Erasmus University Rotterdam

Erasmus School of Economics

Bachelor Thesis: FEB63008

An Assessment of Variable Selection Techniques in a Rare-Weak Feature Setting

Author

P.L. ten Harmsen van der Beek

Student number: 447462

Supervisor

A.J. Koning

Second assessor: U. Karaca

Abstract

This research paper aims to investigate the validity of and relation between several feature selection methods. Specifically, in a setting where possible features are sparse and weak. Higher criticism thresholding has recently been proposed as an efficient and computationally fast alternative to false discovery rate control. Yet, beliefs on which variable selection method to implement in real-life cases remain mixed. I illustrate, using both simulations and real life examples, that higher criticism is a compromise between the Kolmogorov-Smirnov and class boundary threshold. Furthermore, in the region where signal identification is possible, the class boundary and higher criticism threshold become indistinguishable. This shows that higher criticism is an outstanding technique for feature selection. However, in cases where signal identification is possible, feature selection based on false discovery rates is just as appropriate to implement as higher criticism thresholding. In the case of correlation between features the use of innovated higher criticism has recently been advocated. Both the simulated results and empirical data sets show that the power of innovated higher criticism is not significantly higher than the one of higher criticism thresholding. This is in contrast with previous literature. The combination of these results sheds new light on several variable selection techniques and lays a foundation for further research.

July 2, 2020

The views stated in this thesis are those of the author and not necessarily those of the supervisor, second assessor, Erasmus School of Economics or Erasmus University Rotterdam.

Contents

1 Introduction

Feature selection, the process of selecting those features which have the greatest power in predicting the variable of interest, is of great importance in a variety of research fields. Take for example the identification of specific galaxies in astronomy or the production of maps of human settlements, which can be used for investigating population movement. These studies involve a numerous amount of possible features and although performing feature selection in these settings is difficult, if done incorrectly the accuracy and consequently the relevancy of the developed models can vanish (Cantú-Paz et al., 2004). On the one hand, the growth and evolution of big data the past decades has complicated the process of feature selection. On the other hand, it has offered opportunities for the development of new techniques and novel approaches for the improvement of models and the provision of solutions to real-world applications (Dash & Liu, 1997; Donoho & Jin, 2015). Therefore it is of importance to investigate how relevant features can be selected in a real-world big data setting.

Especially in a rare-weak (RW) feature setting the process of identifying features is complicated (Donoho & Jin, 2008). A RW effects setting has two main characteristics. First of all, relatively few of the measured features are expected to be relevant, or in other words show any deviation from the global null hypothesis of no effect. This is referred to as the *effect sparsity*. Secondly, when features do have predictive power, the effect size, or signal strength, is weak. This is referred to as the *effect weakness* (Donoho & Jin, 2015). Settings like these are common in application fields such as genomics, proteomics and astronomy. Therefore, finding an optimal method for identifying signals in a RW setting could contribute to studies and their accompanying discoveries in these fields. This in turn could have major economic, health and well-being implications for the population which endorses the societal relevance of this research.

Previous literature discusses several selection strategies in multiple hypothesis testing. First introduced was the Bonferroni-based family-wise error rate. The family-wise error rate (FWER) is the probability of at least one incorrect rejection of the null hypothesis, and thus at least one false conclusion. This type of false rejection is also called a Type I error. When more hypotheses are tested at the same time, the chance of oberving a rare event increases and thus the probability of incorrectly rejecting a null hypothesis increases. This can be corrected for by testing each individual hypothesis with an adjusted significance level, based on the number of hypotheses (Dunn, 1958). This technique is named the Bonferroni-based FWER as the underlying proof for this correction is based on the Bonferroni inequality. A different technique based on controlling the proportion of falsely rejected hypotheses is the false discovery rate (FDR). To be specific, FDR controls the expected proportion of Type I errors among all rejected hypotheses, also named discoveries. In contrast, FWER controls the probability of making false discoveries among all hypotheses. The FDR method offers a substantial gain in power compared to the Bonferroni FWER (Benjamini & Hochberg, 1995). Another threshold, closely linked to the FDR, is based on an equal probability of a feature belonging to the null or alternative component. This threshold is known as the class boundary (CB) (Klaus & Strimmer, 2013). A more recently favoured approach is called Higher Criticism (HC) thresholding. The HC method uses a second-level test-statistic computed from first-level p-values. This technique is especially advocated for RW feature settings (Klaus & Strimmer, 2013).

On the one hand HC thresholding was shown to outperform strategies based on FDRs or FWER when applied to feature selection (Donoho & Jin, 2008, 2015). On the other hand, studies have proven that FDR is perfectly useful for the identification of features and controlling the FDR is equivalent to HC thresholding (Ahdesmäki, Strimmer, et al., 2010; Klaus & Strimmer, 2013). In fact, Klaus and Strimmer (2010) show that, in the RW model when signal identification is possible, the thresholds resulting from FDR and HC approaches are practically indistinguishable. When the thresholds are notably different, the HC threshold leads to the inclusion of more false positives than the CB threshold and thus the CB threshold is more cautious (Klaus & Strimmer, 2013). These mixed beliefs cause disunity in the methods applied in real-world applications. Some recent studies have adapted the use of HC thresholding when selecting variables (Mihunov et al., 2019) while others still opt for FDR tools (Monroy-Vilchis et al., 2019). The aim of this paper is to settle the division on the use of these thresholding methods and to provide advice for future real-world studies. Therefore, the following research question is formulated:

What is the most effective method for identifying variables in a rare-weak feature setting?

Specifically, I state the following hypotheses:

H1: In a RW setting, HC thresholding outperforms both CB and FDR strategies in terms of prediction errors.

H2: When variable identification is possible, the HC and CB threshold are not significantly different.

H3: Using the HC threshold leads to the inclusion of more false positives than CB and FDR thresholds.

One assumption of HC thresholding is independence between features (Klaus & Strimmer, 2013). This assumption is unlikely to hold in many real-world applications. For instance, in genomics there is a high chance of correlation between markers due to chemical similarities or spatial dependence (Zuber & Strimmer, 2009). Several ways have been proposed to incorporate this dependence. One option is to use an autoregressive model to take dependencies into account (Hand, 2008). Ahdemäski and Strimmer (2010) propose thresholding correlation adjusted t-scores (CAT scores). This method is also applied by Klaus and Strimmer (2013) both in the case of FDR and HC thresholding. Hall and Jin (2010) argue that the possible correlation in the data can be exploited to improve the performance of HC thresholding by taking the correlation into account when setting the detection boundary. They call this concept innovated higher criticism (iHC). This concept has, to my knowledge, not yet been applied or extended in other studies. In this paper, I will apply this concept and generalize it to cases where the correlation matrix can be estimated from the data. This novel and underresearched concept could therefore contribute to the existing literature. Hence, I formulate my last hypothesis.

H4: Innovated higher criticism results in a higher power and lower prediction error than HC thresholding.

To find an answer to the hypotheses and main research question, I conduct a simulation study and apply the thresholding methods to four real-world cancer data sets. The results support the part of the literature which suggests that both HC and FDR based thresholding methods are appropriate methods in a RW feature setting. In case of a correlation structure between features, the results show significantly different patterns from the situation of no correlation. In contrast with the theory posed by Hall and Jin (2010), iHC thresholding does not lead to a significantly higher power than HC variable selection. Therefore, it is of importance for future research to investigate the underlying causes for this difference in results before the merit of iHC can be validated.

The remainder of this paper is structured as follows. First, the models for analysing the research question are presented in Section 2. Section 3 describes the experimental design for both the simulations and empirical evaluation. The corresponding results are presented in Section 4. Thereafter, possible practical applications of the discussed techniques in the economic sector are discussed in Section 5. Lastly, conclusions are drawn and recommendations for further research are given in Section 6.

2 Methodology

In this section, several methods for determining relevant features are presented. These methods can be applied to, for example, assigning data samples to a certain class. In genomics, disease classification is of great importance. Gene expression data can contain large numbers of features which can be used to distinguish among disease subtypes. To select the relevant features for this classification, I propose a combination of linear discriminant analysis and the several methods presented.

The RW model is a normal mean mixture model

$$Z \sim (1-\epsilon)N(0,1) + \epsilon N(\tau,1) \tag{1}$$

where $\epsilon \in [0;1]$ describes the sparsity of the relevant features and $\tau \in [0;\infty]$ describes the effect size of these relevant features. Thus, in a RW setting, where features are sparse and weak, both ϵ and τ are small. When faced with a set of features of sample size d, it is of interest to test which of the following two hypotheses is true for each feature $i \in (1, ..., d)$:

H0: feature i comes from the null component N(0,1)

H1: feature i belongs to the alternative distribution N(τ,1)

Several techniques for this problem are presented in the remainder of this section.

2.1 Family-Wise Error Rate

In statistics, a Type I error is defined as the false rejection of a true null hypothesis. Such a rejection is also known as a false positive. In a single hypothesis setting the per-comparison error rate (PCER) α is controlled by controlling the probability of a Type I error for each hypothesis. This idea was extended to a multiple hypothesis setting by controlling the probability of falsely rejecting at least one true null hypothesis (Saunders, 2014). This approach is more commonly known as the Family-Wise Error Rate (FWER). By controlling the FWER, the PCER is also controlled, yet at a lower level.

The simplest and most frequently applied FWER method is the Bonferroni correction method. This method controls the FWER at a level α when each individual test H_i is controlled at a PCER level α_i under the condition that

$$\alpha_i = \frac{\alpha}{d} \quad \text{for i = 1,...,d} \tag{2}$$

where d is the number of hypotheses tested simultaneously. This condition can be proven by Bonferroni's inequality, which is based on Boole's inequality (Boole, 1847).

Boole's inequality states that, for a countable set of events A1, A2, A3, ... the following holds.

$$\mathbb{P}(\bigcup_i A_i) \leq \sum_i \mathbb{P}(A_i). \tag{3}$$

When applying this inequality to the definition of the FWER, the proof of condition (2) appears. In this proof, d_0 denotes the number of true null hypotheses and p_i the p-value corresponding to hypothesis H_i.

$$FWER = \mathbb{P}\{\bigcup_{i=1}^{d_0}(p_i \leq \frac{\alpha}{d})\} \leq \sum_{i=1}^{d_0}\{\mathbb{P}(p_i \leq \frac{\alpha}{d})\} = d_0\frac{\alpha}{d} \leq d\frac{\alpha}{d} = \alpha \tag{4}$$

The Bonferroni procedure is used frequently in clinical trials and genome-wide studies due to its simplicity. However, it is conservative and rejects too many null hypotheses, especially if the number of simultaneously tested hypotheses is large as in our RW feature setting (D. Wang et al., 2015).

2.2 False Discovery Rate

A more powerful method to control Type I errors in a multiple hypothesis setting was introduced by Benjamini and Hochberg in 1995. They control the expected proportions of errors among all the rejected hypotheses and call this the false discovery rate (FDR) (Benjamini & Hochberg, 1995). Due to this construction, the FDR is equal to the FWER when all tested hypotheses are true.

This method can be split up into two variants. One based on distributions (tail-based FDR) and the other based on densities (local FDR). The tail-based FDR is defined on the p-value scale as follows:

$$Fdr(x) = Pr(H_0|X \leq x) = \frac{\eta_0 F_0(x)}{F(x)} = \frac{\eta_0 x}{F(x)} \tag{5}$$

Thus, Fdr(x) is equal to "the proportion of p-values from the null component found among all p-values smaller than x" (Klaus & Strimmer, 2013). Benjamini and Hochberg (1995) introduced an empirical procedure for controlling the Fdr at level $q \cdot d_0/d \leq q$ where d_0 is the number of true null hypotheses. In this way, the Fdr is exactly equal to the q-value threshold if all tested hypotheses are true and smaller otherwise. This procedure is as follows. First, let $p_{(1)}, p_{(2)}, ..., p_{(d)}$ be the observed p-values sorted in ascending order. Then,

$$k = max\{i : p_{(i)} \leq \frac{i}{d}q\} \tag{6}$$

Subsequently, the hypotheses $H^0_{(0)}, ..., H^0_{(k)}$ are rejected and the corresponding features are thus identified as signals (Benjamini & Hochberg, 1995).

The local FDR is defined as the probability of the null hypothesis under the observed data:

$$fdr(x) = Pr(H_0|X = x) = \frac{\eta_0}{f(x)} \tag{7}$$

If the roles of the alternative and null hypothesis are switched the false non-discovery rate (FNDR) appears. This rate can be viewed as an extension of the Type II error in a multiple hypothesis setting. Specifically, it is defined as the proportion of falsely accepted null hypotheses among all true alternative hypotheses. This concept relates to the power of a test, which is identified as the fraction of features for which a test correctly rejects the null hypothesis when an observation belongs to the alternative. Namely, the power of a test can be calculated as $1 -$ FNDR. The two variants of the FNDR are defined as follows (Klaus & Strimmer, 2013):

$$Fndr(x) = Pr(H_1|X \geq x) = (1 - \eta_0)\frac{1 - F_A(x)}{1 - F(x)} \tag{8}$$

$$fndr(x) = Pr(H_1|X = x) = 1 - fdr(x) \tag{9}$$

The FNDR can be used to identify the true null features. Therefore, the FNDR of a test is an indication of the quality of that procedure.

2.2.1 Variable Selection with False Discovery Rate Thresholding

Relevant features can be selected in several ways using FDR thresholding techniques. A common approach is to set a cutoff value of q in the procedure of Benjamini and Hochberg and to select all features for which the null hypothesis is rejected. This cutoff value is commonly set to 0.05. Another approach is to control the local FDR by requiring $\hat{fdr}(x)$ to be smaller than or equal to 0.2. This conventional threshold corresponds to q-values between 0.05 and 0.15. Pragmatically, it is shown that when increasing the local FDR threshold much above 0.20, this results in high proportions of Type I errors (Efron, 2005).

When the aim is to identify the true null features we impose $\hat{fndr}(x) \leq 0.2$. This is the method we adopt in the rest of this paper. A natural division between the null and alternative features is obtained when we set $f\hat{ndr}(x) = f\hat{dr}(x) = 0.5$. The x for which this condition is satisfied is called the *Class Boundary* (CB) (Klaus & Strimmer, 2013). The CB threshold, $\hat{t}^{CB}$, in a RW feature setting can be expressed analytically as

$$\hat{t}^{CB} = \frac{\tau}{2} + \frac{1}{\tau} log(\frac{1-\epsilon}{\epsilon}) \tag{10}$$

2.3 Higher Criticism

In a rare-weak feature setting, it may easily happen that no predictor has a sufficiently small false discovery rate to be called significant. An alternative approach to select variables which keeps the missed-feature detection rate better under control, is proposed by Donoho and Jin (2008). This method is called higher criticism (HC) and is based on the supremum of a standardized empirical process under the null hypothesis. An empirical process is a stochastic process that represents the proportion of objects in a system in a given state. Empirical processes are useful in the sense that they can be used to establish large sample properties of test statistics and estimators (Andrews, 1994). This characteristic forms the basis of the motivation for HC thresholding.

HC thresholding works as follows. Suppose we are in a situation where we have d features and for each feature a corresponding test statistic $y_1, ...y_d$. For each test statistic y_i the corresponding two-sided p-value p_i is calculated and subsequently these p-values are sorted in ascending order. Next, the HC statistic is calculated for each feature:

$$HC(i;p_i) = \sqrt{d} \frac{i/d - p_i}{\sqrt{p(i)(1-p(i))}} \tag{11}$$

By maximizing over the empirical HC statistics, the HC test statistic $\hat{HC}^*$ is obtained:

$$\hat{HC}^* = \max_i HC(i;p_i) \tag{12}$$

Using the theory of empirical processes, Donoho and Jin (2004) show that for HC thresholding under the null hypothesis H_0,

$$\frac{\hat{HC}_n^*}{\sqrt{2\log\log(n)}} \xrightarrow{p} 1, \quad \text{as} \quad n \to \infty \tag{13}$$

From this it follows that for every combination of τ and ϵ where a likelihood ratio test would completely separate the null and alternative hypothesis, the power of HC feature selection will converge to one (Donoho, Jin, et al., 2004). This demonstrates the theoretical merit of HC thresholding.

Another way to determine the HC threshold is to solve the squared empirical objective function. When both the null and alternative distribution are known, this can be generalised to the population level as

$$HC(x)^2 \propto \frac{(F_A(x) - F_0(x))^2}{F(x)(1 - F(x))} \tag{14}$$

This formulation does not depend on the number of observations and thus can be used to determine the theoretical thresholds, depending only on τ and ϵ (Klaus & Strimmer, 2013) .

2.3.1 Variable Selection with Higher Criticism Thresholding

When selecting features, applying the HC algorithm yields $\hat{HC}^*$ at index $\hat{\imath}^{HC}$. Then the HC threshold is the absolute value of the test statistic corresponding to feature $\hat{\imath}^{HC}$. Thus, $\hat{t}^{HC} = |Z|_{\hat{\imath}}$. All features with test statistics exceeding $\hat{t}^{HC}$ are selected as relevant predictors. Signal identification with HC thresholding is only possible when $\tau \geq \sqrt{-2\log(\epsilon)}$. Below this threshold only the presence but not the location of a signal can be determined (Klaus & Strimmer, 2013).

An interesting observation is that the HC objective function is invariant against transformations in the test statistics. This property can be used to apply HC directly to, for example, correlation adjusted t (CAT) scores instead of only to a set of p-values. Klaus and Strimmer (2013) take advantage of this property and apply both FDR and HC variable selection techniques to CAT scores to take into account possible correlation between features. In this way, the expected power of the tests in the multiple comparison problem rises. These CAT scores are the decorrelated gene-specific t-scores between the mean of a class and the pooled mean. An advantage of using CAT scores, as opposed to other ways to incorporate dependence, is that CAT scores do not require an adjustment of the RW model. Additionally, when there is no correlation, the CAT score reduces to the standard t-score (Zuber & Strimmer, 2009). However, the multiple comparison problem remains.

2.4 Kolmogorov-Smirnov Threshold

The HC statistic can be seen as the standardized Kolmogorov-Smirnov (KS) statistic. Therefore, we can also apply the KS statistic in a same way as the HC statistic in order to find a decision threshold (Klaus & Strimmer, 2013). The KS statistic is given as

$$\sup_x |F_A(x) - F_0(x)| \tag{15}$$

and we can analytically express the KS threshold, $\hat{t}^{KS}$ as

$$\hat{t}^{KS} = \frac{\tau}{2} \tag{16}$$

2.5 Innovated Higher Criticism

Instead of modeling the correlation through CAT scores, Hall and Jin (2010) propose an advanced method of higher criticism to take advantage of the potential dependence structures. This method is also referred to as innovated higher criticism (iHC). Several assumptions are taken before applying iHC. In cases where the correlation decays slowly, the detection boundaries can be identified quite precisely. Therefore, iHC is examined under the assumption of a correlation matrix Σ_d which has polynomial off-diagonal decay. Mathematically, this can be expressed as

$$\Theta_d^*(\lambda, c_0, M) = \{\Sigma_d \in \Theta_d : |\Sigma_d(j,k)| \leq M(1 + |j-k|^{-\lambda}, ||\Sigma_d|| \geq c_0\} \tag{17}$$

where Θ_d is the set of n by n correlation matrices and $\lambda \geq 1$. In addition, the operator norm $||\Sigma_d||$ is uniformly bounded from below. For these matrices, their inverse as well as their Cholesky factorization decay with the same rate as the correlation matrix itself (Hall, Jin, et al., 2010). This property makes it possible to characterize the detection boundary for the identification of signals.

The iHC statistic is constructed according to the following steps. First of all, the Cholesky factorization of the correlation matrix is calculated such that $U_d \Sigma_d U_d^T = I_d$. This Cholesky factorization forms an computationally effective upper triangular matrix U_n. Applying standard HC to $U_d X$ will yield a higher power than applying HC to X directly, where X is a d-dimensional vector of signals in a RW setting (Hall, Jin, et al., 2010). However, due to this transformation the pattern of the signals also changes and appears in clusters as can be seen in Figure 1. In this figure, a signal vector μ is simulated where the signal is equal to 1.357 for 40 features and 0 for the other features. After multiplying the signal vector with U, the signals appear in clusters and 67 features display a signal bigger than 0. By remodeling the signals from clusters to singletons, the signal strength increases and thus the potential power of the variable selection process enhances. This remodeling is done for $U_d = (u_{kj})_{\{1 \le k,j \le d\}}$ as follows

$$\tilde{U}(b_d) = (\tilde{u}_{kj})_{1 \le j,k}, \tilde{u}_{kj} = \begin{cases} u_{kj}, & \text{if } k - b_d + 1 \le j \le k \\ 0, & \text{otherwise} \end{cases} \tag{18}$$

where b_d is a bandwidth which ensures that the lower off-diagonal elements are equal to zero if they are more than or equal to b_d rows in distance from the central diagonal. Hall and Jin (2010) advice to set $b_d = log(d)$ to balance the tradeoff between stronger signals and stronger correlated noise as b_d increases.

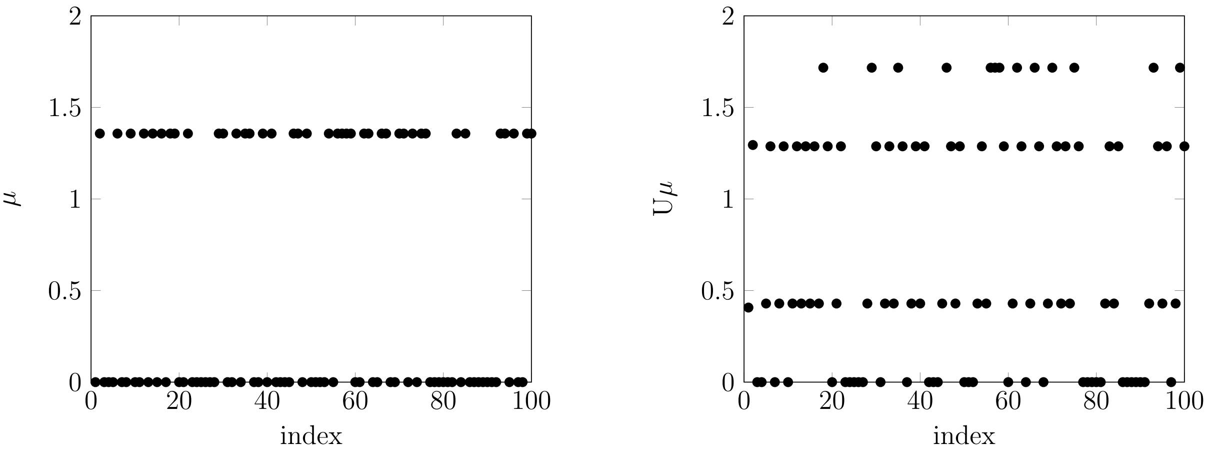

Figure 1: Comparison of μ and Uμ for n = 100 and a correlation matrix with 1 on the main diagonal, 0.3 on the two sub-diagonals and 0 elsewhere.

Next, each column of $\tilde{U}(b_d)$ is normalized by its l^2 norm, the square root of the sum of the elements, such that this norm equals 1 for every column. The resulting matrix is denoted by $\bar{U}(b_d)$. The last step is to apply higher criticism to $\bar{U}_d^T U_d X$, which is also denoted as VX. This results in the following definition:

$$iHC_d^*(b_d) = \frac{1}{\sqrt{2b_d - 1}} \sup_i \{\sqrt{d}\frac{i/d - p(i)}{p(i)(1 - p(i))}\} \tag{19}$$

2.5.1 Variable Selection with Innovated Higher Criticism Thresholding

Applying the iHC algorithm yields $iHC_d^*(b_d)$ at index $i^{i\hat{H}C}$. Just as when implementing the HC algorithm, the iHC threshold is the absolute value of the test statistic corresponding to feature $i^{i\hat{H}C}$. So, $\hat{t}^{iHC} = |VX|_{i^{i\hat{H}C}}$. Subsequently, all features with test statistics exceeding $\hat{t}^{iHC}$ are selected as possible predictors.

When there is a correlation structure between features, the region where signal identification is possible differs from the situation under the assumption of independence. In case of a matrix with constant elements on the diagonals, which is also called a Toeplitz matrix, the threshold for the detectable region changes with a factor $C(f)$. Here, $C(f) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{1}{f(\theta)} d\theta$, where $f(\theta)$ is the underlying function for the Toeplitz matrix (Hall, Jin, et al., 2010). In terms of the parameters τ and ϵ this means that signal identification with iHC thresholding is possible when $\tau \geq C(f) \cdot \sqrt{-2\log(\epsilon)}$.

2.6 Generalised Innovated Higher Criticism

The previously introduced innovated higher criticism assumes a correlation matrix with off-diagonal polynomial decay. However, in real-life applications this matrix often has to be estimated non-parametrically. To investigate the performance of iHC in such situations, I apply iHC to real-life data sets and estimate the correlation matrix using the Pearson correlation, a technique often applied in fields such as genomics (Cheverud, 2001; Li & Ji, 2005). The Pearson correlation is calculated as

$$\rho_{XY} = \frac{Cov(X,Y)}{Var_X Var_Y} \tag{20}$$

where X and Y form a given set of two random variables. For example, this can be a set of gene expressions for two type of genes.

In this paper I apply iHC to a real-life data set using the non-parametric technique for inducing the corrrelation matrix. Then I compare the predictive abilities of iHC with previously introduced feature selection techniques. This way, I expect to show that iHC has theoretical advantages over HC, CB and FDR thresholding but does not perform as well empirically.

2.7 Linear Discriminant Analysis

To use the several variable selection techniques in fields such as genomics, we need to manipulate our data before applying the techniques. Namely, the techniques select features from a certain set of possible features based on a vector of p-values. In real-life data sets, often a matrix of test statistics is given instead of a vector. For example, in genome studies a gene expression is given for each possible feature for n number of observations (persons). Therefore, to be able to apply the proposed techniques, we need to transform the data such that for each possible feature the vector of statistics is reduced to one value. To do this we make use of shrinkage estimators and implement linear discriminant analysis (LDA). This method is used to predict the probability of a sample belonging to a specific class. In this way, we can test the predictive performance of the several decision thresholds.

LDA assumes the following mixture model for the d-dimensional data $\mathbf{x}$.

$$f(\mathbf{x}) = \sum_{j=1}^{K} \pi_j f(\mathbf{x}|j) \tag{21}$$

where K is the number of classes in the data set, each represented by a multivariate normal density.

$$f(\mathbf{x}|k) = (2\pi)^{-p/2}|\mathbf{\Sigma}|^{-1/2} \exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_k)^T \mathbf{\Sigma}^{-1}(\mathbf{x} - \boldsymbol{\mu}_k)\} \tag{22}$$

In this formulation, $\boldsymbol{\mu}_k$ are the class specific centroids, which are calculated by the empirical means, and $\mathbf{\Sigma}$ is the common covariance over all classes (Fisher, 1936). The LDA discriminant score d_k^{LDA} for a sample $\boldsymbol{x}$ can then be calculated for every class according to

$$d_k^{LDA}(\boldsymbol{x}) = \boldsymbol{\mu}_k^T \mathbf{\Sigma}^{-1} \boldsymbol{x} - \frac{1}{2} \boldsymbol{\mu}^T \mathbf{\Sigma}^{-1} \boldsymbol{\mu}_k + log(\pi_k) \tag{23}$$

Subsequently, sample $\mathbf{x}$ is assigned to the class corresponding to the highest LDA discriminant score.

The standard form of the LDA discriminant function (23) can be transformed into another formulation which is better interpretable:

$$\Delta_k^{LDA}(\boldsymbol{x}) = (\boldsymbol{\omega}^{(k,pool)})^T \boldsymbol{\delta}_k(\boldsymbol{x}) + log(\pi_k) \tag{24}$$

In this formulation, $\boldsymbol{\omega}_k$ is called the feature weight vector and $\boldsymbol{\delta}_k(\boldsymbol{x})$ a transformed predictor (Ahdesmäki, Strimmer, et al., 2010). These vectors can be decomposed as

$$\boldsymbol{\delta}_k(\boldsymbol{x}) = \boldsymbol{P}^{-1/2} \boldsymbol{V}^{-1/2} (\boldsymbol{x} - \frac{\boldsymbol{\mu}_k + \boldsymbol{\mu}_{pool}}{2}) \tag{25}$$

$$\boldsymbol{\omega}^{k,pool} = \boldsymbol{P}^{-1/2} \boldsymbol{V}^{-1/2} (\boldsymbol{\mu}_k - \boldsymbol{\mu}_{pool}) \tag{26}$$

where $\boldsymbol{\mu}_{pool}$ is the pooled mean over all classes calculated as $\boldsymbol{\mu}_{pool} = \sum_{k=1}^{K} \frac{n_k}{n} \boldsymbol{\mu}_k$. Furthermore, $\boldsymbol{P} = (\rho_{ij})$ is the correlation matrix and $\boldsymbol{V}$ is a diagonal matrix containing the variances $\boldsymbol{V} = diag\{\sigma_1^2, ..., \sigma_d^2\}$.

The LDA discriminant function (24) is constructed by three shrinkage rules, for the correlations $\boldsymbol{P}$, variances $\boldsymbol{V}$ and proportions π_k, which are based on the theory of James-Stein estimation (James & Stein, 1992). James and Stein proved that an estimator exists which performs better than the population mean in case of several unknown population means. This estimator is based on the fact that for large sample sizes the magnitude of the k-dimensional estimator vector $\mathbf{X}$ is expected to be much more substantial than the magnitude of the estimand vector θ. To correct for this difference in size, James and Stein proposed the shrinkage estimator

$$\hat{\boldsymbol{\theta}} = (1 - \frac{(k-2)}{||\boldsymbol{X}||^2})\boldsymbol{X} \tag{27}$$

These kind of estimators are appropriate for analysing large-dimensional data as they are computationally efficient and hard to improve. Additionally, they can be constructed without making any assumptions on the distribution of the data or the model parameters (Opgen-Rhein & Strimmer, 2007). The key advantage of shrinkage estimation is that it reduces the mean squared error of the sample estimator and subsequently improves the accuracy of the classification rule. The three rules for constructing the LDA discriminant function (24) are discussed in more detail below.

2.7.1 Variances: Opgen-Rhein and Strimmer (2007)

The shrinked variances are estimated by the median of the empirical variances v_i. $v_i^{shrink} = \hat{\lambda_1} v_{median} + (1 - \hat{\lambda_1})v_i$, where

$$\hat{\lambda_1} = min(1, \frac{\sum_{i=1}^{d} \hat{Var}(v_i)}{\sum_{i=1}^{d}(v_i - v_{median})^2}) \tag{28}$$

2.7.2 Correlations: Schäfer and Strimmer (2005)

$\boldsymbol{P}$ is estimated by shrinking the empirical correlations towards zero according to the following rule. $r_{ij}^{shrink} = (1 - \hat{\lambda_2})r_{ij}$, where

$$\hat{\lambda_2} = min(1, \frac{\sum_{i \neq j} \hat{Var}(r_{ij})}{\sum_{i \neq j} r_{ij}^2}) \tag{29}$$

2.7.3 Proportions: Hausser and Strimmer (2009)

The class frequencies are estimated as

$$\hat{\pi}_j^{shrink} = \frac{1}{K} + \frac{n_j}{n} \tag{30}$$

3 Experimental Design

To study the relationship between and to compare the results of the several feature selection methods in a RW feature setting, I analyse both simulated and empirical data.

3.1 Theoretical Evaluation

3.1.1 No Correlation

I identify the decision threshold for the KS, CB and HC threshold under the assumption of independent features for combinations of $\tau \in 2, 3, 4, 5, 6$ and $\epsilon \in 0, 0.001, 0.01, 0.1, 0.5$. The KS and CB threshold are determined analytically by solving equations (16) and (10) respectively. To establish the HC threshold, I numerically solve equation (14).

To compare various output measures of the proposed methods, a simulation study is conducted. To be specific, I compare the number of false positives (FP), false negatives (FN), true positives (TP), true negatives (TN) and total errors (TE). To determine these output measures, I sample 10,000 random z-scores from the mixture model (1) with as input combinations of τ and ϵ such that the model is located at the detection boundary and above. To be specific, I set $\epsilon \in 0.01, 0.05, 0.1, 0.3, 0.5$ and $\tau \in 3, 4, 5, 6$. Next, I transform the simulated z-scores into two sided p-values. Subsequently, the HC threshold is determined by maximization of (10). Furthermore, the CB threshold is determined by setting local FDR $= 0.5$ and the FNDR threshold by setting local FDR $= 0.8$. To do so, the R-package *fdrtool* is employed. Next, I investigate for each threshold the number of FP, FN, TP, TN and TE. This simulation is repeated 1,000 times to determine the means and standard errors of the output measures.

3.1.2 With Correlation

To compare the thresholds in case of correlation, we repeat the simulation under the assumption of a correlation matrix with off-diagonal polynomial decay. Specifically, this involves the following steps. First of all, the correlation matrix is simulated. To be precise, I simulate a tri-diagonal Toeplitz matrix which is generated according to $f(\theta) = 1 + 2\rho\cos(\theta), |\rho| \leq 0.35$.

I let ρ range from -0.35 to 0.35 with increments of 0.05. This results in a matrix with 1 on the main diagonal, ρ on the two sub-diagonals and zero elsewhere. Second, I simulate 1,000 signals for each combination of ρ, $\tau \in 3, 4, 5, 6$ and ϵ at 0.01. For these combinations of τ, ϵ and ρ, the model is located above the detection boundary.

Given the generated correlation matrix, I generate a Gaussian vector with zero mean vector. Following from this, the generated noise vector is added to the simulated signal vector. The third step is to determine the thresholds. Again, the HC threshold is determined by maximizing equation (10) and the CB and FNDR threshold by setting local FDR $= 0.5$ and local FDR $=$ 0.8 respectively. Next to this, the iHC threshold is calculated by solving equation (19). Lastly, I investigate the number of FP, FN, TP, TN and TE for each threshold. Considering the computer power needed for these simulations, I repeat this simulation 500 times to determine the means and standard errors of the output measures.

3.2 Empirical Evaluation

To evaluate the various proposed thresholding techniques, I also apply them to a real-life data set in the field of genomics. In this way, the empirical performance of the proposed variable thresholding techniques can be demonstrated. The experimental data consist of 4 data sets of various cancer types. Each of these data sets contains gene expression values and multiple classes to which each sample can belong. This makes these data sets appropriate to test the predictive performance of the proposed methods. Specifically, I will investigate data on prostate cancer (Singh et al., 2002), lymphoma cancer (Alizadeh et al., 2000), small round blue cell tumors (Khan et al., 2001) and brain cancer (Pomeroy et al., 2002).

Applying linear discriminant analysis in combination with several thresholding methods yields a number of selected variables for each method. Subsequently, these selected variables are used to predict the class of each sample. Following from this, the prediction error of each thresholding technique can be calculated as the fraction of samples assigned to the wrong class. To estimate the prediction error I conduct 10-fold cross validations with 20 repetitions. In this approach, the training set is split into 10 smaller sets which are used to classify the test set. To implement the linear discriminant analysis, I employ the R-package *sda* (Ahdesmaki et al., 2015).

4 Results

4.1 Comparison of Thresholds

The theoretical CB, HC and KS thresholds can be found in Table 1. The results illustrate that as τ is growing, the CB and HC thresholds grow closer. Additionally, for $\epsilon = \frac{1}{2}$ both the CB and HC thresholds reduce to the KS threshold, which is line with the theory. An important observation is that the HC threshold can be seen as a compromise between the KS and CB threshold. This holds for each combination of τ and ϵ. From this we can expect HC thresholding to include more false positives than when the CB threshold would be employed. Lastly, with growing ϵ and thus parameters further in the recoverable region where signal identification is possible, the CB and HC thresholds become increasingly similar. These results are in line with the theory and statements from previous literature (Klaus & Strimmer, 2013).

Table 1: Decision Thresholds for Several Proposed Methods

	KS	CB	HC
$\boldsymbol{\tau = 2}$			
$\varepsilon = 0$	1	∞	3.3514
$\epsilon = 0.001$	1	4.45337	3.0707
$\epsilon = 0.01$	1	3.29755	2.5203
$\epsilon = 0.1$	1	2.09861	1.7574
$\epsilon = 0.5^*$	1	1	1.0000
$\boldsymbol{\tau = 4}$			
$\epsilon = 0$	2	∞	7.6667
$\epsilon = 0.001^*$	2	3.72669	3.6377
$\epsilon = 0.01^*$	2	3.14878	3.0965
$\epsilon = 0.1^*$	2	2.54831	2.5268
$\epsilon = 0.5^*$	2	2	2.0000
$\boldsymbol{\tau = 6}$			
$\epsilon = 0$	3	∞	8.1607
$\epsilon = 0.001^*$	3	4.15113	4.1454
$\epsilon = 0.01^*$	3	3.76585	3.7631
$\epsilon = 0.1^*$	3	3.36620	3.3652
$\epsilon = 0.5^*$	3	3	3.0000

	KS	CB	HC
$\boldsymbol{\tau = 3}$			
$\epsilon = 0$	1.5	∞	5.5305
$\epsilon = 0.001$	1.5	3.8023	3.4927
$\epsilon = 0.01$	1.5	3.0317	2.8406
$\epsilon = 0.1^*$	1.5	2.2324	2.1452
$\epsilon = 0.5^*$	1.5	1.5	1.5000
$\boldsymbol{\tau = 5}$			
$\epsilon = 0$	2.5	∞	8.1607
$\epsilon = 0.001^*$	2.5	3.8814	3.8567
$\epsilon = 0.01^*$	2.5	3.4190	3.4059
$\epsilon = 0.1^*$	2.5	2.9394	2.9343
$\epsilon = 0.5^*$	2.5	2.5	2.5000

Note: Signal identification is possible as $\tau \geq \sqrt{-2\log(\epsilon)}$ when marked with asterisk (*).

4.2 Comparison of Output Measures

The results from the simulation of the output measures of the CB, HC and FNDR thresholding techniques in the case of independence between features can be found in Figure 2 for $\epsilon = 0.01$. The standard errors are not included in the plots as they disorganize the plots due to their amplitude. Yet, as Table 2 shows, the standard errors grow smaller as τ becomes larger and overall they are largest when implementing HC. As expected from the theoretical thresholds, HC produces more false positives than CB thresholding. Furthermore, with growing τ, CB and HC yield increasingly similar results. This can be attributed to the fact that a larger τ means that the effect weakness is smaller and thus signals are more distinguishable. As a result, the error structure becomes more similar and the power of the methods rises.

When the signals are weak, variable selection using HC leads to the most false positives. In contrast, CB is more cautious and leads to more false negatives. Overall, FNDR variable selection results in the most total erroneous outcomes. Lastly, for all signal strengths overall the HC output measures are in between those of CB and FNDR thresholding.

This simulation study is repeated with other effect sparsity settings. The results for those simulations can be found in Appendix A.1. Specifically, we present the results for $\epsilon \in 0.05, 0.1, 0.3, 0.5$. The resulting plots show the same patterns as in Figure 2.

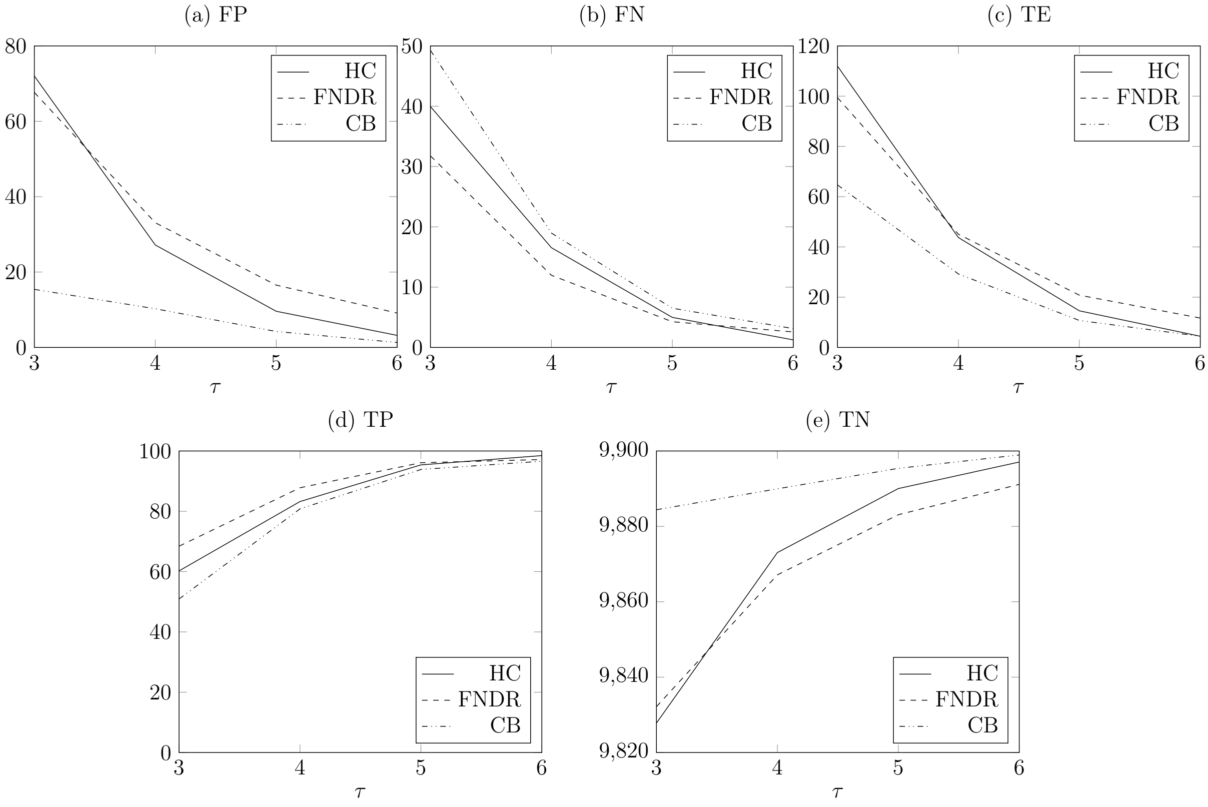

Figure 2: Simulation of Output Measures for HC, CB and FNDR Thresholding in a Rare-Weak Feature Setting with $\epsilon = 0.01$

Table 2: Standard Errors of the Output Measures in Figure 2 for $\epsilon = 0.01$

Method	τ	SE(FP)	SE(FN)	SE(TP)	SE(TN)	SE(TE)
HC	3	69.28	9.75	11.82	70.68	63.16
	4	18.42	5.00	9.94	21.74	16.15
	5	7.73	2.50	10.09	13.23	6.95
	6	4.26	1.05	9.74	10.66	3.99
CB	3	9.05	8.19	9.55	14.58	8.96
	4	6.12	5.10	9.77	12.14	5.96
	5	4.30	4.36	10.34	11.15	4.67
	6	2.21	4.68	10.37	9.90	4.57
FNDR	3	45.05	8.23	11.86	47.80	39.80
	4	27.57	5.55	10.23	29.73	24.51
	5	21.91	4.78	10.63	24.30	20.42
		17.24	4.86	10.48	19.42	16.65

In case of a correlation structure between the features, the plots for $\epsilon = 0.01$ and $\rho = 0.2$ look like those in Figure 3. The plots for false negatives and true positives show similar patterns as in the case of no correlation. For these plots, HC is a compromise between FNDR and CB thresholding. Furthermore, iHC produces more false negatives and less true positives than HC. In contrast to my expectations, these results do not show a significant rise in power when implementing iHC instead of HC. Both the plot which displays the false positives for each method, as well as the plot which displays the true negatives illustrate patterns significantly different from the plots when there is no correlation structure. The resulting plots indicate that both CB and FNDR perform better than HC and iHC in case of correlation between features. Moreover, in both plots HC outperforms iHC which again rejects the hypothesis of improved performance when applying iHC. This simulation study is repeated with other correlation settings. The results for those simulations can be found in Appendix A.2. The resulting plots show the same patterns as in Figure 3.

The results of these simulations are not in line with the claims from Hall and Jin (2010). Namely, in contrast to my results they find a significant increase in power when applying iHC instead of HC thresholding. This difference is striking as the exact steps presented in the paper of Hall and Jin (2010) are implemented. Additionally, the combinations of τ, ϵ and ρ cause the simulated RW model on which the methods are applied, to be above the detection boundary according to the theory of Hall and Jin (2010).

I see several possibilities for the discrepancy in the results of this research paper and the one from Hall and Jin (2010). First of all, it could be the case that the combination of the input variables for my simulated RW model do not put the RW model in the detectable region. However, I am following the theory of Hall and Jin (2010) and the proof included in their paper seems to be correct. Therefore, I deem this reason to be improbable. Secondly, it could be that Hall and Jin (2010) omit the explanation of one or several assumptions taken, which are not generally known. This would make their research fail to obey the rule of replicability. Lastly, the noise vector Z is generated as a Gaussian vector by Hall and Jin (2010). How exactly this is done remains unclear. Consequently, I generated the noise vector using the function *rmvnorm* from the R-package *mvtnorm*. This generates a vector which, when added to the signal vector, contains signal features which are hard to distinguish from the features which do not come from the alternative distribution. It could be the case that Hall and Jin (2010) obtain their noise vector in another way. However, this remains unclear from their explanation and thus again would make their paper incapable of being replicated.

Replicability is an important characteristic for published research papers. Namely, if a research is replicable, other researchers can reproduce, test and further develop the models and theories posed by the authors. If all the theory posed by and Hall and Jin (2010) is correct, this must mean that their paper is not replicable. This should then be rectified by the authors such that further research is able to use and build further on their results. Therefore, the results of my simulations of iHC shed new light on the technique and lay a foundation for further research. It is of importance for future research to further investigate the underlying causes for the difference in conclusions and clearly state them in the literature.

4.3 Cancer Gene Expression Data

The number of selected variables and prediction errors for the four cancer data sets are shown in Table 3. Overall, the CB threshold gives the smallest predictor set. This set is roughly half the amount of predicted genes by HC thresholding, except for the prostate cancer data set. While the number of variables selected using HC thresholding is almost double, the prediction error is only slightly increased. This indicates that almost all of the additionally included predictors are false positives, which confirms the previous theoretical results. FNDR, applied to identify true null features, selects the biggest set of features, which is as expected. These results are in line with those of Klaus and Strimmer (2013).

The performance of iHC varies significantly per data set. In case of the prostate and SRBCT

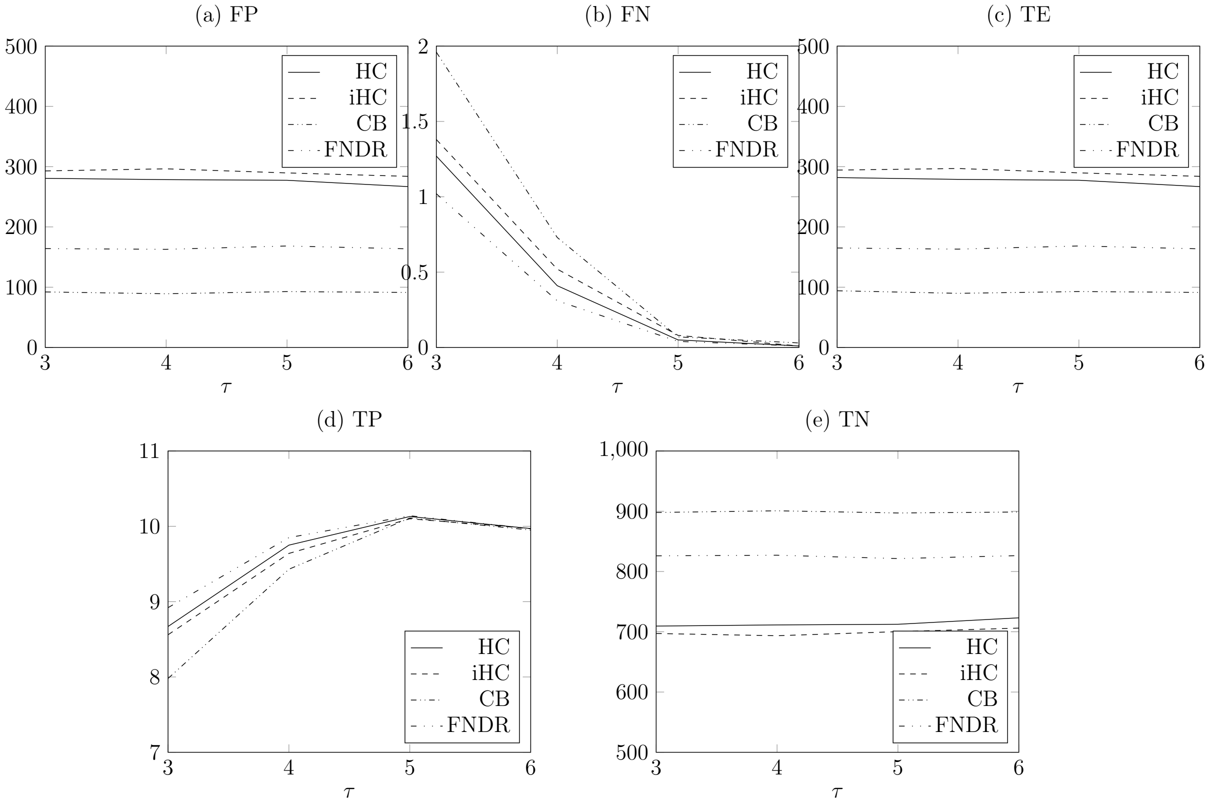

Figure 3: Simulation of Output Measures for HC, CB and FNDR Thresholding in a Rare-Weak Feature Setting with $\epsilon = 0.01$ and $\rho = 0.2$.

data sets, the number of selected variables is in between those of HC and CB thresholding. However, for the other two data sets, iHC eminently selects the most variables while not always significantly lowering the prediction error. Additionally, when the number of selected variables is in between those of the other techniques, the prediction error is not necessarily similar to those of the other methods. This result could be caused by the fact that the empirical correlation matrices are much more complicated than those assumed by the foundation of iHC. However, keeping in mind the theoretical results, it is likely that the paper of Hall and Jin (2010) fails to obey the rule of replicability. This could cause my implementation of iHC to give results different from my expectations. These results again demonstrate that further investigation for this method is necessary before its merit can be validated.

5 Practical Implications

Multiple hypothesis testing in a RW feature setting is common in many fields such as genomics, physics and astronomy. In this section I highlight the possibilities of the discussed techniques in economic sectors as they are not yet widely applied in the economic world. Specifically, I emphasize the possible opportunities in the fields of marketing, finance and health economics.

Table 3: Application of Decision Thresholds to Cancer Gene Expression Data

Method	Prediction Error	Selected variables
$\boldsymbol{Prostate(d = 6033, n = 102, K = 2)}$		
CB	0.0579 (0.0051)	115
HC	0.0564 (0.0051)	116
FNDR	0.0536 (0.0048)	131
iHC	0.0023 (0.0013)	844
$\boldsymbol{Lymphoma(d = 4026, n = 62, K = 3)}$		
CB	0.0113 (0.0031)	178
HC	0.0000 (0.0000)	345
FNDR	0.0057 (0.0021)	392
iHC	0.2175 (0.0087)	217
$\boldsymbol{SRBCT(d = 2308, n = 63, K = 4)}$		
CB	0.0000 (0.0000)	88
HC	0.0007 (0.0007)	174
FNDR	0.0000 (0.0000)	89
iHC	0.0157 (0.0034)	171
$\boldsymbol{Brain(d = 5597, n = 42, K = 5)}$		
CB	0.1582 (0.0131)	78
HC	0.1618 (0.0121)	131
FNDR	0.1768 (0.0145)	102
iHC	0.1575 (0.0124)	653

Note: d: number of possible features,
n: sample size,
K: number of classes in the dependent variable.

In marketing, there is often a search for appropriate variables to classify products, consumers groups or markets. New technologies have enabled marketeers to store enormous amounts of data, in the hope that this data will eventually be useful in their marketing models. This unfocused gathering of data results in a vast quantity of possible explanatory variables. Due to the aimless nature of the obtained data, the structure of the data is likely to meet the characteristics of a RW feature setting. Therefore, it would be of interest to apply the proposed thresholding techniques in this paper to decide which variables to include or not.

Another field which could benefit from the proposed methods is quantitative finance. Namely, with the development of the internet, a huge amount of trading behaviour of each individual agent can be recorded. This can include variables like investment decisions but also consumption patterns or social communications (Z. Wang et al., 2019). Techniques such as HC thresholding could be effective for predicting market trends or analysing market behaviour in such settings.

Lastly, the discussed thresholding techniques could offer benefits for the health economics sector. In recent years, genome-wide association studies (GWAS) have shown that a large part of the genetic basis for most complex traits is built out of small effects of hundreds or even thousands of variants (Euesden et al., 2015). The results from GWASs can be used to create polygenic risk scores (PRSs) for several phenotypes such as obesity. A PRS for an individual is a summation of millions of variants genome-wide, weighted by the strength of their association with a trait of interest (HRS, 2018). Effect sizes are estimated from published GWAS results, and only variants exceeding a certain p-value are included.

PRSs can be used as instrumental variables in many applied studies in the field of health economics. For instance, many studies find a significant negative association between obesity and labor market outcomes (Devaux & Sassi, 2015; Lindeboom et al., 2010). However, due to the reverse causality between obesity and labor market outcomes, instrumental variables are needed. As previous research suggests that the genetic effect on variation in BMI is relatively strong, the PRS would serve as a appropriate instrumental variable. Higher Criticism has, to my knowledge, not yet been applied to the calculation of PRSs. Due to the genomic structure, HC and FDR thresholding seem very appropriate for the calculation of PRSs and their use as instrumental variables. If results would provide new insights to the understanding of the correlation between socioeconomic variables and PRSs, health policy could incorporate this in order to become more effective.

6 Conclusion and Discussion

6.1 Conclusion

This research investigated the effectiveness of several methods for identifying relevant variables in a rare-weak feature setting. Several output measures of these methods were simulated and subsequently the techniques were applied to real-life data sets. The results show that, in the case of independent features, the HC threshold can be seen as a compromise between the KS and CB threshold. Both in the theoretical and empirical setting, HC feature selection leads to output measures in between those of CB and FNDR tresholding. Following from this result, I reject the first hypothesis which states that HC thresholding outperforms both CB and FDR techniques in terms of the prediction error. In addition, when the combination of signal strength and signal sparsity is such that variable identification is possible, the three theoretical thresholds are not significantly different. Therefore, I accept the second hypothesis which states that the CB and HC threshold become indistinguishable when variable identification is possible. Furthermore, the results show that HC variable selection leads to the inclusion of more false positives than the other various thresholding techniques. This leads me to accept the third hypothesis.

These results on the one hand support the studies which claim that HC thresholding is an outstanding technique for feature selection when the assumption of independence is valid. On the other hand, they show that when signal identification is possible, feature selection based on false discovery rates is just as appropriate to implement as HC thresholding. Namely, the theoretical thresholds of the two techniques are insignificantly different. Next to this, the empirical results show that HC thresholding mostly includes extra false positives, which do not raise the predictive power of the model.

When there is a correlation structure between features, the output measures show significantly different patterns from the situation of no correlation. Both CB and FNDR perform better than HC and iHC in terms of Type I errors. Furthermore, the power of iHC thresholding is not significantly better than the one of HC variable selection. This is in contrast with the theory posed by Hall and Jin (2010). Likewise, the empirical results show that iHC does not systematically perform better than HC in terms of prediction error. Consequently, I reject my fourth hypothesis. The difference between my results and the results of Hall and Jin (2010) are striking and likely due to the fact that Hall and Jin (2010) fail to obey to the rule of replicability. The results of my simulations of iHC shed new light on the technique and lay a foundation for further research. In conclusion, it is of importance for future research to investigate the underlying causes for this difference in conclusions before the merit of iHC can be validated.

6.2 Discussion

This research paper contributes to the literature in several ways. First of all, it presents a clear and understandable overview of multiple hypothesis testing techniques and their combination with linear discriminant analysis. Previous papers discussing multiple hypothesis testing methods often only briefly mentioned the combination of LDA and the proposed thresholding techniques but did not clarify how this combination functions. Secondly, both the simulations and empirical results support the part of the literature which suggests that both HC and FDR based thresholding are appropriate methods in a RW feature setting, in case of independence between features. Third, this research investigated iHC, a novel concept which has theoretical advantages over HC. Both the empirical results as the results from simulations in this paper show that more research must be done for this concept to be accepted as an alternative for HC thresholding in case of correlation. The combination of these results establishes a foundation for further research.

Due to time constraints, this paper only discusses CB, FDR, HC and iHC as feature selection techniques. Many other approaches such as the Cramer-Von Mises and Anderson-Darling threshold or machine learning based techniques, which make use of neural networks, exist. Therefore, it is of importance to further compare HC and iHC thresholding to these techniques to get an complete overview of how variable selection techniques can be best applied in a rare-weak feature setting.

Another limitation of this study is that it only applies the techniques to data sets frequently used in genomics, but does not apply it to more economically focused data sets. It would be interesting to see if the presented multiple hypothesis testing techniques would be of value in economic sectors as suggested by Section 5. Therefore, my last suggestion for further research is to apply the techniques to real-world economic data sets.

References

Ahdesmäki, M., Strimmer, K. Et al. (2010). Feature selection in omics prediction problems using cat scores and false nondiscovery rate control. *The Annals of Applied Statistics*, *4*(1), 503–519.

Ahdesmaki, M., Zuber, V., Gibb, S., & Strimmer, K. (2015). Sda: Shrinkage discriminant analysis and cat score variable selection. *R package version*, *1*(7).

Alizadeh, A. A., Eisen, M. B., Davis, R. E., Ma, C., Lossos, I. S., Rosenwald, A., Boldrick, J. C., Sabet, H., Tran, T., Yu, X., Et al. (2000). Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, *403*(6769), 503–511.

Andrews, D. W. (1994). Empirical process methods in econometrics. *Handbook of econometrics*, *4*, 2247–2294.

Benjamini, Y., & Hochberg, Y. (1995). Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, *57*(1), 289–300.

Boole, G. (1847). *The mathematical analysis of logic*. Philosophical Library.

Cantú-Paz, E., Newsam, S., & Kamath, C. (2004). Feature selection in scientific applications, In *Proceedings of the tenth acm sigkdd international conference on knowledge discovery and data mining*.

Cheverud, J. M. (2001). A simple correction for multiple comparisons in interval mapping genome scans. *Heredity*, *87*(1), 52–58.

Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, *1*(3), 131–156.

Devaux, M., & Sassi, F. (2015). The labour market impacts of obesity, smoking, alcohol use and related chronic diseases.

Donoho, D., & Jin, J. (2008). Higher criticism thresholding: Optimal feature selection when useful features are rare and weak. *Proceedings of the National Academy of Sciences*, *105*(39), 14790–14795.

Donoho, D., & Jin, J. (2015). Special invited paper: Higher criticism for large-scale inference, especially for rare and weak effects. *Statistical Science*, 1–25.

Donoho, D., Jin, J. Et al. (2004). Higher criticism for detecting sparse heterogeneous mixtures. *The Annals of Statistics*, *32*(3), 962–994.

Dunn, O. J. (1958). Estimation of the means of dependent variables. *The Annals of Mathematical Statistics*, 1095–1111.

Efron, B. (2005). Local false discovery rates. Division of Biostatistics, Stanford University.

Euesden, J., Lewis, C. M., & O'Reilly, P. F. (2015). Prsice: Polygenic risk score software. *Bioinformatics*, *31*(9), 1466–1468.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of eugenics*, *7*(2), 179–188.

Hall, P., Jin, J. Et al. (2010). Innovated higher criticism for detecting sparse signals in correlated noise. *The Annals of Statistics*, *38*(3), 1686–1732.

Hand, D. J. (2008). Breast cancer diagnosis from proteomic mass spectrometry data: A comparative evaluation. *Statistical applications in genetics and molecular biology*, *7*(2).

James, W., & Stein, C. (1992). Estimation with quadratic loss, In *Breakthroughs in statistics*. Springer.

Khan, J., Wei, J. S., Ringner, M., Saal, L. H., Ladanyi, M., Westermann, F., Berthold, F., Schwab, M., Antonescu, C. R., Peterson, C., Et al. (2001). Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks. *Nature medicine*, *7*(6), 673–679.

Klaus, B., & Strimmer, K. (2013). Signal identification for rare and weak features: Higher criticism or false discovery rates? *Biostatistics*, *14*(1), 129–143.

Li, J., & Ji, L. (2005). Adjusting multiple testing in multilocus analyses using the eigenvalues of a correlation matrix. *Heredity*, *95*(3), 221–227.

Lindeboom, M., Lundborg, P., & van der Klaauw, B. (2010). Assessing the impact of obesity on labor market outcomes. *Economics & Human Biology*, *8*(3), 309–319.

Mihunov, V. V., Lam, N. S., Rohli, R. V., & Zou, L. (2019). Emerging disparities in community resilience to drought hazard in south-central united states. *International Journal of Disaster Risk Reduction*, *41*, 101302.

Monroy-Vilchis, O., Heredia-Bobadilla, R.-L., Zarco-González, M. M., Ávila-Akerberg, V., & Sunny, A. (2019). Genetic diversity and structure of two endangered mole salamander species of the trans-mexican volcanic belt. *Herpetozoa*, *32*, 237.

Opgen-Rhein, R., & Strimmer, K. (2007). From correlation to causation networks: A simple approximate learning algorithm and its application to high-dimensional plant gene expression data. *BMC systems biology*, *1*(1), 37.

Pomeroy, S. L., Tamayo, P., Gaasenbeek, M., Sturla, L. M., Angelo, M., McLaughlin, M. E., Kim, J. Y., Goumnerova, L. C., Black, P. M., Lau, C., Et al. (2002). Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, *415*(6870), 436–442.

Saunders, G. (2014). Family-wise error rate control in quantitative trait loci (qtl) mapping and gene ontology graphs with remarks on family selection.

Singh, D., Febbo, P. G., Ross, K., Jackson, D. G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A. A., D'Amico, A. V., Richie, J. P., Et al. (2002). Gene expression correlates of clinical prostate cancer behavior. *Cancer cell*, *1*(2), 203–209.

Wang, D., Li, Y., Wang, X., Liu, X., Fu, B., Lin, Y., Larsen, L., & Offen, W. (2015). Overview of multiple testing methodology and recent development in clinical trials. *Contemporary clinical trials*, *45*, 13–20.

Wang, Z. Et al. (2019). *The principle of trading economics*. Springer.

Zuber, V., & Strimmer, K. (2009). Gene ranking and biomarker discovery under correlation. *Bioinformatics*, *25*(20), 2700–2707.

A Comparison of Output Measures

The following contains an overview of output measures for several thresholding techniques for various combinations of input variables. The plots in Appendix A.1 show the output measures in case of no correlation and the plots in Appendix A.2 display the results when there is a correlation structure between the possible features. The plots for various other ρ are left out as they displayed exactly the same patterns.

A.1 No Correlation

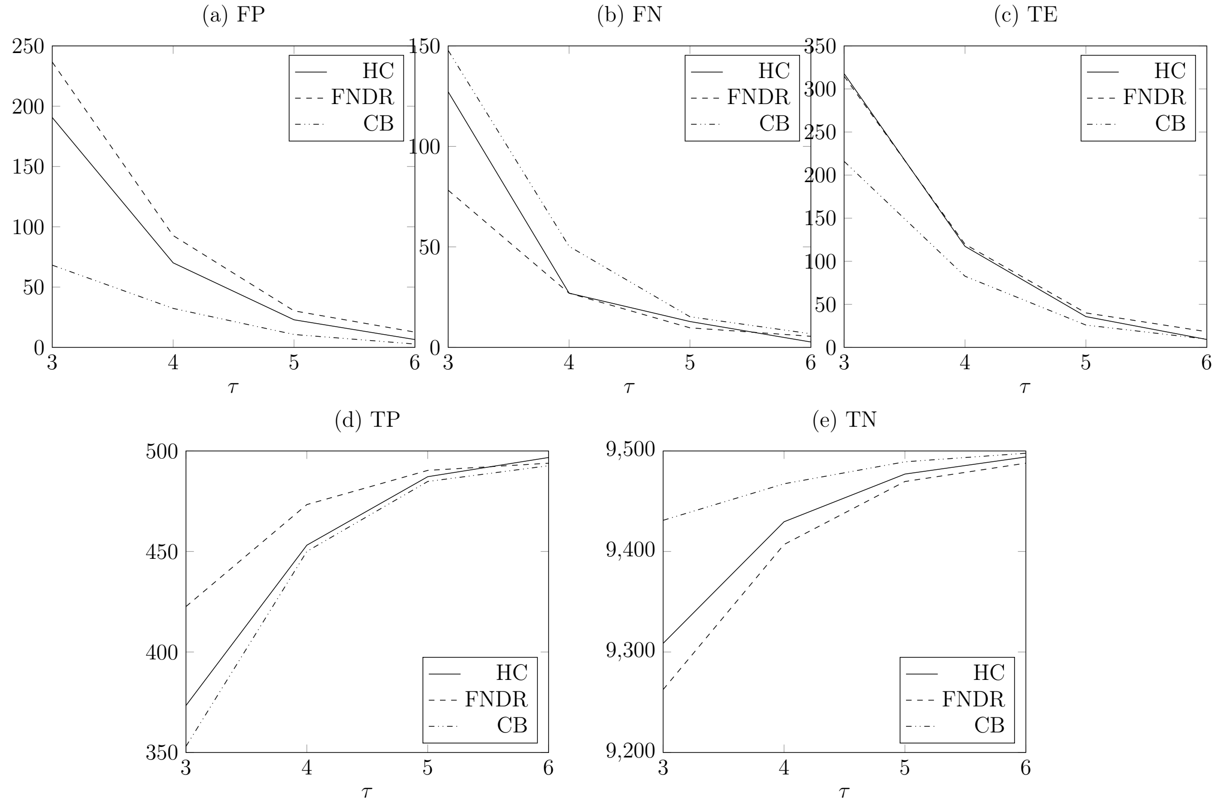

Figure 4: Simulation of Output Measures for HC, CB and FNDR Thresholding in a Rare-Weak Feature Setting with $\epsilon = 0.05$

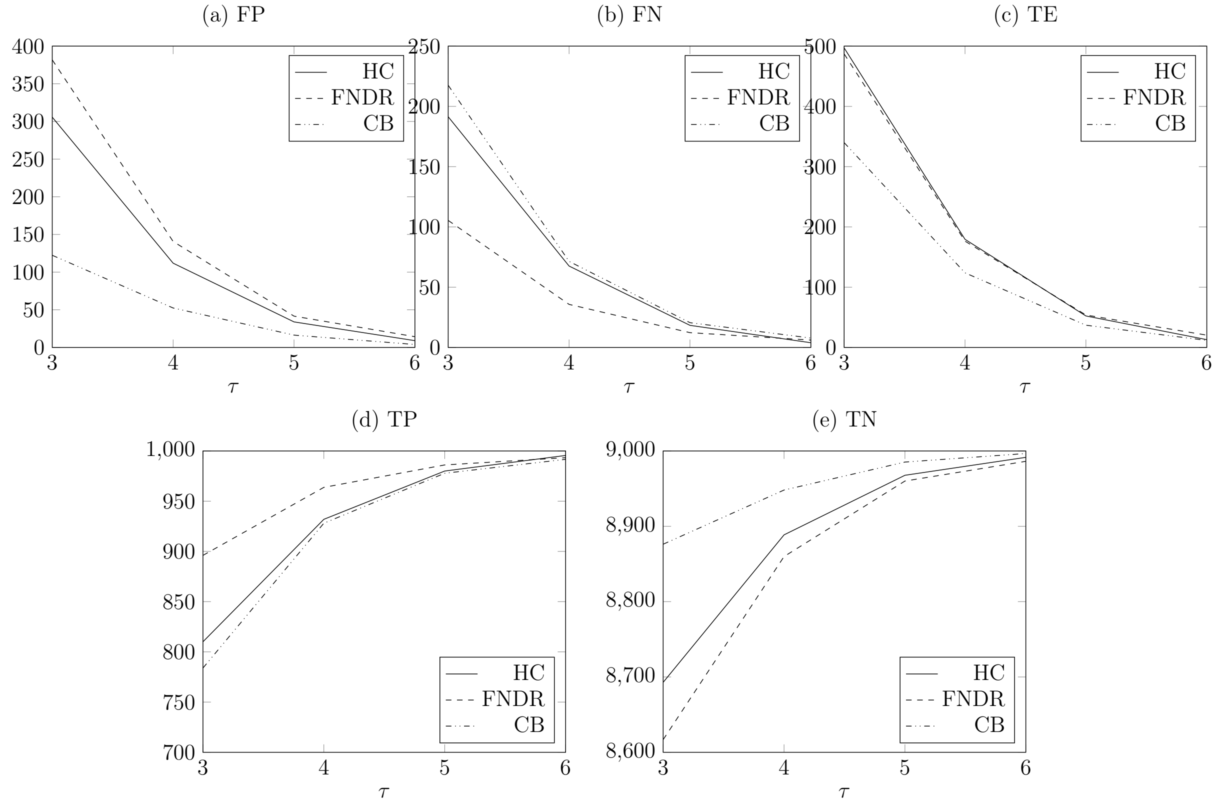

Figure 5: Simulation of Output Measures for HC, CB and FNDR Thresholding in a Rare-Weak Feature Setting with $\epsilon = 0.1$

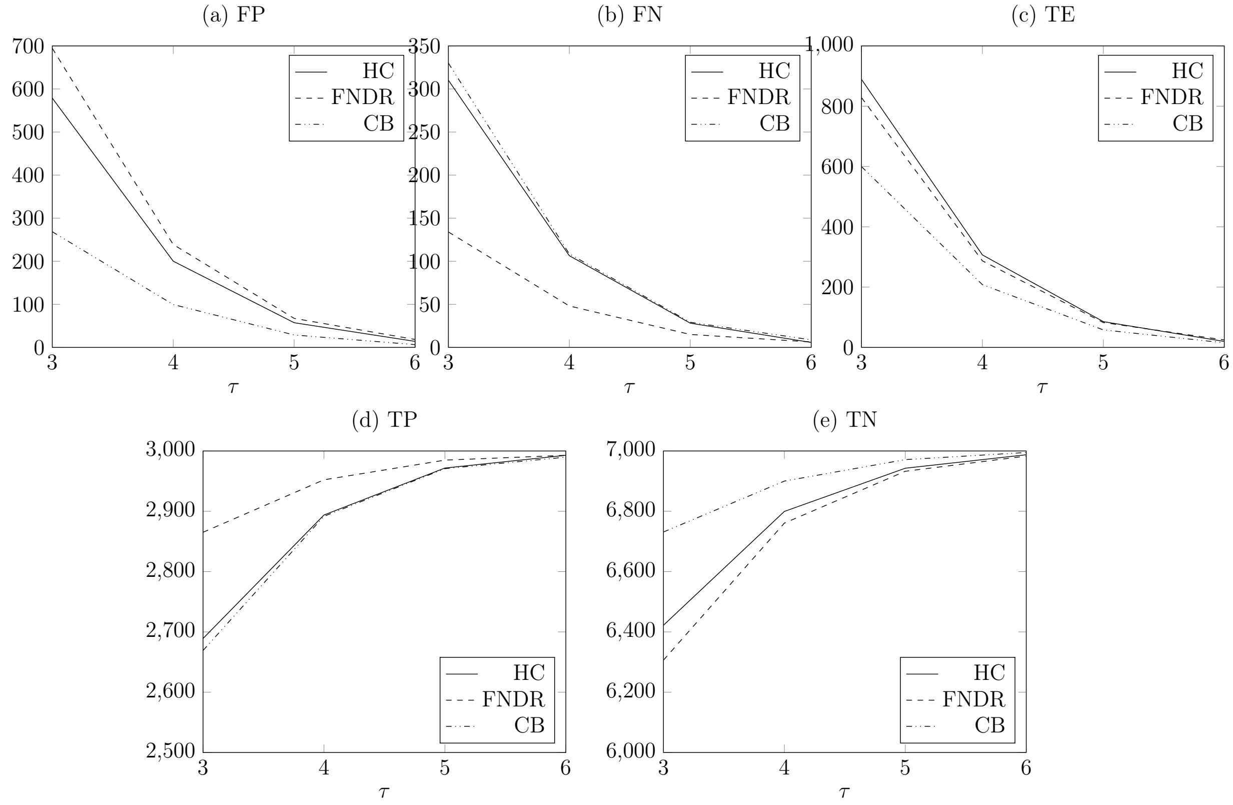

Figure 6: Simulation of Output Measures for HC, CB and FNDR Thresholding in a Rare-Weak Feature Setting with $\epsilon = 0.3$

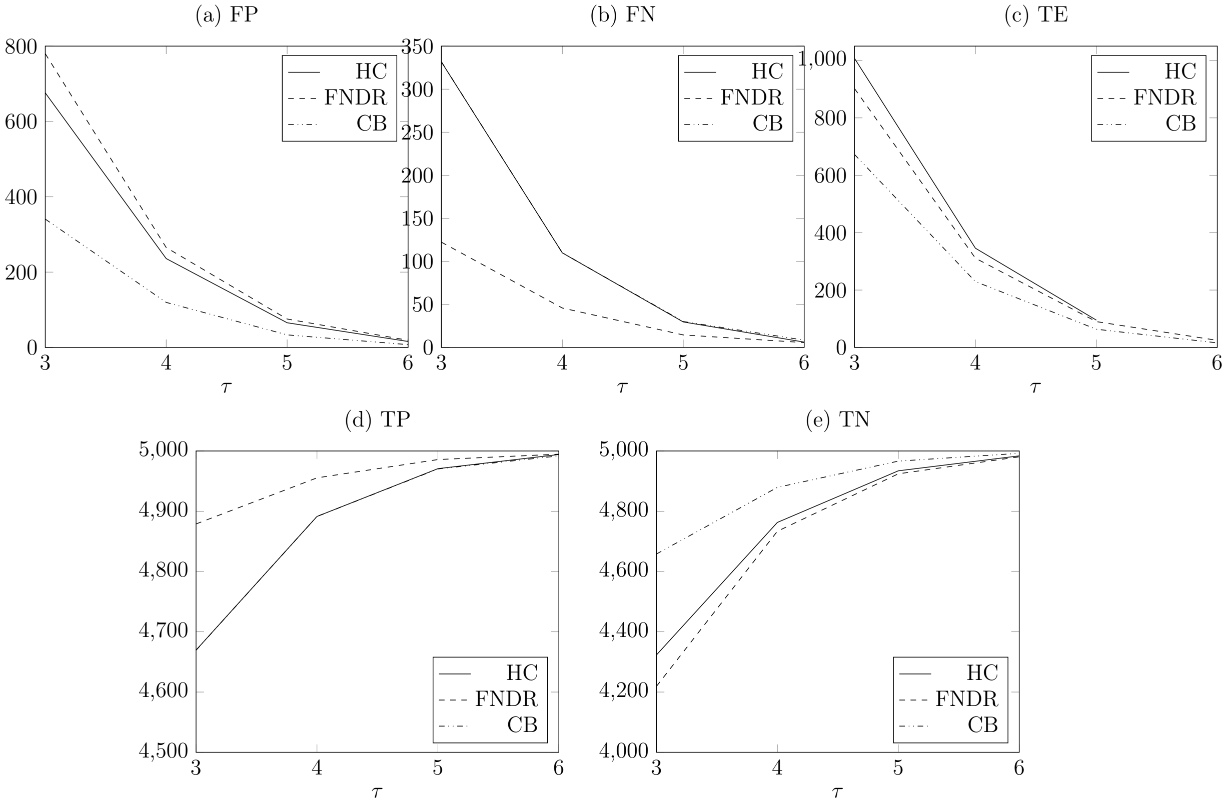

Figure 7: Simulation of Output Measures for HC, CB and FNDR Thresholding in a Rare-Weak Feature Setting with $\epsilon = 0.5$

A.2 With Correlation

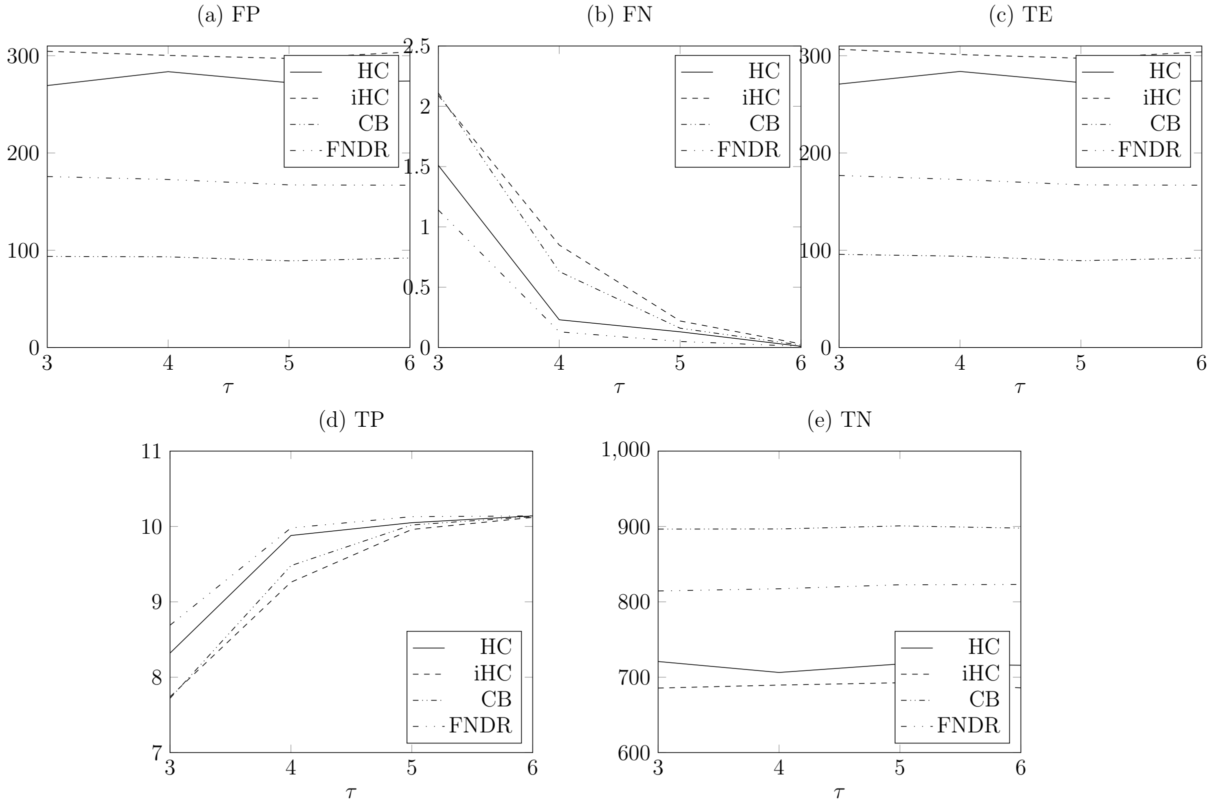

Figure 8: Simulation of Output Measures for HC, CB and FNDR Thresholding in a Rare-Weak Feature Setting with $\epsilon = 0.01$ and $\rho = -0.35$.

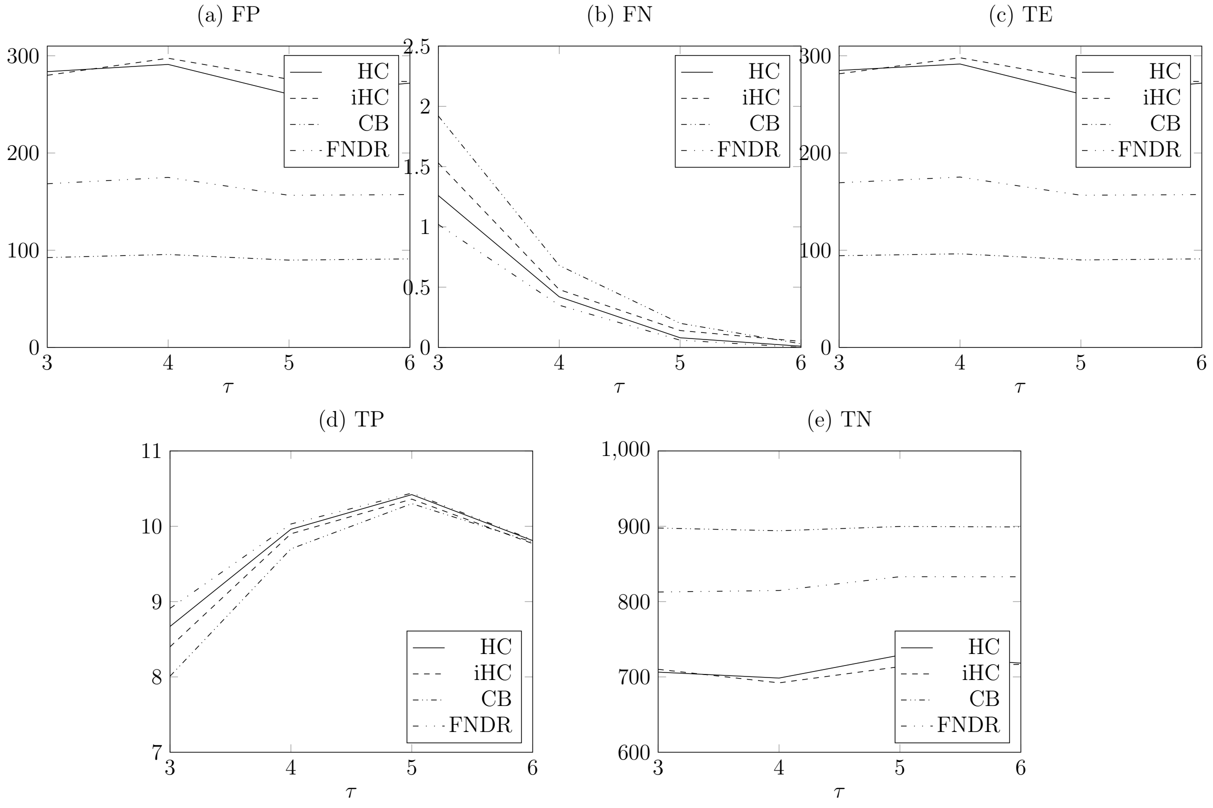

Figure 9: Simulation of Output Measures for HC, CB and FNDR Thresholding in a Rare-Weak Feature Setting with $\epsilon = 0.01$ and $\rho = -0.20$.

B Programming Codes

The simulation studies and empirical evaluation were implemented in R. In total 4 scripts were written to obtain the results. Script 1 contains the functions for obtaining the thresholds in Table 1. Script 2 displays the code for simulating the output measures in case of no correlation. The code for simulating the output measures when there is a correlation structure between features is written in script 3. Script 4 contains the general code for obtaining the results in Table 3.

Listing 1: Code for Obtaining Table 1

```
    basicstyle=\small
]
# simulate 10,000 z-scores and pvalues and calculate zHC
pvalues <- list()
library(distr)
library(fdrtool)
pvector <- vector()
HC <- 0
avrzHC <- 0
df <- data.frame(matrix(ncol = 6, nrow = 0))
x <- c("tau", "eta", "method", "zKS", "zCB", "zHC")
colnames(df) <- x
HCvec <- vector()
alpha0 = 0.1
T=1000
#Thresholds in the RW model
for (tau in c(2,3,4,5,6))
{
  for (eta in c(0,0.001,0.01,0.1,0.5))
  {
    zKS <- tau/2
    zCB <- tau/2 + 1/tau*log((1-eta)/eta)
    HCsquare <- function(x){
      (pnorm(x,tau,1)^2 + pnorm(x,0,1)^2 -2*pnorm(x,tau,1)*pnorm(x,0,1))/
        ((1-eta)*pnorm(x,0,1)+eta*pnorm(x,tau,1)-((1-eta)*pnorm(x,0,1)+eta*pnorm(x,tau,1))^2)
    }
    curve(HCsquare(x), from = -10, to = 10)
    zHC <- optimise(HCsquare, interval = c(-100,100), maximum = TRUE)$maximum
    newrow <- data.frame(tau = tau, eta = eta, zKS = zKS, zCB = zCB, zHC = zHC)
    df <- rbind(df, newrow)
  }
}
```

Listing 2: Code for Simulating Output Measures in Case of no Correlation

```
    # simulate 10,000 z−scores and pvalues and calculate zCB
pvalues <− list()
library(distr)
library(fdrtool)
library(tidyverse)
library(ellipsis)
library(ggplot2)
pvector <− vector()
HC <− 0
avrzHC <− 0
zHC <− vector()
zCB <− vector()
```

```r
zFNDR <- vector()
df <- data.frame(matrix(ncol = 13, nrow = 0))
x <- c("tau", "eta", "method", "FP", "FN", "TP", "TN", "TE", "SEFP", "SEFN", "SETP", "SETN", "SETE")
colnames(df) <- x
methods <- c("CB", "FNDR", "HC")

for (tau in c(3,4,5,6)){
  for (eta in c(0.3))
  {
    Negative <- vector() #1 if an observation is from the standard normal (null) distribution
    FPvecCB <- vector()
    FNvecCB <- vector()
    TNvecCB <- vector()
    TPvecCB <- vector()
    TEvecHC <- vector()
    FPvecHC <- vector()
    FNvecHC <- vector()
    TNvecHC <- vector()
    TPvecHC <- vector()
    TEvecHC <- vector()
    FPvecfndr <- vector()
    FNvecfndr <- vector()
    TNvecfndr <- vector()
    TPvecfndr <- vector()
    TEvecfndr <- vector()
    for (j in 1:1000)
    {
      #The number of samples from the mixture distribution
      N = 10000
      U =runif(N) #Sample N random uniforms U

      #Variable to store the samples from the mixture distribution
      rand.z = rep(NA,N)

      #Sampling from the mixture
      for(i in 1:N){
        if(U[i]< eta){
          rand.z[i] = rnorm(1,tau,1)
          Negative[i] = 0
        }else{
          rand.z[i] = rnorm(1,0,1)
          Negative[i] = 1
        }
      }
      pvector <- list()
      for (i in 1:length(rand.z))
      {
        # calculate p values under the null of a N(0,1) distribution
        p = 1 - pnorm(rand.z[i],0,1)
        pvector <- append(pvector,p) #make a vector of pvalues for a tau and eta combination
      }

      #obtain empirical CB threshold
      tau1 <-  unlist(pvector)
      fdr <- fdrtool(tau1, verbose = FALSE, plot = FALSE, statistic="pvalue")
      lfdr <- fdr$lfdr

      for (method in methods){
        FP <- 0
        TN <- 0
        FN <- 0
        TP <- 0
        if (method == "CB"){
          thresh <- 0.5
          index <- which.min(abs(lfdr - 0.5))
          if (lfdr[index]> 0.5){
            index <- index - 1
          }
          pthresh <- sort(tau1)[index]
          indexz <- match(pthresh, pvector)
        }
        if (method  == "FNDR"){
          thresh <- 0.8 #belong to the null when bigger than 0.8
          index <- which.min(abs(lfdr - 0.8))
          if (lfdr[index]>0.8){
            index <- index -1
          }
          pthresh <- sort(tau1)[index] #gives the pvalue
          indexz <- match(pthresh, pvector)
        }
        if (method == "HC"){
          pthresh <- hc.thresh(tau1, plot = FALSE)
          index <- match(pthresh, pvector)
          indexz <- match(pthresh, pvector)
        }

        if (method == "CB" | method == "FNDR"){
          for (i in 1:length(rand.z)){
            if (lfdr[i]<thresh & Negative[i]==1) {#so we reject and we should not reject
              FP <- FP +1
            }
            if (lfdr[i]<thresh & Negative[i]==0) {#so we reject and we should reject
              TP = TP+1
            }
            if (lfdr[i]>=thresh & Negative[i]==0) {#so we do not reject and we should reject
              FN = FN +1
            }
            if (lfdr[i]>=thresh & Negative[i]==1) {#so we do not reject and we should not reject
              TN = TN +1
            }
          }
        }
        else {
          for (i in 1:length(rand.z)){
            if (abs(rand.z[i])>abs(rand.z[indexz]) & Negative[i]==1) {#so we reject and we should not reject
              FP <- FP +1
            }
            if (abs(rand.z[i])>abs(rand.z[indexz]) & Negative[i]==0) {#so we reject and we should reject
              TP = TP+1
            }
            if (abs(rand.z[i])<=abs(rand.z[indexz]) & Negative[i]==0) {#so we do not reject and we should reject
              FN = FN +1
            }
            if (abs(rand.z[i])<=abs(rand.z[indexz]) & Negative[i]==1) {#so we do not reject and we should not reject
              TN = TN +1
            }
          }
        }
        if (method == "CB"){
          FPvecCB <- append(FPvecCB, FP)
          FNvecCB <- append(FNvecCB, FN)
          TNvecCB <- append(TNvecCB, TN)
          TPvecCB <- append(TPvecCB, TP)
        }
        if (method == "HC"){
          FPvecHC <- append(FPvecHC, FP)
          FNvecHC <- append(FNvecHC, FN)
          TNvecHC <- append(TNvecHC, TN)
          TPvecHC <- append(TPvecHC, TP)
        }
        if (method == "FNDR"){
          FPvecfndr <- append(FPvecfndr, FP)
          FNvecfndr <- append(FNvecfndr, FN)
          TNvecfndr <- append(TNvecfndr, TN)
          TPvecfndr <- append(TPvecfndr, TP)
        }
        TEvecCB <- FPvecCB + FNvecCB
        TEvecHC <- FPvecHC + FNvecHC
        TEvecfndr <- FPvecfndr + FNvecfndr
      }
    }

    for (method in methods){
      if (method == "CB"){
        newrow <- data.frame(tau = tau, eta = eta,method = method, FP = mean(FPvecCB), FN = mean(FNvecCB),
            TP= mean(TPvecCB), TN=mean(TNvecCB), TE=mean(TEvecCB),
                            SEFP = sd(FPvecCB), SEFN = sd(FNvecCB), SETP=sd(TPvecCB), SETN=sd(TNvecCB),
                                SETE=sd(TEvecCB))

      }
      if (method == "HC")
      {
        newrow <- data.frame(tau = tau, eta = eta,method = method, FP = mean(FPvecHC), FN = mean(FNvecHC),
            TP= mean(TPvecHC), TN=mean(TNvecHC), TE=mean(TEvecHC),
                            SEFP = sd(FPvecHC), SEFN = sd(FNvecHC), SETP=sd(TPvecHC), SETN=sd(TNvecHC),
                                SETE=sd(TEvecHC))

      }
      if (method == "FNDR"){
        newrow <- data.frame(tau = tau, eta = eta,method = method, FP = mean(FPvecfndr), FN =
            mean(FNvecfndr), TP= mean(TPvecfndr), TN=mean(TNvecfndr), TE=mean(TEvecfndr),
                            SEFP = sd(FPvecfndr), SEFN = sd(FNvecfndr), SETP=sd(TPvecfndr),
                                SETN=sd(TNvecfndr), SETE=sd(TEvecfndr))
      }

      df<- rbind(df, newrow)
    }
  }
}
pFP<-ggplot(df, aes(x=tau, y=FP, group = method, color = method)) +
  geom_line() +
  geom_point()+
  geom_errorbar(aes(ymin=FP-SEFP, ymax=FP+SEFP), width=.2,
                position=position_dodge(0.05))

pFN<-ggplot(df, aes(x=tau, y=FN, group = method, color = method)) +
  geom_line() +
  geom_point()+
  geom_errorbar(aes(ymin=FN-SEFN, ymax=FN+SEFN), width=.2,
               position=position_dodge(0.05))
pTP<-ggplot(df, aes(x=tau, y=TP, group = method, color = method)) +
  geom_line() +
  geom_point()+
  geom_errorbar(aes(ymin=TP-SETP, ymax=TP+SETP), width=.2,
                position=position_dodge(0.05))

pTN<-ggplot(df, aes(x=tau, y=TN, group = method, color = method)) +
  geom_line() +
  geom_point()+
  geom_errorbar(aes(ymin=TN-SETN, ymax=TN+SETN), width=.2,
                position=position_dodge(0.05))

pTE<-ggplot(df, aes(x=tau, y=TE, group = method, color = method)) +
  geom_line() +
  geom_point()+
  geom_errorbar(aes(ymin=TE-SETE, ymax=TE+SETE), width=.2,
                position=position_dodge(0.05))
library("gridExtra")
grid.arrange(pFP, pFN, pTP, pTN, pTE,
             ncol = 3, nrow = 2)
```

Listing 3: Code for Simulating Output Measures in Case of Correlation between Features

```r
# simulate 10,000 z-scores and pvalues and calculate zCB
pvalues <- list()
library(distr)
library(fdrtool)
library(tidyverse)
library(ellipsis)
library(mvtnorm)
library(ggplot2)
library(wordspace)
pvector <- vector()
HC <- 0
df <- data.frame(matrix(ncol = 14, nrow = 0))
x <- c("tau", "eta", "method", "rho", "FP", "FN", "TP", "TN", "TE", "SEFP", "SEFN", "SETP", "SETN", "SETE")
colnames(df) <- x
methods <- c("CB", "FNDR", "HC", "iHC")

for (tau in c(3,4,5,6)){
  for (eta in c(0.01))
  {
    for (rho in c(-0.35, -0.30, -0.25, -0.20, -0.15, -0.10, -0.05, 0, 0.05,0.10,0.15,0.20,0.25, 0.30,0.35)){
      Negative <- vector() #1 if an observation is from the standard normal (null) distribution
      FPvecCB <- vector()
      FNvecCB <- vector()
      TNvecCB <- vector()
      TPvecCB <- vector()
      TEvecCB <- vector()
      FPvecHC <- vector()
      FNvecHC <- vector()
      TNvecHC <- vector()
      TPvecHC <- vector()
      TEvecHC <- vector()
      FPveciHC <- vector()
      FNveciHC <- vector()
      TNveciHC <- vector()
      TPveciHC <- vector()
      TEveciHC <- vector()
      FPvecfndr <- vector()
      FNvecfndr <- vector()
      TNvecfndr <- vector()
      TPvecfndr <- vector()
      TEvecfndr <- vector()
      for (j in 1:500)
      {
        #The number of samples from the mixture distribution
        n = 1000
        U =runif(n) #Sample N random uniforms U

        #Variable to store the samples from the mixture distribution
        rand.z = rep(NA,n)

        #Sampling from the mixture
        for(i in 1:n){
          if(U[i]< eta){
            rand.z[i] = rnorm(1,tau,1)
            Negative[i] = 0
          }else{
            rand.z[i] = rnorm(1,0,1)
            Negative[i] = 1
          }
        }

        #simulate the correlation matrix
        cormatrix <- matrix( rep( 0, len=n*n), nrow = n)
        for (k in 1:n)
        {
          for (j in 1:n)
          {
            if ((j-k) == 0){
```

```
      cormatrix[j,k] <- 1/(2*pi)* 2*pi
    }
    else if ((j-k) == 1){
      cormatrix[j,k] <- 1/(2*pi)* 2*pi*rho
    }
    else if ((j-k) == 2){
      cormatrix[j,k] <- 1/(2*pi)* 0
    }
    else if ((j-k) == -1){
      cormatrix[j,k] <- 1/(2*pi)* 2*pi*rho
    }
    else {
      cormatrix[j,k] <- 0
    }
  }
}

#generate a Gaussian vector Z~N(0, corrmatrix)

Z <- rmvnorm(1, mean = rep(0, nrow(cormatrix)), sigma = cormatrix, method = ’chol’)

#add the generated noise to the generated signal vector
X <- rand.z + Z
transX <- t(X)

pvector <- list()
for (i in 1:length(rand.z))
{
  # calculate p values under the null of a N(0,1) distribution
  p = 1 - pnorm(X[i],0,1)
  pvector <- append(pvector,p) #make a vector of pvalues for a tau and eta combination
}
U <- chol(cormatrix)
b <- log(n)
Utilde <- U
for (k in 1:n){
  for (j in 1:n){
    if (k - b +1 <= j && j <= k){
      Utilde[k,j] <- Utilde[k,j]
    }
    else {
      Utilde[k,j] <- 0
    }
  }
}
Ubar <- normalize.cols(Utilde)
V <- t(Ubar)%*%U
VX <- X%*%V #check if this is a vector || yes this is a vector!

pvectoriHC <- vector()
for (i in 1:length(X))
{
  # calculate p values under the null of a N(0,1) distribution
  p <- 1 - pnorm(VX[i],0,1)
  pvectoriHC <- append(pvectoriHC, p) #make a vector of pvalues for a tau and eta combination
}


#obtain empirical CB threshold
tau1 <-  unlist(pvector)
fdr <- fdrtool(tau1, verbose = FALSE, plot = FALSE, statistic="pvalue")
lfdr <- fdr$lfdr

for (method in methods){
  FP <- 0
  TN <- 0
  FN <- 0
  TP <- 0
  if (method == "CB"){
```

```
              thresh <- 0.5
              index <- which.min(abs(lfdr - 0.5))
              if (lfdr[index]> 0.5){
                index <- index - 1
              }
              pthresh <- sort(tau1)[index]
              indexz <- match(pthresh, pvector)
            }
            if (method  == "FNDR"){
              thresh <- 0.8 #belong to the null when bigger than 0.8
              index <- which.min(abs(lfdr - 0.8))
              if (lfdr[index]>0.8){
                index <- index -1
              }
              pthresh <- sort(tau1)[index] #gives the pvalue
              indexz <- match(pthresh, pvector)
            }
            if (method == "HC"){
              pthresh <- hc.thresh(tau1, plot = FALSE)
              index <- match(pthresh, pvector)
              indexz <- match(pthresh, pvector)
            }
            if (method == "iHC"){
              pthresh <- hc.thresh(pvectoriHC, plot = FALSE)
              indexz <- match(pthresh, pvectoriHC)
            }
            if (method == "CB" | method == "FNDR"){
              for (i in 1:length(rand.z)){
                if (lfdr[i]<thresh & Negative[i]==1) {#so we reject and we should not reject
                  FP <- FP +1
                }
                if (lfdr[i]<thresh & Negative[i]==0) {#so we reject and we should reject
                  TP = TP+1
                }
                if (lfdr[i]>=thresh & Negative[i]==0) {#so we do not reject and we should reject
                  FN = FN +1
                }
                if (lfdr[i]>=thresh & Negative[i]==1) {#so we do not reject and we should not reject
                  TN = TN +1
                }
              }
            }
            else if (method == "HC"){
              for (i in 1:length(X)){
                if (abs(X[i])>abs(X[indexz]) & Negative[i]==1) {#so we reject and we should not reject
                  FP <- FP +1
                }
                if (abs(X[i])>abs(X[indexz]) & Negative[i]==0) {#so we reject and we should reject
                  TP = TP+1
                }
                if (abs(X[i])<=abs(X[indexz]) & Negative[i]==0) {#so we do not reject and we should reject
                  FN = FN +1
                }
                if (abs(X[i])<=abs(X[indexz]) & Negative[i]==1) {#so we do not reject and we should not reject
                  TN = TN +1
                }
              }
            }
            else if (method == "iHC"){
              for (i in 1:length(rand.z)){
                if (abs(VX[i])>abs(VX[indexz]) & Negative[i]==1) {#so we reject and we should not reject
                  FP <- FP +1
                }
                if (abs(VX[i])>abs(VX[indexz]) & Negative[i]==0) {#so we reject and we should reject
                  TP = TP+1
                }
                if (abs(VX[i])<=abs(VX[indexz]) & Negative[i]==0) {#so we do not reject and we should reject
                  FN = FN +1
                }
                if (abs(VX[i])<=abs(VX[indexz]) & Negative[i]==1) {#so we do not reject and we should not reject
                  TN = TN +1
                }
              }
            }
            if (method == "CB"){
              FPvecCB <- append(FPvecCB, FP)
              FNvecCB <- append(FNvecCB, FN)
              TNvecCB <- append(TNvecCB, TN)
              TPvecCB <- append(TPvecCB, TP)
            }
            if (method == "HC"){
              FPvecHC <- append(FPvecHC, FP)
              FNvecHC <- append(FNvecHC, FN)
              TNvecHC <- append(TNvecHC, TN)
              TPvecHC <- append(TPvecHC, TP)
            }
            if (method == "FNDR"){
              FPvecfndr <- append(FPvecfndr, FP)
              FNvecfndr <- append(FNvecfndr, FN)
              TNvecfndr <- append(TNvecfndr, TN)
              TPvecfndr <- append(TPvecfndr, TP)
            }
            if (method == "iHC"){
              FPveciHC <- append(FPveciHC, FP)
              FNveciHC <- append(FNveciHC, FN)
              TNveciHC <- append(TNveciHC, TN)
              TPveciHC <- append(TPveciHC, TP)
            }
            TEvecCB <- FPvecCB + FNvecCB
            TEvecHC <- FPvecHC + FNvecHC
            TEvecfndr <- FPvecfndr + FNvecfndr
            TEveciHC <- FPveciHC + FNveciHC
          }
        }

        for (method in methods){
          if (method == "CB"){
            newrow <- data.frame(tau = tau, eta = eta,method = method, rho = rho, FP = mean(FPvecCB), FN = mean(FNvecCB), TP=
                 mean(TPvecCB), TN=mean(TNvecCB), TE=mean(TEvecCB),
                                 SEFP = sd(FPvecCB), SEFN = sd(FNvecCB), SETP=sd(TPvecCB), SETN=sd(TNvecCB), SETE=sd(TEvecCB))

          }
          if (method == "HC")
          {
            newrow <- data.frame(tau = tau, eta = eta,method = method, rho = rho, FP = mean(FPvecHC), FN = mean(FNvecHC), TP=
                 mean(TPvecHC), TN=mean(TNvecHC), TE=mean(TEvecHC),
                                 SEFP = sd(FPvecHC), SEFN = sd(FNvecHC), SETP=sd(TPvecHC), SETN=sd(TNvecHC), SETE=sd(TEvecHC))

          }
          if (method == "FNDR"){
            newrow <- data.frame(tau = tau, eta = eta,method = method, rho = rho, FP = mean(FPvecfndr), FN = mean(FNvecfndr),
                 TP= mean(TPvecfndr), TN=mean(TNvecfndr), TE=mean(TEvecfndr),
                                 SEFP = sd(FPvecfndr), SEFN = sd(FNvecfndr), SETP=sd(TPvecfndr), SETN=sd(TNvecfndr),
                                      SETE=sd(TEvecfndr))
          }
          if (method == "iHC"){
            newrow <- data.frame(tau = tau, eta = eta,method = method, rho = rho, FP = mean(FPveciHC), FN = mean(FNveciHC), TP=
                 mean(TPveciHC), TN=mean(TNveciHC), TE=mean(TEveciHC),
                                 SEFP = sd(FPveciHC), SEFN = sd(FNveciHC), SETP=sd(TPveciHC), SETN=sd(TNveciHC),
                                      SETE=sd(TEveciHC))
          }
          df<- rbind(df, newrow)
        }
      }
    }
  }
```

Listing 4: Code for Obtaining the Results in Table 3

```r
# Code for table cancer genes data sets
library(sda)
library(fdrtool)
SRBCT.X = SRBCT$X[1:63,]
SRBCT.Y = SRBCT$Y[1:63]

#calculate correlation matrices and their cholesky decompositions
cormatrixbrain = cor(brain.x)
Ubrain = chol(cormatrixbrain, pivot = TRUE) #because the matrix can be semi positive definite
cormatrixlymphoma = cor(lymphoma$x)
Ulymphoma = chol(cormatrixlymphoma, pivot = TRUE)
cormatrixprostate = cor(prostate$x)
Uprostate = chol(cormatrixprostate, pivot = TRUE)
cormatrixSRBCT = cor(SRBCT.X)
USRBCT = chol(cormatrixSRBCT, pivot = TRUE)

#calculate VX for every data set
bbrain <- log(dim(brain.x)[2])
blymphoma <- log(dim(lymphoma$x)[2])
bprostate <- log(dim(prostate$x)[2])
bSRBCT <- log(dim(SRBCT.X)[2])

Utildebrain <- Ubrain
for (k in 1:dim(brain.x)[2]){
  for (j in 1:dim(brain.x)[2]){
    if (k - bbrain +1 <= j && j <= k){
      Utildebrain[k,j] <- Utildebrain[k,j]
    }
    else {
      Utildebrain[k,j] <- 0
    }
  }
}
Ubarbrain <- normalize.cols(Utildebrain)
Vbrain <- t(Ubarbrain)%*%Ubrain
VXbrain <- brain.x %*%Vbrain #check if this is a matrix || yes this is a matrix with the same
#dimensions as brain.x!

#apply LDA to the produced matrix and check the number of included variables when HC is applied
rabrain <-sda.ranking(VXbrain, brain.y, diagonal = FALSE, fdr = TRUE, ranking.score = "avg", lambda.freqs = 0)
numvarsbrain = which.max( rabrain[, "HC"])

Utildeprostate <- Uprostate
for (k in 1:dim(prostate$x)[2]){
  for (j in 1:dim(prostate$x)[2]){
    if (k - bprostate +1 <= j && j <= k){
      Utildeprostate[k,j] <- Utildeprostate[k,j]
    }
    else {
      Utildeprostate[k,j] <- 0
    }
  }
}
Ubarprostate <- normalize.cols(Utildeprostate)
Vprostate <- t(Ubarprostate)%*%Uprostate
VXprostate <- prostate$x %*%Vprostate

raprostate <-sda.ranking(VXprostate, prostate$y, diagonal = FALSE, fdr = TRUE, ranking.score = "avg", lambda.freqs = 0)
numvarsprostate = which.max( raprostate[, "HC"])


Utildelymphoma <- Ulymphoma
for (k in 1:dim(lymphoma$x)[2]){
  for (j in 1:dim(lymphoma$x)[2]){
    if (k - blymphoma +1 <= j && j <= k){
      Utildelymphoma[k,j] <- Utildelymphoma[k,j]
    }
    else {
      Utildelymphoma[k,j] <- 0
    }
  }
}
Ubarlymphoma <- normalize.cols(Utildelymphoma)
Vlymphoma <- t(Ubarlymphoma)%*%Ulymphoma
VXlymphoma <- lymphoma$x%*%Vlymphoma

ralymphoma <-sda.ranking(VXlymphoma, lymphoma$y, diagonal = FALSE, fdr = TRUE, ranking.score = "avg", lambda.freqs = 0)
numvarslymphoma = which.max( ralymphoma[, "HC"])

UtildeSRBCT <- USRBCT
for (k in 1:dim(SRBCT.X)[2]){
  for (j in 1:dim(SRBCT.X)[2]){
    if (k - bSRBCT +1 <= j && j <= k){
      UtildeSRBCT[k,j] <- UtildeSRBCT[k,j]
    }
    else {
      UtildeSRBCT[k,j] <- 0
    }
  }
}
UbarSRBCT <- normalize.cols(UtildeSRBCT)
VSRBCT <- t(UbarSRBCT)%*%USRBCT
VXSRBCT <- SRBCT.X%*%VSRBCT

### Copyright 2012 Bernd Klaus.
#' Setup prediction function: estimate the accuracy of a predictor with a fixed number of predictors (note
#' this takes into account the uncertainty in estimating the variable ordering).
predfun = function(Xtrain, Ytrain, Xtest, Ytest, numVars, diagonal=FALSE,
                    ranking.score="avg")
{
  # estimate ranking and determine the best numVars variables
  ra = sda.ranking(Xtrain, Ytrain, verbose=FALSE, diagonal=diagonal,
                   fdr=TRUE, ranking.score=ranking.score, lambda.freqs = 0)
  numVars = which.max( ra[, "HC"] )
  #numVars = sum( ra[, "lfdr"]< 0.80)
 #numVars = sum( ra[, "lfdr"]< 0.50)
  selVars = ra[,"idx"][1:numVars]

  # fit and predict
  sda.out = sda(Xtrain[, selVars, drop=FALSE], Ytrain, diagonal=diagonal,
                verbose=FALSE)
  ynew = predict(sda.out, Xtest[, selVars, drop=FALSE], verbose=FALSE)$class

  # compute accuracy
  acc = sum(ynew != Ytest)/length(Ytest)
  return(acc)
}

#' Our setup for crossvalidation:
K = 10 # number of folds
B = 20 # number of repetitions


#'  Crossvalidation estimate of accuracy for
#'  LDA using the top 100 features ranked by CAT scores
#'  (combined across groups using "entropy" for overall ranking):
set.seed(12345)
cv.lda100 = crossval(predfun, Xtrain, Ytrain, K=K, B=B, numVars=numVars,
                     diagonal=FALSE, verbose=FALSE)
```