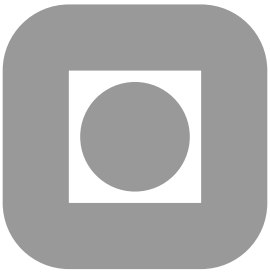
NORGES TEKNISK-NATURVITENSKAPELIGE
UNIVERSITET

Estimating the Proportion of True Null Hypotheses, with Application to DNA Microarray Data

by

Egil Ferkingstad, Mette Langaas and Bo Lindqvist

PREPRINT
STATISTICS NO. 4/2003



NORWEGIAN UNIVERSITY OF SCIENCE AND
TECHNOLOGY
TRONDHEIM, NORWAY

This report has URL
http://www.math.ntnu.no/preprint/statistics/2003/S4-2003.ps
Mette Langaas has homepage: http://www.math.ntnu.no/~mettela
E-mail: mettela@math.ntnu.no
Address: Department of Mathematical Sciences, Norwegian University of Science and Technology, N-7034 Trondheim, Norway.

Estimating the Proportion of True Null Hypotheses, with Application to DNA Microarray Data

Egil Ferkingstad, Mette Langaas and Bo Lindqvist
Department of Mathematical Sciences
Norwegian University of Science and Technology

July 18, 2003

Summary

The problem of estimating the proportion, π_0, of true null hypotheses is important in cases where a large number of hypothesis tests are performed. In addition to being an interesting quantity in itself, the estimate of π_0 can be used as an input to methods for assessing or controlling error rates such as the family wise error rate and the false discovery rate (Benjamini and Hochberg 1995). On the basis of the observed p-values corresponding to the hypothesis tests, several estimators of π_0 are described. Schweder and Spjøtvoll's (1982) nonparametric estimator of π_0 is studied, and different estimators based on estimating the p-value density are developed. Original contributions in this work include a new estimator of π_0 based on Grenander's (1956) nonparametric maximum likelihood density estimator, a novel application of convex decreasing density estimation to the problem of estimating π_0, and the use of kernel density estimation with a choice of smoothing parameter especially tailored to estimate π_0. The estimators are derived under the assumption of independent p-values, and evaluated on simulated data with different degree of dependence. A discussion of the issue of modelling dependencies, with special emphasis on DNA microarray data analysis is presented. Finally, the estimators are applied to real data from DNA microarray experiments.

Contents

1 Introduction

The aim of this work is to develop and present estimators for assessing the proportion of true null hypotheses, π_0, in a multiple hypothesis setup. We apply our research to DNA microarray data, where the goal is to estimate the proportion of genes truly (not) differentially expressed.

Placing the Problem into DNA Microarrays

All of the cells in a living organism have basically the same genetic material. So, the main difference between, say, a liver cell and a brain cell is which genes are *expressed*, i.e. code for proteins. The cell's function is mainly determined by the proteins which it produces. To explain the basics of DNA microarrays, let us consider the *central dogma* of genetics. This can be represented by the following (simplified) diagram:

$$\mathrm{DNA} \rightarrow \mathrm{mRNA} \rightarrow \mathrm{protein},$$

where mRNA is short for "messenger RNA". With DNA microarrays, rather than measuring protein content directly, one uses the middle product, mRNA. Part of the reason for this is that proteins are notoriously difficult to measure on a large scale, because of their extremely intricate molecular structure. Therefore, today, using mRNA is far more tractable. However, it should be noted that using mRNA or proteins directly is not actually equivalent, and in the future protein arrays might be a more desired solution.

Using DNA microarrays[1], we can monitor the expression levels of several thousand genes simultaneously. It is possible to look at analysing data from DNA microarrays in a modular fashion. This is an intuitively appealing way of looking at DNA microarray experiments when focusing on one specific question, but in general one could benefit from a more integrated and coherent view.

In Figure 1 the investigation of a biological question is presented as a series of biological and statistical tasks.

The investigation starts with posing biological questions or presenting biological hypotheses. The biologist will together with the statistician then use statistical design of experiments to plan how microarray experiments should be performed in order to answer the biological questions (e.g. which tissue samples to be applied to the microarray slide in each two-colour DNA microarray experiment, which factors to be varied, number of replications needed). The microarray experiments are then performed by the biologist. The results from microarray experiments are images where the intensity of the spots in an image is related to the expression of the genes in the corresponding tissue sample (the abundance of different species of mRNA). Image analysis tools are used to acquire these intensity measurements. For an review of DNA microarray technology, see e.g. Schulze and Downward (2001).

It is important to assess the quality of data obtained – both on a global and local scale. Global assessment can to a certain degree be performed with the aid of various plotting techniques. If the overall quality of the array is satisfactory, the quality of individual spots are assessed based on physical characteristics of the spot or on agreement with spots from the same gene (or clone). Spots with low quality can be removed (filtered) or kept in the data set.

There are inherently several sources of systematic and random errors present in a microarray experiment. To reach valid inferential conclusions these must be taken properly into account. Currently,

[1] In this presentation, 'DNA microarray" is used as a collective term for different microarray technologies such as two-color spotted cDNA microarrays and oligonucleotide arrays.

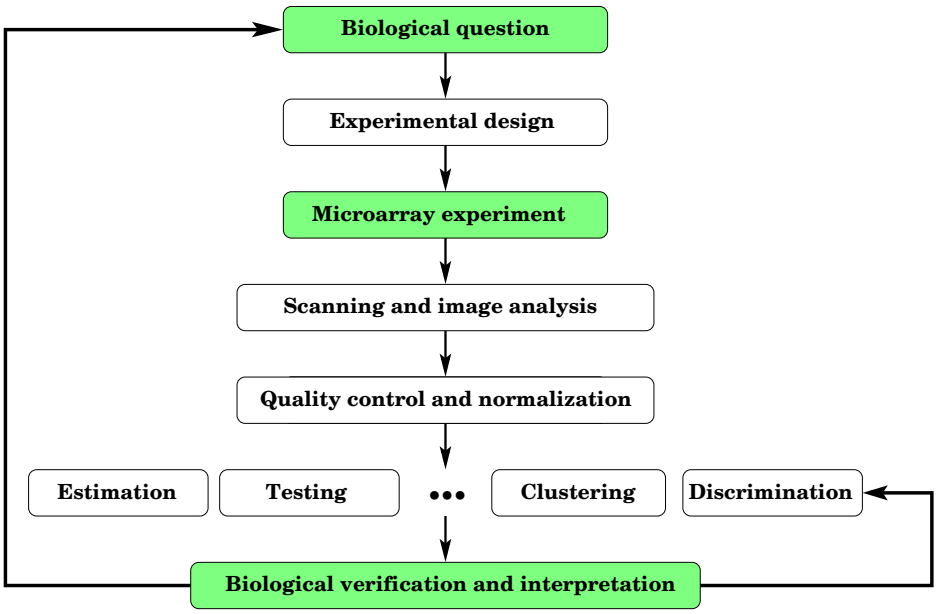


Figure 1: A modular view of investigation of a biological question using microarray technology. The statistical issues are depictured with transparent backgrounds and the biological issues with grey backgrounds. The figure is inspired by a talk by Professor T. P. Speed.

systematic errors are removed as a separate step (called normalization). Statistical tools are then used to find differentially expressed genes, class membership, groupings, rules, etc., which are presented to the biologist for interpretation and verification. These results will then give rise to new and improved biological questions which can be investigated in new microarray experiments or using other techniques.

Thus, the analysis of DNA microarray data involves many challenging statistical problems. A survey of statistical issues and their possible solutions is given by Smyth et al. (2003).

Point of Departure

The inspiration behind developing estimators for the proportion of true null hypotheses has been DNA microarray data. The starting point of this work is after the necessary pre-processing of the DNA microarray data, and we assume that m hypothesis tests are defined, and that a valid p-value is calculated for each test. Our work can also be applied to other situation, besides DNA microarrays, where a large number of hypotheses is tested, e.g. functional magnetic resonance imaging (fMRI) (Turkheimer et al. 2001) and source detection in astrophysics (Miller et al. 2001).

Outline of Presentation

A brief outline of the present work follows. Section 2 contains a presentation of the multiple hypothesis framework, and the notation and model assumptions used. In Section 3, the main reasons for the interest in estimating π_0 are explained. These reasons include the fact that estimation of π_0

is very important for the *False Discovery Rate* methodology introduced by Benjamini and Hochberg (1995). In Section 4, several estimation procedures are described, including Schweder and Spjøtvoll's (1982) nonparametric estimator, different estimators based on semi- or nonparametric p-value density estimation, and a parametric estimator. All of the estimators are based on the observed p-values corresponding to the hypothesis tests. The p-values are assumed to be independent and identically distributed when the estimation methods are derived. Since this assumption might not hold for actual real-life data sets, we discuss different ways of modelling dependence in Section 5. To test the performance of the estimation procedures, a large-scale simulation experiment was carried out, and a description of the simulation methods and the results is presented in Section 6. The estimators were tested on simulated data with different dependence structures, including the case of independence. In Section 7, the estimators are evaluated on data from two DNA microarray studies from Nygaard et al. (2003) and Hedenfalk et al. (2001), respectively. Finally, in Section 8, conclusions and plans for further work are presented.

2 Preliminaries, notation and model assumptions

This section provides a brief review of hypothesis testing, and some necessary notation and model assumptions.

2.1 Single hypothesis testing

Assume that we are given data from a distribution F_θ depending on a population parameter $\theta \in \Theta$. A test of the null hypothesis H_0 versus the alternative hypothesis H_1 can then be formulated as

$$H_0 : \theta \in \Theta_0 \text{ versus } H_1 : \theta \in \Theta_1,$$

where Θ_0 and Θ_1 are subsets of the parameter space Θ such that $\Theta_0 \cup \Theta_1 = \Theta$ and $\Theta_0 \cap \Theta_1 = \emptyset$. The test is based on a test statistic (i.e. a function of the data) T by defining a rejection set Γ. If $T \in \Gamma$ we reject H_0 and accept H_1, otherwise we accept H_0. In this situation we can commit two types of errors:

Type I error: We reject H_0 when it is true. This happens when $\theta \in \Theta_0$ and $T \in \Gamma$.

Type II error: We accept H_0 when it is false. This happens when $\theta \in \Theta_1$ and $T \notin \Gamma$.

The maximal probability of committing a type I error is called the level of significance and is denoted α. This means that $\Pr(T \in \Gamma | H_0 \text{ true}) \leq \alpha$ for a test of level α. The power function $\beta(\cdot)$ of a test is defined as the probability of rejecting the null hypothesis given the true value θ of the parameter. For false H_0, $\beta(\theta) = 1 - \Pr(\text{Type II error})$. Usually, one would attempt to maximize the power of a test for a given level α. We can base the test on the p-value corresponding to an observed statistic $T = t$:

$$\text{p–value}(t) = \inf_{\{\Gamma : t \in \Gamma\}} \Pr(T \in \Gamma | H_0 \text{ true}),$$

where Γ is the rejection set (Lehmann 1986). If the p-value is less than α we reject H_0.

2.2 Multiple hypothesis testing framework and model assumptions

In a multiple hypothesis setup, each of m related null hypothesis are tested, i.e. we test

$$H_{0i} \text{ versus } H_{1i};\ i = 1, \ldots, m.$$

We denote the corresponding random p-values $P_1, \ldots, P_m$, and the observed p-values $p_1, \ldots, p_m$.

The quantity π_0 that we want to estimate, is the probability that any given null hypothesis H_{0i} is true. Genovese and Wasserman (2002) formalize this in the following mixture model: We define random variables $H_0, \ldots, H_m$ as

$$H_i = \begin{cases} 0, & \text{if } H_{0i} \text{ is true,} \\ 1, & \text{otherwise,} \end{cases}$$

for $i = 1, \ldots, m$. We assume that each $H_i \sim \text{Bernoulli}(\pi_0)$ (i.e. $\text{Prob}(H_i = 0) = \pi_0$ and $\text{Prob}(H_i = 1) = 1 - \pi_0$), and that the H_i's are independent.

The number of true null hypotheses is then the random variable $M_0 = \sum_{i=1}^{m}(1 - H_i)$. Clearly M_0 is binomially distributed: $M_0 \sim \text{Bin}(m, \pi_0)$. A realized value of M_0 is denoted m_0.

We further assume that the p-values are continuous random variables. Then, the null p-values are uniformly distributed on $[0, 1]$. This follows almost immediately from the probability unit transformation: The cumulative distribution function $F_X(X) \sim \text{Unif}[0, 1]$ for a continuous random variable X. If we for each hypothesis $H_{0j}; j = 1, \ldots, m_0$, reject H_{0j} when a test statistic X_j is large (this situation can always be achieved by applying a suitable transformation to any test statistic), then

$$F_{X_j}(X_j) \sim \text{Unif}[0, 1] \Rightarrow P_j = 1 - F_{X_j}(X_j) \sim \text{Unif}[0, 1].$$

All the estimation methods in Section 4 are derived under the assumption of independent and identically distributed p-values. The question of dependence is addressed is Section 5, and the estimators are tested on simulated data with different dependence structures in Section 6.

2.3 Mixture model and identifiability

The estimation methods are based on the following mixture model for the density f of the p-values:

$$f = \pi_0 + (1 - \pi_0)h,$$

where h is the density of the p-values corresponding to false null hypotheses. This model is a direct concequence of the assumptions made in Section 2.2.

On the basis of the mixture model, consider the question of identifiability of π_0, as illustrated in Figure 2. In Figure 2, two p-value histograms, from simulated p-values, are shown. The portion of each histogram corresponding to the null p-values is shaded, and the portion corresponding to the alternative is transparent. The left histogram shows the situation when π_0 is identifiable, which in our situation basically means that the density h of the alternative p-values equals zero when p gets large. The right histogram illustrates that in the case when many alternative p-values are large, π_0 will probably be overestimated. Assuming that f is decreasing, if $h(1) = 0$ then π_0 is identifiable.

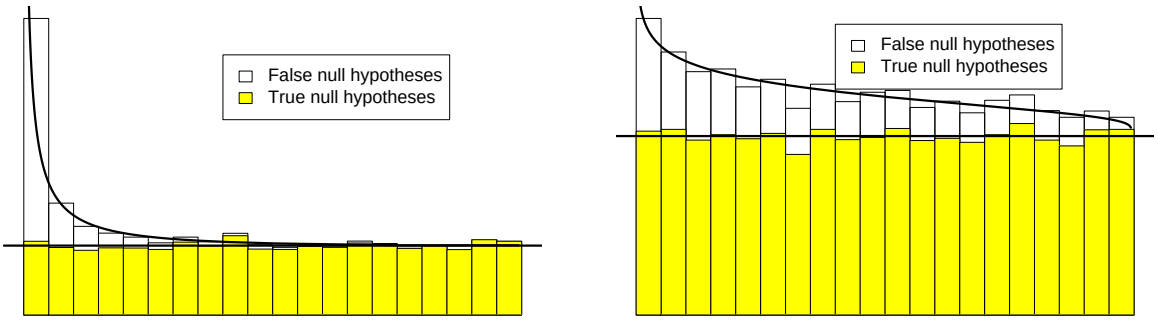


Figure 2: The mixture model and the problem of identifiability.

It is beyond the scope of this work to thoroughly discuss identifiability of π_0. See Genovese and Wasserman (2003, Section 3.1) for a discussion of identifiability of π_0 and Prakasa Rao (1992, Section 8) for a general introduction to the problem of identifiability in mixture models. When deriving the estimation methods, we shall assume that π_0 is identifiable.

2.4 Multiple testing error rates

In this multiple hypothesis setup we would like to control some adequate error measure ER. We distinguish between *weak* and *strong* control of ER:

Weak control of an error rate ER at level α, say, means that $\mathrm{ER} \leq \alpha$ for $m_0 = m$ (all null hypotheses are true).

Strong control means that we keep $\mathrm{ER} \leq \alpha$ for all values of m_0 simultaneously.

Generally, each test has its own type I and type II error probabilities, and it is not obvious how we should measure the overall error rate. For each test, there are four possibilities:

1. H_0 is true and accepted (correct decision)
2. H_0 is true and rejected (type I error)
3. H_0 is false and accepted (type II error)
4. H_0 is false and rejected (correct decision)

Each of the m tests will be of one of the above categories. Let U,V,T and S of the tests be of categories 1, 2, 3 and 4, respectively, let $R = U + T$ be the number of rejected null hypothesis, and let $W = m - R$ be the number of accepted null hypotheses. These are all random variables, of course, but only R and W can be observed. This situation is displayed in Table 1 below.

	Accept	Reject	Total
H_0 true	U	V	m_0
H_0 false	T	S	$m - m_0$
Total	W	R	m

Table 1: Outcomes from m hypothesis tests

The traditional error rate to control is the Family Wise Error Rate (FWER). This is defined as $\mathrm{Prob}(V \geq 1)$, the probability of committing at least one type I error. To control the FWER at level α requires that each individual test is conducted at a lower level. For example, using the familiar Bonferroni procedure to control the FWER at level α, each null hypothesis H_{0i} is rejected when $p_i \leq \alpha/m$. Demanding strong control of the FWER is a strict criterion, and can result in low power to reject alternative hypotheses.

Benjamini and Hochberg (1995) introduced a new multiple testing error rate called the *false discovery rate* (FDR). The FDR is discussed in Section 3.1.

3 Why estimate the proportion π_0 of true null hypotheses?

The most important reason for wanting to estimate π_0 is that it is a quantity of independent interest in many situations. In the case of testing for differential expression in DNA microarrays, the proportion of differentially expressed genes is $1 - \pi_0$. Clearly it is important to know whether 5% or 35% of the genes are differentially expressed, even if we cannot identify these genes.

In addition to this, a reliable estimate of π_0 is crucial when we want to assess or control the so-called false discovery rate (FDR), and it is also central in the estimation of Storey's (2002b) q-values. In Section 3.1 we review the FDR and its relation to the estimation of π_0, and in Section 3.2 we present Storey's q-values.

Throughout this section, we use the notation $\hat{\pi}_0$ for any estimate of π_0. Methods for estimating π_0 will be discussed in Section 4.

3.1 The false discovery rate (FDR)

The false discovery rate (FDR), introduced by Benjamini and Hochberg (1995), is an error rate which can provide a substantial gain in power in situations where control of the FWER is not necessary. As an example, when we test thousands of genes for differential expression using DNA microarrays, it seems unnecessarily restrictive to control the probability of making a single mistake. In most cases, the DNA microarray experiment would only be a screening to pick out genes for further study, and then one would prefer to accept a few false discoveries rather than sacrificing the power of the testing procedure.

In these situations, Benjamini and Hochberg (1995) argue that the interesting quantity is the proportion of erroneously rejected hypotheses among the rejected ones, i.e. V/R in the notation from Table 1 on page 6. The FDR is defined as the expectation of this proportion if $R > 0$, and zero if $R = 0$:

$$\mathrm{FDR} = \begin{cases} \mathrm{E}(V/R) & \text{if } R > 0, \\ 0 & \text{if } R = 0. \end{cases}$$

Equivalently, this can be written as

$$\mathrm{FDR} = \mathrm{E}(V/R \,|\, R > 0)\,\mathrm{Prob}(R > 0).$$

Storey (2002a) prefers an alternative "false discovery rate" definition, which he terms the *positive false discovery rate* (pFDR). The pFDR is defined to be

$$\mathrm{pFDR} = \mathrm{E}(V/R \,|\, R > 0).$$

It is called "positive" because it is conditioned on the fact that $R > 0$. The introduction of the pFDR is motivated by concerns about what happens when $\mathrm{Prob}(R > 0)$ is much less than one (in which case the FDR might be misleading), as well as mathematical tractability. Whether the FDR or pFDR is the most appropriate error measure is still under dispute; we will not get into that debate here, but only refer to Storey (2002a, 2002b) for an argument in favor of the pFDR.

The main reason for mentioning the pFDR in this work is that it forms the basis for the q-values described in Section 3.2.

3.1.1 Controlling the FDR

Simes (1986) provides a procedure which has weak control of the FWER at level α: Given the ordered, observed p-values $p_{(1)}, \ldots, p_{(m)}$, let $\hat{k} = \max\{k : p_{(k)} \leq \alpha\frac{k}{m}\}$. Then, reject all null hypotheses corresponding to $p_{(i)}$ for $i \leq \hat{k}$.

Benjamini and Hochberg (1995) showed that this procedure strongly controls the FDR at level $\pi_0\alpha$. Therefore, if an estimate $\hat{\pi}_0$ of π_0 is available, calculating $\hat{l} = \max\{l : \hat{\pi}_0 p_{(l)} \leq \alpha\frac{l}{m}\}$ and rejecting the null hypotheses corresponding to $p_{(1)}, \ldots, p_{(\hat{l})}$ provides strong control at approximately level α.

If π_0 is significantly less than one, and $\hat{\pi}_0$ is a sufficiently good estimate, then the utilization of $\hat{\pi}_0$ in this way leads to an increase in power, while still achieving control at (approximately) level α.

Using $\hat{\pi}_0$ as above in Simes's (1986) procedure is discussed both by Storey (2002b) and Reiner et al. (2003), although from somewhat different viewpoints.

3.1.2 Estimating the FDR for a fixed rejection region

Storey (2002a) investigates the estimation of the false discovery rate when the rejection region Γ is fixed beforehand. Determining Γ *a priori* might seem like putting the cart before the horse, but Storey (2002b, p. 41) argues that "experts in a particular field (for example, DNA microarrays) run similar experiments over and over. Often they are able to judge from their experience which statistics are likely to be significant". In such situations, fixing Γ would make sense.

To simplify our discussion here, we follow Storey (2002a) and assume that each test has the same rejection region Γ (this should be natural in most cases), and (without loss of generality) that we base the tests on the observed p-values $p_1, \ldots, p_m$. Then, the rejection region $\Gamma = [0, \gamma]$ for some $\gamma \in [0, 1]$, and we reject all null hypotheses corresponding to p-values less than γ.

Now, approximately $m\pi_0$ of the observed p-values correspond to the true null hypotheses. Since these p-values are uniformly distributed, about $m\gamma\pi_0$ null p-values are less then γ, and therefore approximately $m\gamma\pi_0$ true null hypotheses will be rejected. The *total* number of rejected hypotheses (null or alternative) is simply $R(\gamma) = \#\{p_i \leq \gamma\}$. Since the false discovery rate is the expectation of the proportion of rejected null hypotheses among all rejected hypotheses, a natural estimate of the FDR is then

$$\widehat{\mathrm{FDR}}(\gamma) = \frac{m\gamma\hat{\pi}_0}{\max(1, R(\gamma))}, \tag{3.1}$$

where the "max" in the denominator is needed in case $R(\gamma) = 0$ (Storey 2002b).

The pFDR can be estimated in a similar way, as suggested by Storey (2002b). Since $\mathrm{pFDR} = \mathrm{FDR}/\mathrm{Prob}(R > 0)$, an estimate of the FDR divided by an estimate of $\mathrm{Prob}(R > 0)$ is an estimate of the pFDR. Clearly $\mathrm{Prob}(R > 0)$ must be greater than $1 - (1 - \gamma)^m$, since the latter expression is simply the probability that $R > 0$ given that all null hypothesis are true (and all p-values uniformly distributed). Increasing the number of false null hypotheses would only increase $\mathrm{Prob}(R > 0)$. Therefore, an estimate of the pFDR is given by

$$\widehat{\mathrm{pFDR}}(\gamma) = \frac{m\gamma\hat{\pi}_0}{((1 - (1 - \gamma)^m)\max(1, R(\gamma))}, \tag{3.2}$$

Looking at Equations (3.1) and (3.2) makes it clear that the quality of the estimation of both the false discovery rate and the positive false discovery rate relies heavily on having a good estimator $\hat{\pi}_0$ at our

disposal. This is one reason for our interest in estimating π_0.

3.2 Storey's q-values

Storey's (2002b) q-value is an error measure for *each* observed statistic (or p-value) in terms of the pFDR. The q-value is defined as

$$\text{q–value}(t) = \inf_{\{\Gamma:\ t\in\Gamma\}} \text{pFDR}(\Gamma)$$

for an observed statistic $T = t$.

To understand this definition, we first take a closer look at the interpretation of the pFDR. Assume that m tests are performed with the test statistics $T_1, \ldots, T_m$, each with rejection region Γ. Then, Storey (2002b) shows that the pFDR can be written as the following posterior probability:

$$\text{pFDR}(\Gamma) = \text{Prob}(H = 0|T \in \Gamma)$$

(where we drop the index i since $\text{Prob}(H_i = 0|T_i \in \Gamma)$ are the same for each i). This implies that the q-value can be written as

$$\text{q–value}(t) = \inf_{\{\Gamma:\ t\in\Gamma\}} \text{Prob}(H = 0|T \in \Gamma) \tag{3.3}$$

for $T = t$. From the definition of the p-value, the p-value for an observed statistic $T = t$ is

$$\text{p–value}(t) = \inf_{\{\Gamma:\ t\in\Gamma\}} \text{Prob}(T \in \Gamma|H = 0).$$

This nice relation between the p–value and the q–value is the reason for the q–value's name, as well as one of the main reasons for its appeal. The expression for the pFDR in Equation (3.3) can be given a Bayesian interpretation — it is the posterior probability that a null hypothesis is true given that it is rejected. This means that the q–value of a test statistic can be interpreted as the minimal posterior probability that the corresponding null hypothesis is true, where the minimum is over all rejection regions containing the test statistic. So, the q–value provides us with an error measure for *each* observed statistic.

It might be instructive to phrase this in terms of the case where the test statistics are p-values. Given the observed ordered p-values $p_{(1)}, \ldots, p_{(m)}$, the naturally occuring rejection regions Γ are of the form $[0, p_{(k)}]$, where k is the number of null hypotheses that are rejected (k is a realization of R from Table 1). So, for each p-value $P_{(i)} = p_{(i)}$, the corresponding q–value is

$$\text{q–value}(\gamma) = \inf_{\{\gamma:\gamma\geq p_{(i)}\}} \text{pFDR}(\gamma), \tag{3.4}$$

where $\text{pFDR}(\gamma)$ denotes $\text{pFDR}([0, \gamma])$.

The q-values can be estimated from the p-values, using the estimator $\widehat{\text{pFDR}}(\gamma)$ from (3.2) for the pFDR. Considering the expression (3.4) for the q–value, a natural estimator $\hat{q}(p_i)$ is given by

$$\hat{q}(p_i) = \min_{\gamma\geq p_i} \widehat{\text{pFDR}}(\gamma) = \min_{\gamma\geq p_i} \frac{m\gamma\hat{\pi}_0}{\left((1-(1-\gamma)^m)\right)\max\left(1, R(\gamma)\right)}, \tag{3.5}$$

where $R(\gamma) = \#\{p_j \leq \gamma\}$.

Clearly it is important to have a good estimate $\hat{\pi}_0$ to estimate the q–values, and this is also part of our motivation for estimating π_0.

4 Estimators of π_0

In this section, some known estimators of π_0 are described, and new estimators are developed.

A brief outline of this section: In Section 4.1, Schweder and Spjøtvoll's (1982) estimator $\widehat{\pi}_0(\lambda)$ is presented, along with Storey's (2002b) suggestions for the choice of the tuning parameter λ, as well as some other considerations regarding this estimator. In Sections 4.2, 4.3 and 4.4 we describe different estimators based on semi- or nonparametric estimation of the p-value density, and in Section 4.5, a parametric estimator of π_0 is presented.

4.1 Schweder and Spjøtvoll's estimator of π_0

Schweder and Spjøtvoll (1982) suggest an estimator $\widehat{\pi}_0(\lambda)$ which is based on the following reasoning. Let $W(\lambda) = \#\{p_j > \lambda\}$, the number of p-values greater than some value λ. Since the p-values associated with the false null hypotheses should be small, a large majority of the p-values in the interval $[\lambda, 1]$ should be corresponding to the true null hypothesis, and thus Unif[0,1]-distributed. This implies that the expected value of $W(\lambda)$ should be approximately equal to the product of $m\pi_0$ and the length of the interval$[\lambda, 1]$, that is,

$$\mathrm{E}(W(\lambda)) \approx m\pi_0(1-\lambda).$$

Therefore,

$$\widehat{\pi}_0(\lambda) = \frac{W(\lambda)}{m(1-\lambda)} = \frac{\#\{p_j > \lambda\}}{m(1-\lambda)}$$

is a reasonable estimator for π_0 for a given λ.

Schweder and Spjøtvoll (1982) plotted the ordered values $q_{(i)} = 1 - p_{(i)}$, sorted in ascending order, versus their rank. An example of such a plot, based on simulated p-values, is shown in Figure 3. The p-values corresponding to the true null hypotheses should fall approximately on a straight line in the left portion of this plot, since they are uniformly distributed. In Figure 3, this straight line is simply fitted by eye, and the estimate of π_0 is found as the height at the right end of the line. Note that this can be seen as a way of choosing λ. Methods for choosing λ will be discussed later in this section. A method directly based on Schweder and Spjøtvoll's (1982) plot is described in Section 4.1.5.

4.1.1 Expectation, variance and mean square error

Expressions for the expectation, variance and mean square error of Schweder and Spjøtvoll's (1982) estimator $\widehat{\pi}_0(\lambda)$ for a given λ are derived in this section. These expressions shed some light on the properties of the estimator, and will be useful for the discussion in the subsequent sections.

Let the cumulative distribution function corresponding to the density h of the alternative p-values be H. Furthermore, we let M_0 denote the random number of true null hypothesis, and let m_0 be any realized value of M_0. We define random functions $U(\cdot)$ and $T(\cdot)$ by $U(\lambda) = \#\{\text{null } p_j > \lambda\}$ and $T(\lambda) = \#\{\text{alternative } p_j > \lambda\}$ (this notation is in accordance with Table 1 on page 6), such that $W(\lambda) = U(\lambda) + T(\lambda)$. The derivation of expectation and variance are based on the observation that

$$\begin{aligned} M_0 &\sim \mathrm{Bin}(m, \pi_0) \\ U(\lambda)|M_0 &\sim \mathrm{Bin}(M_0, 1-\lambda), \\ T(\lambda)|M_0 &\sim \mathrm{Bin}(m - M_0, 1 - H(\lambda)), \end{aligned}$$

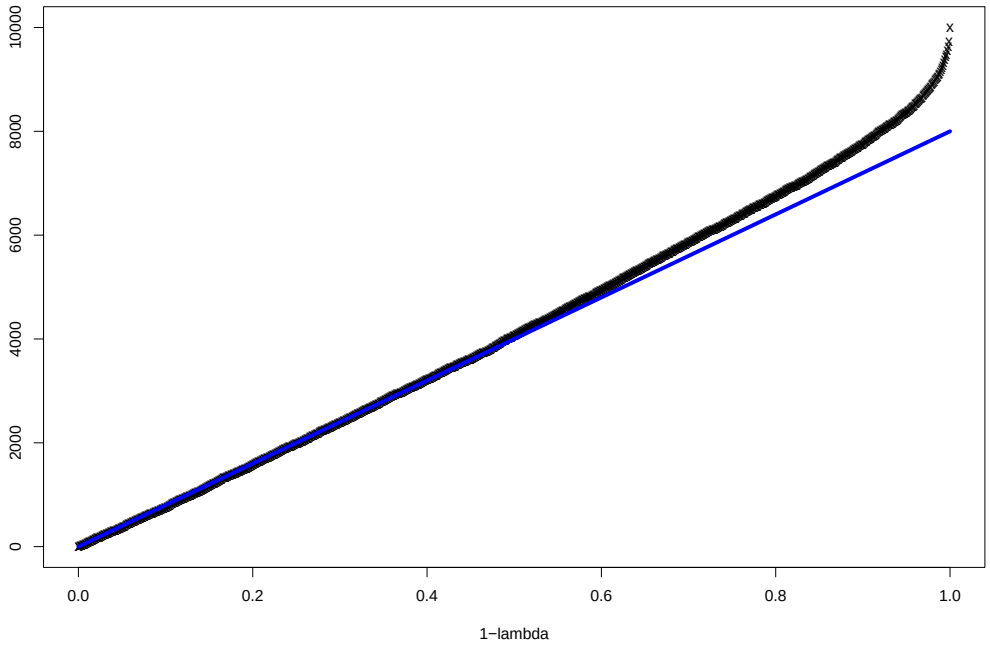


Figure 3: Schweder and Spjøtvoll's p-value plot

and the well-known result that

$$\begin{aligned} \mathrm{E}(X) &= \mathrm{E}(\mathrm{E}(X|Y)), \\ \mathrm{Var}(X) &= \mathrm{E}(\mathrm{Var}(X|Y)) + \mathrm{Var}(\mathrm{E}(X|Y)) \end{aligned}$$

for any random variables X and Y (Casella and Berger 1990).

We first consider the expectation of $\widehat{\pi}_0(\lambda)$:

$$\begin{aligned} \mathrm{E}(\widehat{\pi}_0(\lambda)) &= \frac{\mathrm{E}(U(\lambda)) + \mathrm{E}(T(\lambda))}{m(1-\lambda)} \\ &= \frac{\mathrm{E}(\mathrm{E}(U(\lambda)|M_0)) + \mathrm{E}(\mathrm{E}(T(\lambda)|M_0))}{m(1-\lambda)} \\ &= \frac{\mathrm{E}((1-\lambda)M_0) + \mathrm{E}((1-H(\lambda))(m-M_0))}{m(1-\lambda)} \\ &= \frac{(1-\lambda)m\pi_0 + (1-H(\lambda))m(1-\pi_0)}{m(1-\lambda)} \\ &= \pi_0 + \frac{1-H(\lambda)}{1-\lambda}(1-\pi_0). \end{aligned} \tag{4.1}$$

Notice that $\mathrm{E}(\widehat{\pi}_0(\lambda)) \geq \pi_0$, a point we will return to shortly.

The variance can be calculated similarly:

$$\begin{aligned}
\mathrm{Var}(\widehat{\pi}_0(\lambda)) &= \frac{\mathrm{Var}(U(\lambda)) + \mathrm{Var}(T(\lambda))}{m^2(1-\lambda)^2} \\
&= \frac{\mathrm{E}(\mathrm{Var}(U|M_0)) + \mathrm{Var}(\mathrm{E}(U|M_0)) + \mathrm{E}(\mathrm{Var}(T|M_0)) + \mathrm{Var}(\mathrm{E}(T|M_0))}{m^2(1-\lambda)^2} \\
&= \frac{\lambda(1-\lambda)m\pi_0 + (1-\lambda)^2 m\pi_0(1-\pi_0) + H(\lambda)(1-H(\lambda))m(1-\pi_0)}{m^2(1-\lambda)^2} \\
&\quad + \frac{(1-H(\lambda))^2 m\pi_0(1-\pi_0)}{m^2(1-\lambda)^2} \\
&= \frac{\lambda\pi_0}{m(1-\lambda)} + \frac{H(\lambda)(1-H(\lambda))(1-\pi_0)(1+(1-H(\lambda))\pi_0)}{m(1-\lambda)^2} + \frac{\pi_0(1-\pi_0)}{m}. \quad (4.2)
\end{aligned}$$

The mean square error (MSE) is defined as

$$\mathrm{MSE}(\widehat{\pi}_0(\lambda)) = \mathrm{E}[(\widehat{\pi}_0(\lambda) - \pi_0)^2]. \quad (4.3)$$

It is easy to show (see e.g. Casella and Berger (1990)) that

$$\mathrm{MSE}(\widehat{\pi}_0(\lambda)) = \mathrm{Var}(\widehat{\pi}_0(\lambda)) + [\mathrm{Bias}(\widehat{\pi}_0(\lambda))]^2, \quad (4.4)$$

so the mean square error combines the error resulting from bias and variance. Given the expression for the expectation and variance in Equations (4.1) and (4.2), the MSE of $\widehat{\pi}_0(\lambda)$ is therefore given by

$$\begin{aligned}
\mathrm{MSE}(\widehat{\pi}_0(\lambda)) &= \frac{\lambda\pi_0}{m(1-\lambda)} + \frac{H(\lambda)(1-H(\lambda))(1-\pi_0)(1+(1-H(\lambda))\pi_0)}{m(1-\lambda)^2} + \frac{\pi_0(1-\pi_0)}{m} \\
&\quad + \left(\frac{1-H(\lambda)}{1-\lambda}(1-\pi_0)\right)^2. \quad (4.5)
\end{aligned}$$

4.1.2 Choice of λ

To use Schweder and Spjøtvoll's (1982) estimator we need to choose a value for the 'tuning parameter' λ. From the expression for the variance of $\widehat{\pi}_0(\lambda)$ in Equation (4.2), it is easily seen that choosing a small value for λ will give a small variance. However, for λ small, many of the non-null (nonuniformly distributed) p-values will be included in $W(\lambda)$, and this will give a larger bias. We see that there is a trade-off between bias and variance when choosing λ. Therefore, we aim to find the λ which minimizes $\mathrm{MSE}(\widehat{\pi}_0(\lambda))$.

The true MSE is unknown, so we need to find an estimate $\widehat{\mathrm{MSE}}(\lambda)$. Storey (2002b) suggests estimating $\mathrm{MSE}(\widehat{\pi}_0(\lambda))$ for fixed λ using bootstrapping, which is a computer-based resampling technique. Taking (4.3) as the starting point, the bootstrap estimator of $\mathrm{MSE}(\widehat{\pi}_0(\lambda))$ is

$$\widehat{\mathrm{MSE}}(\lambda) = \frac{1}{B}\sum_{b=1}^{B}\left(\widehat{\pi}_0^{*b}(\lambda) - \widehat{\pi}_0^{p}\right)^2,$$

where $\widehat{\pi}_0^{p}(\lambda)$ is a plug-in estimator of π_0 (see Section 4.1.3 for how this is chosen by Storey), and $\widehat{\pi}_0^{*b}(\lambda), b = 1, \ldots, B$ are bootstrap versions of $\widehat{\pi}_0(\lambda)$, obtained as follows: For a given λ, we draw B

with-replacement samples of m p-values each from the original p-values, and for each such sample we calculate $\widehat{\pi}_0^{*b}(\lambda)$ from the resampled p-values.

Given $\widehat{\pi}_0^p$, the bootstrap estimate of the MSE is then

$$\widehat{\mathrm{MSE}}(\lambda) = \frac{1}{B}\sum_{b=1}^{B}\left(\widehat{\pi}_0^{*b}(\lambda) - \widehat{\pi}_0^p\right)^2.$$

Now, to estimate the MSE-optimal λ, we choose a range $\mathcal{R}$ of values for λ (e.g. $\mathcal{R} = \{0, 0.05, \ldots, 0.95\}$), and the choice $\hat{\lambda}$ is simply the $\lambda \in \mathcal{R}$ which minimizes $\widehat{\mathrm{MSE}}(\lambda)$.

4.1.3 Choice of plug-in estimator of π_0

How should we choose the plug-in estimator $\widehat{\pi}_0^p$? Storey (2002b) suggests choosing

$$\widehat{\pi}_0^p = \min_{\lambda' \in \mathcal{R}}[\widehat{\pi}_0(\lambda')].$$

This is motivated by the following observations. From Equation (4.1) in Section 4.1.1 we know that for any given λ, $\mathrm{E}(\widehat{\pi}_0(\lambda)) \geq \pi_0$. This immediately implies that $\min_{\lambda' \in \mathcal{R}}[\mathrm{E}(\widehat{\pi}_0(\lambda'))] \geq \pi_0$, and it is obviously also the case that $\mathrm{E}(\widehat{\pi}_0(\lambda)) \geq \min_{\lambda' \in \mathcal{R}}[\mathrm{E}(\widehat{\pi}_0(\lambda'))]$ for and $\lambda \in \mathcal{R}$. All of this means that, for any $\lambda \in \mathcal{R}$, we have

$$\mathrm{E}(\widehat{\pi}_0(\lambda)) \geq \min_{\lambda' \in \mathcal{R}}[\mathrm{E}(\widehat{\pi}_0(\lambda'))] \geq \pi_0.$$

From this inequality, Storey (2002b) immediately concludes that $\min_{\lambda' \in \mathcal{R}}[\widehat{\pi}_0(\lambda')]$ is the natural plug-in estimator. At first sight, the inequality looks quite convincing. The plug-in estimator seems to be "sandwiched" between the true π_0 and the expectation of its estimator $\widehat{\pi}_0(\lambda)$. However, the interesting quantity here is the expectation of the plug-in estimator, $\mathrm{E}\left(\min_{\lambda' \in \mathcal{R}}[\widehat{\pi}_0(\lambda')]\right)$, not $\min_{\lambda' \in \mathcal{R}}[\mathrm{E}(\widehat{\pi}_0(\lambda'))]$. Therefore, in order to conclude that Storey's (2002b) choice of plug-in estimator is reasonable, we would rather want to have

$$\mathrm{E}(\widehat{\pi}_0(\lambda)) \geq \mathrm{E}\left(\min_{\lambda' \in \mathcal{R}}[\widehat{\pi}_0(\lambda')]\right) \geq \pi_0,$$

which is does *not* hold in general. All we can say is that $\mathrm{E}(\widehat{\pi}_0(\lambda)) \geq \mathrm{E}\left(\min_{\lambda' \in \mathcal{R}}[\widehat{\pi}_0(\lambda')]\right)$ and $\mathrm{E}(\widehat{\pi}_0(\lambda)) \geq \pi_0$ for any $\lambda \in \mathcal{R}$, so it could might as well be the case that

$$\mathrm{E}(\widehat{\pi}_0(\lambda)) \geq \pi_0 \geq \mathrm{E}\left(\min_{\lambda' \in \mathcal{R}}[\widehat{\pi}_0(\lambda')]\right).$$

In the latter situation, the plug-in estimator would tend to underestimate π_0. Simulations have shown that this is often the case in practice (data not included).

From these observations we conclude that Storey (2002b) provides insufficient justification for his choice of plug-in estimator, which does not necessarily mean that it is a poor choice. We will use Storey's plug-in estimator in what follows.

4.1.4 Bootstrap-algorithm for choosing optimal λ

Now that we have decided on the plug-in estimator

$$\widehat{\pi}_0^p = \min_{\lambda' \in \mathcal{R}}[\widehat{\pi}_0(\lambda')],$$

we can formally specify the algorithm for choosing the value $\hat{\lambda}$ of λ, which is given in pseudocode as Algorithm 1 below.

Algorithm 1 Storey's algorithm for optimal choice of λ

for all $\lambda \in \mathcal{R}$ **do**
  for $b = 1$ to B **do**
    draw $p_1^{*b}, \ldots, p_m^{*b}$ with replacement from the m p-values
    $\widehat{\pi}_0^{*b}(\lambda) \leftarrow \#\{p_j^{*b} : p_j^{*b} > \lambda\}/[m(1-\lambda)]$
  end for
  $\widehat{\mathrm{MSE}}(\lambda) \leftarrow \frac{1}{B} \sum_{b=1}^{B} \left(\widehat{\pi}_0^{*b}(\lambda) - \min_{\lambda' \in \mathcal{R}}[\widehat{\pi}_0(\lambda')]\right)^2$
end for
$\hat{\lambda} \leftarrow \mathrm{argmin}_{\lambda \in \mathcal{R}} \widehat{\mathrm{MSE}}(\lambda)$

The overall estimate of π_0 is then

$$\widehat{\pi}_0 = \widehat{\pi}_0(\hat{\lambda}).$$

4.1.5 An alternative choice of λ

Turkheimer et al. (2001) suggest an alternative way to choose λ. Their approach is based on Schweder and Spjøtvoll's (1982) p-value plot, described in the beginning of this section, and illustrated in Figure 3.

Turkheimer et al. (2001) observe that there should exist a break-point in the plot at the point where the p-values are from the alternative hypotheses, as they as expected to be decidedly non-uniform. Turkheimer et al.'s (2001) method chooses λ to be at this break-point. In practice, this is done by testing smaller and smaller sets of p-values for independence, where the largest p-value is left out in each iteration if the hypothesis of independence is rejected. Then, λ is chosen to be the largest p-value in the first such set which is not rejected by this uniformity test. We refer to the paper by Turkheimer et al. (2001) for further details[2].

4.1.6 Smoothing of Schweder and Spjøtvoll's estimator over λ

The Schweder and Spjøtvoll's (1982) estimator, can be plotted as a function of λ, as shown in Figure 4. Storey and Tibshirani (2003) present another procedure for estimating π_0 based on Schweder and Spjøtvoll's (1982) estimator. Storey and Tibshirani (2003) proceed as follows: First $\widehat{\pi}_0(\lambda)$ are calculated over a (fine) grid of λ — in the article the range $\{0, 0.01, 0.02, \ldots, 0.95\}$ is used as an example. Then, a weighted natural cubic spline y, with 3 degrees of freedom, is fitted to the $(\lambda, \widehat{\pi}_0(\lambda)))$, where the weight $1 - \lambda$ is used for each λ. Finally, π_0 is estimated by $\widehat{\pi}_0 = y(1)$. (In the R code provided by Storey and Tibshirani (2003) $\widehat{\pi}_0 = y(0.95)$.)

We find this procedure to be somewhat *ad hoc*. In particular, the choice of grid and smoothing method seems quite arbitrary. In addition, the smoothing of $\widehat{\pi}_0(\lambda)$ over a range of λ is essentially equivalent to using estimates of the p-value density as the basis for estimation for π_0. This can be seen by the following considerations. Let $\hat{F}$ be the empirical distribution of the observed p-values, F the true

[2]Matlab code for estimation of π_0 with this method is freely available at http://www.irsl.org/∼fet/pplot/pplot.html

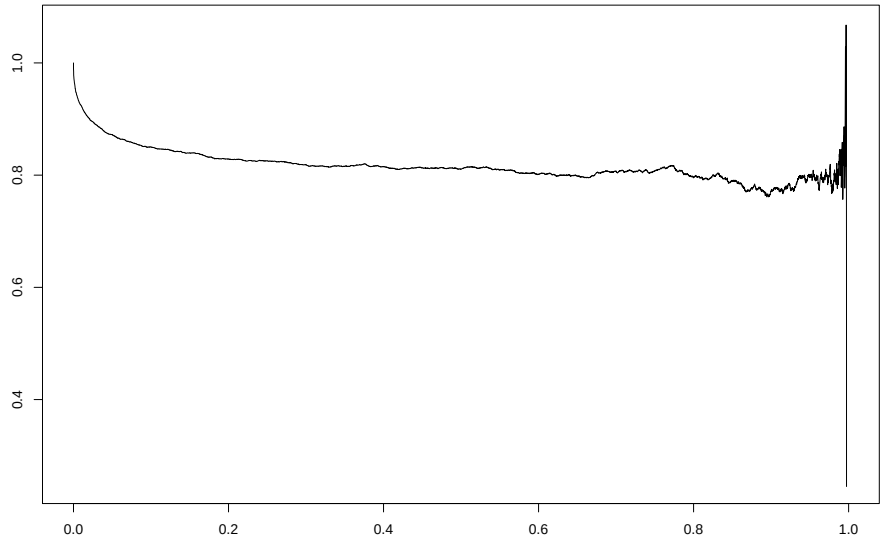


Figure 4: Schweder and Spjøtvoll's estimator as a function of λ. (Data generated with true value π_0=0.8)

(unknown) cumulative distribution function and f the density of the p-values. Note that $\widehat{\pi}_0(\lambda)$ can be written as

$$\widehat{\pi}_0(\lambda) = \frac{1 - \hat{F}(\lambda)}{1 - \lambda}.$$

Therefore, we can view $\widehat{\pi}_0(\lambda)$ as a plug-in estimator for the quantity $\pi_0(\lambda)$, defined by

$$\pi_0(\lambda) = \frac{1 - F(\lambda)}{1 - \lambda}.$$

A Taylor expansion of $F(p)$ about the point λ gives

$$F(p) = F(\lambda) + (p - \lambda)f(\lambda) + \frac{1}{2}(p - \lambda)^2 f'(\lambda) + \cdots .$$

Inserting $p = 1$ we obtain

$$1 = F(1) = F(\lambda) + (1 - \lambda)f(\lambda) + \frac{1}{2}(1 - \lambda)^2 f'(\lambda) + \cdots ,$$

which implies that

$$\pi_0(\lambda) = f(\lambda) + \frac{1}{2}(1 - \lambda)f'(\lambda) + \cdots .$$

This means that smoothing over $\widehat{\pi}_0(\lambda)$ near $\lambda = 1$ is asymptotically related to density estimation of f in the setting of estimating π_0 — the two approaches are strongly related.

In Sections 4.2, 4.3 and 4.4 we present several estimators of π_0 which are based on estimation of p-value density. We feel that this formulation is more natural and intuitively pleasing, since it is based on making interpretable restrictions on the p-value density. For these reasons, we have chosen not to further discuss Storey and Tibshirani's (2003) smoothing-procedure.

4.2 Two estimators of π_0 based on decreasing density estimation

In this section, we present two estimators of π_0 based on the non-parametric maximum likelihood decreasing density estimate of the p-values. This density estimator is known as the Grenander estimator (Grenander 1956). The idea of using the Grenander estimator for estimation of π_0 was also mentioned by Genovese and Wasserman (2003).

4.2.1 The Grenander estimator of the p-value density

Consider the problem of finding an estimate $\hat{f} \in \mathcal{F}$, of the density f of the p-values, where $\mathcal{F}$ is the set of decreasing density functions on $\mathbb{R}^+$, the positive real numbers. (Note that the p-values are all in $\mathbb{R}^+$). Let $p_{(1)} \leq \ldots \leq p_{(m)}$ be the ordered observed p-values from the m hypothesis tests. We choose $\hat{f}$ to be the nonparametric maximum likelihood estimator (NPMLE) of f. This is defined as

$$\hat{f} = \arg\max_{z \in \mathcal{F}} \prod_{i=1}^{m} z(p_{(i)}). \tag{4.6}$$

Intuitively, this means that we choose the decreasing density function which "maximizes the probability" of the observed sample. The solution $\hat{f}$ is known as the Grenander estimator (Grenander 1956). There is a simple expression for the values of $\hat{f}_i = \hat{f}(p_{(i)}); i = 1, \ldots, m$, in terms of minima and maxima (Robertson et al. 1988):

$$\hat{f}_i = \min_{l \leq i-1} \max_{k \geq i} \frac{\hat{F}(p_{(k)}) - \hat{F}(p_{(l)})}{p_{(k)} - p_{(l)}}, \tag{4.7}$$

where $\hat{F}$ is the empirical cumulative distribution function of the p-values. Determining only $\hat{f}_1, \ldots, \hat{f}_m$ is sufficient, because $\hat{f}$ is constant on each interval $(p_{(i)}, p_{(i+1)}]$. If it was not, we could replace $\hat{f}$ on $(p_{(i)}, p_{(i+1)}]$ with the mean $\tilde{f}(p_{(i+1)})$ of $\hat{f}$ over the interval (a constant). Since $\hat{f}$ is decreasing, $\tilde{f}(p_{(i+1)}) > \hat{f}(p_{(i+1)})$, and then (4.6) implies that $\tilde{f}$ gives a higher likelihood than $\hat{f}$. But then $\hat{f}$ cannot be the NPMLE, which is a contradiction (cf. Robertson et al. (1988)).

Equation (4.7) means that the Grenander estimate is the (left-hand) slope of the least concave majorant (LCM) of the empirical cumulative distribution function of the p-values[3].

4.2.2 Estimating π_0 by the minimum of the Grenander estimator

Suppose that we have calculated the Grenander estimate $\hat{f}$ of Section 4.2.1 for a sample of p-values from the density f. Considering the mixture model $f = \pi_0 + (1 - \pi_0)h$, the estimator

$$\hat{\pi}_0^g = \hat{f}(1)$$

immediately suggests itself.

Note that to compute $\hat{\pi}_0^g = \hat{f}(1) = \hat{f}_m$, we actually only need to calculate

$$\hat{f}_m = \min_{l \leq m-1} \frac{\hat{F}(p_{(m)}) - \hat{F}(p_{(l)})}{p_{(m)} - p_{(l)}} = \min_{l \leq m-1} \frac{1 - l/m}{p_{(m)} - p_{(l)}}, \tag{4.8}$$

[3]The LCM of a given function $\phi(t)$ at each point t is defined as the infimum at each point t of the values at t of all concave functions whose graphs lie entirely above ϕ (Robertson et al. 1988). Informally, this means that the LCM is the smallest concave function above ϕ.

by the definition of $\hat{F}$, which limits the computational effort considerably. However, in this study we will still make use of the complete density estimate. This is because we want to examine the density estimates for simulated p-values, and also because we will consider another estimator of π_0 based on $\hat{f}$ in Section 4.2.3, namely $\hat{\pi}_0^l$.

Note that the estimator $\hat{\pi}_0^g$ is very similar to Storey's (2002b) plug-in estimator $\widehat{\pi}_0^p$, described in Section 4.1.3:

$$\widehat{\pi}_0^p = \min_{\lambda \in \mathcal{R}} \frac{\hat{F}(1) - \hat{F}(\lambda)}{1 - \lambda} = \min_{\lambda \in \mathcal{R}} \frac{1 - \hat{F}(\lambda)}{1 - \lambda},$$

where $\mathcal{R}$ is a grid over $[0, 1)$. The only differences between $\hat{\pi}_0^g$ and $\widehat{\pi}_0^p$ is that $p_{(m)}$ is replaced by 1 in the denominator, and the fact that the minimum is taken over the grid $\mathcal{R}$ for $\widehat{\pi}_0^p$ and over all observed p-values for $\hat{\pi}_0^g$. Clearly, $p_{(m)} \approx 1$ in most cases.

Simply computing $\hat{f}$ by direct use of Equation (4.7) is computationally inefficient. We have chosen to make use of the fact that NPMLE density estimation can be considered a special case of *antitonic regression*, which implies that $\hat{f}$ can be calculated more efficiently using an algorithm known as the *Pool Adjacent Violators Algorithm* (PAVA). We omit the description of these implementational matters, interested readers are referred to the book by Robertson et al. (1988).

4.2.3 Estimating π_0 at the longest constant interval in the Grenander estimator

Given the ordered p-values $p_1, \ldots, p_{(m)}$ we can now find the decreasing density estimate $\hat{f}(p)$ and compute the estimator

$$\hat{\pi}_0^g = \hat{f}(1)$$

described in Section 4.2.2. However, this does not work well in practice.

The problem is illustrated in Figure 5, where the Grenander estimates are plotted for the two real DNA microarray data sets which will be analyzed in Section 7. In both cases it is seen that the density estimates "flatten out" when p increases, and then suddenly drop down to a significantly smaller value for p close to 1. This "drop-down"-effect was also seen in the simulation experiment described in Section 6. It is intuitively reasonable that this should happen for a density estimate which is constrained to be decreasing.

For the reasons explained above, it does not seem like a good idea to estimate π_0 by $\hat{\pi}_0^g = \hat{f}(1)$. If we still want to base our estimator on $\hat{f}$, how should we proceed? First, as noted in the Section 4.2.1, recall that $\hat{f}$ is constant over each interval $(p_{(i)}, p_{(i+1)}]$, and of course also could be constant over longer intervals $(p_a, p_b]$, $a < b$. Let $r_1, \ldots, r_k$ denote the indices of the "change points", i.e. $p_{(r_1)}, \ldots, p_{(r_k)}$ is the set of p-values $p_{(i)}$ for which $p_{(i)} > p_{(i+1)}$.

As Figure 5 would suggest, the typical $\hat{f}$ seems to be constant over increasingly long intervals, and the sizes of the jumps between the constant level decrease. Then, for some p close to 1 there is a relatively big jump, and $\hat{f}$ is constant over one or more very short interval(s). Also, according to our mixture model $f = \pi_0 + (1 - \pi_0)h$ (where h is the density of the alternative p-values), the density should eventually settle close to the constant level π_0. These observations suggest a new estimator for π_0, based on the NPMLE decreasing density estimate $\hat{f}$. The idea of this estimator, denoted $\hat{\pi}_0^l$, is to choose the value of $\hat{f}(\cdot)$ in the region where $\hat{f} \leq 1$ for which $\hat{f}$ is constant over the longest interval $(p_{(r_{l-1})}, p_{(r_l)}]$. This "longest-length interval" estimator should be valid asymptotically, because then the p-value density will be nearly constant for large p. The simulation experiment carried out in Section 6 shows that $\hat{\pi}_0^l$ is a great improvement over $\hat{\pi}_0^g$.

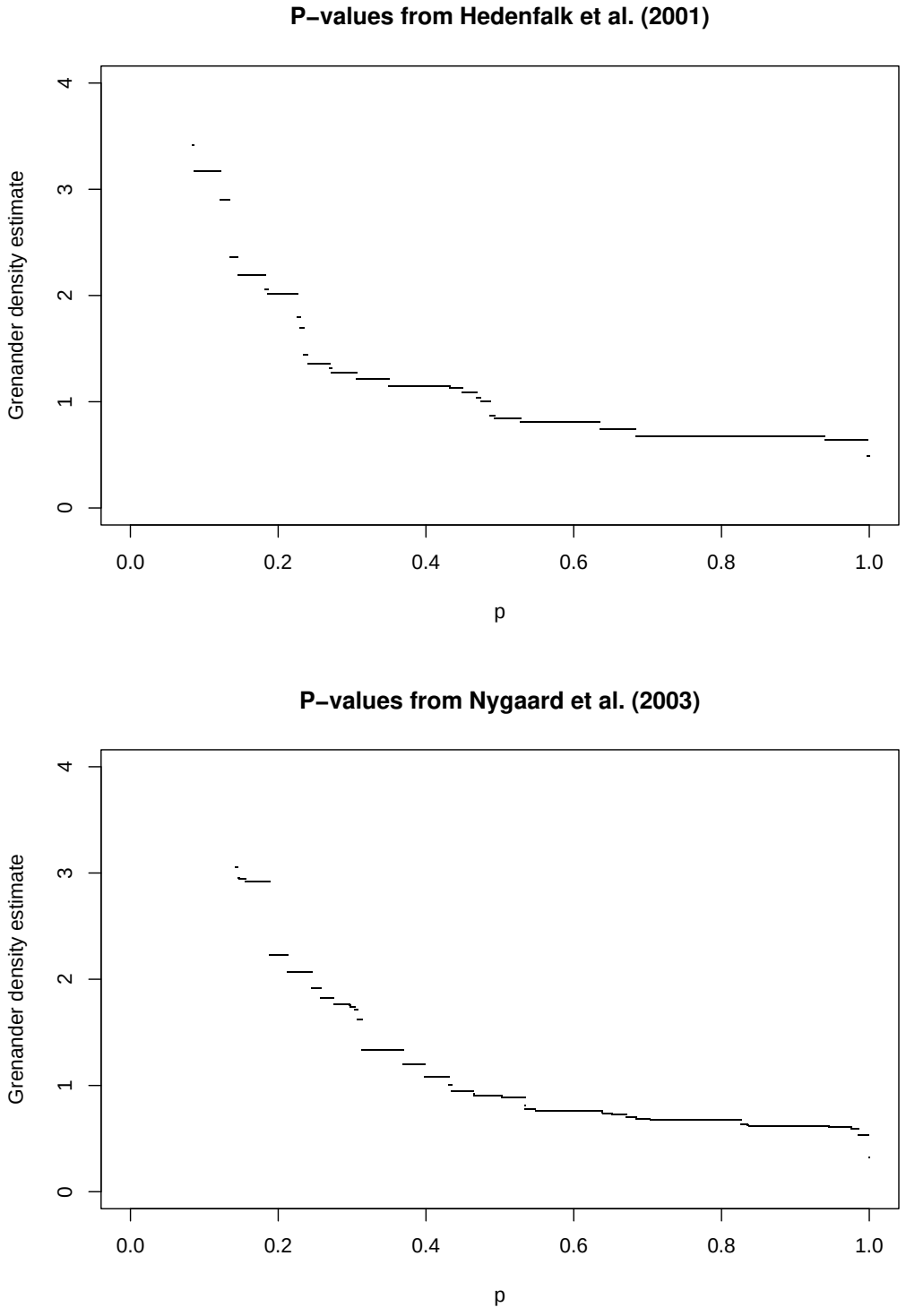


Figure 5: Grenander NPMLE decreasing density estimates for the p-values from Hedenfalk et al. (2001) and Nygaard et al. (2003).

Formally, the estimator $\hat{\pi}_0^l$ is defined as follows:

$$\hat{\pi}_0^l = \hat{f}_j; \text{ where } j = \underset{\{r_l : \hat{f}_{(r_l)} \leq 1\}}{\arg\max} \{p_{(r_l)} - p_{(r_{l-1})}\},$$

where we let $p_{(r_0)} = 0$ for ease of notation.

4.3 Estimating π_0 using convex decreasing density estimation

In this section we use a density estimate $\hat{f}^c(p)$ of the p-values, restricted to be convex and decreasing on $[0, 1]$, to estimate π_0. The motivation for the convexity assumption is the 'drop-down'-effect (underestimation of the density near $p = 1$) discussed earlier.

In this section, we will first establish a new mixture representation for f. Using this mixture representation, we are able to characterize the maximum likehood estimate for f. This characterization is then the basis for an iterative algorithm to calculate an approximate maximum likelihood estimate, which we use as the $\hat{f}^c$ for calculating $\hat{\pi}_0^c$.

We want to estimate f under the assumption that it is convex and decreasing on $[0, 1]$ (f vanishes outside this interval). For this assumption to hold, it is clearly necessary and sufficient that h is convex and decreasing on (the convex set) $[0, 1]$ — since $g \equiv 1$ is a convex function and f is a convex combination of g and h.

The strategy we will follow is to estimate f by a certain finite mixture of the uniform density on $[0, 1]$ and a collection of densities f_θ, where $\theta > 0$. Groeneboom et al.'s (2002) proof that any convex decreasing density on $[0, \infty)$ has a mixture representation is presented in Section 4.3.1. A novel extension of their result, described in Section 4.3.2, allows us to find a mixture representation for *p-value* density f. This extension is needed because f is convex only on $[0, 1]$, not on $[0, \infty)$. Based on the mixture representation, we establish necessary and sufficient conditions for an estimate $\hat{f}$ to be the maximum likehood estimate of f in Section 4.3.3. This characterization is used in Section 4.3.4 to specify an algorithm for computation of an approximate maximum likelihood estimate, which we use as our convex decreasing p-value density estimate $\hat{f}^c$.

The final estimate is then

$$\hat{\pi}_0^c = \hat{f}^c(1).$$

4.3.1 Mixture representation for convex decreasing densities on $[0, \infty)$

In Groeneboom et al. (2002) it is shown that any convex decreasing density y on $[0, \infty)$ can be written as a continuous mixture

$$y(x) = \int_0^\infty f_\theta(x)\, \gamma_y(\theta)\, \mathrm{d}\theta, \tag{4.9}$$

where γ_y is a probability density on $[0, \infty)$ and

$$f_\theta(x) = \frac{2(\theta - x)}{\theta^2} I_{(0,\theta)}(x),\ \theta > 0. \tag{4.10}$$

Here I_A is the indicator function on the set A, i.e. $I_A(x) = 1$ if $x \in A$ and equals zero otherwise. Notice that each density f_θ is a very simple triangular density, as illustrated in Figure 6.

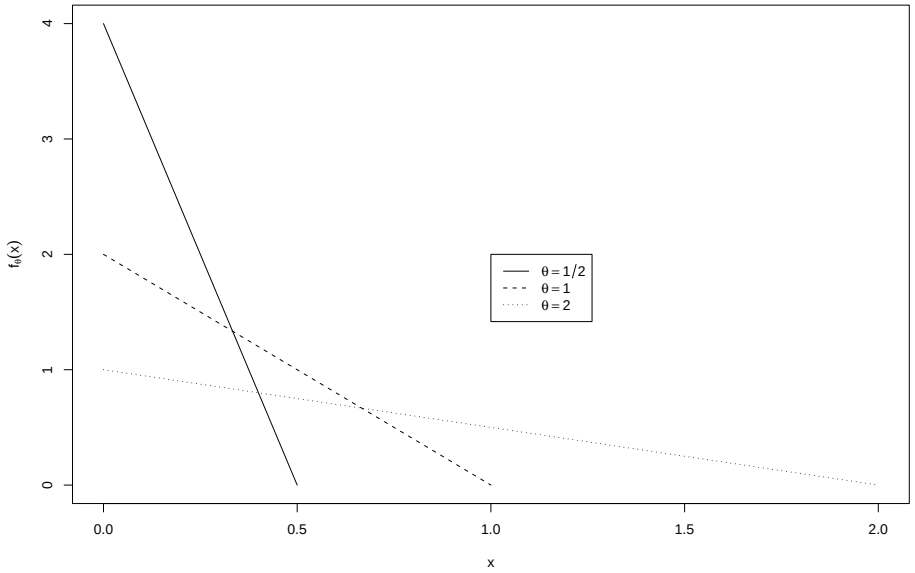


Figure 6: The mixing density $f_\theta(x)$ for different θ.

Using the class of mixing densities γ defined by

$$\gamma_y(\theta) = \frac{1}{2}\theta^2 y''(\theta) \tag{4.11}$$

we see that

$$\begin{aligned}
\int_0^\infty f_\theta(x)\gamma_y(\theta)\,\mathrm{d}\theta &= \int_x^\infty \frac{2(\theta - x)}{\theta^2}\gamma_y(\theta)\,\mathrm{d}\theta \\
&= \int_x^\infty (\theta - x) y''(\theta)\,\mathrm{d}\theta \\
&= \int_x^\infty \theta\, y''(\theta)\,\mathrm{d}\theta - x\int_x^\infty y''(\theta)\,\mathrm{d}\theta \qquad \text{(by partial integration)} \\
&= \theta\, y'(\theta)|_x^\infty - \int_x^\infty y'(\theta)\,\mathrm{d}\theta - xy'(\theta)|_x^\infty \\
&= (0 - xy'(x)) - (0 - y(x)) - (0 - xy'(x)) \\
&= y(x), \qquad (4.12)
\end{aligned}$$

which establishes Groeneboom et al.'s (2002) result.

Note that the condition that y is convex is necessary, since otherwise γ_y could be negative (and hence not a density). The demand that y be decreasing is used in the evalution of the integrals in the previous calculation.

4.3.2 Mixture representation for convex decreasing p-value densities

We want to find a mixture representation for f, the density of the p-values, but f is only convex on the interval $[0, 1]$ and not on $[0, \infty)$. Therefore, the result of Groeneboom et al.'s (2002) described in Section 4.3.1 does not apply. In this subsection we demonstrate an extension of their result, which allows us to find a mixture representation for f.

For simplicity, we start by finding a mixture representation for h, the density of the alternative p-values. We will assume that h is twice differentiable and that $h(1) = 0$. The support of h is $[0, 1]$, and $h'(x) \leq 0$ and $h''(x) \geq 0$ for all $x \in [0, 1]$. We claim that h can be written as a continuous mixture of f_θ on $[0, 1]$ with respect to the mixing density $\gamma_h = \frac{1}{2}\theta^2 h''(\theta)$ (of the class (4.11)) plus a point mass on the density $f_1(x) = 2(1 - x)$:

$$h(x) = \int_0^1 f_\theta(x)\gamma_h(\theta)\,\mathrm{d}\theta + 2(1-x)a, \text{ where } a = 1 - \int_0^1 \gamma_h(\theta)\,\mathrm{d}\theta. \tag{4.13}$$

To prove this, we examine Equation (4.13) term by term.

We first evaluate the first term in Equation (4.13) (recall that $h(1) = 0$):

$$\begin{aligned}
\int_0^1 f_\theta(x)\gamma_h(\theta)\,\mathrm{d}\theta &= \int_x^1 \frac{2(\theta - x)}{\theta^2}\frac{1}{2}\theta^2 h''(\theta)\,\mathrm{d}\theta \\
&= \int_x^1 \theta h''(\theta)\,\mathrm{d}\theta \; - x\int_x^1 h''(\theta),\mathrm{d}\theta \\
&= \theta h'(\theta)|_x^1 - \int_x^1 h'(\theta)\,\mathrm{d}\theta - x(h'(1) - h'(x)) \qquad \text{(by partial integration)} \\
&= h'(1) - xh'(x) - h(1) + h(x) - xh'(1) + xh'(x) \\
&= h(x) + (1 - x)h'(1).
\end{aligned} \tag{4.14}$$

Considering the second term in (4.13), we see that

$$\begin{aligned}
\int_0^1 \gamma_h(\theta)\,\mathrm{d}\theta &= \int_0^1 \frac{1}{2}\theta^2 h''(\theta)\,\mathrm{d}\theta \\
&= \frac{1}{2}\theta^2 h'(\theta)|_0^1 - \int_0^1 \theta h'(\theta)\,\mathrm{d}\theta \qquad \text{(by partial integration)} \\
&= \frac{1}{2}h'(1) - \int_0^1 \theta h'(\theta)\,\mathrm{d}\theta \\
&= \frac{1}{2}h'(1) - \left(\theta h(\theta)|_0^1 - \int_0^1 h(\theta)\,\mathrm{d}\theta\right) \qquad \text{(by partial integration)} \\
&= \frac{1}{2}h'(1) + 1.
\end{aligned} \tag{4.15}$$

From (4.15) it immediately follows that γ_h is a density if and only if $h'(1) = 0$. There is obviously no reason that $h'(1)$ has to be zero — and, intuitively, the reason to introduce the point mass in (4.13) is to allow that $h'(1) < 0$. The weight a in Equation (4.13) can now be found explicitely;

$$a = 1 - \int_0^1 \gamma_h(\theta)\,\mathrm{d}\theta = -\frac{1}{2}h'(1). \tag{4.16}$$

Insertion of (4.14) and (4.16) into the right side of the mixture representation in Equation (4.13) gives

$$\begin{aligned}\int_0^1 f_\theta(x)\gamma_h(\theta)\,\mathrm{d}\theta + 2(1-x)a &= h(x) + (1-x)h'(1) + 2(1-x)\left(-\frac{1}{2}h'(1)\right)\\ &= h(x),\end{aligned}$$

and the mixture representation for h is established.

Since the density f of the p-values is a mixture of the convex function $g \equiv 1$ and h, this immediately implies that f also has a mixture representation: For notational simplicity, we write $f_0(x) \equiv g$. Then

$$\begin{aligned}f(x) &= \int_0^\infty f_\theta(x)\,\mathrm{d}\mu(\theta)\\ &\equiv \pi_0 f_0(x) + (1-\pi_0)\left(\int_0^1 f_\theta(x)\gamma_h(\theta)\,\mathrm{d}\theta + f_1(x)\left(1 - \int_0^1 \gamma_h(\theta)\,\mathrm{d}\theta\right)\right),\end{aligned} \tag{4.17}$$

which is the desired mixture representation.

4.3.3 Characterization of the maximum likelihood estimate of f

Before we consider maximum likelihood estimation of f, we need to review some terminology and basic results regarding the minimization of a convex functional defined on a convex set of functions. The maximum likelihood estimate will later be defined as the minimizer of a certain convex functional, and these more general results are the basis for our estimation procedure. The following results and definitions are taken from Groeneboom et al. (2002).

Let C be a convex set of functions, and ϕ a convex functional defined on C. Consider the optimization problem

$$\text{minimize } \phi(z) \text{ for } z \in C, \tag{4.18}$$

where we assume that the minimizer exists and is unique.

For each $z \in C$ we define the *directional derivative* of z with respect to ϕ as follows. Let ψ be a function such that $z + \epsilon\psi \in C$ for some number $\epsilon > 0$. The directional derivative *in the direction* ψ is then

$$D_\phi(\psi; z) = \lim_{\epsilon \downarrow 0} \frac{\phi(z+\epsilon\psi) - \phi(z)}{\epsilon}.$$

One possible way of choosing ψ is letting $\psi = u - z$ for some $u \in C$.

Equipped with the operator D_ϕ we can characterize the solution of the minimization problem (4.18):

$$\hat{z} = \arg\min_{z \in C} \phi(z) \text{ if and only if } D_\phi(u - \hat{z}; \hat{z}) \geq 0;\ \forall u \in C. \tag{4.19}$$

This equivalence is not difficult to show. To prove necessity ($\Rightarrow$), let $\hat{z}$ be the minimum of ϕ over C and u be a function in the set C. Note that for any $\epsilon > 0$,

$$\frac{\phi(\hat{z} + \epsilon(u - \hat{z})) - \phi(\hat{z})}{\epsilon} \geq 0.$$

Taking the limit as $\epsilon \downarrow 0$ on both sides of the inequality we get that $D_\phi(u - \hat{z}; \hat{z}) \geq 0$. To show sufficiency ($\Leftarrow$), let u be an arbitrary function in C. By assumption $D_\phi(u - \hat{z}; \hat{z}) \geq 0$. Define the

function τ by $\tau(\epsilon) = \phi((1-\epsilon)\hat{f} + \epsilon u) = \phi(\hat{z} + \epsilon(u - \hat{z}))$. Clearly τ is convex, because it is the composition of the functions $(1-\epsilon)\hat{z} + \epsilon u$ (which is convex because it is a convex combination of the convex functions $\hat{z}$ and u) and ϕ (which is also convex). By convexity of τ, $\tau(1) - \tau(0) \geq \tau'(0+)$ where $\tau'(0+)$ is the right-hand derivative of τ. By the definitions of τ and D_ϕ, $\phi(u) - \phi(\hat{z}) = \tau(1) - \tau(0)$ and $\tau'(0+) = D_\phi(u - \hat{z}; \hat{z}) \geq 0$. This implies that $\phi(u) - \phi(\hat{z}) \geq 0$. Since $u \in C$ was arbitrary, it follows that $\hat{z}$ is the minimizer over C.

If the functional ϕ has the linearity property that for any probability measure μ and any mixture $u(x) = \int z_\theta(x) \mathrm{d}\mu(\theta)$ (where $z_\theta \in C$ for all $\theta \geq 0$) and any $z \in C$

$$D_\phi(u - z; z) = \int_0^\infty D_\phi(z_\theta - z; z)\, \mathrm{d}\mu(\theta), \tag{4.20}$$

then it follows from the equivalence (4.19) that

$$\hat{z} = \arg\min_{z \in C} \phi(z) \text{ if and only if } D_\phi(z_\theta - \hat{z}; \hat{z}) \geq 0;\ \forall \theta \geq 0. \tag{4.21}$$

So, the (rather strict) condition in (4.19); that $D_\phi(u - \hat{z}; \hat{z}) \geq 0$ for all $u \in C$, can be relaxed to the condition that $D_\phi(z_\theta - \hat{z}; \hat{z})$ for all $\theta \geq 0$ in this case.

This concludes the general convex optimization results that are needed. We now consider maximum likelihood estimation of some density function $v \in C$. Assume that we are given a sample $x_1, \ldots, x_m$ from v. The maximum likelihood estimate $\hat{v}$ is defined as the solution of the minimization function (4.18), where

$$\phi(v) = -\sum\nolimits' \log v(x_i), \tag{4.22}$$

and $\sum'$ denotes the sum over all i such that $v(x_i) > 0$. Note that ϕ is a negative loglikelihood, so minimizing it makes sense. Also, $\phi(v)$ is a convex functional of v, since the sum of logarithms is concave.

For the "maximum likelihood functional" $\phi(v) = -\sum' \log v(x_i)$ there is a simple expression for $D_\phi(\psi; v)$:

$$\begin{aligned}
D_\phi(\psi; v) &= \lim_{\epsilon \downarrow 0} \frac{-\sum' \log(v(x_i) + \epsilon\psi(x_i)) - (-\sum' \log v(x_i))}{\epsilon} \\
&= -\sum\nolimits' \lim_{\epsilon \downarrow 0} \frac{\log\left(1 + \epsilon(\psi(x_i)/v(x_i))\right)}{\epsilon} \\
&= -\sum\nolimits' \lim_{\epsilon \downarrow 0} \frac{\psi(x_i)/v(x_i)}{1 + \epsilon(\psi(x_i)/v(x_i))} \qquad \text{(by L'Hopital's rule)} \\
&= -\sum\nolimits' \frac{\psi(x_i)}{v(x_i)}.
\end{aligned} \tag{4.23}$$

The linearity property (4.20) holds for $\phi(v) = -\sum' \log v(x_i)$. This is seen by noting that in this case

$$\begin{aligned} D_\phi(u - v; v) &= -\sum' \frac{u(x_i) - v(x_i)}{v(x_i)} \\ &= -\sum' \frac{\int_0^\infty (v_\theta(x_i) - v(x_i))\, \mathrm{d}\mu(\theta)}{v(x_i)} \\ &= \int_0^\infty \left(-\sum' \frac{v_\theta(x_i) - v(x_i)}{v(x_i)} \right) \mathrm{d}\mu(\theta) \\ &= \int_0^\infty D_\phi(v_\theta - v; v)\, \mathrm{d}\mu(\theta). \end{aligned}$$

Given the observed p-values $p_1, \ldots, p_m$, our aim is to find an estimate $\hat{f} \in C$ of the density of the p-values. In Section 4.3.2 we proved that f can be represented by a mixture;

$$f(x) = \int_0^1 f_\theta(x)\, \mathrm{d}\mu(\theta), \tag{4.24}$$

where the mixing density is

$$f_\theta(x) = \begin{cases} I_{[0,1]}(x) & \text{if } \theta = 0, \\ \frac{2(\theta - x)}{\theta^2} I_{(0,\theta)}(x) & \text{if } \theta \in (0, 1]. \end{cases}$$

This means that we can use the linearity property (4.20) of ϕ, which again implies that the equivalence in (4.21) holds. For the mixture representation we have for f, equivalence (4.21) then implies that $\hat{f}$ is the maximum likelihood estimate if and only if

$$D_\phi(f_\theta - \hat{f}; \hat{f}) \geq 0 \text{ for all } \theta \in [0, 1]. \tag{4.25}$$

Using the expression for D_ϕ from (4.23), we see that

$$D_\phi(f_\theta - \hat{f}; \hat{f}) = \sum' \frac{\hat{f}(p_i) - f_\theta(p_i)}{\hat{f}(p_i)};\ \theta \in [0, 1].$$

Let

$$\hat{\theta} = \operatorname*{arg\,min}_{\theta \in [0,1]} D_\phi(f_\theta - \hat{f}; \hat{f}) = \operatorname*{arg\,min}_{\theta \in [0,1]} \sum' \frac{\hat{f}(p_i) - f_\theta(p_i)}{\hat{f}(p_i)}. \tag{4.26}$$

Then $\hat{f}$ is the likelihood estimate of f if and only if

$$D_\phi(f_{\hat{\theta}} - \hat{f}; \hat{f}) \geq 0 \tag{4.27}$$

or equivalently

$$\sum' \frac{\hat{f}(p_i) - f_{\hat{\theta}}(p_i)}{\hat{f}(p_i)} \geq 0. \tag{4.28}$$

4.3.4 An algorithm for calculating an approximate MLE of f

From Equation (4.28) in Section 4.3.3, we now have a characterization of the maximum likelihood estimate $\hat{f}$ in terms of the observed p-values. This result provides the basis for calculation of an *approximate* maximum likelihood estimate $\hat{f}^c$. Since our only reason for estimating f is that it provides a way to estimate π_0, a good approximation should be sufficient for our purposes.

The algorithm we present for estimation of f was proposed by Fedorov (1972) and Wynn (1970) (in a completely different context). This procedure works as follows: We first specify a convex and decreasing initial value $\hat{f}_0$ (e.g. $\hat{f}_0 = I[0,1]$). Then for $j = 0, 1, 2, \ldots$, given the current iterate $\hat{f}_j$, we determine $\hat{\theta}$ (where $\hat{f}_j$ replaces $\hat{f}$ in Equation (4.26)). If $D_\phi(f_{\hat{\theta}} - \hat{f}_j; \hat{f}_j) \geq 0$, then the current iterate $\hat{f}_j$ is optimal by (4.27) and we are done. Otherwise, the next iterate is

$$\hat{f}_{j+1} = (1 - \hat{\varepsilon})\hat{f}_j + \hat{\varepsilon} f_{\hat{\theta}}, \tag{4.29}$$

where

$$\hat{\varepsilon} = \underset{\varepsilon \in [0,1)}{\arg\min}\, \phi((1-\varepsilon)\hat{f}_j + \varepsilon f_{\hat{\theta}}) = \underset{\varepsilon \in [0,1)}{\arg\min} \left(-\sum{}' \log((1-\varepsilon)\hat{f}_j(p_i) + \varepsilon f_{\hat{\theta}}(p_i)) \right). \tag{4.30}$$

This procedure is an analogue to the "steepest descent"-algorithms used for optimizing functions on the Euclidian n-space $\mathbb{R}^n$. In each step, the next iterate is the optimal convex combination of the current iterate and the mixing density, $f_{\hat{\theta}}$, corresponding to the most negative directional derivative (which is "the best direction").

In practice, we calculate an approximate $\hat{\theta}$ by finding the $\theta \in \mathcal{T}$ which minimizes $\sum'((f_\theta(p_i) - \hat{f}_j(p_i))/\hat{f}_j(p_i))$ where $\mathcal{T}$ is a grid over $[0,1)$, e.g. $\mathcal{T} = \{0, 0.01, 0.02, \ldots, .99\}$. This reduces the problem of calculating $\hat{\theta}$ to finding the minimal element in a vector. Since the function $\tau(\varepsilon) = \phi((1-\varepsilon)\hat{f}_j + \varepsilon f_{\hat{\theta}})$ is convex, $\hat{\varepsilon}$ can be found by a bisection search. Note that

$$\tau'(\varepsilon) = -\sum{}' \frac{\mathrm{d}}{\mathrm{d}\varepsilon} \log((1-\varepsilon)\hat{f}_j(p_i) + \varepsilon f_{\hat{\theta}}(p_i)) = \sum{}' \frac{\hat{f}_j(p_i) - f_{\hat{\theta}}(p_i)}{(1-\varepsilon)\hat{f}_j(p_i) + \varepsilon f_{\hat{\theta}}(p_i)}.$$

Since τ is convex, we know that if $\tau'(0) \geq 0$ then $\hat{\varepsilon} = 0$. If this is not the case, then a proposed value ε^* is too small if $\tau'(\varepsilon^*) < 0$ and too large if $\tau'(\varepsilon^*) > 0$, so we can use a bisection search in the obvious way.

The iteration is run until $D_\phi(f_{\hat{\theta}} - \hat{f}_j; \hat{f}_j) > -\delta$ where δ is a positive accuracy parameter. In addition, we recommend to specify a maximal number k of f_θ (for $\theta > 0$) that one is willing to include in the mixture.

Let $\hat{f}$ be the last $\hat{f}_j$ which is calculated before the iteration terminates. Our estimate of π_0 is then

$$\hat{\pi}_0^c = \hat{f}(1).$$

The entire estimation procedure is formally specified as Algorithm 2 below.

4.4 Estimating π_0 using kernel density estimation

In this section we will describe an estimator based on a kernel density estimate of the p-value density $f(p)$. As for the other density estimation based estimators (except $\hat{\pi}_0^l$), the basic approach here is using the value at $p = 1$ of an estimate of f as an estimator of π_0.

Algorithm 2 Calculation of an approximate MLE of f (Fedorov/Wynn)

$p_1, \ldots, p_m$: observed p-values
$\mathcal{T}$: grid covering $[0,1]$ (e.g. $\{0, 0.01, 0.02, \ldots, 1\}$)
k : maximal number of f_θ where $\theta > 0$ to include in the mixture
$\delta, \nu > 0$: accuracy parameters
$\hat{f} \leftarrow I_{[0,1]}$
$\hat{\theta} \leftarrow \arg\min_{\theta \in \mathcal{T}} \sum{}'(\hat{f}(p_i) - f_\theta(p_i))/\hat{f}(p_i)$
$j \leftarrow 0$
while $j \leq k$ and $\sum{}'(\hat{f}(p_i) - f_{\hat{\theta}}(p_i))/\hat{f}(p_i)) \leq -\delta$ [i.e. $D_\phi(f_{\hat{\theta}} - \hat{f}_j; \hat{f}_j) \leq -\delta$] **do**
  if $\sum{}'(\hat{f}(p_i) - f_{\hat{\theta}}(p_i))/\hat{f}(p_i) \geq 0$ [i.e. $\tau'(0) \geq 0$] **then**
    $\hat{\varepsilon} \leftarrow 0$
  else
    $l \leftarrow 0; u \leftarrow 1$
    while $u - l > \nu$ **do**
      $\hat{\varepsilon} \leftarrow (l+u)/2$
      if $\sum{}'(\hat{f}(p_i) - f_{\hat{\theta}}(p_i))/((1-\hat{\varepsilon})\hat{f}(p_i) + \hat{\varepsilon} f_{\hat{\theta}}(p_i)) < 0$ [i.e. $\tau'(\hat{\varepsilon}) < 0$] **then**
        $l \leftarrow \hat{\varepsilon}$
      else
        $u \leftarrow \hat{\varepsilon}$
      end if
    end while
  end if
  if $\hat{\theta} > 0$ and $\hat{\varepsilon} > 0$ **then**
    $j \leftarrow j + 1$
  end if
  $\hat{f} \leftarrow (1-\hat{\varepsilon})\hat{f} + \hat{\varepsilon} f_{\hat{\theta}}$
  $\hat{\theta} \leftarrow \arg\min_{\theta \in \mathcal{T}} \sum{}'(\hat{f}(p_i) - f_\theta(p_i))/\hat{f}(p_i)$
end while
$\hat{f}^c \leftarrow \hat{f}$
$\hat{\pi}_0^c \leftarrow \hat{f}^c(1)$

The kernel method for density estimation is presented in e.g. Silverman (1986). Formally, it can be described as follows: Let K be a function satisfying

$$\int_{-\infty}^{\infty} K(t)\,\mathrm{d}t = 1.$$

K is called a *kernel function*. (A symmetric probability density is a common choice for K.) Given independent, identically distributed observation $x_1, \ldots, x_n$ from a density y, the kernel estimator with kernel K and *smoothing parameter* ω is then defined by

$$\hat{y}(x) = \frac{1}{n\omega} \sum_{i=1}^{n} K\left(\frac{x - x_i}{\omega}\right). \tag{4.31}$$

It turns out that the choice of kernel K is not really critical to the performance of the kernel estimator (see Silverman (1986, Section 3.3.2)). We will use the standard normal density as the kernel. This is a conventional (and convenient) choice, and is known to work well.

The choice of smoothing parameter ω is very important for the performance of the estimation. In many situations ω is determined subjectively, but for our purpose that will not do. We describe two ways of choosing ω automatically. In Section 4.4.1 we present a standard "rule of thumb" described by Silverman (1986), and in Section 4.4.2 a new method especially tailored for our particular situation is developed. The latter method is based on some of the ideas of the former, and therefore we need to consider both.

After describing the two different choices for ω, we consider the estimation of π_0 is Section 4.4.3.

4.4.1 A standard choice of the smoothing parameter ω

The development in this section is based on the book by Silverman (1986). However, we will make more restrictive assumptions to simplify the presentation. We assume that the kernel K is a symmetric probability density with zero mean and unit variance, and that the unknown density y is sufficiently smooth.

Assume that we have an iid sample $x_1, \ldots, x_n$ from y, and that we construct a kernel density estimate $\hat{y}$ with smoothing parameter ω and kernel K, as in Equation (4.31).

The main idea is to choose ω such that it minimizes the distance between $\hat{y}$ and y. As a first step towards a workable definition of "distance", we consider the mean and variance of $\hat{y}$ at a given point x. It follows easily from the definition (4.31) that

$$\begin{aligned} \mathrm{E}(\hat{y}(x)) &= \frac{1}{n\omega} \sum_{i=1}^{n} \mathrm{E}\left(K\left(\frac{x - x_i}{\omega}\right)\right) \\ &= \int_{-\infty}^{\infty} \frac{1}{\omega} K\left(\frac{x - t}{\omega}\right) y(t)\,\mathrm{d}t, \end{aligned} \tag{4.32}$$

and

$$\begin{aligned}\operatorname{Var}(\hat{y}(x)) &= \frac{1}{(n\omega)^2}\sum_{i=1}^{n}\operatorname{Var}K\left(\frac{x-x_i}{\omega}\right)\\ &= \frac{1}{(n\omega)^2}\sum_{i=1}^{n}\left(\left(\mathrm{E}\left(K\left(\frac{x-x_i}{\omega}\right)\right)^2\right)-\left(\mathrm{E}\left(K\left(\frac{x-x_i}{\omega}\right)\right)\right)^2\right)\\ &= \frac{1}{n}\left(\int_{-\infty}^{\infty}\frac{1}{\omega^2}K\left(\frac{x-t}{\omega}\right)^2 y(t)\,\mathrm{d}t-\left(\int_{-\infty}^{\infty}\frac{1}{\omega}K\left(\frac{x-t}{\omega}\right)y(t)\,\mathrm{d}t\right)^2\right). \qquad (4.33)\end{aligned}$$

The bias of $\hat{y}(x)$ is

$$\operatorname{Bias}(\hat{y}(x)) = \int_{-\infty}^{\infty}\frac{1}{\omega}K\left(\frac{x-t}{\omega}\right)y(t)\,\mathrm{d}t - y(x). \qquad (4.34)$$

At the fixed point x, the mean square error (MSE) of $\hat{y}(x)$ is

$$\operatorname{MSE}(\hat{y}(x)) = \mathrm{E}(\hat{y}(x)-y(x))^2 = \operatorname{Bias}^2(\hat{y}(x)) + \operatorname{Var}(\hat{y}(x))$$

(Casella and Berger 1990). A common measure of the accuracy of the estimate $\hat{y}$ at all points x simultaneously is the *mean integrated square error* (MISE), which is defined as

$$\begin{aligned}\operatorname{MISE}(\hat{y}) &= \mathrm{E}\left(\int_{-\infty}^{\infty}(\hat{y}(x)-y(x))^2\,\mathrm{d}x\right)\\ &= \int_{-\infty}^{\infty}\mathrm{E}(\hat{y}(x)-y(x))^2\,\mathrm{d}x\\ &= \int_{-\infty}^{\infty}\operatorname{MSE}(\hat{y}(x))\,\mathrm{d}x\\ &= \int_{-\infty}^{\infty}\operatorname{Bias}^2(\hat{y}(x))\,\mathrm{d}x + \int_{-\infty}^{\infty}\operatorname{Var}(\hat{y}(x))\,\mathrm{d}x \qquad (4.35)\end{aligned}$$

(Silverman 1986).

We now choose the value ω which minimizes an approximation to the integrated mean square error. By a Taylor series expansion,

$$y(x-\omega t) \approx f(x) - \omega t f'(x) + \frac{1}{2}\omega^2 t^2 f''(x). \qquad (4.36)$$

Using Equation (4.36) we obtain an approximation to the integrated squared bias:

$$\begin{aligned}
\int_{-\infty}^{\infty} \mathrm{Bias}^2(\hat{y}(x))\,\mathrm{d}x &= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} \frac{1}{\omega} K\left(\frac{x-t}{\omega}\right) y(t)\,\mathrm{d}t - y(x) \right)^2 \mathrm{d}x \\
&\quad \text{(by Equation 4.34)} \\
&= \int_{-\infty}^{\infty} \left(\int_{-\infty}^{\infty} K(u)(y(x-\omega u) - y(x))\,\mathrm{d}u \right)^2 \mathrm{d}x \\
&\quad \text{(by the substitution } t = x - \omega u \text{ and the fact that } K \text{ is a density)} \\
&\approx \int_{-\infty}^{\infty} \left(-\omega f'(x) \int_{-\infty}^{\infty} uK(u)\,\mathrm{d}u + \frac{1}{2}\omega^2 y''(x) \int_{-\infty}^{\infty} u^2 K(u)\,\mathrm{d}u \right)^2 \mathrm{d}x \\
&\quad \text{(by the Taylor approximation in (4.36))} \\
&= \int_{-\infty}^{\infty} \left(\frac{1}{2}\omega^2 y''(x) \right)^2 \mathrm{d}x \\
&\quad \text{(by the assumptions on } K\text{)} \\
&= \frac{1}{4}\omega^4 \int_{-\infty}^{\infty} y''(x)^2\,\mathrm{d}x. \qquad (4.37)
\end{aligned}$$

A similar calculation, again using the Taylor approximation in (4.36), yields

$$\begin{aligned}
\int_{-\infty}^{\infty} \mathrm{Var}(\hat{y}(x))\,\mathrm{d}x &= \int_{-\infty}^{\infty} \left(\frac{1}{n} \int_{-\infty}^{\infty} \frac{1}{\omega^2} K\left(\frac{x-t}{\omega}\right)^2 y(t)\,\mathrm{d}t - \frac{1}{n}\mathrm{E}^2(\hat{f}(x)) \right) \mathrm{d}x \\
&\quad \text{(by Equations (4.32) and (4.33))} \\
&\approx \int_{-\infty}^{\infty} n^{-1}\omega^{-1} y(x) \int_{-\infty}^{\infty} K(u)^2\,\mathrm{d}u\,\mathrm{d}x \\
&\quad \text{(by the Taylor approximation and the substitution } t = x - \omega u\text{)} \\
&= n^{-1}\omega^{-1} \int_{-\infty}^{\infty} K(u)^2\,\mathrm{d}u. \qquad (4.38)
\end{aligned}$$

Now, inserting Equations (4.37) and (4.38) into the expression (4.35) for the MISE, we obtain the approximation

$$\mathrm{MISE}(\hat{y}) \approx \frac{1}{4}\omega^4 \int_{-\infty}^{\infty} y''(x)^2\,\mathrm{d}x + n^{-1}\omega^{-1} \int_{-\infty}^{\infty} K(t)^2\,\mathrm{d}t. \qquad (4.39)$$

To find the $\hat{\omega}$ which minimizes the approximation for the MISE, we simply differentiate and set the result equal to zero;

$$\hat{\omega}^3 \int_{-\infty}^{\infty} y''(x)^2\,\mathrm{d}x - \hat{\omega}^{-2} n^{-1} \int_{-\infty}^{\infty} K(t)^2\,\mathrm{d}t = 0,$$

which implies that

$$\hat{\omega} = \left(\int_{-\infty}^{\infty} K(t)^2\,\mathrm{d}t \right)^{1/5} \left(\int_{-\infty}^{\infty} y''(x)^2\,\mathrm{d}x \right)^{-1/5} n^{-1/5}. \qquad (4.40)$$

It is easily checked that the second derivative is positive at the point $\hat{\omega}$, so $\hat{\omega}$ is the minimizer of the approximation for the MISE.

The expression (4.40) for $\hat{\omega}$ depends on the density y, which of course is unknown. Silverman (1986, Section 3.4.2) suggests to use a refence family of distribution, specifically, the familty of normal densities with variance σ^2, to replace y in Equation (4.40). If the true density y is near normal, this approach should work well. Using this reference family,

$$\int_{-\infty}^{\infty} y''(x)^2 \, \mathrm{d}x = \frac{3}{8}\pi^{-1/2}\sigma^{-5}. \tag{4.41}$$

Using a standard normal density as the kernel K, we similarly obtain

$$\int_{-\infty}^{\infty} K(t)^2 \, \mathrm{d}t = (4\pi)^{-1/2}. \tag{4.42}$$

(We omit the details of the evaluation of these integrals.) Inserting (4.41) and (4.42) into the expression for $\hat{\omega}$ in Equation 4.40, we obtain

$$\begin{aligned}\hat{\omega} &= (4\pi)^{-1/10}(\frac{3}{8}\pi^{-1/2})^{-1/5}\sigma n^{-1/5} \\ &= \left(\frac{4}{3}\right)^{1/5}\sigma n^{-1/5} \\ &\approx 1.06\sigma n^{-1/5}. \end{aligned} \tag{4.43}$$

Accordingly, we can choose the smoothing parameter by using the sample standard deviation s as an estimate for σ, and inserting this estimate into (4.43). However, Silverman (1986) recommends to use a more robust measure of variability, namely the interquartile range R, which is defined by

$$R = x_{.75} - x_{.25},$$

where $x_{.25}$ and $x_{.75}$ are the observed 25% and 75% quantiles of the data, respectively. If the data are really normally distributed with variance σ, then $R \approx 1.34s$ or equivalently $s \approx 0.75R$. Then, replacing σ by $0.75R$ in Equation (4.43) gives

$$\hat{\omega} = 0.79Rn^{-1/5}.$$

An even more robust estimate of spread is available by using

$$A = \min(s, R/1.34)$$

instead of σ in Equation (4.43). In addition to this, we should also consider the robustness to deviations from normality in y. It turns out (Silverman 1986, p. 47–48) that the robustness in this respect can be improved by reducing the factor 1.06 in Equation (4.43). For this reason, Silverman (1986) recommends choosing the smoothing parameter

$$\hat{\omega}_1 = 0.9An^{-1/5}. \tag{4.44}$$

This choice has become a standard reference. For example, it is the default choice of smoothing parameter of the density estimation routines in the software packages R and S-Plus.

We will use $\hat{\omega}_1$ as the first of our two choices of ω. The following section deals with the second choice.

4.4.2 An alternative choice of ω

In this section a new way of choosing ω is presented. This method is especially taylored for the problem of estimating the p-value density f at the point $p = 1$.

Assume that we have observed m p-values $p_1, \ldots, p_m$ from the density f and want to make a kernel density estimate $\hat{f}$. The choice $\hat{\omega}_1$ from Section 4.4.1 is the minimizer of the approximate mean *integrated* square error (MISE) of Equation (4.35) (a global error measure). The minimizer depends on the second derivative of f, and therefore we replace f by a reference density, namely a normal density with variance σ, in Equation (4.40).

In our opinion there are two potential drawbacks associated with using this approach when the goal is to estimate π_0 by estimating f. The first is that the global error measure (MISE) is basically irrelevant — we only need the estimate to be accurate at the value $p = 1$. The second is that the p-value density f is certainly non-normal, indeed f is neither unimodal nor positive on the whole real line.

Let us first consider the second objection. A more realistic, but still very simple reference family of densities is the $\mathrm{Beta}(c, 1)$-family. The $\mathrm{Beta}(c, 1)$-density $b(p)$ is given by

$$b(p|c) = cp^{c-1};\ 0 \leq p \leq 1, c > 0.$$

One major advantage of using this family is that the maximum likelihood estimator $\hat{c}$ of c is available in closed form: The loglikelihood is given by

$$l(c) = \sum_{i=1}^{m} \log\left(cp_i^{c-1}\right) = m \log c + (c-1) \sum_{i=1}^{m} \log p_i.$$

The likelihood equation is then

$$\frac{\partial l}{\partial c} = \frac{m}{c} + \sum_{i=1}^{m} \log p_i = 0,$$

and

$$\hat{c} = -\frac{m}{\sum_{i=1}^{m} \log p_i} \tag{4.45}$$

is the maximum likelihood estimate of c.

Now, instead of minimizing the approximate MISE of $\hat{f}$, we minimize an approximation for the mean square error at the point $p = 1$. It follows immediately from the calculations in Section 4.4.1 (simply by dropping the integration over x) that

$$\mathrm{MSE}(\hat{f}(1)) \approx \frac{1}{4}\omega^4 f''(1)^2 + m^{-1}\omega^{-1} f(1) \int_{-\infty}^{\infty} K(t)^2 \,\mathrm{d}t, \tag{4.46}$$

and that the ω which minimizes this is given by

$$\hat{\omega} = \left(\int_{-\infty}^{\infty} K(t)^2 \,\mathrm{d}t\right)^{1/5} f''(1)^{-2/5} f(1)^{1/5} m^{-1/5}. \tag{4.47}$$

For the $\mathrm{Beta}(c, 1)$-density we have $b(1|c) = c$ and $b''(1|c) = c(c-1)(c-2)$. Therefore, by replacing

f in Equation (4.47) with the $\mathrm{Beta}(c,1)$-density, the minimizing $\hat{\omega}$ would be given by

$$\begin{aligned}\hat{\omega} &= \left(\int_{-\infty}^{\infty} K(t)^2 \,\mathrm{d}t\right)^{1/5} b''(1|c)^{-2/5} f(1|c)^{1/5} m^{-1/5} \\ &= (4\pi)^{-1/10}(c(c-1)(c-2))^{-2/5}\hat{c}^{1/5}m^{-1/5} \\ &= (4\pi)^{-1/10}c^{-1/5}((c-1)(c-2))^{-2/5}m^{-1/5} \end{aligned} \tag{4.48}$$

if c was known.

It is a well known result that maximum likelihood estimators are invariant under functional transformations (Casella and Berger 1990). Therefore, we may simply plug the MLE $\hat{c}$ from Equation (4.45) into (4.48) to obtain

$$\hat{\omega}_2 = (4\pi)^{-1/10}\hat{c}^{-1/5}((\hat{c}-1)(\hat{c}-2))^{-2/5}m^{-1/5},$$

which then is the proposed choice of ω.

4.4.3 P-value reflection and estimation of π_0

To estimate π_0, we need to estimate a density $f(p)$ with bounded support ([0,1]) at the boundary $p = 1$. As is well known, the bounded domain will lead to underestimation near $p = 1$ (Silverman 1986). This is due to the fact that $\hat{f}_k$ is a density defined on the whole real line. Since this density will be positive also for $p > 1$, and obviously still has to integrate to one, this leads to deflation of the $\hat{f}_k$ near $p = 1$. In our case, a simple way to avoid this problem is to *mirror* the p-values around the point $p = 1$. This means that we augment the observed p-values $p_1, \ldots, p_m$ with the values $2 - p_1, \ldots, 2 - p_m$ and construct a kernel density estimate $\hat{f}_k^*$ based on these $2m$ p-values. This density estimate is defined on [0,2], and therefore the boundary effect should be negligable at $p = 1$. The density estimate based on the original p-values is then

$$\hat{f}_k(p) = 2\hat{f}_k^*(p) \text{ for } 0 < p \leq 1,$$

and zero otherwise. We can use either of the smoothing parameters $\hat{\omega}_1$ or $\hat{\omega}_2$ when calculating this kernel density estimate.

Our estimate of π_0 is given by

$$\hat{\pi}_0^k = \hat{f}_k(1).$$

4.5 Parametric estimation of π_0 using a Beta mixture model

The estimators described in Sections 4.1, 4.2, 4.3 and 4.4 are all based on a nonparametric estimates of the p-value density. In this section, we try the parametric approach due to Heller and Qin (2003). This is based on our usual mixture model for the density f of the p-values:

$$f = \pi_0 g + (1 - \pi_0)h = \pi_0 + (1 - \pi_0)h.$$

Heller and Qin (2003) assume that h, the density of the alternative p-values, is a member of the $\mathrm{Beta}(\xi, \theta)$ family, with $0 < \xi \leq 1$ and $\theta \leq 1$, i.e.

$$h(p) = c(\xi, \theta)p^{\xi-1}(1 - p)^{\theta-1};\ 0 < \xi \leq 1,\ \theta \geq 1,\ 0 < p \leq 1,$$

where the normalizing constant $c(\xi,\theta) = \Gamma(\xi+\theta)/(\Gamma(\xi)+\Gamma(\theta))$ where Γ denotes the gamma function.

It is easy to show that $h(p)$ is decreasing and convex:

$$\frac{\partial h}{\partial p} = c(\xi,\theta)p^{\xi-2}(1-p)^{\theta-2}\left((\xi-1)(1-p)-(\theta-1)p\right) \leq 0$$

since $\xi - 1 \leq 0$ and $\theta - 1 \geq 0$. Note that $h(p)$ is strictly decreasing unless $\xi = \theta = 1$, i.e. $h(p)$ is the uniform density. Also,

$$\frac{\partial^2 h}{\partial p^2} = c(\xi,\theta)p^{\xi-3}(1-p)^{\theta-3}((\xi-2)(\xi-1)(1-p)^2 - 2(\xi-1)(\theta-1)p(1-p) + (\theta-2)(\theta-1)p^2) \geq 0$$

since $(\xi-2)(\xi-1) \geq 0$, $(\theta-2)(\theta-1) \geq 0$ and $(\xi-1)(\theta-1) \leq 0$, so $h(p)$ is convex.

The density f of the p-values is represented by the following mixture model:

$$f(p) = \pi_0 + (1-\pi_0)c(\xi,\theta)p^{\xi-1}(1-p)^{\theta-1};\ 0 < \xi \leq 1,\ \theta \geq 1,\ 0 < p \leq 1, \tag{4.49}$$

and since h is decreasing and convex, it follows immediately that f is also decreasing and convex.

We now need to estimate the parameters π_0, ξ and θ. This can be done on the basis of the loglikelihood

$$l(\pi_0,\xi,\theta) = \sum_{i=1}^{m} \log\{\pi_0 + (1-\pi_0)c(\xi,\theta)p_i^{\xi-1}(1-p_i)^{\theta-1}\} \tag{4.50}$$

The maximum likelihood estimates of the parameters are the π_0^*, ξ^* and θ^* for which

$$l(\pi_0^*,\xi^*,\theta^*) \geq l(\pi_0,\xi,\theta) \text{ for all } 0 \leq \pi_0 \leq 1,\ 0 < \xi \leq 1,\ \theta \geq 1.$$

The values of π_0^*, ξ^* and θ^* are calculated numerically. The final estimate of π_0 is then given by

$$\hat{\pi}_0^{\beta} = \pi_0^*.$$

The task of maximizing the loglikelihood is alleviated by the observation that $l(\pi_0,\xi,\theta)$, regarded as a function of π_0 for fixed (ξ,θ), is concave: Letting $y_i = c(\xi,\theta)p_i^{\xi-1}(1-p_i)^{\theta-1}$,

$$\frac{\partial l}{\partial \pi_0} = \sum_{i=1}^{m} \frac{1-y_i}{\pi_0 + (1-\pi_0)y_i}$$

and

$$\frac{\partial^2 l}{\partial {\pi_0}^2} = \sum_{i=1}^{m} \frac{-(1-y_i)^2}{(\pi_0 + (1-\pi_0)y_i)^2} \leq 0,$$

since each term of the last sum is nonpositive. This implies that it is easy to maximize l with respect to π_0 for fixed (ξ,θ): We first check if $\partial l/\partial\pi_0|_{\pi_0=0} < 0$ or $\partial l/\partial\pi_0|_{\pi_0=1} > 0$, which would mean that the maximum is at $\pi_0 = 0$ and $\pi_0 = 1$, respectively (since l would then be monotonically decreasing/increasing). If neither is the case, then l is unimodal as a function of π_0, so a bisection search can be used to find the maximizing π_0.

Now, with π_0 fixed at the value found by bisection, any good constrained optimizition algorithm can be used to find the maximizing (ξ,θ).

There were some problems with the convergence of the algorithm for calculation of $\hat{\pi}_0^{\beta}$ in our implementation. Also, we feel that the parametric assumptions might be to strong to be reasonable for analysis of real-life data sets. For these reasons, the estimator $\hat{\pi}_0^{\beta}$ is not further considered in this work. In particular, $\hat{\pi}_0^{\beta}$ is not included in the simulation experiment in Section 6 and the application to data from DNA microarrays in Section 7

5 Dependence

All of the estimation methods considered in Section 4 are based on the assumption of independent p-values. This assumption might not hold for some applications, such as for example DNA microarrays. However, through simulation experiments using dependent data presented in Section 6 we aim to show that the estimators are relatively robust to the assumption of independence and work well also for p-values with different dependence structures. We feel that the actual performance of the estimators is more important than absolute rigour in their derivation.

5.1 Modelling dependencies in DNA microarray data

Ideally, for a DNA microarray dataset, we would want to model the specific dependence structure inherent in the particular selection of genes included in the experiment. We have considered different possible ways of modelling dependencies in microarray data. One idea is to apply the theory of *copulae*. A copula is a function which links identically distributed marginals to their multivariate joint distribution. It is seen that the dependence structure is inherent in the copula. Unfortunately, we were not able to apply this theory. It seems like the problem of estimating π_0 is too high-dimensional; the applications of copulae known to us concerns problems in only a few dimensions, not thousands as for DNA microarray data. An excellent introduction to the theory and applications of copulae is given by Nelson (1999).

Another possibility would be to specify a graphical model of the dependencies. This could be based on a priori knowledge of signalling pathways and functional relationships in the particular genetical material under study. One type of graphical model which might prove useful in this context is the so-called *vines* introduced by Bedford and Cooke (2002). Vines, which are a generalization of the more familiar Bayesian belief nets, can be used to model conditional dependencies. One could imagine specifying a vine for each functional group within the genes studied, and somehow use this formalization to obtain more accurate estimates of π_0. However, it is far from obvious how this should be done in practice, and to our knowledge, vines have not yet been applied to the analysis of DNA microarray data.

5.2 Schweder and Spjøtvoll's analysis of dependence

The problem of dependence when estimating π_0 is discussed as far back as in the paper by Schweder and Spjøtvoll (1982). Their treatment is in the context of testing for independence in 2×2 subtables of contingency tables, as well as the problem of testing for non-zero values in correlation matrices. However, the dimensionality of their examples is much less than in the situations that interest us here — the largest number of hypothesis tests they consider is $m = 136$. Schweder and Spjøtvoll's (1982) analysis of dependence is only used to assess the variance of their estimator of π_0.

5.3 Positive regression dependence

Benjamini and Yekutieli (2001) show that Simes's (1986) step-up procedure controls the FDR if the test statistics are so-called *positive regression dependent*, which the authors view as a sufficiently general assuptions to cover many situations. For further details and the definition of positive regression dependence we refer to Benjamini and Yekutieli's (2001) article.

5.4 "General" and "clumpy" dependence

Storey (2002b) describes two kinds of dependence which he calls "clumpy dependence"[4] and "general dependence". "General dependence" means that all of the p-values (or, equivalently, test statistics) are mutually dependent to some extent.

In the setting of DNA microarray data analysis, Storey (2002b) suggests that what he calls "clumpy dependence" is a more likely form of dependence. "Clumpy dependence" means that the p-values are dependent within groups, and that the p-values in any particular group are independent of all the p-values in the other groups. Considering p-values from DNA microarray experiments, Storey (2002b) mentions two reasons why this dependence structure is plausible. The first reason is a part of the biological reality, namely the fact that genes interact in (rather small) functional groups which are called pathways. This so-called co-regulation of genes results in dependent test statistics (and dependent p-values). The second reason is more of a technical issue concerning the microarray experimental situation, namely the occurence of cross-hybridization — which is what happens when two non-complementary strands of DNA hybridize in a microarray. This might occur when the genes have a similar molecular structure, which does not happen by chance, but when there is some real relationship between the genes. Therefore, also in this case the dependence should be confined within groups.

The clumpy dependence is emulated by splitting the data into G groups, generating $\frac{m}{G}$ $\mathrm{N}(0, \sigma_{clump})$-distributed random variables (where m is the number of hypothesis tests), and then adding the ith such varible to the ith observation in each group. General dependence is generated similarly, by setting $G = 1$. See Section 6 for details.???

5.5 Pairwise correlations

Dependencies among observations (observed log-ratios of gene expression in two-color DNA microarray experiments, transformed gene expressions in one colour experiments) or p-values, might be described by pairwise correlations.

Let us assume that observed values come from a multivariate Gaussian distribution with variance-covariance matrix $\mathbf{\Sigma}$. This gives us the possibility to flexibly describe a large number of correlation structures. We will here focus on grouped correlations.

To describe data with dependence structure similar to Storey's (2002b) "clumpy" dependence, let the variance-covariance matrix $\mathbf{\Sigma}$ be block diagonal, i.e. complete correlations within the gene group and independence between the gene groups. [5]

Correlated observations can be among other things be caused by co-regulated genes, spatial effects on the microarray slide and cross-hybridization. Some of these effects might be removed by proper preprocessing of the data. In theory the observation of one gene can be negatively correlated with the observation of another gene (when the first gene is up-regulated this causes the second gene to be down-regulated). With negative correlations, care must be taken to assure that the specified variance-covariance matrix is positive definite.

[4] In Storey (2002b), several results (such as the asymptotic behaviour of his estimators) are proved under the assumption of "weak dependence", of which "clumpy dependence" is a special case.

[5] As a generalization of the clumpy dependence we could let the strength of the correlations be different for each gene group (i.e. some groups with low correlation, some with moderate correlations and some with high correlations).

If the dependence is due to co-regulation of genes (e.g. from gene pathways), the group size of the genes could be in the order of tens to thousands. We believe that the strength of the correlations are low to moderate. In the simulations experiment of Section 6 group sizes of 50 and 100, and low (0.25), moderate (0.5) and high (0.75) correlations have been explored.

6 Simulation experiment

To investigate the properties of the estimators described in Section 4, we have carried out a simulation experiment. The generation of simulated data and the calculation of the estimates were both done in the language R, Ihaka and Gentleman (1996).

6.1 Testing scenario

A total of $m = 5000$ features (e.g. log intensity ratios for each gene in the case of DNA microarrays) were simulated for each of $J = 10$ inviduals (e.g. pasients, tissue samples, etc.). Let these random variables be $X_{ij};\ i = 1, \ldots, m,\ j = 1, \ldots, J$, and the corresponding realizations x_{ij}. We assume that each $\mathbf{X}_j \sim \mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. For each $i = 1, \ldots, m$ we test

$$H_{0i} : \mu_i = 0 \text{ versus } H_{1i} : \mu_i \neq 0.$$

For each i, a two-sided p-value p_i is then calculated on the basis of a one-sample t-test:

$$p_i = 2 \cdot \mathrm{Prob}\left(T_{J-1} \geq \left|\bar{x}_i / \sqrt{s_i/J}\right|\right), \tag{6.1}$$

where $\bar{x}_i = \sum_{j=1}^{J} x_{ij}/J$ and $s_i = \sum_{j=1}^{J}(x_{ij} - \bar{x}_i)^2/(J-1)$ are the sample mean and variance, respectively, and T_{J-1} is a t-distributed random variable with $J-1$ degrees of freedom. (See e.g. Casella and Berger (1990) for details on the one-sample t-test.)

6.2 Generation of simulated data

For the generation of simulated data four different choices of π_0 were considered, namely 0.5, 0.8, 0.9 and 0.95. For each π_0, we first drew the number of true null hypotheses m_0 from the appropriate binomial distribution, i.e. $m_0 \sim \mathrm{Bin}(m, \pi_0)$.

Secondly, a vector of expected values, $\boldsymbol{\mu} = (\mu_1, ..., \mu_m)$, was constructed. The expected values for the true null hypotheses were set to 0, $\mu_1 = \mu_2 = \cdots = \mu_{m_0} = 0$. Then the expected values for the false null hypotheses, $\mu_{m_0+1}, ..., \mu_m$, were drawn from the symmetric bi-triangular density $\kappa(\cdot|a, b)$ given by

$$\kappa(\Delta|a,b) = \begin{cases} \frac{\Delta+b}{(b-a)(b-1)} & \text{if } -b \leq \Delta < -1, \\ \frac{\Delta+a}{(b-a)(a-1)} & \text{if } -1 \leq \Delta \leq -a, \\ \frac{\Delta+a}{(b-a)(1-a)} & \text{if } \quad a \leq \Delta < 1, \\ \frac{\Delta+b}{(b-a)(1-b)} & \text{if } \quad 1 \leq \Delta \leq b, \\ 0 & \text{otherwise}, \end{cases}$$

where $a < 1 < b$. The values $a = \log_2(1.2)$ and $b = \log_2(6)$ were chosen. This particular choice of a and b is motivated by the case of DNA microarray data, where the measurements are often $\log_2$-transformed spot intensities; these values for a and b seem reasonable in this situation. The graph of the density κ is shown in Figure 7. Samples from κ were generated in two steps: In the first step Δ was determined to be positive or negative, each with probability $\frac{1}{2}$. Then, in the second step, inversion sampling was used to draw from the part of the (appropiately rescaled) density κ where $\Delta > 0$ or $\Delta < 0$, respectively. It is straightforward to calculate the cumulative distribution function $\mathcal{K}$ of this density, and by the probability unit transform $\mathcal{K}^{-1}(u)$ is a sample from κ if $u \sim \mathrm{Unif}[0, 1]$.

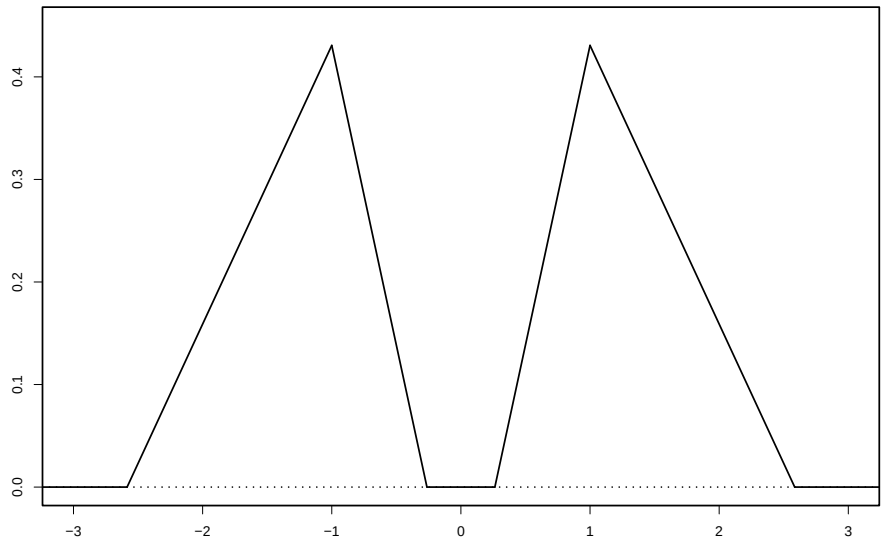


Figure 7: Density of Δ

Thirdly, a block diagonal covariance matrix, $\mathbf{\Sigma}$, was constructed. The variance of the logratio data for each gene were set to 1. Genes were separated into groups of size g (values 50 and 100 selected). Correlations betweens the groups of genes were set to 0 (independence between groups), and correlations within groups were set to ρ. Values $\rho = \{0, 0.25, 0.5, 0.75\}$ where explored in separate experiments. See Figure 8 for a sketch of the covariance matrix with group size 100.

The 5000-dimensional vectors, $\mathbf{X}_j$, $j = 1, ..., 10$ were then drawn independently from the multivariate Gaussian distribution $N(\boldsymbol{\mu}, \mathbf{\Sigma})$, and finally p-values were calculated using Equation (6.1).

For each $\pi_0 \in \{0.5, 0.8, 0.9, 0.95\}$, group sizes $g \in \{50, 100\}$, and correlations $\rho \in \{0, 0.25, 0.5, 0.75\}$, a total of $N = 1000$ sets of p-values were calculated. Then, each of the estimation procedures presented in Section 4 was applied to each set of p-values. Note that this simulation experiment is on a quite large scale. We needed to generate $4 \times 2 \times 3 + 4 = 28$ different 5000×1000 matrices of p-values. Generation of data and evaluation of all estimation procedures took approximately 16 hrs for each of the 28 situations, giving a total time of compuation of 19 days (on a PC running FreeBSD 4.8-RELEASE, CPU: Intel(R) Pentium(R) 4 CPU 2.40GHz).

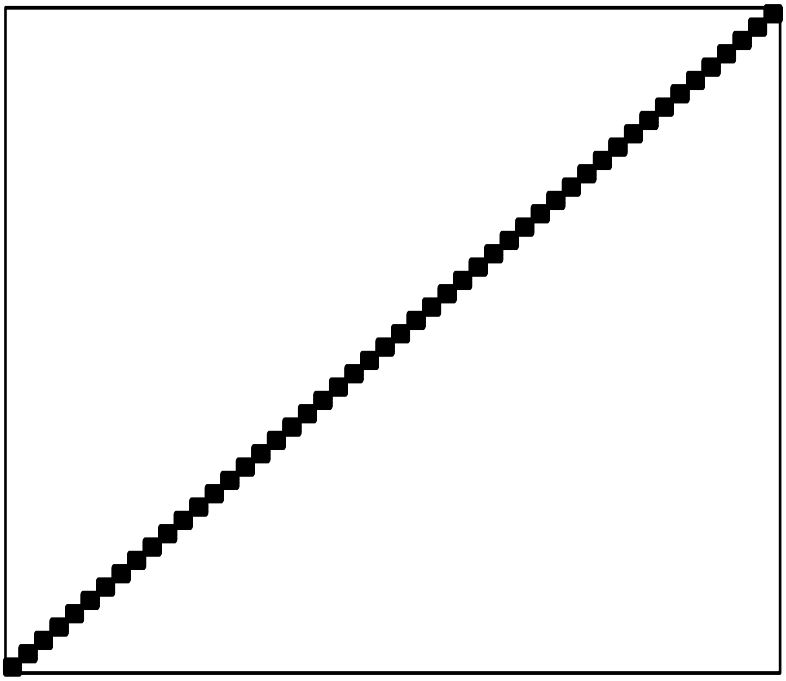


Figure 8: Sketch of the covariance matrix with group size 100 for the simulation experiment.The black squares denotes the non-null entries.

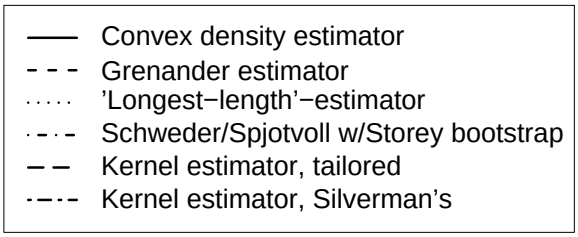


Figure 9: Plotting symbols for figures.

6.3 Results of simulation study

The results from the simulation study is a set of estimates of π_0, $\hat{\pi}_0^{(1)}, \ldots, \hat{\pi}_0^{(N)}$, for each $\pi_0 \in \{0.5, 0.8, 0.9, 0.95\}$, group sizes $g \in \{50, 100\}$, and correlations $\rho \in \{0, 0.25, 0.5, 0.75\}$ for each of the following estimators:

- "Convex": the estimator $\hat{\pi}_0^c$ described in Section 4.3.
- "Grenander": the estimator $\hat{\pi}_0^g$ from Section 4.2.2.
- "Longest-length": the estimator $\hat{\pi}_0^l$ was introduced in Section 4.2.3.
- "Kernel, tailored" is the kernel density estimation based method using the choice of smoothing parameter developed in Section 4.4.2. Estimated values are given as the minimum of the value provided by the density estimation and 1.
- "Kernel, Silverman" uses the Silverman's rule of thumb, explained in Section 4.4.1. Estimated values are given as the minimum of the value provided by the density estimation and 1.
- "SchSpjSto" denotes Schweder and Spjøtvoll's (1982) estimator $\widehat{\pi}_0(\lambda)$ with Storey's (2002b) choice of tuning parameter λ, described in Section 4.1.

For each estimator and each of the 28 sets of $N = 1000$ p-values, the minimum, maximum, first and third quartiles, median, bias, standard deviation and root mean square error (RMSE) are reported. The RMSE combines the errors from bias and variance in a natural way. Given the empirical standard deviation s and the empirical bias $b = \sum_{n=1}^{N} \hat{\pi}_0^{(n)}/N - \pi_0$ of an estimator $\hat{\pi}_0$, the RMSE is given by

$$\mathrm{RMSE}(\hat{\pi}_0) = \sqrt{s^2 + b^2}.$$

In Table 2 the summary statistics for $\pi_0 = 0.9$ are presented, and in Tables 4-6 results for π_0 equal to 0.5, 0.8, and 0.95 are found. The summary statistics are also visualized in plots. Separate plots are produced for the two group sizes and each value of π_0 as functions of the correlation. Plots of bias are found in Figure 10, and of RMSE in Figure 11. The remaining plots (median, 1st quantile, 3rd quantile and standard deviation) are found in Figures 21-24 in Appendix A.

Density estimates of the estimates based on independent data is found in Figures 12. For $\pi_0 = 0.9$ density estimates for correlated data are displayed in Figures 13 for group size 50 and in Figure 14 for group size 100. Corresponding density estimates for correlated data for the other values of π_0, are given in Figures 25-30 in Appendix A. Plotting symbols for all figures are given in the figure cation of each figure and are shown in Figure 9.

Comparison of the simulated estimates on a case-by-case basis may also reveal patterns and interesting features. Since all of the estimation methods were tried on the same 1000 data sets of p-values for each π_0 and each dependence structure, it makes sense to the ith simulated value from each estimator against the ith simulated value from each of the other estimators, for $i = 1, \ldots, 1000$. This is shown in Figure 15 for the case of independence with $\pi_0 = 0.9$.

Indep.:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.8405	0.8901	0.8998	0.8984	0.9087	0.9266	0.0137	0.0138
Grenander	0.1273	0.4948	0.6644	0.6322	0.791	0.9196	0.1856	0.3258
'Longest-length'	0.85	0.8985	0.909	0.9095	0.9195	0.9751	0.018	0.0204
Kernel, tailored	0.8403	0.8913	0.905	0.9049	0.9185	0.9613	0.0194	0.02
Kernel, Silverman	0.8017	0.881	0.9038	0.9043	0.9284	1	0.0324	0.0326
'SchSpjSto'	0.68	0.86	0.8868	0.8767	0.9022	0.9262	0.0344	0.0416
Group 50 Corr 0.25:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.816	0.8842	0.9	0.8971	0.9128	0.9435	0.0207	0.0209
Grenander	0.1531	0.4852	0.64	0.6202	0.7704	0.9435	0.1883	0.3372
'Longest-length'	0.8295	0.897	0.9103	0.9093	0.923	0.99	0.0217	0.0236
Kernel, tailored	0.8278	0.8863	0.9041	0.9043	0.9212	0.9876	0.026	0.0264
Kernel, Silverman	0.8071	0.878	0.904	0.9038	0.9283	1	0.0366	0.0368
'SchSpjSto'	0.73	0.8554	0.8835	0.8767	0.9046	0.9433	0.0361	0.0429
Group 50 Corr 0.50:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.7603	0.8717	0.897	0.8919	0.9184	0.9501	0.0335	0.0344
Grenander	0.1362	0.4838	0.6548	0.6268	0.7862	0.9412	0.1933	0.3347
'Longest-length'	0.7667	0.8876	0.9103	0.905	0.9273	0.9866	0.0305	0.031
Kernel, tailored	0.776	0.8761	0.9027	0.9037	0.9327	1	0.0422	0.0424
Kernel, Silverman	0.7501	0.8688	0.9024	0.9027	0.9375	1	0.0509	0.0509
'SchSpjSto'	0.6971	0.8436	0.8794	0.873	0.9111	0.9498	0.0473	0.0545
Group 50 Corr 0.75:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.6569	0.8506	0.8918	0.8802	0.9204	0.9537	0.051	0.0547
Grenander	0.1241	0.4803	0.6326	0.6178	0.7794	0.9484	0.1968	0.344
'Longest-length'	0.7224	0.8771	0.9086	0.9001	0.9311	0.9997	0.0447	0.0447
Kernel, tailored	0.6966	0.8558	0.9009	0.8996	0.9459	1	0.0628	0.0628
Kernel, Silverman	0.6558	0.8467	0.8966	0.8965	0.953	1	0.0707	0.0707
'SchSpjSto'	0.585	0.8249	0.876	0.863	0.9136	0.9545	0.0618	0.072
Group 100 Corr 0.25:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.7855	0.8838	0.9002	0.8969	0.9154	0.9459	0.0246	0.0248
Grenander	0.157	0.4941	0.6357	0.6201	0.7718	0.9266	0.1853	0.3356
'Longest-length'	0.8074	0.8946	0.9126	0.9087	0.9251	0.9845	0.0237	0.0252
Kernel, tailored	0.7992	0.8858	0.906	0.9053	0.926	0.9966	0.0302	0.0307
Kernel, Silverman	0.7655	0.8772	0.9039	0.9039	0.929	1	0.0392	0.0394
'SchSpjSto'	0.7067	0.854	0.8845	0.8761	0.906	0.9466	0.0397	0.0463
Group 100 Corr 0.50:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.7222	0.8637	0.898	0.8891	0.9232	0.9518	0.0423	0.0437
Grenander	0.1048	0.4918	0.6643	0.6297	0.7862	0.9405	0.1917	0.3314
'Longest-length'	0.7405	0.8807	0.9112	0.9028	0.9298	0.9936	0.0368	0.0369
Kernel, tailored	0.7418	0.8694	0.9042	0.9044	0.9441	1	0.052	0.0521
Kernel, Silverman	0.7175	0.8622	0.9038	0.9026	0.9462	1	0.059	0.0591
'SchSpjSto'	0.664	0.8371	0.8809	0.8718	0.9182	0.9507	0.0539	0.0609
Group 100 Corr 0.75:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.6442	0.8392	0.8908	0.8742	0.9262	0.9573	0.0649	0.0699
Grenander	0.1286	0.4806	0.6476	0.6274	0.7867	0.9517	0.1941	0.3347
'Longest-length'	0.6405	0.8627	0.9081	0.8924	0.9339	0.9984	0.0586	0.0591
Kernel, tailored	0.6784	0.8453	0.9062	0.899	0.9672	1	0.078	0.078
Kernel, Silverman	0.6585	0.8381	0.9068	0.8963	0.975	1	0.0851	0.0852
'SchSpjSto'	0.604	0.8086	0.8759	0.8583	0.9192	0.9579	0.0732	0.0842

Table 2: Summary statistics for set of estimates, $\hat{\pi}_0$, for $\pi_0 = 0.9$.

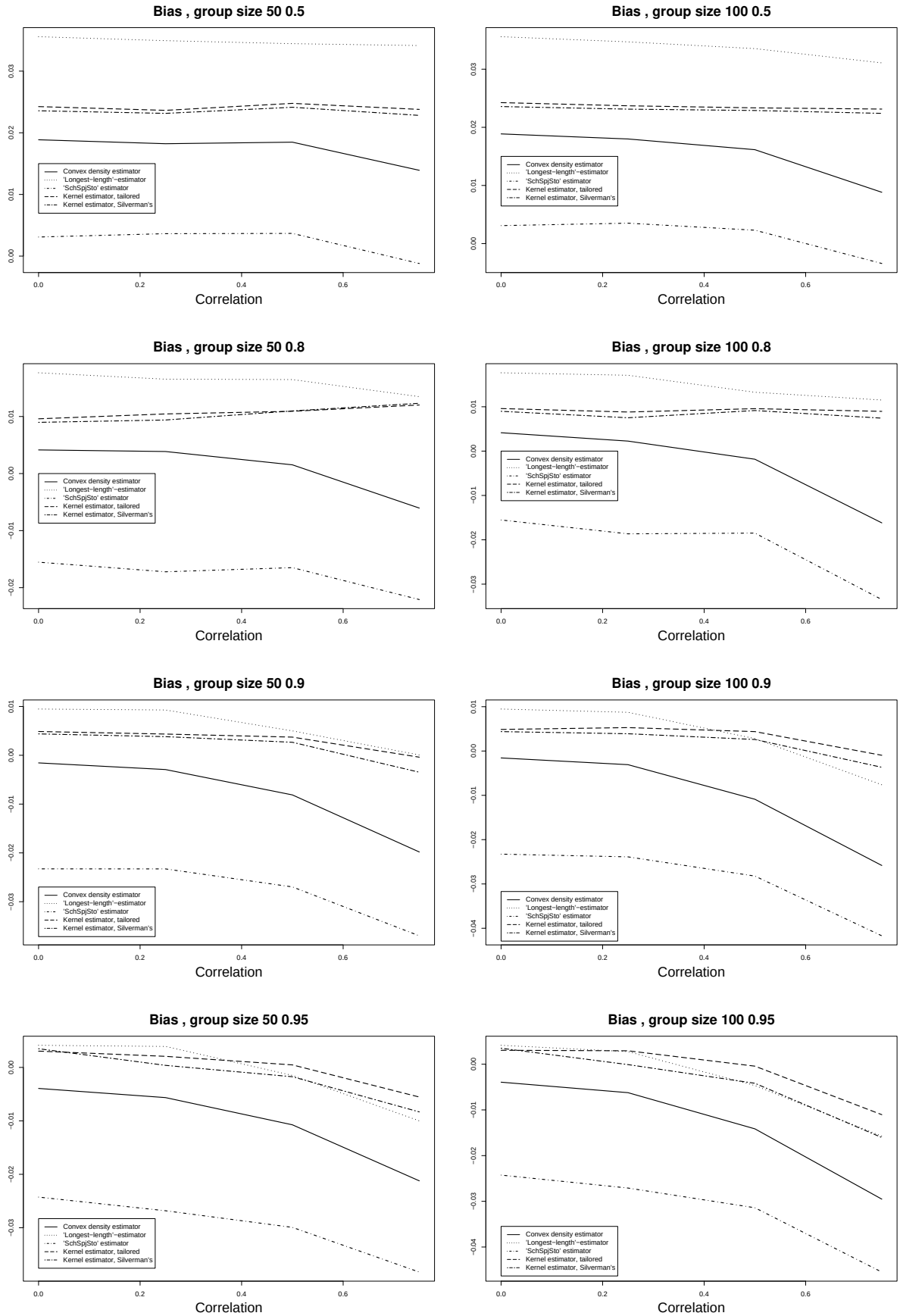


Figure 10: Bias, $b = \sum_{n=1}^{N} \hat{\pi}_0^{(n)}/N - \pi_0$ of each estimator $\hat{\pi}_0$ in data sets of N=1000, as a function of correlation for group sizes 50 and 100 and values of $\pi_0 \in \{0.5, 0.8, 0.9, 0.95\}$ for five of the six methods considere (the Grenander estimator is excluded from the plot due to high negative bias).

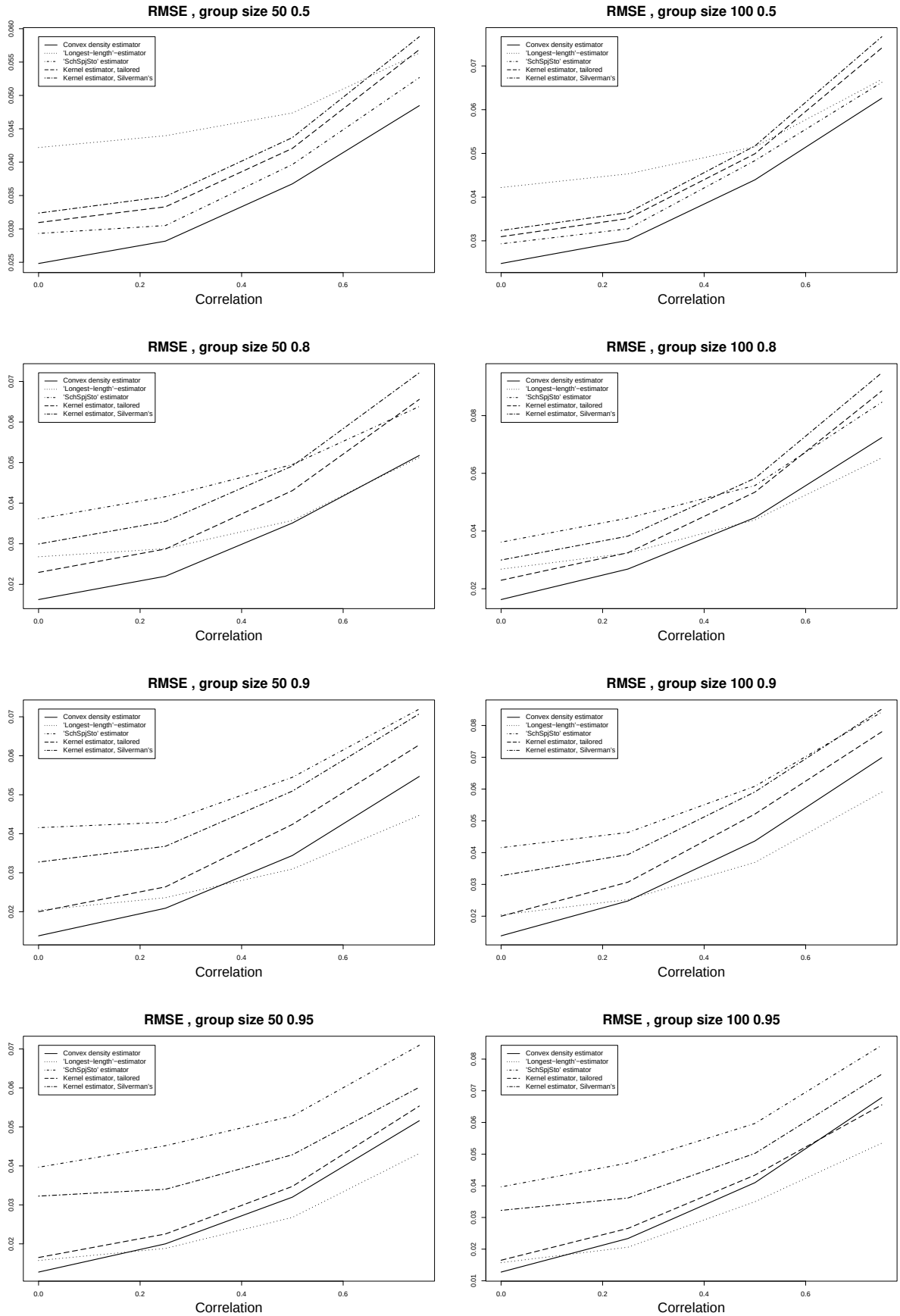


Figure 11: Root mean square error (RMSE), in data sets of N=1000, as a function of correlation for group sizes 50 and 100 and values of $\pi_0 \in \{0.5, 0.8, 0.9, 0.95\}$ for for five of the six methods considere (the Grenander estimator is excluded from the plot due to high RMSE).

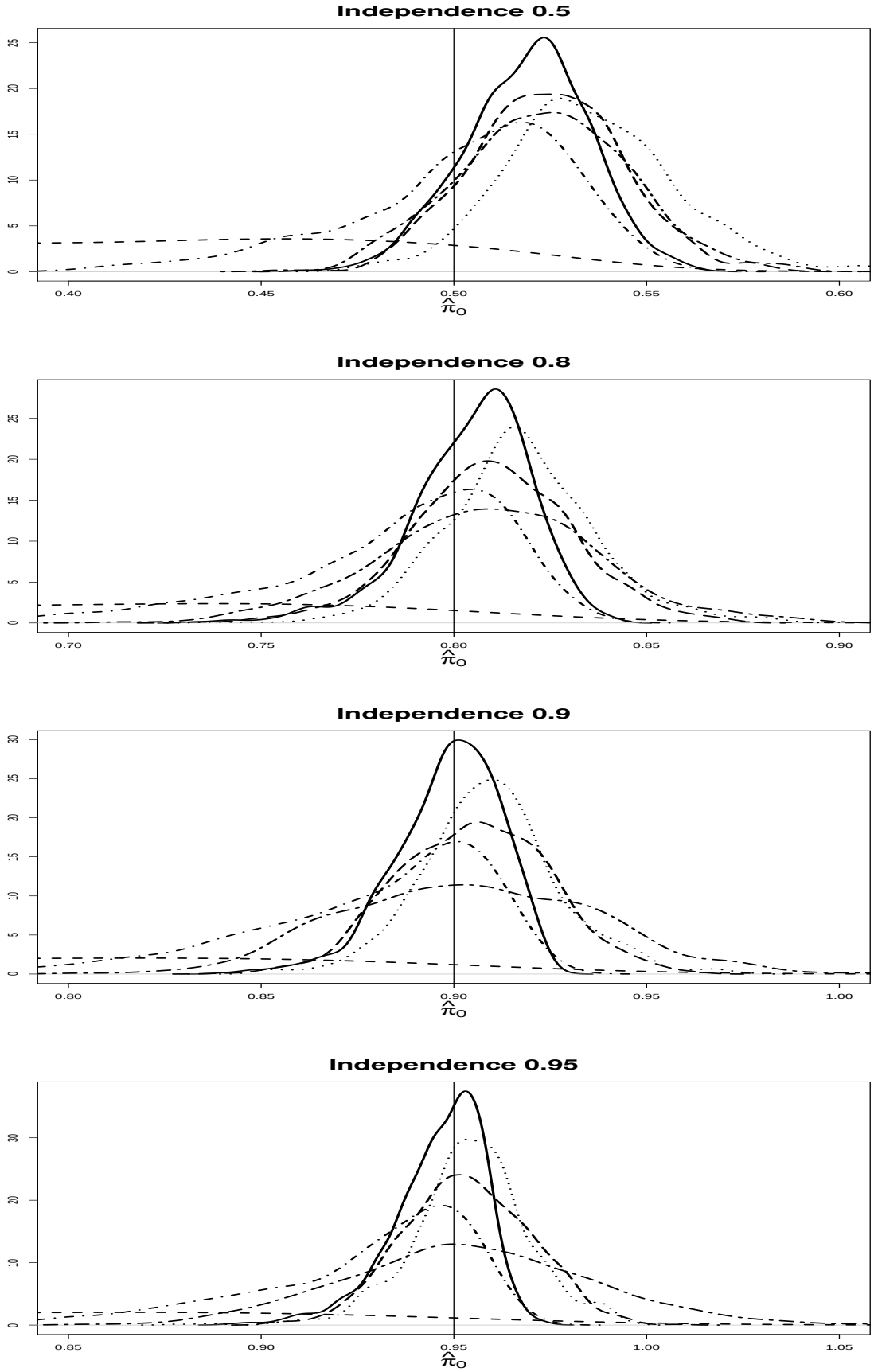
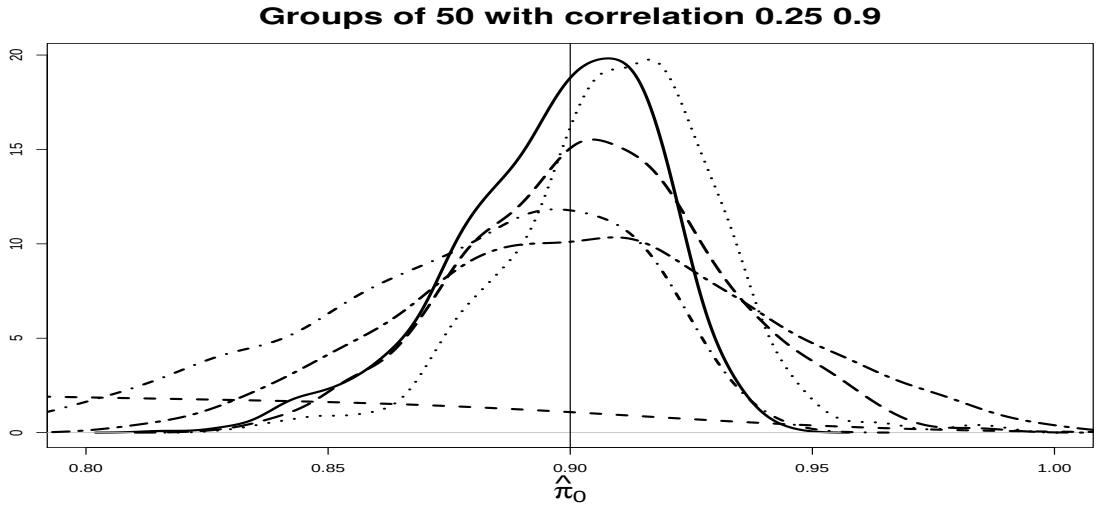
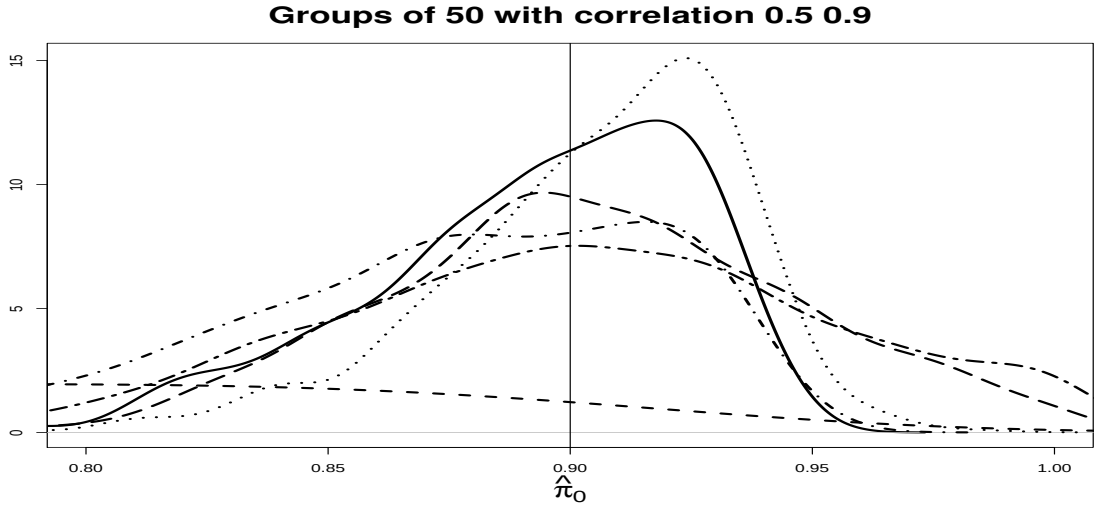


Figure 12: Density estimates of $\hat{\pi}_0$ for independent data and $\pi_0 = \{0.5, 0.8, 0.9, 0.95\}$."Convex" is solid, "Grenander" is dashed, "Longest-length" is dotted, "SchSpjSto" is dotdash "Kernel, tailord" is longdash and "Kernel, Silverman" is twodash (plotting symbols are shown in Figure 9).





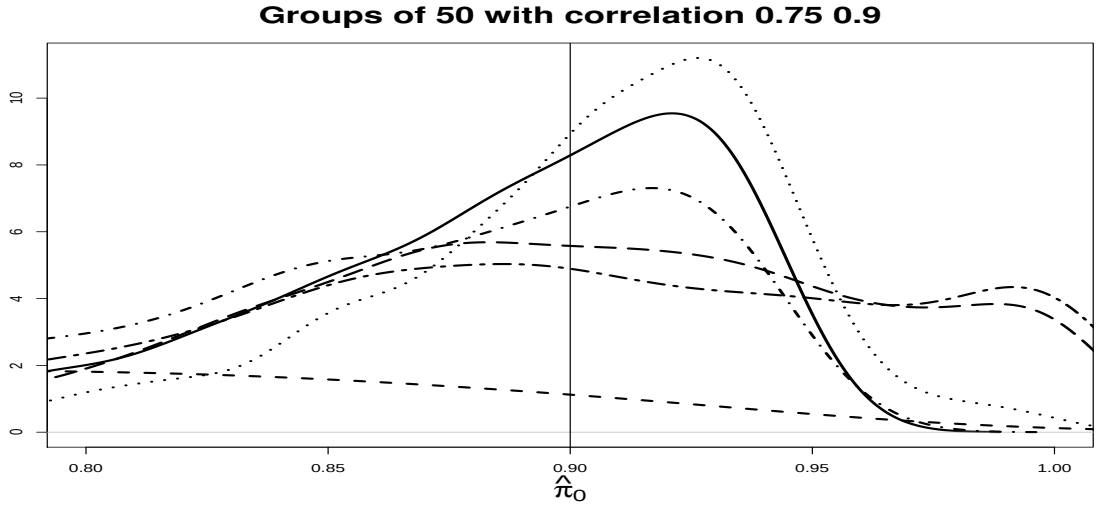
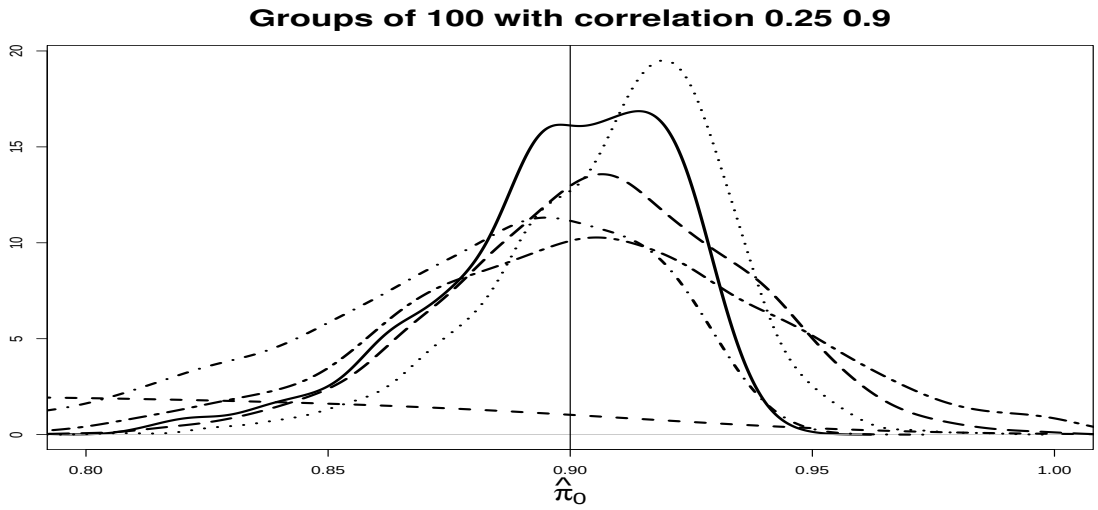
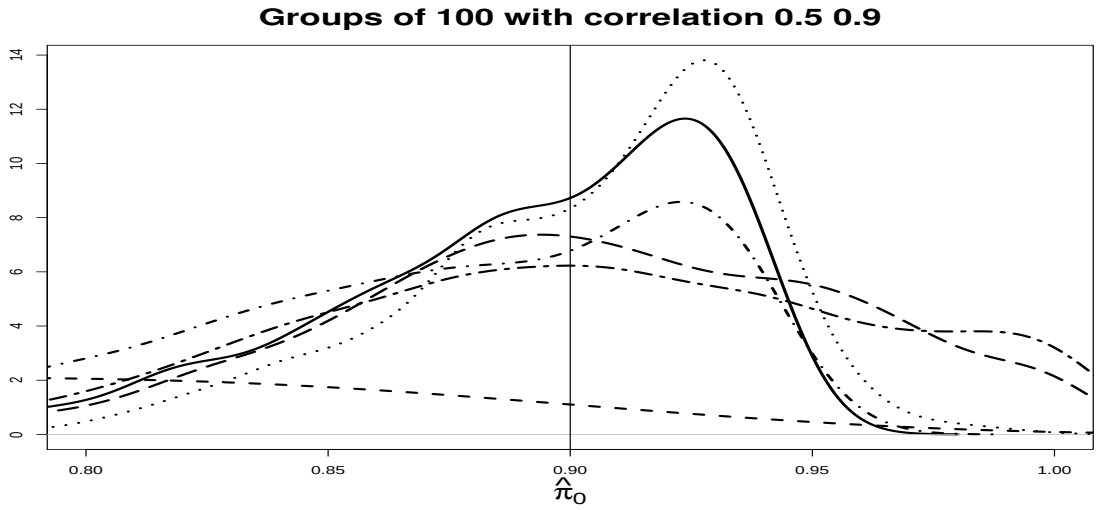


Figure 13: Density estimates of $\hat{\pi}_0$ for group size 50 for π_0=0.9. "Convex" is solid, "Grenander" is dashed, "Longest-length" is dotted, "SchSpjSto" is dotdash "Kernel, tailord" is longdash and "Kernel, Silverman" is twodash (plotting symbols are shown in Figure 9).





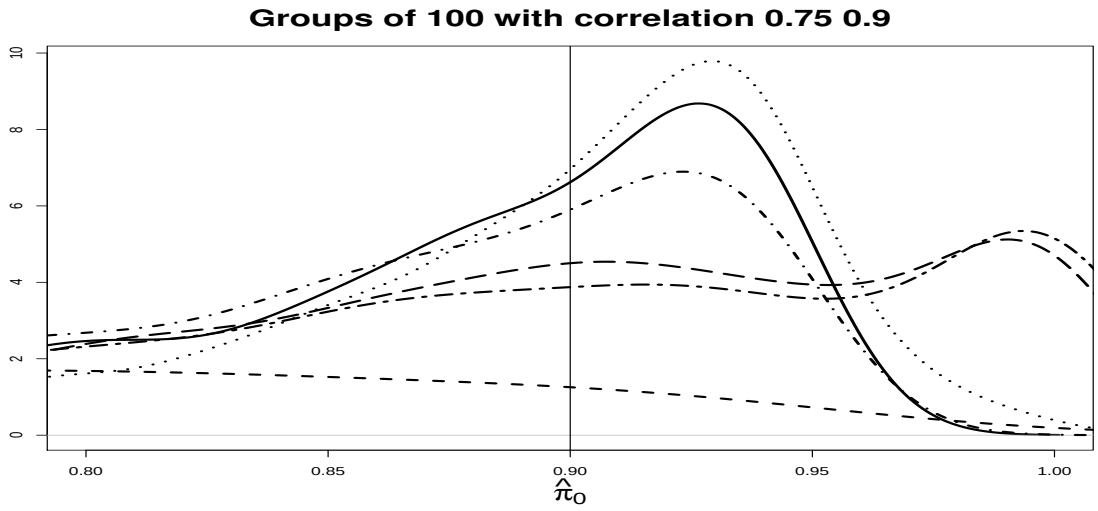


Figure 14: Density estimates of $\hat{\pi}_0$ for group size 100 for π_0=0.9. "Convex" is solid, "Grenander" is dashed, "Longest-length" is dotted, "SchSpjSto" is dotdash "Kernel, tailord" is longdash and "Kernel, Silverman" is twodash (plotting symbols are shown in Figure 9).

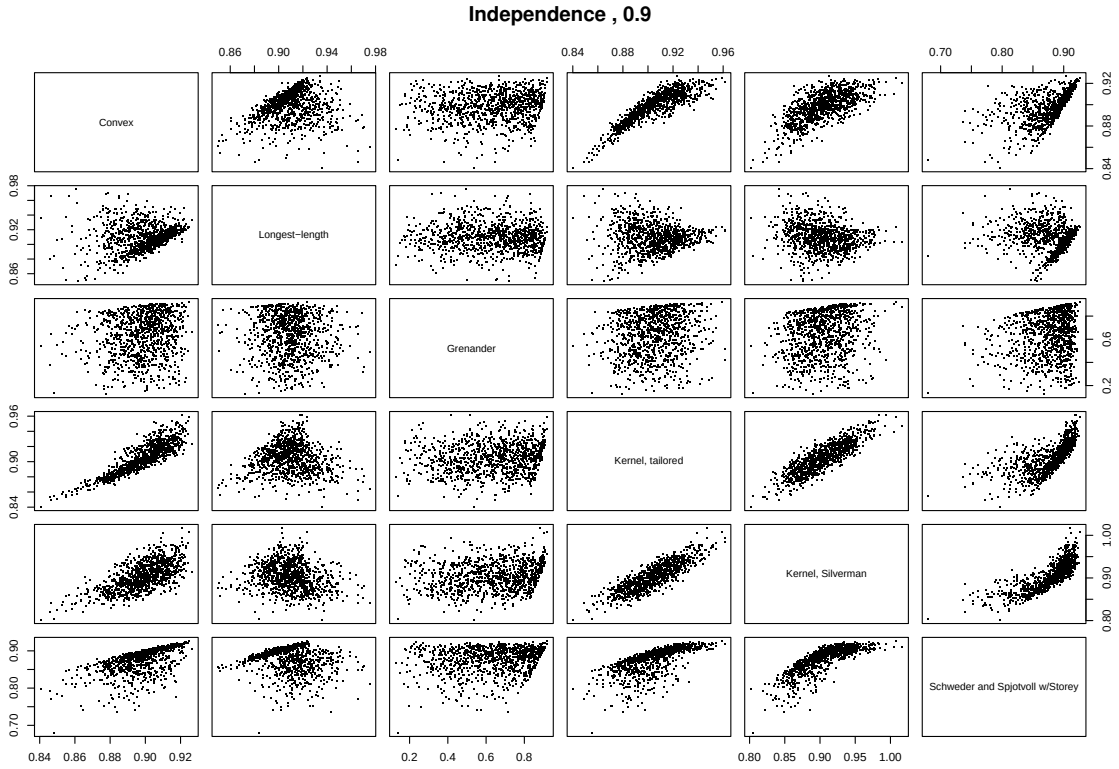


Figure 15: Comparison of the different estimators based on the simulated data, for $\pi_0 = 0.9$, independence.

6.4 Interpretation of the results

The generation of simulated data has resulted in 28 000 sets of 5000 p-values (1000 sets of 5000 p-values each for the 28 situations studied). Looking at a selection of these histograms we have attempted a crude division into three groups. For all situations considered (as a function of the value of the p-values) the histogram of p-values first decreases as the number of false null hypothesese decreases. The position of the end of this phase is dependent on the distribution of the false p-values. The second phase of the histogram can be described as constant, increasing or decreasing, and can be overlapping the first phase.

Constant This is the typical situation for independent data, and for many of the data sets for dependent data. See the second row of Figure 16.

Increasing After the fist decreasing phase, the histogram increases. See the first row of Figure 16.

Decreasing The histogram is decreasing for the whole interval. See the third row of Figure 16.

In our simulation experiment, the existence of the "increasing" and "decreasing" groups are not due to π_0 being unidentiafiable, but an effect introduced by the correlation between p-values.

From the histograms in Figure 16 we see that the form of the histograms are governed by the distribution of the true null hypothesis (since the situation in these plots are for π_0=0.95, but in general we often come across the situation where the number of true null hypotheses are much higher than the number of false null hypotheses). Let us look at histograms for the data from the true null hypotheses generated in our simulations experiment for π_0=0.95. In Figure 17 we have randomly chosen 16 data sets of p-values from the true null hypotheses. First we look at the case of independent data. In the upper (4×4) plots data are generated under independence. We see that these histograms do not show any clear increasing or decreasing trend. Then we look at dependent data; the lower (4×4) plots of Figure 17 are based on dependent data with group size 100 and correlation 0.75. We see that some of these histograms show a clear decreasing or increasing trend.

All estimators of π_0 presented in Section **??** are developed under the assumption of independence between p-values, but different restrictions have in addition been made for the estimators (e.g. decreasing, convex distribution of p-values). Still, many of the estimators are found to perform good both on independent and on dependent data. Let us first look at results for independent data and then for dependent data.

Independent data

First we look at the mean over the 1000 data sets for each value of π_0. The Grenander estimator always has the lowest value of the mean, followed by the Schweder and Spjøtvoll estimator with Storey's bootstrap routine ('SchSpjSto'), the 'Convex' estimator, the two kernel estimators and finally the 'Longest-length' estimator. The same ranking is valid for the 1st quantile and the median. For the 3rd quantile the ranking is the same, exept for $\pi_0 = 0.9$ and 0.95, where the 'Longest-length' estimator is below the kernel estimators.

For $\pi_0 = 0.5$ the 'SchSpjSto'-estimator has the smallest bias (relative to the mean), for $\pi_0 = 0.8$ and 0.9 the 'Convex'-estimator has the smallest bias, and for $\pi_0 = 0.95$ the two kernel estimators have the smallest bias. The "Convex" estimator clearly has the smallest standard deviation and RMSE of the

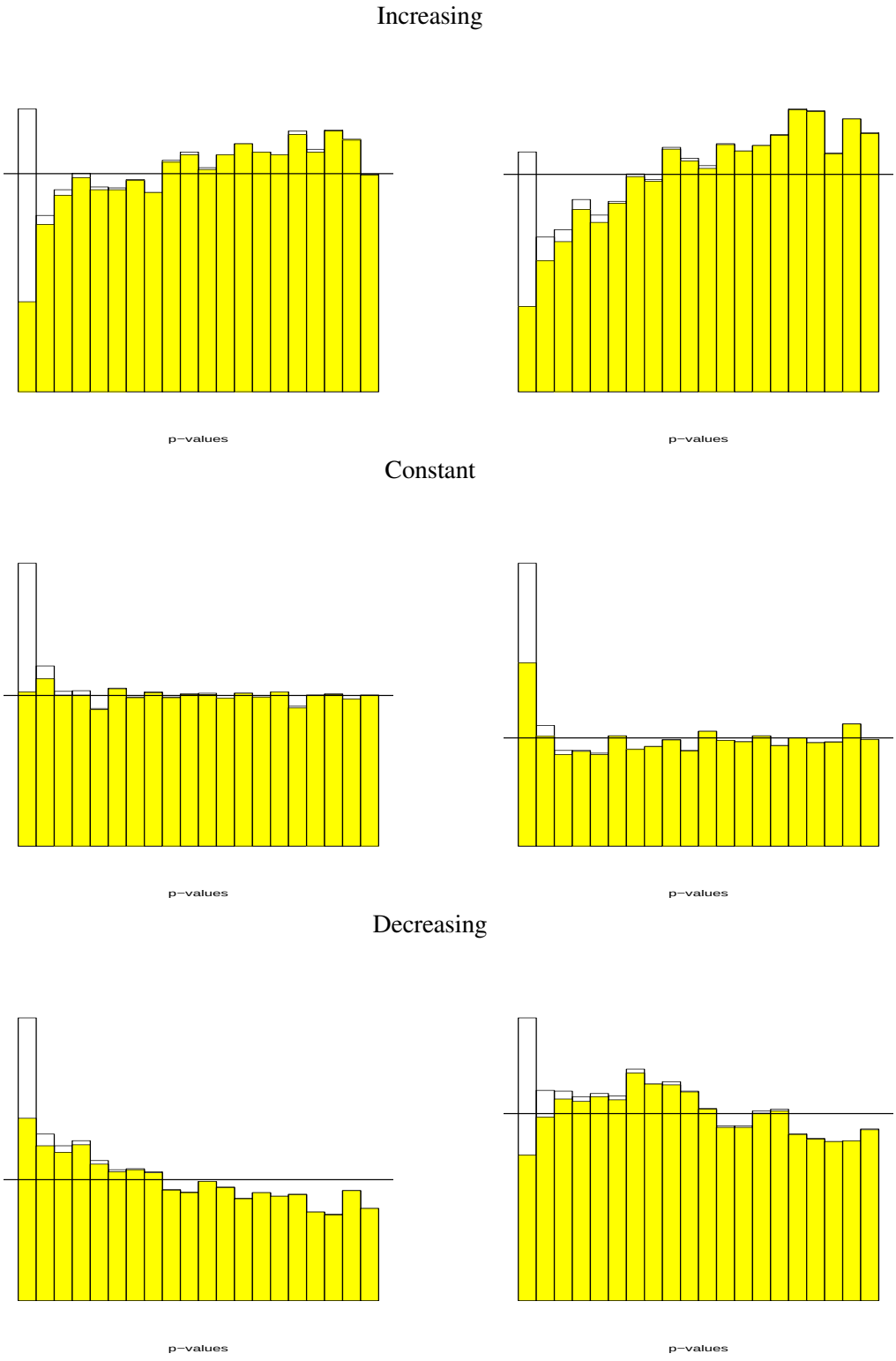


Figure 16: Typical examples of histograms of p-values. In the first row the histograms are "increasing", in the second row the histograms are "constant", and in the third row the histograms are "decreasing". The shaded areas reflects the portion of true null hypotheses and the transparent areas the portion of false null hypotheses. All examples are taken from the situation where π_0=0.95, group size is 100 and correlation is 0.75.

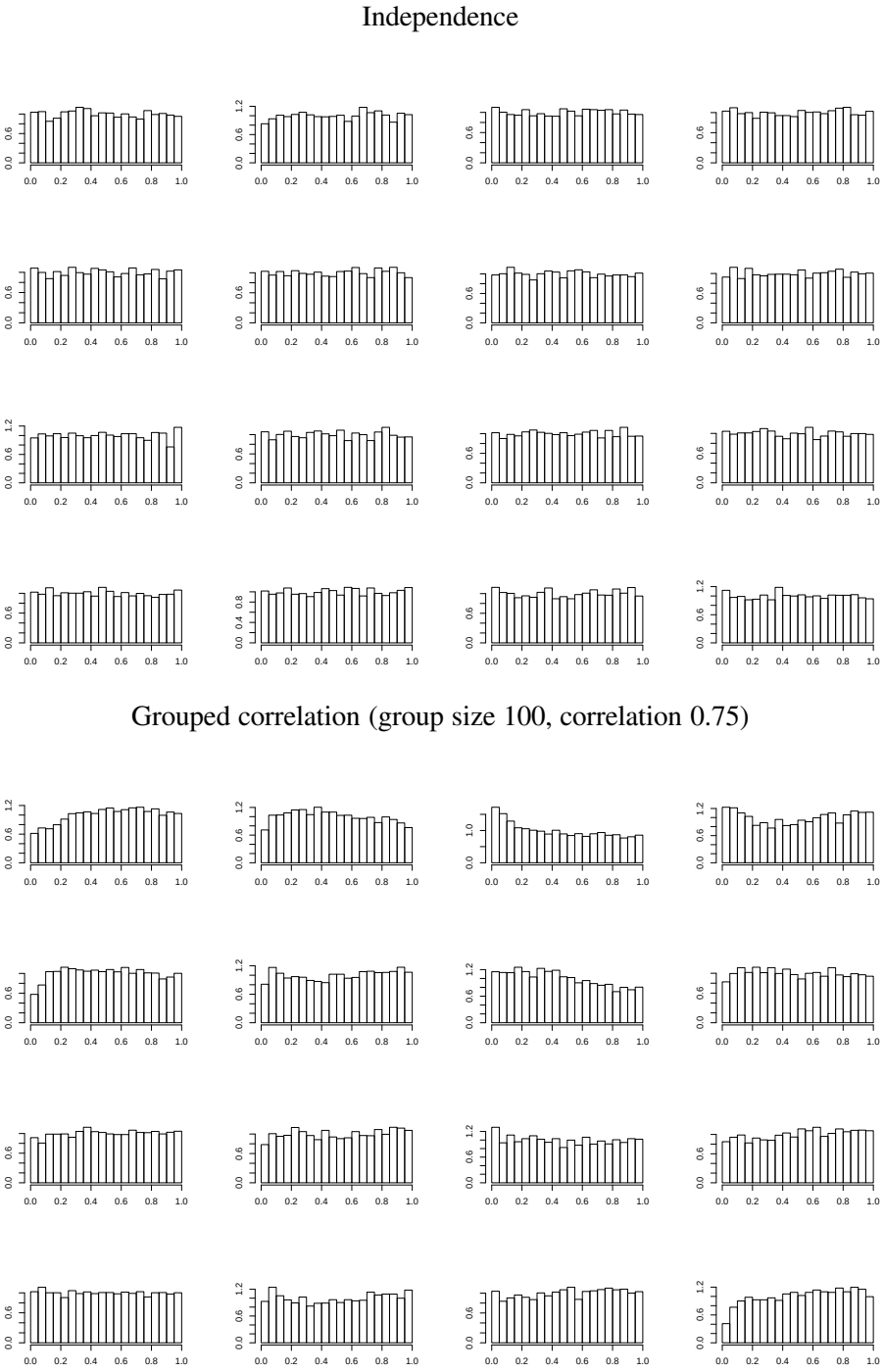


Figure 17: Randomly selected histograms of p-values from true null hypotheses for π_0=0.95. Upper (4×4) plots shows independent data and lower (4×4) plots dependent data with group size 100 and correlation 0.75.

estimators over all values of π_0. For the 'Convex' estimator the RMSE is decreasing with increasing values of π_0, i.e. RMSE equal to 0.0248, 0.0162,0.0138 and 0.0128 for π_0=(0.5,0.8,0.90.95). Also, it is interesting to note that the RMSEs of this estimator as well as 'Longest-length' and the tailored kernel method decrease with increasing π_0. By contrast, the RMSEs of the other estimators are seen to increase with increasing π_0.

Let us now consider the Grenander estimator and the 'Longest-length'-estimator. Obviously the Grenander estimator is a terrible choice (it could scarcely be called an estimator at all). However, the 'Longest-length'-idea salvages most of the damage and performs quite well. It overestimates π_0 on average for all values of π_0 under independence.

Schweder and Spjøtvoll's estimator with the Storey bootstrap routine $\widehat{\pi}_0(\hat{\lambda})$ has little bias in the case $\pi_0 = 0.5$, but the variance is quite high. Storey (2002b) claims that $\widehat{\pi}_0(\hat{\lambda})$ is a *conservative* estimator, which means that is should overestimate π_0 on average. Our results show the contrary: π_0 is clearly on the average *underestimated* for $\pi_0 = 0.8, 0.9$ and 0.95.

Considering the two kernel density estimate-based estimation methods, the "especially tailored" estimator seems to be clearly superior to the estimator using Silverman's (1986) rule of thumb. The variance is uniformly smaller, although the bias is slightly larger for the "tailored" estimator. However, the reduction of variance is so large that this should be a small price too pay. It also appears that the improvement of the "tailored" estimator over Silverman's estimator increases with π_0. For large values of π_0, i.e.$\pi_0 = 0.95$, the kernel estimators have the undesired property of a large build-up of values at 1.0 (due to 1 being an upper limit).

If we use the RMSE as a measure of the quality of estimation, the 'Convex' estimate based method is clearly best for independent data. This estimation method has the lowest RMSE for all the situations we considered. The only drawback with this estimator is that is seen to slightly underestimate π_0 when $\pi_0 = 0.95$, e.g. mean of 0.9461, whereas the 'Longest-length' and the kernel estimators gives a slight overestimation. In applications underestimation would be considered more serious than overestimation.

Finally, we make some observations based on the comparison between the simulated $\hat{\pi}_0$ from the different estimation methods depicted in Figure 15. As we might expect, the two kernel density estimation based methods are seen to be strongly linearly correlated. The "tailored" kernel method also seems to be related to the 'Convex' estimator. In addition, we see that the for 'Convex' estimator a subset of the estimated values are linearly dependent to the 'Longest-length' estimator and the 'SchSpjSto' estimator. A subset of the estimates from the 'SchSpjSto' estimator is again correlated to the estimates from the other methods, except the Grenander estimator.

Dependent data

The general observations when introducing increased degree of grouped correlation into the data are as follows. The 1st quantile and the mean clearly decreases. The median of the data also decreases, but to a smaller extent. The 3rd quantile of the data, the standard deviation and the RMSE clearly increases. Increased group size has similar effect as increased correlation. Looking at the density estimates, the skewness of the distribution of the estimates increases with increased dependence.

Looking at the density plots, we see that the degree of skewness of the estimators increases with increasing value of π_0.

For all cases investigated, the 'Convex' and/or the 'Longest-length' estimator performs the best wrt.

RMSE. For moderately dependent data the 'Convex' estimator gives the lowest RMSE, but for moderate to high dependency the 'Longest-length' estimator outperforms the 'Convex' estimator. This change is dependent on the value of π_0. For moderate π_0 ($\pi_0 = 0.5$), the 'Convex' estimator is better than the 'Longest-length' estimator (wrt. RMSE) for all values of group size and correlation investigated. Here the 'Longest-length' estimator has the largest bias (after the Grenander estimator) and is found to substantially overestimate π_0. For larger values of π_0, the bias of the 'Longest-length' estimator is smaller, and for $\pi_0 = 0.8$ the 'Convex' estimator and the 'Longest-length' estimator has approximate the same RMSE for $\rho = 0.5$ and $\rho = 0.75$. For $\pi_0 = 0.9$ and 0.95 the 'Longest-length' estimator outperformes the 'Convex' estimator for correlations equal to and above $\rho = 0.25$.

We have looked more closely at the case of $\pi_0 = 0.95$, group size 100 and correlation $\rho = 0.75$, to try to find reasons why the 'Longest-length' estimator performs better than the 'Convex' estimator. We have pin-pointed a typical situation in the case of decreasing p-value histograms, where the 'Longest-length' estimator finds a plateau and thus gives a higher estimate than the 'Convex' estimator. Examples of this are found in Figure 18.

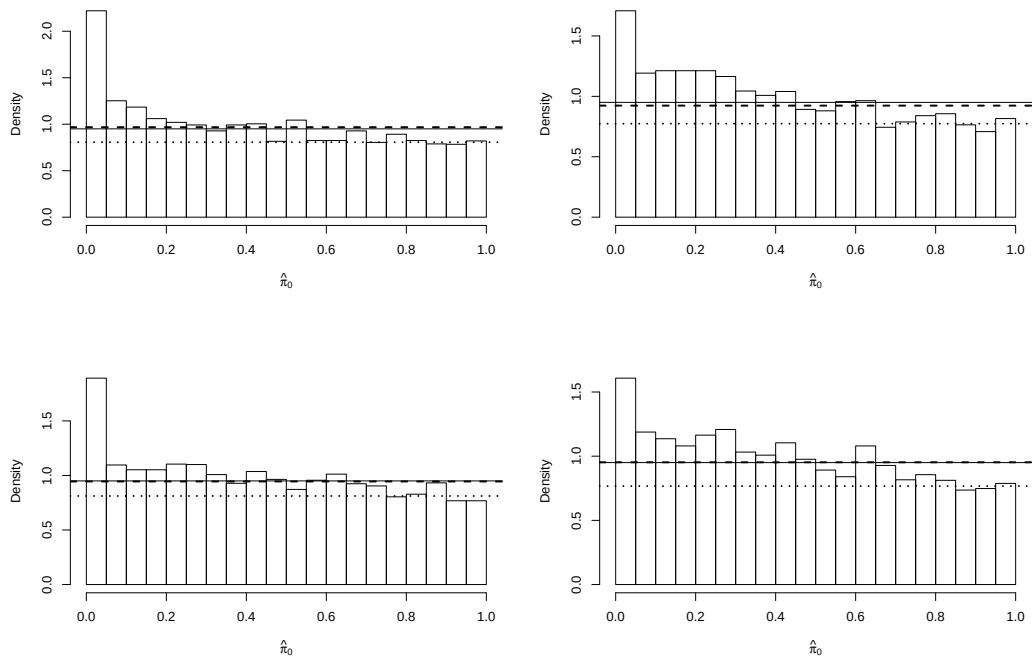


Figure 18: Cases where the 'Convex' estimator underestimates π_0, while the 'Longest-length' estimator gives a better estimate. Cases are taken from $\pi_0 = 0.95$, group size 100 and correlation $\rho = 0.75$ The solid line is at $\pi_0 = 0.95$, the dashed line is the 'Longest-length' estimator and the dotted line is the 'Convex' estimator.

7 Application to DNA microarray data

In this section the estimators are evaluated on real data sets two from DNA microarray experiments. Both data sets are from studies where each of m genes were tested for differential expression. The estimates were calculated using the same R code as was used in the simulation experiment described in Section 6

7.1 Description of data set

The first data set we consider is from Nygaard et al. (2003). The objective of this study was to investigate the impact of mRNA amplificationon gene expression ratios. mRNA amplification is a technique which is useful in cases where insufficient mRNA to perform a DNA microarray experiment is available. The main contribution of Nygaard et al.'s (2003) paper was an analysis of variance (ANOVA) model to investigate the sources of variation when using mRNA amplification. To verify the results found by the ANOVA model, the data was split in two groups by testing each gene for differential expression using a two-sample t-test (i.e. this was not done to find differences between groups for each gene). The authors emphasize that a two sample t-test is not an optimal choice of method for this small data set, which consists of two groups of 4 and 8 individuals, respectively. We will nevertheless use the p-values from the two-sample t-tests for estimating π_0. $m = 4331$ tests were performed.

The second data set is from Hedenfalk et al. (2001). This is a study regarding breast cancer, where one objective was to discover differentially expressed genes in tumors with a mutated *BRCA1* gene and the *BRCA2* gene, respectively. Gene expression levels were measured for 7 individuals (tumors) with the BRCA1 mutation and 8 individuls with the BRCA2 mutation. The p-values used here are calculated in Storey and Tibshirani (2003), on the basis of permutation tests. $m = 3170$ tests were performed.

7.2 Estimates of π_0

The estimates of π_0 for each of the data sets described in Section 7.1 and each estimator is shown in Table 3. Here, $\widehat{\pi}_0(\hat{\lambda})$ is Schweder and Spjøtvoll's (1982) estimator with Storey's (2002b) bootstrap choice of λ from Section 4.1, $\hat{\pi}_0^g$ is the "Grenander" estimator from Section 4.2.2, $\hat{\pi}_0^l$ is "longest" from Section 4.2.3, $\hat{\pi}_0^c$ is the estimator based on convex decreasing density estimation developed in Section 4.3, and $\hat{\pi}_0^{k1}$ and $\hat{\pi}_0^{k2}$ are the kernel density estimate based estimators from Section 4.4, with Silverman's (1986) and the especially tailored choice of smoothing parameter, respectively.

Dataset / estimator	$\widehat{\pi}_0(\hat{\lambda})$	$\hat{\pi}_0^g$	$\hat{\pi}_0^l$	$\hat{\pi}_0^c$	$\hat{\pi}_0^{k1}$	$\hat{\pi}_0^{k2}$
Nygaard et al. (2003)	0.5864	0.3260	0.6789	0.6077	0.6100	0.6164
Hedenfalk et al. (2001)	0.6719	0.4892	0.6717	0.6753	0.6900	0.6899

Table 3: Estimates of π_0

A dotchart of the estimates for each data set in shown in Figure 19. We see that $\hat{\pi}_0^g$ is much smaller than the other estimators for both data sets, as was the case in the simulation experiment. For the data from Nygaard et al. (2003), $\hat{\pi}_0^l$ is quite large, $\widehat{\pi}_0(\hat{\lambda})$ is rather small. The other three estimates are not very different. For the data from Hedenfalk et al. (2001), we see that the kernel based estimates

are almost exactly the same, and have a larger value than $\hat{\pi}_0^l$, $\hat{\pi}_0^c$ and $\widehat{\pi}_0(\hat{\lambda})$, which in turn are almost equal.

In Figure 20, Grenander's (1956) nonparametric maximum likelihood density estimate and the convex decreasing density estimate is shown for each set of p-values.

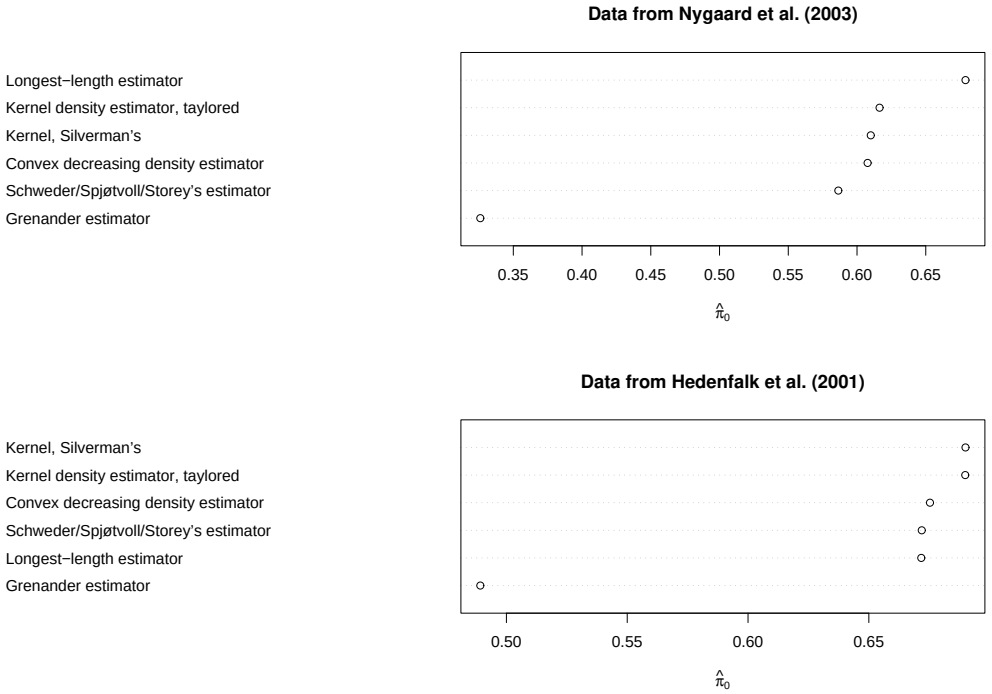


Figure 19: Dotcharts of estimated π_0 for each data set.

7.3 Accuracy of the estimates

In addition to the point estimates shown in Table 3, we would obviously also like to say something about the accuracy of the estimates.

For the estimates of π_0 based on the data from Hedenfalk et al. (2001), empirical confidence intervals could be estimated using bootstrapping, in the following way: Assume that the individuals (tumors, pasients) on which we measure gene expression are mutually independent. (Within each pasient, the measurements are not assumed to be independent.) For each bootstrap replication $i = 1, \ldots, B$, we form two groups: One with 7 individuals drawn with replacement from the BRCA1 group, and the other with 8 individuals drawn with replacement from the BRCA2 group. Then, for each i, we calculate an estimate $\hat{\pi}_0^{(i)}$ on the basis of the p-values from permutation tests using the bootstrap-sampled groups. Let $\hat{\pi}_{0,\alpha}$ be the empirical α-quantile of the bootstrap-sampled $\hat{\pi}_0^{(i)}$, $i = 1, \ldots, B$. Then, a $1 - 2\alpha$ bootstrap percentile interval is given by $(\hat{\pi}_{0,\alpha}, \hat{\pi}_{0,1-\alpha})$ (Efron and Tibshirani 1993). Note that this procedure is very computationally expensive, due to its two-level structure: For each bootstrap replication, a large number of permuation replications are needed to calculate the p-values. For this reason, bootstrap confidence intervals are not calculated in this work.

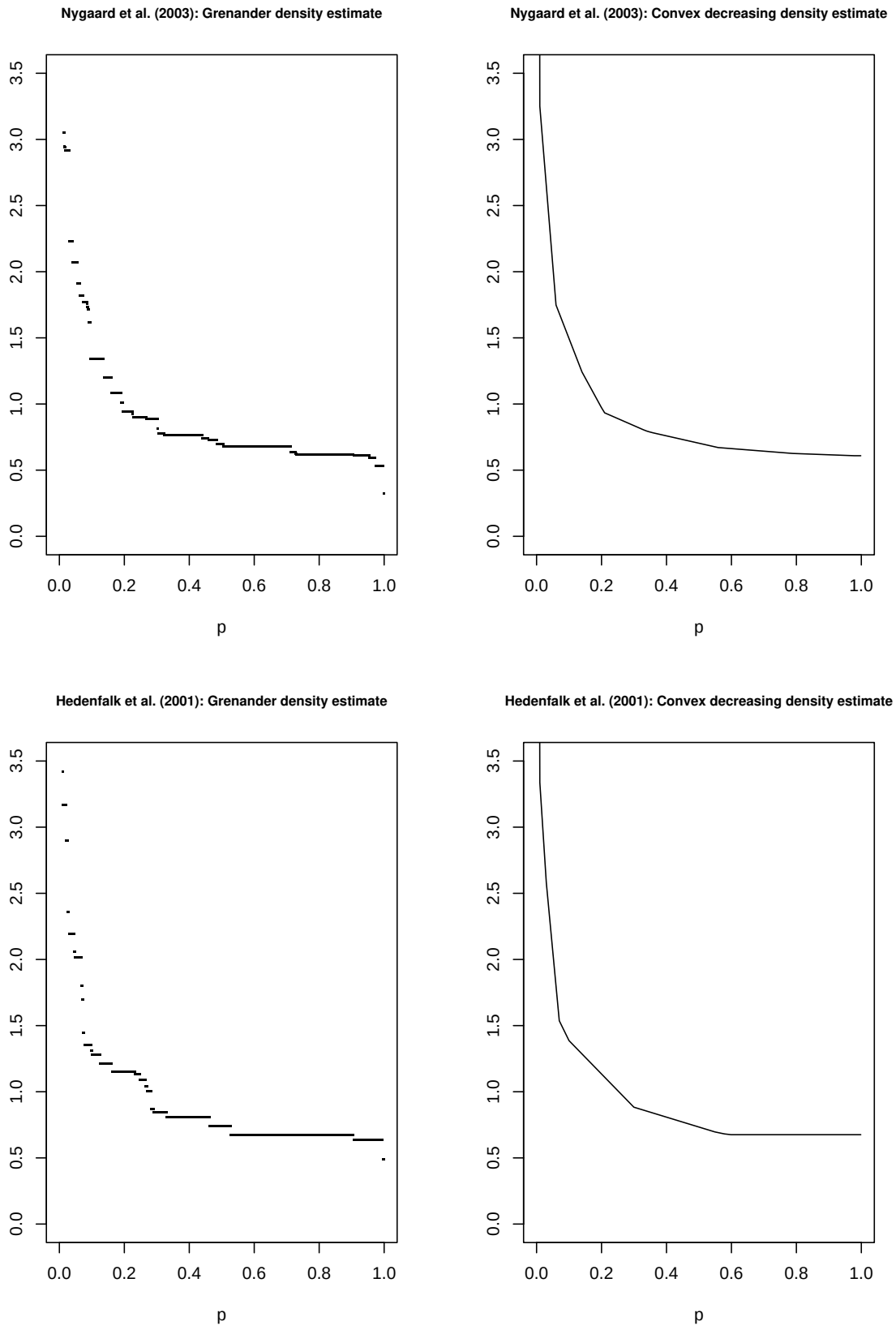


Figure 20: Grenander density estimate and convex decreasing density estimate for each data set.

In principle, bootstrap percentile intervals could be calculated for the Nygaard et al. (2003) data in the same way as described above for the Hedenfalk et al. (2001) data. To do so, an investigation into the choice of method for calculating the p-values would be needed. Therefore, we have chosen not to calculate bootstrap confidence intervals for the Nygaard et al. (2003) data.

Thus, we must refer the reader to the simulation experiment in Section 6 for an assessment of the bias and variance of the estimators.

8 Conclusions and further work

The new contributions of this work includes the estimator $\hat{\pi}_0^c$ based on convex decreasing density estimation developed in Section 4.3, the estimator $\hat{\pi}_0^l$ ("longest") from Section 4.2.3, and kernel density estimation with the smoothing parameter specifically chosen for estimation of π_0, developed in Section 4.4.2

Considering the results of the simulation experiments in Section 6, the new estimation procedures appear to work well. Particularly the 'Convex' estimator $\hat{\pi}_0^c$ and the 'Longest-length' estimator $\hat{\pi}_0^l$ performs very well, and seems to be clearly better than Schweder and Spjøtvoll's (1982) estimator with Storey's (2002b) bootstrap choice of λ, which is probably the most used of the previously known estimation methods. This is true both for the situation with independent data and for all degrees of grouped dependence and values of π_0 studied in our simulation experiment. For moderate degree of dependence, the 'Convex' estimator is found to perform the best, but for a substantial degree of dependence the 'Longest-length' estimator performes the best. Our belief when considering DNA microarray data, is that the degree of correlation is small to moderate, and thus we find the 'Convex' estimator to be the best choice.

The estimators based on kernel density estimation provide a very fast and simple method for estimating π_0. The especially tailored choice of smoothing parameter seems more appropriate than Silverman's (1986) rule of thumb when estimating π_0.

There are still some open issues regarding estimation of π_0 and its application to DNA microarrays. In particular, more work should be done on modelling dependence among the hypothesis tests. This should be done both in general terms and specifically for DNA microarrays. In the latter case, a thorough knowledge of biological sources of dependence, such as pathways in the genome, seems necessary. Specific questions include; does microarray data exhibit correlation within groups and independence between groups, how large are the groups, and how high are the correlations? Further attempts should also be made to design a large set of simulation experiments that are relevant to the application of DNA microarrays.

More work is also needed on addressing how bias and variability should be estimated when applying the estimators to data from DNA microarray experiments.

In the introduction we emphasized that our starting point was a set of p-values, and we assumed that the p-values were "correctly" calculated. The question of which method that should be used for calculating these p-values remains unanswered.

References

Bedford, T. and Cooke, R. M. (2002). Vines — a new graphical model for dependent random variables, *Annals of Statistics* **30**(4): 1031–1068.

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

Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency, *Annals of Statistics* **29**(4): 1165–1188.

Casella, G. and Berger, R. L. (1990). *Statistical Inference*, Duxbury Press, Belmont, California.

Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*, Chapman & Hall, New York.

Fedorov, V. (1972). *Theory of optimal experiments*, Academic Press, New York.

Genovese, C. and Wasserman, L. (2002). False discovery rates, *Technical report*, Carnegie Mellon University. http://www.stat.cmu.edu/tr/tr762/tr762.pdf.

Genovese, C. and Wasserman, L. (2003). A stochastic process approach to false discovery rates, *Technical Report 762*, Department of Statistics, Carnegie Mellon University.

Grenander, U. (1956). On the theory of mortality measurement. Part II, *Skandinavisk Aktuarietidskrift* **39**: 125–53.

Groeneboom, P., Jongbloed, G. and Wellner, J. A. (2002). The support reduction algorithm for computing nonparametric function estimates in mixture models, *Technical Report 2002-13*, Department of Stochastics, Vrije Universiteit.

Hedenfalk, I., Duggan, D., Chen, Y., Radmacher, M., Bittner, M., Simon, R., Meltzer, P., Gusterson, B., Esteller, M., Kallioniemi, O. P., Wilfond, B., Borg, Å. and Trent, J. (2001). Gene-expression profiles in hereditary breast cancer, *New England Journal of Medicine* **344**(8): 539–548.

Heller, G. and Qin, J. (2003). A mixture model for finding informative genes in microarray studies, *Technical report*, Department of Epidemiology and Biostatistics, Memorial Sloan-Kettering Cancer Center.

Ihaka, R. and Gentleman, R. (1996). R: A language for data analysis and graphics, *Journal of Computational and Graphical Statistics* **5**(3): 299–314.

Lehmann, E. L. (1986). *Testing Statistical Hypotheses. 2nd edition*, John Wiley & Sons, New York.

Miller, C. J., Genovese, C., Nichol, R. C., Wasserman, L., Connolly, A., Reichart, D., Hopkins, A., Schneider, J. and Moore, A. (2001). Controlling the false discover rate in astrophysical data analysis, *Technical Report 747*, Department of Statistics, Carnegie Mellon University.

Nelson, R. B. (1999). *An Introduction to Copulas*, Springer, New York.

Nygaard, V., Løland, A., Holden, M., Langaas, M., Rue, H., Liu, F., Myklebost, O., Fodstad, Ø., Hovig, E. and Smith-Sørensen, B. (2003). Effects of mRNA amplification on gene expression ratios in cDNA experiments estimated by analysis of variance, *BMC Genomics*.

Prakasa Rao, B. L. S. (1992). *Identifiability in Stochastic Models: Characterization of Probability Distributions*, Academic Press.

Reiner, A., Yekutieli, D. and Benjamini, Y. (2003). Identifying differentially expressed genes using false discovery rate controlling procedures, *Bioinformatics* **19**(3): 368–375.

Robertson, T., Wright, F. T. and Dykstra, R. L. (1988). *Order Restricted Statistical Inference*, John Wiley & Sons, New York.

Schulze, A. and Downward, J. (2001). Navigating gene expression using microarrays — a technology review, *Nature Cell Biology* **3**: E190–E195.

Schweder, T. and Spjøtvoll, E. (1982). Plots of p-values to evaluate many tests simultaneously, *Biometrika* **69**(3): 493–502.

Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, New York.

Simes, R. J. (1986). An improved Bonferroni procedure for multiple tests of significance, *Biometrika* **73**: 751–754.

Smyth, G. K., Yang, Y. H. and Speed, T. (2003). Statistical issues in cDNA microarray data analysis, *in* M. J. Brownstein and A. B. Khodursky (eds), *Functional Genomics: Methods and Protocols*, Humana Press, New Jersey, chapter 9.

Storey, J. D. (2002a). A direct approach to false discovery rates, *Journal of the Royal Statistical Society, Series B* **64**(3): 479–498.

Storey, J. D. (2002b). *False discovery rates: Theory and applications to DNA microarrays*, PhD thesis, Department of Statistics, Stanford University.

Storey, J. D. and Tibshirani, R. (2003). Statistical significance for genome-wide experiments, Preprint, Department of Statistics, University of California, Berkeley.

Turkheimer, F. E., Smith, C. B. and Schmidt, K. (2001). Estimation of the number of "true" null hypotheses in multivariate analysis of neuroimaging data, *NeuroImage* **13**(5): 920–930.

Wynn, H. (1970). Some algorithmic aspects of the theory of optimal design, *Annals of Mathematical Statistics* **6**: 1286–1301.

A Additional tables and plots from the simulation experiment

Additional tables and plots from the simulation experiment described in Section 6 are found here.

In particular,

- Tables 4, 5, and 6 show summary statistics for the simulation experiment with π_0 equal to 0.5, 0.8 and 0.95, respectively.
- Figures 21 through 24 show the first quantile, median, third quantile and standard deviation (in data sets of N=1000) as functions of correlation for group sizes 50 and 100 and values of $\pi_0 \in \{0.5, 0.8, 0.9, 0.95\}$ for the six methods considered.
- Figures 25 through and 30 show density estimates of the estimated π_0 for the different estimators, for group sizes 50 and 100, for correlations $\rho \in \{0.25, 0.5, 0.75\}$, and for π_0 equal to 0.5, 0.8 and 0.95.

Indep.:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.4606	0.5083	0.5202	0.5189	0.5299	0.5635	0.0161	0.0248
Grenander	0.0624	0.2845	0.3791	0.366	0.4564	0.5537	0.1076	0.1719
'Longest-length'	0.4737	0.5209	0.534	0.5356	0.5496	0.6441	0.0227	0.0422
Kernel, tailored	0.461	0.5113	0.5242	0.5242	0.5376	0.5859	0.0192	0.031
Kernel, Silverman	0.4547	0.5089	0.5239	0.5236	0.5383	0.5914	0.0222	0.0324
'SchSpjSto'	0.4	0.4883	0.509	0.5031	0.5239	0.5633	0.0292	0.0293
Group 50 Corr 0.25:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.4367	0.5045	0.5202	0.5182	0.5342	0.5744	0.0215	0.0282
Grenander	0.0778	0.2904	0.3801	0.3662	0.4592	0.5552	0.1095	0.1729
'Longest-length'	0.4502	0.5183	0.5343	0.5349	0.5505	0.6587	0.0267	0.044
Kernel, tailored	0.4499	0.5072	0.5248	0.5236	0.5403	0.5909	0.0235	0.0333
Kernel, Silverman	0.4456	0.5049	0.5255	0.5231	0.541	0.6016	0.0261	0.0349
'SchSpjSto'	0.3875	0.4844	0.5078	0.5036	0.5264	0.5657	0.0303	0.0305
Group 50 Corr 0.50:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.4123	0.4975	0.5203	0.5185	0.5405	0.5999	0.0318	0.0368
Grenander	0.0556	0.272	0.3734	0.3621	0.4562	0.5741	0.1146	0.1793
'Longest-length'	0.4266	0.5146	0.5353	0.5345	0.5557	0.7144	0.0325	0.0474
Kernel, tailored	0.4204	0.5019	0.5249	0.5248	0.5484	0.6372	0.034	0.0421
Kernel, Silverman	0.4089	0.4997	0.5239	0.5241	0.5491	0.6456	0.0364	0.0437
'SchSpjSto'	0.34	0.4779	0.5076	0.5037	0.5326	0.5956	0.0395	0.0397
Group 50 Corr 0.75:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.3686	0.4823	0.5198	0.5139	0.5483	0.6252	0.0464	0.0485
Grenander	0.0662	0.2661	0.3673	0.354	0.4422	0.6075	0.1141	0.1853
'Longest-length'	0.3679	0.5055	0.5378	0.5342	0.5655	0.7054	0.0448	0.0563
Kernel, tailored	0.3784	0.4865	0.5244	0.5238	0.5603	0.712	0.0516	0.0569
Kernel, Silverman	0.3723	0.4845	0.5234	0.5228	0.5612	0.7239	0.0542	0.0588
'SchSpjSto'	0.3433	0.464	0.5033	0.4988	0.5385	0.6093	0.0527	0.0527
Group 100 Corr 0.25:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.4169	0.5031	0.5204	0.518	0.5349	0.5757	0.0241	0.0301
Grenander	0.0673	0.2877	0.3866	0.3682	0.4597	0.5627	0.1087	0.1709
'Longest-length'	0.4335	0.5156	0.5349	0.5347	0.5532	0.629	0.0291	0.0453
Kernel, tailored	0.4267	0.5068	0.5247	0.5237	0.5418	0.6078	0.0259	0.0351
Kernel, Silverman	0.4178	0.5044	0.5233	0.5231	0.5423	0.6164	0.0282	0.0364
'SchSpjSto'	0.3625	0.4845	0.5067	0.5035	0.5259	0.5739	0.0326	0.0327
Group 100 Corr 0.50:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.3677	0.4908	0.5194	0.5162	0.5449	0.6105	0.0409	0.0439
Grenander	0.06	0.2808	0.3708	0.3605	0.4486	0.5877	0.1135	0.1799
'Longest-length'	0.4005	0.5104	0.537	0.5335	0.5603	0.6857	0.0391	0.0515
Kernel, tailored	0.3766	0.4949	0.5233	0.5233	0.5515	0.6635	0.0441	0.0499
Kernel, Silverman	0.3703	0.4927	0.5228	0.5229	0.553	0.6747	0.0464	0.0517
'SchSpjSto'	0.3286	0.4732	0.5053	0.5023	0.5376	0.6079	0.0483	0.0483
Group 100 Corr 0.75:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.2567	0.4699	0.5172	0.5088	0.5555	0.632	0.062	0.0626
Grenander	0.0708	0.2767	0.3688	0.3633	0.4515	0.6219	0.1184	0.1808
'Longest-length'	0.2669	0.4954	0.5379	0.5311	0.5714	0.8949	0.0593	0.0669
Kernel, tailored	0.2717	0.4751	0.5238	0.5231	0.5696	0.7941	0.0704	0.0741
Kernel, Silverman	0.263	0.4719	0.522	0.5224	0.5712	0.8201	0.0733	0.0767
'SchSpjSto'	0.2537	0.4509	0.5031	0.4966	0.5467	0.633	0.0662	0.0662

Table 4: Summary statistics for set of estimates, $\hat{\pi}_0$ for $\pi_0 = 0.5$.

Indep.:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.7333	0.7953	0.8066	0.8041	0.8152	0.8405	0.0157	0.0162
Grenander	0.1105	0.4429	0.5882	0.5637	0.7124	0.8355	0.1717	0.2921
'Longest-length'	0.7377	0.8056	0.817	0.8177	0.8292	0.9315	0.0201	0.0268
Kernel, tailored	0.7319	0.7964	0.8094	0.8096	0.8237	0.8716	0.0208	0.0229
Kernel, Silverman	0.7129	0.7899	0.81	0.809	0.8278	0.8995	0.0286	0.0299
'SchSpjSto'	0.628	0.7694	0.7921	0.7845	0.8076	0.8395	0.0326	0.0361
Group 50 Corr 0.25:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.7223	0.7906	0.8071	0.8039	0.8195	0.8528	0.0217	0.022
Grenander	0.0906	0.4415	0.5861	0.5618	0.7054	0.8454	0.1696	0.2924
'Longest-length'	0.7345	0.8014	0.8174	0.8165	0.8319	0.9226	0.0236	0.0288
Kernel, tailored	0.721	0.7918	0.8119	0.8105	0.8296	0.9017	0.0267	0.0287
Kernel, Silverman	0.7054	0.7857	0.8096	0.8094	0.8342	0.9128	0.0342	0.0355
'SchSpjSto'	0.6286	0.7618	0.7896	0.7828	0.8113	0.8508	0.0379	0.0416
Group 50 Corr 0.50:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.6498	0.7795	0.8063	0.8015	0.8291	0.8691	0.0351	0.0351
Grenander	0.0985	0.4489	0.5816	0.565	0.7057	0.8505	0.1683	0.2891
'Longest-length'	0.6764	0.7971	0.8221	0.8165	0.8384	0.9288	0.0317	0.0358
Kernel, tailored	0.6667	0.7806	0.8118	0.8109	0.8411	0.9551	0.0417	0.0431
Kernel, Silverman	0.6428	0.7782	0.8105	0.811	0.8451	0.9802	0.0479	0.0492
'SchSpjSto'	0.6067	0.7544	0.79	0.7835	0.8215	0.8703	0.0467	0.0495
Group 50 Corr 0.75:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.5724	0.7639	0.8036	0.794	0.8338	0.8798	0.0514	0.0518
Grenander	0.1016	0.4359	0.5835	0.5626	0.7006	0.8647	0.1712	0.2928
'Longest-length'	0.6079	0.785	0.8221	0.8135	0.8462	0.9634	0.0495	0.0513
Kernel, tailored	0.6059	0.7686	0.8123	0.812	0.8586	1	0.0645	0.0656
Kernel, Silverman	0.5807	0.7636	0.8115	0.8123	0.8601	1	0.0711	0.0721
'SchSpjSto'	0.5433	0.741	0.7857	0.7779	0.8261	0.8793	0.0599	0.0638
Group 100 Corr 0.25:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.7065	0.7857	0.8064	0.8023	0.8219	0.858	0.0267	0.0268
Grenander	0.1077	0.4295	0.5764	0.5545	0.6976	0.8444	0.1701	0.2987
'Longest-length'	0.6913	0.7999	0.8198	0.8171	0.8351	0.9031	0.0275	0.0323
Kernel, tailored	0.7084	0.7879	0.8091	0.8088	0.8305	0.9005	0.0313	0.0325
Kernel, Silverman	0.6672	0.7825	0.8071	0.8075	0.8333	0.9285	0.0375	0.0382
'SchSpjSto'	0.6	0.7578	0.7873	0.7813	0.8117	0.8569	0.0404	0.0445
Group 100 Corr 0.50:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.6428	0.7698	0.805	0.7982	0.8345	0.8722	0.0446	0.0446
Grenander	0.1155	0.4485	0.5809	0.5664	0.6974	0.8632	0.1668	0.287
'Longest-length'	0.6067	0.7891	0.8211	0.8133	0.8445	0.934	0.0418	0.0438
Kernel, tailored	0.6447	0.7737	0.8092	0.8096	0.8451	0.9607	0.0526	0.0535
Kernel, Silverman	0.6317	0.7708	0.808	0.8092	0.8474	0.9914	0.0576	0.0583
'SchSpjSto'	0.568	0.746	0.7849	0.7815	0.8245	0.8734	0.0526	0.0558
Group 100 Corr 0.75:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.5008	0.741	0.8029	0.7838	0.8387	0.8875	0.0705	0.0724
Grenander	0.1129	0.4173	0.5674	0.5547	0.6927	0.8856	0.177	0.3025
'Longest-length'	0.5614	0.7743	0.8227	0.8115	0.8561	0.9981	0.0643	0.0653
Kernel, tailored	0.5306	0.7476	0.8103	0.809	0.8678	1	0.0881	0.0885
Kernel, Silverman	0.5125	0.7415	0.8085	0.8074	0.8723	1	0.0946	0.0949
'SchSpjSto'	0.4743	0.7161	0.7775	0.7665	0.8303	0.8874	0.0777	0.0846

Table 5: Summary statistics for set of estimates, $\hat{\pi}_0$ for $\pi_0 = 0.8$.

Indep.:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.8918	0.9392	0.9481	0.9461	0.9548	0.9731	0.0121	0.0128
Grenander	0.144	0.5027	0.6932	0.6599	0.8367	0.9618	0.2039	0.3546
'Longest-length'	0.8778	0.9458	0.9544	0.9541	0.963	0.9957	0.0152	0.0157
Kernel, tailored	0.9034	0.942	0.9528	0.953	0.9646	1	0.0162	0.0165
Kernel, Silverman	0.8538	0.9326	0.9519	0.9523	0.9741	1	0.0296	0.0297
'SchSpjSto'	0.772	0.91	0.9349	0.9257	0.9487	0.9705	0.0314	0.0397
Group 50 Corr 0.25:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.8597	0.9333	0.9489	0.9444	0.9589	0.9793	0.0192	0.02
Grenander	0.1373	0.5035	0.6807	0.6547	0.8179	0.9665	0.1951	0.3539
'Longest-length'	0.8847	0.9438	0.9563	0.9539	0.9664	0.9988	0.0184	0.0188
Kernel, tailored	0.8732	0.9363	0.9536	0.9521	0.9679	1	0.0224	0.0225
Kernel, Silverman	0.8339	0.9275	0.9517	0.9504	0.9774	1	0.034	0.034
'SchSpjSto'	0.77	0.904	0.9326	0.9232	0.9515	0.9765	0.0364	0.0452
Group 50 Corr 0.50:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.8031	0.9247	0.9478	0.9393	0.9625	0.9794	0.0301	0.032
Grenander	0.1372	0.5106	0.6733	0.6601	0.823	0.9758	0.1952	0.3495
'Longest-length'	0.8167	0.9355	0.9552	0.9485	0.9676	0.9988	0.0267	0.0268
Kernel, tailored	0.8144	0.9289	0.9531	0.9505	0.9763	1	0.0347	0.0347
Kernel, Silverman	0.7761	0.9183	0.9526	0.9483	0.9867	1	0.0428	0.0428
'SchSpjSto'	0.728	0.8928	0.9278	0.9201	0.957	0.9802	0.0435	0.0528
Group 50 Corr 0.75:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.6796	0.9061	0.9446	0.9288	0.9644	0.9869	0.047	0.0516
Grenander	0.1368	0.5074	0.6869	0.659	0.8236	0.985	0.2072	0.3572
'Longest-length'	0.7205	0.9209	0.9532	0.94	0.9692	0.9987	0.042	0.0432
Kernel, tailored	0.404	0.9126	0.9546	0.9445	0.992	1	0.0551	0.0554
Kernel, Silverman	0.6787	0.9021	0.9536	0.9417	1	1	0.0596	0.0602
'SchSpjSto'	0.592	0.8765	0.9299	0.9117	0.9598	0.9879	0.0597	0.0709
Group 100 Corr 0.25:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.8253	0.9317	0.9493	0.9438	0.9612	0.9789	0.0225	0.0234
Grenander	0.1247	0.5223	0.7004	0.6643	0.8338	0.9691	0.1983	0.3478
'Longest-length'	0.8402	0.942	0.9569	0.9528	0.967	0.9993	0.0204	0.0206
Kernel, tailored	0.8448	0.9362	0.9542	0.9529	0.9725	1	0.0264	0.0266
Kernel, Silverman	0.8108	0.9262	0.9519	0.9499	0.9789	1	0.0362	0.0362
'SchSpjSto'	0.74	0.9004	0.9299	0.9229	0.954	0.9804	0.0386	0.0472
Group 100 Corr 0.50:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.7634	0.917	0.9471	0.9359	0.9655	0.988	0.0385	0.041
Grenander	0.1247	0.5204	0.6815	0.6625	0.8255	0.979	0.1981	0.3492
'Longest-length'	0.771	0.9283	0.9567	0.9453	0.9704	0.9981	0.0347	0.035
Kernel, tailored	0.7986	0.9232	0.9548	0.9496	0.9877	1	0.0434	0.0434
Kernel, Silverman	0.7548	0.9116	0.9529	0.9458	0.997	1	0.0501	0.0502
'SchSpjSto'	0.73	0.8867	0.9307	0.9186	0.962	0.9876	0.0507	0.0597
Group 100 Corr 0.75:	Min.	1st Qu.	Med.	Mean	3rd Qu.	Max.	St.Dev.	RMSE
Convex	0.6733	0.8913	0.9421	0.9205	0.9664	0.9948	0.0611	0.0679
Grenander	0.1324	0.5118	0.6836	0.6606	0.835	0.984	0.2101	0.3576
'Longest-length'	0.6684	0.9103	0.9503	0.9343	0.9719	0.9999	0.0511	0.0535
Kernel, tailored	0.7095	0.9014	0.9564	0.939	1	1	0.0646	0.0656
Kernel, Silverman	0.6801	0.8895	0.9593	0.934	1	1	0.0735	0.0752
'SchSpjSto'	0.6367	0.8667	0.9262	0.9045	0.962	0.9954	0.0711	0.0844

Table 6: Summary statistics for set of estimates, $\hat{\pi}_0$ for $\pi_0 = 0.95$.

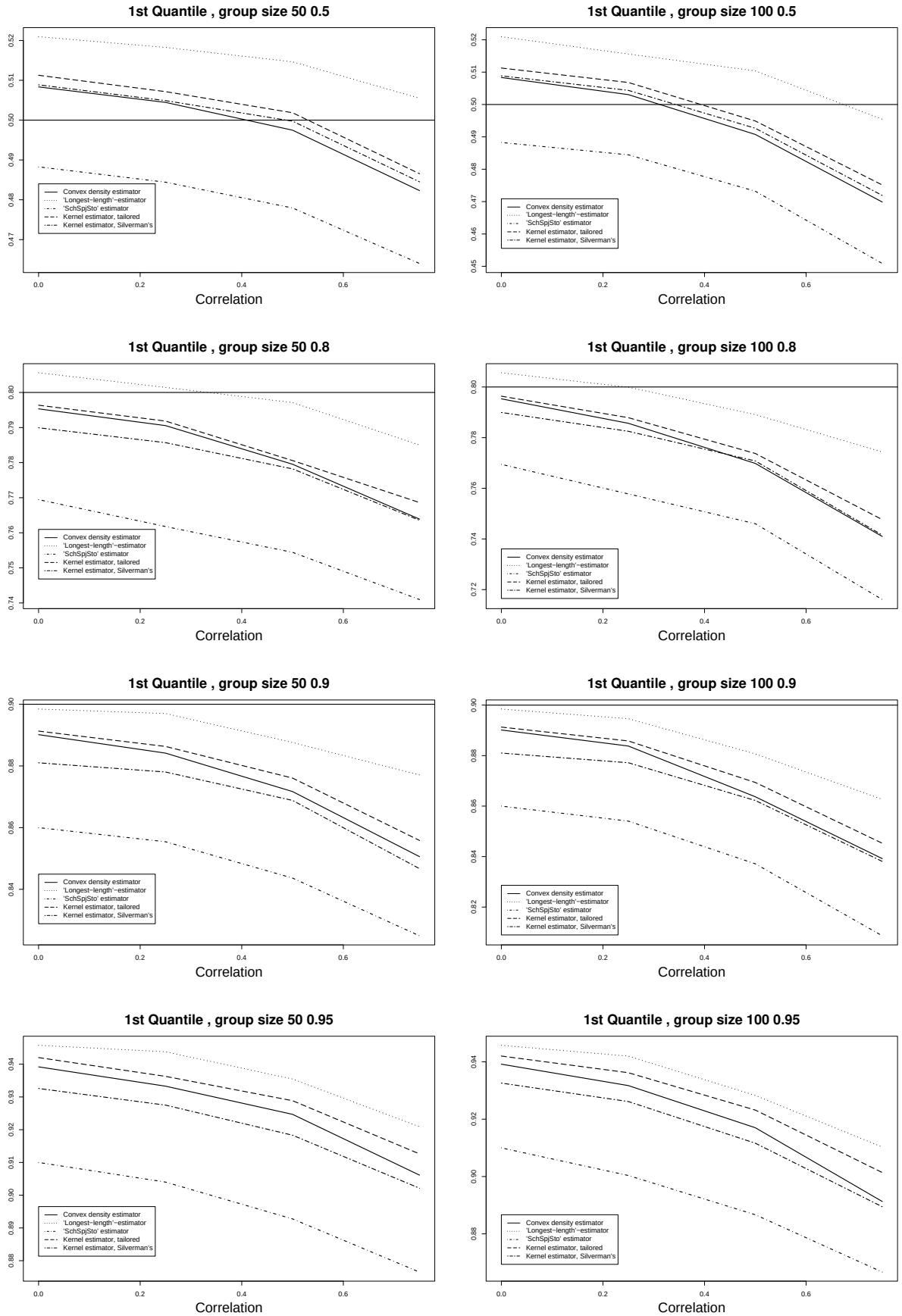


Figure 21: First quantile (in data sets of N=1000) as a function of correlation for group sizes 50 and 100 and values of $\pi_0 \in \{0.5, 0.8, 0.9, 0.95\}$ for five of the six methods considered (the Grenander estimator is excluded from the plot).

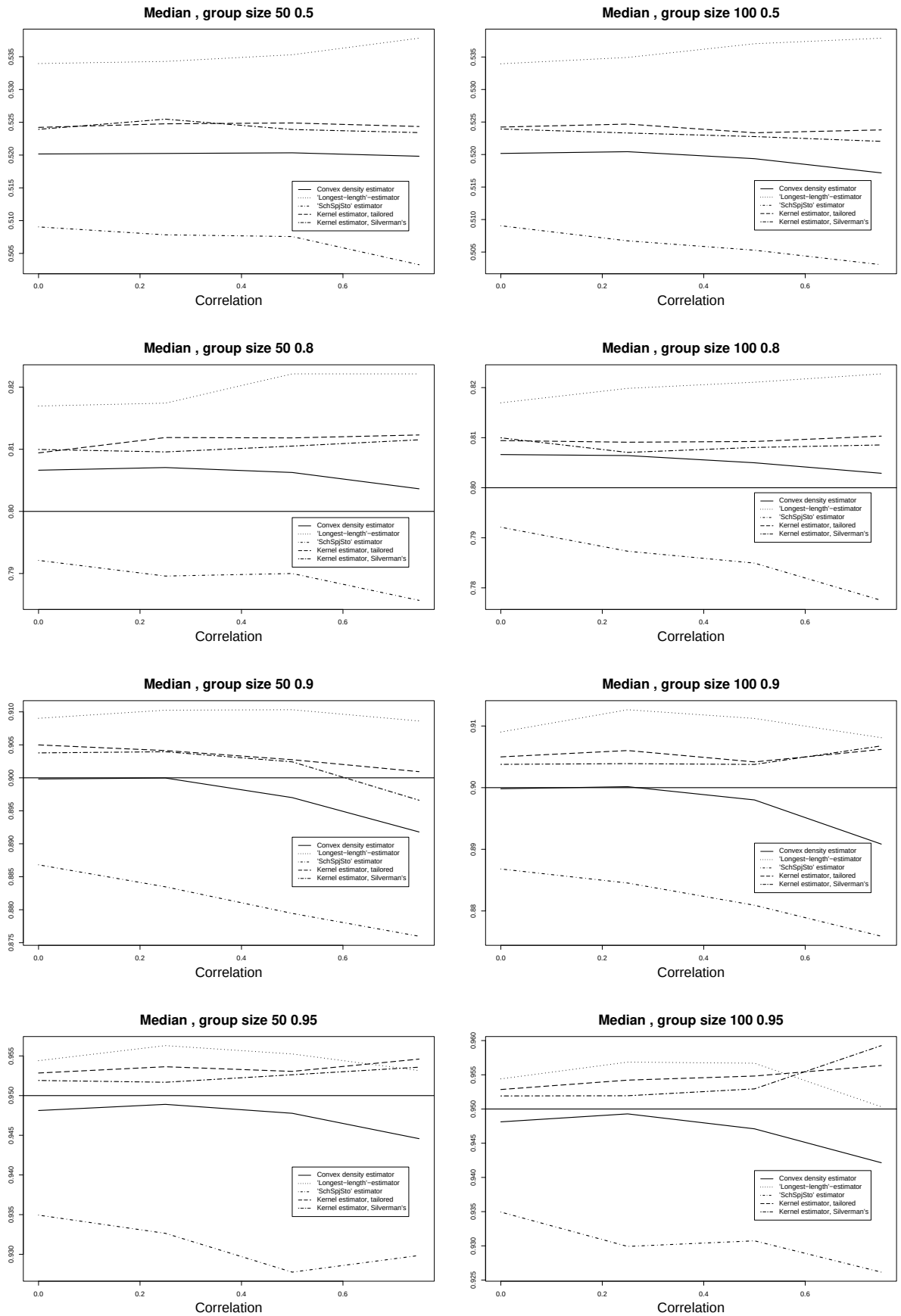


Figure 22: Median (in data sets of N=1000) as a function of correlation for group sizes 50 and 100 and values of $\pi_0 \in \{0.5, 0.8, 0.9, 0.95\}$ for five of the six methods considered (the Grenander estimator is excluded from the plot).

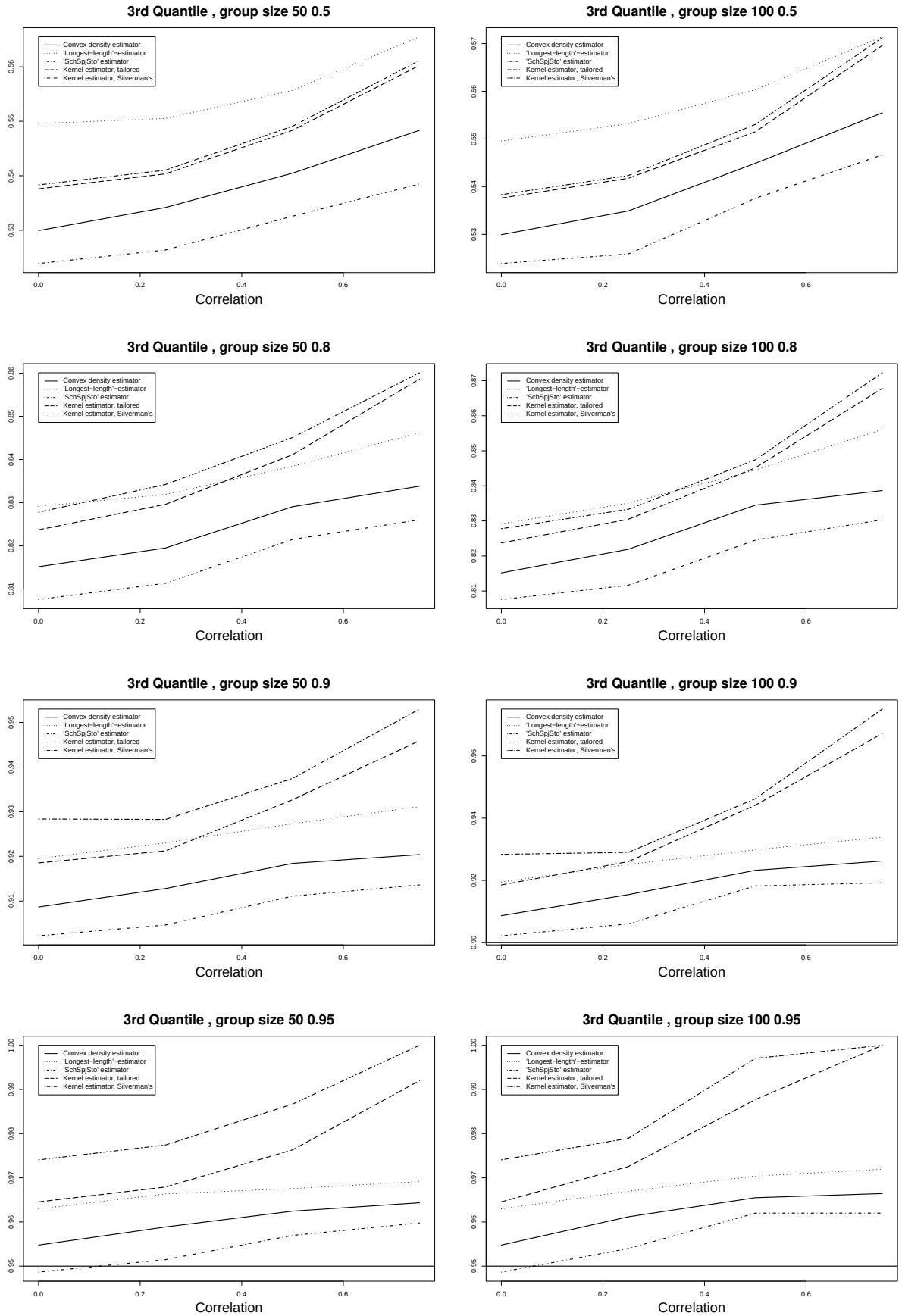


Figure 23: Third quantile (in data sets of N=1000) as a function of correlation for group sizes 50 and 100 and values of $\pi_0 \in \{0.5, 0.8, 0.9, 0.95\}$ for five of the six methods considered (the Grenander estimator is excluded from the plot).

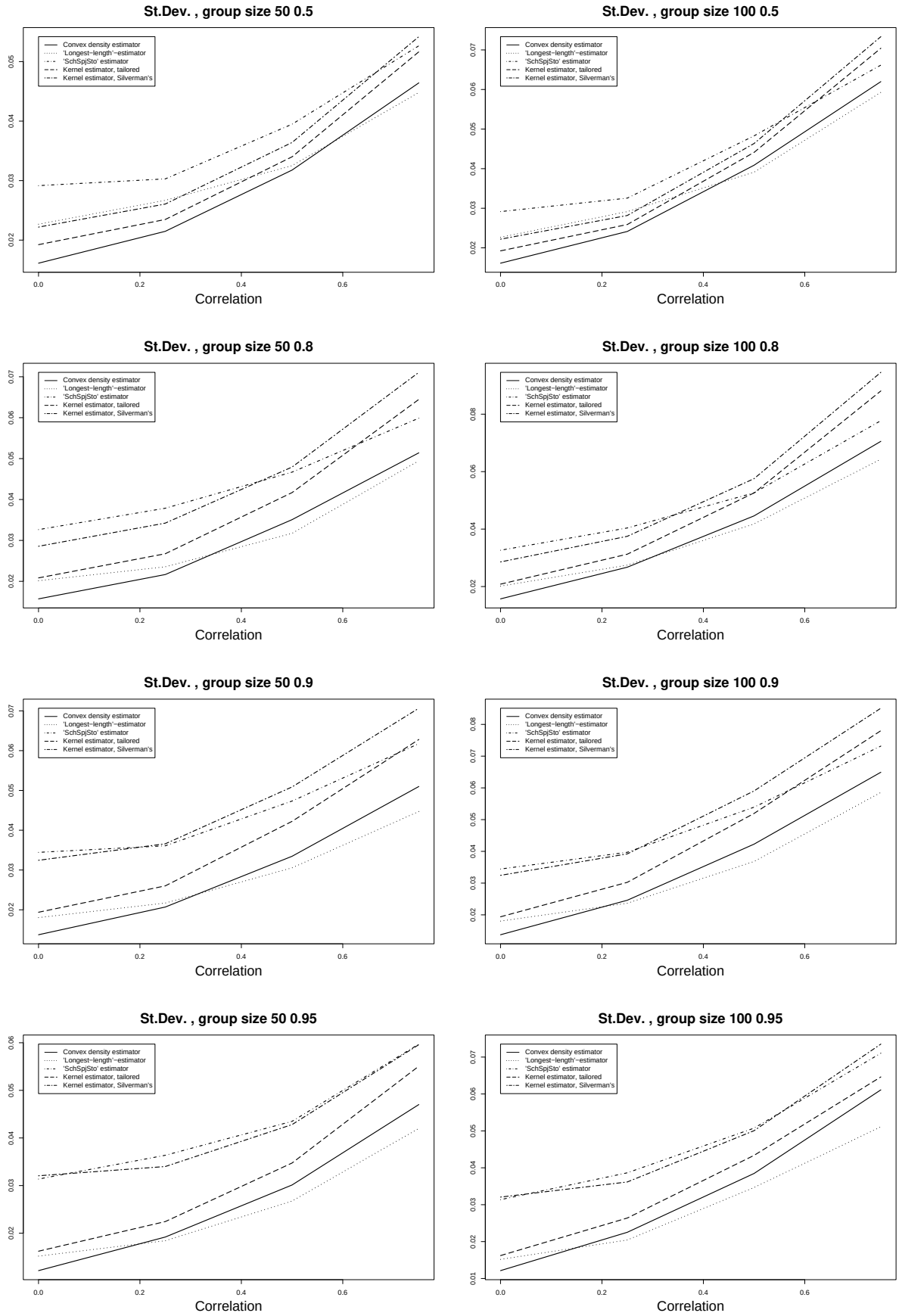
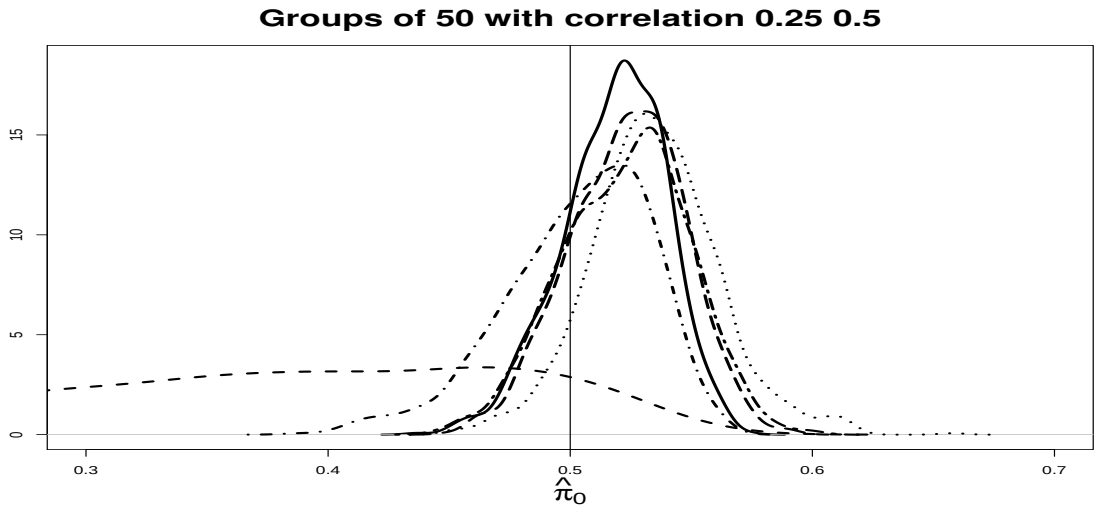
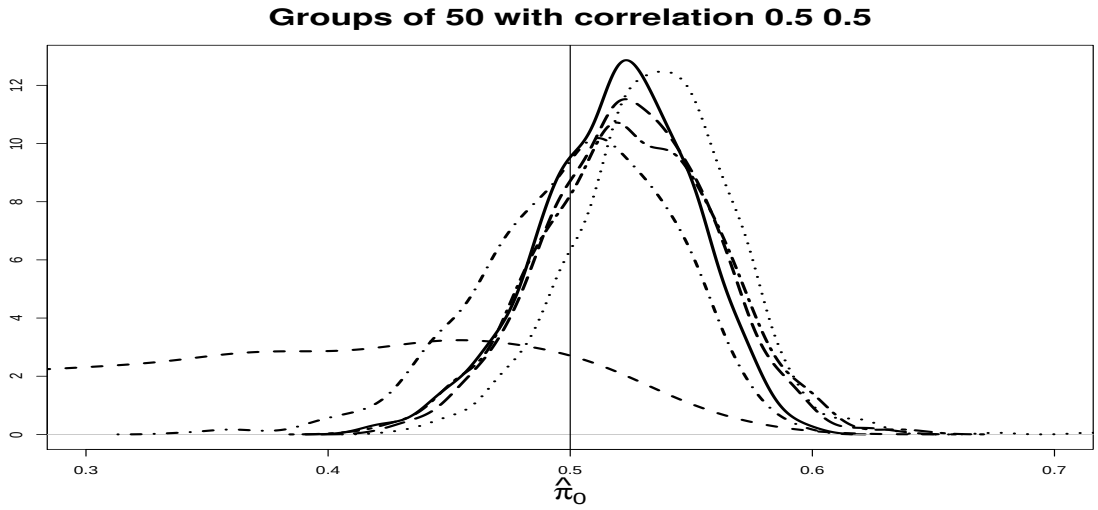


Figure 24: Standard deviation (in data sets of N=1000) as a function of correlation for group sizes 50 and 100 and values of $\pi_0 \in \{0.5, 0.8, 0.9, 0.95\}$ for five of the six methods considered (the Grenander estimator is excluded from the plot).





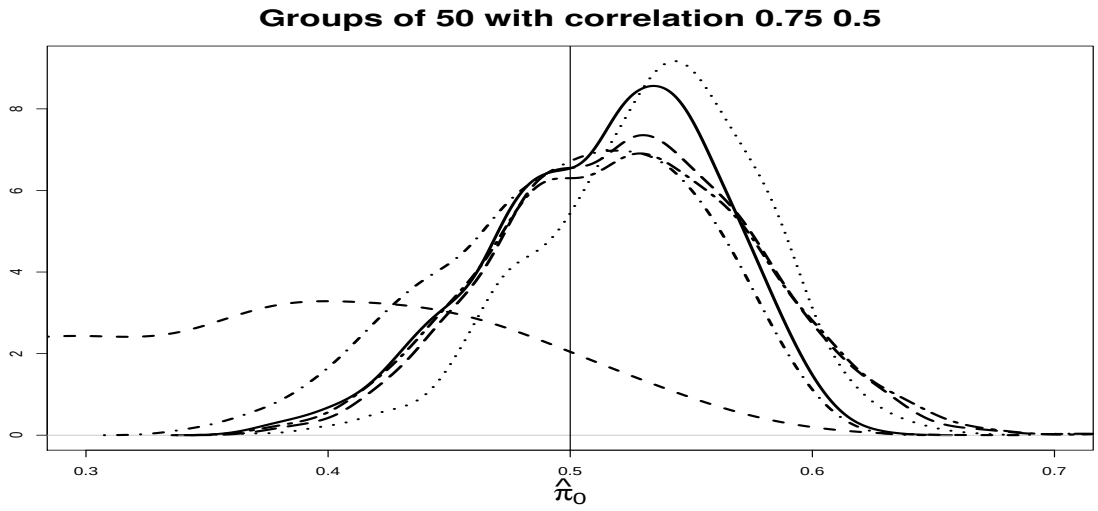
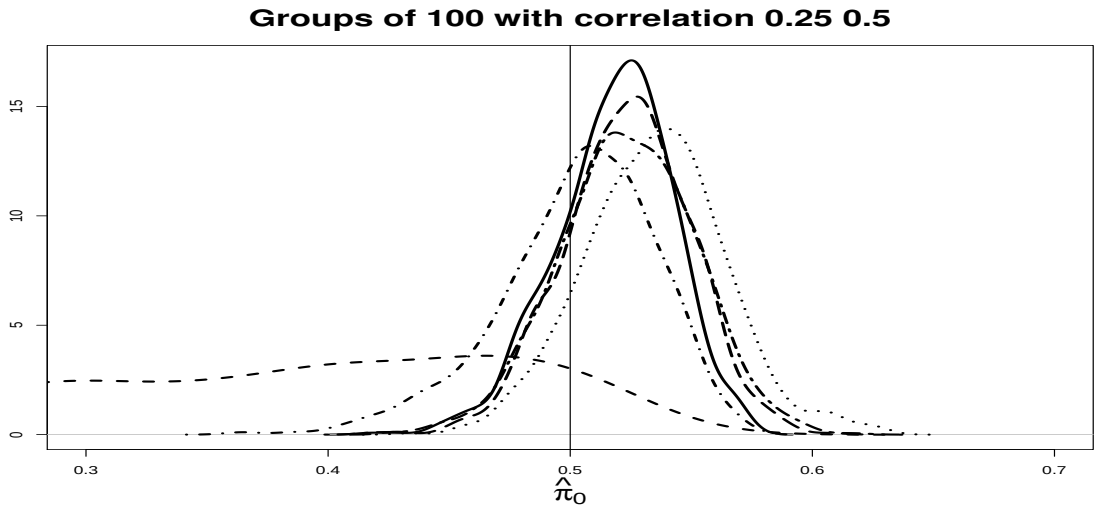
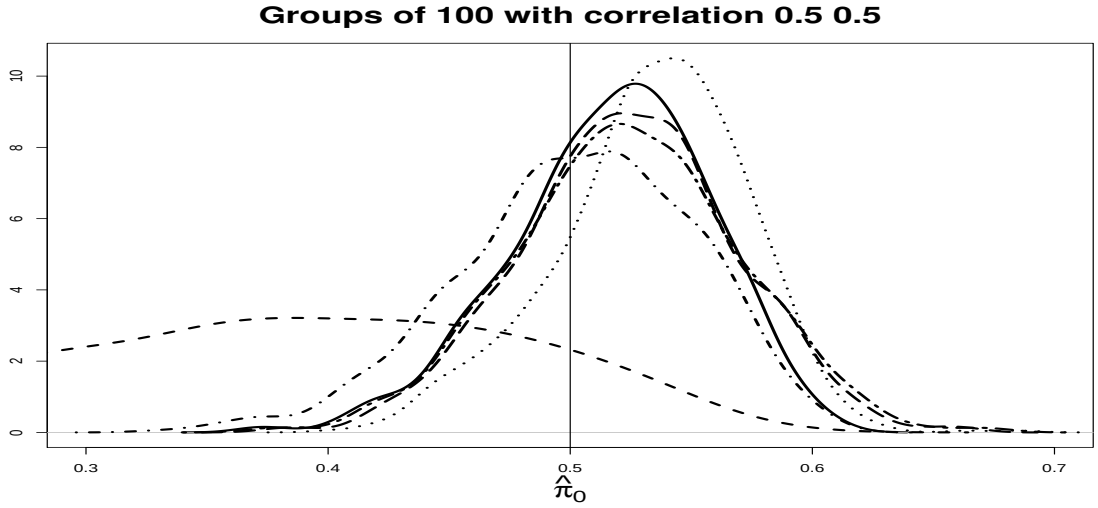


Figure 25: Density estimates of $\hat{\pi}_0$ for group size 50 for π_0=0.5. "Convex" is solid, "Grenander" is dashed, "Longest-length" is dotted, "SchSpjSto" is dotdash "Kernel, tailord" is longdash and "Kernel, Silverman" is twodash (plotting symbols are shown in Figure 9).





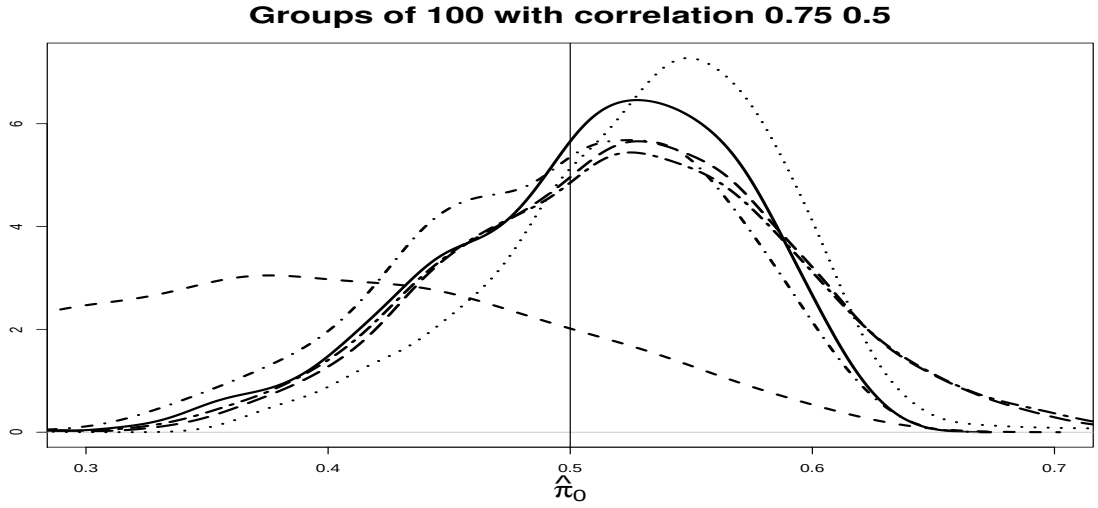
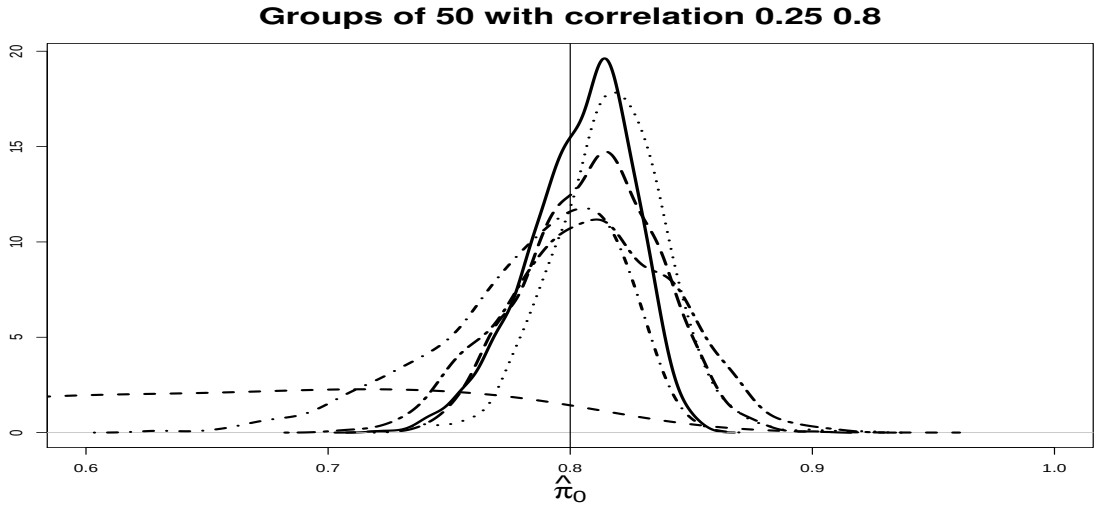
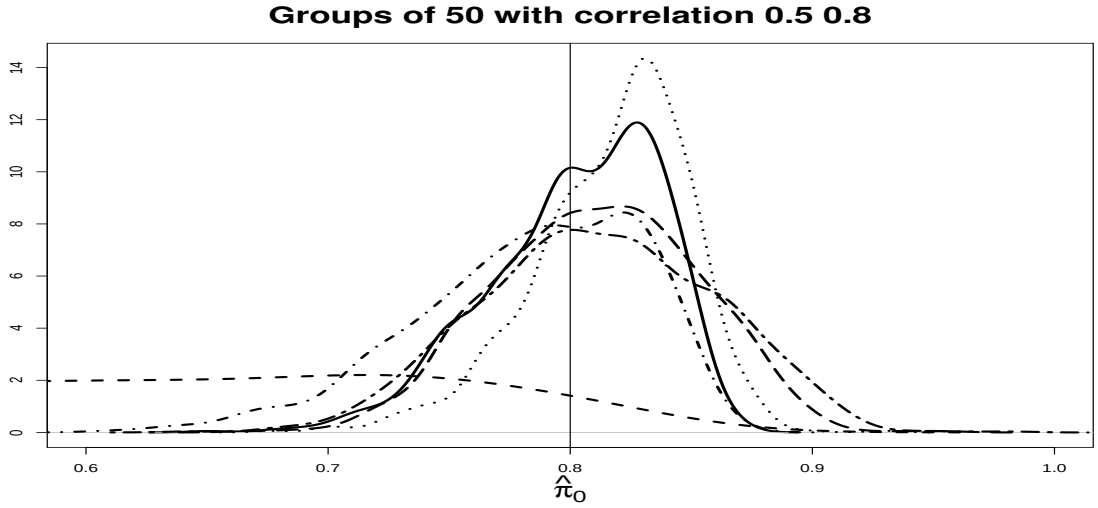


Figure 26: Density estimates of $\hat{\pi}_0$ for group size 100 for π_0=0.5. "Convex" is solid, "Grenander" is dashed, "Longest-length" is dotted, "SchSpjSto" is dotdash "Kernel, tailord" is longdash and "Kernel, Silverman" is twodash (plotting symbols are shown in Figure 9).





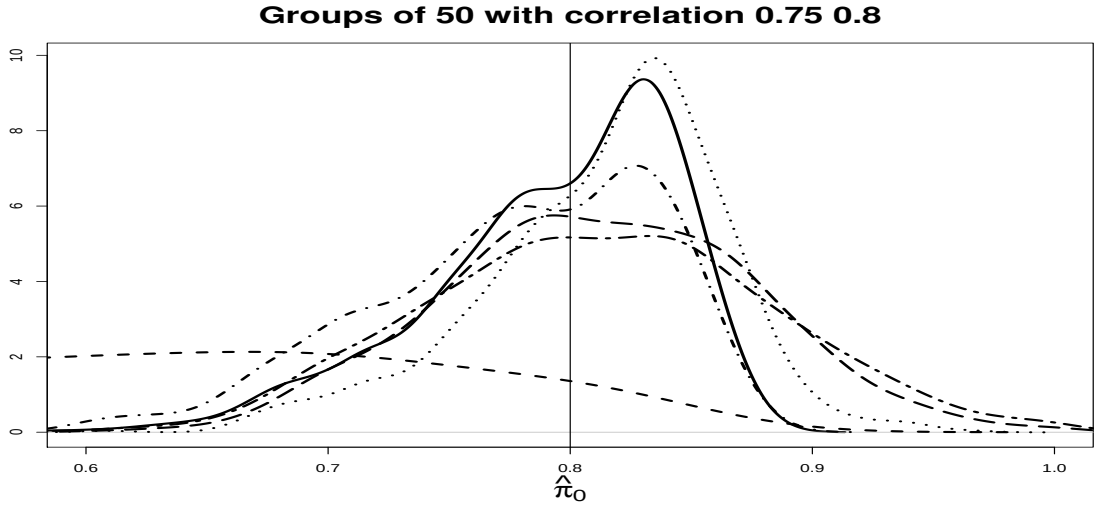
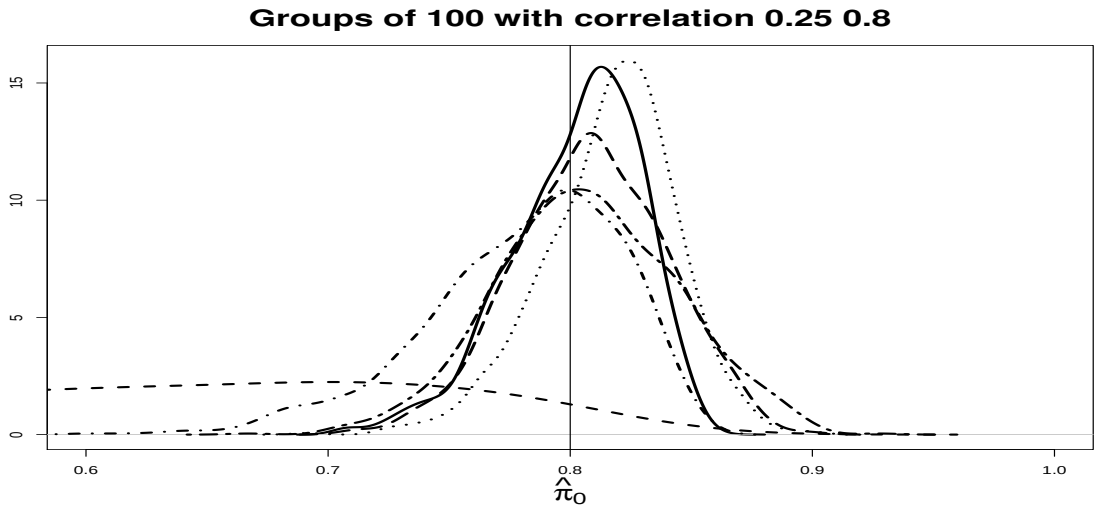
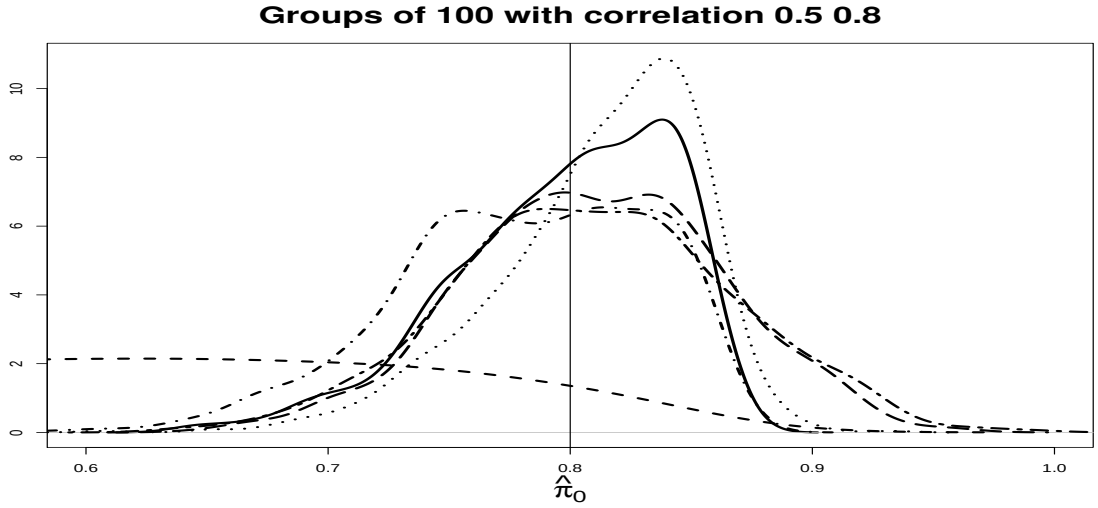


Figure 27: Density estimates of $\hat{\pi}_0$ for group size 50 for π_0=0.8. "Convex" is solid, "Grenander" is dashed, "Longest-length" is dotted, "SchSpjSto" is dotdash "Kernel, tailord" is longdash and "Kernel, Silverman" is twodash (plotting symbols are shown in Figure 9).





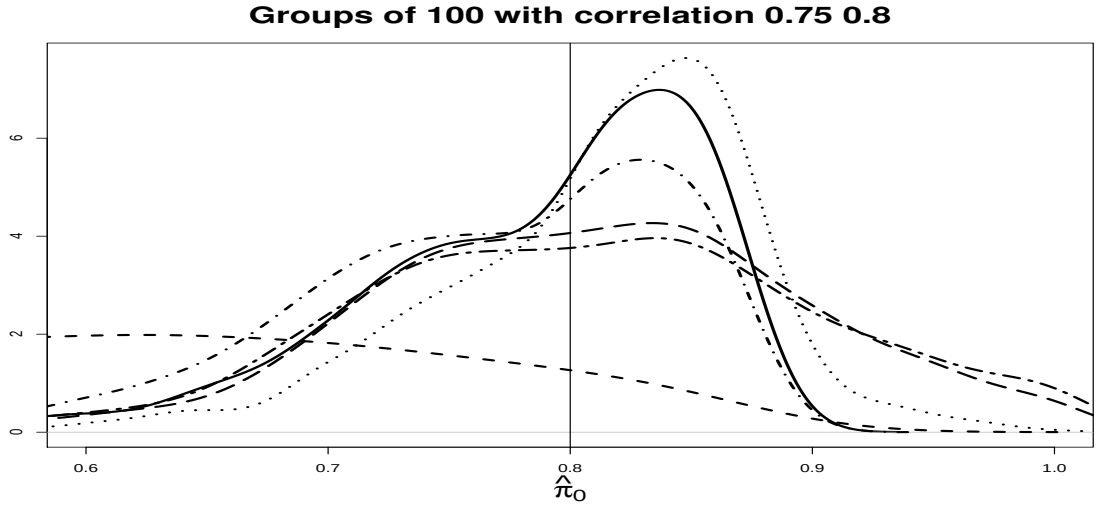
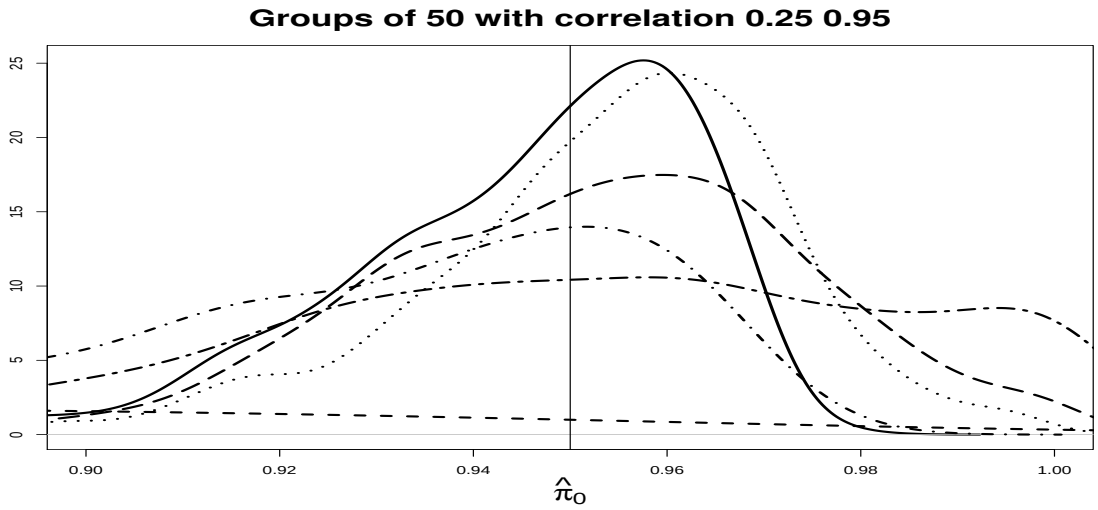
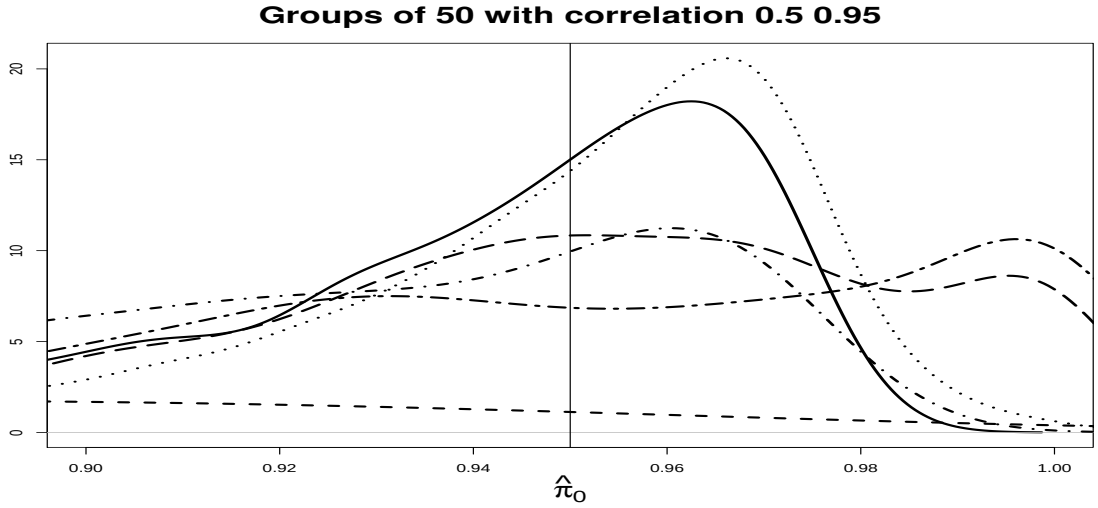


Figure 28: Density estimates of $\hat{\pi}_0$ for group size 100 for π_0=0.8. "Convex" is solid, "Grenander" is dashed, "Longest-length" is dotted, "SchSpjSto" is dotdash "Kernel, tailord" is longdash and "Kernel, Silverman" is twodash (plotting symbols are shown in Figure 9).





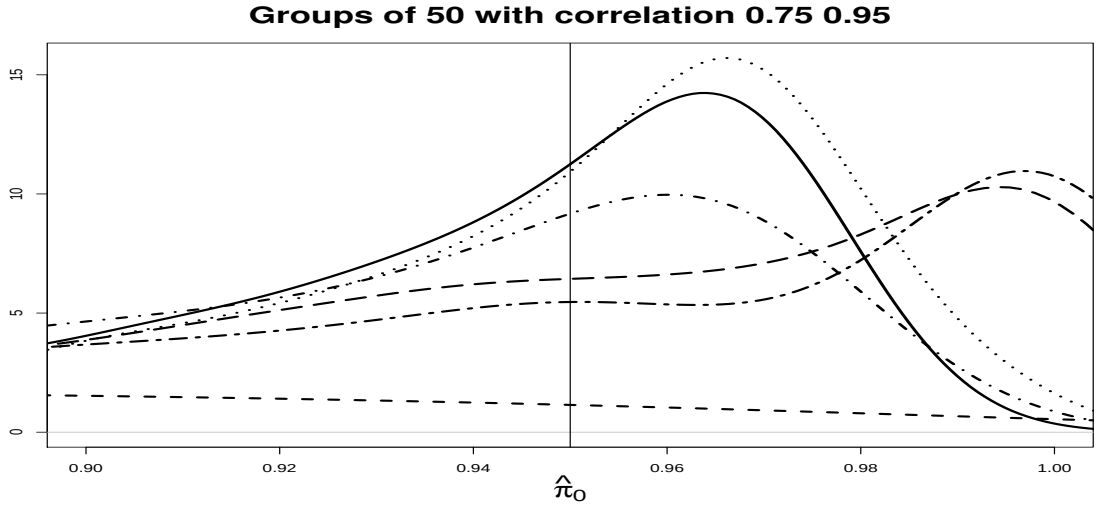
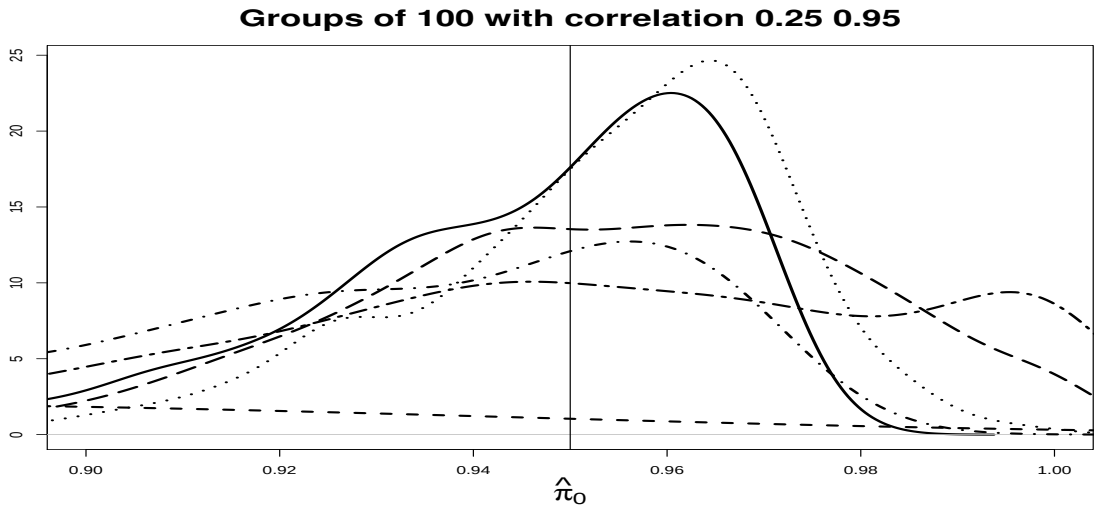
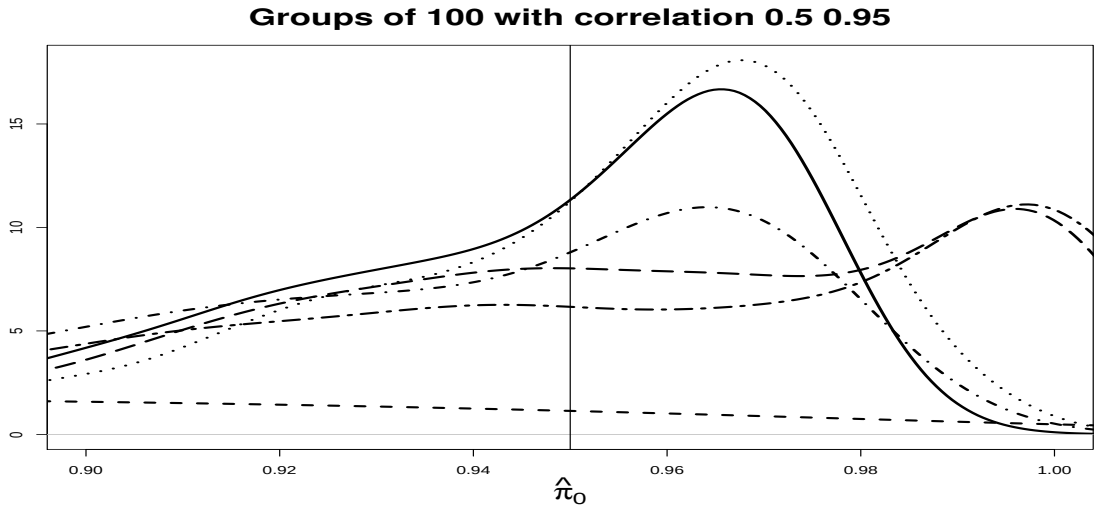


Figure 29: Density estimates of $\hat{\pi}_0$ for group size 50 for π_0=0.95. "Convex" is solid, "Grenander" is dashed, "Longest-length" is dotted, "SchSpjSto" is dotdash "Kernel, tailord" is longdash and "Kernel, Silverman" is twodash (plotting symbols are shown in Figure 9).





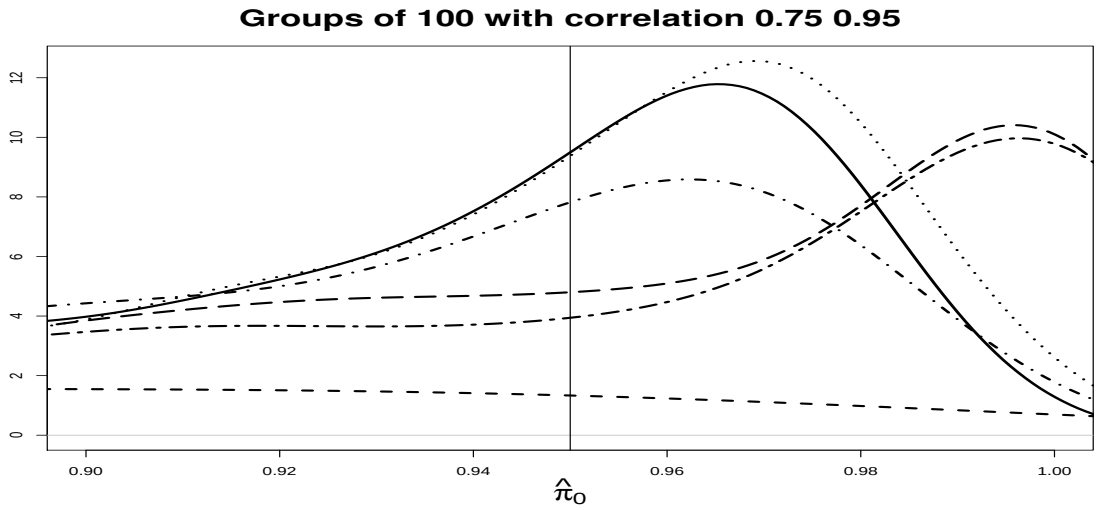


Figure 30: Density estimates of $\hat{\pi}_0$ for group size 100 for π_0=0.95. “Convex” is solid, “Grenander” is dashed, “Longest-length” is dotted, “SchSpjSto” is dotdash “Kernel, tailord” is longdash and “Kernel, Silverman” is twodash (plotting symbols are shown in Figure 9).

B R source code

Here all R source code for calculating the estimators of π_0 is listed. All code is written by Egil Ferkingstad, except

- the code for the PAVA algorithm (the function `pava`), written by R. F. Raubertas, and
- the code for calculating Storey's estimate, written by John D. Storey (available at http://www.stat.berkeley.edu/∼storey/).

```
#==============================================================================
# Estimation methods for pi0:
#
# grenest(p): Grenander estimate at p=1
# longest(p): Grenander estimate and the 'longest-length' idea
# convest(p): Convex decreasing density estimation
# kernest.sm(p): Kernel density estimation with Silverman's choice of h
# kernest.my(p): Kernel density estimation with the alternative choice of h
# storeyest(p): Schweder & Spjøtvoll's est. with Storey's bootstrap choice of lambda
#
# All the function take a vector with the observed p-values as input,
# and return the estimate of pi0.
#==============================================================================



########################################
# CONVEX DECREASING DENSITY ESTIMATION #
########################################

convest <- function(p)
# Estimates pi0 using a convex decreasing density estimate
# Input: Observed p-values
# Returns: An estimate of pi0
  {
    delta <- .00001
    k <- 200
    ny <- 1e-6
    p <- sort(p)
    m <- length(p)
    p.c <- ceiling(100*p)/100
    p.f <- floor(100*p)/100
    t.grid <- (1:100)/100
    x.grid <- (0:100)/100
    t.grid.mat <- matrix(t.grid,ncol=1)
    f.hat <- rep(1,101) #f.hat at the x-grid
    f.hat.p <- rep(1,m) #f.hat at the p-values
    theta.hat <- 0.01*which.max(
     apply(t.grid.mat,1,function(theta) sum((2*(theta-p)*(p<theta)/theta^2))))
    # f.theta.hat at the x-grid
    f.theta.hat <- 2*(theta.hat-x.grid)*(x.grid<theta.hat)/theta.hat^2
    # f.theta.hat at the p-vales
    f.theta.hat.p <- 2*(theta.hat-p)*(p<theta.hat)/theta.hat^2
    i<-1
    j<-0
    z<-1
    thetas <- numeric()
    for(j in 1:k) {
      if (sum((f.hat.p-f.theta.hat.p)/f.hat.p)>0) eps <- 0
        else
          {
            l <- 0
            u <- 1
            while (abs(u-l)>ny)
              {
                eps <- (l+u)/2
                if (sum(((f.hat.p-f.theta.hat.p)/
                 ((1-eps)*f.hat.p+eps*f.theta.hat.p))[f.hat.p>0])<0) l <- eps
                else u <- eps
              }
          }
        #if (theta.hat>0 & eps>0) j<-j+1
        f.hat <- (1-eps)*f.hat + eps*f.theta.hat
        pi.0.hat <- f.hat[101]
        d <- -sum((f.theta.hat.p-f.hat.p)/f.hat.p)
        f.hat.p <- 100*(f.hat[100*p.f+1]-f.hat[100*p.c+1])*(p.c-p)+f.hat[100*p.c+1]
        theta.hat <- 0.01*which.max(apply(t.grid.mat,1,function(theta)
             sum((2*(theta-p)*(p<theta)/theta^2)/f.hat.p)))
        f.theta.hat <- 2*(theta.hat-x.grid)*(x.grid<theta.hat)/theta.hat^2
        f.theta.hat.p <- 2*(theta.hat-p)*(p<theta.hat)/theta.hat^2
        if (sum(f.theta.hat.p/f.hat.p)<sum(1/f.hat.p))
          {
```

```
            theta.hat <- 0
            f.theta.hat <- rep(1,101)
            f.theta.hat.p <- rep(1,m)
          }
        if (sum(thetas==theta.hat)==0)
          {
            thetas[i] <- theta.hat
            thetas <- sort(thetas)
            i <- i + 1
          }
    z <- z+1
    }
    pi.0.hat <- f.hat[101]
    pi.0.hat
  }

##############################
# KERNEL DENSITY ESTIMATION #
##############################

kernest.sm <- function(p)
# Estimation of pi0 using kernel density estimation
# with Silverman's choice of smoothing parameter
# Input: Vector of observed p-values
# Returns: An estimate of pi0
  {
    m <- length(p)
    n <- 2*m
    pa <- p
    pa[(m+1):n] <- 2-p
    h <- 0.9*min(sd(pa),(quantile(pa,.75)-quantile(pa,.25))/1.34)*n^(-1/5)
    dens <- n^-1*h^-1*(2*pi)^-(1/2)*sum(exp(-(1/2)*((1-pa)/h)^2))
    pi.0.hat <- 2*dens
    #cat("pi.0.hat:",pi.0.hat,"\th:",h,"\n")
    pi.0.hat
  }

kernest.my <- function(p)
# Estimation of pi0 using kernel density estimation
# with Ferkingstad's choice of smoothing parameter
# Input: Vector of observed p-values
# Returns: An estimate of pi0
 {
    m <- length(p)
    n <- 2*m
    pa <- p
    pa[(m+1):n] <- 2-p
    c.hat <- -m/sum(log(p[p>0])) # the maximum likelihood estimate of c
    h <- (4*pi)^(-1/10)*(c.hat*(c.hat-1)*(c.hat-2))^-(2/5)*n^(-1/5)*c.hat^(1/5)
    dens <- n^-1*h^-1*(2*pi)^-(1/2)*sum(exp(-(1/2)*((1-pa)/h)^2))
    pi.0.hat <- 2*dens
    #cat("pi.0.hat:",pi.0.hat,"\tc.hat:",c.hat,"\th:",h,"\n")
    pi.0.hat
  }

############################################################
# DECREASING (GRENANDER) DENSITY ESTIMATES (WITH LONGEST) #
############################################################


grenlongest <- function(p)
  # Input: Vector of observed p-values
  # Returns: Vector with grenest as first element and longest as second argument
  {
    p <- sort(p)
    pi.0.hats <- numeric()
    npmle <- grenander(p)
    n <- length(npmle)
    pi.0.hats[1] <- min(npmle[n],1)
    pi.0.hats[2] <- findlong.e(npmle,p)
    pi.0.hats
  }


grenest <- function(p)
# Grenander estimate
# Input: Vector of observed p-values
# Returns: An estimate of pi0
  {
    npmle <- grenander(p)
    n <- length(npmle)
    pi.0.hat <- npmle[n]
    pi.0.hat
  }

longest <- function(p)
# 'Longest-length estimate
# Input: Vector of observed p-values
# Returns: An estimate of pi0
   {
```

```
     npmle <- grenander(p)
     pi.0.hat <- findlong.e(npmle,p)
     pi.0.hat
   }

grenander <- function(p)
# Compute the nonparametric maximum likehood decreasing density estimate f.gren
# of the p-values. Uses the function pava.R
# Input: Vector of observed p-values
# Returns: Vector with NPMLE decreasing density estimate
  {
    p <- sort(p)
    y <- w <- npmle <- numeric()
    y <- 1/(length(p)*c(p[1],diff(p)))
    w <- c(p[1],diff(p))
    npmle <- pava(y,w)
    npmle
  }

pava <- function(x, wt=rep(1,length(x)))
# Compute the antitonic regression of numeric vector 'x', with
# weights 'wt', with respect to simple order. The pool-adjacent-
# violators algorithm is used. Returns a vector of the same length
# as 'x' containing the regression.

# 02 Sep 1994 / R.F. Raubertas

# Modified by Egil Ferkingstad (changed from isotonic to antitonic regression)

{
   n <- length(x)
   if (n <= 1) return (x)
   if (any(is.na(x)) || any(is.na(wt))) {
      stop ("Missing values in 'x' or 'wt' not allowed")
   }
   x[x==Inf] <- 10^100
   lvlsets <- (1:n)
   repeat {
     viol <- (as.vector(diff(x)) > 0) # Find adjacent violators
      if (!(any(viol))) break

      i <- min( (1:(n-1))[viol]) # Pool first pair of violators

      lvl1 <- lvlsets[i]
      lvl2 <- lvlsets[i+1]
      ilvl <- (lvlsets == lvl1 | lvlsets == lvl2)
      x[ilvl] <- sum(x[ilvl]*wt[ilvl]) / sum(wt[ilvl])
      lvlsets[ilvl] <- lvl1
   }
   x
}

findlong.e <- function(x,p)
  {
    pi.0.hat <- 2
    n <- length(x)
    p <- sort(p)
    nlto <- sum(x<=1)
    if (nlto<=1) pi.0.hat<-1
    else
      {
        x <- x[(n-nlto):n]
        p <- p[(n-nlto):n]
        ix <- 1:(length(x)-1)
        ch.pt <- unique(c(ix[diff(x)!=0],max(ix)))
        diffs <- diff(p[ch.pt+1])
        ch.idx <- which.max.last(diffs)
        long.idx <- ch.pt[ch.idx]
        pi.0.hat <- x[long.idx+1]
      }
    pi.0.hat
  }


which.max.last <- function(x)
{
    y <- seq(length(x))[x == max(x)]
        if(length(y) > 1)
                max(y)
        else y
}

########################################
# SCWEDER/SPJØTVOLL/STOREY ESTIMATION #
########################################

storeyest <- function(p)
  {
    qvalue(p,lam.meth="bootstrap")$pi0
  }
```

```
# CODE WRITTEN BY JOHN D. STOREY, AVAILABLE AT:
# http://www.stat.berkeley.edu/~storey/

qvalue <- function(p, alpha=NULL, lam=NULL, lam.meth="smoother", robust=F) {
#This is a function for estimating the q-values for a given set of p-values. The
#methodology mainly comes from:
#Storey JD. (2002) A direct approach to false discovery rates.
#Journal of the Royal Statistical Society, Series B, 64: 479-498.
#See http://www.stat.berkeley.edu/~storey/ for more info.
#This function was written by John D. Storey. Copyright 2002 by John D. Storey.
#All rights are reserved and no responsibility is assumed for mistakes
#in or caused by
#the program.
#
#Input
#=============================================================================
#p: a vector of p-values (only necessary input)
#alpha: a level at which to control the FDR (optional)
#lam: the value of the tuning parameter to estimate pi0 (optional)
#lam.method: either "smoother" or "bootstrap"; the method for automatically
#           choosing tuning parameter lam if it is not specified
#robust: an indicator of whether it is desired to make the estimate more robust
#         for small p-values (optional)
#
#Output
#=============================================================================
#remarks: tells the user what options were used, and gives any relevant warnings
#pi0: an estimate of the proportion of null p-values
#qvalues: a vector of the estimated q-values (the main quantity of interest)
#pvalues: a vector of the original p-values
#significant: if alpha is specified, and indicator of
# whether the q-value fell below alpha
#    (taking all such q-values to be significant controls FDR at level alpha)

#This is just some pre-processing
    if(min(p)<0 || max(p)>1) {
    print("ERROR: p-values not in valid range"); return(0)
    }
    m <- length(p)
#These next few functions are the various ways to estimate pi0
    if(!is.null(lam)) {
        pi0 <- mean(p>lam)/(1-lam)
        pi0 <- min(pi0,1)
        remark <- "The user prespecified lam in the calculation of pi0."
    }
    else{
        lam <- seq(0,0.95,0.01)
        pi0 <- rep(0,length(lam))
        for(i in 1:length(lam)) {
            pi0[i] <- mean(p>lam[i])/(1-lam[i])
        }
        if(lam.meth=="smoother") {
            remark <- "A smoothing method was used in the calculation of pi0."
            library(modreg)
            spi0 <- smooth.spline(lam,pi0,df=3,w=(1-lam))
            pi0 <- predict.smooth.spline(spi0,x=0.95)$y
            pi0 <- min(pi0,1)
        }
        if(lam.meth=="bootstrap") {
            remark <- "A bootstrap method was used in the calculation of pi0."
            minpi0 <- min(pi0)
            mse <- rep(0,length(lam))
            pi0.boot <- rep(0,length(lam))
            for(i in 1:100) {
                p.boot <- sample(p,size=m,replace=T)
                for(i in 1:length(lam)) {
                    pi0.boot[i] <- mean(p.boot>lam[i])/(1-lam[i])
                }
                mse <- mse + (pi0.boot-minpi0)^2
            }
            pi0 <- min(pi0[mse==min(mse)])
            pi0 <- min(pi0,1)
        }
    }
    if(pi0 <= 0) {
    print("ERROR: Check that you have valid p-values. The estimated pi0 < 0."); return(0)
    }
#The q-values are actually calculated here
    u <- order(p)
    v <- rank(p)
    qvalue <- pi0*m*p/v
    if(robust) {
        qvalue <- pi0*m*p/(v*(1-(1-p)^m))
        remark <- c(remark, "The robust version of the q-value was
                    calculated. See Storey JD (2002) JRSS-B 64: 479-498.")
    }
    qvalue[u[m]] <- min(qvalue[u[m]],1)
    for(i in (m-1):1) {
    qvalue[u[i]] <- min(qvalue[u[i]],qvalue[u[i+1]],1)
```

```
    }
#Here the results are returned
    if(!is.null(alpha)) {
        return(remarks=remark, pi0=pi0, qvalues=qvalue,
               significant=(qvalue <= alpha), pvalues=p)
    }
    else {
        return(remarks=remark, pi0=pi0, qvalues=qvalue, pvalues=p)
    }
}
```