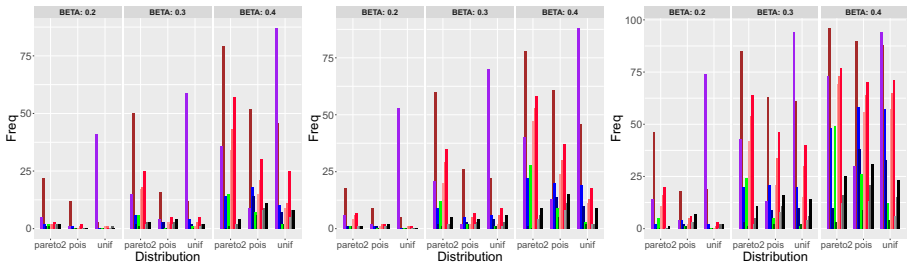


Figure A.9: Selection frequency of different distributions for Poisson Model



(P30) pois+unif+pare$_2$, MRates=(.8, 1, 2)

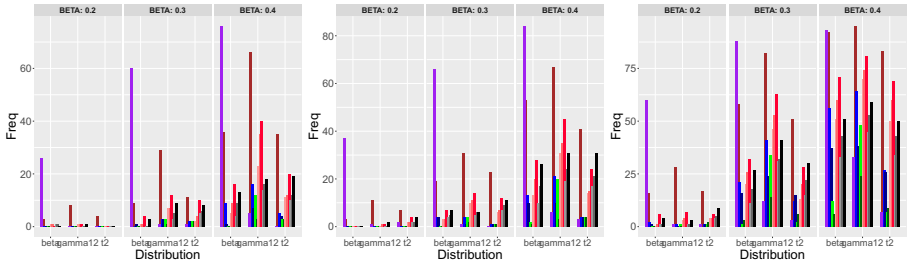
(P31) pois+gam$_{12}$+pare$_2$, MRates=(.8, 1, 2)

(P32) beta+unif+lnorm, MRates=(.8, 1, 2)

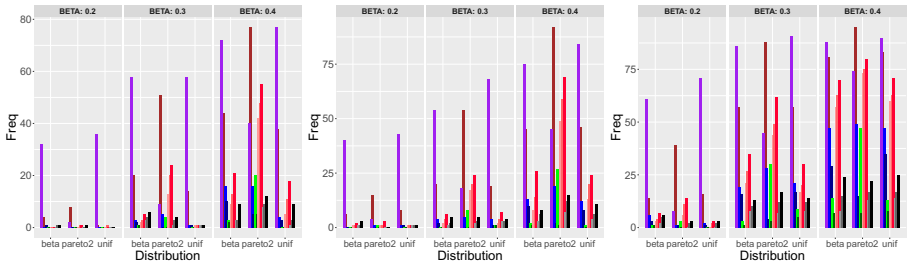
(P33) beta+gam$_{12}$+lnorm, MRates=(.8, 1, 2)

(P34) beta+unif+t$_2$, MRates=(.8, 1, 2)

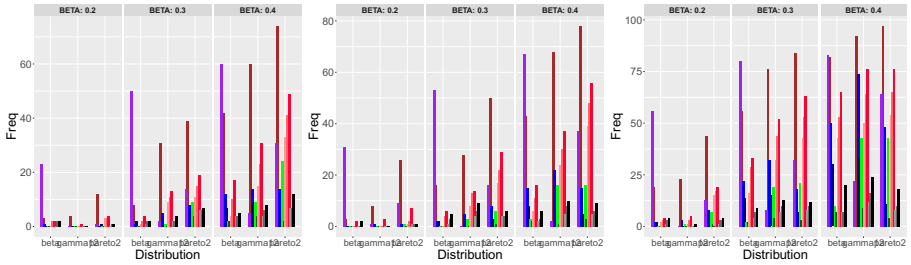
Figure A.9: Selection frequency of different distributions for Poisson Model



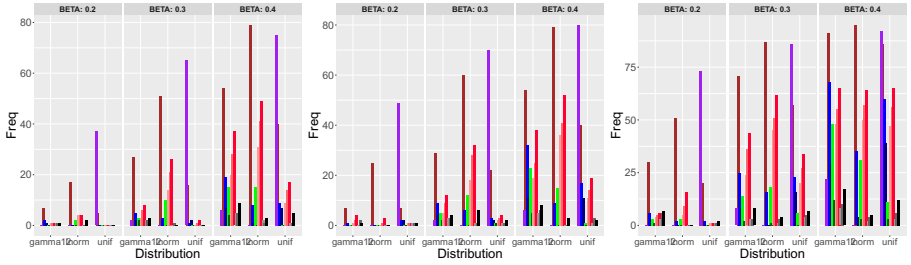
(P35) beta+gam$_{12}$+t$_2$, MRates=(.8, 1, 2)



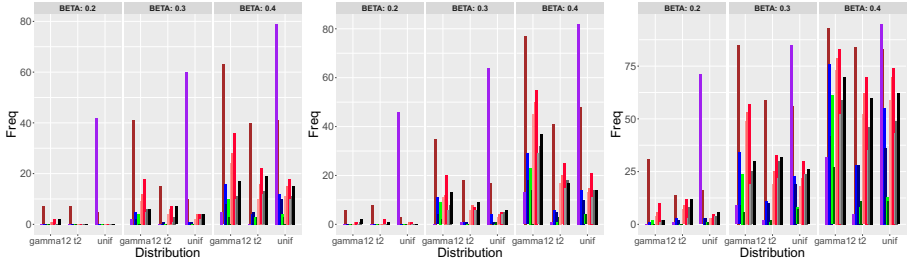
(P36) beta+unif+pare$_2$, MRates=(.8, 1, 2)



(P37) beta+gam$_{12}$+pare$_2$, MRates=(.8, 1, 2)



(P38) unif+gam$_{12}$+lnorm, MRates=(.8, 1, 2)



(P39) unif+gam$_{12}$+t$_2$, MRates=(.8, 1, 2)

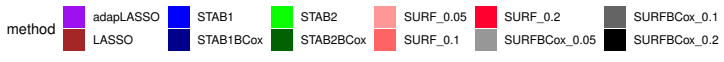
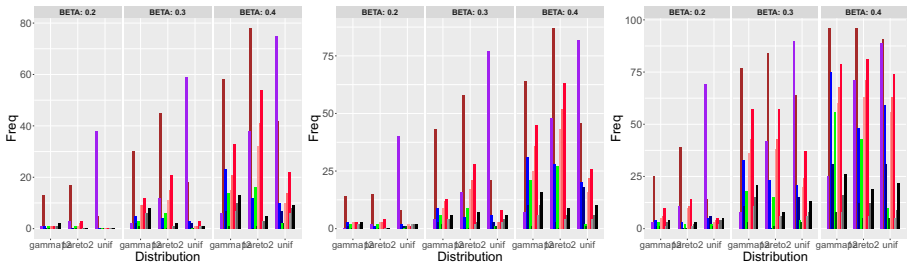
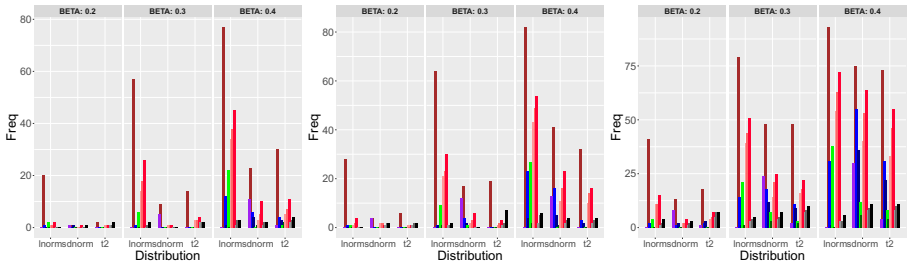


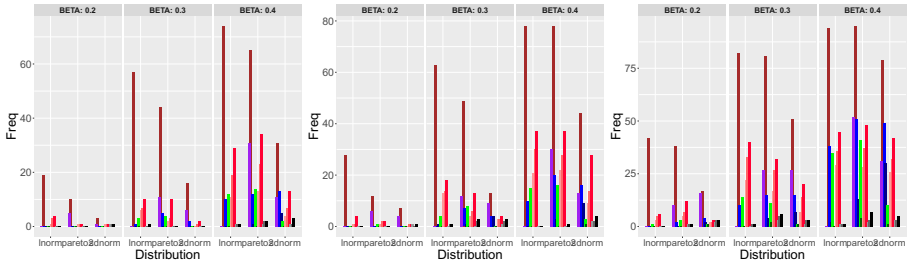
Figure A.9: Selection frequency of different distributions for Poisson Model



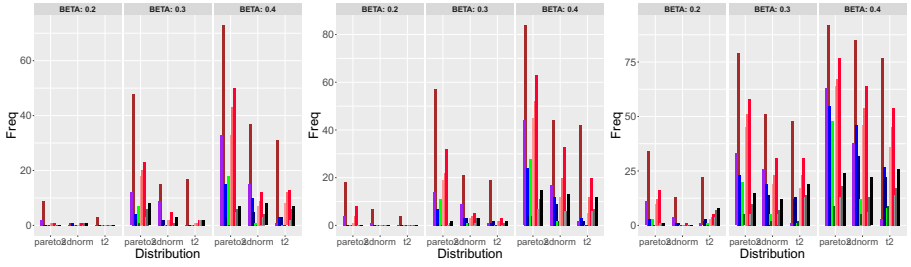
(P40) unif+gam$_{12}$+pare$_2$, MRates=(.8, 1, 2)



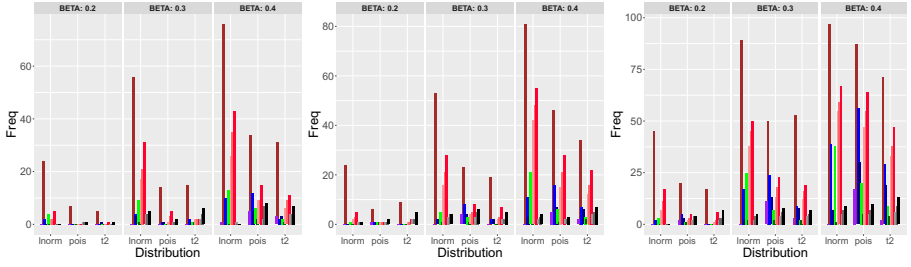
(P41) sdn+lnorm+t$_2$, MRates=(.8, 1, 2)



(P42) sdn+lnorm+pare$_2$, MRates=(.8, 1, 2)



(P43) sdn+t$_2$+pare$_2$, MRates=(.8, 1, 2)



(P44) pois+lnorm+t$_2$, MRates=(.8, 1, 2)

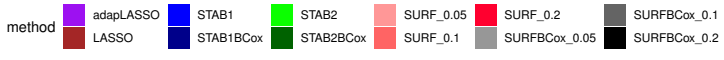
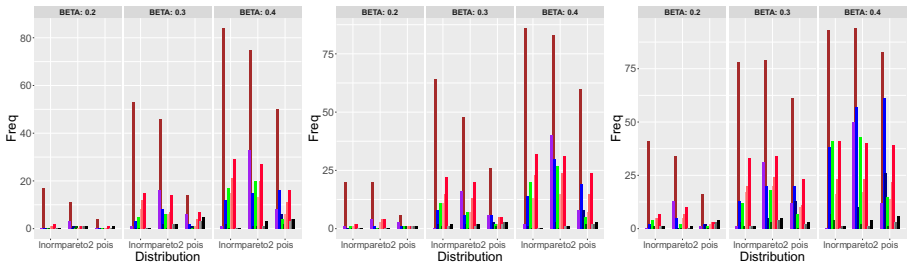
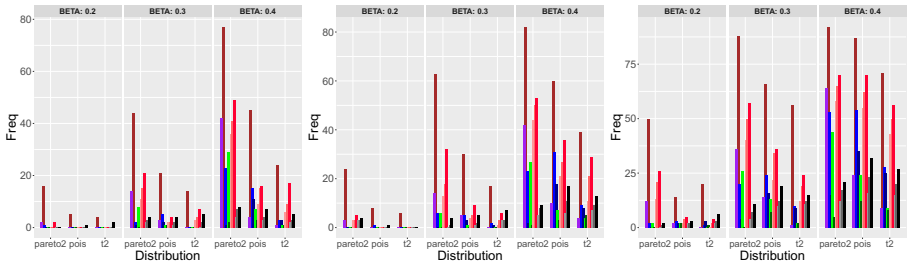


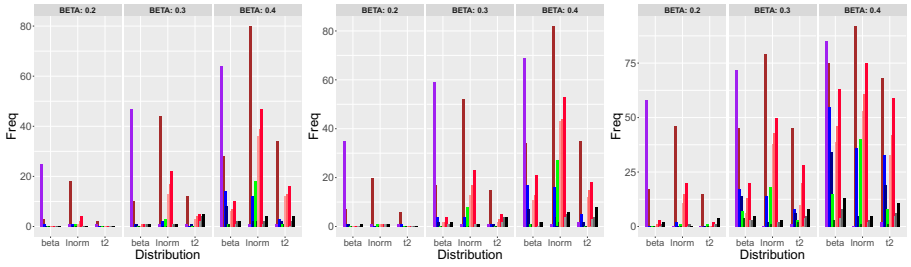
Figure A.9: Selection frequency of different distributions for Poisson Model



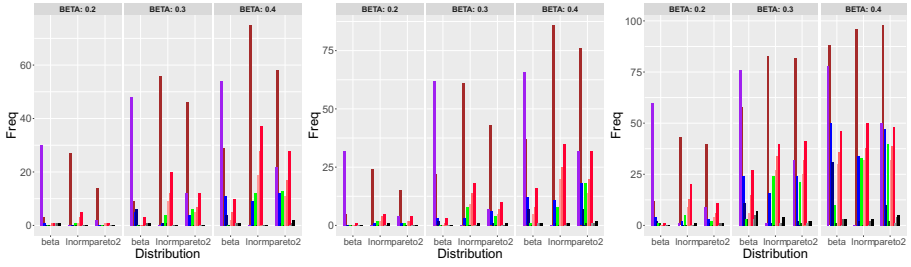
(P45) pois+lnorm+pare$_2$, MRates=(.8, 1, 2)



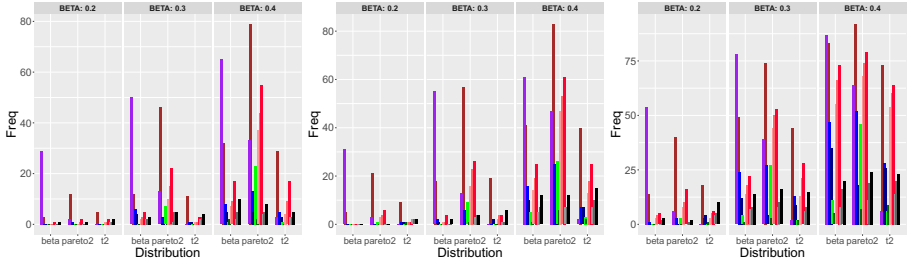
(P46) pois+t$_2$+pare$_2$, MRates=(.8, 1, 2)



(P47) beta+lnorm+t$_2$, MRates=(.8, 1, 2)



(P48) beta+lnorm+pare$_2$, MRates=(.8, 1, 2)



(P49) beta+t$_2$+pare$_2$, MRates=(.8, 1, 2)

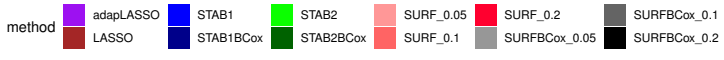
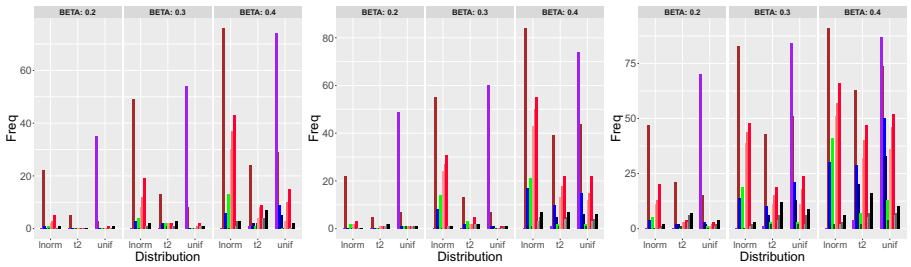
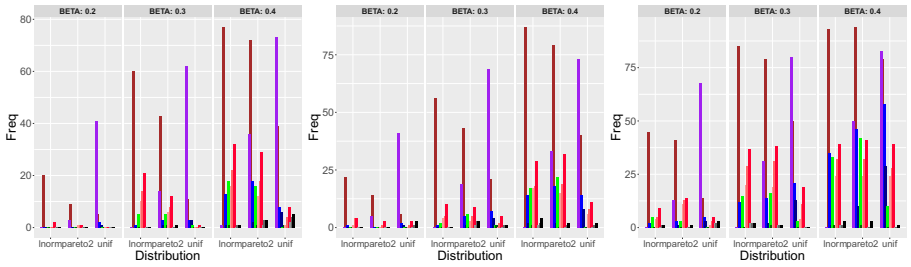


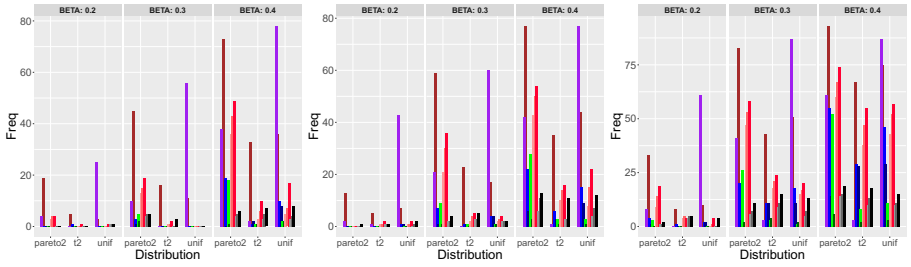
Figure A.9: Selection frequency of different distributions for Poisson Model



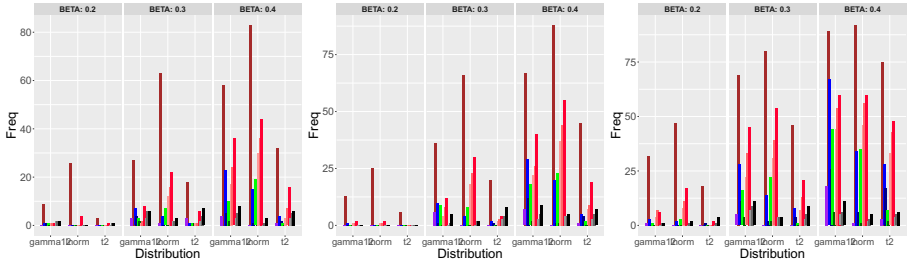
(P50) unif+lnorm+t_2, MRates=(.8, 1, 2)



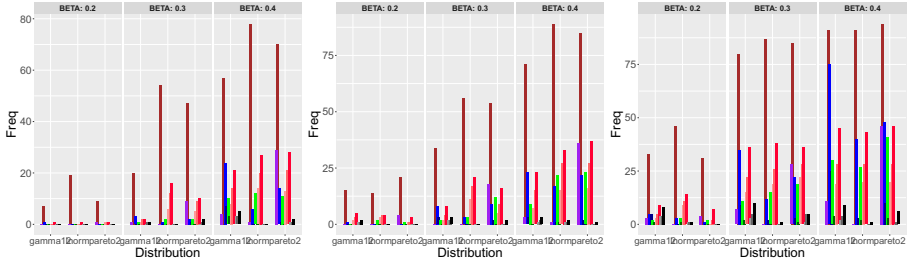
(P51) unif+lnorm+pare$_2$, MRates=(.8, 1, 2)



(P52) unif+t2+pare$_2$, MRates=(.8, 1, 2)



(P53) gam$_{12}$+lnorm+t_2, MRates=(.8, 1, 2)



(P54) gam$_{12}$+lnorm+pare$_2$, MRates=(.8, 1, 2)

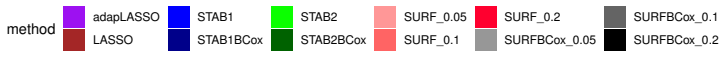
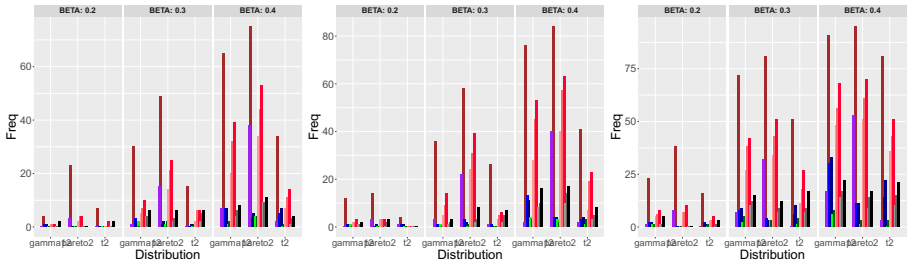
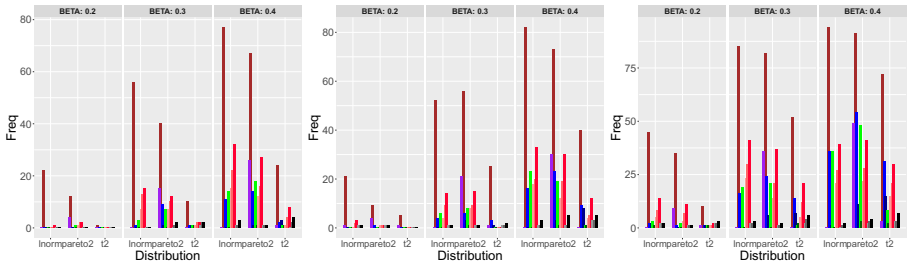


Figure A.9: Selection frequency of different distributions for Poisson Model



(P55) gam_{12}+t_2+$pare_2$, MRates=(.8, 1, 2)



(P56) lnorm+t_2+$pare_2$, MRates=(.8, 1, 2)

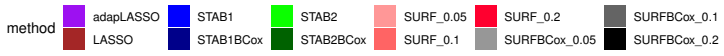


Figure A.9: Selection frequency of different distributions for Poisson Model

Bibliography

[1] Alessandra Amendola, Francesco Giordano, Maria Lucia Parrella, and Marialuisa Restaino. Variable selection in high-dimensional regression: a nonparametric procedure for business failure prediction. *Applied Stochastic Models in Business and Industry*, 33(4):355–368, 2017.

[2] Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The Annals of Statistics*, pages 813–852, 2016.

[3] Paul Bjorndahl, Joseph P Bielawski, Lihui Liu, Wei Zhou, and Hong Gu. Novel application of survival models for predicting microbial community transitions with variable selection for environmental dna. *Applied and Environmental Microbiology*, 88(6):e02146–21, 2022.

[4] Barbara Bodinier, Sabrina Rodrigues, Maryam Karimi, Sarah Filippi, Julien Chiquet, and Marc Chadeau-Hyam. Stability selection and consensus clustering in r: The r package sharp.

[5] CE Bonferroni. Teoria statistica delle classi e calcolo delle probabilità comm, 1935.

[6] Peter Bühlmann and Jacopo Mandozzi. High-dimensional variable screening and bias in subsequent inference, with an empirical comparison. *Computational Statistics*, 29(3-4):407–430, 2014.

[7] Emmanuel Candes and Terence Tao. The dantzig selector: Statistical estimation when p is much larger than n. 2007.

[8] J Gregory Caporaso, Christian L Lauber, Elizabeth K Costello, Donna Berg-Lyons, Antonio Gonzalez, Jesse Stombaugh, Dan Knights, Pawel Gajer, Jacques Ravel, Noah Fierer, et al. Moving pictures of the human microbiome. *Genome biology*, 12(5):1, 2011.

[9] Anna Catherine Cardall, Riley Chad Hales, Kaylee Brooke Tanner, Gustavious Paul Williams, and Kel N Markert. Lasso (l1) regularization for development of sparse remote-sensing models with applications in optically complex waters using gee tools. *Remote Sensing*, 15(6):1670, 2023.

[10] Ilseung Cho and Martin J Blaser. The human microbiome: at the interface of health and disease. *Nature Reviews Genetics*, 13(4):260–270, 2012.

[11] Alexandra Chouldechova and Trevor Hastie. Generalized additive model selection. *arXiv preprint arXiv:1506.03850*, 2015.

[12] Jose C Clemente, Luke K Ursell, Laura Wegener Parfrey, and Rob Knight. The impact of the gut microbiota on human health: an integrative view. *Cell*, 148(6):1258–1270, 2012.

[13] Giada De Palma, Patricia Blennerhassett, J Lu, Y Deng, AJ Park, W Green, E Denou, MA Silva, A Santacruz, Y Sanz, et al. Microbiota and host determinants of behavioural phenotype in maternally separated mice. *Nature communications*, 6:7735, 2015.

[14] Aurore Delaigle and Peter Hall. Effect of heavy tails on ultra high dimensional variable ranking methods. *Statistica Sinica*, pages 909–932, 2012.

[15] Ruben Dezeure, Peter Bühlmann, Lukas Meier, Nicolai Meinshausen, et al. High-dimensional inference: Confidence intervals, p-values and r-software hdi. *Statistical science*, 30(4):533–558, 2015.

[16] Jianqing Fan, Yingying Fan, and Emre Barut. Adaptive robust variable selection. *Annals of statistics*, 42(1):324, 2014.

[17] Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 70(5):849–911, 2008.

[18] Jianqing Fan and Jinchi Lv. A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20(1):101, 2010.

[19] Noah Fierer, Christian L Lauber, Nick Zhou, Daniel McDonald, Elizabeth K Costello, and Rob Knight. Forensic identification using skin bacterial communities. *Proceedings of the National Academy of Sciences*, 107(14):6477–6481, 2010.

[20] Joel B Fontanarosa and Yang Dai. Using lasso regression to detect predictive aggregate effects in genetic studies. In *BMC proceedings*, volume 5, pages 1–5. BioMed Central, 2011.

[21] Robin Genuer, Jean-Michel Poggi, and Christine Tuleau-Malot. Vsurf: an r package for variable selection using random forests. *The R Journal*, 7(2):19–33, 2015.

[22] Debashis Ghosh and Arul M Chinnaiyan. Classification and selection of biomarkers in genomic data using lasso. *Journal of Biomedicine and Biotechnology*, 2005(2):147, 2005.

[23] Jack A Gilbert, Janet K Jansson, and Rob Knight. The earth microbiome project: successes and aspirations. *BMC biology*, 12:1–4, 2014.

[24] Edouard Grave, Guillaume R Obozinski, and Francis R Bach. Trace lasso: a trace norm regularization for correlated designs. In *Advances in Neural Information Processing Systems*, pages 2187–2195, 2011.

[25] Trevor Hastie and Robert Tibshirani. Generalized additive models: some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987.

[26] Trevor Hastie, Robert Tibshirani, Jerome Friedman, T Hastie, J Friedman, and R Tibshirani. *The elements of statistical learning*, volume 2. Springer, 2009.

[27] Junxiu He, Xiaoting Ge, Hong Cheng, Yu Bao, Xiuming Feng, Gaohui Zan, Fei Wang, Yunfeng Zou, and Xiaobo Yang. Sex-specific associations of exposure to metal mixtures with telomere length change: Results from an 8-year longitudinal study. *Science of The Total Environment*, 811:151327, 2022.

[28] Benjamin Hofner, Luigi Boccuto, and Markus Göker. Controlling false discoveries in high-dimensional situations: boosting with stability selection. *BMC bioinformatics*, 16:1–17, 2015.

[29] Tao Hou, Fu Liu, Yun Liu, Qing Yu Zou, Xiao Zhang, and Ke Wang. Classification of metagenomics data at lower taxonomic level using a robust supervised classifier. *Evolutionary Bioinformatics*, 11:EBO–S20523, 2015.

[30] Jian Huang, Shuangge Ma, and Cun-Hui Zhang. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, pages 1603–1618, 2008.

[31] Shiqiong Huang and Micol Marchetti-Bowick. Summary and discussion of : " stability selection " statistics journal club , 36-825. 2014.

[32] Jenny Jin, Kenji Schorpp, Daniel Samaga, Kristian Unger, Kamyar Hadian, and Brent R Stockwell. Machine learning classifies ferroptosis and apoptosis cell death modalities with tfr1 immunostaining. *ACS Chemical Biology*, 17(3):654–660, 2022.

[33] Xiaobei Li and Ross Jacobucci. Regularized structural equation modeling with stability selection. *Psychological Methods*, 27(4):497, 2022.

[34] Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate nonparametric regression. *The Annals of Statistics*, pages 2272–2297, 2006.

[35] Zhengyan Lin, Yanbiao Xiang, and Caiya Zhang. Adaptive lasso in high-dimensional settings. *Journal of Nonparametric Statistics*, 21(6):683–696, 2009.

[36] Lihui Liu, Hong Gu, Johan Van Limbergen, and Toby Kenney. Surf: a new method for sparse variable selection, with application in microbiome data analysis. *Statistics in Medicine*, 40(4):897–919, 2021.

[37] Richard Lockhart, Jonathan Taylor, Ryan J Tibshirani, and Robert Tibshirani. A significance test for the lasso. *Annals of statistics*, 42(2):413, 2014.

[38] Yin Lou, Jacob Bien, Rich Caruana, and Johannes Gehrke. Sparse partially linear additive models. *Journal of Computational and Graphical Statistics*, 25(4):1126–1140, 2016.

[39] Nicolai Meinshausen and Peter Bühlmann. Stability selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(4):417–473, 2010.

[40] Sonia Michail, Matthew Durbin, Dan Turner, Anne M Griffiths, David R Mack, Jeffrey Hyams, Neal Leleiko, Harshavardhan Kenche, Adrienne Stolfi, and Eytan Wine. Alterations in the gut microbiome of children with severe ulcerative colitis. *Inflammatory bowel diseases*, 18(10):1799–1808, 2012.

[41] Julia Oh, Allyson L Byrd, Morgan Park, Heidi H Kong, Julia A Segre, NISC Comparative Sequencing Program, et al. Temporal stability of the human skin microbiome. *Cell*, 165(4):854–866, 2016.

[42] Trevor Park and George Casella. The bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[43] Ashley Petersen and Daniela Witten. Data-adaptive additive modeling. *Statistics in medicine*, 38(4):583–600, 2019.

[44] Edwin JG Pitman. Significance tests which may be applied to samples from any populations. *Supplement to the Journal of the Royal Statistical Society*, 4(1):119–130, 1937.

[45] Pradeep Ravikumar, John Lafferty, Han Liu, and Larry Wasserman. Sparse additive models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 71(5):1009–1030, 2009.

[46] Leah Reshef, Amir Kovacs, Amos Ofer, Lior Yahav, Nitsan Maharshak, Nirit Keren, Fred M Konikoff, Hagit Tulchinsky, Uri Gophna, and Iris Dotan. Pouch inflammation is associated with a decrease in specific bacterial taxa. *Gastroenterology*, 149(3):718–727, 2015.

[47] Giacomo Rossi, Graziano Pengo, Marco Caldin, Angela Palumbo Piccionello, Jörg M Steiner, Noah D Cohen, Albert E Jergens, and Jan S Suchodolski. Comparison of microbiological, histological, and immunomodulatory parameters in response to treatment with either combination therapy with prednisone and metronidazole or probiotic vsl# 3 strains in dogs with idiopathic inflammatory bowel disease. *PloS one*, 9(4):e94699, 2014.

[48] Srikanth Ryali, Tianwen Chen, Kaustubh Supekar, and Vinod Menon. Estimation of functional connectivity in fmri data using stability selection-based sparse partial correlation with elastic net penalty. *Neuroimage*, 59(4):3852–3861, 2012.

[49] Veeranjaneyulu Sadhanala and Ryan J Tibshirani. Additive models with trend filtering. *The Annals of Statistics*, 47(6):3032–3068, 2019.

[50] Diego Franco Saldana and Yang Feng. Sis: An r package for sure independence screening in ultrahigh-dimensional statistical models. *Journal of Statistical Software*, 83:1–25, 2018.

[51] Kun Shan, Xiaoxiao Wang, Hong Yang, Botian Zhou, Lirong Song, and Mingsheng Shang. Use statistical machine learning to detect nutrient thresholds in microcystis blooms and microcystin management. *Harmful algae*, 94:101807, 2020.

[52] Bo Shen. Pouchitis: what every gastroenterologist needs to know. *Clinical Gastroenterology and Hepatology*, 11(12):1538–1549, 2013.

[53] Martin Sill, Sebastian Kaiser, Axel Benner, and Annette Kopp-Schneider. Robust biclustering by sparse singular value decomposition incorporating stability selection. *Bioinformatics*, 27(15):2089–2097, 2011.

[54] Rasnik K Singh, Hsin-Wen Chang, DI Yan, Kristina M Lee, Derya Ucmak, Kirsten Wong, Michael Abrouk, Benjamin Farahnik, Mio Nakamura, Tian Hao Zhu, et al. Influence of diet on the gut microbiome and implications for human health. *Journal of translational medicine*, 15(1):1–17, 2017.

[55] Tingni Sun and Cun-Hui Zhang. Scaled sparse linear regression. *Biometrika*, 99(4):879–898, 2012.

[56] J Kenneth Tay and Robert Tibshirani. Reluctant generalised additive modelling. *International Statistical Review*, 88:S205–S224, 2020.

[57] Linda V Thomas, Theo Ockhuizen, and Kaori Suzuki. Exploring the influence of the gut microbiota and probiotics on health: a symposium report. *British Journal of Nutrition*, 112(S1):S1–S18, 2014.

[58] Elizabeth Thursby and Nathalie Juge. Introduction to the human gut microbiota. *Biochemical journal*, 474(11):1823–1836, 2017.

[59] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

[60] Robert Tibshirani, Michael Saunders, Saharon Rosset, Ji Zhu, and Keith Knight. Sparsity and smoothness via the fused lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(1):91–108, 2005.

[61] Ryan J Tibshirani, Jonathan Taylor, Richard Lockhart, and Robert Tibshirani. Exact post-selection inference for sequential regression procedures. *Journal of the American Statistical Association*, 111(514):600–620, 2016.

[62] Andrea D Tyler, Natalie Knox, and Boyko and Kabakchiev. Characterization of the gut-associated microbiome in inflammatory pouch complications following ileal pouch-anal anastomosis. *PloS one*, 8(9):e66934, 2013.

[63] Anita Y Voigt, Paul I Costea, Jens Roat Kultima, Simone S Li, Georg Zeller, Shinichi Sunagawa, and Peer Bork. Temporal and technical variability of human gut metagenomes. *Genome biology*, 16(1):1, 2015.

[64] Hansheng Wang, Runze Li, and Chih-Ling Tsai. Tuning parameter selectors for the smoothly clipped absolute deviation method. *Biometrika*, 94(3):553–568, 2007.

[65] Canhong Wen, Aijun Zhang, Shijie Quan, and Xueqin Wang. Bess: an r package for best subset selection in linear, logistic and cox proportional hazards models. *Journal of Statistical Software*, 94:1–24, 2020.

[66] Samuel S Wilks. The large-sample distribution of the likelihood ratio for testing composite hypotheses. *The annals of mathematical statistics*, 9(1):60–62, 1938.

[67] Simon N Wood. *Generalized additive models: an introduction with R*. CRC press, 2017.

[68] Yilei Wu, Yingli Qin, and Mu Zhu. Quadratic discriminant analysis for high-dimensional data. *Statistica Sinica*, 29(2):939–960, 2019.

[69] Jian Xiao, Li Chen, Stephen Johnson, Yue Yu, Xianyang Zhang, and Jun Chen. Predictive modeling of microbiome data using a phylogeny-regularized generalized linear mixed model. *Frontiers in microbiology*, 9:1391, 2018.

[70] Xiaohan Yan and Jacob Bien. Rare feature selection in high dimensions. *Journal of the American Statistical Association*, 116(534):887–900, 2021.

[71] In-Kwon Yeo and Richard A Johnson. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, 2000.

[72] Ping Yin, Ning Mao, Chao Zhao, Jiangfen Wu, Chao Sun, Lei Chen, and Nan Hong. Comparison of radiomics machine-learning classifiers and feature selection for differentiation of sacral chordoma and sacral giant cell tumour based on 3d computed tomography features. *European radiology*, 29:1841–1847, 2019.

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

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

[75] Roman Zakharov and Pierre Dupont. Stable lasso for high-dimensional feature selection through proximal optimization. *Regularization and Optimization and Kernel Methods and Support Vector Machines: Theory and Applications, Brussels and Belgium*, 2013.

[76] Tao Zhang and Joseph E Cavanaugh. A multistage algorithm for best-subset model selection based on the kullback–leibler discrepancy. *Computational statistics*, 31(2):643–669, 2016.

[77] Yue Zhang, Weihong Guo, and Soumya Ray. On the consistency of feature selection with lasso for non-linear targets. In *International Conference on Machine Learning*, pages 183–191. PMLR, 2016.

[78] Peng Zhao and Bin Yu. On model selection consistency of lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

[79] Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

[80] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.