



Hierarchical inference for genome-wide association studies: a view on methodology with software

Claude Renaux¹ · Laura Buzdugan¹ · Markus Kalisch¹ · Peter Bühlmann¹

© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract

We provide a view on high-dimensional statistical inference for genome-wide association studies. It is in part a review but covers also new developments for meta analysis with multiple studies and novel software in terms of an R-package `hierinf`. Inference and assessment of significance is based on very high-dimensional multivariate (generalized) linear models: in contrast to often used marginal approaches, this provides a step towards more causal-oriented inference.

1 Introduction

We provide a selective review (or a view) on high-dimensional statistical inference for genome-wide association studies (GWAS). In doing so, we give an illustration of our own software in terms of the R-package `hierinf` and we also include some novel methodological aspects and results. Both of the mentioned topics, high-dimensional inference and GWAS, have been rapidly evolving over the last years and we do not aim here to present a broad overview. Instead, we focus on the combination of the two and consider inference in a multivariate model which quantifies effects after adjusting for all remaining single nucleotide polymorphism (SNP) covariates. Assigning uncertainties in such a multiple regression model has received fairly little attention so far, perhaps because of the difficulty to deal in practice with the very high-dimensionality in GWAS with $p \approx O(10^6)$ SNP covariates.

Univariate approaches for significance of a SNP being marginally associated to a response variable (sometimes denoted as phenotype) have been widely adopted in the last decades. The main challenge with such marginal approaches is the multiple

This invited paper is discussed in comments available at: <https://doi.org/10.1007/s00180-019-00940-9>; <https://doi.org/10.1007/s00180-019-00940-9>; <https://doi.org/10.1007/s00180-019-00940-9>; <https://doi.org/10.1007/s00180-019-00940-9>.

✉ Peter Bühlmann
peter.buehlmann@stat.math.ethz.ch

¹ Seminar for Statistics, ETH Zürich, Zürich, Switzerland

testing adjustment: the false discovery rate (FDR) (Benjamini and Hochberg 1995) has become very popular as an error criterion which is less conservative than the familywise error rate (FWER), see for example Storey and Tibshirani (2003), Sabatti et al. (2003), Benjamini and Yekutieli (2005). Peterson et al. (2016) consider a hierarchical formulation for the FDR: their hierarchical procedure is very different though than the hierarchical inference scheme which we propagate in this article [in their work, the hierarchy originates from having multiple phenotypes; in contrast, the hierarchy in our approach addresses the issue of highly correlated covariates in a (generalized) linear model].

There are several proposals which consider a multivariate regression model. Baierl et al. (2006) consider model selection in low-dimensional QTL modeling which is not of high-dimensional nature. Frommlet et al. (2012) and Dolejsi et al. (2014) have further developed the methodology from Baierl et al. (2006) and applied it to real GWAS data, i.e. in the high-dimensional context. Furthermore for GWAS, Bayesian approaches (Hoggart et al. 2008; Carbonetto and Stephens 2012), Ridge regression (Malo et al. 2008) or the Lasso for screening important covariates (including interactions) (Wu et al. 2010a, b) have been considered and used, further proposals include a combination of the Lasso and linear mixed models (Rakitsch et al. 2013; Zhou et al. 2013), the Bayesian Lasso (Li et al. 2011) or stability selection for sparse estimators (Alexander and Lange 2011; He and Lin 2011). None of these proposals compute frequentist p values for single or groups of SNPs, but methods based on stability selection lead to control of the number of false positives (Meinshausen and Bühlmann 2010). More recently, interesting work has been pursued for control of the FDR after selection in a multivariate regression model (Brzyski et al. 2017). The proposed procedure is first pre-screening for the level of resolution to identify regions or groups of SNPs and then controls the FDR for the pre-screened first-stage regions, see also Heller et al. (2017) when using marginal tests. In a sense, the work by Brzyski et al. (2017) comes closest to our proposal with a hierarchical structure: both approaches share the point that for GWAS, data-driven aggregation of hypotheses results in more power. For an overview of univariate and multivariate methods which have been published until the year of 2015, we also refer to the monograph by Frommlet et al. (2016).

We have recently proposed high-dimensional hierarchical inference for assigning statistical significance in terms of p values for groups of SNPs being associated to a response variable: Buzdugan et al. (2016) considers this approach for human GWAS and Klasen et al. (2016) for GWAS with plants. The methodological and theoretical concepts have been worked out in Mandozzi and Bühlmann (2016a) and Mandozzi and Bühlmann (2016b). The hierarchy enables in a fully data-driven way to infer significant groups or regions of SNPs at an adaptive resolution, by controlling the familywise error rate (FWER). Although the FWER seems overly conservative, we still detect small groups of SNPs in real datasets. We will review this approach and extend it to the setting of multiple studies using concepts of meta-analysis. The difference to pre-screening and selection techniques as in e.g. Brzyski et al. (2017) is that we do not need to choose the amount of pre-screening at the beginning: the entire procedure is fully data-driven, leading to high resolution (small groups of SNPs) if the signal is strong in relation to the strength of correlation among the SNPs and vice-versa yielding low resolution if the signal is weak. The hierarchy itself is constructed by either clustering

the SNPs according to their strength of squared correlations or by partitioning of the genomic sequence into blocks of consecutive genomic positions corresponding to groupings of the SNPs. Our procedure is based on an efficient hierarchical multiple testing adjustment from Meinshausen (2008). The power of a sequential approach for controlling the FWER has been thoroughly discussed in Goeman and Solari (2010). The scheme by Meijer et al. (2015) could be an interesting alternative when looking at region-based groups of SNPs: it is more flexible at the price of a higher multiple testing adjustment and thus, it is unclear whether it would exhibit more power.

2 High-dimensional hierarchical inference

We build the statistical inference on a multiple regression model where all the measured SNPs enter as covariates in the model. We will mainly focus on a linear model:

$$Y = \mu + \mathbf{X}\beta^0 + \varepsilon, \quad (1)$$

with $n \times 1$ response vector Y , $n \times p$ design matrix \mathbf{X} and $n \times 1$ vector of stochastic errors ε and intercept μ . The superscript “0” denotes the “true” underlying parameter of the data-generating distribution. We usually assume fixed design and i.i.d. errors with $\mathbb{E}[\varepsilon_i] = 0$, $\text{Var}(\varepsilon_i) = \sigma^2$. We denote the i th row and the j th column of \mathbf{X} by X_i and $X^{(j)}$, respectively. The assumption about fixed design is not really a loss of generality as long as the linear model is correct: if the covariates are random, we can always condition on \mathbf{X} (and the linear model is still correct) and perform the statistical inference conditional on \mathbf{X} .

Genome-wide association study In a GWAS, the covariates corresponding to the columns of \mathbf{X} are the SNPs. The response variable can be continuous like e.g. a growth rate of a plant or binary encoding the status “healthy” or “diseased” (see Sect. 3.4 for some examples). For the latter, we would then consider a logistic regression model as in (2). In general, the sample size is about $n \approx 3'000$ whereas the number of SNP covariates is in the order of $p \approx 10^6$. Obviously the model in (1) in the setting of GWAS is very high-dimensional with many more unknown parameters than sample size, i.e., $p \gg n$.

We note that the multiple model in (1) is very different from a marginal model

$$Y = \mu_j + \gamma_j X^{(j)} + \tilde{\varepsilon}^{(j)},$$

where the response variable is modeled for every covariate $X^{(j)}$ individually. The marginal model does not take into account how much of an effect is due to other covariates (i.e., the marginal effect is not adjusted for other covariates), and the multiple regression model is much more powerful towards causal inference as discussed in Sect. 2.4.

The model has to be adapted if the response is not continuous. The response could be a binary variable encoding a disease status, e.g. if a patient has diabetes or is healthy. We can extend the methodology to generalized linear models of the form

Y_i independent with,

$$\eta_i = g(\mathbb{E}[Y_i]) = \mu + \sum_{j=1}^p \beta_j^0 X_i^{(j)}, \quad (2)$$

where $g(\cdot)$ is a real-valued link function. The most prominent example which we will use is a logistic regression model, where $Y_i \in \{0, 1\}$ is binary, $\pi_i = \pi_i(X_i) = \mathbb{P}[Y_i = 1|X_i]$ and link function $g(\pi) = \log(\pi)/\log(1 - \pi)$. We will illustrate such an extension for GWAS analysis in Sect. 3.4.

In the sequel, for simplicity, we usually consider a linear model. The extension of the methodology and computations for generalized linear models is straightforward and the case of a logistic model is implemented in our software `hierinf`, which is an R package available on bioconductor, as described in Sect. 3. The theoretical results which we review in the subsequent sections carry over to generalized linear models: the underlying analysis is more delicate though, see for example Bühlmann and van de Geer (2011).

2.1 High-dimensional inference

A first goal is to infer the very many unknown regression parameters β^0 in (1) or (2), respectively. This means that we are interested in estimating the regression coefficients in one of the afore mentioned models. A next important aim is to perform statistical hypothesis testing, which is described in Sect. 2.2.

Because of the high-dimensionality of the problem at hand, the estimated regression parameters $\hat{\beta}$ are regularized and enforced to be sparse, i.e. many of its components are equal to zero. We restrict ourselves for the moment to the case of a linear model (1). The Lasso (Tibshirani 1996) has become a very popular tool for point estimation:

$$\hat{\beta}(\lambda) = \operatorname{argmin}_{\beta} (\|Y - \mathbf{X}\beta\|_2^2/n + \lambda\|\beta\|_1), \quad (3)$$

where $\lambda > 0$ is a regularization parameter which needs to be chosen. The Euclidean or L_2 norm is denoted by $\|\cdot\|_2$ and the Manhattan or L_1 norm by $\|\cdot\|_1$. The first term in the above equation, the sum of squared residuals, is identical to the case of least squares estimation for a low-dimensional regression problem. The sum of squared residuals is divided by the number of observations n in order to achieve a proper scaling but it does not change the methodology. The second term penalizes the size of the regression parameters: because of the “geometry” of the L_1 -norm, Lasso is a sparse estimator with many components being exactly equal to zero (depending on the value of λ).

The columns or covariables of the $n \times p$ design matrix \mathbf{X} are denoted as before by $X^{(j)}$ with $j = 1, \dots, p$. Here, the Y response and all the covariates $X^{(j)}$ are assumed to be mean centered so that the intercept μ can be dropped from the model. This is a convenient way to estimate the unknown parameters β^0 . Furthermore, the Lasso (3) usually makes most sense if all the covariates are on the same scale, as implemented per default in the R-package `glmnet` (Friedman et al. 2010). The penalty term penalizes all the variables with the same amount which only makes sense if they are standardized. For GWAS, the SNP covariates take values in $\{0, 1, 2\}$ (minor allele

frequencies) and are treated as numerical values: since they are “on the same scale”, we do not standardize them to equal standard deviation. Treating them as numerical or continuous rather than categorical or ordinal variables has the advantage of using only one parameter for each SNP covariate, whereas a categorical approach with main effects or full interactions would require $2 \cdot p$ or $3^p - 1$ parameters, respectively. Using a continuous scale modeling for SNPs is a rather common approach, see for example Cantor et al. (2010) or Bush and Moore (2012). This means that we are searching for additive effects. One typically has reasonable power to detect additive and dominant effects whereas for recessive effects the study might be underpowered (Bush and Moore 2012).

Throughout the paper, whenever we will make some asymptotic statements, they are meant to be that the dimension p as well as the sample size n tend to infinity, i.e., we adopt a “changing model” (sometimes called “triangular array”) asymptotics. That is, the dimension $p = p_n$ and the model parameters $\beta = \beta_n$, $\mu = \mu_n$ and $\sigma = \sigma_n$ (for linear models) depend on n and typically the ratio $p_n/n \rightarrow \infty$ as $n \rightarrow \infty$.

2.1.1 Statistical properties of the Lasso

An executive summary The statistical properties of the Lasso in (3) have been extensively studied during the last decade. The Lasso is a nearly optimal method for prediction and parameter estimation when making the main assumptions on sparsity of the parameter vector [assumption (A1) below] and identifiability in terms of “well-posedness” of the design matrix [assumption (A2) below]. For accurate selection of the active set of variables (having non-zero regression coefficients), one necessarily needs a “beta-min” condition [assumption (A3) below] which requires that the non-zero regression coefficients are sufficiently large. In addition, one would necessarily need a rather strong irrepresentable condition on the design matrix: this can be avoided guaranteeing instead a variable screening property. The latter is most useful in practice, and in fact a standard workhorse in many applications, allowing to screen for the important variables and achieving a drastic dimensionality reduction in terms of the original variables.

The two main assumptions leading to good or near optimal properties of the Lasso for (point) estimation of β^0 are a sparsity assumption on the parameter vector β^0 and an identifiability assumption on the design \mathbf{X} . The Lasso itself is a sparse estimator and hence it is expected that it leads to good performance if the true underlying parameter β^0 is sparse as well: the support of β^0 , sometimes also called the active set, is denoted by

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

and we will assume that its cardinality $s_0 = |S_0|$ is smaller than $\text{rank}(\mathbf{X}) \leq n$. Regarding identifiability, since $\text{rank}(\mathbf{X}) \leq n < p$, the null-space of \mathbf{X} is not trivial and we can write

$$\mathbf{X}\beta^0 = \mathbf{X}\theta \text{ for } \theta = \beta^0 + \xi \text{ with any } \xi \text{ in the null-space of } \mathbf{X}.$$

Thus, in order to estimate β^0 we must make an additional identifiability assumption on the design \mathbf{X} which again relies on sparsity with a not too large set S_0 .

The main assumptions are as follows:

(A1) Sparsity: The cardinality of the support or active set of β^0 satisfies

$$s_0 = |S_0| = o(a_n), \quad a_n \rightarrow \infty,$$

with typical values being $a_n = n/\log(p)$ or $a_n = \sqrt{n/\log(p)}$, see for example Bühlmann and van de Geer (2011, Eq. (2.22)).

(A2) Compatibility condition (van de Geer 2007): An identifiability assumption on the design \mathbf{X} .

For some $\phi_0 > 0$ and for all β satisfying $\|\beta_{S_0^c}\|_1 \leq 3\|\beta_{S_0}\|_1$ it holds that

$$\|\beta_{S_0}\|_1^2 \leq (\beta^\top \hat{\Sigma} \beta) s_0 / \phi_0^2,$$

where $\hat{\Sigma} = \mathbf{X}^\top \mathbf{X} / n$ and β_S , for an index set $S \subseteq \{1, \dots, p\}$, has elements set to zero outside the set S , i.e., $(\beta_S)_j = 0$ ($j \notin S$) and $(\beta_S)_j = \beta_j$ ($j \in S$). The value $\phi_0 > 0$ is called the compatibility constant.

Assuming conditions (A1) and (A2) (with the compatibility constant ϕ_0) one can establish an oracle inequality of the following form, see for example Bühlmann and van de Geer (2011, Th.6.1). Consider a linear model as in (1) with fixed design \mathbf{X} , Gaussian or sub-Gaussian errors ε and when using the Lasso (3) with regularization parameter $\lambda \asymp \sqrt{\log(p)/n}$:

$$\|\mathbf{X}(\hat{\beta}(\lambda) - \beta^0)\|_2^2/n + \lambda \|\hat{\beta}(\lambda) - \beta^0\|_1 \leq O_P(\lambda^2 s_0 / \phi_0^2).$$

The parameter λ cannot be chosen smaller than of the order $\sqrt{\log(p)/n}$ since otherwise, the probability in the “ $O_P(\cdot)$ ” notation would not become large and the statement would not hold anymore. When choosing $\lambda \asymp \sqrt{\log(p)/n}$ and assuming that the compatibility constant $\phi_0 \geq L > 0$ is bounded away from zero, we obtain for

$$\text{prediction: } \|\mathbf{X}(\hat{\beta} - \beta^0)\|_2^2/n \leq O_P(s_0 \log(p)/n), \tag{4}$$

$$\text{parameter estimation: } \|\hat{\beta} - \beta^0\|_1 \leq O_P(s_0 \sqrt{\log(p)/n}). \tag{5}$$

Here, we have dropped the dependence of $\hat{\beta}$ on λ . The second statement is more relevant for inferring the true underlying β^0 . In particular, it is straightforward to derive a screening property as discussed next.

The Lasso, being a sparse estimator, is often used as a variable selection and screening tool. We denote by

$$\hat{S}(\lambda) = \{j; \hat{\beta}_j(\lambda) \neq 0\}.$$

The aim would be that $\hat{S} \approx S_0$, which is a highly ambitious goal (see below). Clearly, to infer the active set from data, the regression coefficients in S_0 must be sufficiently large. This can be ensured by an additional “beta-min” condition:

$$(A3) \min\{|\beta_j^0|; \beta_j^0 \neq 0\} = \min_{j \in S_0} |\beta_j^0| \geq C(s_0, p, n), \text{ where } C(s_0, p, n) \asymp \sqrt{s_0 \log(p)/n}.$$

Assuming (A1), a slightly stronger version than (A2) in terms of a restricted eigenvalue condition (Bickel et al. 2009), and (A3), we have the following screening result. For a linear model as in (1) with fixed design \mathbf{X} , Gaussian or sub-Gaussian errors ε and when using the Lasso (3) with regularization parameter $\lambda \asymp \sqrt{\log(p)/n}$:

$$\mathbb{P}[\hat{S} \supseteq S_0] \rightarrow 1 \text{ (} p \geq n \rightarrow \infty \text{)}. \tag{6}$$

When using the weaker compatibility condition (A2), we would then require the beta-min condition with a larger $C(s_0, p, n) \asymp s_0 \sqrt{\log(p)/n}$. This is an immediate consequence of (5).

The variable screening property is a highly efficient dimension reduction technique in terms of the original covariates. Because it holds that $|\hat{S}(\lambda)| \leq \min(n, p)$ for all λ [assuming (A1) and (the weaker version) of (A2)], and the latter equals n in the high-dimensional regime with $p \gg n$, we can greatly reduce the dimension without losing an active variable from \hat{S}_0 . Obviously, it would be even better if variable selection would consistently estimate the true underlying active set,

$$\mathbb{P}[\hat{S}(\lambda) = S_0] \rightarrow 1 \text{ (} p \geq n \rightarrow \infty \text{)}.$$

However, such a consistent variable selection property necessarily requires a much stronger so-called irrepresentable condition on the design \mathbf{X} than the assumption in (A2) (Meinshausen and Bühlmann 2006; Zou 2006; Zhao and Yu 2006).

Practical considerations For the task of inference described in Sect. 2.2.1 below, we aim for a regularization parameter λ such that the screening property (6) holds, i.e., that $\hat{S}(\lambda) \supseteq S_0$ holds in a reliable way. Choosing the regularization parameter by cross-validation (by default 10-fold CV), denoted by λ_{CV} typically leads to a good set $\hat{S}(\lambda_{CV})$ in comparison to other values of λ . It isn't true that there is a monotone relationship between λ and $\hat{S}(\lambda)$ and thus, a smaller value $\lambda < \lambda_{CV}$ does not necessarily lead to a superset $\hat{S}(\lambda) \supseteq \hat{S}(\lambda_{CV})$. Bühlmann and Mandozzi (2014) illustrate the success of various variable screening methods with respect to true and false positives, without considering the issue of choosing a good regularization parameter: overall, the Lasso leads to a competitive performance in comparison to other methods. It is not so unlikely though that the property $\hat{S} \supseteq S_0$ can be rather far from being entirely correct: it is rare that all of the variables in S_0 are contained in $\hat{S}(\lambda)$ but hopefully a reasonable good sized fraction of S_0 is contained in the set \hat{S} from the Lasso.

The assumptions in the context of GWAS We discuss here whether the theoretical assumptions hold at least approximately in the context of GWAS. The assumption (A1) is about sparsity: it is a speculation whether the true underlying biological phenomena are sparse: the model is always a simplification and a best sparse approximation, achieved by the Lasso, is often still very useful. More details about best sparse approximation properties and weak sparsity are given in Bühlmann and van de Geer (2011) and van de Geer (2016). Assumption (A2) can be justified as follows: assume that the

covariates are i.i.d. sampled from a population distribution with covariance matrix Σ , having smallest eigenvalue bounded away from zero. Then, if the population distribution is e.g. sub-Gaussian, the condition (A2) holds with high probability for sparse sets S_0 (Bühlmann and van de Geer 2011, Cor.6.8). It seems quite plausible that the population distribution in a GWAS context has spatially decaying covariance behavior such that the smallest eigenvalue is bounded away from zero, e.g. for a Toeplitz matrix model. The main assumption is again sparsity for the set S_0 as in (A1). Assumption (A3) is severe and not realistic to hold exactly in many applications: however, it is not required for (4) and (5), and it can be avoided also for hypothesis testing as pointed out in Sect. 2.2.2: our proposed multi sample splitting procedure in this paper has no theoretical guarantees without (a weaker form of) (A3) but thanks to multiple sample splitting and averaging, it still performs empirically reasonably well in absence of condition (A3), see for example Dezeure et al. (2015).

2.2 Statistical hypothesis testing

Our main goal is to provide p values for statistical hypothesis tests. We consider the following null and alternative hypotheses for the regression parameters in the model (1) or (2). For individual variables

$$H_{0,j} : \beta_j^0 = 0 \text{ versus } H_{A,j} : \beta_j^0 \neq 0,$$

or for a group $G \subseteq \{1, \dots, p\}$ of variables:

$$H_{0,G} : \beta_j^0 = 0 \text{ for all } j \in G \text{ versus } H_{A,G} : \text{there exists } j \in G \text{ with } \beta_j^0 \neq 0. \quad (7)$$

The challenge is to construct p values in the very high-dimensional setting with $p \gg n$ which control the error rate of falsely rejecting the null-hypothesis (the type I error rate). There is also a computational difficulty involved and the methods from Sect. 2.2.2 are not feasible in the context of GWAS with $p \approx 10^6$ covariates. And finally, there is the issue of multiple testing: this is addressed in Sect. 2.3 advocating a very powerful hierarchical approach.

2.2.1 Multi sample splitting and aggregation of p-values

An executive summary Sample splitting and its improved version of multiple sample splitting (Meinshausen et al. 2009) is rather straightforward and, as a modular technique, it is easy to implement. It is justified to yield valid p values which control (possibly conservatively) the type I error rate under the assumptions (A1)–(A3): while (A1)–(A2) are essentially unavoidable, the beta-min assumption (A3) is rather unpleasant since the p value or statistical test itself is a method to investigate whether a regression coefficient is “smallish” or sufficiently large [while (A3) is simply assuming the latter]. However, the method has been empirically found to be rather reliable to control the type I error rate and yet having often reasonable power (Dezeure et al. 2015) to detect a variety of alternative hypotheses. From a computational view point,

the procedure is scaling very nicely for very high-dimensional problems making it feasible to be used for GWAS with $p \approx 10^6$.

The idea of the procedure is as follows. We do variable screening with an estimated set of variables \hat{S} such that (6) holds, at least in an approximate sense. We can then use standard low-dimensional inference methods based on the selected variables from \hat{S} only. To avoid to use the data twice for screening and inference, we split the dataset into two halves: select or screen for variables in the first half and pursue the inference on the second remaining part of the dataset. This procedure is implicitly given in the work by Wasserman and Roeder (2009).

Sample splitting for p values

1. Randomly split the sample into two parts of equal size. Denote the corresponding indices by I_1, I_2 with $I_i \subset \{1, \dots, n\}$ ($i = 1, 2$) such that $I_1 \cap I_2 = \emptyset$, $I_1 \cup I_2 = \{1, \dots, n\}$ and $|I_1| = \lfloor n/2 \rfloor$, $|I_2| = n - \lfloor n/2 \rfloor$.
2. Do variable selection or screening with the Lasso based on data with samples from I_1 : denote the selected variables by \hat{S}_{I_1} . (The Lasso can be used for linear or also generalized linear models). We use the regularization such that \hat{S}_{I_1} consists of the first $\lfloor n/6 \rfloor$ variables entering in the Lasso regularization path.
3. Derive p values for individual or group hypotheses based on data with covariates from \hat{S}_{I_1} and samples from I_2 . Since $|\hat{S}_{I_1}| = \lfloor n/6 \rfloor$ and assuming that $\text{rank}(\mathbf{X}_{I_2, \hat{S}_{I_1}}) = |\hat{S}_{I_1}| = \lfloor n/6 \rfloor$ we can use classical techniques based on least squares or likelihood ratio testing.

For the linear regression model (1) we use for a single variable $j \in \{1, \dots, p\}$,

if $j \in \hat{S}_{I_1}$: p-value P_j from the two-sided t-test for $H_{0,j}$ based on $(Y_{I_2}, \mathbf{X}_{I_2, \hat{S}_{I_1}})$;

if $j \notin \hat{S}_{I_1}$: set $P_j = 1$.

Similarly, for a group $G \subseteq \{1, \dots, p\}$,

if $G \cap \hat{S}_{I_1} \neq \emptyset$: p-value P_G from the partial F-test for $H_{0, \tilde{G}}$,

where $\tilde{G} = G \cap \hat{S}_{I_1}$;

if $G \cap \hat{S}_{I_1} = \emptyset$: set $P_G = 1$.

For a generalized linear model (2) we use the likelihood ratio test instead of the t- or partial F-test.

The sample splitting method is valid and controls the type I error if the screening property $\hat{S} \supseteq S_0$ holds. This is due to the fact that we have all the relevant variables in the model in the second inference step based on data from I_2 . The requirement for the screening property can be a bit relaxed as analyzed in Bühlmann and Mandozzi (2014), allowing also for not too many small non-zero regression coefficients.

Note that if the intersection between a given group G and the selected set of variables with Lasso \hat{S}_{I_1} based on a half-sample is empty, then the p value is set to the value one. For some given large group, the intersection between this group and the selected set of variables from Lasso has cardinality at most equal to $|\hat{S}_{I_1}|$ which is bounded by

the half-sample size $\lfloor n/2 \rfloor$. In particular, not all the variables of such a group G are considered for calculating the p value in the other half-sample I_2 . This works fine since we assume that the screening property $\hat{S}_{I_1} \supseteq S_0$ of the Lasso holds, which implies that we control for all the relevant variables.

Unfortunately, sample splitting very much depends on how the dataset is split into two parts, e.g., the random choice of partitioning the data into two groups. To avoid this dependence on how the dataset is split, one can do the sample splitting and inference procedure many times (e.g. 100 times) and then aggregate the corresponding p values in a way so that the type I error is controlled. This aggregation step requires special attention and is detailed below in (8). The method has been invented by Meinshausen et al. (2009) and works as follows.

Multiple sample splitting for p -values The multiple sample splitting approach uses the steps 1.-3. from the sample splitting procedure above B times. For a group of variables G , including the case of $G = \{j\}$ being a singleton, this leads to B p values

$$P_G^{(1)}, \dots, P_G^{(B)}.$$

The question is how to aggregate these B p values to a single one such that the type I error rate is still controlled. In particular, since the B p values arise from different random splits of the data, they are dependent, and we thus need to develop a method to aggregate arbitrarily dependent p values. This can be done by the following rule:

$$P_G = \min \left(1, \log(1 - \gamma_{\min}) \inf_{\gamma \in (\gamma_{\min}, 1)} Q_G(\gamma) \right),$$

$$Q_G(\gamma) = q_\gamma \left(\{P_G^{(b)} / \gamma; b = 1, \dots, B\} \right), \tag{8}$$

where $q_\gamma(\{P_G^{(b)} / \gamma; b = 1, \dots, B\})$ is the γ -quantile of the B p values multiplied by $1/\gamma$. The factor $\log(1 - \gamma_{\min})$ guarantees to adjust for the fact that we are searching the smallest quantiles over the range $(\gamma_{\min}, 1)$.

As argued for the single sample splitting procedure, the multiple sample splitting method is valid if the screening property $\hat{S} \supseteq S_0$ holds. Thus, for asymptotic validity in terms of controlling the type I error, we require the screening property as in (6). This itself holds for the Lasso under the assumptions (A1)–(A3) discussed in Sect. 2.1.1. In particular, this approach calls for a beta-min assumption as in (A3) which is somewhat unpleasant: the p value or statistical test should quantify to what extent a regression parameter is “smallish” or “sufficiently large” while the beta-min assumption is simply assuming that there are no “smallish nonzero” coefficients. A slight relaxation of the screening property is discussed in Bühlmann and Mandozzi (2014), allowing for not too many small non-zero true regression coefficients.

From a computational point of view, the method requires the computational cost $O(np \min(n, p))$ for screening the variables with the Lasso and then at most $O(n|\hat{S}|^2)$ for inference based on the selected variables: thus, for $p \gg n$ and since $|\hat{S}| \leq n$, the total computational cost is $O(Bn^2 p)$ which is linear in the dimensionality p . We typically take $B = 100$ and parallel implementation over the B repetitions can easily be

done. The main cost is fitting a Lasso regression for variable screening in the setting where p is very large and n is a substantial number. Computational speed-ups for the Lasso using random projections (in sample space) have been recently proposed (Pilanci and Wainwright 2015) and might be useful in practice; similarly, computationally fast Ridge regression (Lu et al. 2013) and thresholding (Shao and Deng 2012) could be used for reasonably accurate screening, though perhaps a bit worse than Lasso (Bühlmann and Mandozzi 2014).

2.2.2 Other methods

Other methods which do not require a beta-min assumption can be used for statistical hypothesis testing: for a comparison, see Dezeure et al. (2015). The most prominent example is perhaps the de-biased or de-sparsified Lasso estimator proposed by Zhang and Zhang (2014) and further analyzed in van de Geer et al. (2014); a related technique has been proposed in Javanmard and Montanari (2014). A Ridge projection method (Bühlmann 2013) is another option, often leading to more conservative inferential statements.

Bootstrapping the Lasso or versions of it has been proposed in Chatterjee and Lahiri (2011, 2013), Liu and Yu (2013) but due to the sparsity of the underlying estimator, these approaches are exposed to the super-efficiency phenomenon (i.e. estimation of parameters being equal to zero is very accurate while it can be very poor for non-zero components). Bootstrapping the de-biased Lasso estimator, where super-efficiency does not occur, has been analyzed in Dezeure et al. (2017). A very different resampling strategy for obtaining p values for rather general hypotheses about “goodness of fit” has been proposed in Shah and Bühlmann (2018).

Finally, one can use “stability selection” for obtaining statistical error measures (Meinshausen and Bühlmann 2010; Shah and Samworth 2013): it is a very generic subsampling technique but does not lead to rigorous p values corresponding to the hypothesis in (7) as we require it here.

2.3 Hierarchical inference

An executive summary Hierarchical inference is a key technique for computationally and statistically efficient hypothesis testing and multiple testing adjustment. It provides a convincing way to address the main problems occurring in high-dimensional scenarios. First, due to high pairwise absolute empirical correlation between covariates, or near linear dependence among a small set of covariates, one cannot (or at least not sufficiently well) identify single regression coefficients β_j^0 . However, the problem is much better posed if we ask for identifying whether there is an association between a group of variables $G \subseteq \{1, \dots, p\}$ and a response, i.e., to test a group hypothesis as in (7). Hierarchical inference is a method for sequentially testing many such group hypotheses, thereby automatically adapting to the “resolution level” without the need to pre-specify the precise form or size of the groups.

The hierarchy for the inference is described in terms of a tree \mathcal{T} where each node corresponds to a group $G(\subseteq \{1, \dots, p\})$ and a group hypothesis $H_{0,G}$: the hierarchical

constraint means that for a node (or group) G , any descendant node G' must satisfy $G' \subset G$. Furthermore, we require that the child nodes of G (the direct descendants of G) build a partition of G . The tree \mathcal{T} typically starts with the top node $G_{\text{top}} = \{1, \dots, p\}$ and then branches downward to smaller groups until the p single variable nodes $\{1\}, \dots, \{p\}$ at the bottom of the tree, see Figs. 1 and 2. A typical construction of such a tree is given by hierarchical clustering which results in a binary tree, see at the end of this section.

Given a hierarchical tree \mathcal{T} , the main idea of hierarchical inference is to pursue testing of the groups in a sequential fashion, starting with the top node and then successively moving down the hierarchy until a group doesn't exhibit a significant effect. Figure 2 illustrates this point, showing that we might proceed rather deep in the hierarchy at some parts of the tree whereas at other parts the testing procedure stops due to a group which is not found to exhibit a significant effect. We need some multiple testing adjustment of the p values: interestingly, due to the hierarchical nature, it is not overly severe at the upper parts of the hierarchy as described below.

The procedure works as follows. Denote by P_G the raw p value of the statistical test for the null-hypothesis $H_{0,G}$ versus $H_{A,G}$ defined as in (7). We correct for multiplicity in a simple way:

$$P_{G;\text{adjusted}} = P_G \cdot p/|G|. \tag{9}$$

This corresponds to a depth-wise Bonferroni correction for a balanced tree. Denote by $d(G)$ the level of the tree of the node (or group) G and by $n(G)$ the number of nodes at level $d(G)$: for example, when $G = \{1, \dots, p\}$ corresponds to the top node in a tree containing all variables, we have that $d(G) = 1$ and $n(G) = 1$. If the tree has the same number of offspring (e.g. a binary tree with two offspring throughout the entire tree), we could also use the unweighted version,

$$\text{depth-wise Bonferroni correction: } P_{G;\text{adjusted}} = P_G \cdot n(G), \tag{10}$$

see for example Bühlmann [2017, eq. after eq. (22)]. If in addition the groups would have the same size in each level of depth (up to rounding errors), then the rules in (9) and (10) coincide. The formula (10) is only given here for the sake of interpretation as a depth-wise Bonferroni in the case of balanced trees with the same number of offspring. See also Fig. 1 for an illustration of such a depth-wise Bonferroni correction if the groups are balanced.

The sequential nature with stopping can be formulated in terms of p values by adding a hierarchical constraint:

$$P_{G;\text{hierarchically-adjusted}} = \max_{G' \supset G} P_{G',\text{adjusted}}, \tag{11}$$

implying that once we stop rejecting a node, we cannot reject further down in the tree hierarchy and thus, we can simply stop the procedure when a node is not found as being significant. The main advantage of the procedure is the statistically efficient correction for multiple testing in (9) which is much more powerful than a standard Bonferroni correction over all the nodes in the tree, see also (10).

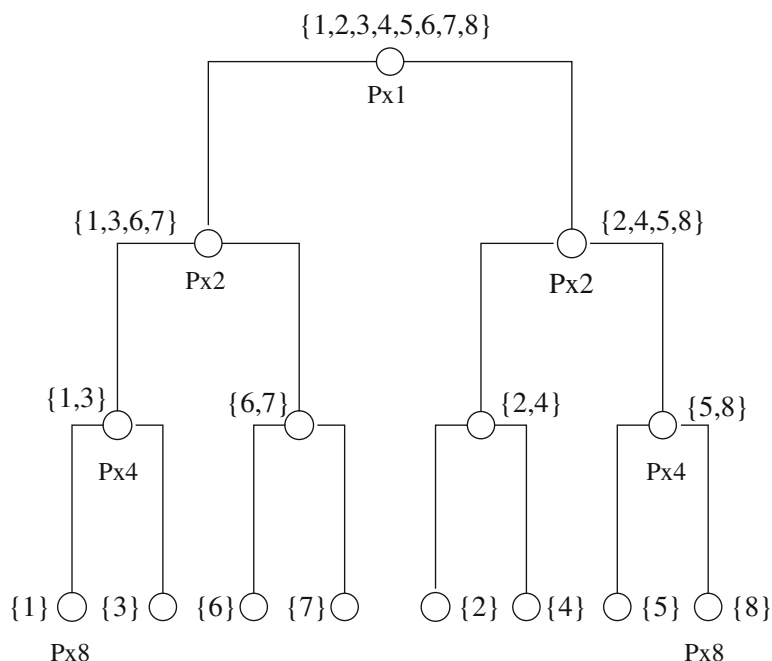


Fig. 1 Hierarchical grouping of 8 variables where different groups are denoted by $\{ \dots \}$. The capital letter “ P ” is a generic notation for the raw p value corresponding to a group hypothesis $H_{0,G}$ of a group G , which is then adjusted as in (10). Since the hierarchy has the same number of offspring throughout the tree, the adjustment is the depth-wise Bonferroni correction which amounts to multiply the p values in every depth of the tree by the number of nodes in the corresponding depth; no multiplicity adjustment at the top node, then multiplication by the factor 2 (depth 2), 4 (depth 3), and 8 (depth 4). The figure is taken from Bühlmann (2017)

The following then holds.

Proposition 1 (Meinshausen 2008) *Consider an arbitrary hierarchy of hypotheses tests in terms of a tree structure \mathcal{T} . Consider the procedure described above with depth-wise adjustment in (9) and with hierarchy constraint as in (11). Then, the familywise error rate (FWER) is controlled: that is, for $0 < \alpha < 1$, when rejecting a hypothesis $H_{0,G}$ if and only if $P_{G;\text{hierarchically-adjusted}} \leq \alpha$, we have that $\text{FWER} = \mathbb{P}[\text{at least one false rejection}] \leq \alpha$.*

The procedure described above and justified in Proposition 1 has a few features to be pointed out. First, it relies on the premise that large groups should be easier to detect and found to be significant, due to the fact that the identifiability is much better posed. We address this issue at the end of this section. In fact, the method has indeed built in the hierarchical constraint (11) that once we cannot reject $H_{0,G}$ for some group G , we do not consider any other sub-groups of G which arise as descendants further down in the tree hierarchy. Due to the sequential nature of the testing procedure, multiple testing adjustment for controlling the familywise error rate is rather mild (for upper parts in the tree) as we only correct for multiplicity at each depth of the tree, i.e., the root node does not need any adjustment, and if it were found to be significant, the next children nodes only need a correction according to the number of nodes at depth 2 of the tree, and similarly for deeper levels; see Fig. 1.

Improvements over the rule in (9) and (11) are possible, based on exploiting the logical relationships among the tests with the Schaffer improvement (Meinshausen

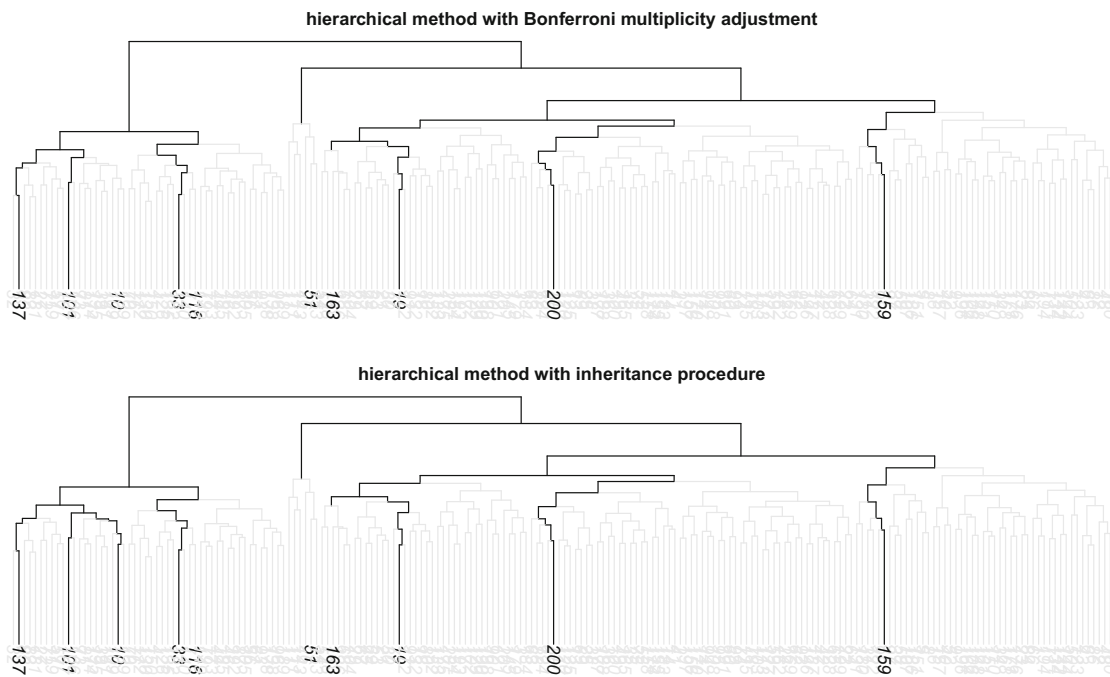


Fig. 2 Hierarchical inference within a tree of clustered variables for a simulated example with $p = 500$ and $n = 100$. The numbers at the bottom in black (bold) denote the indices j of active variables with $\beta_j^0 \neq 0$ (and corresponding to $H_{0,j}$ being false). The black lines graphically encode the significant groups of variables. Top panel: hierarchical procedure with the rule in (9). Bottom panel: A refined procedure which detects in addition the single variable 10; for details see Mandozzi and Bühlmann (2016b). The figure is taken from Mandozzi and Bühlmann (2016b) as well

2008; Mandozzi and Bühlmann 2016a) or using more complete improvements from sequential testing (Mandozzi and Bühlmann 2016b) based on ideas from Goeman and Solari (2010), Goeman and Finos (2012). Our software uses the improved hierarchical adjustment of Mandozzi and Bühlmann (2016a). But the essential gain in computational and statistical power is in terms of the sequential and hierarchical nature of the procedure as illustrated in Figs. 1 and 2. In particular, the method automatically adapts to the resolution level: if the regression parameter of a single variable is very large in absolute value, the procedure might detect such a single variable as being significant; on the other hand, if the signal is not sufficiently strong or if there is substantial correlation (or near linear dependence) within a large number of variables in a group, the method might only identify such a group as being significant. Figure 2 illustrates this point. Naturally, finding a large group to be significant (coarse resolution) is much less informative than detecting a small group or even a single variable.

Methods for p -values The hierarchical procedure with the rules in (9) and (11) requires p values as input which are valid in the sense that they control the type I errors of single tests. We advocate here the use of the multi sample splitting method described in Sect. 2.2.1, implemented in our software. This method is computationally feasible for very high dimension p and it is empirically shown to be competitive, with respect to type I error and power, over a range of scenarios (Dezeure et al. 2015).

The power of the hierarchical method is mainly hinging on the assumption that null-hypotheses further up in the tree are easier to reject, that is the p values are typically getting larger when moving downwards the tree. In low-dimensional regression prob-

lems this is typically true when using partial F-tests for testing $H_{0,G} : \beta_j^0 = 0 \forall j \in G$. Since our p values rely on the partial F test after variable screening with the Lasso, as described in Sect. 2.2.1, the same phenomenon is expected to hold also in the high-dimensional regime.

Clustering and partitioning methods for constructing the hierarchical tree We describe two partitioning methods for constructing a hierarchical tree of the measured SNP variables.

Motivated by the problem of identifiability among correlated variables, we aim to construct a tree such that highly correlated variables are in the same groups: this can be achieved by a standard hierarchical clustering algorithm [Hartigan (1975), cf.], for example using average linkage and the dissimilarity matrix given by $1 - (\text{empirical correlation})^2$. Other clustering algorithms can be used, for example based on canonical correlation (Bühlmann et al. 2013).

Alternatively, we can build a hierarchical tree by using the genomic positions of SNPs. We start with an entire chromosome (or even with the full genome sequence) and use a top-down recursive binary partitioning of the genomic sequence into blocks of consecutive genomic positions, corresponding to a binary tree, such that partitions at every depth of the tree contain about the same number of measured SNPs. Such a spatial recursive partitioning is computationally very fast, and it has the advantage that it can be used for multiple studies with SNPs being measured at different locations for different studies. We note that the approach by Meijer et al. (2015) also involves spatial grouping of SNPs, using a different and computationally more demanding procedure than hierarchical testing described above.

2.4 Causal inference

Causal inference deals with “directional associations”, thereby going beyond regression which is non-directional. A main tool for formalizing this are structural equation models (Pearl 2000, cf.). The analogue to a linear model in (1) is then a structural equation model with a linear structural equation for Y : the data are i.i.d. realizations of

$$\begin{aligned} X^{(j)} &\leftarrow f_j^0(X^{(\text{pa}(j))}, \varepsilon^{(j)}), \quad j = 1, \dots, p, \\ Y &\leftarrow \sum_{k \in \text{pa}(Y)} \theta_k^0 X^{(k)} + \varepsilon^{(Y)}, \\ \varepsilon^{(1)}, \dots, \varepsilon^{(p)}, \varepsilon^{(Y)} &\text{ jointly independent.} \end{aligned} \tag{12}$$

Here $\text{pa}(j) = \text{pa}_D(j)$ denotes the parental set of the node j in a graph D , and the graph D is assumed to be acyclic and encodes the true underlying causal influence diagram (and the random variables $X^{(1)}, \dots, X^{(p)}, Y$ correspond to the nodes in the graph). Furthermore, $f_j^0(\cdot, \cdot)$ are arbitrary measurable potentially nonlinear functions and the “ \leftarrow ” symbol equals an algebraic “ $=$ ” sign but emphasizes that the left-hand side is a direct “causal” function of the right-hand side. We note that the covariates are random: when conditioning on them, assuming that Y is childless (see below in

Proposition 2), we have a fixed design linear model for the data vector $Y = \mathbf{X}\theta + \varepsilon^{(Y)}$ with $\mathbb{E}[\varepsilon^{(Y)}|\mathbf{X}] = 0$.

In absence of knowing the true causal DAG D , the structure D and the corresponding parameter-matrix β are typically non-identifiable from the observational probability distribution. However, there is an interesting exception which is relevant for the case with GWAS, namely when the node Y is childless (i.e. all edges of Y point into Y): this simply means that the response (e.g. disease status) is caused by the genetic SNP biomarkers and there are no causal effects from the response to the genetic variables. The following result holds.

Proposition 2 *Assume a structural equation model with a linear structural equation for Y as in (12) and suppose that Y is childless. Consider the true linear regression coefficients β^0 in the linear regression of Y versus all $X^{(1)}, \dots, X^{(p)}$ and assume that $\text{Cov}((X^{(1)}, \dots, X^{(p)})^\top)$ is positive definite. Then, it holds that $\beta_k^0 = \theta_k^0$ for $k \in \text{pa}(Y)$ and $\beta_k^0 = 0$ for $k \notin \text{pa}(Y)$. Thus, if $\beta_k^0 \neq 0$ it holds that $k \in \text{pa}(Y)$ and there is a directed edge $X^{(k)} \rightarrow Y$ (i.e., a direct causal effect from $X^{(k)}$ to Y).*

Proof The DAG D induces an ordering among the variables such that $\text{pa}_j \subseteq \{j - 1, \dots, 1\}$, assuming for notational simplicity that the variables have already been ordered (according to such an order). Since Y is childless we can choose an ordering where Y is the last element. The conditional distribution then satisfies thanks to the Markov property:

$$\mathcal{L}(Y|X^{(1)}, \dots, X^{(p)}) = \mathcal{L}(Y|X^{(\text{pa}(Y))}).$$

This completes the proof. □

Causal interpretation As a consequence, under the assumptions in Proposition 2, the inference techniques for multiple regression lead to a causal interpretation. The main assumptions for such a substantially more sharpened interpretation are: (i) the underlying true model is a structural equation model with a DAG structure and a linear or generalized linear form for the structural equation of Y (for the latter case, using the analogous argument, we would use a generalized linear model of Y versus all $X^{(1)}, \dots, X^{(p)}$ to obtain the causal variables and effects); (ii) there are no hidden confounding variables between Y and some of the $X^{(j)}$'s; (iii) the response variable Y is childless. The assumption about a positive definite population covariance matrix is weak, even in the context of GWAS; see also the discussion at the end of Sect. 2.1.1. The last assumption (iii) is rather plausible for GWAS since one believes that the genetic factors are the causes for the disease and ruling out that the disease would cause a certain constellation of genetic factors. A notable exception are retroviruses, including e.g. HIV. The second assumption (ii) is rather strong and perhaps the main additional assumption: relaxing it in a very high-dimensional setting is an open problem. In view of measuring thousands of genetic markers, the premise of having measured all the relevant factors is somewhat less unrealistic. The first assumption (i) about the acyclicity of the causal influence diagram is not important as long as there is no feedback from the response Y to the X variables (which is plausible for GWAS), while the requirement for a linear or logistic form might be problematic in view of possible

interactions among the X -variables and/or nonlinear regression functions. The latter is a misspecification and of the same nature as when having misspecified the functional form in a regression model, a topic which we will discuss in Sect. 2.5.

One should always be careful when adopting a causal interpretation. However, and this is a main point, the regression model taking all the variables into account is much more appropriate than a marginal approach where the response Y is marginally regressed or correlated to one SNP variable at a time. This has been the standard approach over many years in GWAS, including extensions with mixed models and adjusting for a few other covariates (Zhou and Stephens 2014). The approach based on inference in a high-dimensional linear or generalized linear model statistics comes much closer to a causal interpretation as described in Proposition 2. And that is among the main reasons why we believe that such multiple regression methods should lead to more reliable results for GWAS in comparison to older marginal techniques.

In case of complex traits, several issues with marginal testing have been pointed out by Frommlet et al. (2012). The work shows that model misspecification can result in a severe loss of power to detect important SNPs and problems occur when ranking the SNPs with respect to their p values. Small correlations between causal and non-causal SNPs may lead to a large number of false positives.

2.5 Misspecification of the model

The results in the previous sections for statistical confidence or testing of linear model parameters rely on the correctness of a linear or generalized linear model as in (1) or (2). If the model is not correct, we have to distinguish more carefully between random and fixed design matrix X (and the latter case may also arise when conditioning on X).

For fixed design and assuming $\text{rank}(X) = n$, we can always represent any $n \times 1$ vector f as $f = X\beta^*$ for some (non-unique) β^* . Therefore, for $f = \{\mathbb{E}[Y_i|X_i]; i = 1, \dots, n\}$ in a regression or $f = \{g(\mathbb{E}[Y_i|X_i]); i = 1, \dots, n\}$ in a generalized regression, we can represent any (nonlinear in x) function f evaluated at the data points as $X\beta^*$. The only question is whether there is a representation with a *sparse* β^* .

For random design, a fit with a linear or generalized linear model to a potentially nonlinear model is to be interpreted as the best approximation with a (generalized) linear model. A linear model approximation has some interesting properties for Gaussian design but the latter is not relevant for GWAS with discrete values for the covariates.

A detailed treatment of model misspecification in the high-dimensional context is given in Bühlmann and van de Geer (2015). A more general perspective in the low-dimensional regime is given in Buja et al. (2014).

3 Software

The R package `hierinf` (available on bioconductor: <https://bioconductor.org/packages/devel/bioc/html/hierinf.html>) is an implementation of the hierarchical inference described in the Sect. 2.3 and it is easy to use for GWAS. The package is a re-implementation of the R package `hierGWAS` (Buzdugan 2019) and includes new features like straightforward parallelization, an additional option for constructing a

hierarchical tree based on spatially contiguous genomic positions, and the possibility of jointly analyzing multiple datasets. To summarize the method, one starts by clustering the data hierarchically. This means that the clusters can be represented by a tree. The main idea is to pursue testing top-down and successively moving downwards until the null-hypotheses cannot be rejected, see Sect. 2.3. The p value of a given cluster is calculated based on the multiple sample splitting approach and aggregation of those p values as described in Sect. 2.2.1. The work flow is straightforward and is composed in two function calls. We note that the package `hierinf` requires complete observations, i.e. no missing values in the data, because the testing procedure is based on all the SNPs which is in contrast to marginal tests. If missing values are present, they can be imputed prior to the analysis. This can be done in R using e.g. `mice` (van Buuren and Groothuis-Oudshoorn 2011), `mi` (Shi et al. 2011), or `missForest` (Stekhoven and Bühlmann 2012).

A small simulated toy example with two chromosomes is used to demonstrate the procedure. The toy example is taken from (Buzdugan 2019) and was generated using PLINK where the SNPs were binned into different allele frequency ranges. The response is binary with 250 controls and 250 cases. Thus, there are $n = 500$ samples, the number of SNPs is $p = 1000$, and there are two additional control variables with column names “age” and “sex”. The first 990 SNPs have no association with the response and the last 10 SNPs were simulated to have a population odds ratio of 2. The functions of the package `hierinf` require the input of the SNP data to be a `matrix` (or a list of matrices for multiple datasets). We use a `matrix` instead of a `data.frame` since this makes computation faster.

```
# load the package
library(hierinf)

# random number generator (for parallel computing)
RNGkind("L'Ecuyer-CMRG")

# We use a small build-in dataset for our toy example.
data(simGWAS)

# The genotype, phenotype and the control variables are
# saved in different objects.
sim.geno <- simGWAS$x
sim.pheno <- simGWAS$y
sim.clvar <- simGWAS$clvar
```

The two following sections correspond to the two function calls in order to perform hierarchical testing. The third section states some remarks about running the code in parallel.

3.1 Software for clustering

The package `hierinf` offers two possibilities to build a hierarchical tree for corresponding hierarchical testing. The function `cluster_var` performs hierarchical clustering based on some dissimilarity matrix and is described first. The function

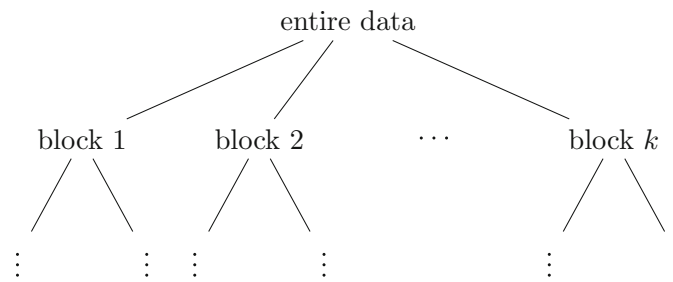


Fig. 3 The top two levels of a hierarchical tree used to perform multiple testing. The user can optionally specify the second level of the tree with the advantage that one can easily run the code in parallel over different clusters in the second level, denoted by block 1, ..., block k . A natural choice is to choose the chromosomes as the second level of the hierarchical tree, which define a partition of the SNPs. If the second level is not specified, then the first split is estimated based on clustering the data, i.e. it is a binary split. The user can define the second level of the tree structure using the argument `block` in the functions `cluster_var/cluster_position`. The function `cluster_var/cluster_position` builds a separate binary hierarchical tree for each of the blocks

`cluster_position` builds a tree based on recursive binary partitioning of consecutive positions of the SNPs. For a short description, see at the end of Sect. 2.3.

Hierarchical clustering is computationally expensive and prohibitive for large datasets. Thus, it makes sense to pre-define dis-joint sets of SNPs which can be clustered separately. One would typically assume that the second level of a cluster tree structure corresponds to the blocks given by the chromosomes as illustrated in Fig. 3. For the method based on binary partitioning of consecutive positions of SNPs, we recommend to pre-define the second level of the hierarchical tree as well. This allows to run the building of the hierarchical tree and the hierarchical testing for each block or in our case for each chromosome in parallel, which can be achieved using the function calls below. If one does not want to specify the second level of the tree, then the argument `block` in both function calls can be omitted.

In the toy example, we define the second level of the tree structure as follows. The first and second 500 SNPs of the SNP data `sim.geno` correspond to chromosome 1 and chromosome 2, respectively. The object `block` is a `data.frame` which contains two columns identifying the two blocks. The blocks are defined in the second column and the corresponding column names of the SNPs are stored in the first column. The argument `stringsAsFactors` of the function `data.frame` is set to `FALSE` because we want both columns to contain integers or strings.

```

# Define the second level of the tree structure.
block <- data.frame("colname" = paste0("SNP.", 1:1000),
                   "block" = rep(c("chrom 1",
                                   "chrom 2"),
                                each = 500),
                   stringsAsFactors = FALSE)

# Cluster the SNPs
dendr <- cluster_var(x = sim.geno,
                    block = block,
                    # the following arguments have to
                    # be specified for parallel
                    # computation
  
```

```
parallel = "multicore",
ncpus = 2)
```

By default, the function `cluster_var` uses the agglomeration method average linkage and the dissimilarity matrix given by $1 - (\text{empirical correlation})^2$.

Alternatively, `cluster_position` builds a hierarchical tree using recursive binary partitioning of consecutive genomic positions of the SNPs. As for `cluster_var`, the function can be run in parallel if the argument `block` defines the second level of the hierarchical tree.

```
# Store the positions of the SNPs.
position <- data.frame("colnames" = paste0("SNP.",
                                           1:1000),
                      "position" = seq(from = 1,
                                       to = 1000),
                      stringsAsFactors = FALSE)

# Build the hierarchical tree based on the position.
# The argument block defines the second level of the
# tree structure.
dendr.pos <- cluster_position(position = position,
                             block = block,
                             # the following arguments
                             # have to be specified
                             # for parallel
                             # computation
                             parallel = "multicore",
                             ncpus = 2)
```

3.2 Software for hierarchical testing

The function `test_hierarchy` is executed after the function `cluster_var` or `cluster_position` since it requires the output of one of those two functions as an input (argument `dendr`).

The function `test_hierarchy` first randomly splits the data into two halves (with respect to the observations), by default $B = 50$ times, and performs variable screening on the second half. Then, the function `test_hierarchy` uses those splits and corresponding selected variables to perform the hierarchical testing according to the tree defined by the output of one of the two functions `cluster_var` or `cluster_position`.

As mentioned in Sect. 3.1, we can exploit the proposed hierarchical structure which assumes the chromosomes to form the second level of the tree structure as illustrated in Fig. 3. This allows to run the testing in parallel for each block, which are the chromosomes in the toy example.

The following function call performs first the global null-hypothesis test for the group containing all the variables/SNPs and continues testing in the hierarchy of the two chromosomes and their children.

```
# Test the hierarchy using multi sample split
set.seed(1234)
result <- test_hierarchy(x = sim.geno,
                        y = sim.pheno,
                        clvar = sim.clvar,
                        # alternatively:
                        # dendr = dendr.pos
                        dendr = dendr,
                        family = "binomial",
                        # the following arguments have
                        # to be specified for parallel
                        # computation
                        parallel = "multicore",
                        ncpus = 2)
```

The function `test_hierarchy` allows to fit models with continuous or binary response, the latter being based on logistic regression. The argument `family` is set to "binomial" because the response variable in the toy example is binary.

The output looks as follows:

```
> print(result, n.terms = 4)
  block    p.value  significant.cluster
1 chrom 1 NA          NA
2 chrom 2 0.0489170 SNP.605, SNP.792, SNP.636, SNP.857,
  ... [8]
3 chrom 2 0.0020718 SNP.992
4 chrom 2 0.0001637 SNP.991
5 chrom 2 0.0047820 SNP.1000
6 chrom 2 0.0065060 SNP.994
7 chrom 2 3.858e-05 SNP.993
```

The output shows significant groups of SNPs or even single SNPs if there is sufficiently strong signal in the data. The block names, the p values, and the column names (of the SNP data) of the significant clusters are returned. There is no significant cluster in chromosome 1. That's the reason why the p value and the column names of the significant cluster are NA in the first row of the output. Note that the large significant cluster in the second row of the output is shortened to better fit on screen. In our toy example, the last 8 column names are replaced by "... [8]". The maximum number of terms can be changed by the argument `n.terms` of the `print` function. One can evaluate the object `result` in the console and the default values of the `print` function are used. In this case, it would only display the first 5 terms.

The only difference in the R code when using a hierarchical tree based on binary recursive partitioning of the genomic positions of the SNPs (whose output is denoted as `dendr.pos`) is to specify the corresponding hierarchy: `test_hierarchy(..., dendr = dendr.pos, ...)`.

We can access part of the output by `result$res.hierarchy` which we use below to calculate the R^2 value of the second row of the output, i.e.

`result$res.hierarchy[[2, "significant.cluster"]]`. Note that we need the double square brackets to access the column names stored in the column `significant.cluster` of the output since the last column is a list where each element contains a character vector of the column names. The two other columns containing the block names and the p values can both be indexed using single square brackets as for any `data.frame`, e.g. `result$res.hierarchy[2, "p.value"]`.

```
> (coln.cluster <- result$res.hierarchy[[2,
                                     "significant.cluster"]])
[1] "SNP.605" "SNP.792" "SNP.636" "SNP.857" "SNP.858"
[6] "SNP.911" "SNP.571" "SNP.998" "SNP.708" "SNP.867"
[11] "SNP.612" "SNP.932"
```

The function `compute_r2` calculates the adjusted R^2 value or coefficient of determination of a cluster for a continuous response. The Nagelkerke's R^2 (Nagelkerke et al. 1991) is calculated for a binary response as e.g. in our toy example.

```
> compute_r2(x = sim.geno, y = sim.pheno,
             clvar = sim.clvar,
             res.test.hierarchy = result,
             family = "binomial",
             colnames.cluster = coln.cluster)
```

The function `compute_r2` is based on multi-sample splitting. The R^2 value is calculated per split based on the second half of observations and based on the intersection of the selected variables and the user-specified cluster. Then, the R^2 values are averaged over the different splits. If one does not specify the argument `colnames.cluster`, then the R^2 value of the whole dataset is calculated.

3.3 Software for parallel computing

The function calls of `cluster_var`, `cluster_position`, and `test_hierarchy` above are evaluated in parallel since we set the arguments `parallel = "multicore"` and `ncpus = 2`. The argument `parallel` can be set to "no" for serial evaluation (default value), to "multicore" for parallel evaluation using forking, or to "snow" for parallel evaluation using a parallel socket cluster (PSOCKET); see below for more details. The argument `ncpus` corresponds to the number of cores to be used for parallel computing. We use the `parallel` package for our implementation which is already included in the base R installation (R Core Team 2019).

The user has to select the "L'Ecuyer-CMRG" pseudo-random number generator and set a seed such that the parallel computing of `hierinf` is reproducible. This pseudo-random number generator can be selected by `RNGkind("L'Ecuyer-CMRG")` and has to be executed once for every new R session; see R code at the beginning of Sect. 3. This allows us to create multiple streams of pseudo-random numbers, one for each processor/computing node, using the `parallel` package; for more details see the vignette of the `parallel` package published by R Core Team (2019).

We recommend to set the argument `parallel = "multicore"` which will work on Unix/Mac (but not Windows) operation systems. The function is then evaluated in parallel using forking which is leaner on the memory usage. This is a neat

feature for GWAS since e.g. a large SNP dataset does not have to be copied to the new environment of each of the processors. Note that this is only possible on a multicore machine and not on a cluster.

On all operation systems, it is possible to create a parallel socket cluster (PSOCKET) which corresponds to setting the argument `parallel = "snow"`. This means that the computing nodes or processors do not share the memory, i.e. an R session with an empty environment is initialized for each of the computing nodes or processors.

How many processors should one use? If the user specifies the second level of the tree, i.e. defines the `block` argument of the functions `cluster_var / cluster_position` and `test_hierarchy`, then the building of the hierarchical tree and the hierarchical testing can be easily performed in parallel across the different blocks. Note that the package can make use of as many processors as there are blocks, say, 22 chromosomes. In addition, the multi sample splitting and screening step, which is performed inside the function `test_hierarchy`, can always be executed in parallel regardless if we defined blocks or not. It can make use of at most B processors where B is the number of sample splits.

3.4 Illustration: hierarchical inference on real datasets

Hierarchical inference for GWAS has been successfully applied in some of our own previous work (Buzdugan et al. 2016; Klasen et al. 2016).

One dataset is about type 1 diabetes with a binary response variable (“healthy”/“diseased”): The Wellcome Trust Case Control Consortium (2007) measured 500’568 SNPs of 2’000 cases and 3’000 controls. Some of the results from Buzdugan et al. (2016) are described in Table 1. Buzdugan et al. (2016) found a significant association of the response and eight single SNPs: five of those SNPs have been found to be significant in the study of The Wellcome Trust Case Control Consortium (2007). One of the other three SNPs was found to have a moderate association in an independent study (Plagnol et al. 2011).

Buzdugan et al. (2016) identified two small significant groups of SNPs for the type 2 diabetes dataset which has the same sample size and number of SNPs as the type 1 diabetes dataset. Their results are described in Table 2. Both groups contain one SNP which was originally found significant by The Wellcome Trust Case Control Consortium (2007). There are two SNPs, one in each of the two groups, that were shown significant in an independent study by Zeggini et al. (2007) and only one of those two SNPs by Scott et al. (2007).

Klasen et al. (2016) compare hierarchical testing with linear mixed effect models and stress that the hierarchical testing seems less exposed to population structure and often does not need a corresponding correction. One of the studied datasets is about the association between the root development and the genotype of 201 world-wide collected natural *Arabidopsis* accessions. They found one significant locus with a linear mixed effect model whereas with the hierarchical testing they discovered three additional loci which are located in two neighboring genes. Klasen et al. (2016) made a follow-up randomized treatment-control experiment to validate an effect of one of

Table 1 List of small significant groups of SNPs for type 1 diabetes

Significant group of SNPs	Chr	Gene	p value	R^2
rs6679677	1	PHTF1	3.6×10^{-11}	0.03
rs17388568	4	ADAD1	2.7×10^{-2}	0.006
rs9272346	6	HLA-DQA1	2.4×10^{-3}	0.17
rs9272723	6	HLA-DQA1	2.2×10^{-4}	0.17
rs2523691	6	Intergenic	6.04×10^{-5}	0.004
rs11171739	12	Intergenic	1.3×10^{-2}	0.01
rs17696736	12	NAA25	6.5×10^{-4}	0.018
rs12924729	16	CLEC16A	3.4×10^{-2}	0.007

The smallest groups of SNPs whose null hypothesis was rejected are displayed. The SNPs in this group are jointly significant. The rsIDs are taken from dbSNP. Chromosome is abbreviated by Chr. If the group of SNPs belongs to a gene, then the gene symbol from Entrez Gene is stated in the corresponding column. The p values are adjusted for multiple testing (controlling the FWER) and the R^2 value is the explained variance by the group of SNPs. The table is taken from Buzdugan et al. (2016)

Table 2 List of small significant groups of SNPs for type 2 diabetes

Significant group of SNPs	Chr	Gene	p value	R^2
rs4074720, rs10787472, rs7077039, rs11196208, rs11196205, rs10885409, rs12243326, rs4132670, rs7901695, rs4506565	10	TCF7L2	1.7×10^{-5}	0.015
rs9926289, rs7193144, rs8050136, rs9939609	16	FTO	4.7×10^{-2}	0.007

The smallest groups of SNPs whose null hypothesis was rejected are displayed. The SNPs in this group are jointly significant. The rsIDs are taken from dbSNP. Chromosome is abbreviated by Chr. If the group of SNPs belongs to a gene, then the gene symbol from Entrez Gene is stated in the corresponding column. The p values are adjusted for multiple testing (controlling the FWER) and the R^2 value is the explained variance by the group of SNPs. The table is taken from Buzdugan et al. (2016)

these two genes on the root growth (namely the PEPR2 gene): it turned out to be successful exhibiting a significant effect.

4 Meta-analysis for several datasets

Consider the general situation with m datasets

$$Y^{(\ell)}, \mathbf{X}^{(\ell)}, \ell = 1, \dots, m,$$

with $n_\ell \times 1$ response vector $Y^{(\ell)}$ and $n_\ell \times p_\ell$ design matrix $\mathbf{X}^{(\ell)}$. For each of them we assume a potentially high-dimensional linear model

$$Y^{(\ell)} = \mathbf{X}^{(\ell)}\beta^{(\ell)} + \varepsilon^{(\ell)},$$

with $\varepsilon_1^{(\ell)}, \dots, \varepsilon_{n_\ell}^{(\ell)}$ i.i.d. having $\mathbb{E}[\varepsilon_i^{(\ell)}] = 0$, $\text{Var}(\varepsilon_i^{(\ell)}) = (\sigma^{(\ell)})^2$. To simplify notation, we drop here the superscript “0” for denoting the true underlying parameter. Note that the treatment for generalized linear models is analogous.

For simplicity, we consider here only the case where the measured covariates are the same across all the m datasets. This implies that $p_\ell \equiv p$ for all $\ell = 1, \dots, m$. We consider the null-hypothesis for single variables

$$\tilde{H}_{0,j} : \beta_j^{(\ell)} = 0 \quad \text{for all } \ell = 1, \dots, m, \tag{13}$$

versus the alternative

$$\tilde{H}_{A,j} : \text{there exists } \ell \in \{1, \dots, m\} \text{ with } \beta_j^{(\ell)} \neq 0. \tag{14}$$

For groups of variables $G \subseteq \{1, \dots, p\}$ we have the analogous hypotheses:

$$\tilde{H}_{0,G} : \beta_G^{(\ell)} \equiv 0 \quad \text{for all } \ell = 1, \dots, m, \tag{15}$$

versus the alternative

$$\tilde{H}_{A,G} : \text{there exists } j \in G \text{ and } \ell \in \{1, \dots, m\} \text{ with } \beta_j^{(\ell)} \neq 0. \tag{16}$$

If $\tilde{H}_{0,j}$ is rejected we conclude that covariate j is significant in at least one dataset. From an abstract point of view, $\tilde{H}_{0,j}$ or $\tilde{H}_{0,G}$ as in (13) or (15) are again group hypothesis with coefficient indices in the group $(\ell, j) \in \{1, \dots, m\} \times \{j\}$ or $(\ell, j) \in \{1, \dots, m\} \times G$, respectively.

A simple way to test the hypotheses in (13) or (15) is to aggregate the corresponding p values for the datasets $\ell = 1, \dots, m$. Denote by $P_G^{(\ell)}$ the p value for testing the null-hypothesis $H_{0,G}^{(\ell)} : \beta_G^{(\ell)} \equiv 0$ for the dataset ℓ .

We advocate here the use of Tippett’s rule (Tippett 1931):

$$P_{\text{Tippett};G} = 1 - \left(1 - \min\{P_G^{(\ell)}, \ell = 1, \dots, m\}\right)^m, \tag{17}$$

where $P_G^{(1)}, \dots, P_G^{(m)}$ are the raw p values. This aggregated p value controls the familywise error rate at level α for the decision rule: reject $\tilde{H}_{0,G}$ if and only if $P_{\text{Tippett};G} \leq \alpha$ for some significance level α .

Alternatively, p values can be aggregated by Stouffer’s rule (Stouffer et al. 1949):

$$P_{\text{Stouffer};G} = \Phi \left(\sum_{\ell=1}^m w_\ell \Phi^{-1} \left(P_G^{(\ell)} \right) \right), \quad w_\ell = \sqrt{n_\ell/n}, \quad n = \sum_{\ell=1}^m n_\ell. \tag{18}$$

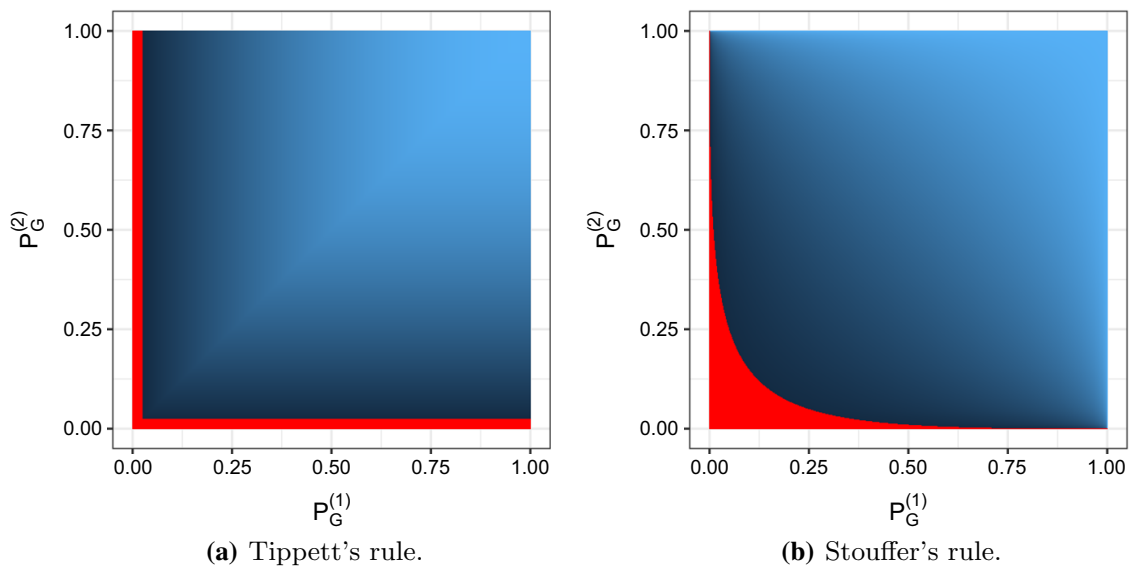


Fig. 4 Aggregated p value based on two datasets. The red areas (lower left corner) highlight the aggregated p values which are below 0.05 (color figure online)

This p value controls the familywise error rate at level α for the decision rule: reject $\tilde{H}_{0,G}$ if and only if $P_{\text{Stouffer};G} \leq \alpha$.

For illustration purposes, we consider the case $m = 2$ in Fig. 4. The individual p values $P_G^{(1)}$ and $P_G^{(2)}$ are plotted on the x - and y -axis, respectively and the aggregated values $P_{\text{Tippett};G}$ and $P_{\text{Stouffer};G}$ are color-coded in the respective plots. Both red areas are equal to 0.05. The difference between the two plots is that Stouffer's rule is more powerful in the case of two datasets with weak signal and Tippett's rule is more powerful in the case of one dataset with a strong signal and the other having a very weak or no signal.

We advocate the use of Tippett's rule because it performs best in our simulations for all scenarios; see Fig. 5 and Sect. 4.1 for more details. This seems partially due to the hierarchical multiple sample splitting inference method which is unstable, especially for weaker signals: it happens fairly often that a cluster turns out to be clearly significant in one dataset and not significant at all in another, a situation where Tippett's rule is much more powerful. See also the paragraph at the end of Sect. 4.2

The naive (and conceptually wrong) approach would be to pool the different datasets and proceed as if it would be one homogeneous dataset. This would then result in p values $P_{\text{pooled};G}$ by using the methods from Sect. 2.

Fast computational methods for pooled GWAS There has been a considerable interest for fast algorithms for GWAS with very large sample size in the order of 10^5 ; see Lippert et al. (2011), Zhou and Stephens (2014). Often though, such large sample size comes from pooling different studies or sub-populations. We argue in favor of meta analysis and aggregating corresponding p values. Besides more statistical robustness against heterogeneity (arising from the different sub-populations), meta-analysis is also computationally very attractive: the computations can be trivially implemented in parallel for every sub-population and the p value aggregation step comes essentially without any computational cost.

4.1 Empirical results for aggregating p values and pooling of two datasets

We perform a simulation study to compare power and error rate for three methods. We consider aggregating the p values using Tippett's rule as described in (17), Stouffer's method in (18), and pooling the datasets. The latter is a very simple method where we ignore that we deal with different datasets or studies and run the hierarchical testing on the pooled set of observations but allowing for a different intercept per dataset.

For simplicity, we consider the case of two datasets, i.e. $m = 2$. Denote the true underlying parameter by $\beta^{(\ell)}$ for $\ell = 1, 2$ and the corresponding active set by

$$S_0^{(\ell)} = \{j; \beta_j^{(\ell)} \neq 0\}, \ell = 1, 2.$$

As an easy case we assume here that the active sets of the two datasets coincide $S_0^{(1)} = S_0^{(2)}$ and that the true underlying parameters $\beta^{(1)}$ and $\beta^{(2)}$ take the values 1 and -1 on the active set, respectively. If one pools the two datasets, then those effects roughly cancel each other (when the datasets have approximately the same sample sizes). On the other hand, when aggregating p values from individual datasets, effects do not cancel out.

To compare the two methods, we generate semi-synthetic data which is based on data from openSNP (<https://opensnp.org/>), where people donate their raw genotypic data into the public domain (using CC0 license). We generate two datasets $\mathbf{X}^{(\ell)}$, $\ell = 1, 2$, with $n = 300$ observations each and two (consecutive) blocks of 500 SNPs from chromosome 1 and 2, respectively. This makes in total $p = 1000$ SNPs. Both datasets share the same 1000 SNPs and are kept fixed for the simulation.

For the generation of those two datasets, columns with many missing values are excluded and remaining columns are imputed using the median. We further exclude columns with standard deviation zero and omit columns in order not to have a set of collinear columns of set size up to 10.

For each simulation run, we randomly pick an active set of size 10 which is the same for both datasets. Thus, $S_0 = S_0^{(1)} = S_0^{(2)}$. We simulate a continuous response using

$$Y^{(\ell)} = \mathbf{X}^{(\ell)}\beta^{(\ell)} + \varepsilon^{(\ell)}, \ell = 1, 2,$$

where each element of $\varepsilon^{(\ell)}$ is drawn from a $\mathcal{N}(0, (\sigma^{(\ell)})^2)$ -distribution. For the simulation, we vary the values of $(\sigma^{(2)})^2$ and the values of $\beta^{(\ell)}$ for the corresponding elements which are in the active set $S_0^{(\ell)}$, $\ell = 1, 2$.

The two datasets play different roles. The variance $(\sigma^{(1)})^2 = 1$ is fixed for the dataset $Y^{(1)}, \mathbf{X}^{(1)}$ and only the value of the non-zero elements of $\beta^{(1)}$ are varied. The dataset $Y^{(1)}, \mathbf{X}^{(1)}$ carries a strong signal in general. The dataset $Y^{(2)}, \mathbf{X}^{(2)}$ shows a weak signal especially when we inflate the variance $(\sigma^{(2)})^2$. The elements of $\beta^{(2)}$ corresponding to the active set take only values 0, 0.5 and 1.

We use a modified definition of the power as the performance measure for the simulation study because it takes the size of the significant clusters into account. We define the adaptive power by

$$\text{Power}_{\text{adap}} = \frac{1}{|S_0|} \sum_{C \in \text{MTD}} \frac{1}{|C|}$$

where MTD stands for Minimal True Detections which means that the cluster has to be significant (“Detection”), there is no significant subcluster (“Minimal”), and the cluster contains at least one active variable (“True”). This is the same definition as in Mandozzi and Bühlmann (2016a).

Figure 5 illustrates the adaptive power of the simulation study. Aggregating the p values using Tippett’s rule is clearly better than pooling and outperforms Stouffer’s method. The two different aggregation methods and their advantages have been already discussed at the beginning of Sect. 4. Pooling the datasets seems to work fine especially for situations where the values of the non-zero elements of $\beta^{(1)}$ and $\beta^{(2)}$ are similar and the standard deviation $\sigma^{(2)}$ takes values 0.5 or 1, i.e. similar standard deviations for both datasets. But in these situations, aggregating the p values using Tippett’s rule works comparably well. We note that with pooling, the power can slightly decrease when the true regression parameters in one dataset increase in size: this is somewhat counter-intuitive but might occur because misspecification with pooling can become stronger when increasing the regression parameters in one dataset. In general aggregation with Tippett’s rule performs more reliably than pooling since the latter is conceptually wrong. Figure 6 illustrates that the familywise error rate (FWER) is controlled for all three methods, for most scenarios even conservatively.

The conceptual correctness together with the results of the simulation study support our recommendation to aggregate the p values from different datasets or studies rather than a simple-minded pooling of the datasets. Aggregating the p values of multiple studies is very easy to perform using the R package `hierinf` as described in Sect. 4.4.

4.2 Empirical results for aggregating p values and pooling of multiple datasets

We consider two simulations for the case of $m = 10$ semi-synthetic datasets $Y^{(\ell)}$, $\mathbf{X}^{(\ell)}$, $\ell = 1, \dots, 10$, with $n = 150$ observations and $p = 10,000$ SNPs as described in Sect. 4.1. The response is simulated as

$$Y^{(\ell)} = \mathbf{X}^{(\ell)} \beta^{(\ell)} + \varepsilon^{(\ell)}, \ell = 1, \dots, 10,$$

where each element of $\varepsilon^{(\ell)}$ is drawn from a $\mathcal{N}(0, 1)$ -distribution, i.e. all the variances are kept fixed.

We examine two scenarios where the support of the parameter vectors $\beta^{(\ell)}$ is the same across all datasets. In particular, the non-zero elements of $\beta^{(\ell)}$, $\ell = 1, \dots, 5$, respectively, $\ell = 1, \dots, 8$, are varied by one number while the non-zero elements of $\beta^{(k)}$ of the remaining datasets are equal to 0.5.

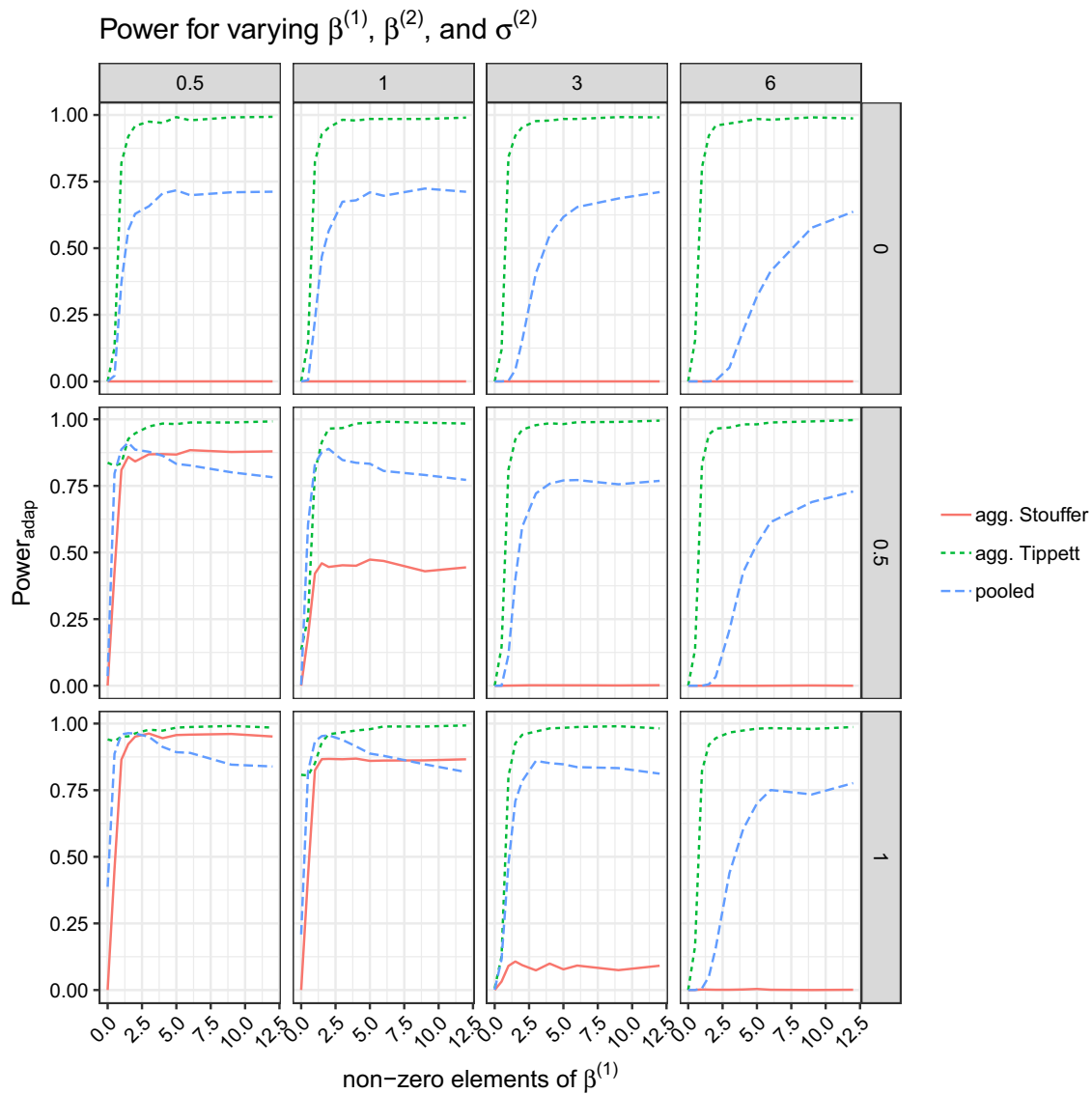


Fig. 5 Two datasets: comparison of the adaptive power of aggregating the p values using Tippett’s rule, Stouffer’s rule or by simply pooling multiple studies. The values of the active or non-zero element of both datasets are varied, i.e. $\beta^{(1)} \in \{0, 0.5, 1, 1.5, 2, 3, 4, 5, 6, 9, 12\}$ (x -axis) and $\beta^{(2)} \in \{0, 0.5, 1\}$ (multi panels: rows). The standard deviation of the error is varied for the second dataset, i.e. $\sigma^{(1)} = 1$ and $\sigma^{(2)} \in \{0.5, 1, 3, 6\}$ (multi panels: columns). The active set is of size 10 and is randomly selected for each simulation run. The adaptive power was calculated based on 100 independent simulations for each combination of the parameters

Aggregating the p values using Tippett’s rule performs worse than pooling while aggregation with Stouffer’s rule performs poorly. The results are illustrated in Figs. 7 and 8. The number of observations per dataset is halved compared to the simulation in Sect. 4.1 and the number of SNPs is 10 times larger, both being favourable for pooling. We also note that the active sets of the 10 datasets are identical and thus, the different datasets are perhaps still rather “homogeneous”. It can be dangerous to pool the datasets because in general there is no theoretical guarantee that the FWER is controlled.

Performance of Stouffer’s rule. The main reason why Stouffer’s rule for aggregation of p values performs so poorly seems to be the *instability* of the hierarchical inference

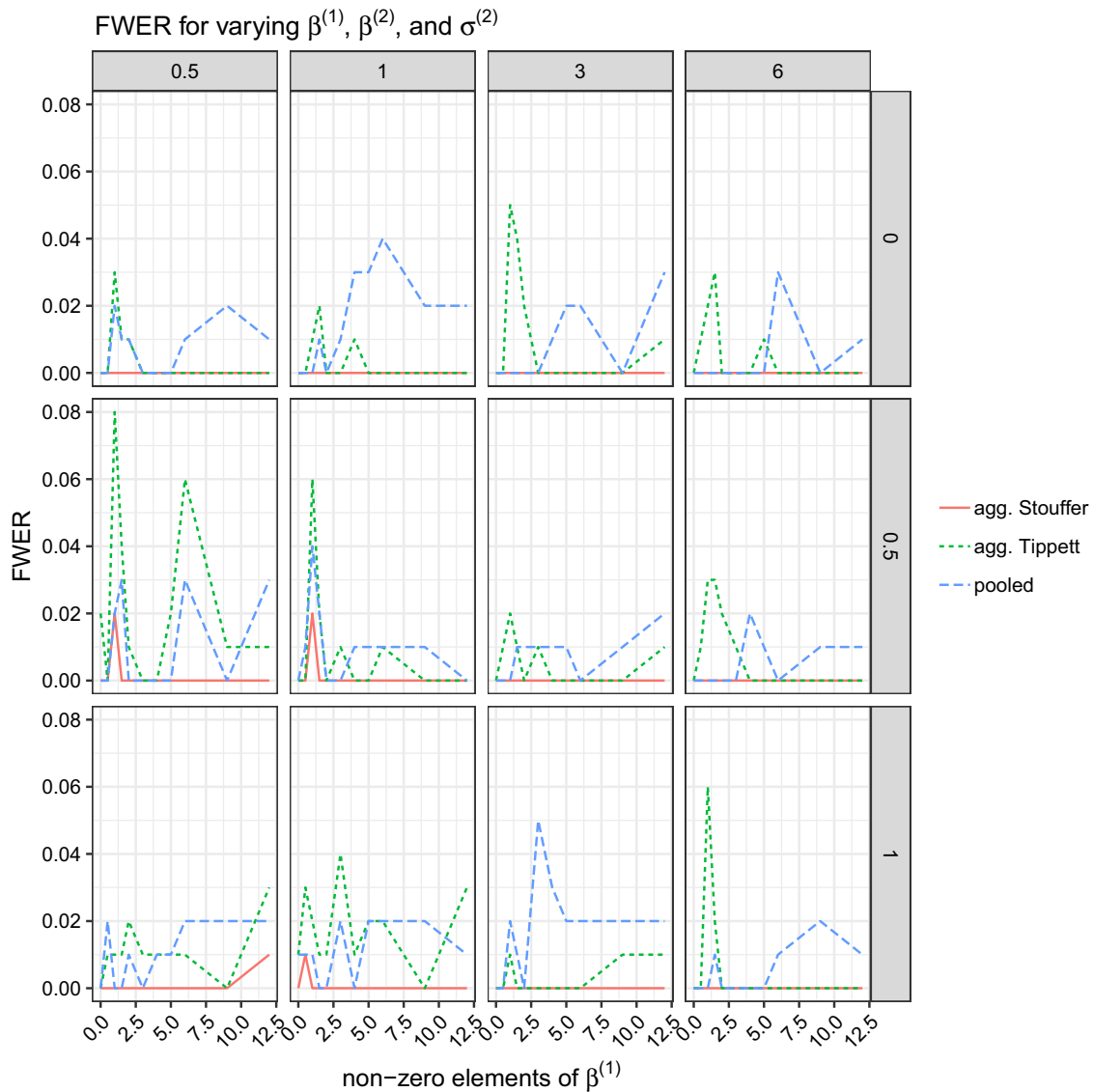


Fig. 6 Two datasets: comparison of the familywise error rate (FWER) of aggregating the p values using Tippett’s rule, Stouffer’s rule or by simply pooling multiple studies. All three methods control the FWER at level 0.05

scheme. For two datasets having even the same generating distribution, it can easily happen that the hierarchical inference scheme provides once a highly significant and once a non-significant result. And analogously, a similar pattern arises with more than two datasets. In such situations, Stouffer’s rule performs poorly, as indicated also by Fig. 4. In the worst case, if one of the p values from the different datasets is 1, then Stouffer’s rule won’t reject for sure.

The explanation of the observed instability is as follows. The p values arising from multiple sample splits are aggregated using (8) where the correction factor $1/\gamma$ is the price to pay for using multi sample splitting. An aggregated p value can be large or even 1 if a mix of moderate to large (and perhaps also some very few small) p values is aggregated. Furthermore, the raw p values of an active cluster can be large or even equal to 1 if the signal is weak or if the selected variables from Lasso pre-screening have an empty intersection with the cluster of interest, respectively. The latter issue

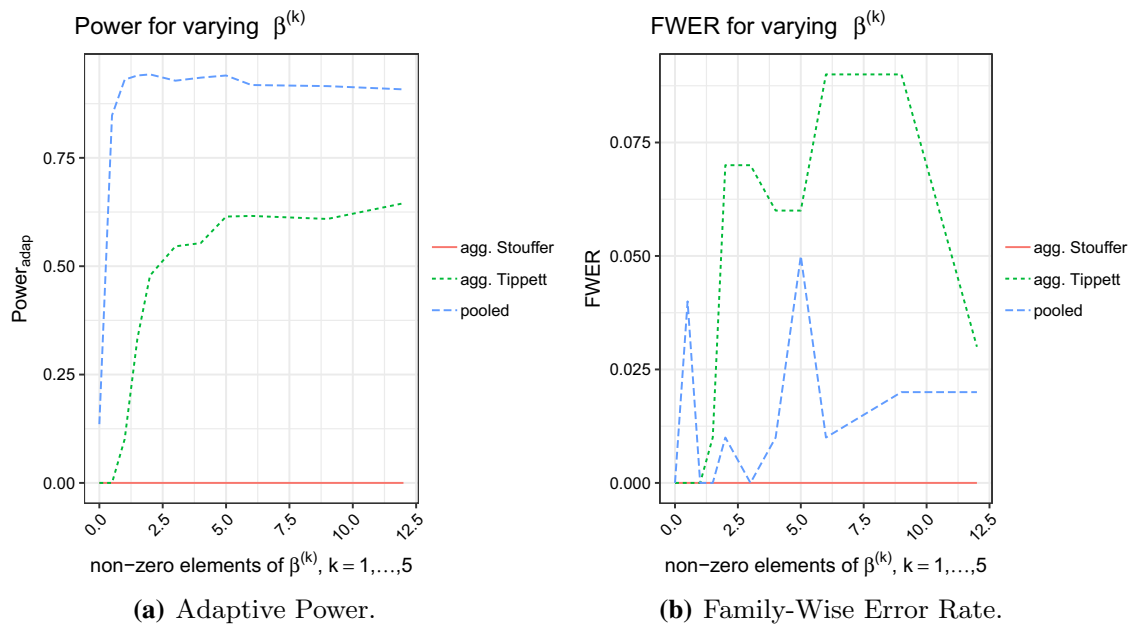


Fig. 7 Ten datasets: comparison of the adaptive power and FWER of aggregating p values using Tippett's or Stouffer's rule or by simply pooling multiple studies. The single value of the non-zero elements of $\beta^{(\ell)}$, $\ell = 1, \dots, 5$ is varied, while the non-zero elements of $\beta^{(\ell)}$, $\ell = 6, \dots, 10$ all take the value 0.5. The common active set is of size 10 and is randomly selected for each simulation run. The results are based on 100 simulation runs

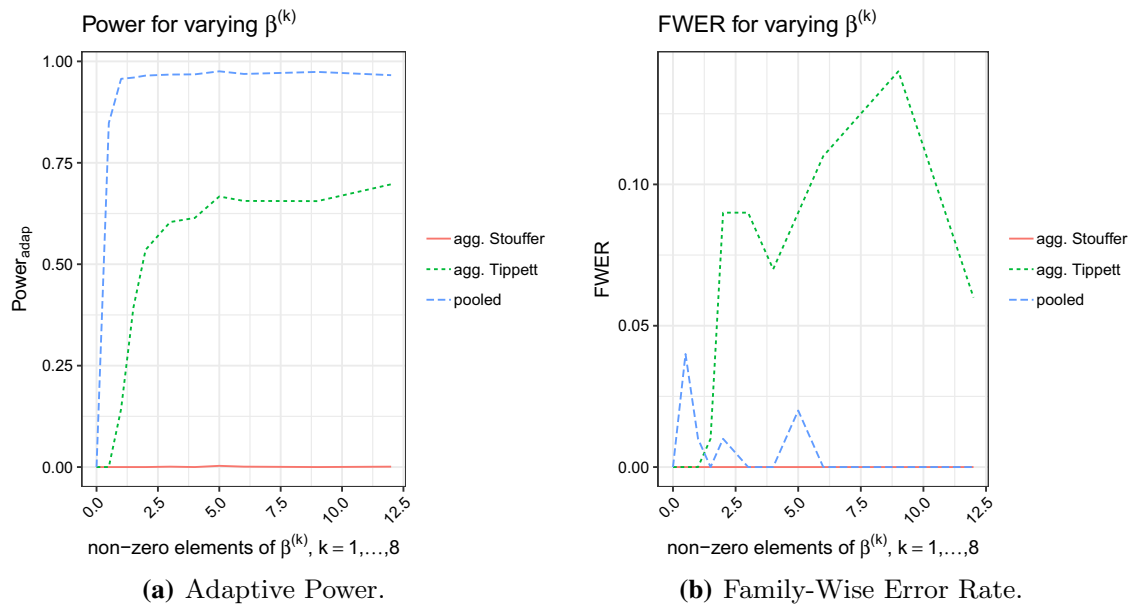


Fig. 8 Ten datasets: comparison of the adaptive power and FWER of aggregating p values using Tippett's or Stouffer's rule or by simply pooling multiple studies. The single value of the non-zero elements of $\beta^{(\ell)}$, $\ell = 1, \dots, 8$ is varied, while the non-zero elements of $\beta^{(\ell)}$, $\ell = 9, 10$ both take the value 0.5. The common active set is of size 10 and is randomly selected for each simulation run. The results are based on 100 simulation runs

arises because of the difficulty of variable screening in very high-dimensional settings with high correlations among the variables.

4.3 Theoretical considerations for aggregating p values and pooling of multiple datasets

We have illustrated in Figs. 7 and 8 that pooling can be clearly better than using Tippett’s multiple testing correction.

To shed some light on the issue, we consider the situation with linear models as mentioned at the beginning of Sect. 4,

$$Y_i^{(\ell)} = \sum_{j=1}^p \beta_j^{(\ell)} X_i^{(\ell)} + \varepsilon_i^{(\ell)}, \quad i = 1, \dots, n_\ell,$$

over various datasets $\ell = 1 \dots, m$. For simplicity, we assume that $n_\ell \equiv n$ for all ℓ and that the $X_i^{(\ell)}$ are fixed variables which have been i.i.d. sampled from a distribution with covariance matrix $\Sigma_X^{(\ell)}$, where $\Sigma_X^{(\ell)} \equiv \Sigma_X$ for all ℓ . The latter might be far from being true but our aim here is only to present a simple argument.

Consider a statistic for testing $\beta_j^{(\ell)}$:

$$T_j^{(\ell)}, \quad T_j^{(\ell)} \sim \mathcal{N}(0, 1) \text{ under } \tilde{H}_{0,j},$$

with $\tilde{H}_{0,j}$ as in (13). The t -test statistic in a linear model satisfies this asymptotically under mild distributional assumptions on the error term, and under the assumptions from Sect. 2.2.1, this also holds in a sample splitting context as used in our approach and software.

Tippett’s multiple testing correction (17) is slightly more powerful than Bonferroni correction, and the latter amounts to consider the maximum of the test-statistics

$$\max_{\ell=1, \dots, m} |T_j^{(\ell)}|.$$

It is well known, that due to the Gaussian assumption and under the null-hypothesis $\tilde{H}_{0,j}$:

$$\mathbb{P}\left[\max_{\ell=1, \dots, m} |T_j^{(\ell)}| > \sqrt{c^2 + 2 \log(m)} \right] \leq 2 \exp(-c^2/2).$$

This implies that for $T_j^{(\ell)}$ being the t -test statistics for $\beta_j^{(\ell)}$, the test has power converging to 1 (for any fixed significance level) if

$$\max_{\ell=1, \dots, m} \frac{|\beta_j^{(\ell)}|}{\sigma^{(\ell)} (\Sigma_X)_{jj}^{-1}} \gg \sqrt{\log(m)/n}, \tag{19}$$

where $\sigma^{(\ell)} = \sqrt{\text{Var}(\varepsilon^{(\ell)})}$ is the standard deviation of the noise term $\varepsilon^{(\ell)}$. Thus, we see from (19) that Tippett’s correction pays a price with a factor $\sqrt{\log(m)}$, due to multiple testing, instead of the usual detection rate $1/\sqrt{n}$.

With pooling as described at the beginning of Sect. 4.1, we consider the pooled parameter in the linear model over all the m datasets:

$$\beta^{\text{pool}} = \operatorname{argmin}_{\beta} \mathbb{E} \left[n_{\text{tot}}^{-1} \sum_{i=1}^{n_{\text{tot}}} (Y_i - X_i^T \beta)^2 \right],$$

with corresponding noise term $\varepsilon_i^{\text{pool}} = Y_i - X_i^T \beta^{\text{pool}}$ and $n_{\text{tot}} = \sum_{\ell=1}^m n_{\ell} = mn$. We then obtain that

$$\beta^{\text{pool}} = \sum_{\ell=1}^m \beta^{(\ell)} \mathbb{P}[Z = \ell],$$

where Z denotes the random variable encoding the index of the dataset (assuming here a mixture model for the m datasets). In comparison to (19), the t -test with pooled data then leads to the detection

$$\frac{|\beta_j^{\text{pool}}|}{\sigma^{\text{pool}} (\Sigma_X)_{jj}^{-1}} \gg \sqrt{1/(mn)}. \tag{20}$$

For comparing (19) with (20), we consider two special cases.

Case I (equal β_j 's) Suppose that $\beta_j^{(\ell)} \equiv \beta_j$ for all ℓ , implying also that the supports of $\beta^{(\ell)}$ are the same. Then it holds that $\beta_j^{\text{pool}} = \beta_j$ and the detection boundary in (20) is clearly in favor of the pooled method. This case is “fairly close” to the scenario in Figs. 7 and 8 where all the β_j just take two values over the $m = 10$ datasets.

Case II (fully distinct supports of β 's) Suppose that all the supports of $\beta^{(\ell)}$ are disjoint and thus: if $\beta_j^{(\ell)} \neq 0$ it must be that $\beta_j^{\ell'} = 0$ for all $\ell' \neq \ell$. In the balanced case where $\mathbb{P}[Z = \ell] \equiv 1/m$, the pooled parameter then equals $\beta_j^{\text{pool}} = \beta_j^{(\ell)}/m$. Then, the detection boundary for coefficient j in (20) becomes

$$\frac{|\beta_j^{(\ell)}|}{\sigma^{\text{pool}} (\Sigma_X)_{jj}^{-1}} \gg \sqrt{m/n}, \tag{21}$$

and in this case, the Tippett scheme is better (assuming that σ^{pool} is comparable to $\sigma^{(\ell)}$): compare (21) to (19).

The conclusion from this little calculation is as expected, that pooling can be better than aggregation of p values if the different datasets substantially share the supports and signs of the regression coefficients, as illustrated in Figs. 7 and 8. In general, including e.g. different covariances of the covariates, pooling can be inadequate and is exposed to a misspecified model. Thus, Tippett's aggregation of p values is the safer procedure (and e.g. Stouffer's aggregation rule is not really a competitor in our setting with multi sample splitting for hierarchical testing, as pointed out in the paragraph at the end of Sect. 4.2).

4.4 Software for aggregating p values of multiple studies

It is very convenient to combine the information of multiple studies by aggregating p values as described in Sect. 4. The package `hierinf` offers two methods for jointly estimating a single hierarchical tree for all datasets using either of the functions `cluster_var` or `cluster_position`; compare with Sect. 3.1. Testing is performed by the function `test_hierarchy` in a top-down manner given by the joint hierarchical tree. For a given cluster, p values are calculated based on the intersection of the cluster and each dataset (corresponding to a study) and those p values are then aggregated to obtain one p value per cluster using either Tippett's rule (17) or Stouffer's method (18); see argument `agg.method` of the function `test_hierarchy`. The difference and issues of the two methods for estimating a joint hierarchical tree are described in the following two paragraphs.

The function `cluster_var` estimates a hierarchical tree based on clustering the SNPs from all the studies. Problems arise if the studies do not measure the same SNPs and thus, some of the entries of the dissimilarity matrix cannot be calculated. By default, pairwise complete observations for each pair of SNPs are taken to construct the dissimilarity matrix. This issue affects the building of the hierarchical tree but the testing of a given cluster remains as described before.

The function `cluster_position` estimates a hierarchical tree based on the genomic positions of the SNPs from all the studies. The problems mentioned above do not show up here since SNPs, maybe different ones for various datasets, can still be uniquely assigned to genomic regions.

The only difference in all the function calls is that the arguments `x`, `y`, and `clvar` are now each a list of matrices instead of just a single matrix. Note that the order of the list elements of the arguments `x`, `y`, and `clvar` matter, i.e. the user has to stick to the order that the first element of the three lists corresponds to the first dataset, the second element to the second datasets, and so on. One would replace the corresponding element of the list containing the control covariates (argument `clvar`) by `NULL` if some dataset has no control covariates. If none of the datasets have control covariates, then one can simply omit the argument. Note that the argument `block` defines the second level of the tree which is assumed to be the same for all datasets or studies. The argument `block` has to be a `data.frame` which contains all the column names (of all the datasets or studies) and their assignment to the blocks. The aggregation method can be chosen using the argument `agg.method` of the function `test_hierarchy`, i.e. it can be set to either "Tippett" or "Stouffer". The default aggregation method is Tippett's rule (17).

The example below demonstrates the functions `cluster_var` and `test_hierarchy` for two datasets/studies measuring the same SNPs.

```
# The datasets need to be stored in different elements
# of a list. Note that the order has to be the same
# for all the lists.

# As a simple example, we artificially split the
# observations of the toy dataset in two parts,
# i.e. two datasets.
```

```

set.seed(89)
ind1 <- sample(1:500, 250)
ind2 <- setdiff(1:500, ind1)

sim.geno.2dat <- list(sim.geno[ind1, ],
                    sim.geno[ind2, ])
sim.clvar.2dat <- list(sim.clvar[ind1, ],
                    sim.clvar[ind2, ])
sim.pheno.2dat <- list(sim.pheno[ind1],
                    sim.pheno[ind2])

# Cluster the SNPs
dendr <- cluster_var(x = sim.geno.2dat,
                   block = block,
                   # the following arguments have to
                   # be specified for parallel
                   # computation
                   parallel = "multicore",
                   ncpus = 2)

# Test the hierarchy using multi sample split
set.seed(1234)
result <- test_hierarchy(x = sim.geno.2dat,
                       y = sim.pheno.2dat,
                       clvar = sim.clvar.2dat,
                       dendr = dendr,
                       family = "binomial",
                       # the following arguments have
                       # to be specified for parallel
                       # computation
                       parallel = "multicore",
                       ncpus = 2)

```

The above R code is evaluated in parallel; compare with Sect. 3.3 for more details about the software for parallel computing.

The output shows one significant group of SNPs and one single SNP.

```

> print(result, n.terms = 4)
  block    p.value    significant.cluster
1 chrom 1 NA      NA
2 chrom 2 0.02659100 SNP.532, SNP.721, SNP.882,
  SNP.520, ... [15]
3 chrom 2 0.01100256 SNP.993

```

The significance of a cluster is based on the information of both datasets. For a given cluster, the p values of each dataset were aggregated using Tippett's rule as in (17). Those aggregated p values are displayed in the output above. We cannot judge which dataset (or both or combined) inherits a strong signal such that a cluster is shown significant but that is not the goal. The goal is to combine the information of multiple studies.

The crucial point is that the testing procedure goes top-down through a single jointly estimated tree for all the studies and only continues if at least one child is significant

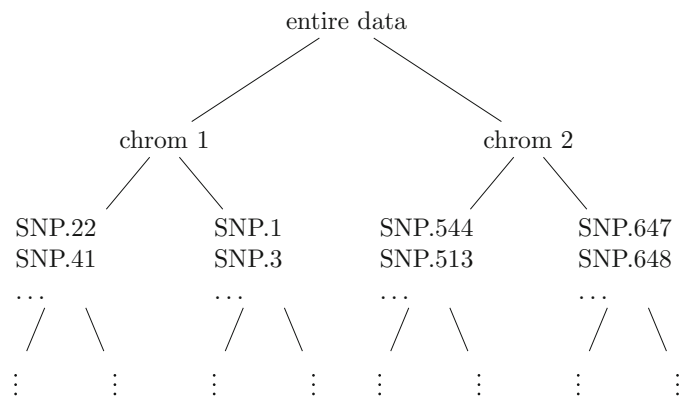


Fig. 9 Illustration of a possible single jointly estimated tree for multiple studies based on clustering the SNPs. The second level of the hierarchical tree is defined by chromosome 1 and 2 (defined by the argument block of the functions `cluster_var/cluster_position`). The function `cluster_var / cluster_position` builds a separate hierarchical tree for each of the chromosomes

(based on the aggregated p values of the multiple datasets) of a given cluster. The algorithm determines where to stop and naturally we get one output for all the studies. A possible single jointly estimated tree of the above R code is illustrated in Fig. 9. In our example, both datasets measure the same SNPs. If that would not be the case, then intersection of the cluster and each dataset is taken before calculating a p value per dataset/study and then aggregating those.

5 Discussion and conclusions

We provide a review of hierarchical inference for high-dimensional (generalized) linear models, particularly aiming for the analysis of genome-wide association studies (GWAS) where the dimensionality is in the order $O(10^6)$ and sample size typically in the thousands. Inferring statistical significance in such high-dimensional settings is very challenging: we believe that *hierarchical* inference is a very natural and powerful approach towards better and more reliable inference in GWAS. Obviously, multiple datasets or studies contain more information. We advocate the use of meta-analysis within a single hierarchical structure which is simple and coherent.

Our new implementation in the R-package `hierinf` provides many possibilities: two options for constructing hierarchical structures, fitting linear and logistic linear response models with possible additional adjustment for external control variables, and efficient parallel computation. Our software is a major cornerstone for enabling the practical use of hierarchical inference for GWAS, controlling the FWER. A different way of performing hierarchical inference can be done within the framework of selective inference for controlling the FDR or a conditional FDR (Brzyski et al. 2017; Heller et al. 2017).

Many open problems remain. Among them, we name here a few. (1) The issue of hidden confounders: even when taking all measured SNPs into the analysis, unobserved confounding can lead to spurious and wrong associations. An extreme example is given by Novembre et al. (2008), and mixed models (Rakitsch et al. 2013; Zhou et al. 2013) may only account in part for hidden confounders. (2) Another point is the

debate whether the familywise error rate (FWER) is a too strict criterion to work with, in contrast to the false discovery rate (FDR): the FWER is simpler to control, especially in hierarchical and closed testing schemes. We refer also to Goeman and Solari (2011) for an interesting discussion on this point. In the classical non-hierarchical inference, the ranking about significance of single hypotheses is not influenced whether the user chooses adjustment of p values with the Bonferroni–Holm or the Benjamini–Hochberg procedure to control the FWER or FDR, respectively. In the hierarchical case though, this remains unclear. In addition, the p values from multi-sample splitting, as used in our procedure and software, might be unreliable: it is challenging, in particular for logistic regression (Sur and Candès 2019), to come up with reliable and powerful p values for testing single or groups of regression coefficients which are reliable and powerful in high-dimensional settings. (3) The role of the hierarchy is an issue of power, as long as we assume fixed design and a correct model specification. The FWER control holds for any fixed hierarchical structure, but the power typically depends on the chosen hierarchy. We have not considered here the region-based approach from Meijer et al. (2015) which allows for a supervised form of the groups or clusters at the price of a more severe multiple testing correction: in absence of a broad comparison, we do not want to give general recommendations. Our view is (in part) to enable the users to try our approach and make their own judgment: our R-software package `hierinf` should provide substantial support to do so.

Acknowledgements We thank the Ricardo Cao for arranging the possibility present our results as a discussion paper. We also thank two anonymous reviewers for very insightful and helpful comments.

References

- Alexander D, Lange K (2011) Stability selection for genome-wide association. *Genet Epidemiol* 35:722–728
- Baierl A, Bogdan M, Frommlet F n, Futschik A (2006) On locating multiple interacting quantitative trait loci in intercross designs. *Genetics* 173:1693–1703
- Benjamini Y, Hochberg Y (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Ser B* 57:289–300
- Benjamini Y, Yekutieli D (2005) Quantitative trait loci analysis using the false discovery rate. *Genetics* 171:783–790
- Bickel P, Ritov Y, Tsybakov A (2009) Simultaneous analysis of Lasso and Dantzig selector. *Ann Stat* 37:1705–1732
- Brzyski D, Peterson CB, Sobczyk P, Candès EJ, Bogdan M, Sabatti C (2017) Controlling the rate of GWAS false discoveries. *Genetics* 205:61–75
- Bühlmann P (2013) Statistical significance in high-dimensional linear models. *Bernoulli* 19:1212–1242
- Bühlmann P (2017) High-dimensional statistics, with applications to genome-wide association studies. *EMS Surv Math Sci* 4:45–75
- Bühlmann P, Mandozzi J (2014) High-dimensional variable screening and bias in subsequent inference, with an empirical comparison. *Comput Stat* 29:407–430
- Bühlmann P, Rütimann P, van de Geer S, Zhang C-H (2013) Correlated variables in regression: clustering and sparse estimation. *J Stat Plan Inference* 143:1835–1858
- Bühlmann P, van de Geer S (2011) *Statistics for high-dimensional data: methods, theory and applications*. Springer, New York
- Bühlmann P, van de Geer S (2015) High-dimensional inference in misspecified linear models. *Electron J Stat* 9:1449–1473

- Buja A, Berk R, Brown L, George E, Pitkin E, Traskin M, Zhan K, Zhao L (2014) Models as approximations, part I: a conspiracy of nonlinearity and random regressors in linear regression. Preprint [arXiv:1404.1578](https://arxiv.org/abs/1404.1578)
- Bush WS, Moore JH (2012) Genome-wide association studies. *PLOS Comput Biol* 8:e1002822
- Buzdugan L (2019) hierGWAS: assessing statistical significance in predictive GWA studies. R package version 1.17.0. <https://www.bioconductor.org/packages/devel/bioc/html/hierGWAS.html>
- Buzdugan L, Kalisch M, Navarro A, Schunk D, Fehr E, Bühlmann P (2016) Assessing statistical significance in multivariable genome wide association analysis. *Bioinformatics* 32:1990–2000
- Cantor RM, Lange K, Sinsheimer JS (2010) Prioritizing GWAS results: a review of statistical methods and recommendations for their application. *Am J Hum Genet* 86:6–22
- Carbonetto P, Stephens M (2012) Scalable variational inference for Bayesian variable selection in regression, and its accuracy in genetic association studies. *Bayesian Anal* 7:73–108
- Chatterjee A, Lahiri S (2011) Bootstrapping Lasso estimators. *J Am Stat Assoc* 106:608–625
- Chatterjee A, Lahiri S (2013) Rates of convergence of the adaptive LASSO estimators to the oracle distribution and higher order refinements by the bootstrap. *Ann Stat* 41:1232–1259
- Dezeure R, Bühlmann P, Meier L, Meinshausen N (2015) High-dimensional inference: confidence intervals, p-values and R-software hdi. *Stat Sci* 30:533–558
- Dezeure R, Bühlmann P, Zhang C-H (2017) High-dimensional simultaneous inference with the bootstrap (with discussion). *TEST* 26:685–719
- Dolejsi E, Bodenstorfer B, Frommlet F (2014) Analyzing genome-wide association studies with an FDR controlling modification of the bayesian information criterion. *PLoS One* 9(7):e103322
- Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* 33(1):1–22
- Frommlet F, Bogdan M, Ramsey D (2016) Phenotypes and genotypes: the search for influential genes. Springer, New York
- Frommlet F, Ruhhaltinger F, Twaróg P, Bogdan M (2012) Modified versions of Bayesian information criterion for genome-wide association studies. *Comput Stat Data Anal* 56(5):1038–1051
- Goeman JJ, Finos L (2012) The inheritance procedure: multiple testing of tree-structured hypotheses. *Stat Appl Genet Mol Biol* 11:1–18
- Goeman JJ, Solari A (2010) The sequential rejection principle of familywise error control. *Ann Stat* 38:3782–3810
- Goeman JJ, Solari A (2011) Multiple testing for exploratory research. *Stat Sci* 26:584–597
- Hartigan J (1975) Clustering algorithms. Wiley, New York
- He Q, Lin D-Y (2011) A variable selection method for genome-wide association studies. *Bioinformatics* 27:1–8
- Heller R, Chatterjee N, Krieger A, Shi J, (2017) Post-selection inference following aggregate level hypothesis testing in large scale genomic data. *J Am Stat Assoc*. <https://doi.org/10.1080/01621459.2017.1375933>
- Hoggart CJ, Whittaker JC, De Iorio M, Balding DJ (2008) Simultaneous analysis of all SNPs in genome-wide and re-sequencing association studies. *PLOS Genet* 4:e1000130
- Javanmard A, Montanari A (2014) Confidence intervals and hypothesis testing for high-dimensional regression. *J Mach Learn Res* 15:2869–2909
- Klasen J, Barbez E, Meier L, Meinshausen N, Bühlmann P, Koornneef M, Busch W, Schneeberger K (2016) A multi-marker association method for genome-wide association studies without the need for population structure correction. *Nat Commun* 7:Article number 13299. <https://doi.org/10.1038/ncomms13299>
- Li J, Das K, Fu G, Li R, Wu R (2011) The Bayesian Lasso for genome-wide association studies. *Bioinformatics* 27:516–523
- Lippert C, Listgarten J, Liu Y, Kadie CM, Davidson RI, Heckerman D (2011) Fast linear mixed models for genome-wide association studies. *Nat Methods* 8:833
- Liu H, Yu B (2013) Asymptotic properties of Lasso + mLS and Lasso + Ridge in sparse high-dimensional linear regression. *Electron J Stat* 7:3124–3169
- Lu Y, Dhillon P, Foster DP, Ungar L (2013) Faster ridge regression via the subsampled randomized hadamard transform. In: *Advances in neural information processing systems*, vol 26, pp 369–377
- Malo N, Libiger O, Schork N (2008) Accommodating linkage disequilibrium in genetic-association analyses via ridge regression. *Am J Hum Genet* 82:375–385
- Mandozzi J, Bühlmann P (2016a) Hierarchical testing in the high-dimensional setting with correlated variables. *J Am Stat Assoc* 111:331–343

- Mandozzi J, Bühlmann P (2016b) A sequential rejection testing method for high-dimensional regression with correlated variables. *Int J Biostat* 12:79–95
- Meijer RJ, Krebs TJ, Goeman JJ (2015) A region-based multiple testing method for hypotheses ordered in space or time. *Stat Appl Genet Mol Biol* 14:1–19
- Meinshausen N (2008) Hierarchical testing of variable importance. *Biometrika* 95:265–278
- Meinshausen N, Bühlmann P (2006) High-dimensional graphs and variable selection with the Lasso. *Ann Stat* 34:1436–1462
- Meinshausen N, Bühlmann P (2010) Stability selection (with discussion). *J R Stat Soc Ser B* 72:417–473
- Meinshausen N, Meier L, Bühlmann P (2009) P-values for high-dimensional regression. *J Am Stat Assoc* 104:1671–1681
- Nagelkerke NJ et al (1991) A note on a general definition of the coefficient of determination. *Biometrika* 78:691–692
- Novembre J, Johnson T, Bryc K, Kutalik Z, Boyko RA, Auton A, Indap A, King K, Bergmann S, Nelson M, Stephens M, Bustamante C (2008) Genes mirror geography within Europe. *Nature* 456:98–101
- Pearl J (2000) *Causality: models, reasoning and inference*. Cambridge University Press, Cambridge
- Peterson CB, Bogomolov M, Benjamini Y, Sabatti C (2016) Many phenotypes without many false discoveries: error controlling strategies for multitrait association studies. *Genet Epidemiol* 40:45–56
- Pilanci M, Wainwright MJ (2015) Randomized sketches of convex programs with sharp guarantees. *IEEE Trans Inf Theory* 61:5096–5115
- Plagnol V, Howson JM, Smyth DJ, Walker N, Hafler JP, Wallace C, Stevens H, Jackson L, Simmonds MJ, Bingley PJ et al (2011) Genome-wide association analysis of autoantibody positivity in type 1 diabetes cases. *PLOS Genet* 7:e1002216
- R Core Team (2019) *R: a language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna. <https://www.R-project.org/>
- Rakitsch B, Lippert C, Stegle O, Borgwardt K (2013) A Lasso multi-marker mixed model for association mapping with population structure correction. *Bioinformatics* 29:206–214
- Sabatti C, Freimer N et al (2003) False discovery rate in linkage and association genome screens for complex disorders. *Genetics* 164:829–833
- Scott LJ, Mohlke KL, Bonnycastle LL, Willer CJ, Li Y, Duren WL, Erdos MR, Stringham HM, Chines PS, Jackson AU, Prokunina-Olsson L, Ding C-J, Swift AJ, Narisu N, Hu T, Pruim R, Xiao R, Li X-Y, Conneely KN, Riebow NL, Sprau AG, Tong M, White PP, Hetrick KN, Barnhart MW, Bark CW, Goldstein JL, Watkins L, Xiang F, Saramies J, Buchanan TA, Watanabe RM, Valle TT, Kinnunen L, Abecasis GR, Pugh EW, Doheny KF, Bergman RN, Tuomilehto J, Collins FS, Boehnke M (2007) A genome-wide association study of type 2 diabetes in finns detects multiple susceptibility variants. *Science* 316:1341–1345
- Shah R, Bühlmann P (2018) Goodness of fit tests for high-dimensional linear models. *J R Stat Soc Ser B* 80:113–135
- Shah R, Samworth R (2013) Variable selection with error control: another look at stability selection. *J R Stat Soc Ser B* 75:55–80
- Shao J, Deng X (2012) Estimation in high-dimensional linear models with deterministic design matrices. *Ann Stat* 40:812–831
- Shi G, Boerwinkle E, Morrison AC, Gu CC, Chakravarti A, Rao D (2011) Mining gold dust under the genome wide significance level: a two-stage approach to analysis of GWAS. *Genet Epidemiol* 35:111–118
- Stekhoven DJ, Bühlmann P (2012) Missforest - non-parametric missing value imputation for mixed-type data. *Bioinformatics* 28(1):112–118
- Storey JD, Tibshirani R (2003) Statistical significance for genomewide studies. *Proc Natl Acad Sci* 100:9440–9445
- Stouffer SA, Suchman EA, DeVinney LC, Star SA, Williams RM Jr (1949) *The American soldier: adjustment during army life*. (Studies in social psychology in World War II), vol 1. Princeton University Press, Princeton
- Sur P, Candès E (2019) A modern maximum-likelihood theory for high-dimensional logistic regression. *Proc Nat Acad Sci* 116(29):14516–14525
- The Wellcome Trust Case Control Consortium (2007) Genome-wide association study of 14,000 cases of seven common diseases and 3,000 shared controls. *Nature* 447:661–678
- Tibshirani R (1996) Regression shrinkage and selection via the Lasso. *J R Stat Soc Ser B* 58:267–288
- Tippett LHC (1931) *Methods of statistics*, 1st edn. Williams Norgate, London

- van Buuren S, Groothuis-Oudshoorn K (2011) mice: Multivariate imputation by chained equations in R. *J Stat Soft Articles* 45:1–67
- van de Geer S (2007) The deterministic Lasso. In: *JSM proceedings, 2007*, 140. American Statistical Association
- van de Geer S (2016) Estimation and testing under sparsity: École d'Été de Probabilités des Saint-Flour XLV–2015. *Lecture Notes in Mathematics*, vol 2159. Springer, New York
- van de Geer S, Bühlmann P, Ritov Y, Dezeure R (2014) On asymptotically optimal confidence regions and tests for high-dimensional models. *Ann Stat* 42:1166–1202
- Wasserman L, Roeder K (2009) High dimensional variable selection. *Ann Stat* 37:2178–2201
- Wu J, Devlin B, Ringquist S, Trucco M, Roeder K (2010a) Screen and clean: a tool for identifying interactions in genome-wide association studies. *Genet Epidemiol* 34:275–285
- Wu M, Kraft P, Epstein M, Taylor D, Chanock S, Hunter D, Lin X (2010b) Powerful SNP-set analysis for case-control genome-wide association studies. *Am J Hum Genet* 86:929–942
- Zeggini E, Weedon MN, Lindgren CM, Frayling TM, Elliott KS, Lango H, Timpson NJ, Perry JRB, Rayner NW, Freathy RM, Barrett JC, Shields B, Morris AP, Ellard S, Groves CJ, Harries LW, Marchini JL, Owen KR, Knight B, Cardon LR, Walker M, Hitman GA, Morris AD, Doney ASF, McCarthy MI, Hattersley AT (2007) Replication of genome-wide association signals in uk samples reveals risk loci for type 2 diabetes. *Science* 316:1336–1341
- Zhang C-H, Zhang S (2014) Confidence intervals for low dimensional parameters in high dimensional linear models. *J R Stat Soc Ser B* 76:217–242
- Zhao P, Yu B (2006) On model selection consistency of Lasso. *J Mach Learn Res* 7:2541–2563
- Zhou X, Carbonetto P, Stephens M (2013) Bayesian sparse linear mixed models. *PLOS Genet* 9:e1003264
- Zhou X, Stephens M (2014) Efficient multivariate linear mixed model algorithms for genome-wide association studies. *Nat Methods* 11:407–409
- Zou H (2006) The adaptive Lasso and its oracle properties. *J Am Stat Assoc* 101:1418–1429

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.