High-dimensional Inference: Confidence intervals, p-values and R-Software `hdi`

Ruben Dezeure, Peter Bühlmann, Lukas Meier and Nicolai Meinshausen

ETH Zurich

Abstract. We present a (selective) review of recent frequentist high-dimensional inference methods for constructing p-values and confidence intervals in linear and generalized linear models. We include a broad, comparative empirical study which complements the viewpoint from statistical methodology and theory. Furthermore, we introduce and illustrate the R-package **hdi** which easily allows the use of different methods and supports reproducibility.

Key words and phrases: Clustering, Confidence interval, Generalized linear model, High-dimensional statistical inference, Linear model, Multiple testing, P-value, R-software.

1. INTRODUCTION

Over the last 15 years, a lot of progress has been achieved in high-dimensional statistics where the number of parameters can be much larger than sample size, covering (nearly) optimal point estimation, efficient computation, and applications in many different areas; see for example the books by Hastie et al. (2009), Bühlmann and van de Geer (2011) or the review article by Fan and Lv (2010). The core task of statistical inference accounting for uncertainty, in terms of frequentist confidence intervals and hypothesis testing, is much less developed. Recently, a few methods for assigning p-values and constructing confidence intervals have been suggested (Wasserman and Roeder, 2009; Meinshausen et al., 2009; Bühlmann, 2013; Zhang and Zhang, 2014; Lockhart et al., 2014; van de Geer et al., 2014; Javanmard and Montanari, 2013; Meinshausen, 2013).

The current paper has three main pillars: (i) a (selective) review of the development in frequentist high-dimensional inference methods for p-values and confidence regions; (ii) presenting the first broad, comparative empirical study among different methods, mainly for linear models: since the methods are mathematically justified under non-checkable and sometimes non-comparable assumptions, a thorough simulation study should lead to additional insights about reliability and performance of various procedures; (iii) presenting the R-package **hdi** (**h**igh-**d**imensional **i**nference) which enables to easily use many of the different methods for inference in high-dimensional generalized linear models. In addition, we include a recent line of methodology allowing to detect significant groups of highly correlated variables which could not be inferred as individually significant single

ETH Zurich, Main building, Rämistrasse 101, 8092 Zürich, Switzerland, (e-mail: dezeure@stat.math.ethz.ch; buhlmann@stat.math.ethz.ch; meier@stat.math.ethz.ch; meinshausen@stat.math.ethz.ch)

variables (Meinshausen, 2013). The review and exposition in Bühlmann et al. (2014a) is vaguely related to points (i) and (iii) above, but much more focusing on an application oriented viewpoint and covering much less statistical methodology, theory and computational details.

Our comparative study, the point (ii) mentioned above, exhibits interesting results indicating that more "stable" procedures based on Ridge-estimation or random sample splitting with subsequent aggregation are somewhat more reliable for type I error control than asymptotically power-optimal methods. Such results cannot be obtained by comparing underlying assumptions of different methods, since these assumptions are often too crude and far from necessary. As expected, we are unable to pinpoint to a method which is (nearly) best in all considered scenarios. In view of this, we also want to offer a collection of useful methods for the community, in terms of our R-package `hdi` mentioned in the point (iii) above.

2. INFERENCE FOR LINEAR MODELS

We consider first a high-dimensional linear model, while extensions are discussed in Section 3:

$$Y = \mathbf{X}\beta^0 + \varepsilon \tag{2.1}$$

with $n \times p$ fixed or random design matrix $\mathbf{X}$, $n \times 1$ response and error vectors Y and ε, respectively. The errors are assumed to be independent of $\mathbf{X}$ (for random design) with i.i.d. entries having $\mathbb{E}[\varepsilon_i] = 0$. We allow for high-dimensional settings where $p \gg n$. In the further development, the active set or the set of relevant variables

$$S_0 = \{j;\ \beta_j^0 \neq 0,\ j = 1, \ldots, p\},$$

as well as its cardinality $s_0 = |S_0|$, are important quantities. The main goals of this section are the construction of confidence intervals and p-values for individual regression parameters β_j^0 $(j = 1, \ldots, p)$ and corresponding multiple testing adjustment. The former is a highly non-standard problem in high-dimensional settings while for the latter we can use standard well-known techniques: when considering both goals simultaneously, though, one can develop more powerful multiple testing adjustments. The Lasso (Tibshirani, 1996) is among the most popular procedures for estimating the unknown parameter β^0 in a high-dimensional linear model. It exhibits desirable or sometimes even optimal properties for point estimation such as prediction of $\mathbf{X}\beta^0$ or of a new response Y_{new}, estimation in terms of $\|\hat{\beta} - \beta^0\|_q$ for e.g. $q = 1, 2$, and variable selection or screening; see for example in the book of Bühlmann and van de Geer (2011). For assigning uncertainties in terms of confidence intervals or hypothesis testing, however, the plain Lasso seems inappropriate: it is very difficult to characterize the distribution of the estimator in the high-dimensional setting: Knight and Fu (2000) derive asymptotic results for fixed dimension as sample size $n \to \infty$ and already for such simple situations, the asymptotic distribution of the Lasso has point mass at zero. This implies, because of non-continuity of the distribution, that standard bootstrapping and subsampling schemes are delicate to apply and uniform convergence to the limit seems hard to achieve. The latter means that the estimator is exposed to undesirable super-efficiency problems, as illustrated in Section 2.4. All the problems mentioned are expected to apply not only for the Lasso but for other sparse estimators as well.

In high-dimensional settings and for general fixed design $\mathbf{X}$, the regression parameter is not identifiable. However, when making some restrictions on the design, one can ensure that the regression vector is identifiable. The so-called compatibility condition on the design $\mathbf{X}$ (van de Geer, 2007) is a rather weak assumption (van de Geer and Bühlmann, 2009) which guarantees identifiability and oracle (near) optimality results for the Lasso. For the sake of completeness, the compatibility condition is described in the Appendix A.1.

When assuming the compatibility condition with constant ϕ_0^2 (ϕ_0^2 is close to zero for rather ill-posed designs, and sufficiently larger than zero for well-posed designs), the Lasso has the following property: for Gaussian errors and if $\lambda \asymp \sqrt{\log(p)/n}$, we have with high probability that

$$\|\hat{\beta} - \beta^0\|_1 \leq 4 s_0 \lambda / \phi_0^2. \tag{2.2}$$

Thus, if $s_0 \ll \sqrt{n/\log(p)}$ and $\phi_0^2 \geq M > 0$, we have $\|\hat{\beta} - \beta^0\|_1 \to 0$ and hence, the parameter β^0 is identifiable.

Another often-used assumption, although not necessary by any means, is the so-called beta-min assumption:

$$\min_{j \in S_0} |\beta_j^0| \geq \beta_{\min}, \tag{2.3}$$

for some choice of constant $\beta_{\min} > 0$. The result in (2.2) immediately implies the screening property: if $\beta_{\min} > 4 s_0 \lambda / \phi_0^2$, then

$$\hat{S} = \{j;\ \hat{\beta}_j \neq 0\} \supseteq S_0. \tag{2.4}$$

Thus, the screening property holds when assuming the compatibility and beta-min condition. The power of the screening property is a massive dimensionality reduction (in the original variables) because $|\hat{S}| \leq \min(n,p)$; thus, if $p \gg n$, the selected set $\hat{S}$ is much smaller than the full set of p variables. Unfortunately, the required conditions are overly restrictive and exact variable screening seems rather unrealistic in practical applications (Bühlmann and Mandozzi, 2014).

2.1 Different methods

We describe here three different methods for construction of statistical hypothesis tests or confidence intervals. Alternative procedures are presented in Sections 2.3 and 2.4.

2.1.1 Multi sample-splitting A generic way for deriving p-values in hypotheses testing is given by splitting the sample with indices $\{1, \ldots, n\}$ into two equal halves denoted by I_1 and I_2, i.e., $I_r \subset \{1, \ldots, n\}$ $(r = 1, 2)$ with $|I_1| = \lfloor n/2 \rfloor$, $|I_2| = n - \lfloor n/2 \rfloor$, $I_1 \cap I_2 = \emptyset$ and $I_1 \cup I_2 = \{1, \ldots n\}$. The idea is to use the first half I_1 for variable selection and the second half I_2 with the reduced set of selected variables (from I_1) for statistical inference in terms of p-values. Such a sample-splitting procedure avoids the over-optimism to use the data twice for selection and inference after selection (without taking the effect of selection into account).

Consider a method for variable selection based on the first half of the sample:

$$\hat{S}(I_1) \subset \{1, \ldots, p\}.$$

A prime example is the Lasso which selects all the variables whose corresponding estimated regression coefficients are different from zero. We then use the second half of the sample I_2 for constructing p-values, based on the selected variables $\hat{S}(I_1)$. If the cardinality $|\hat{S}(I_1)| \leq n/2 \leq |I_2|$, we can run ordinary least squares estimation using the subsample I_2 and the selected variables $\hat{S}(I_1)$, i.e., we regress Y_{I_2} on $\mathbf{X}_{I_2}^{(\hat{S}(I_1))}$ where the sub-indices denote the sample half and the super-index stands for the selected variables, respectively. Thereby, we implicitly assume that the matrix $\mathbf{X}_{I_2, \hat{S}(I_1)}$ has full rank $|\hat{S}(I_1)|$. Thus, from such a procedure, we obtain p-values $P_{\text{t-test},j}$ for testing $H_{0,j}:\ \beta_j^0 = 0$, for

$j \in \hat{S}(I_1)$, from the classical t-tests, assuming Gaussian errors or relying on asymptotic justification by the central limit theorem. To be more precise, we define (raw) p-values

$$P_{\text{raw},j} = \begin{cases} P_{\text{t-test},j} \text{ based on } Y_{I_2}, \mathbf{X}_{I_2}^{(\hat{S}(I_1))} & \text{if } j \in \hat{S}(I_1), \\ 1 & \text{if } j \notin \hat{S}(I_1). \end{cases}$$

An interesting feature of such a sample-splitting procedure is the adjustment for multiple testing. For example, if we wish to control the familywise error rate over all considered hypotheses $H_{0,j}$ $(j = 1, \ldots, p)$, a naive approach would employ a Bonferroni-Holm correction over the p tests. This is not necessary: we only need to control over the considered $|\hat{S}(I_1)|$ tests in I_2. Therefore, a Bonferroni corrected p-value for $H_{0,j}$ is given by

$$P_{\text{corr},j} = \min(P_{\text{raw},j} \cdot |\hat{S}(I_1)|, 1).$$

In high-dimensional scenarios, $p \gg n > \lfloor n/2 \rfloor = |\hat{S}(I_1)|$, where the latter inequality is an implicit assumption which holds for the Lasso (under weak assumptions), and thus, the correction factor employed here is rather small. Such corrected p-values control the familywise error rate in multiple testing when assuming the screening property in (2.4) for the selector $\hat{S} = \hat{S}(I_1)$ based on the first half I_1 only, exactly as stated in Fact 1 below. The reason is that the screening property ensures that the reduced model is a correct model, and hence the result is not surprising. In practice, the screening property typically does not hold exactly but it is not a necessary condition for constructing valid p-values Bühlmann and Mandozzi (2014).

The idea about sample-splitting and subsequent statistical inference is implicitly contained in Wasserman and Roeder (2009). We summarize the whole procedure as follows:

Single sample-splitting for multiple testing of $H_{0,j}$ among $j = 1, \ldots, p$:

1. Split (partition) the sample $\{1, \ldots, n\} = I_1 \cup I_2$ with $I_1 \cap I_2 = \emptyset$ and $|I_1| = \lfloor n/2 \rfloor$ and $|I_2| = n - \lfloor n/2 \rfloor$.
2. Using I_1 only, select the variables $\hat{S} \subseteq \{1, \ldots, p\}$. Assume or enforce that $|\hat{S}| \le |I_1| = \lfloor n/2 \rfloor \le |I_2|$.
3. Denote the design matrix with the selected set of variables by $\mathbf{X}^{(\hat{S})}$. Based on I_2 with data $(Y_{I_2}, \mathbf{X}_{I_2}^{(\hat{S})})$, compute p-values $P_{\text{raw,j}}$ for $H_{0,j}$, for $j \in \hat{S}$, from classical least squares estimation (i.e. t-test which can be used since $|\hat{S}(I_1)| \le |I_2|$). For $j \notin \hat{S}$, assign $P_{\text{raw},j} = 1$.
4. Correct the p-values for multiple testing: consider

$$P_{\text{corr},j} = \min(P_j \cdot |\hat{S}|, 1)$$

which is an adjusted p-value for $H_{0,j}$ for controlling the familywise error rate.

A major problem of the single sample-splitting method is its sensitivity with respect to the choice of splitting the entire sample: sample splits lead to wildly different p-values. We call this undesirable phenomenon a p-value lottery, and Figure 1 provides an illustration. To overcome the "p-value lottery" we can run the sample-splitting method B times, with B large. Thus, we obtain a collection of p-values for the jth hypothesis $H_{0,j}$:

$$P_{\text{corr},j}^{[1]}, \ldots, P_{\text{corr},j}^{[B]} \ (j = 1, \ldots, p).$$

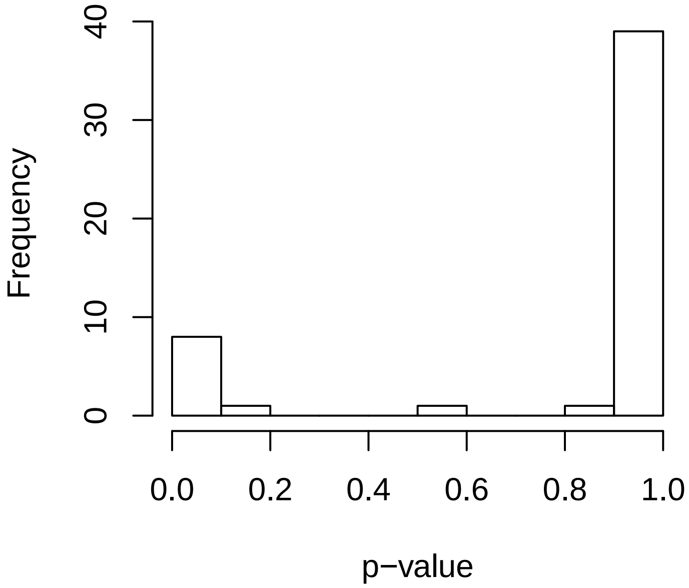

FIG 1. *Histogram of p-values $P_{\text{corr},j}$ for a single covariable, in the* `riboflavin` *data-set, when doing 50 different (random) sample splits. The figure is taken from Bühlmann et al. (2014a).*

The task is now to do an aggregation to a single p-value. Because of dependence among $\{P^{[b]}_{\text{corr},j};\ b = 1,\ldots,B\}$, because all the different halve samples are part of the same full sample, an appropriate aggregation needs to be developed. A simple solution is to use an empirical γ-quantile with $0 < \gamma < 1$:

$$Q_j(\gamma) = \min\left(\text{emp. } \gamma\text{-quantile}\{P^{[b]}_{\text{corr},j}/\gamma;\ b = 1,\ldots,B\}, 1\right).$$

For example, with $\gamma = 1/2$, this amounts to taking the sample median $\{P^{[b]}_{\text{corr},j};\ b = 1,\ldots,B\}$ and multiply it with the factor 2. A bit more sophisticated approach is to choose the best and properly scaled γ-quantile in the range $(\gamma_{\min}, 1)$ (e.g., $\gamma_{\min} = 0.05$), leading to the aggregated p-value

$$P_j = \min\left((1 - \log(\gamma_{\min})) \inf_{\gamma \in (\gamma_{\min},1)} Q_j(\gamma)\right) \ (j = 1,\ldots,p). \tag{2.5}$$

Thereby, the factor $(1-\log(\gamma_{\min}))$ is the price to be paid for searching for the best $\gamma \in (\gamma_{\min}, 1)$. This Multi sample-splitting procedure has been proposed and analyzed in Meinshausen et al. (2009), and we summarize it below. Before doing so, we remark that the aggregation of dependent p-values as described above is a general principle as described in the Appendix A.1.

Multi sample-splitting for multiple testing of $H_{0,j}$ among $j = 1,\ldots,p$:

1. Apply the single sample-splitting procedure B times leading to p-values $\{P^{[b]}_{\text{corr},j};\ b = 1,\ldots,B\}$. Typical choices are $B = 50$ or $B = 100$.

2. Aggregate these p-values as in (2.5) leading to P_j which are adjusted p-values for $H_{0,j}$ ($j = 1, \dots, p$), controlling the familywise error rate.

The Multi sample-splitting method enjoys the property that the resulting p-values are approximately reproducible and not subject to a "p-value lottery" anymore, and it controls the familywise error rate under the following assumptions:

(A1) The screening property as in (2.4) for the first half of the sample: $\mathbb{P}[\hat{S}(I_1) \supseteq S_0] \geq 1 - \delta$ for some $0 < \delta < 1$.

(A2) The reduced design matrix for the second half of the sample satisfies: $\mathrm{rank}(\mathbf{X}_{I_2}^{(\hat{S}(I_1))}) = |\hat{S}(I_1)|$.

Fact 1. *(Meinshausen et al. (2009))*
Consider a linear model as in (2.1) with fixed design $\mathbf{X}$ *and Gaussian errors. Assume (A1)–(A2). Then, for a significance level* $0 < \alpha < 1$ *and denoting by* B *the number of sample splits:*

$$\mathbb{P}[\cup_{j \in S_0^c} I(P_j \leq \alpha)] \leq \alpha + B\delta,$$

that is, the familywise error rate (FWER) is controlled up to the additional (small) value $B\delta$*.*

A proof is given in Meinshausen et al. (2009). Assumption (A2) typically holds for sparse estimators such as the Lasso satisfying $|\hat{S}(I_1)| \leq |I_1| = \lfloor n/2 \rfloor \leq |I_2| = n - \lfloor n/2 \rfloor$.

The screening property (A1). The screening property (A1) is not a necessary condition for constructing valid p-values and can be replaced by a zonal assumption requiring the following: there is a gap between large and small regression coefficients and there are not too many small nonzero regression coefficients Bühlmann and Mandozzi (2014). Still, such a zonal assumption makes a requirement about the unknown β^0 and the absolute values of its components: but this is the essence of the question in hypothesis testing to infer whether coefficients are sufficiently different from zero, and one would like to do such a test without an assumption on the true values.

The Lasso satisfies (A1) when assuming the compatibility condition (A.1) on the design $\mathbf{X}$ and a beta-min condition (2.3), as shown in (2.4). Again, the beta-min assumption should be avoided when constructing a hypothesis test about the unknown components of β^0.

Fact 1 has a corresponding asymptotic formulation where the dimension $p = p_n$ and the model depend on sample size n: if (A1) is replaced by $\lim_{n\to\infty} \mathbb{P}[\hat{S}(I_{1;n}) \supseteq S_{0;n}] \to 1$ and for a fixed number B, $\limsup_{n\to\infty} \mathbb{P}[\cup j \in S_0^c I(P_j \leq \alpha)] \leq \alpha$. In such an asymptotic setting, the Gaussian assumption in Fact 1 can be relaxed by invoking the central limit theorem (for the low-dimensional part).

The Multi sample-splitting method is very generic: it can be used for many other models, and its basic assumptions are an approximate screening property (2.4) and that the cardinality $|\hat{S}(I_1)| < |I_2|$ so that we only have to deal with a fairly low-dimensional inference problem. See for example Section 3 for GLMs. An extension for testing group hypotheses of the form $H_{0,G} : \ \beta_j = 0$ for all $j \in G$ is indicated in Section 4.1.

Confidence intervals can be constructed based on the duality with the p-values from equation (2.5). A procedure is described in detail in the Appendix A.2. The idea to invert the p-value method is to apply a bisection method having a point in and a point outside of the confidence interval. To verify if a point is inside the *aggregated* confidence interval, one looks at the fraction of confidence intervals from the splits which cover the point.

2.1.2 Regularized projection: de-sparsifying the Lasso We describe here a method, first introduced by Zhang and Zhang (2014), which does not require an assumption about β^0 except for sparsity.

It is instructive to give a motivation starting with the low-dimensional setting where $p < n$ and $\mathrm{rank}(\mathbf{X}) = p$. The jth component of the ordinary least squares estimator $\hat{\beta}_{\mathrm{OLS};j}$ can be obtained as follows. Do an OLS regression of $X^{(j)}$ versus all other variables $\mathbf{X}^{(-j)}$ and denote the corresponding residuals by $Z^{(j)}$. Then:

$$\hat{\beta}_{\mathrm{OLS};j} = Y^T Z^{(j)} / (X^{(j)})^T Z^{(j)} \tag{2.6}$$

can be obtained by a linear projection. In a high-dimensional setting, the residuals $Z^{(j)}$ would be equal to zero and the projection is ill-posed.

For the high-dimensional case with $p > n$, the idea is to pursue a regularized projection. Instead of ordinary least squares regression, we use a Lasso regression of $\mathbf{X}^{(j)}$ versus $\mathbf{X}^{(-j)}$ with corresponding residual vector $Z^{(j)}$: such a penalized regression involves a regularization parameter λ_j for the Lasso, and hence $Z^{(j)} = Z^{(j)}(\lambda_j)$. As in (2.6), we immediately obtain (for any vector $Z^{(j)}$):

$$\begin{aligned}\frac{Y^T Z^{(j)}}{(X^{(j)})^T Z^{(j)}} &= \beta^0 + \sum_{k \neq j} P_{jk} \beta_k^0 + \frac{\varepsilon^T Z^{(j)}}{(X^{(j)})^T Z^{(j)}},\\ P_{jk} &= (X^{(k)})^T Z^{(j)} / (X^{(j)})^T Z^{(j)}.\end{aligned} \tag{2.7}$$

We note that in the low-dimensional case with $Z^{(j)}$ being the residuals from ordinary least squares, due to orthogonality, $P_{jk} = 0$. When using the Lasso-residuals for $Z^{(j)}$, we do not have exact orthogonality and a bias arises. Thus, we make a bias correction by plugging in the Lasso estimator $\hat{\beta}$ (of the regression Y versus $\mathbf{X}$): the bias-corrected estimator is

$$\hat{b}_j = \frac{Y^T Z^{(j)}}{(X^{(j)})^T Z^{(j)}} - \sum_{k \neq j} P_{jk} \hat{\beta}_k. \tag{2.8}$$

Using (2.7) we obtain:

$$\sqrt{n}(\hat{b}_j - \beta_j^0) = \frac{n^{-1/2} \varepsilon^T Z^{(j)}}{n^{-1} (X^{(j)})^T Z^{(j)}} + \sum_{k \neq j} \sqrt{n} P_{jk} (\hat{\beta}_k - \beta_k^0).$$

The first term on the right-hand side has a Gaussian distribution, when assuming Gaussian errors; otherwise, it has an asymptotic Gaussian distribution assuming that $\mathbb{E}|\varepsilon_i|^{2+\kappa} < \infty$ for $\kappa > 0$. We will argue in Appendix A.1 that the second term is negligible under the following assumptions:

(B1) The design matrix $\mathbf{X}$ has compatibility constant bounded away from zero, and the sparsity is $s_0 = o(\sqrt{n}/\log(p))$.

(B2) The rows of $\mathbf{X}$ are fixed realizations of i.i.d. random vectors $\sim \mathcal{N}_p(0, \Sigma)$, and the minimal eigenvalue of Σ is bounded away from zero.

(B3) The inverse Σ^{-1} is row-sparse with $s_j = \sum_{k \neq j} I(\Sigma_{jk}^{-1} \neq 0) = o(n/\log(p))$.

FACT 2. *(van de Geer et al., 2014) Consider a linear model as in (2.1) with fixed design and Gaussian errors. Assume (B1) and (B2). Then,*

$$\begin{aligned}&\sqrt{n}\sigma_\varepsilon^{-1}(\hat{b} - \beta^0) = W + \Delta,\\ &W \sim \mathcal{N}_p(0, \Omega),\ \Omega_{jk} = \frac{(Z^{(j)})^T Z^{(k)}}{[(X^{(j)})^T Z^{(j)}][(X^{(k)})^T Z^{(k)}]},\\ &\|\Delta\|_\infty = o_P(1).\end{aligned}$$

The asymptotic implications of Fact 2 are as follows:

$$\sigma_\varepsilon^{-1}\Omega_{jj}^{-1/2}\sqrt{n}(\hat{b}_j - \beta_j^0) \Rightarrow \mathcal{N}(0,1),$$

from which we can immediately construct a confidence interval or hypothesis test by plugging in an estimate $\hat{\sigma}_\varepsilon$ as briefly discussed in 2.1.4.

More general than the statements in Fact 2, the following holds assuming additionally (B3) (van de Geer et al., 2014): the asymptotic variance $\sigma_\varepsilon^2\Omega_{jj}$ reaches the Cramer-Rao lower bound, which equals $\sigma_\varepsilon^2\Sigma_{jj}^{-1}$, and the estimator $\hat{b}_j$ is efficient in the sense of semiparametric inference. Furthermore, the convergence in Fact 2 is uniform over the subset of the parameter space where the number of non-zero coefficients $\|\beta^0\|_0$ is small. and therefore, we obtain *honest* confidence intervals and tests. In particular, both of these results say that all the complications in post-model selection do not arise (Leeb and Pötscher, 2003) and yet, $\hat{b}_j$ is optimal for construction of confidence intervals of a single coefficient β_j^0.

From a practical perspective, we need to choose the regularization parameters λ (for the Lasso regression of Y versus $\mathbf{X}$) and λ_j (for the nodewise Lasso regressions (Meinshausen and Bühlmann, 2006) of $X^{(j)}$ versus all other variables $\mathbf{X}^{(-j)}$). Regarding the former, we advocate a choice using cross-validation; for the latter, we favor a proposal for a smaller λ_j than the one from CV, and the details are described in the Appendix A.1.

Furthermore, for a group $G \subseteq \{1,\ldots,p\}$, we can test a group hypothesis $H_{0,G}:\ \beta_j^0 = 0$ for all $j \in G$ by considering the test-statistic

$$\max_{j\in G}\sigma_\varepsilon^{-1}\Omega_{jj}^{-1/2}\sqrt{n}|\hat{b}_j| \Rightarrow \max_{j\in G}\Omega_{jj}^{-1/2}|W_j|,$$

where the limit on the right hand side occurs if the null-hypothesis $H_{0,G}$ holds true. The distribution of $\max_{j\in G}|\Omega_{jj}^{-1/2}W_j|$ can be easily simulated from dependent Gaussian random variables. We also remark that sum-type statistics for large groups cannot be easily treated because $\sum_{j\in G}|\Delta_j|$ might get out of control.

2.1.3 Ridge projection and bias correction Related to the de-sparsified Lasso estimator $\hat{b}$ in (2.8) is an approach based on Ridge estimation. We sketch here the main properties and refer to Bühlmann (2013) for a detailed treatment.

Consider

$$\hat{\beta}_{\text{Ridge}} = (n^{-1}\mathbf{X}^T\mathbf{X} + \lambda I)^{-1}n^{-1}\mathbf{X}^TY.$$

A major source of bias occurring in Ridge estimation when $p > n$ comes from the fact that the Ridge estimator is estimating a projected parameter

$$\theta^0 = P_R\beta^0, \quad P_R = \mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^-\mathbf{X},$$

where $(\mathbf{X}\mathbf{X}^T)^-$ denotes a generalized inverse of $\mathbf{X}\mathbf{X}^T$. The minor bias for θ^0 then satisfies:

$$\max_j|\mathbb{E}[\hat{\beta}_{\text{Ridge};j}] - \theta_j^0| \le \lambda\|\theta^0\|_2\lambda_{\min\neq 0}(\hat{\Sigma})^{-1},$$

where $\lambda_{\min\neq 0}(\hat{\Sigma})$ denotes the minimal non-zero eigenvalue of $\hat{\Sigma}$ (Shao and Deng, 2012). The quantity can be made small by choosing λ small. Therefore, for $\lambda \searrow 0^+$ and assuming Gaussian errors we have that

$$\sigma_\varepsilon^{-1}(\hat{\beta}_{\text{Ridge}} - \theta^0) \approx W, \quad W \sim \mathcal{N}_p(0,\Omega_R), \tag{2.9}$$

where $\Omega_R = (\hat{\Sigma} + \lambda)^{-1}\hat{\Sigma}(\hat{\Sigma} + \lambda)^{-1}$. Since

$$\frac{\theta^0}{P_{R;jj}} = \beta_j^0 + \sum_{k \neq j} \frac{P_{R;jk}}{P_{R;jj}} \beta_k^0,$$

the major bias for β_j^0 can be estimated and corrected with

$$\sum_{k \neq j} \frac{P_{R;jk}}{P_{R;jj}} \hat{\beta}_k,$$

where $\hat{\beta}$ is the ordinary Lasso. Thus, we construct a bias corrected Ridge estimator, which addresses the potentially substantial difference between θ^0 and the target β^0:

$$\hat{b}_{R;j} = \frac{\hat{\beta}_{\text{Ridge};j}}{P_{R;jj}} - \sum_{k \neq j} \frac{P_{R;jk}}{P_{R;jj}} \hat{\beta}_k, \;\; j = 1, \ldots, p. \tag{2.10}$$

Based on (2.9) we derive in Appendix A.1 that

$$\begin{aligned} &\sigma_\varepsilon^{-1} \Omega_{R;jj}^{-1/2} (\hat{b}_{R;j} - \beta_j^0) \approx \Omega_{R;jj}^{-1/2} W_j + \sigma_\varepsilon^{-1} \Omega_{R;jj}^{-1/2} \Delta_{R;j}, \quad W \sim \mathcal{N}_p(0, \Omega_R), \\ &|\Delta_{R;j}| \leq \Delta_{R\text{bound};j} := \max_{k \neq j} \left| \frac{P_{R;jk}}{P_{R;jj}} \right| (\log(p)/n)^{1/2 - \xi}, \end{aligned} \tag{2.11}$$

with the typical choice $\xi = 0.05$. Sufficient conditions for deriving (2.11) are assumption (B1) and that the sparsity satisfies $s_0 = O((n/\log(p))^{\xi})$ for ξ as above.

Unlike as in Fact 2, the term $\Delta_{R;j}$ is typically not negligible and we correct the Gaussian part in (2.11) by the upper bound $\Delta_{R\text{bound};j}$. For example, for testing $H_{0,j} : \beta_j^0 = 0$ we use the upper bound for the p-value

$$2(1 - \Phi(\sigma_\varepsilon^{-1} \Omega_{R;jj}^{-1/2} |P_{R;jj}| (|\hat{b}_{R;j}| - \Delta_{R\text{bound};j})_+)).$$

Similarly, for two-sided confidence intervals with coverage $1 - \alpha$ we use

$$\begin{aligned} &[\hat{b}_{R;j} - c_j, \hat{b}_{R;j} + c_j], \\ &c_j = \Delta_{R\text{bound};j} + \sigma_\varepsilon \Omega_{R;jj}^{1/2} / |P_{R;jj}| \Phi^{-1}(1 - \alpha/2). \end{aligned}$$

For testing a group hypothesis for $G \subseteq \{1, \ldots, p\}$, $H_{0,G} : \beta_j^0 = 0$ for all $j \in G$, we can proceed similarly as at the end of Section 2.1.2: under the null-hypotheses $H_{0,G}$, the statistic $\sigma_\varepsilon^{-1} \max_{j \in G} \Omega_{R;jj}^{-1/2} |\hat{b}_{R;j}|$ has a distribution which is approximately stochastically upper bounded by

$$\max_{j \in G} (\Omega_{R;jj}^{-1/2} |W_j| + |\Delta_{R;j}|),$$

see also (2.11). When invoking an upper bound for $\Delta_{R\text{bound};j} \geq |\Delta_{R;j}|$ as in (2.11), we can easily simulate this distribution from dependent Gaussian random variables which in turn can be used to construct a p-value: we refer for further details to Bühlmann (2013).

2.1.4 Additional issues: estimation of the error variance and multiple testing correction Unlike the Multi sample-splitting procedure in Section 2.1.1, the de-sparsified Lasso and Ridge projection method outlined in Sections 2.1.2–2.1.3 require to plug-in an estimate of σ_ε and to adjust for multiple testing. The scaled Lasso (Sun and Zhang, 2012) leads to a consistent estimate of the error variance: it is a fully automatic method which does not need any specification of a tuning parameter. In Reid et al. (2013), an empirical comparison of various estimators suggests that the estimator based on residual sum of squares of a cross-validated Lasso solution often yields good finite-sample performance.

Regarding the adjustment when doing many test for individual regression parameters or groups thereof, one can use any valid standard method to correct the p-values from the de-sparsified Lasso or Ridge projection method. The prime example are the Bonferroni-Holm procedure for controlling the familywise error rate or the method from Benjamini and Yekutieli (2001) for controlling the false discovery rate. An approach for familywise error control which explicitly takes the dependence among the multiple hypotheses is proposed in Bühlmann (2013), based on simulations for dependent Gaussian random variables.

2.1.5 Conceptual differences between the methods We briefly outline here conceptual differences while Section 2.4 presents empirical results.

The Multi sample-splitting method is very generic and in the spirit of Breiman's appeal for stability (Breiman, 1996a,b), it enjoys some kind of stability due to multiple sample splits and aggregation. The disadvantage is that, in the worst case, the method needs a beta-min or a weaker zonal assumption on the underlying regression parameters: this is somewhat unpleasant since a significance test should *find out* whether a regression coefficient is sufficiently large or not.

Both the de-sparsified Lasso and Ridge projection procedures do not make any assumption on the underlying regression coefficient except sparsity. The former is most powerful and asymptotically optimal if the design were generated from a population distribution whose inverse covariance matrix is sparse. Furthermore, the convergence is uniform over all sparse regression vectors and hence, the method yields honest confidence regions or tests. The Ridge projection method does not require any assumption on the fixed design but does not reach the asymptotic Cramer-Rao efficiency bound. The construction with the additional correction term in (A.3) leads to reliable type I error control at the cost of power.

In terms of computation, the Multi sample-splitting and Ridge projection method are substantially less demanding than the de-sparsified Lasso.

2.2 `hdi` for linear models

In the R-package `hdi`, available on R-Forge (Meier et al., 2014) , we provide implementations for the Multi sample-splitting, the Ridge projection and the de-sparsified Lasso method. Using the R functions is straightforward:

```
> outMssplit <- multi.split(x = x, y = y)
> outRidge   <- ridge.proj(x = x, y = y)
> outLasso   <- lasso.proj(x = x, y = y)
```

For users that are very familiar with the procedures, we provide flexible options. For example, we can easily use an alternative model selection or another "classical" fitting procedure using the arguments `model.selector` and `classical.fit` in `multi.split`. The default options should be satisfactory for standard usage.

All procedures return p-values and confidence intervals. The Ridge and de-sparsified Lasso methods return both single testing p-values as well as multiple testing corrected p-values unlike the Multi sample-splitting procedure which only returns multiple testing corrected p-values. The confidence intervals are for individual parameters only (corresponding to single hypothesis testing).

The single-testing p-values and the multiple testing corrected p-values are extracted from the fit as follows

```
> outRidge$pval
> outRidge$pval.corr
```

By default, we correct for controlling the familywise error rate for the p-values `pval.corr`.

Confidence intervals are acquired through the usual `confint` interface. Below we extract the 95 % confidence intervals for those p-values that are smaller than `0.05`

```
> confint(outMssplit, parm = which(outMssplit$pval.corr <= 0.05), level = 0.95)
```

Due to the fact that the de-sparsified Lasso method is quite computationally intensive, we provide the option to parallelize the method on a user-specified number of cores.

We refer to the manual of the package for more detailed information.

2.3 Other methods

Recently, a few other procedures have been suggested for construction of p-values and confidence intervals.

Residual-type bootstrap approaches are proposed and analyzed in Chatterjee and Lahiri (2013) and Liu and Yu (2013). A problem with these approaches is the non-uniform convergence to a limiting distribution and exposure to the super-efficiency phenomenon: that is, if the true parameter equals zero, a confidence region might be the singleton $\{0\}$ (due to finite amount of bootstrap resampling) while for non-zero true parameter values, the coverage might be very poor or big length of the confidence interval.

The covariance test (Lockhart et al., 2014) is another proposal which relies on the solution path of the Lasso and provides p-values for conditional tests that all relevant variables enter the Lasso solution path first.

In Javanmard and Montanari (2013), a procedure was proposed that is very similar to the one described in Section 2.1.2, with the only difference being that Z is picked as the solution of a convex program, rather than using the Lasso. The method is aiming to relax the sparsity assumption (B3) for the design.

A conservative *Group-bound* method which needs no regularity assumption for the design, e.g. no compatibility assumption (A.1), has been proposed by Meinshausen (2013). The method has the capacity to automatically determine whether a regression coefficient is identifiable or not, and this makes the procedure very robust against ill-posed designs. The main motivation of the method is in terms of testing groups of correlated variables, and we discuss it in more detail in Section 4.1.

While all the methods mentioned above are considered in a comparative simulation study in Section 2.4, we refer to another procedure which we do not include in that study: based on stability selection (Meinshausen and Bühlmann, 2010), Shah and Samworth (2013) propose a version which leads to p-values for testing individual regression parameters.

2.4 A broad comparison

We compare a variety of methods on the basis of multiple testing corrected p-values and single testing confidence intervals. The methods we look at are the multiple sample-splitting method *MS-Split* (Section 2.1.1), the de-sparsified Lasso method *Lasso-Pro* (Section 2.1.2), the Ridge projection method *Ridge* (Section 2.1.3), the covariance test *Covtest* (Section 2.3), the method by Javanmard and Montanari *Jm2013* (Section 2.3) and the two bootstrap procedures mentioned in Section 2.3 (*Res-Boot* corresponds to Chatterjee and Lahiri (2013) and *liuyu* to Liu and Yu (2013)).

2.4.1 Specific details for the methods For the estimation of the error variance, for the Ridge projection or the de-sparsified Lasso method, the scaled lasso is used as mentioned in Section 2.1.4.

For the choice of tuning parameters for the nodewise Lasso regressions (discussed in Section 2.1.2), we look at the two alternatives of using either cross-validation or our more favored alternative procedure (denoted by Z&Z) discussed in the Appendix A.1.

We don't look at the bootstrap procedures in connection with multiple testing adjustment due to the fact that the required number of bootstrap samples grows out of proportion to go far enough in the tails of the distribution; some additional importance sampling might help to address such issues.

Regarding the covariance test, the procedure doesn't directly provide p-values for the hypotheses we are interested in. For the sake of comparison though, we use the interpretation as in Bühlmann et al. (2014b). This interpretation doesn't have a theoretical reasoning behind it and functions more as a heuristic. Thus, the results of the covariance test procedure should be interpreted with caution.

For the method *Jm2013*, we used our own implementation instead of the code provided by the authors. The reason for this is that we had already implemented our own version when we discovered that code was available and our own version was (orders of magnitude) better in terms of error control. Posed with the dilemma of fair comparison, we stuck to the best performing alternative.

2.4.2 Data used For the empirical results, simulated design matrices as well as design matrices from real data are used. The simulated design matrices are generated $\sim \mathcal{N}_p(0, \Sigma)$ with covariance matrix Σ of the following three types:

$$\begin{aligned} \text{Toeplitz:} \quad & \Sigma_{j,k} = 0.9^{|j-k|}, \\ \text{Exp.decay:} \quad & (\Sigma^{-1})_{j,k} = 0.4^{|j-k|/5}, \\ \text{Equi.corr:} \quad & \Sigma_{j,k} \equiv 0.8 \text{ for all } j \neq k,\ \Sigma_{j,j} \equiv 1 \text{ for all } j. \end{aligned}$$

The sample size and dimension are fixed at $n = 100$ and $p = 500$, respectively. The design matrix RealX from real gene expression data of Bacillus Subtilis ($n = 71, p = 4088$) was kindly provided by DSM (Switzerland) and is publicly available (Bühlmann et al., 2014a). To make the problem somewhat comparable in difficulty to the simulated designs, the number of variables is reduced to $p = 500$ by taking the variables with highest empirical variance.

The cardinality of the active set is picked to be one of two levels $s_0 \in \{3, 15\}$. For each of the active set sizes, we look at 6 different ways of picking the sizes of the non-zero coefficients:

Randomly generated : U(0,2), U(0,4), U(-2,2),
A fixed value : 1, 2 or 10.

The positions of the non-zero coefficients as columns of the design $\boldsymbol{X}$ are picked at random. Results where the non-zero coefficients were positioned to be the first s_0 columns of $\boldsymbol{X}$ can be found in Appendix A.3.

Once we have the design matrix $\boldsymbol{X}$ and coefficient vector β^0, the responses Y are generated according to the linear model equation with $\varepsilon \sim \mathcal{N}(0, 1)$.

2.4.3 P-values We investigate multiple testing corrected p-values for two-sided testing of the null hypotheses $H_{0,j} : \beta_j^0 = 0$ for $j = 1, \dots, p$. We report the power and the familywise error rate (FWER) for each method:

$$\text{Power} = \sum_{j \in S_0} \mathbb{P}[H_{0,j} \text{ is rejected}]/s_0,$$
$$\text{FWER} = \mathbb{P}[\exists j \in S_0^c : H_{0,j} \text{ is rejected}].$$

We calculate empirical versions of these quantities based on fitting 100 simulated responses Y coming from newly generated ε.

For every design type, active set size and coefficient type combination we obtain 50 datapoints of the empirical versions of "Power" and "FWER", from 50 independent simulations. Thereby, each datapoint has a newly generated X, β^0 (if not fixed) and active set positions $S_0 \in \{1 \dots p\}$: thus the 50 datapoints indicate the variability with respect to the three quantities in the data generation (for the same covariance model of the design, the same model for the regression parameter and its active set positions). The datapoints are grouped in plots by design type and active set size.

We also report the average number of false positives `AVG(V)` over all data points per method next to the FWER plot.

The results, illustrating the performance for various methods, can be found in Figures 2, 3, 4 and 5.

2.4.4 Confidence intervals We investigate confidence intervals for the one particular setup of Toeplitz design, active set size $s_0 = 3$ and coefficients $\beta_j^0 \sim U[0, 2]$ $(j \in S_0)$. The active set positions are chosen to be the first s_0 columns of $\boldsymbol{X}$. The results we show will correspond to a single datapoint in the p-value results.

In Figure 6, 100 confidence intervals are plotted for each coefficient for each method. These confidence intervals are the results of fitting 100 different responses Y resulting from newly generated ε error terms.

For the Multi sample-splitting method from Section 2.1.1, if a variable did not get selected often enough in the sample splits, there is not enough information to draw a confidence interval for it. This is represented in the plot by only drawing confidence intervals when this wasn't the case. If the (uncheckable) beta-min condition (2.3) would be fulfilled, we would know that those confidence intervals cover zero.

For the bootstrapping methods, an invisible confidence interval is the result of the coefficient being set to zero in all bootstrap iterations.

2.4.5 Summarizing the empirical results As a first observation, the impact of the sparsity of the problem on performance cannot be denied. The power clearly gets worse for $s_0 = 15$ for the Toeplitz and Exp.decay setups. The FWER becomes too high for quite a few methods for $s_0 = 15$ in the cases of Equi.corr and RealX.

For the sparsity $s_0 = 3$, the Ridge projection method manages to control the FWER as desired for all setups. In the case of $s_0 = 15$, it is the Multi sample-splitting method that comes out best in comparison to the other methods. Generally speaking, good error control tends to be associated with a lower power which is not too surprising since we are dealing with the trade-off between type I and type II errors. The de-sparsified Lasso method turns out to be a less conservative alternative with not perfect but reasonable FWER control as long as the problem is sparse enough ($s_0 = 3$). The method has a slightly too high FWER for the Equi.corr and RealX setups but FWER around

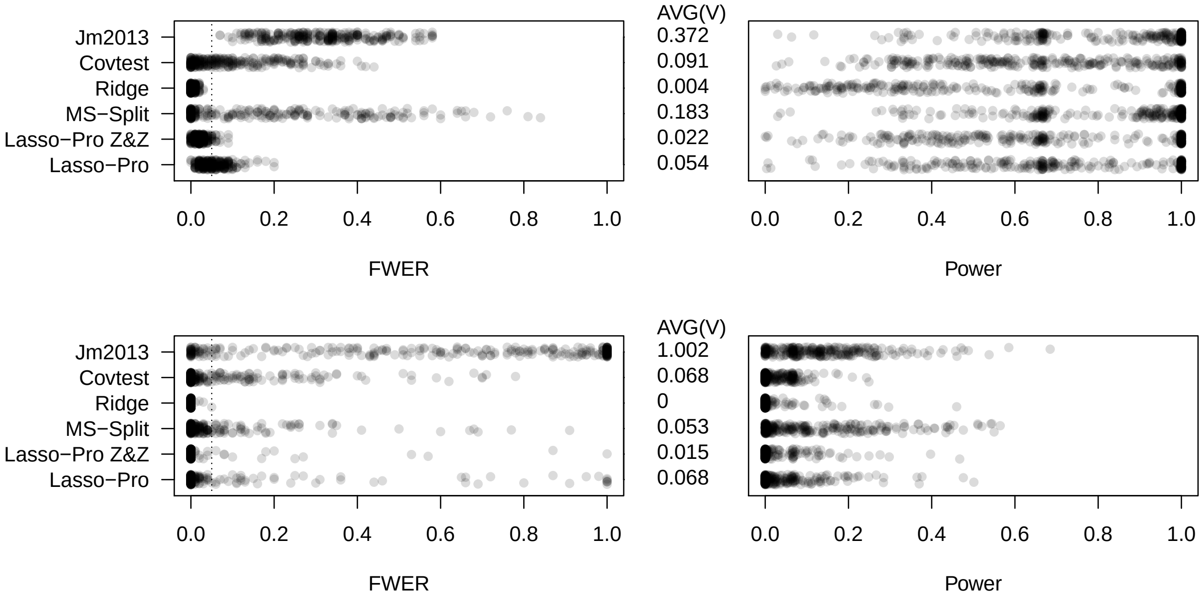

Fig 2. *Familywise error rate (FWER), average number of false positive (AVG(V)) and power for multiple testing based on various methods for a linear model. The desired control level for the FWER is $\alpha = 0.05$. The average number of false positives AVG(V) for each method is shown in the middle. The design matrix is of type **Toeplitz**, and the active set size being $s_0 = 3$ (top) and $s_0 = 15$ (bottom).*

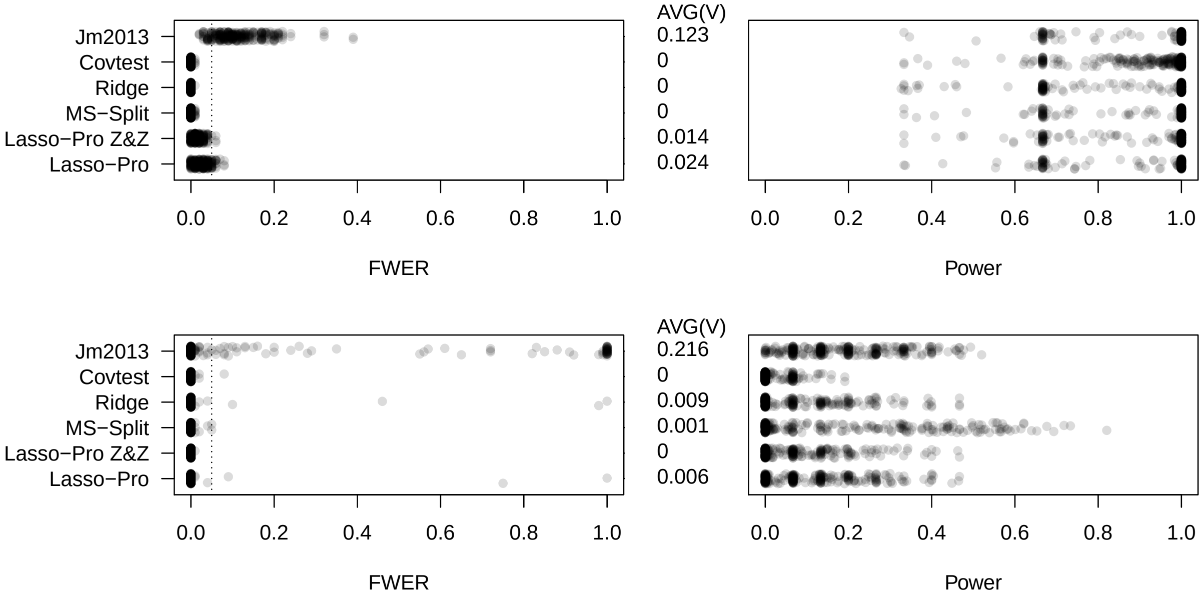

Fig 3. *See caption of Figure 2 with the only difference being the type of design matrix. In this plot the design matrix type is **Exp.decay**.*

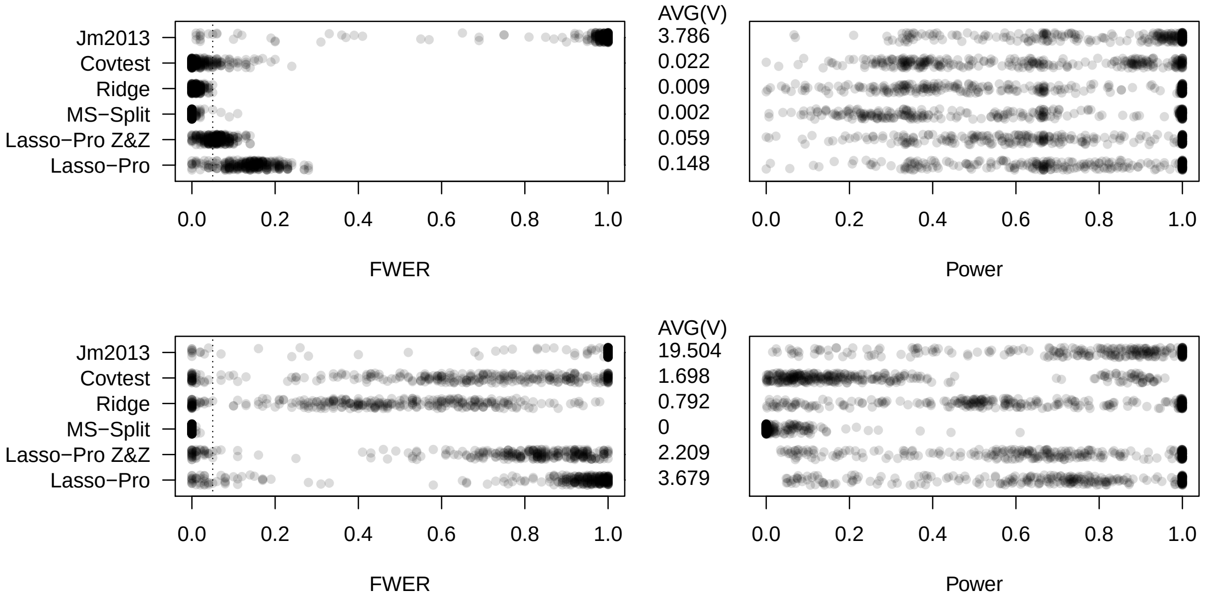

FIG 4. *See caption of Figure 2 with the only difference being the type of design matrix. In this plot, the design matrix type is* ***Equi.corr****.*

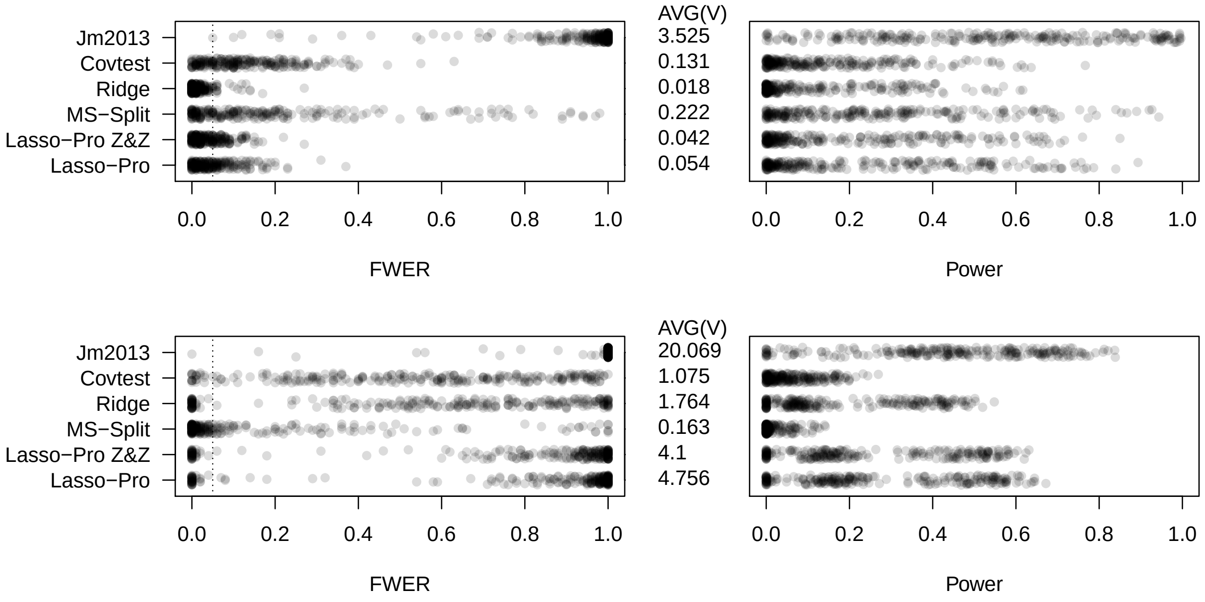

FIG 5. *See caption of Figure 2 with the only difference being the type of design matrix. In this plot, the design matrix type is* ***RealX****.*

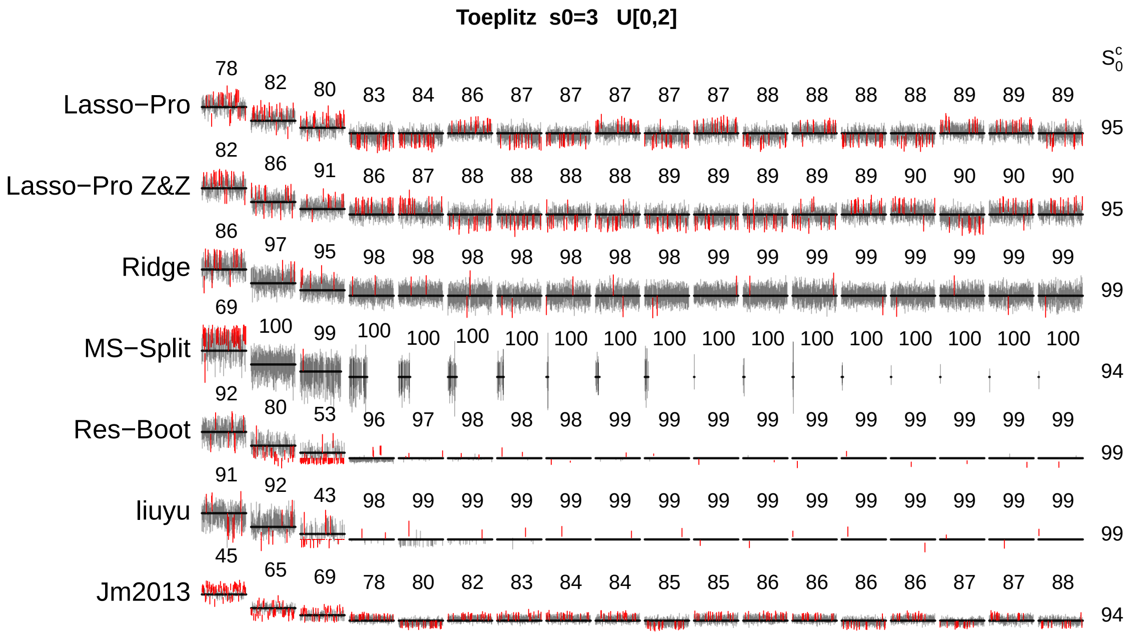

FIG 6. *100 confidence intervals and their empirical coverage of the true coefficients for a linear model. Black confidence intervals cover the truth, red confidence intervals do not. The 15 zero coefficients shown are those with the worst coverage for that method. The black numbers are the coverage rates in percentage. The numbers in the last column are the average coverage rates of the zero coefficients S_0^c. The confidence intervals for the* MS-Split *method are only drawn if the variable was selected in the sample-splitting procedure.*

0.05 for Toeplitz and Exp.decay designs. Doing the Z&Z tuning procedure helps the error control as can be seen most clearly in the Equi.corr setup.

The results for the simulations where the positions for the non-zero coefficients were not randomly chosen, presented in Appendix A.3, largely give the same picture. In comparison to the results presented before, the Toeplitz setup is easier while the Exp.decay setup is more challenging. The Equi.corr results are very similar to the ones from before which is to be expected from the covariance structure.

Looking into the confidence interval results, it is clear that the confidence intervals of the Multi sample-splitting method and the Ridge projection method are wider than the rest. For the bootstrapping methods, the super-efficiency phenomenon mentioned in Section 2.3 is visible. Important to note here is that the smallest non zero coefficient, the third column, has very poor coverage from these methods.

We can conclude that the coverage of the zero coefficients is decent for all methods and that the coverage of the non-zero coefficients is in line with the error rates for the p-values.

Confidence interval results for many other setup combinations are provided in the appendix A.4. The observations are to a large extent the same.

3. GENERALIZED LINEAR MODELS

Consider a generalized linear model

$$\begin{gathered} Y_1, \dots, Y_n \text{ independent}, \\ g(\mathbb{E}[Y_i|X_i = x]) = \mu^0 + \sum_{j=1}^{p} \beta_j^0 x^{(j)}, \end{gathered}$$

where $g(\cdot)$ is a real-valued, known link function. As before, the goal is to construct confidence intervals and statistical tests for the unknown parameters $\beta_1^0, \dots, \beta_p^0$, and maybe μ^0 as well.

3.1 Methods

The Multi sample-splitting method can be modified for GLMs in an obvious way: the variable screening step using the first half of the data can be based on ℓ_1-norm regularized MLE, and p-values and confidence intervals using the second half of the sample are constructed from the asymptotic distribution of the (low-dimensional) MLE. Multiple testing correction and aggregation of the p-values from multiple sample splits are done exactly as for linear models in Section 2.1.1.

A de-sparsified Lasso estimator for GLMs can be constructed as follows (van de Geer et al., 2014). The ℓ_1-norm regularized MLE $\hat{\theta}$ for the parameters $\theta^0 = (\mu^0, \beta^0)$ is de-sparsified with a method based on the Karush-Kuhn-Tucker (KKT) conditions for $\hat{\theta}$, leading to an estimator with an asymptotic Gaussian distribution. The Gaussian distribution can then be used to construct confidence intervals and hypothesis tests.

3.2 Weighted squared error approach

The problem can be simplified in such a way that we can apply the approaches for the linear model from Section 2. This can be done for all types of Generalized linear models (as shown in Appendix A.5), but we restrict ourselves in this Section to the specific case of logistic regression. Logistic regression is usually fitted by applying the iteratively reweighted least squares (IRLS) algorithm where at every iteration one solves a weighted least squares problem (Hastie et al., 2009).

The idea is now to apply standard l1-penalized fitting of the model, build up the weighted least squares problem at the l1-solution and then apply our linear model methods on this problem.

We use the notation $\hat{\pi}_i, i = 1, \dots, n$ for the estimated probability of the binary outcome. $\hat{\pi}$ is the vector of these probabilities.

From Hastie et al. (2009), the adjusted response variable becomes

$$Y_{adj} = \boldsymbol{X}\hat{\beta} + \boldsymbol{W}^{-1}(Y - \hat{\pi}),$$

and the weighted least squares problem is

$$\hat{\beta}_{new} = \text{argmin}_\beta (Y_{adj} - \boldsymbol{X}\beta)^T \boldsymbol{W} (Y_{adj} - \boldsymbol{X}\beta),$$

with weights

$$\boldsymbol{W} = \begin{pmatrix} \hat{\pi}_1(1-\hat{\pi}_1) & 0 & \dots & 0 \\ 0 & \hat{\pi}_2(1-\hat{\pi}_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & \hat{\pi}_n(1-\hat{\pi}_n) \end{pmatrix}.$$

We rewrite, $Y_w = \sqrt{\boldsymbol{W}} Y_{adj}$ and $X_w = \sqrt{\boldsymbol{W}} \boldsymbol{X}$ to get

$$\hat{\beta}_{new} = \text{argmin}_\beta (Y_w - \boldsymbol{X}_w\beta)^T (Y_w - \boldsymbol{X}_w\beta).$$

The linear model methods can now be applied to Y_w and $\boldsymbol{X}_w$. Thereby, the estimate $\hat{\sigma}_\varepsilon$ has to be set to the value 1. We note that in the low-dimensional case, the resulting p-values (with unregularized residuals Z_j) are very similar to the p-values provided by the standard `R`-function `glm`.

3.3 Small empirical comparison

We provide a small empirical comparison of the methods mentioned in 3.1 and 3.2. When applying the linear model procedures, we use the naming from 2.4. The new GLM-specific methods from 3.1 are referred to by their linear model names with a capital G added to them.

For simulating the data, we use a subset of the variations presented in Section 2.4.2. We only look at Toeplitz and Equi.corr and an active set size of $s_0 = 3$. The number of variables is fixed at $p = 500$ but the sample size is varied $n \in \{100, 200, 400\}$. The coefficients were randomly generated

$$\text{Randomly generated}: \quad \text{U(0,1), U(0,2), U(0,4).}$$

The non-zero coefficient positions are chosen randomly in one case and fixed as the first s_0 columns of $\boldsymbol{X}$ in the other.

For every combination (of type of design, type of coefficients, sample size and coefficient positions), 100 responses Y are simulated to calculate empirical versions of the "Power" and "FWER" described in Section 2.4.3. In contrast to the p-value results from Section 2.4.3, there is only one resulting datapoint per setup combination (i.e., no additional replication with new random covariates, random coefficients and random active set). For each method, there are 18 datapoints, corresponding to 18 settings, in each plot. The results can be found in Figure 7.

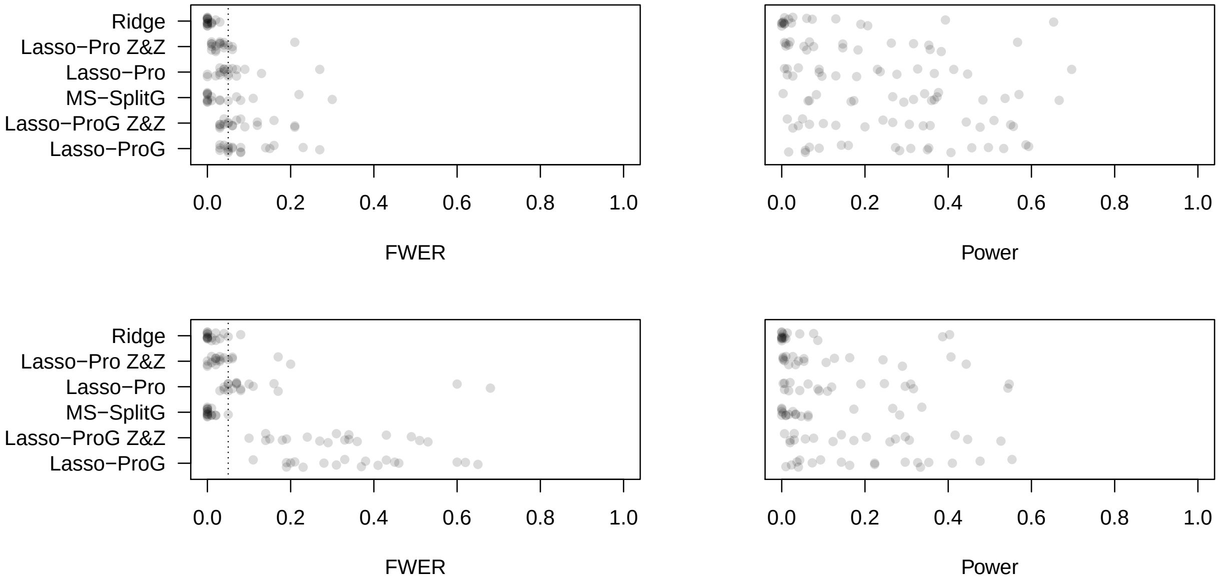

Fig 7. *Familywise error rate (FWER) and power for multiple testing based on various methods for logistic regression. The desired control level for the FWER is $\alpha = 0.05$. The design matrix is of type* ***Toeplitz*** *in the top plot and* ***Equi.corr*** *in the bottom plot. If the method name contains a capital* `G` *it is the modified glm version, otherwise the linear model methods are used using the weighted squared error approach.*

Both the modified GLM methods as well as the weighted squared error approach works adequately. The Equi.corr setup does prove to be challenging for *Lasso-ProG*.

3.4 hdi for generalized linear models

In the hdi R-package (Meier et al., 2014) we also provide the option to use the Ridge projection method and the de-sparsified Lasso method with the weighted squared error approach.

We provide the option to specify the family of the response Y as done in the R-package glmnet.

```
> outRidge <- ridge.proj(x = x, y = y, family = "binomial")
> outLasso <- lasso.proj(x = x, y = y, family = "binomial")
```

P-values and confidence intervals are extracted in the exact same way as for the linear model case, see Section 2.2.

4. HIERARCHICAL INFERENCE IN PRESENCE OF HIGHLY CORRELATED VARIABLES

The previous sections and methods assume in some form or another that the effects are strong enough to enable accurate estimation of the contribution of *individual variables*.

Variables are often highly correlated for high-dimensional data. Working with a small sample size, it is impossible to attribute any effect to an individual variable if the correlation between a block of variables is too high. Confidence intervals for individual variables are then very wide and uninformative. Asking for confidence intervals for individual variables thus leads to poor power of all procedures considered so far. Perhaps even worse, under high correlation between variables the coverage of some procedures will also be unreliable as the necessary conditions for correct coverage (such as the compatibility assumption) are violated.

In such a scenario, the individual effects are not granular enough to be resolved. However, it might yet still be possible to attribute an effect to a group of variables. The groups can arise naturally due to a specific structure of the problem, such as in applications of the *group Lasso* (Yuan and Lin, 2006).

Perhaps more often, the groups are derived via hierarchical clustering (Hartigan, 1975), using the correlation structure or some other distance between the variables. The main idea is as follows. A hierarchy $\mathcal{T}$ is a set of clusters or groups $\{\mathcal{C}_k;\ k\}$ with $\mathcal{C}_k \subseteq \{1, \ldots, p\}$. The root node (cluster) contains all variables $\{1, \ldots, p\}$. For any two clusters $\mathcal{C}_k, \mathcal{C}_\ell$, either one cluster is a subset of the other or they have an empty intersection. Usually, a hierarchical clustering has an additional notion of a level such that on each level, the corresponding clusters build a partition of $\{1, \ldots, p\}$. We consider a hierarchy $\mathcal{T}$ and first test the root node cluster $\mathcal{C}_0 = \{1, \ldots, p\}$ with hypothesis $H_{0,\mathcal{C}_0}:\ \beta_1 = \beta_2 = \ldots = \beta_p = 0$. If this hypothesis is rejected, we test the next clusters $\mathcal{C}_k$ in the hierarchy (all clusters whose supersets are the root node cluster $\mathcal{C}_0$ only): the corresponding cluster hypotheses are $H_{0,\mathcal{C}_k} : \beta_j = 0$ for all $j \in \mathcal{C}_k$. For the hypotheses which can be rejected, we consider all smaller clusters whose only supersets are clusters which have been rejected by the method before, and we continue to go down the tree hierarchy until no more cluster hypothesis can be rejected.

With the hierarchical scheme in place, we still need a test for the null hypothesis $H_{0,\mathcal{C}}$ of a cluster of variables. The tests have different properties. For example, whether a multiplicity adjustment is necessary will depend on the chosen test. We will describe below some methods that are useful for testing the effect of a group of variables and which can be used in such a hierarchical approach. The nice and interesting feature of the procedures is that they adapt automatically to the level of the hierarchical tree: if a signal of a small cluster of variables is strong, and if that cluster is sufficiently uncorrelated from all other variables or clusters, the cluster will be detected as significant. Vice versa, if the signal is weak or if the cluster has too high correlation with other variables or clusters, the cluster will not become significant. For example, a single variable cannot be detected as significant if it has too much correlation to other variables or clusters.

4.1 Group-bound confidence intervals without design assumptions

The *Group-bound* proposed in (Meinshausen, 2013) gives confidence intervals for the ℓ_1-norm $\|\beta^0_{\mathcal{C}_k}\|_1$ of a group $\mathcal{C}_k \subseteq \{1, \ldots, p\}$ of variables. If the lower-bound of the $1-\alpha$ confidence interval is larger than 0, then the null hypothesis $\beta^0_{\mathcal{C}_k} \equiv 0$ can be rejected for this group. The method combines a few properties:

(i) The confidence intervals are valid without an assumption like the compatibility condition (A.1).

(ii) The test is hierarchical. If a set of variables can be rejected, all supersets will also be rejected. And vice versa: if a group of variables cannot be rejected, none of its subsets can be rejected.

(iii) The estimation accuracy has an optimal detection rate under the so-called group effect compatibility condition, which is weaker than the compatibility condition necessary to detect the effect of individual variables.

(iv) The power of the test is unaffected by adding highly or even perfectly correlated variables in $\mathcal{C}_k$ to the group. The compatibility condition would fail to yield a non-trivial bound but the group effect compatibility condition is unaffected by the addition of perfectly correlated variables to a group.

The price to pay for the assumption-free nature of the bound is a weaker power than with previously discussed approaches when the goal is to detect the effect of individual variables. However, for groups of highly correlated variables, the approach can be much more powerful than simply testing all variables in the group.

We remark that previously developed tests can be adapted to the context of hierarchical testing of groups with hierarchical adjustment for familywise error control (Meinshausen, 2008); for the Multi sample-splitting method, this is described next.

4.2 Hierarchical multi sample-splitting

The Multi sample-splitting method (Section 2.1.1) can be adapted to the context of hierarchical testing of groups by using hierarchical adjustment of familywise error control (Meinshausen, 2008). When testing a cluster hypotheses $H_{0,\mathcal{C}}$, one can use a modified form of the partial F-test for high-dimensional settings; and the multiple testing adjustment due to the multiple cluster hypotheses considered can be taken care of a hierarchical adjustment scheme proposed in Meinshausen (2008). A detailed description of the method, denoted here by *Hier. MS-Split*, together with theoretical guarantees is given in Mandozzi and Bühlmann (2013).

4.3 Simultaneous inference with the Ridge or de-sparsified Lasso method

Simultaneous inference for all possible groups can be achieved by considering p-values P_j of individual hypotheses $H_{0,j}:\ \beta^0_j = 0$ $(j = 1, \ldots, p)$ and adjusting them for simultaneous coverage, namely $P_{\text{adjusted},j} = P_j \cdot p$. The individual p-values P_j can be obtained by the Ridge or de-sparsified Lasso method in Section 2.

We can then test any group hypothesis $H_{0,G}:\ \beta^0_j = 0$ for all $j \in G$ by simply looking whether $\min_{j \in G} P_{\text{adjust},j} \leq \alpha$, and we can consider as many group hypotheses as we want without any further multiple testing adjustment.

4.4 Illustrations

A semi-real data example is shown in Figure 8, where the predictor variables are taken from the Riboflavin dataset (Bühlmann et al., 2014a) $(n = 71, p = 4088)$ and the coefficient vector is taken to have entries 0, except for 2 clusters of highly correlated variables. In example 1, the clusters both have size 3 with nonzero coefficient sizes equal to 1 for all the variables in the clusters and Gaussian noise level $\sigma = 0.1$. In example 2, the clusters are bigger and have different sizes 11 and 21, the coefficient sizes for all the variables in the clusters is again 1 but the Gaussian noise level here is chosen to be $\sigma = 0.5$.

In the first example, 6 out of the 6 relevant variables are discovered as individually significant by the *Lasso-Pro*, *Ridge* and *MS-Split* methods (as outlined in Sections 2.1.1–2.1.2), after adjusting for multiplicity.

In the second example, the methods cannot reject the single variables individually any longer. The results for the *Group-bound* estimator are shown in the right column. The *Group-bound* can reject a group of 4 and 31 variables in the first example, each contains a true cluster of 3 variables. The method can also detect a group of 2 variables (a subset of the cluster of 4) which contains 2 out of the 3 highly correlated variables. In the second example, a group of 34 variables is rejected with the *Group-bound* estimator, containing 16 of the group of 21 important variables. The smallest group of variables containing the cluster of 21 that the method can detect, is of size 360. It can thus be detected that the variables jointly have a substantial effect even though the null hypothesis cannot be rejected for any variable individually. The Hierarchical multi sample-splitting method (outlined in Section 4.2) manages to detect the same clusters as the *Group-bound* method. It even goes one step further by detecting a smaller subcluster.

We also consider the following simulation model. The design matrix was chosen to be Gaussian with mean zero and population block-diagonal covariance matrix Σ with blocks of dimension 20×20 with equi-correlated off-diagonal entries ρ. We chose the dimension $p = 500$, sample size $n = 100$ and Gaussian noise with level $\sigma = 1$. There were only 3 nonzero coefficients chosen with three different signal levels (realizations of the β's) from Uniform distributions U[0,2], U[0,4] and U[0,8]. Aside from varying signal level, we studied two cases where either all the nonzero coefficients were contained in one single highly correlated block or each of the active variables were in a different block. Figures 9 and 10 show the coverage and power as a function of the correlation ρ between variables in another simulation model. The type of design matrix was chosen to be such that the population covariance matrix Σ is a block-diagonal matrix with blocks of dimension 20×20 being of the same type as Σ for Equi.corr (see Section 2.4.2) with off-diagonal ρ instead of 0.8. The dimensions of the problem were chosen to be $p = 500$ number of variables, $n = 100$ number of samples and noise level $\sigma = 1$. There were only 3 nonzero coefficients chosen with three different signal levels U[0,2], U[0,4] and U[0,8] being used for the simulations. Aside from varying signal level, we studied the two cases where in one case all the nonzero coefficients were contained in one single highly correlated block and in the other case each of those variables was in a different block. We look at 3 different measures of power. One can define the power as the fraction of the 100 repeated simulations that the method managed to reject the group of all variables $G = 1, \ldots, p$. This is shown at the top in figure 9. Alternatively, one can look at the rejection rate of the hypothesis for the group G that contains all variables in the highly correlated blocks that contain a variable from S_0. This is the plot at the bottom in figure 9. Finally, one can look at the rejection rate of the hypothesis where the group G contains only the variables in S_0 (of size 3 in this case). The type I error we define to be the fraction of the simulations in which the method rejected the group hypothesis H_{0,S_0^c} where all regression coefficients are equal to zero. These last two are presented in Figure 10.

The power of the Ridge-based method (Bühlmann, 2013) drops substantially for high correlations. The power of the *Group-bound* stays close to 1 at the level of the highly correlated groups (Block-power) and above (Power $G = 1 \ldots p$) throughout the entire range of correlation values. The *Lasso-*

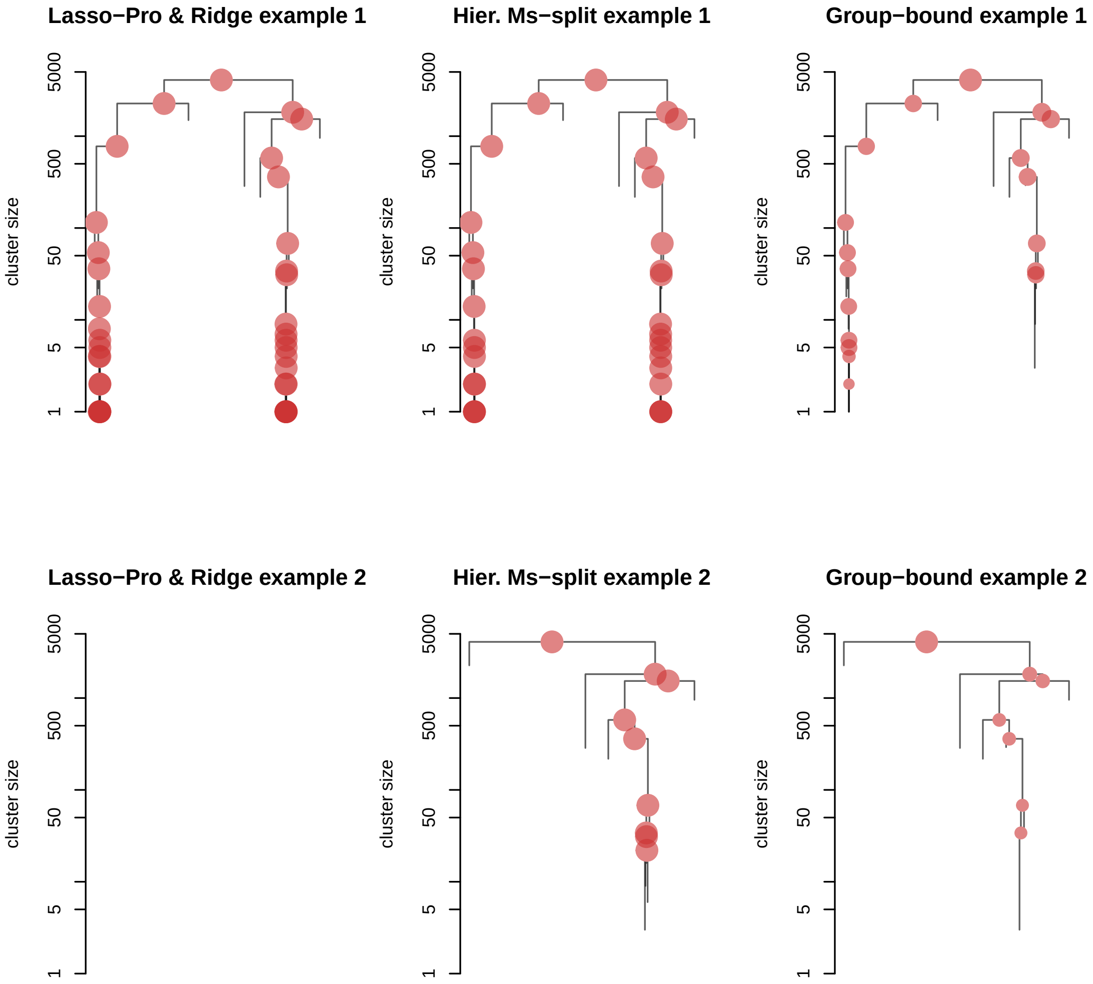

FIG 8. *A visualization of the hierarchical testing scheme. Rejected clusters are shown with a red circle, and the size of the cluster is on the vertical axis. The top row is the first example mentioned in the text, the bottom row corresponds to the second example with a lower signal-to-noise ratio and less sparsity. In the left column, the outcomes when using the de-sparsified Lasso-pro and Ridge-based methods are shown, which find the individually important variables in the first example and nothing in the second example.The right column shows the outcome when using the* Group-bound *method, which find two significant clusters in the first example (containing the true set of important variables) and a single significant cluster in the second example (again containing a set of important variables). The middle column shows the outcome for the Hierarchical Multi sample-splitting method which performs similar to Ridge and Lasso-Pro in example 1 and similar to* Group-bound *in example 2.*

Pro and *MS-Split* perform well here as well. The power of the *Group-bound* is 0 when attempting to reject the small groups H_{0,S_0}. The type I error rate is supposedly controlled at level $\alpha = 0.05$ with all three methods. However, the *Lasso-Pro* and the hierarchical *MS-Split* methods fail to control the error rates, with the type I error rate even approaching 1 for large values of the correlation. The *Group-bound* and Ridge-based estimator have, in contrast, a type I error rate close to 0 for all

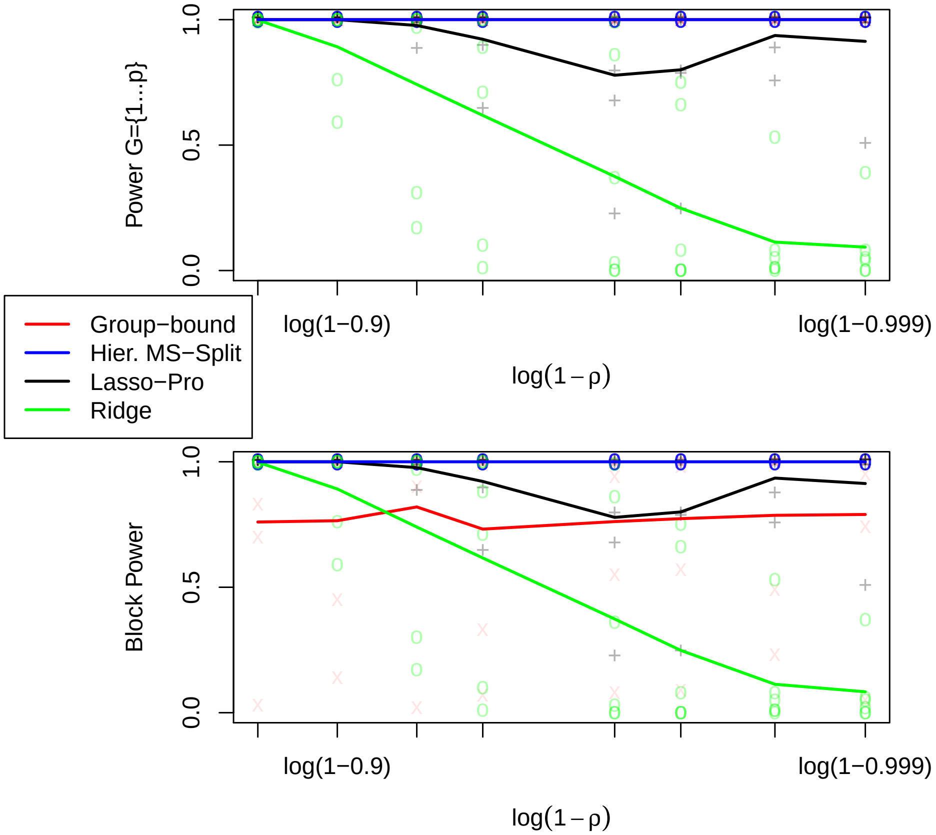

Fig 9. *The power for the rejection of the group-hypothesis of all variables (top) and the power for the rejection of the group-hypothesis of the variables in blocks highly correlated with S_0 variables (bottom). The design matrix used is of type* ***Block Equi.corr*** *which is similar to the Equi.corr setup in that Σ is block diagonal with blocks (of size 20×20) being the Σ of Equi.corr. The power is plotted as a function of the correlations in the blocks, quantified by ρ. The Ridge-based method loses power as the correlation between variables increases, while the group bound, Hier. MS-Split and Lasso-Pro methods can maintain power close to 1 for both measures of power.*

values of the correlation.

For highly correlated groups of variables, trying to detect the effect of individual variables has thus two inherent dangers. The power to detect interesting groups of variables might be very low. And the assumptions for the methods might be violated, which invalidates the type I error control. The assumption-free *Group-bound* method provides a powerful test for the group effects even if variables are perfectly correlated but suffers in power, relatively speaking, when variables are not highly correlated.

4.5 `hdi` for hierarchical inference

An implementation of the *Group-bound* method is provided in the `hdi` R-package (Meier et al., 2014).

For specific groups, one can provide a vector or a list of vectors where the elements of the vector

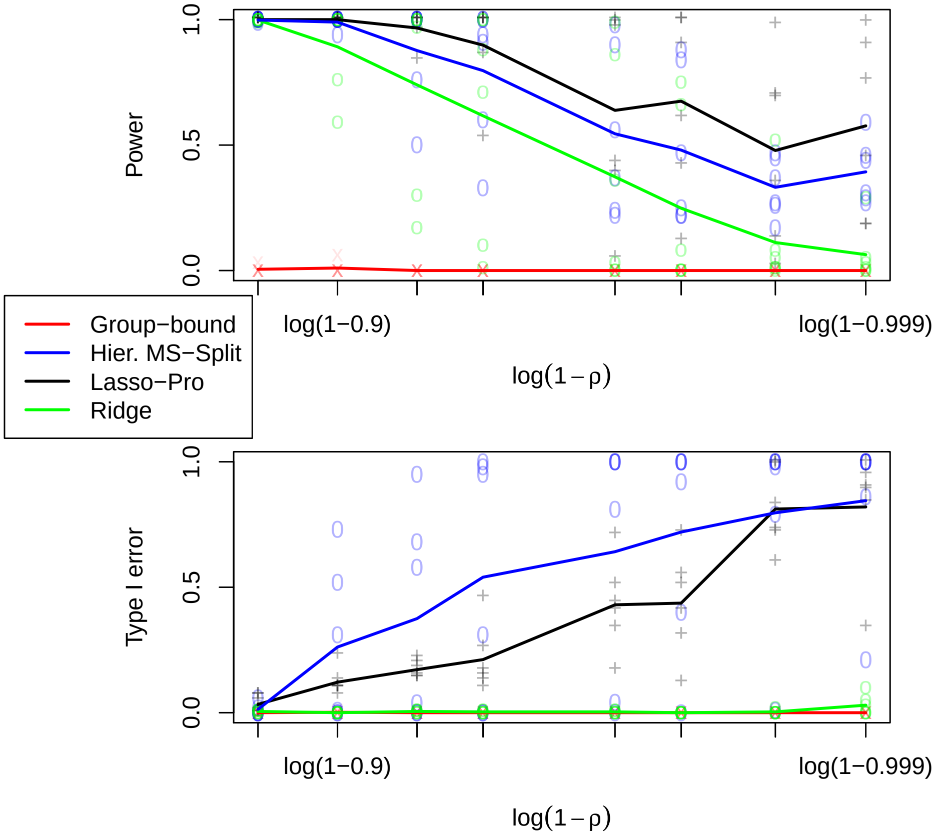

FIG 10. *The power for the rejection of the group-hypothesis of all S_0 variables (top) and type I error rate corresponding to the rejection of the group-hypothesis of all S_0^c variables (bottom) for the design matrix of type* ***Block Equi.corr*** *when changing the correlation ρ between variables. The design matrix type is described in detail in the caption of Figure 9 and in the text. The de-sparsified Lasso, Hier. MS-Split and the Ridge-based method lose power as the correlation between variables increases, while the* Group-bound *cannot reject the small group of variables S_0 (3 in this case). The de-sparsified Lasso and MS-Split methods also exceed the nominal type I error rate for high correlations (as the design assumptions break down), whereas the Ridge-based method and the* Group-bound *are both within the nominal 5% error rate for every correlation strength.*

specify the desired columns of $\mathbf{X}$ to be tested for. The following code tests the group hypothesis if the group contains all variables

```
> group          <- 1:ncol(x)
> outGroupBound <- groupBound(x = x, y = y, group = group, alpha = 0.05)
> rejection     <- outGroupBound > 0
```

Note that one needs to specify the significance level α.

One can also let the method itself apply the hierarchical clustering scheme as described at the beginning of Section 4. This works as follows:

```
> outClusterGroupBound <- clusterGroupBound(x = x, y = y, alpha = 0.05)
```

The output contains all clusters that were tested for significance in `members`. The corresponding lower bounds are contained in `lowerBound`.

To extract the significant clusters, one can do

```
> significant.cluster.numbers <- which(outClusterGroupBound$lowerBound > 0)
> significant.clusters <- outClusterGroupBound$members[[significant.cluster.numbers]]
```

The figures in the style of figure 8 can be achieved by using the function `plot` on `outClusterGroupBound`.

Note that one can specify the distance matrix used for the hierarchical clustering, as done for `hclust`.

To test group hypotheses $H_{0,G}$ for the Ridge and de-sparsified Lasso method as described in Section 4.3, one uses the output from the original single parameter fit, as illustrated for the group of all variables:

```
> outRidge <- ridge.proj(x = x, y = y)
> outLasso <- lasso.proj(x = x, y = y)
> group <- 1:ncol(x)
> outRidge$groupTest(group)
> outLasso$groupTest(group)
```

To apply an hierarchical clustering scheme as done in `clusterGroupBound`, one calls `clusterGroupTest`

```
> outRidge$clusterGroupTest(alpha = 0.95)
```

To summarize, the R-package provides functions to test individual groups as well as to test according to a hierarchical clustering scheme for the methods *Group-bound*, Ridge and de-sparsified Lasso. An implementation of the hierarchical Multi sample-splitting method is not provided at this point in time.

5. STABILITY SELECTION AND ILLUSTRATION WITH HDI

Stability selection (Meinshausen and Bühlmann, 2010) is another methodology to guard against false positive selections, by controlling the expected number of false positives $\mathbb{E}[V]$. The focus is on selection of a single or a group of variables in a regression model, or on a selection of more general discrete structures such as graphs or clusters. For example, for a linear model in (2.1) and with selection of single variables, stability selection provides a subset of variables $\hat{S}_{\text{stable}}$ such that for $V = |\hat{S}_{\text{stable}} \cap S_0^c|$ we have that $\mathbb{E}[V] \leq M$, where M is a pre-specified number.

For selection of single variables in a regression model, the method does not need a beta-min assumption but the theoretical analysis of stability selection for controlling $\mathbb{E}[V]$ relies on a restrictive exchangeability condition (which e.g. is ensured by a restrictive condition on the design matrix). This exchangeability condition seems far from necessary though (Meinshausen and Bühlmann, 2010). A refinement of stability selection is given in Shah and Samworth (2013).

An implementation of the stability selection procedure is available in the `hdi` R-package. It is called in a very similar way as the other methods. If we want to control e.g. $\mathbb{E}[V] \leq 1$ we use

```
> outStability <- stability(x = x, y = y, EV = 1)
```

The "stable" predictors are available in the element `select`. The default model selection algorithm is the Lasso (the first q variables entering the lasso paths). The option `model.selector` allows to apply a user defined model selection function.

6. R WORKFLOW EXAMPLE

We go through a possible R workflow based on the Riboflavin dataset (Bühlmann et al., 2014a) and methods provided in the hdi R-package.

```
> library(hdi)
> data(riboflavin)
```

We assume a linear model and we would like to investigate which effects are statistically significant on a significance level of $\alpha = 0.05$. Moreover, we want to construct the corresponding confidence intervals.

We start by looking at the individual variables. We want a conservative approach and, based on the results from Section 2.4, we choose the Ridge projection method for its good error control.

```
> outRidge <- ridge.proj(x = riboflavin$x, y = riboflavin$y)
```

We investigate if any of the multiple testing corrected p-values is smaller than our chosen significance level.

```
> any(outRidge$pval.corr <= 0.05)
[1] FALSE
```

We calculate the 95% confidence intervals for the first 3 predictors

```
> confint(outRidge,parm=1:3,level=0.95)
             lower    upper
AADK_at -0.8848403 1.541988
AAPA_at -1.4107374 1.228205
ABFA_at -1.3942909 1.408472
```

Disappointed with the lack of significance for testing individual variables, we want to investigate if we can find a significant group instead. From the procedure proposed for the Ridge method in Section 4, we know that if the Ridge method can't find any significant individual variables, it won't find a significant group either.

We apply the Group-bound method with its clustering option to try to find a significant group.

```
> outClusterGroupBound <- clusterGroupBound(x = riboflavin$x,
                                            y = riboflavin$y,
                                            alpha = 0.05)
> significant.cluster.numbers <- which(outClusterGroupBound$lowerBound > 0)
> significant.clusters <- outClusterGroupBound$members[[significant.cluster.numbers]]
> str(significant.clusters)
 num [1:4088] 1 2 3 4 5 6 7 8 9 10 ...
```

Only a single group, being the root node of the clustering tree, is found significant.

These results are in line with the results achievable in earlier studies of the same dataset in Bühlmann et al. (2014a) and van de Geer et al. (2014).

7. CONCLUSIONS

We present a (selective) overview of recent developments in frequentist high-dimensional inference for constructing confidence intervals and assigning p-values for the parameters in linear and generalized linear models. We include some methods which are able to detect significant groups of highly correlated variables which cannot be individually detected as single variables. We complement the methodology and theory viewpoints with a broad empirical study. The latter indicates that more "stable" procedures based on Ridge estimation or sample splitting with subsequent aggregation might be more reliable for type I error control, at the price of losing power; asymptotically, power-optimal methods perform nicely in well-posed scenarios but are more exposed to fail for error control in more difficult settings where the design or degree of sparsity are more ill-posed. We introduce the R-package hdi which allows the user to choose from a collection of frequentist inference methods and eases reproducible research.

APPENDIX A: APPENDIX

A.1 Additional definitions and descriptions

Compatibility condition (Bühlmann and van de Geer, 2011, p.106). Consider a fixed design matrix $\mathbf{X}$. We define the following.

The compatibility condition holds if for some $\phi_0 > 0$ and all β satisfying $\|\beta_{S_0^c}\|_1 \le 3\|\beta_{S_0}\|_1$:

$$\|\beta_{S_0}\|_1^2 \le \beta^T \hat{\Sigma} \beta s_0/\phi_0^2, \quad \hat{\Sigma} = n^{-1}\mathbf{X}^T\mathbf{X}. \tag{A.1}$$

Here β_A denotes the components $\{\beta_j;\ j \in A\}$ where $A \subseteq \{1,\dots,p\}$. The number ϕ_0 is called the compatibility constant.

Aggregation of dependent p-values Aggregation of dependent p-values can be generically done as follows.

Lemma 1. *(Implicitly contained in Meinshausen et al. (2009)).*
Assume that we have B p-values $P^{(1)},\dots,P^{(B)}$ for testing a null-hypothesis H_0, i.e., for every $b \in \{1,\dots,B\}$ and any $0 < \alpha < 1$, $\mathbb{P}_{H_0}[P^{(b)} \le \alpha] \le \alpha$. Consider for any $0 < \gamma < 1$ the empirical γ-quantile

$$Q(\gamma) = \ \min\left(\textit{empirical } \gamma\textit{-quantile } \{P^{(1)}/\gamma,\dots,P^{(B)}/\gamma\}, 1\right),$$

and the minimum value of $Q(\gamma)$, suitably corrected with a factor, over the range $(\gamma_{\min}, 1)$ for some positive (small) $0 < \gamma_{\min} < 1$:

$$P = \min\left((1 - \log(\gamma_{\min})) \min_{\gamma \in (\gamma_{\min},1)} Q(\gamma), 1\right).$$

Then, both $Q(\gamma)$ (for any fixed $\gamma \in (0,1)$) and P are conservative p-values satisfying for any $0 < \alpha < 1$: $\mathbb{P}_{H_0}[Q(\gamma) \le \alpha] \le \alpha$ or $\mathbb{P}_{H_0}[P \le \alpha] \le \alpha$, respectively.

Bounding the error of the estimated bias correction in the de-sparsified Lasso. We will argue now why the error from the bias correction

$$\sum_{k \neq j} \sqrt{n} P_{jk}(\hat{\beta}_k - \beta_k^0)$$

is negligible. From the KKT conditions when using the Lasso of $\mathbf{X}^{(j)}$ versus $\mathbf{X}^{(-j)}$ we have (Bühlmann and van de Geer, 2011, cf. Lem 2.1):

$$\max_{k \neq j} 2|n^{-1}(X^{(k)})^T Z^{(j)}| \leq \lambda_j. \tag{A.2}$$

Therefore,

$$\begin{aligned} |\sqrt{n}\sum_{k\neq j} P_{jk}(\hat{\beta}_k - \beta_k^0)| &\leq \sqrt{n}\max_{k\neq j}|P_{jk}|\|\hat{\beta}-\beta^0\|_1 \\ &\leq 2\sqrt{n}\lambda_j\|\hat{\beta}-\beta^0\|_1(n^{-1}(\mathbf{X}^{(j)})^T Z^{(j)})^{-1}. \end{aligned}$$

Assuming sparsity and the compatibility condition (A.1), and when choosing $\lambda_j \asymp \sqrt{\log(p)/n}$, one can show that $(n^{-1}(\mathbf{X}^{(j)})^T Z^{(j)})^{-1} = O_P(1)$ and $\|\hat{\beta}-\beta^0\|_1 = O_P(s_0\sqrt{\log(p)/n})$ (for the latter, see (2.2)). Therefore,

$$|\sqrt{n}\sum_{k\neq j} P_{jk}(\hat{\beta}_k - \beta_k^0)| \leq O_P(\sqrt{n}s_0\sqrt{\log(p)/n}\lambda_j) = O_P(s_0\log(p)n^{-1/2}),$$

where the last bound follows by assuming $\lambda_j \asymp \sqrt{\log(p)/n}$. Thus, if $s_0 \ll n^{1/2}/\log(p)$, the error from bias correction is asymptotically negligible.

Choice of λ_j for de-sparsified Lasso. We see from (A.2), that the numerator of the error in the bias correction term (i.e. the P_{jk}'s) is decreasing as $\lambda_j \searrow 0$; for controlling the denominator, λ_j shouldn't be too small to ensure that the denominator (i.e. $n^{-1}(\mathbf{X}^{(j)})^T Z^{(j)}$) behaves reasonable (staying away from zero) for a fairly large range of λ_j.

Therefore, the strategy is as follows.

1. Compute a Lasso regression of $X^{(j)}$ versus all other variables $\mathbf{X}^{(-j)}$ using CV, and the corresponding residual vector is denoted by $Z^{(j)}$.
2. Compute $\|Z^{(j)}\|_2^2/((\mathbf{X}^{(j)})^T Z^{(j)})^2$ which is the asymptotic variance of $\hat{b}_j/\sigma_\varepsilon$, assuming that the error in the bias correction is negligible.
3. Increase the variance by 25%, i.e., $V_j = 1.25\|Z^{(j)}\|_2^2/((\mathbf{X}^{(j)})^T Z^{(j)})^2$.
4. Search for the smallest λ_j such that the corresponding residual vector $Z^{(j)}(\lambda_j)$ satisfies:

$$\|Z^{(j)}(\lambda_j)\|_2^2/((\mathbf{X}^{(j)})^T Z^{(j)}(\lambda_j))^2 \leq V_j.$$

This procedure is similar to the choice of λ_j advocated in Zhang and Zhang (2014).

Bounding the error of bias correction for the Ridge projection. The goal is to derive the formula (2.11). Based on (2.9) we have:

$$\begin{aligned} &\sigma_\varepsilon^{-1}\Omega_{R;jj}^{-1/2}(\hat{b}_{R;j} - \beta_j^0) \approx \Omega_{R;jj}^{-1/2}W_j + \sigma_\varepsilon^{-1}\Omega_{R;jj}^{-1/2}\Delta_{R;j}, \quad W \sim \mathcal{N}_p(0,\Omega_R), \\ &|\Delta_{R;j}| \leq \max_{k\neq j}\left|\frac{P_{R;jk}}{P_{R;jj}}\right| \|\hat{\beta}-\beta^0\|_1. \end{aligned}$$

In relation to the result in Fact 2 for the de-sparsified Lasso, the problem here is that the behavior of $\max_{k\neq j}|P_{R;jj}^{-1}P_{R:jk}|$ and of the diagonal elements $\Omega_{R;jj}$ are hard to control but fortunately, these quantities are fixed and observed for fixed design $\mathbf{X}$.

By invoking the compatibility constant for the design $\mathbf{X}$, we obtain the bound for $\|\hat{\beta}-\beta^0\|_1 \leq s_0 4\lambda/\phi_0$ in (2.2) and therefore, we can upper-bound

$$|\Delta_{R;j}| \leq 4 s_0 \lambda/\phi_0^2 \max_{k\neq j} \left|\frac{P_{R;jk}}{P_{R;jj}}\right|.$$

Asymptotically, for Gaussian errors, we have with high probability

$$\begin{aligned} |\Delta_{R;j}| &= O(s_0\sqrt{\log(p)/n} \max_{k\neq j}\left|\frac{P_{R;jk}}{P_{R;jj}}\right|) \\ &\leq O((\log(p)/n)^{1/2-\xi} \max_{k\neq j}\left|\frac{P_{R;jk}}{P_{R;jj}}\right|), \end{aligned} \tag{A.3}$$

where the last inequality holds due to assuming $s_0 = O((n/\log(p))^{\xi})$ for some $0 < \xi < 1/2$. In practice, we use the bound from (A.3) in the form:

$$\Delta_{R\text{bound};j} := \max_{k\neq j}\left|\frac{P_{R;jk}}{P_{R;jj}}\right| (\log(p)/n)^{1/2-\xi},$$

with the typical choice $\xi = 0.05$.

A.2 Confidence intervals for Multi sample-splitting

We construct confidence intervals that satisfy the duality with the p-values from equation (2.5), and thus, they are corrected already for multiplicity:

$$\begin{aligned} (1-\alpha)\%\text{ CI} &= \text{Those values } c \text{ for which the p-value } \geq \alpha \text{ for testing the null hypothesis } H_{0,j}: \beta = c \\ &= \text{Those } c \text{ for which the p-value resulting from the p-value aggregation procedure is} \geq \alpha, \\ &= \{c | P_j \geq \alpha\}, \\ &= \{c | (1-\log\gamma_{min}) \inf_{\gamma\in(\gamma_{min},1)} Q_j(\gamma) \geq \alpha\}, \\ &= \{c | \forall\gamma \in (\gamma_{min},1) : (1-\log\gamma_{min}) Q_j(\gamma) \geq \alpha\}, \\ &= \{c | \forall\gamma \in (\gamma_{min},1) : \min(1, \mathit{emp.}\ \gamma\ \mathit{quantile}(P^{[b]}_{corr;j})/\gamma) \geq \alpha/(1-\log\gamma_{min})\}, \\ &= \{c | \forall\gamma \in (\gamma_{min},1) : \mathit{emp.}\ \gamma\ \mathit{quantile}(P^{[b]}_{corr;j})/\gamma \geq \alpha/(1-\log\gamma_{min})\}, \\ &= \{c | \forall\gamma \in (\gamma_{min},1) : \mathit{emp.}\ \gamma\ \mathit{quantile}(P^{[b]}_{corr;j}) \geq \frac{\alpha\gamma}{(1-\log\gamma_{min})}\}. \end{aligned}$$

We will use the notation $\gamma^{[b]}$ for the position of $P^{[b]}_{corr;j}$ in the ordering by increasing value of the corrected p-values $P^{[i]}_{corr;j}$, divided by B.
We can now rewrite our former expression in a form explicitly using our information from every sample split

$$\begin{aligned} &(1-\alpha)\%\text{ CI} \\ &= \{c | \forall b = 1,\ldots,B : (\gamma^{[b]} \leq \gamma_{min}) \vee (P^{[b]}_{corr;j} \geq \frac{\alpha\gamma^{[b]}}{(1-\log\gamma_{min})})\} \\ &= \{c | \forall b = 1,\ldots,B : (\gamma^{[b]} \leq \gamma_{min}) \vee (c \in \text{ the } \left(1 - \frac{\alpha\gamma^{[b]}}{(1-\log\gamma_{min})|\hat{S}^{[b]}|}\right) 100\%\text{CI for split b})\}. \end{aligned}$$

For single testing (not adjusted for multiplicity), the corresponding confidence interval becomes:

$$\begin{aligned}&(1-\alpha)\%\ \text{CI}\\ &= \{c|\forall b = 1,\dots,B : (\gamma^{[b]} \le \gamma_{min}) \vee (c \in \text{ the } \left(1 - \frac{\alpha\gamma^{[b]}}{(1-\log\gamma_{min})}\right) 100\%\text{CI for split b})\}.\end{aligned}$$

If one has starting points with one being in the confidence interval and the other one outside of it, one can apply the bisection method to find the bound in between these points.

A.3 Broad comparison results with fixed active set positions

In Figures 11, 12 and 13 the simulation results can be found where the positions of the non-zero coefficients were fixed to the first s_0 positions.

Note that in comparison to the case with random active set positions, the type I error is better controlled for the Toeplitz design whereas it is poor for all methods for the Exp.decay setup.

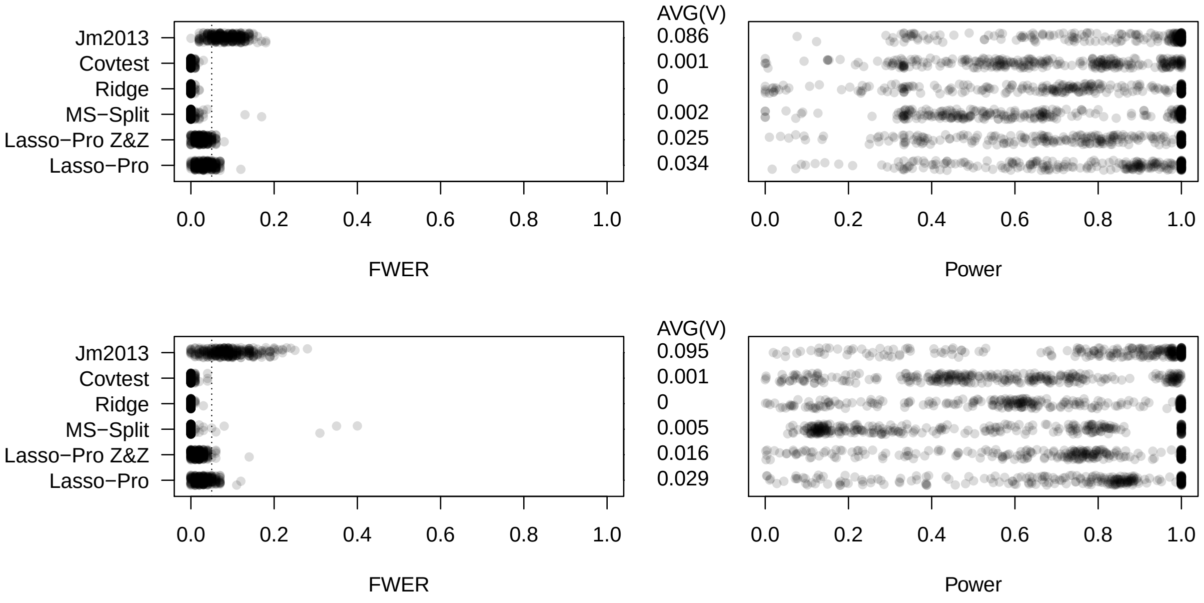

Fig 11. *Familywise error rate (FWER) and power for multiple testing based on various methods for a linear model. The desired control level for the FWER is $\alpha = 0.05$. The average number of false positives AVG(V) for each method is shown in the middle. The design matrix is of type* ***Toeplitz****, and the active set with fixed position of active variables has sizes $s_0 = 3$ (top) and $s_0 = 15$ (bottom).*

A.4 More empirical confidence interval results

We present the confidence interval results corresponding to the setup combinations from the p-value results presented in Section 2.4.2. Some results are also for non-ordered non-zero coefficients with which is meant that their positions as columns of $\boldsymbol{x}$ were chosen randomly.

Generally speaking we can say that the overall coverage of the zero-coefficients is good. (The number to the right of the plot is always bigger than or equal to 95). Depending on the ordering of the variables and the correlation structure of the design, coverage can be poor for the non-zero coefficients or a few zero coefficients.

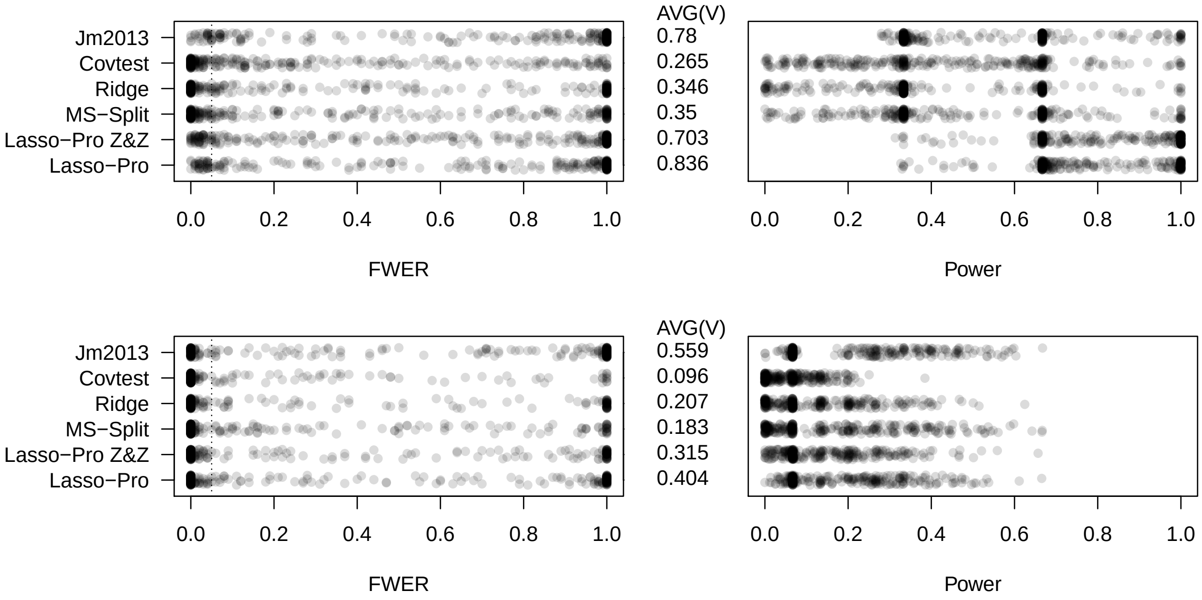

FIG 12. *See caption of Figure 11 with the only difference being the type of design matrix. In this plot the design matrix type is* ***Exp.decay***.

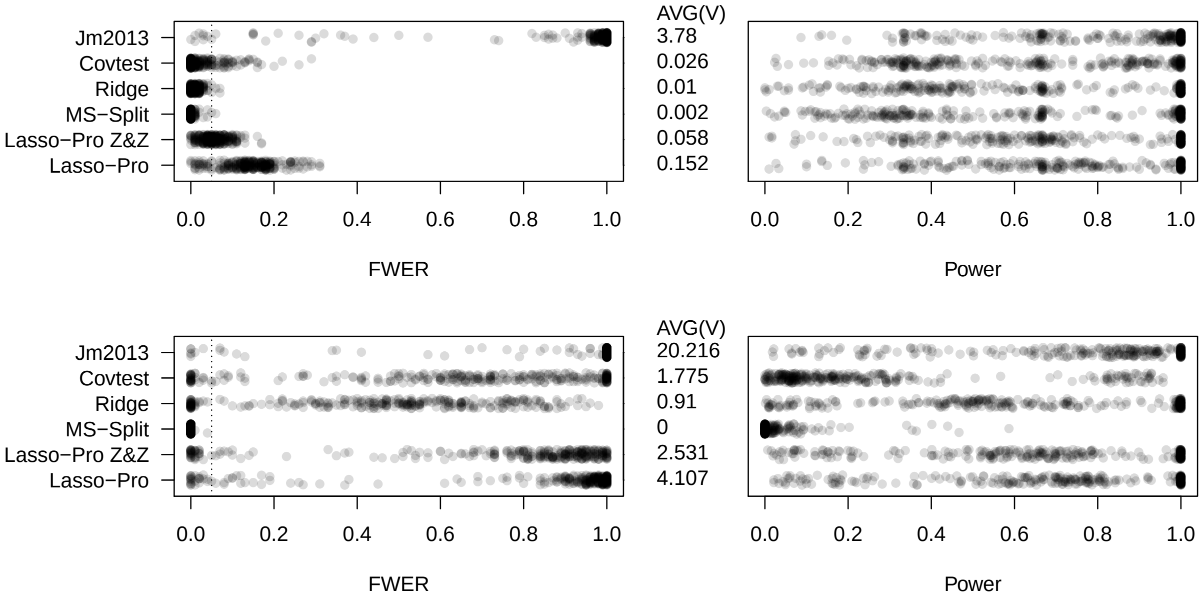

FIG 13. *See caption of Figure 11 with the only difference being the type of design matrix. In this plot, the design matrix type is* ***Equi.corr***.

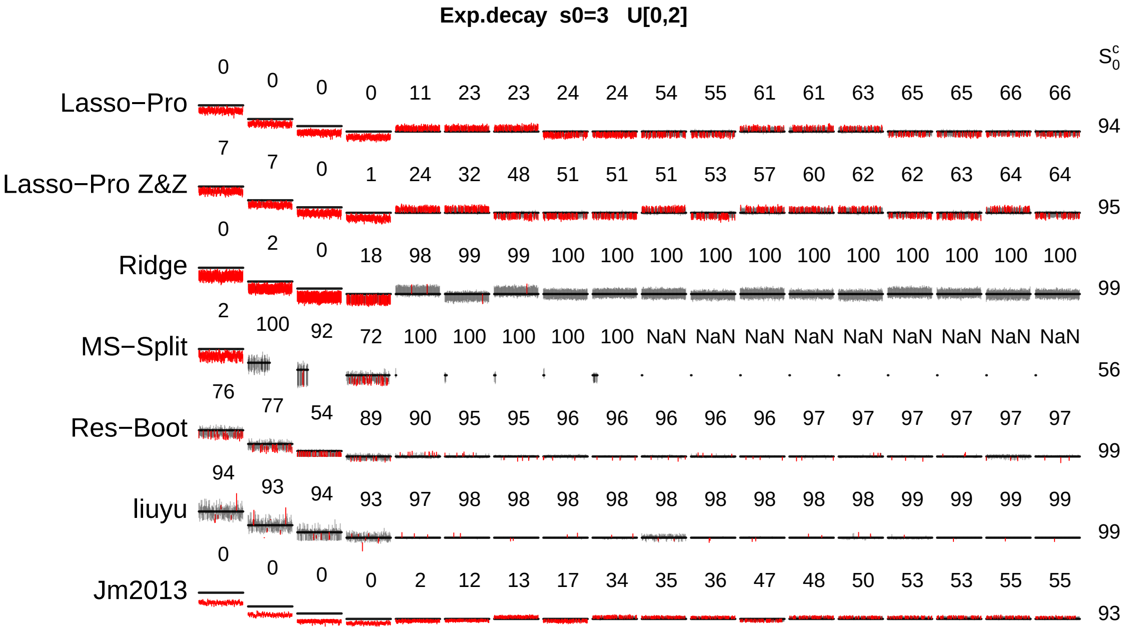

Fig 14. *See caption of Figure 6 with the only difference being the type of design matrix. In this plot, the design matrix type is* ***Exp.decay***

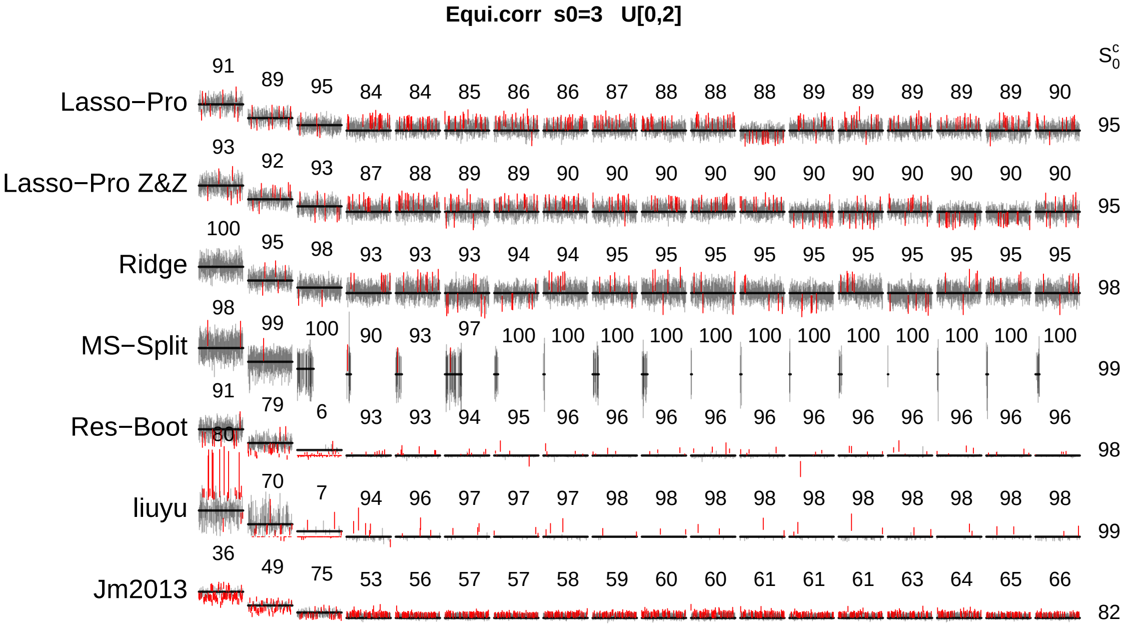

Fig 15. *See caption of Figure 6 with the only difference being the type of design matrix. In this plot, the design matrix type is* ***Equi.corr***

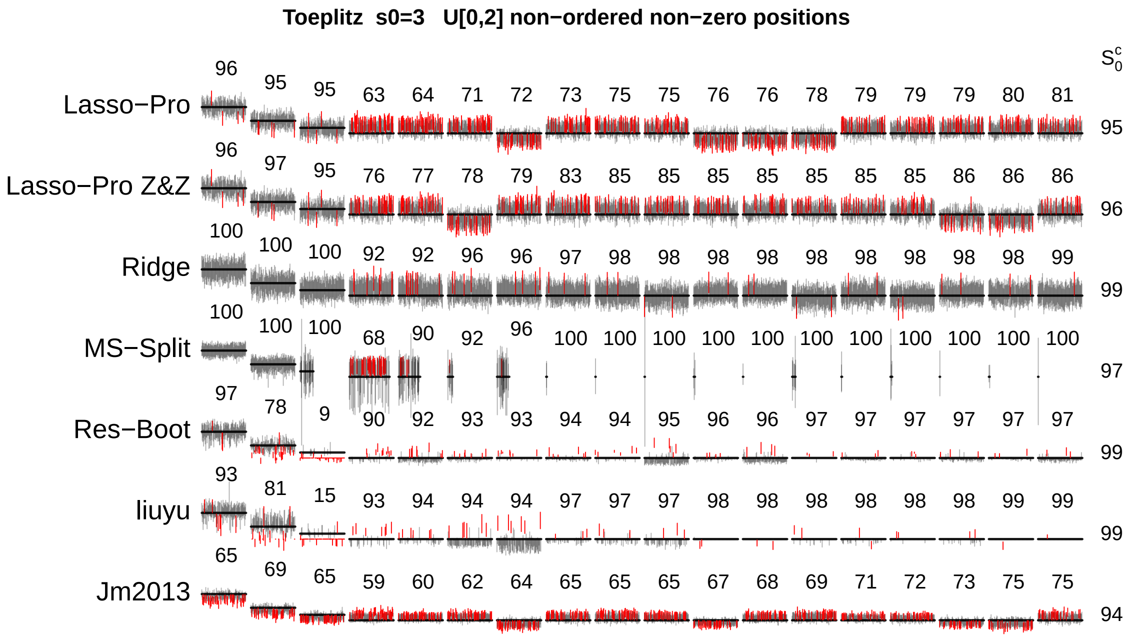

FIG 16. *See caption of Figure 6 with the only difference being the positions of the non-zero coefficients being non-ordered (as in not the first s_0 columns of $\boldsymbol{X}$).*

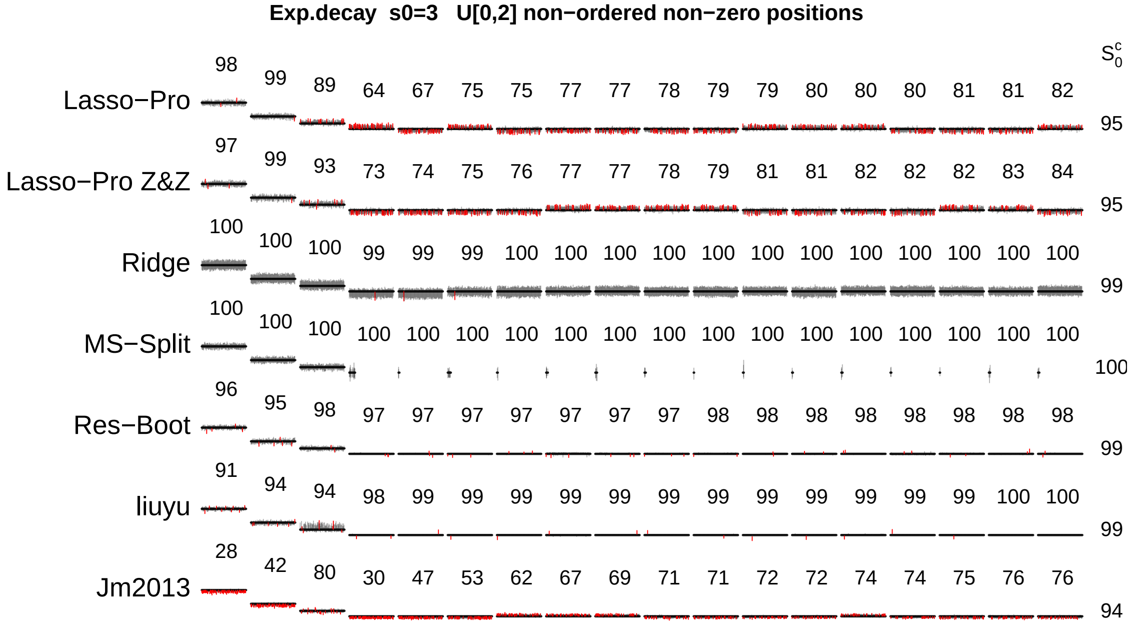

FIG 17. *See caption of Figure 6 with the only differences being the positions of the non-zero coefficients being non-ordered (as in not the first s_0 columns of $\boldsymbol{X}$) and the design matrix type to be* ***Exp.decay****.*

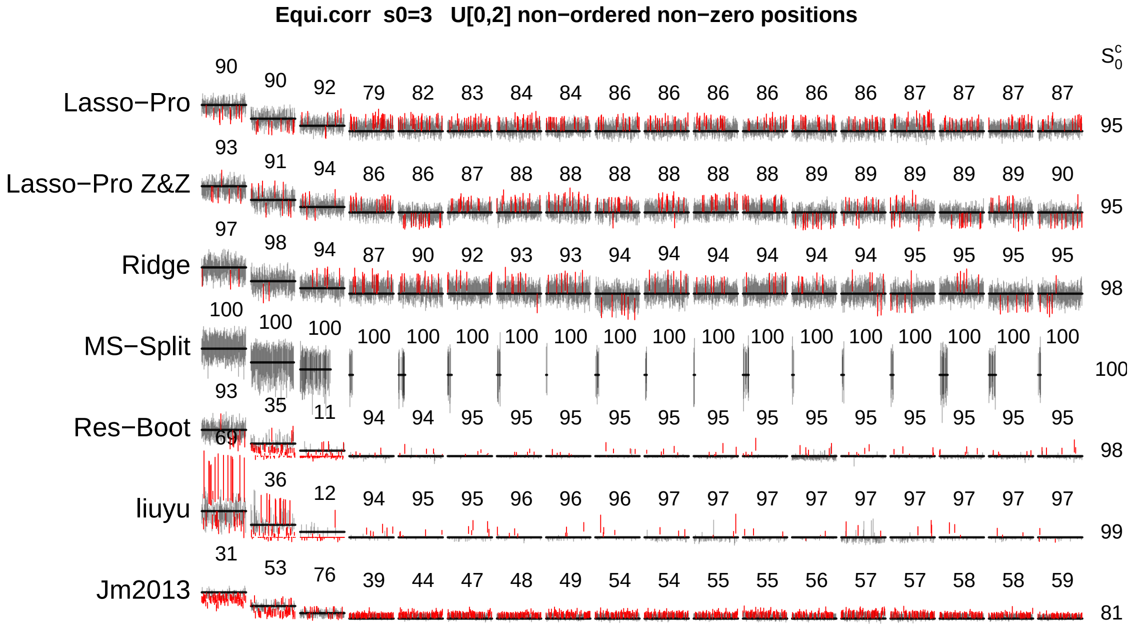

FIG 18. *See caption of Figure 6 with the only differences being the positions of the non-zero coefficients being non-ordered (as in not the first s_0 columns of $\boldsymbol{X}$) and the design matrix type to be* ***Equi.corr****.*

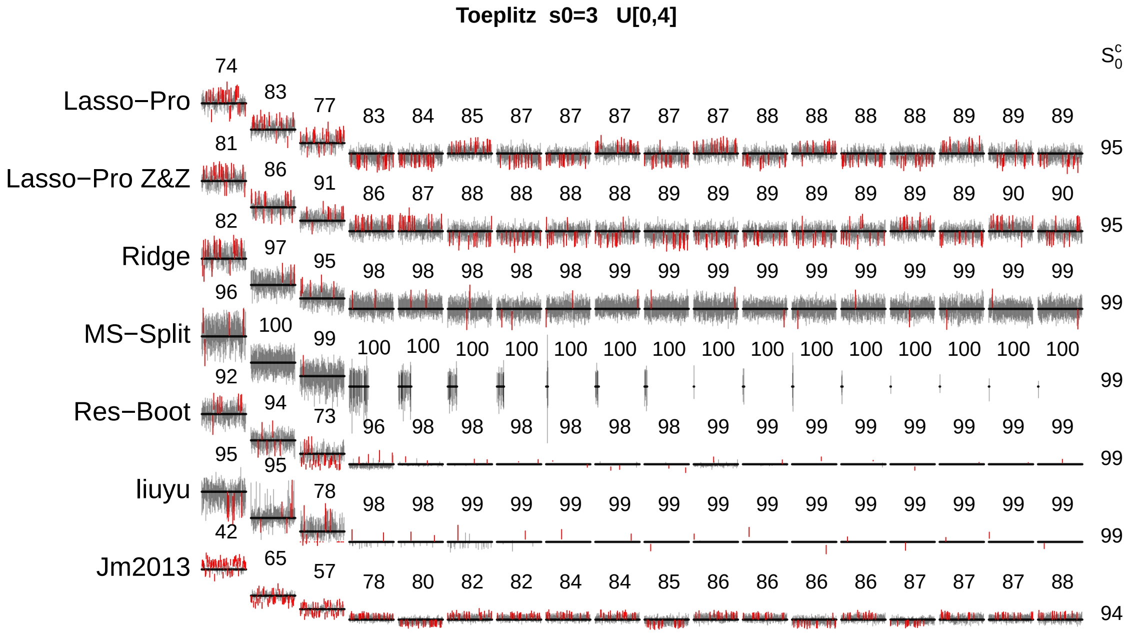

FIG 19. *See caption of Figure 6 with the only difference being the size of the non-zero coefficients being $U[0, 4]$.*

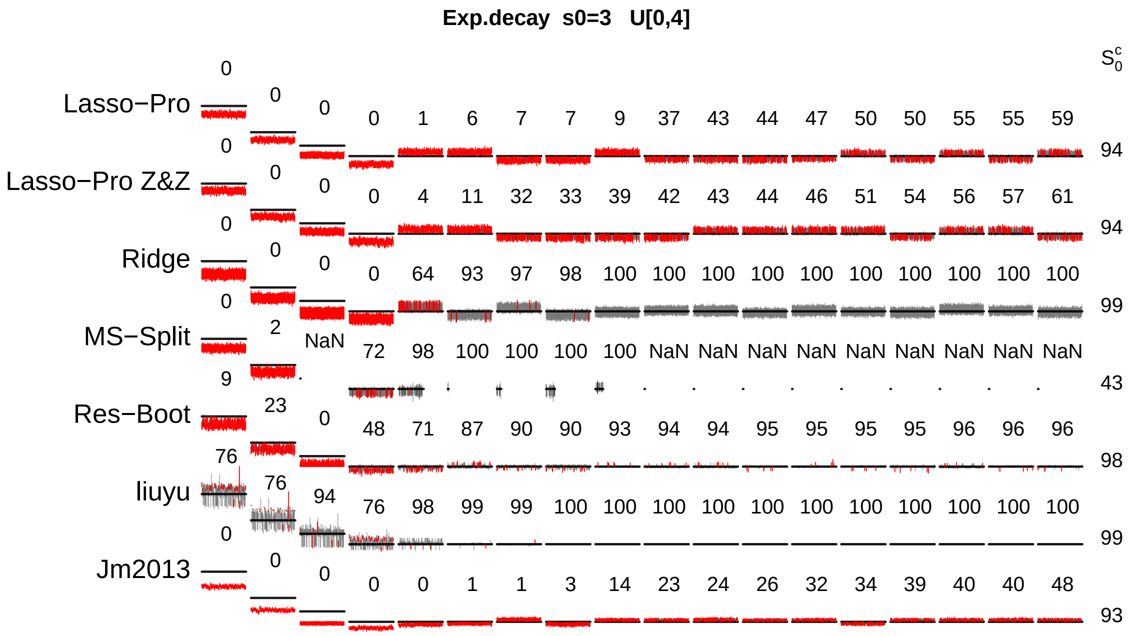

Fig 20. *See caption of Figure 6 with the only differences being the size of the non-zero coefficients being $U[0,4]$ and the design matrix type being* ***Exp.decay****.*

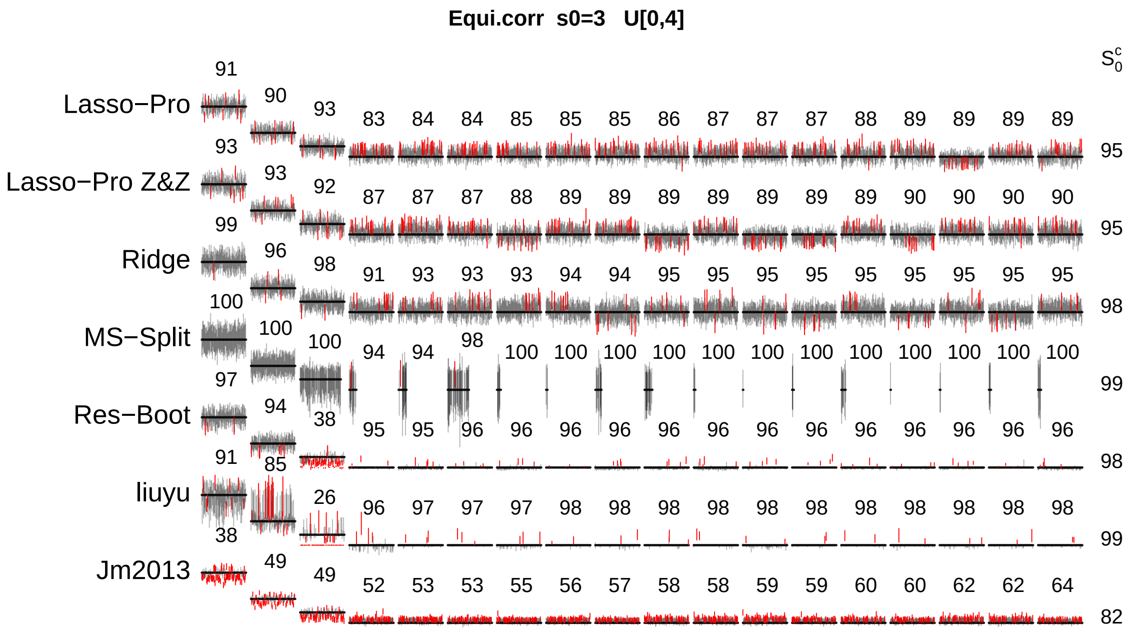

Fig 21. *See caption of Figure 6 with the only differences being the size of the non-zero coefficients being $U[0,4]$ and the design matrix type being* ***Equi.corr****.*

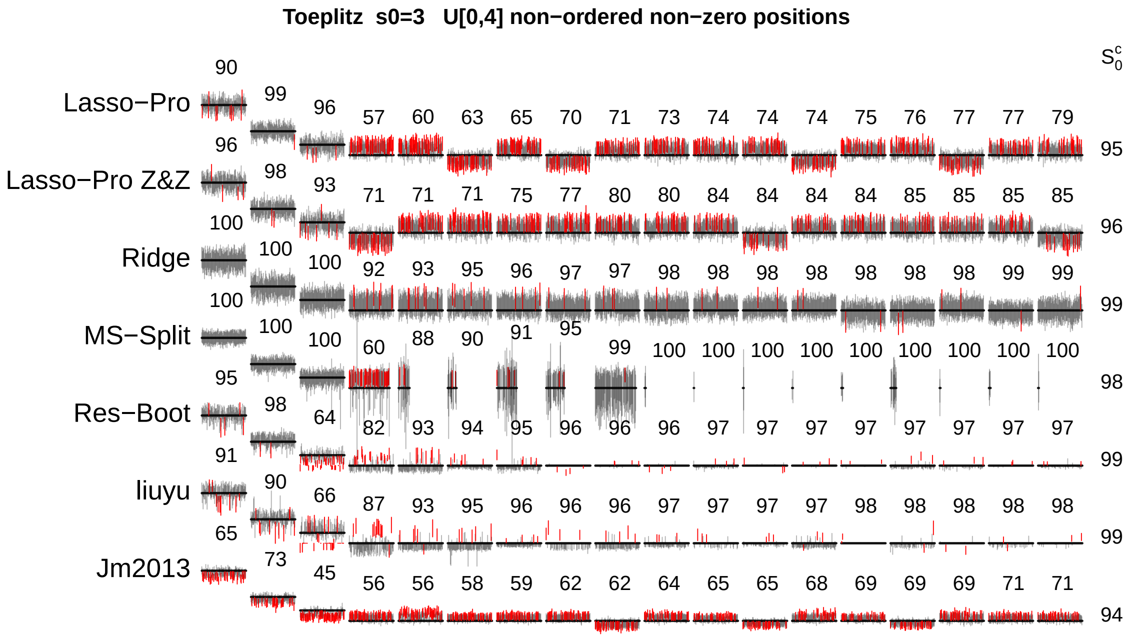

FIG 22. *See caption of Figure 6 with the only differences being the size of the non-zero coefficients being $U[0,4]$ and the positions of the non-zero coefficients being non-ordered (as in not the first s_0 columns of $\boldsymbol{X}$).*

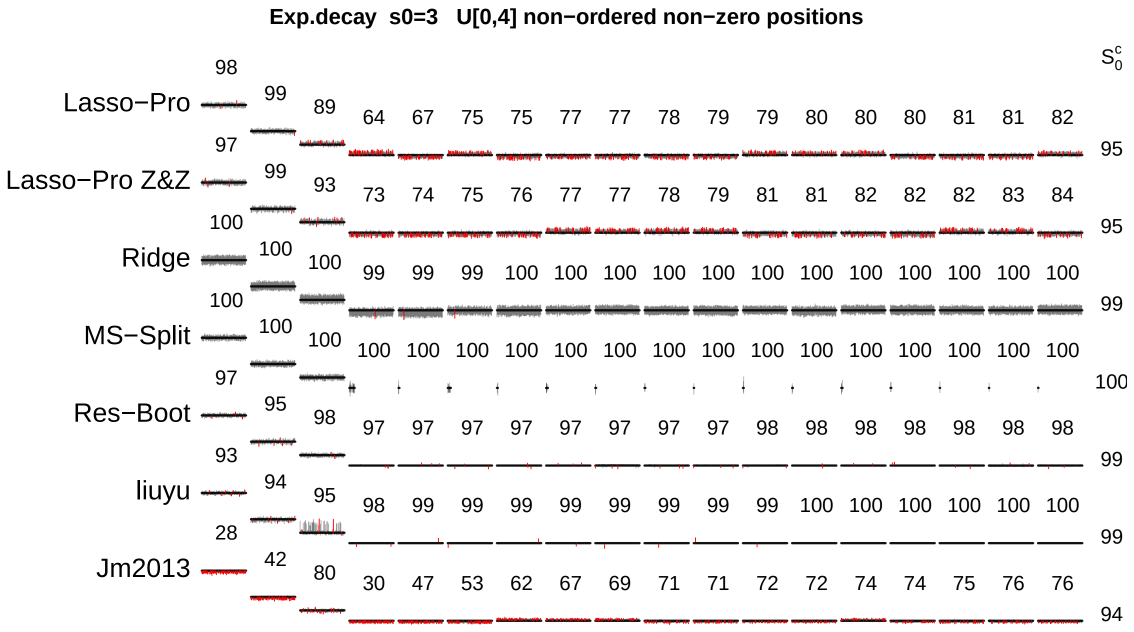

FIG 23. *See caption of Figure 6 with the differences being the type of design matrix being* ***Exp.decay****, the size of the non-zero coefficients being $U[0,4]$ and the positions of the non-zero coefficients being non-ordered (as in not the first s_0 columns of $\boldsymbol{X}$).*

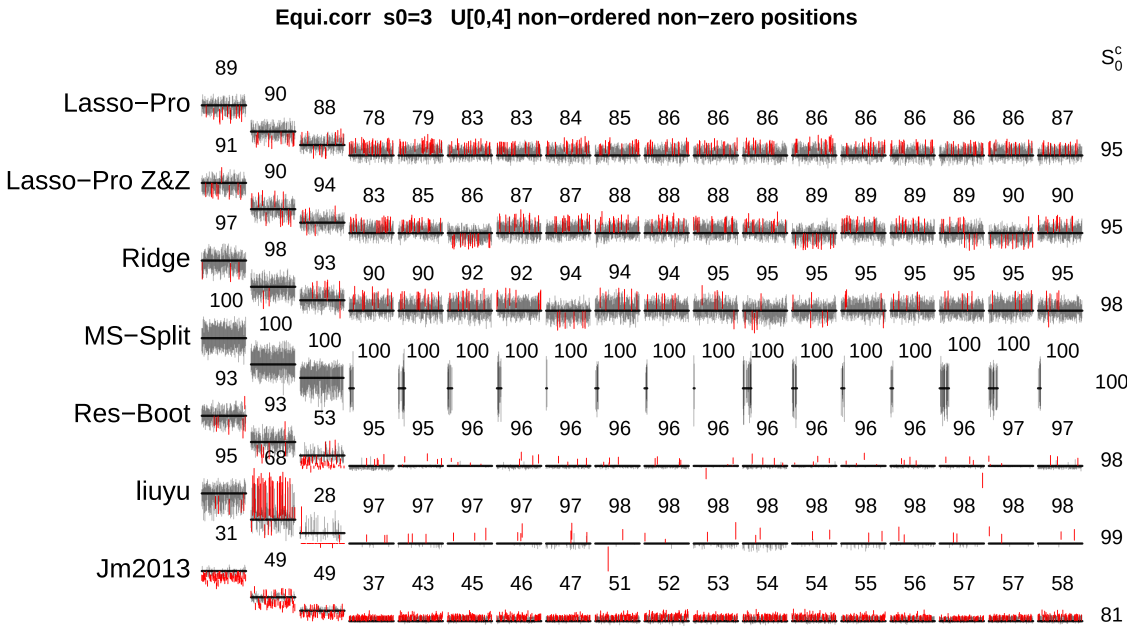

FIG 24. *See caption of Figure 6 with the differences being the type of design matrix being* ***Equi.corr****, the size of the non-zero coefficients being* $U[0,4]$ *and the positions of the non-zero coefficients being non-ordered (as in not the first* s_0 *columns of* $\boldsymbol{X}$*).*

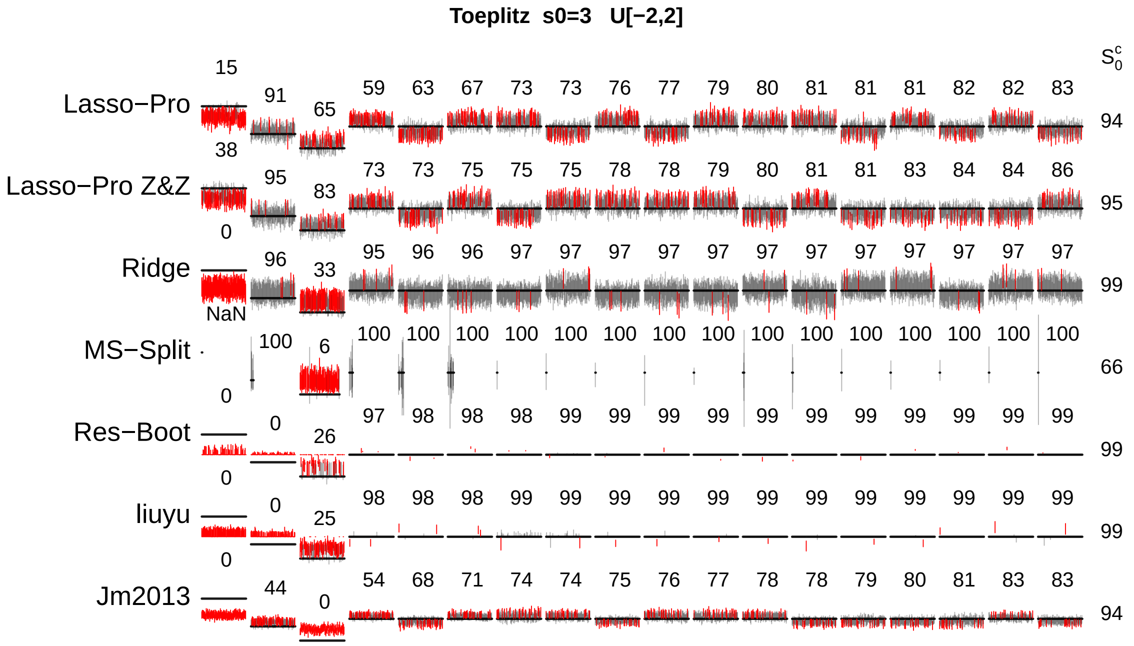

FIG 25. *See caption of Figure 6 with the only difference being the size of the non-zero coefficients being* $U[-2,2]$.

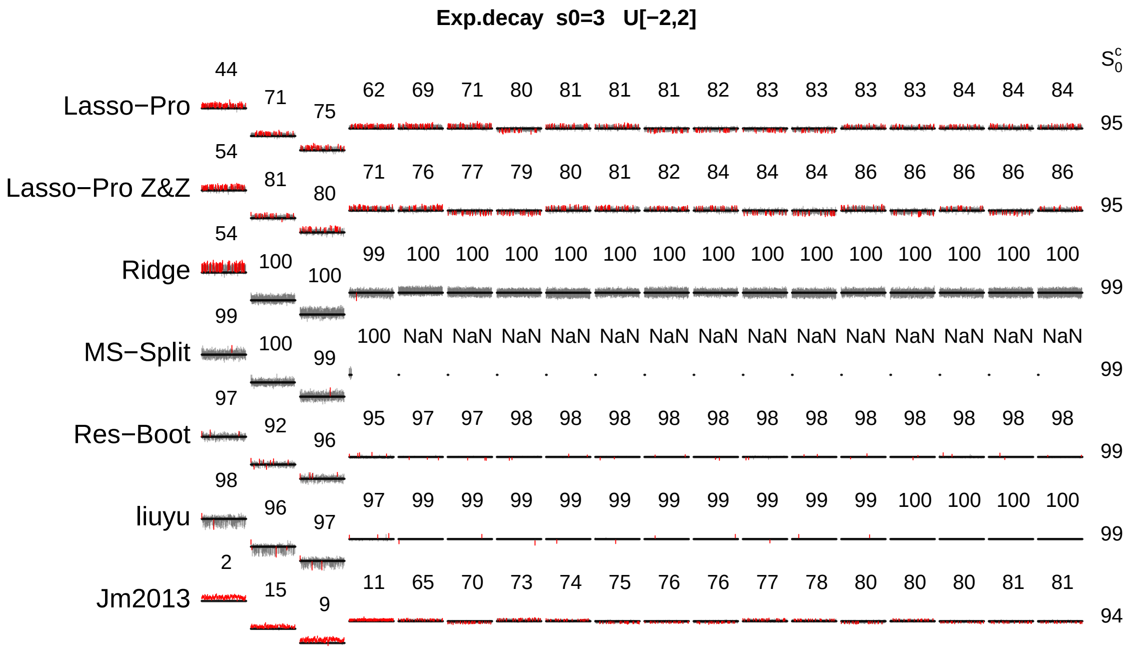

FIG 26. *See caption of Figure 6 with the only differences being the size of the non-zero coefficients being $U[-2, 2]$ and the design matrix type being **Exp.decay**.*

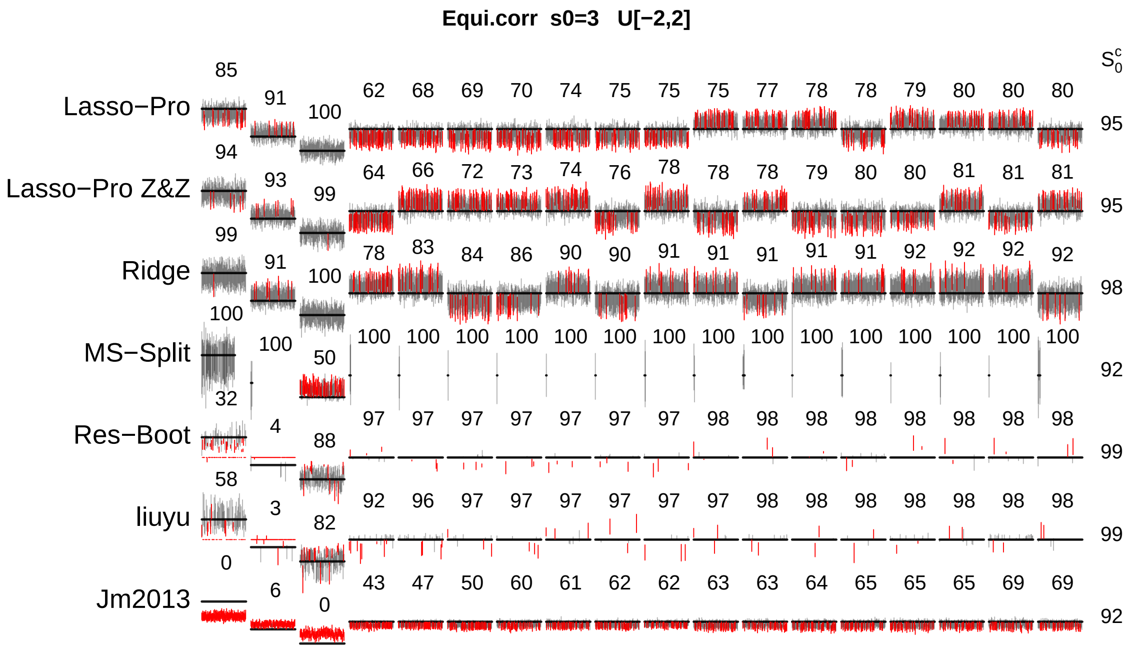

FIG 27. *See caption of Figure 6 with the only differences being the size of the non-zero coefficients being $U[-2, 2]$ and the design matrix type being **Equi.corr**.*

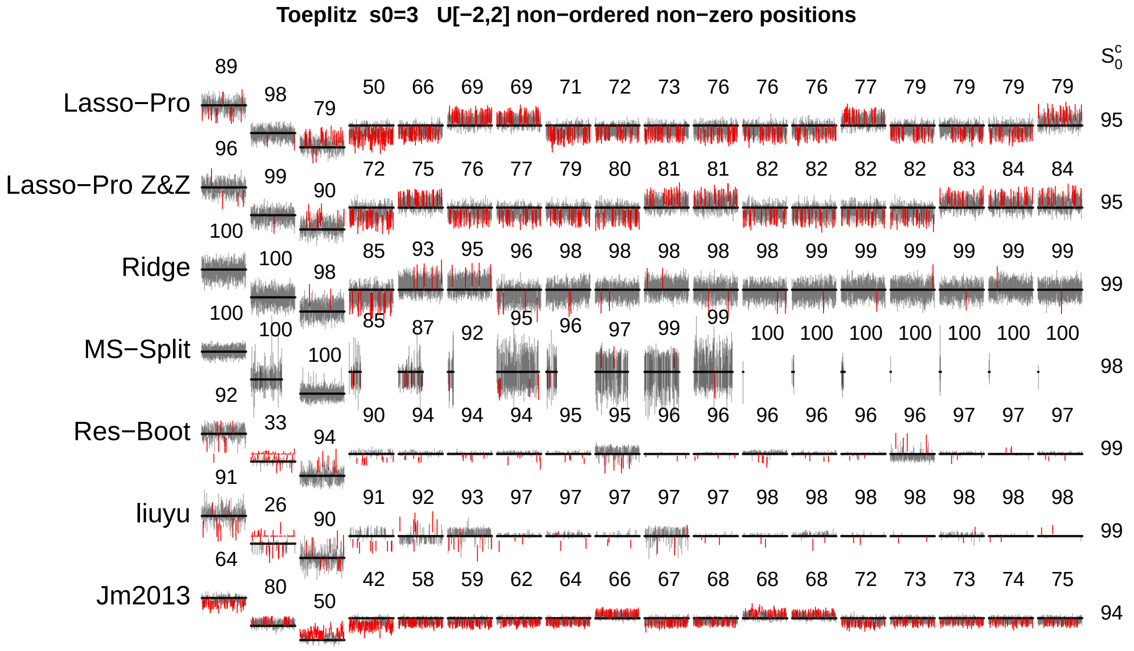

FIG 28. *See caption of Figure 6 with the only differences being the size of the non-zero coefficients being $U[-2,2]$ and the positions of the non-zero coefficients being non-ordered (as in not the first s_0 columns of $\boldsymbol{X}$).*

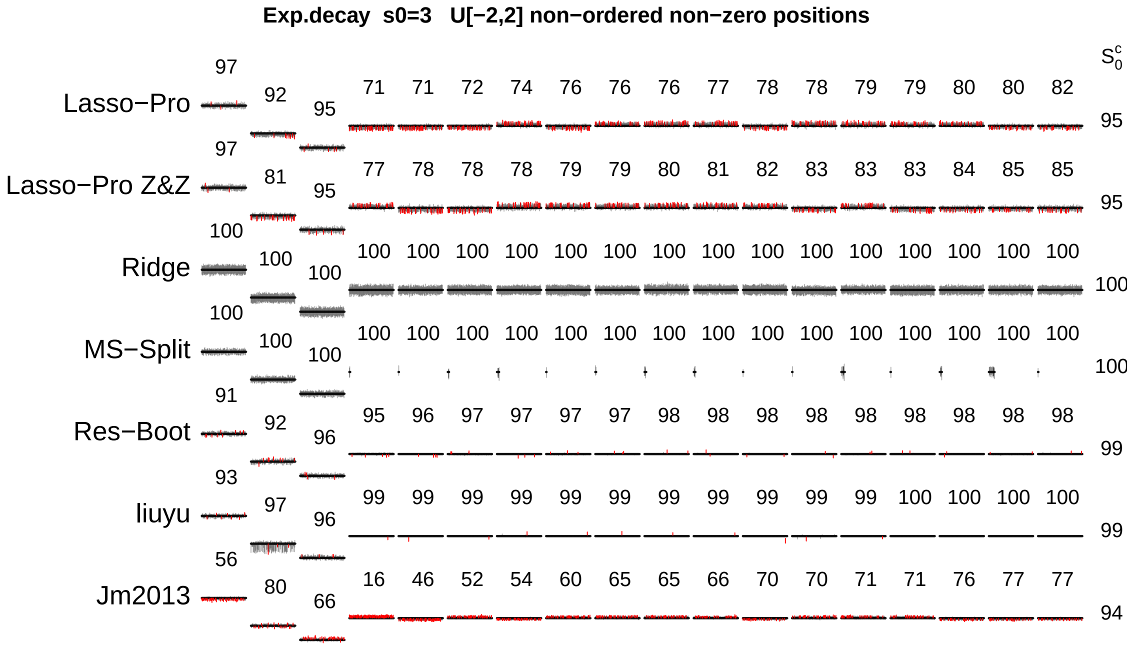

FIG 29. *See caption of Figure 6 with the differences being the type of design matrix being **Exp.decay**, the size of the non-zero coefficients being $U[-2,2]$ and the positions of the non-zero coefficients being non-ordered (as in not the first s_0 columns of $\boldsymbol{X}$).*

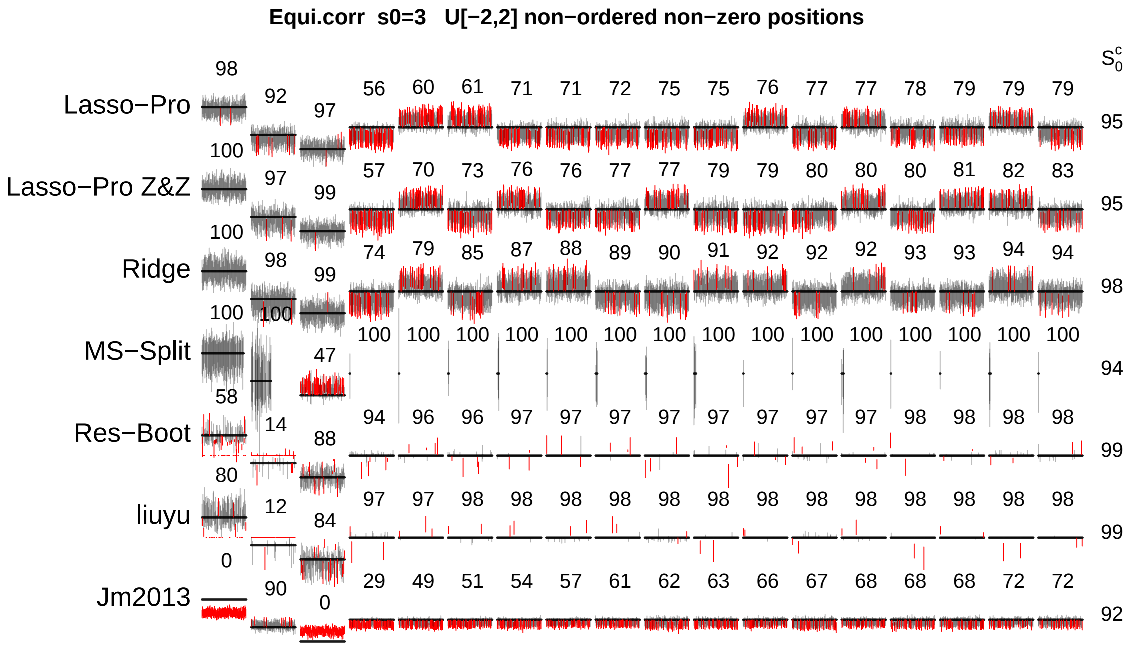

FIG 30. *See caption of Figure 6 with the differences being the type of design matrix being* ***Equi.corr****, the size of the non-zero coefficients being* $U[-2,2]$ *and the positions of the non-zero coefficients being non-ordered (as in not the first* s_0 *columns of* $\boldsymbol{X}$*).*

A.5 Weighted squared error approach for general GLM

We describe the approach presented in Section 3.2 in a more general way. One algorithm for fitting generalized linear models is to calculate the maximum likelihood estimates $\hat{\beta}$ by applying iterative weighted least squares (McCullagh and Nelder, 1989).

As in Section 3.2, the idea is now to apply standard l1-penalized fitting of the model, then build up the weighted least squares problem at the l1-solution and apply our linear model methods on this problem.

From McCullagh and Nelder (1989), using the notation $\hat{z}_i = g^{-1}((\boldsymbol{X}\hat{\beta})_i), i = 1, \dots, n$, the adjusted response variable becomes

$$Y_{i,adj} = (\boldsymbol{X}\hat{\beta})_i + (Y_i - \hat{z}_i)\frac{\partial g(z)}{\partial z}\bigg|_{z=\hat{z}_i}, i = 1, \dots, n.$$

We then get a weighted least squares problem

$$\hat{\beta}_{new} = \text{argmin}_{\beta}(Y_{adj} - \boldsymbol{X}\beta)^T \boldsymbol{W}(Y_{adj} - \boldsymbol{X}\beta),$$

with weights

$$\boldsymbol{W}^{-1} = \begin{pmatrix} (\frac{\partial g(z)}{\partial z})^2\Big|_{z=\hat{z}_1} V(\hat{z}_1) & 0 & \dots & 0 \\ 0 & (\frac{\partial g(z)}{\partial z})^2\Big|_{z=\hat{z}_2} V(\hat{z}_2) & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & (\frac{\partial g(z)}{\partial z})^2\Big|_{z=\hat{z}_n} V(\hat{z}_n) \end{pmatrix},$$

with variance function $V(z)$.

The variance function $V(z)$ is related to the variance of the response Y. To more clearly define this relation, we assume that the response Y has a distribution of the form described in McCullagh and Nelder (1989)

$$f_Y(y; \theta, \phi) = \exp\left[(y\theta - b(\theta))/a(\phi) + c(y, \phi)\right],$$

with known functions $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$. θ is the canonical parameter and ϕ is the dispersion parameter.

As defined in McCullagh and Nelder (1989), the variance function is then related to the variance of the response in the following way

$$\mathrm{Var}(Y) = b^{''}(\theta)a(\phi) = V(g^{-1}(\boldsymbol{X}\beta^0))a(\phi).$$

We rewrite, $Y_w = \sqrt{\boldsymbol{W}} Y_{adj}$ and $X_w = \sqrt{\boldsymbol{W}}\boldsymbol{X}$ to get

$$\hat{\beta}_{new} = \mathrm{argmin}_{\beta}(Y_w - \boldsymbol{X}_w\beta)^T(Y_w - \boldsymbol{X}_w\beta).$$

The linear model methods can now be applied to Y_w and $\boldsymbol{X}_w$. Thereby, the estimate $\hat{\sigma}_\varepsilon$ has to be set to the value 1.

REFERENCES

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. *The Annals of Statistics*, 29:1165–1188.

Breiman, L. (1996a). Bagging predictors. *Machine Learning*, 24:123–140.

Breiman, L. (1996b). Heuristics of instability and stabilization in model selection. *Annals of Statistics*, 24:2350–2383.

Bühlmann, P. (2013). Statistical significance in high-dimensional linear models. *Bernoulli*, 19:1212–1242.

Bühlmann, P., Kalisch, M., and Meier, L. (2014a). High-dimensional statistics with a view towards applications in biology. *Annual Review of Statistics and its Applications*, 1:255–278.

Bühlmann, P. and Mandozzi, J. (2014). High-dimensional variable screening and bias in subsequent inference, with an empirical comparison. *Computational Statistics*, 29(3-4):407–430.

Bühlmann, P., Meier, L., and van de Geer, S. (2014b). Invited discussion on ”a significance test for the lasso (R. Lockhart, J. Taylor, R. Tibshirani and R. Tibshirani)”. *Annals of Statistics*, 42:469–477.

Bühlmann, P. and van de Geer, S. (2011). *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer.

Chatterjee, A. and Lahiri, S. (2013). Rates of convergence of the adaptive LASSO estimators to the oracle distribution and higher order refinements by the bootstrap. *Annals of Statistics*, 41:1232–1259.

Fan, J. and Lv, J. (2010). A selective overview of variable selection in high dimensional feature space. *Statistica Sinica*, 20:101–148.

Hartigan, J. (1975). *Clustering algorithms*. Wiley.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. Springer, New York, second edition.

Javanmard, A. and Montanari, A. (2013). Confidence intervals and hypothesis testing for high-dimensional regression. To appear in Journal of Machine Learning Research. Preprint arXiv:1306.3171.

Knight, K. and Fu, W. (2000). Asymptotics of Lasso-type estimators. *The Annals of Statistics*, 28:1356–1378.

Leeb, H. and Pötscher, B. (2003). The finite-sample distribution of post-model-selection estimators and uniform versus non-uniform approximations. *Econometric Theory*, 19:100–142.

Liu, H. and Yu, B. (2013). Asymptotic properties of lasso+mls and lasso+ridge in sparse high-dimensional linear regression. *Electronic Journal of Statistics*, 7:3124–3169.

Lockhart, R., Taylor, J., Tibshirani, R. J., and Tibshirani, R. (2014). A significance test for the lasso. *Annals of Statistics*, 42.

Mandozzi, J. and Bühlmann, P. (2013). Hierarchical testing in the high-dimensional setting with correlated variables. Preprint arXiv:1312.5556v1.

McCullagh, P. and Nelder, J. A. (1989). *Generalized linear models.* London England Chapman and Hall 1983., second edition.

Meier, L., Meinshausen, N., and Dezeure, R. (2014). *hdi: High-Dimensional Inference.* R package version 0.1-2.

Meinshausen, N. (2008). Hierarchical testing of variable importance. *Biometrika*, 95(2):265–278.

Meinshausen, N. (2013). Group-bound: confidence intervals for groups of variables in sparse high-dimensional regression without assumptions on the design. *Preprint arXiv:1309.3489*.

Meinshausen, N. and Bühlmann, P. (2006). High-dimensional graphs and variable selection with the Lasso. *Annals of Statistics*, 34:1436–1462.

Meinshausen, N. and Bühlmann, P. (2010). Stability Selection (with discussion). *Journal of the Royal Statistical Society Series B*, 72:417–473.

Meinshausen, N., Meier, L., and Bühlmann, P. (2009). P-values for high-dimensional regression. *Journal of the American Statistical Association*, 104:1671–1681.

Reid, S., Tibshirani, R., and Friedman, J. (2013). A study of error variance estimation in lasso regression. Preprint arXiv:1311.5274v1.

Shah, R. and Samworth, R. (2013). Variable selection with error control: another look at Stability Selection. *Journal of the Royal Statistical Society Series B*, 75:55–80.

Shao, J. and Deng, X. (2012). Estimation in high-dimensional linear models with deterministic design matrices. *The Annals of Statistics*, 40:812–831.

Sun, T. and Zhang, C.-H. (2012). Scaled sparse linear regression. *Biometrika*, 99:879–898.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society Series B*, 58:267–288.

van de Geer, S. (2007). The deterministic Lasso. In *JSM proceedings, 2007, 140*. American Statistical Association.

van de Geer, S. and Bühlmann, P. (2009). On the conditions used to prove oracle results for the Lasso. *Electronic Journal of Statistics*, 3:1360–1392.

van de Geer, S., Bühlmann, P., Ritov, Y., and Dezeure, R. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *Annals of Statistics*, 42:1166–1202.

Wasserman, L. and Roeder, K. (2009). High dimensional variable selection. *Annals of Statistics*, 37:2178–2201.

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

Zhang, C.-H. and Zhang, S. S. (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):217–242.