3 A review on false discovery rates

3.1 Introduction and outline

The arrival of the microarray technology brought new challenges to the statistical community. A microarray measures the expression of thousands of genes at the same time. With these micro devices it is now possible to compare the gene expression of different tissue samples in a comprehensive way. For each gene we might ask: is there a difference between the tissues? And: for a single gene, do we have evidence of observing a reliable difference?

The first question is answered by computing the difference in mean gene expression between the different tissue samples. The significance of the observed difference is expressed by computing an empirical p-value as in Equation (2.13). The smaller the p-value, the more evidence we have that we really observed a significant difference and not just random noise. Surely, biological data suffers from all kinds of noise contaminations and it is questionable whether even a small p-value provides reliable evidence. Thus we need to employ a filter mechanism that safeguards against random noise. The classical approach in multiple testing theory is to define an adjustment procedure, which we term a *p-value filter*, that leaves only genes with p-values lower than a certain threshold. The final list of differentially expressed genes contains only those genes with low p-values. If we draw the histogram of p-values as shown in Figure 3.1, the p-value filter corresponds to a vertical line dividing the p-value range into two parts: the left part supporting differential expression, the right part not supporting differential expression.

The p-value filter works well for a small number of genes. With present microarrays researchers measure the expression of several tens of thousands of genes simultaneously. The large number of genes brings along a second problem that is more severe than random noise: often a large percentage of the genes behaves

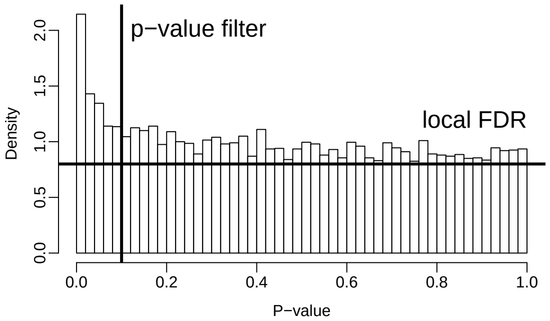

Figure 3.1: Exemplary distribution of p-values in a microarray experiment. The two solid lines correspond to two opposite approaches. The vertical line is linked to classical p-value filters, which divide the range of p-values into one part supporting differential expression and into another part not supporting it. The horizontal line corresponds to the estimation of the local false discovery rate: the p-value distribution is divided into a uniform block of consistent genes and a rest indicating differential expression. Local false discovery rates provide a global view on differential gene expression over the whole range of p-values while filters concentrate only on genes on the left-hand side of the vertical line.

consistently among the tissue samples. We call those genes consistent that are not differentially expressed. Due to random fluctuation in the data a consistent gene might get a low p-value simply by chance. Theory suggests that if we only observed data of consistent genes, their p-values would be uniformly distributed in the resulting histogram. This uniform distribution is indicated by the horizontal line in Figure 3.1. Whenever we observe a uniform block of p-values in the histogram, the gene set contains a certain percentage of consistent genes. The height of the uniform block corresponds to this percentage. The histogram parts above the horizontal line provide the number and distribution of the differentially expressed genes. If we now return to the set of genes on the left side of the histogram, which passed the p-value filter, we observe that we included a reasonable percentage of consistent genes. Even if we restrict the filter to smaller p-values, we cannot omit the inclusion of a uniform part. However, in this example it is more likely that a gene with a small p-value is differentially expressed than a gene with a higher p-value. Even genes with p-values exceeding 0.2 seem not to correspond to consistent genes only. Yet they have a higher probability of not being differentially expressed than genes with p-values passing the filter. Assigning such a probability to every gene answers the second question above.

The probability of not being differentially expressed is measured by the *local false discovery rate*. Estimating the local false discovery rate means to draw a horizontal line in the p-value distribution instead of a vertical line as a p-value filter does. Thus the idea of the local false discovery rate is totally opposite to the idea of p-value filters. Local false discovery rates are perfectly suited for large microarray experiments: the larger the number of genes is, the more accurately the rate can be estimated. When analyzing microarray data, the estimation of the local false discovery rate is the natural procedure to explore differential gene expression.

In the following, we review the concepts and ideas behind the false discovery rate in Section 3.2. Global false discovery rates define special kinds of p-value filters and we show in Section 3.3 why such a p-value filter may not be the best choice for the analysis of microarray data. In Section 3.4 we introduce the local variant of the false discovery rate and close with a comprehensive overview on various estimators of the local false discovery rate.

3.2 The false discovery rate

In the previous section, we introduced the concept of p-value filters. With p-value filters, significance is based on the size of a p-value: the smaller the p-value, the higher the evidence for a significant difference in gene expression. The concept of p-value filters was developed to safeguard against random noise, which often cannot be avoided in biological data. The filter divides the distribution of p-values vertically into two parts: the left part corresponds to differential expression, the right part corresponds to non-differential expression, see Figures 3.1 and 3.2. There exist various classical concepts for the definition of p-value filters. For details on these concepts, we refer to the paper series of Dudoit *et al.* (2004) and van der Laan *et al.* (2004).

A novel concept of p-value filters was introduced in the seminal work of Benjamini and Hochberg (1995): the *false discovery rate*. The authors intended to draw the vertical line such that the resulting list of differentially expressed genes does only contain a certain rate of genes that are truly not differentially expressed. As p-values of non-induced genes take by chance any value in the interval $[0, 1]$, they are uniformly distributed. Hence the fraction of non-induced genes for a given vertical separation at value p relates to the rectangular area that is marked with $V(p)$ in Figure 3.2. These genes are called *false positives* because they appear in the list of induced genes, thus being *positive*, but they are there just by chance

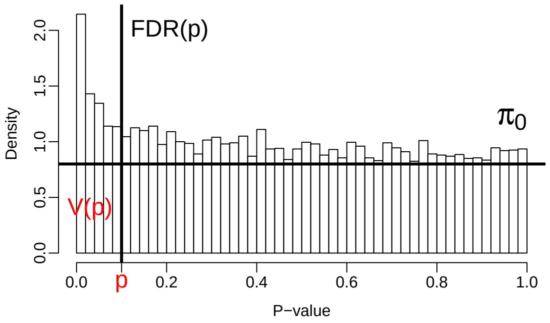

Figure 3.2: Exemplary distribution of p-values in a microarray experiment. As depicted in Figure 3.1, the analysis methods for microarray data divide into two opposite classes: the false discovery rate of Benjamini and Hochberg (1995) is a p-value filter and thus corresponds to the vertical separation drawn at a value of p. The gene set thus includes a certain amount $V(p)$ of genes being non-induced. The method of Storey (2003) incorporates the estimation of the fraction π_0 of non-induced genes and thus relates to the horizontal separation line.

and not because of true induction. Benjamini and Hochberg defined the false discovery rate as the "expected rate of false positives among all positive genes".

Let P_i be the random variable of the ith p-value. The number of all positives $R(p)$, that is the size of the list of induced genes, and the number of false positives $V(p)$ are defined as

$$R(p) = \sum_{i=1}^{m} I\{P_i \leq p\} \quad \text{and} \quad V(p) = \sum_{i=1}^{m} I\{P_i \leq p, H = 0\}, \tag{3.1}$$

where p is the value at which the vertical line is drawn. Then the false discovery rate is defined as the expected rate of false positives among all positives, that is

$$\text{FDR}(p) = \text{E}\left[\frac{V(p)}{R(p)} I\{R(p) > 0\}\right]. \tag{3.2}$$

The false discovery rate of Benjamini and Hochberg (1995) is the expectation of the ratio of $V(p)$ and $R(p)$ with the natural restriction that the list of induced genes is not empty, that is $R(p) > 0$. Equation (3.2) can be rewritten as a conditioned expectation

$$\text{FDR}(p) = \text{E}\left[\frac{V(p)}{R(p)} \mid R(p) > 0\right] Pr[R(p) > 0], \tag{3.3}$$

with $Pr[R(p) > 0]$ being the probability that at least one positive gene occurs. The authors developed a procedure to estimate the false discovery rate for every choice of p. Let $p_1, \ldots, p_m$ be the set of observed p-values. Let $p_{(1)}, \ldots, p_{(m)}$ be the set of ordered p-values such that $p_{(1)} \leq p_{(2)} \leq \cdots \leq p_{(m)}$. If $i^\star$ is the largest index i for which

$$p_{(i)} \leq \frac{i}{m}\alpha, \tag{3.4}$$

then the set of genes with ordered p-values $p_{(1)}, \ldots, p_{(i^\star)}$ defines the set of induced genes with a false discovery rate equal or below $\alpha \cdot 100\%$. With this procedure, a certain choice of FDR $\leq \alpha$ leads to a certain value $p = p_{(i^\star)}$ and thus to a list of induced genes with p-values equal or below p.

A substantial change of viewpoint. The procedure defined in Equation (3.4) also works in reversed mode: for every value of p it leads to an estimated value of α. As the number of possible values p depends on the number of unique observed p-values, it is convenient to associate the resulting α to every single p-value. The estimated level α is called ***FDR-adjusted p-value***. For each ordered p-value $p_{(i)}$, the FDR-adjusted p-value $\tilde{p}_{(i)}$ is given as

$$\tilde{p}_{(i)} = \min_{k=i,\ldots,m}\left\{\min\left(\frac{m}{k}p_{(k)}, 1\right)\right\}. \tag{3.5}$$

The term $\frac{m}{k}p_{(k)}$ is the upper bound for α and is derived from inverting $p_{(k)} \leq \frac{k}{m}\alpha$ in Equation (3.4). The inner minimum in Equation (3.5) prevents from FDR-adjusted p-values greater than one. The outer minimum causes the values to increase monotonically. Now each gene is associated with an estimated false discovery rate, which is reached if all genes with lower or equal FDR-adjusted p-values pass the filter. With these adjustments one can conveniently explore how the inclusion or exclusion of a certain gene changes the estimated false discovery rate. The switch from a fixed false discovery rate threshold to an estimated false discovery rate for each genes is an important step that is directly connected to the q-value concept of Storey (2003), which we introduce in the following. FDR-adjusted p-values shift the emphasis from hypothesis-based inference to an inference based on the observed set of p-values. This results in shifting the line of separation in Figure 3.2.

Switching from vertical to horizontal separation. Storey (2003) claimed that the false discovery rate as defined above is “the rate that false discoveries occur”.

Storey argued that one is more interested in "the rate that discoveries are false", which relates to shifting the emphasis from the vertical line drawn by a p-value filter to the horizontal line estimating the overall amount of non-induced genes. Storey proposed the use of the *positive* false discovery rate, which relates to one definition of the "rate of false positives" in Benjamini and Hochberg (1995). The positive false discovery rate is defined as

$$\mathrm{pFDR}(p) = \mathrm{E}\left[\frac{V(p)}{R(p)} \mid R(p) > 0\right]. \tag{3.6}$$

The definition of the positive false discovery rate equals the definition of the false discovery rate in Equation (3.3) except for lacking the factor $Pr[R(p) > 0]$, which is the probability of observing at least one differentially expressed gene. In a usual experiment we might assume $Pr[R(p) > 0] = 1$ such that FDR = pFDR and we can examine the rate that discoveries are false. However, Storey argues that there are experiments where $Pr[R(p) > 0] = 1$ does not hold because of weak signals or weak induction. Say $Pr[R(p) > 0] = 0.5$ and we are interested in a (positive) false discovery rate of 5%. If we adjust the p-values according to Benjamini and Hochberg (1995) as given above, we actually control the positive false discovery rate at 10% because $Pr[R(p) > 0] \cdot \mathrm{pFDR} = 0.5 \cdot 10\% = 5\%$.

Further, Storey (2003) showed that the positive false discovery rate is equivalent to the *conditional* false discovery rate also suggested in Benjamini and Hochberg (1995):

$$\mathrm{cFDR}(p) = \mathrm{E}\left[\frac{V(p)}{R(p)} \mid R(p) = r(p)\right] \tag{3.7}$$

$$= \frac{\mathrm{E}\left[V(p) \mid R(p) = r(p)\right]}{r(p)}, \tag{3.8}$$

where $r(p) > 0$ denotes the actually observed number of positives at value p. We refer to Tsai *et al.* (2003) for a comparative review on the three false discovery rate variants.

Storey also introduced the term *q-value*. Similar to FDR-adjusted p-values, the q-value is assigned to each gene and denotes the estimated positive false discovery rate that we can reach if we include this gene into the list of differentially expressed genes. Storey showed that the positive false discovery rate can be rewritten in terms of conditional probabilities such that

$$\text{pFDR}(p) = Pr\left[H = 0 \,|\, P \leq p\right] \tag{3.9}$$

$$= Pr\left[H = 0\right] \frac{Pr\left[P \leq p \,|\, H = 0\right]}{Pr\left[P \leq p\right]}. \tag{3.10}$$

The probability $Pr[H = 0]$ in Equation (3.10) is the probability of not being differentially expressed, which affects the probability of being non-induced conditioned on a p-value filter $P \leq p$ in Equation (3.9). Storey argues that a q-value is not a pFDR-adjusted p-value, which is due to the probability $Pr[H = 0]$. How does one interpret $Pr[H = 0]$? It is the overall probability of being non-induced in an observed microarray experiment. Thus it relates to the size of the uniform distribution of p-values from non-induced genes, that is the height of the horizontal line in Figure 3.2, which is denoted by

$$\pi_0 = Pr[H = 0]. \tag{3.11}$$

The height π_0 has to be estimated from the set of p-values, whereas FDR-adjusted p-values are derived by direct conversion of the p-values. Q-values are estimated positive false discovery rates. The computation of q-values given in Storey and Tibshirani (2003) equals the procedure of Benjamini and Hochberg (1995) in Equation (3.5) except for the inclusion of the estimated prior probability. The estimator will be introduced in Section 3.5, Equation (3.22). For the time being, assume we have an estimate $\widehat{Pr[H = 0]} = \widehat{\pi_0}$. For a set of ordered p-values $p_{(1)} \leq \cdots \leq p_{(m)}$ the q-value of the ith ordered p-value is given as

$$q_{(i)} = \min_{k=i,\ldots,m} \left\{ \min\left(\widehat{\pi_0} \frac{m}{k} p_{(k)}, 1 \right) \right\}. \tag{3.12}$$

FDR-adjusted p-values are a p-value filter method and thus relate to the vertical line in Figure 3.2. The same applies for q-values. They filter p-values with respect to the positive false discovery rate. Both false discovery rate variants result in one estimated value for a *set of genes* and do not provide single-gene information. We will discuss in the following section how this might give misleading results.

3.3 Pitfalls of global false discovery rates

The false discovery rate variants introduced in Section 3.2 work globally: for a vertical separation of the p-value distribution, the estimated positive false discovery rate of the resulting list of induced genes is the largest q-value of the genes in the list. The q-value is not a feature of a single gene but of the entire gene list. Finner and Roters (2001) criticize that one might easily include a non-induced gene into the list without changing the estimated false discovery rate too much. Consider the following example: assume we found 100 genes with p-values such that the estimated false discovery rate is below 1% if we draw a vertical line at value p. Say, 99 genes have p-values well below p whereas gene i's p-value is just slightly below p. By definition of the false discovery rate, we expect one false positive among the list of 100 genes. Intuitively we assume that gene i is the most likely candidate for being *the* false positive. However, as the estimated value is a property of the entire list we must not imply that gene i is non-induced. This first ambiguity calls for a local probability of being a false positive for each gene in the list—or better for all genes under study.

Along the same lines comes a second pitfall of global false discovery rates: q-values do not react immediately to the inclusion of false positives when we observe many highly induced genes. Assume we have a set of highly induced genes with low estimated positive false discovery rate. We might enlarge the list with some less induced genes before the positive false discovery rate reaches an undesirable value. Since the positive false discovery rate estimates are not fit to react instantly to a contamination with false positives, we need a probability measure that accounts for local changes in significance. We illustrate this second disadvantage of global false discovery rate estimates with a simulation. We randomly drew 10000 values from a standard normal distribution $\mathcal{N}(0,1)$ serving as scores of non-induced genes. From these, we chose 2000 values and induced them by adding a constant value μ. The parameter μ accounts for the strength of induction. We set $\mu \in \{2,3,4\}$. Thus the induced genes were distributed according to $\mathcal{N}(\mu,1)$. All scores were transformed to p-values with respect to the true null distribution $\mathcal{N}(0,1)$. We computed q-values using package *qvalue* by A. Dabney and J. D. Storey. For each choice of positive false discovery rate, that is for each observed q-value, we stored the number of false positives within the resulting list. The resulting curves of false positives up to a number of 200 against the respective

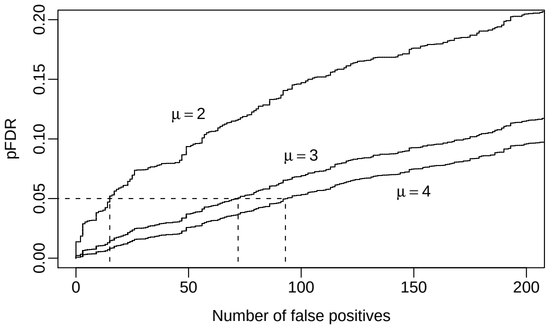

Figure 3.3: Reaction of global false discovery rate estimates to the inclusion of false positives. Shown are the results of a simulation study with 2000 induced genes among 10000. Parameter μ reflects the strength of induction. In experiments with high induction ($\mu = 4$), the q-values increase slowly such that the list of induced genes contains more false positives than in studies with less induction ($\mu < 4$). The choice of pFDR $= 5\%$ leads to 15, 72 and 93 false positives, respectively.

q-values are shown in Figure 3.3. Note that we kept the two sets of 2000 induced and 8000 non-induced genes fixed, which implies that the order of the false positives within the lists is independent of μ. For example, we included always the same set of 100 false positives into the list if we selected the positive false discovery rate with respect to this number. Figure 3.3 illustrates that one can include more false positives into the list if the data set contains highly induced genes. With the same choice of pFDR $= 5\%$, we received 15 false positives for a slight induction with $\mu = 2$ whereas $\mu = 4$ lead to 93 false rejections.

3.4 A local measure of significance

To overcome the problems discussed in the previous section, Efron *et al.* (2001) introduced the *local* false discovery rate. Similar to the definition of the positive false discovery rate in Equations (3.9) and (3.10), the local false discovery rate is defined as

$$\mathrm{fdr}(p) = Pr\left[H = 0 \,\middle|\, P = p\right] \tag{3.13}$$

$$= Pr\left[H = 0\right] \frac{Pr\left[P = p \,\middle|\, H = 0\right]}{Pr\left[P = p\right]}. \tag{3.14}$$

In contrast to the positive false discovery rate, the local false discovery rate is conditioned on $P = p$ instead of $P \leq p$. The definition above has to be interpreted with care since a point probability $Pr\,[P = p]$ equals zero. Here the term "$P = p$" refers to "in the vicinity of p", meaning that we are interested in a local probability measure closely around the p-value level p. The local false discovery rate is the probability that a gene is not differentially expressed given its p-value $p_i = p$ and conditional on the set of all observed p-values.

We return to the simulation underlying Figure 3.3, where we observed scores drawn from a mixture of two normal distributions with different location parameters for the induced part. In Figure 3.4, the local false discovery rates of the three mixture models are plotted over the range of p-values. For enhanced interpretation, we propose to plot the posterior probability of differential expression as in the right panel of Figure 3.4, that is $Pr\left[H = 1 \,\middle|\, P = p\right] = 1 - \mathrm{fdr}(p)$ for all $p \in [0, 1]$. We use the term "local false discovery rate" for both posterior probabilities in parallel although $1 - \mathrm{fdr}(p)$ rather relates to a local *true* discovery rate. In the simulation with less induction given by $\mu = 2$, the overlap of the two score distributions caused the local false discovery rate to spread over the whole range of p-values. Even a p-value close to one has a slight chance to belong to an induced gene. Here the experiment exhibits a broad *twilight zone* of p-values supporting both differential and non-differential expression. We further observe a substantial difference to the q-value curves in Figure 3.3: now it is the local false discovery rate curve with highest induction that has the largest slope, see left panel of Figure 3.4. In contrast to q-values, the local false discovery rate estimates react immediately to the inclusion of highly induced genes.

The effect becomes more prominent in Figure 3.5. Displayed are the estimated positive and local false discovery rates with the corresponding numbers of genes with smaller or equal values. The left panel equals in principle Figure 3.3 with reverted axes but with the number of all positives shown instead of false positives. Again, the same pFDR-based p-value filter will lead to a higher number of differentially expressed genes in case of high induction ($\mu = 4$) than in case of moderate induction ($\mu < 4$). In contrast to this, the local false discovery rate does not suffer from high induction. In our artificial example, we observe that for $\mathrm{fdr} = 0.5$ the three curves coincide in 2000 differentially expressed genes, perfectly resembling the simulation setting of 2000 induced and 8000 non-induced genes. In this ideal

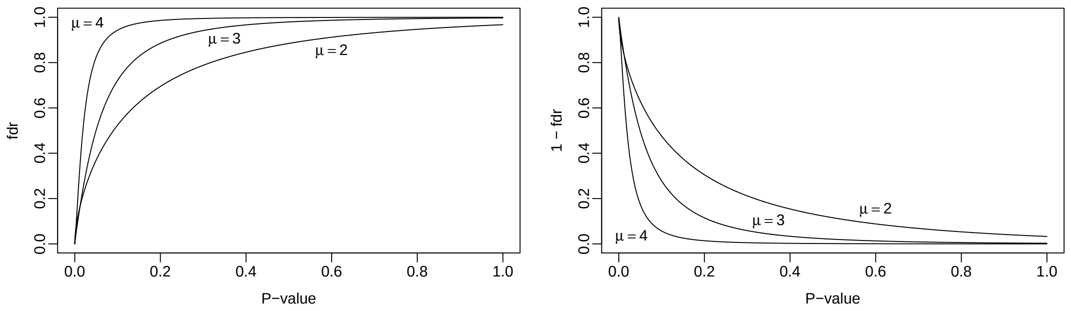

Figure 3.4: Local false discovery rates over the range of p-values. Shown are the rates corresponding to the simulation in Figure 3.3. Left panel: Posterior probability of non-differential expression. Right panel: Posterior probability of differential expression. In an experiment with moderate induction $\mu = 2$, the curve declines slowly and spreads over a twilight zone of both induction and non-induction.

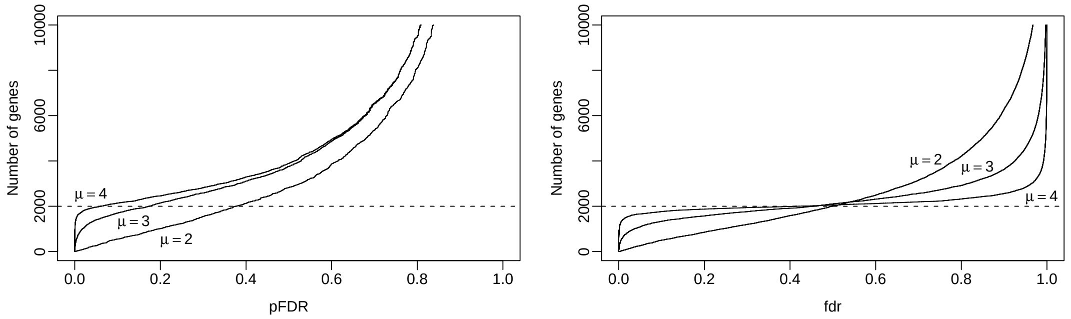

Figure 3.5: Positive and local false discovery rates with the corresponding number of genes with smaller or equal rates, same data as shown in Figures 3.3 and 3.4. With increasing induction μ, the local false discovery rate separates clearer between the 2000 induced and the 8000 non-induced genes (right panel), which is not the case for the positive false discovery rate (left panel).

setting, a fifty-fifty chance of being differentially expressed is naturally assigned to the genes on the border between induction and non-induction.

Modeling the p-value distribution. The estimation of the local false discovery rate is based on the distribution of the p-values. The p-value distribution is commonly modeled using a mixture with at least two components

$$f(p) = \pi_0 f_0(p) + \pi_1 f_1(p), \tag{3.15}$$

where f is the probability density function of the observed p-values. The overall density is decomposed into the density component f_0 referring to the p-value density of non-induced genes, and into f_1, the p-value density of induced genes. The factor π_0 denotes the global proportion of non-induced genes in the experiment. It corresponds to the prior probability of being non-induced in Equations (3.11) and (3.14), and to the height of the horizontal line in Figure 3.2:

$$\pi_0 = Pr\left[H = 0\right]. \tag{3.16}$$

The second mixture parameter is simply $\pi_1 = 1 - \pi_0$, which we interpret as the prior probability of being differentially expressed: $\pi_1 = Pr\left[H = 1\right]$. In terms of the mixture model (3.15), the local false discovery rate definition in Equation (3.14) can be translated into

$$\mathrm{fdr}(p) = \pi_0 \frac{f_0(p)}{f(p)}. \tag{3.17}$$

The estimation of the local false discovery rate amounts to estimating the single components of the right-hand side of Equation (3.17). The overall density f is estimated from the set of observed p-values for example by applying kernel density estimation or similar smoothing techniques. The estimation of the null density f_0 needs the selection of a data-generating model. We might either choose a fully parameterized model for the expected null distribution or we estimate f_0 from the given data by using permutation techniques. In the empirical Bayes approach of Efron (2004) the two variants are combined by estimating the parameters of a normal mixture from scores observed under class-label permutation. The same options apply to the estimation of the prior probability π_0. The prior can be determined by an expert or estimated from the observed data. We will review estimators of π_0 in more detail in Section 3.5 and turn now to existing estimation procedures for the local false discovery rate, which differ with respect to the chosen models.

First, we can further simplify the mixture model in Equation (3.15): by probability theory, p-values are uniformly distributed if they were derived under the null hypothesis. With $f_0(p) = 1$ for all $p \in [0, 1]$, the local false discovery rate estimation amounts to the estimation of the two unknown terms π_0 and f_1 in

$$\mathrm{fdr}(p) = \frac{\pi_0}{f(p)} \quad \text{with} \quad f(p) = \pi_0 + (1 - \pi_0)\, f_1(p). \tag{3.18}$$

The mixture model depends on the choice of the mixture parameter π_0 and on distributional parameters of the density f_1. Several configurations of the parameter set might be chosen, which all explain the mixture density equally well. Hence, the mixture model is not uniquely defined and the parameters are not identifiable unless we apply additional assumptions on π_0 or on f_1. The identification problem gives rise to two different approaches. First, a fully parameterized representation of the alternative part f_1 is chosen ensuring that the prior π_0 is identifiable. Second, density f_1 is modeled using non-parametric techniques. Here additional assumptions on both f_1 and π_0 must be set to ensure identification. In the following, we briefly review estimation approaches that use the uniformity assumption above. Other approaches that do not exploit the p-value but the original score distribution can be adopted to p-values and we review them here as well. Most of the methods will be explained in technical detail in Section 3.5.

Fully parameterized mixture models. Pounds and Morris (2003) selected a beta-distribution for f_1. The first parameter of the beta-distribution is set to one. The unknown second parameter and prior π_0 are determined using maximum likelihood estimation. This simple beta-uniform model was generalized by Allison *et al.* (2002) who modeled the alternative density f_1 as a finite mixture of beta-distributions, now allowing both parameters to vary. Model selection with respect to the number of beta components was done using a bootstrap approach. Liao *et al.* (2004) described a local version of the beta-uniform mixture model. The authors split the p-value range into bins and fit separate models similar to that of Pounds and Morris (2003) for each bin. For model fitting, a full Bayesian model with conjugate prior distributions was used to derive the joint posterior distribution of all model parameters including π_0.

Non-parametric mixture models. The parameterized models are restricted to the choice of distribution. The justification of such a choice might be questionable and a non-parametric approach is preferable in that case. However, non-parametric models for f_1 need additional assumptions. Efron *et al.* (2001) assumed that there exists an upper bound on prior π_0. They defined the upper bound in terms of the observed scores t as

$$\pi_0 \leq \min_t \left\{ \frac{f(t)}{f_0(t)} \right\}. \tag{3.19}$$

The interpretation of the upper bound becomes clearer if we base the approach on the corresponding p-values such that Equation (3.19) simplifies to

$$\pi_0 \le \min_p \{f(p)\}, \tag{3.20}$$

since $f_0(p) = 1$ for all $p \in [0,1]$. The upper bound for π_0 is given as the minimum of the p-value density. Since we assume the p-values to be uniformly distributed under the null hypothesis, the upper bound estimate $\pi_0 = \min_p \{f(p)\}$ is equivalent to the assumption that the null density f_0 consists of the largest possible uniform fraction given f. In other words, the alternative density f_1 does not contain any uniform parts. If $\pi_0 < \min_p \{f(p)\}$, we allow f_1 to include an additional uniform part.

Efron *et al.* (2001) suggested estimating f by smoothed logistic regression and then using the upper bound $\min_p\{\hat{f}(p)\}$ as an estimator for π_0. Pounds and Cheng (2004) applied assumption (3.20) in the context of a mixture model with a uniform component for the non-induced genes using a histogram estimator for deriving the mixture density f. Since the prior π_0 is estimated using the data, the procedures of Efron *et al.* (2001) and Pounds and Cheng (2004) are empirical Bayes methods. Assumption (3.20) is equivalent to assuming that f_1 has no uniform component. If it has, the method of Efron *et al.* (2001) overestimates π_0. Do *et al.* (2005) criticized the biased estimation of π_0 and developed a full non-parametric Bayesian mixture model using Dirichlet processes. Instead of a data driven plug-in estimate of π_0, they imposed a uniform prior distribution on it.

In the context of the global false discovery rate, Genovese and Wassermann (2004) assumed in addition to the upper bound derived from Equation (3.20) that f is monotonously decreasing, implying

$$\pi_0 = \min_p \{f(p)\} = f(1). \tag{3.21}$$

Equation (3.21) implies that π_0 can be determined by estimating $f(1)$. The same strategy was suggested by Storey and Tibshirani (2003), who described a smoothed extrapolation based estimator for $f(1)$. The original paper of Tusher *et al.* (2001) contains a simplified version of the extrapolation based estimator, which is implemented in their SAM software.

In Chapter 5, we introduce a novel estimator of the local false discovery rate, termed the successive exclusion procedure (SEP). In an extensive simulation study we compared our SEP estimates to various other estimation procedures including some of the previously reviewed methods. An overview of the competitors is given in the following section.

3.5 Estimating the proportion of non-induced genes

The uniformity assumption for the null density f_0 reduces the local false discovery rate estimation problem to an estimation of two components: prior π_0 and mixture density f. The latter can be done efficiently by applying density estimators to the observed p-value distribution. The crucial point is the estimation of prior π_0. Simply setting $\widehat{\pi_0} = 1$ is the most conservative choice and will lead in many cases to an overestimated local false discovery rate. The goal is to estimate π_0 from the data without underestimating it severely, as an underestimated π_0 results in overly optimistic conclusions on the percentage of induced genes $\pi_1 = 1 - \pi_0$.

In the following, we introduce a selection of π_0 estimators, which we evaluated in a comprehensive simulation study. The simulation covers different settings regarding the true percentage π_0, strength of induction and number of genes. Results are shown in Section 5.6. The estimation of the mixture parameter π_0 is the first step in the estimation of the local false discovery rate. Such a parameter is needed not only in the context of microarray experiments. Similar problems with many hypotheses and probably sparse signals are for example the analysis of allele frequencies (Mosig *et al.*, 2001) but also the detection of novel stars as discussed in Meinshausen and Rice (2006).

In Section 3.4, we reviewed main representatives of estimation concepts for the local false discovery rate. The methods split into two groups depending on whether a parametric or non-parametric model was used. Now we divide the methods differently. Most methods need transformations of the data such as certain mixture models, which have to be fitted to the data using direct parameter estimation, smoothing techniques or an iterative design. Other estimators are defined in closed form. We call the first set of estimators *iterative* and the second ones *analytic*. In the following, we review 13 estimators for π_0. If no other name was given, the methods are called by their first author's name. Some methods have

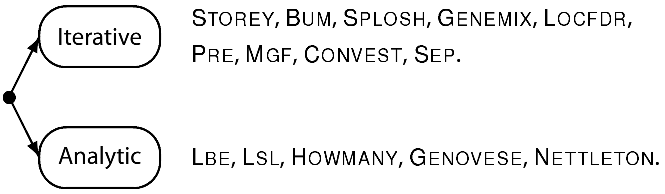
Iterative
Storey, Bum, Splosh, Genemix, Locfdr, Pre, Mgf, Convest, Sep.
Analytic
Lbe, Lsl, Howmany, Genovese, Nettleton.

Figure 3.6: Overview on estimation procedures under evaluation. Method Sep denotes our novel approach introduced in Chapter 5.

been published recently, others have performed well in earlier comparisons (Ferkingstad *et al.*, 2003; Broberg, 2005). In Figure 3.6 we display how the methods split into analytic and iterative estimators.

Storey The procedure introduced in the work on positive false discovery rates by Storey (2003) may be the most prominent and commonly used π_0 estimator. The underlying algorithm is given explicitly in Storey and Tibshirani (2003). Assuming uniformity for p-values of non-induced genes, the ratio of the observed and the expected number of p-values is computed for a range of threshold values $p \in \{0, 0.01, \dots, 0.95\}$ as

$$\widehat{\pi_0}(p) = \frac{\sum_{i=1}^{m} I\{p_i > p\}}{m(1-p)}, \tag{3.22}$$

where again m is the number of genes. This estimator was first introduced in Schweder and Spjøtvoll (1982). Equation (3.22) is a natural estimator for the p-value density f. At $p = 1$, that is at the rightmost end of the p-value histogram, we assume to observe only p-values stemming from non-induced genes. Thus with $\hat{f}$ being a smoothed version of $\widehat{\pi_0}(p)$, the final estimator is derived by extrapolation such that

$$\widehat{\pi_0} = \hat{f}(1), \tag{3.23}$$

which relates to the upper bound definition in Equation (3.21). The extrapolation uses a natural cubic spline with three degrees of freedom. A cubic spline is a smoothing technique often used to fit non-linear data. It consists of piecewise fitted cubic polynomials with continuity at the border between two pieces. The natural cubic spline requires a linear instead of cubic fit at the boundaries of the data range. The π_0 estimator is implemented in Storey's package *qvalue*.

Bum Pounds and Morris (2003) chose a beta-uniform mixture model (BUM) to express the observed p-value distribution. The mixture density f is modeled by the sum of a uniform component and a beta-distributed component:

$$f(p|a,\lambda) = \lambda + (1-\lambda)ax^{a-1}, \tag{3.24}$$

with λ being the mixture parameter of the uniform part and a being the first parameter of the beta distribution. Note that the second parameter is set to one ($b = 1$). The model parameters λ and a are fitted by maximum likelihood estimation. To this end, a logit transformation is applied to the two model parameters such that the density is expressed in terms of two new parameters $\psi = \text{logit}(a) = \ln\left(\frac{a}{1-a}\right)$ and $\varphi = \text{logit}(\lambda) = \ln\left(\frac{\lambda}{1-\lambda}\right)$. The log-likelihood of f is given as

$$l(\psi,\varphi|p) = \sum_{i=1}^{m} \log(f(p_i|\psi,\varphi)). \tag{3.25}$$

Values of ψ and φ have to be found that maximize the log-likelihood. As a closed form solution does not exist, the authors applied numerical optimization to find the optimal values. The maximum likelihood estimates of the initial parameters then follow as $\hat{a} = \frac{\exp(\hat{\psi})}{1+\exp(\hat{\psi})}$ and $\hat{\lambda} = \frac{\exp(\hat{\varphi})}{1+\exp(\hat{\varphi})}$. An upper bound estimator of π_0 is then given as

$$\widehat{\pi_0} = \hat{\lambda} + (1-\hat{\lambda})\hat{a}. \tag{3.26}$$

The original code is available at http://www.stjuderesearch.org/statistics. The function was written for the software S-plus® by Insightful Corporation, Seattle, WA, USA. Both R and S-Plus® are implementations based on the programming language S, and are thus closely related. As some functions differ between the two environments, the optimization step in the Bum code had to be translated into R. We exchanged the S-Plus® optimizer *nlminb* with the R function *optim*.

Splosh Pounds and Cheng (2004) estimated the mixture density by applying a spacing LOESS histogram (SPLOSH) estimator. The "spacings" are intervals of the p-value range. To estimate the mixture density f the p-value range is divided into k intervals and a local polynomial regression model (LOESS) is applied to the interval frequencies. In LOESS regression a weighted polynomial model is fitted to each data point. The fit is based on a subset of the data in the neighborhood of the target point. The weights are assigned such that they decrease with increasing distance from the target point. Weights are defined by a kernel function, here the Epachnechnikov kernel that yields weights w proportional to

$$w(x) = \begin{cases} (1 - |x|^3)^3 & \text{if } |x| \leq 1 \text{ and} \\ 0 & \text{otherwise,} \end{cases} \tag{3.27}$$

where x is the distance of any neighboring point to the target point divided by a smoothing parameter that determines the width of the neighborhood. Prior to LOESS regression, the SPLOSH approach ensures via certain data transformations such as logarithmic transformation that the resulting density function is strictly positive and that boundary effects near $p = 0$ or $p = 1$ do not occur.

Once the density f has been estimated, the prior probability is determined at the rightmost end of the density yielding the upper bound estimate $\widehat{\pi_0} = \hat{f}(1)$. As in Pounds and Morris (2003), the π_0 estimate is only a by-product of the false discovery rate estimation. The authors showed that SPLOSH yields better estimates than the earlier method BUM of the same first author. The approach is available as an S-Plus® function at http://www.stjuderesearch.org/statistics. To work under R, we exchanged one *predict.loess* statement with *predict*.

GENEMIX The approach of Liao *et al.* (2004) is an extension of the beta-uniform mixture model of Pounds and Morris (2003), who used a single beta distribution as alternative density f_1. Liao *et al.* did not apply a global beta distribution but fitted local beta distributions to small intervals of the p-value range. To this end, the authors applied a proportional hazard model. The p-value range is divided into k intervals with cut points $0 = t_0 < t_1 < \cdots < t_k = 1$. Let $\lambda_i > 1$ be the hazard ratio of the alternative density f_1 over the null density f_0 on the interval $[t_{i-1}, t_i)$, that is

$$\lambda_i = \frac{f_1(p)}{1 - F_1(p)} \frac{1 - F_0(p)}{f_0(p)} = \frac{f_1(p)}{1 - F_1(p)}(1 - p), \tag{3.28}$$

with $p \in [t_{i-1}, t_i)$. We further define

$$\theta(p) = \sum_{i=1}^{k} \lambda_i I\{[t_{i-1}, t_i)\}. \tag{3.29}$$

With these ingredients the piecewise proportional hazard model is given as

$$\frac{f_1(p)}{1 - F_1(p)} = \frac{\theta(p)}{1 - p}. \tag{3.30}$$

It follows that the alternative density f_1 can be written as a piecewise function. Let l be the index for which $p \in [t_{l-1}, t_l)$ and λ the set of hazard ratios, $\lambda = (\lambda_1, \ldots, \lambda_k)$. The authors show that the alternative density is then given as

$$f_1(p|\lambda) = \lambda_l(1 - p)^{\lambda_l - 1} \prod_{i=1}^{l-1} (1 - t_i)^{\lambda_i - \lambda_{i+1}}. \tag{3.31}$$

The model parameters are estimated by Bayesian inference as follows. The hazard ratios are transformed to $\tau_i = \log(\lambda_i - 1)$ and a normal distribution is assumed for

$$\tau_{i+1} - \tau_i \sim \mathcal{N}(0, \sigma^2). \tag{3.32}$$

Variance σ^2 determines the strength of smoothing. The variance is further controlled by a hyper prior ν, which serves as the final tuning parameter for smoothness. In addition, a beta(1,1) prior distribution is imposed on the mixture parameter π_0. For a given number k of cut points, the iteration starts with smoothing parameter $\nu = 1$. Parameter ν increases while evaluating the posterior distributions of π_0 and $\lambda_1, \ldots, \lambda_k$. The algorithm stops after a specified number of iteration or earlier if density f_1 reached a certain level of smoothness. Finally, estimate $\widehat{\pi_0}$ is taken as the mean of the posterior distribution of π_0.

The authors' implementation in function *gene.mixture* is available at http://www.geocities.com/jg_liao/software/. We set the number of iterations to 500 to reduce computation time and used $k = 60$ cut points as given in an example. The smoothing parameter is set to $\nu = 1$. Instead of being incremented as described in the original paper, the smoothing parameter appears to be kept at a fixed value in the function code.

Locfdr Another upper bound estimator is based on the model of Efron (2004). The procedure works on observed scores, which are assumed to follow a standard normal distribution under the null hypothesis. We used the implementation in package *locfdr* by B. Efron and B. Narasimhan. The mixture density is estimated by smoothing techniques, here a natural spline with seven degrees of freedom. Similar to method Splosh, the p-value range is divided into k intervals and the smoothing spline is applied on the interval frequencies. The function was used with default values, that is $k = 120$ intervals. In addition, 1‰ of the tail proportions were omitted when fitting the mixture density f. In the software, the null density can either be estimated empirically or set to standard normal. As the simulated null distribution introduced in Section 5.6 was standard normal, we used the latter setting with default values otherwise. The model differs from the seminal publication of the empirical Bayes approach in Efron *et al.* (2001), where logistic regression was applied to estimate the local false discovery rate. Since no original implementation of this procedure was available, we used the most recent method of the first author.

Pre Broberg (2005) adapted the Locfdr approach to work on p-values instead of scores. As above, the p-value range is divided into intervals and the interval counts are fitted by Poisson regression (PRE), the classical model for count data. Here, a polynomial function was fitted to the data. Let t_k be the midpoint of the kth interval and μ_k^0 the observed frequency therein. The expected frequency is then modeled as

$$\mu_k(\beta) = \mu_k^0 \exp(\beta_0 + \beta_1 t_k + \beta_2 t_k^2 + \beta_3 t_k^3 + \beta_4 t_k^4), \qquad (3.33)$$

with $\beta = (\beta_0, \beta_1, \beta_2, \beta_3, \beta_4)$ being the set of model parameters. The Poisson model is fitted via maximum likelihood estimation. The final estimate of the mixture density $\hat{f}$ is derived from a smoothing spline with four degrees of freedom on the expected relative frequencies. The percentage of non-induced genes is taken as the minimum density: $\widehat{\pi_0} = \min_p \hat{f}(p)$. According to Equation (3.20), the minimum density estimator is an upper bound estimator for π_0. The estimation procedure is implemented in the author's package *SAGx* in function ***p0.mom***. The location and number of intervals k is determined internally depending on the number of genes m.

Mgf Besides Pre, Broberg (2005) proposed a second estimator, which is based on the moment generating function (MGF) of a mixture density. The moment generating function M of any random variable P with density $f(p)$ can be written as the expectation

$$M(s) = \int \exp(sp)\, f(p)\mathrm{d}p, \tag{3.34}$$

integrated over the support of f. The moment generating function of a mixture distribution is again a mixture of moment generating functions. For the mixture in Equation (3.15) thus follows

$$M(s) = \pi_0\, M_0(s) + (1 - \pi_0) \int \exp(sp)\, f_1(p)\mathrm{d}p, \tag{3.35}$$

with

$$M_0(s) = \frac{\exp(s) - 1}{s} \tag{3.36}$$

being the moment generating function of the uniform distribution. Let

$$M_1(s) = \int \exp(sp)\, f_1(p)\mathrm{d}p \tag{3.37}$$

denote the unknown transform of the alternative density, The mixture parameter π_0 is then given as

$$\pi_0 = \frac{M(s) - M_1(s)}{M_0(s) - M_1(s)}. \tag{3.38}$$

The author derived a recursive formula to derive M_1 from M, M_0 and M_1 for increasing values of s and with

$$\widehat{M}(s) = \frac{1}{m} \sum_{i=1}^{m} \exp(sp_i) \tag{3.39}$$

being the estimated moment generating function of the mixture distribution. The ratio in Equation (3.38) is computed for increasing values $s \in [0, 1]$. The iteration stops if a certain choice of M_1 and s provides a stable estimate of π_0. Like PRE, the MGF estimator is available from function *p0.mom* in package *SAGx*.

CONVEST Langaas *et al.* (2005) assumed that the p-value density f is a convex and decreasing function. The authors decomposed the mixture density f into a mixture of triangular densities f_θ:

$$f_\theta(p) = \frac{2(\theta - p)^+}{\theta^2}, \tag{3.40}$$

with parameter $\theta \in (0,1]$. The density parameters are estimated by iterative maximum likelihood approximation starting with a single uniform density $\hat{f}(p) = 1$ for all $p \in [0,1]$. In each iteration $j = 0,1,2\ldots$ the current estimate $\hat{f}_j$ replaces $\hat{f}$ and parameter θ is determined by evaluating

$$\hat{\theta} = \arg\min_{\theta\in(0,1]} \left\{ \sum_{i:f(p_i)>0} \frac{\hat{f}(p_i) - f_\theta(p_i)}{\hat{f}(p_i)} \right\}. \tag{3.41}$$

Within each step, the density is a mixture of the current iterate and the new optimal density:

$$\hat{f}_{j+1} = (1-\hat{\varepsilon})\hat{f}_j + \hat{\varepsilon} f_{\hat{\theta}}, \tag{3.42}$$

where

$$\hat{\varepsilon} = \arg\min_{\varepsilon\in[0,1)} \left\{ - \sum_{i:f(p_i)>0} \log((1-\varepsilon)\hat{f}_j(p_i) + \varepsilon f_{\hat{\theta}}(p_i)) \right\}. \tag{3.43}$$

This procedure ensures to find the new mixture density with the highest improvement of accuracy, which relates to a "steepest-descent" down-hill search algorithm. The iteration stops when a specified accuracy is meet. In practice, the optimal value for parameter θ is found over a grid on the interval $(0,1)$.

The final mixture estimate $\hat{f}$ leads to the upper bound estimate $\widehat{\pi_0} = \hat{f}(1)$. The procedure was introduced in a work by the same authors providing a comparison of earlier π_0 estimators, in which the Convest method outperforms its competitors (Ferkingstad *et al.*, 2003). The convex density estimator is implemented in the *limma* package by G. Smyth, which is designed for linear model fitting of microarray data, and was applied with default number of iterations set to 100.

Lbe Dalmasso *et al.* (2005) introduced a π_0 estimator that is not derived by an iterative procedure but given in closed form. The method is termed *location based estimator* (LBE), where "location" refers to expectation. The authors did not follow the density estimation approach but took expected values such that the mixture in Equation (3.15) transforms to

$$\frac{\mathrm{E}(P)}{\mathrm{E}_0(P)} = \pi_0 + (1-\pi_0)\frac{\mathrm{E}_1(P)}{\mathrm{E}_0(P)}, \tag{3.44}$$

where E_0 and E_1 are the expectations under the null and under the alternative hypothesis and P is the p-value random variable. An upper bound of π_0 is simply the left-hand side of the equation above, that is

$$\frac{E(P)}{E_0(P)} \geq \pi_0. \tag{3.45}$$

Dalmasso *et al.* showed that any transformation $\varphi(P)$ of P leads to tighter bounds since

$$\frac{E_1(\varphi(P))}{E_0(\varphi(P))} \leq \frac{E_1(P)}{E_0(P)}. \tag{3.46}$$

With $\varphi(p) = -\log(1-p)$, the upper bound might be further reduced with respect to the increasing power k in

$$\frac{E_1(\varphi(P)^{k+1})}{E_0(\varphi(P)^{k+1})} \leq \frac{E_1(\varphi(P)^{k})}{E_0(\varphi(P)^{k})}. \tag{3.47}$$

The final estimator is then derived as

$$\widehat{\pi_{0(k)}} = \frac{E(\varphi(P)^k)}{E_0(\varphi(P)^k)} = \frac{\frac{1}{m}\sum_{i=1}^{m} \varphi(p_i)^k}{E_0(\varphi(P)^k)}. \tag{3.48}$$

The authors showed that $E_0(\varphi(P)^k) = k!$ for $\varphi(p) = -\log(1-p)$, such that the estimator is finally given in closed form as

$$\widehat{\pi_{0(k)}} = \frac{\frac{1}{m}\sum_{i=1}^{m} \left(-\log(1-p_i)\right)^k}{k!}. \tag{3.49}$$

Parameter k is an integer value that increases with the number of genes m. The authors suggest to set $k = 1$ for 1000 genes and $k = 3$ for 10000 genes. We applied the original implementation of LBE available at http://ifr69.vjf.inserm.fr/lbe, which also provides false discovery rate estimates.

Lsl In Benjamini and Hochberg (2000), the authors refined their original procedure to estimate the false discovery rate as given in Equation (3.5). In particular, they exchanged the number of genes m with the estimated number of non-induced genes $\widehat{m_0} = \widehat{\pi_0}m$, which equals the definition of q-values in Equation (3.12). The number m_0 is estimated from the slope of the line when drawing p-values over their respective ranks. That is, the ordered p-values $p_{(1)}, \dots, p_{(m)}$ are plotted versus their expectations $1, \dots, m$ under the null hypothesis of no induction. If not a single gene is induced, p-values are assumed to be uniformly distributed and the curve of all points $(i, p_{(i)})$ is a straight line passing through the origin and the point $(m+1, 1)$ with slope $\beta = 1/(m+1)$. The more induced genes there are in the experiment, the more small p-values we observe and the curve of $(i, p_{(i)})$ will depart from the straight line when approaching zero. The slope of the line fitted through the larger p-values is a natural estimator for m_0 since $\beta = 1/(m_0 + 1)$. With an appropriate slope estimate $\hat{\beta}$, the authors estimate $\widehat{\pi_0}m = \widehat{m_0} = \hat{\beta}^{-1}$.

The estimation of the slope β depends on how many large p-values are taken into account for the linear fit. The authors propose a *Lowest Slope* estimator (LSL) that works as follows. For all $i = 1 \dots, m$, we compute the slopes

$$\beta_i = \frac{1 - p_{(i)}}{m + 1 - i}. \tag{3.50}$$

We loop once through the set of p-values starting with the smallest p-value $p_{(1)}$ and search for the smallest $i^\star$ with decreasing slope, that is with $\beta_{i^\star} < \beta_{i^\star - 1}$. The final estimator is then given as

$$\widehat{m_0} = \min\left\{\frac{1}{\beta_{i^\star}} + 1, m\right\}. \tag{3.51}$$

Adding 1 to $\beta_{i^\star}^{-1}$ before taking the minimum ensures that the estimator is conservative and tends to overestimate m_0.

The Lsl estimator is implemented in package *GeneTS* by K. Fokianos, J. Schäfer, and K. Strimmer. The package is designed for time-series analysis of gene expression data. The Lsl method is available via function *fdr.estimate.eta0* with argument *method="adaptive"*.

Howmany Meinshausen and Rice (2006) discussed the problem of identifying unknown objects in outer space by monitoring light fluxes of known stars. The number of p-values exceeds that of a typical microarray experiment by far. The authors proposed an estimate of the lower bound of π_1, which in our case serves as an upper bound to π_0. The method is motivated by the theory of bounding functions and bounding sequences.

Let U be the uniform distribution on [0,1] and $U_m(p)$ the empirical cdf of m observations of a random variable P with probability distribution U. Then

$$V_{m,\delta} = \sup_{p \in (0,1)} \frac{U_m(p) - p}{\delta(p)} \tag{3.52}$$

defines the supremum of an empirical distribution weighted by a *bounding function* $\delta(p)$. We require the bounding function to be real-valued on [0,1] and strictly positive on (0,1). The series $\beta_{m,\alpha}$ is called a *bounding sequence* for $\delta(p)$ if $m\beta_{m,\alpha}$ is monotonically increasing with m and

$$Pr[V_{m,\delta} > \beta_{m,\alpha}] < \alpha \tag{3.53}$$

for all m and constant level α. Inserting (3.52) into (3.53) leads to

$$Pr[\sup_{p \in (0,1)} U_m(p) > p + \beta_{m,\alpha}\,\delta(p)] < \alpha. \tag{3.54}$$

Let $F_m(p)$ be the empirical cdf of m observed p-values. Assuming that a certain proportion of these p-values is uniformly distributed and given the bounding statement above, a natural lower bound for the percentage of induced genes π_1 is $\widehat{\pi_1} = \sup_{p \in (0,1)} \{F_m(p) - p - \beta_{m,\alpha}\,\delta(p)\}$. The authors showed that an additional factor $1/(1-p)$ can be gained such that the final lower bound estimate is given as

$$\widehat{\pi_1} = \sup_{p \in (0,1)} \frac{F_m(p) - p - \beta_{m,\alpha}\,\delta(p)}{1-p}. \tag{3.55}$$

For a given bounding function $\delta(p)$ and accompanying bounding sequence $\beta_{m,\alpha}$ the theory of bounding functions then guarantees that $\widehat{\pi_1}$ is a lower bound for π_1 at confidence level α, that is

$$Pr[\widehat{\pi_1} \le \pi_1] \ge 1 - \alpha. \tag{3.56}$$

Meinshausen and Rice (2006) chose the bounding function $\delta(p) = \sqrt{p(1-p)}$ and showed that this choice is optimal. The associated bounding sequence follows as

$$\beta_{m,\alpha} = \frac{-\log(-\log(1-\alpha)) + 2\log(\log(m)) + 0.5\log(\log(\log(m))) - 0.5\log(4\,\pi)}{\sqrt{2\,m\log(\log(m))}}. \tag{3.57}$$

The lower bound estimator above is implemented in package *howmany* by N. Meinshausen. We kept the default value of the confidence level at $\alpha = 0.05$. The package also includes a second estimator, introduced in Meinshausen and Bühlmann (2005). Here the approach takes the correlation structure between genes into account and the usual permutation regime is applied to evaluate the data under randomness. The random positives count as false positives and are natural estimates of the measure $V(p)$ in Equation (3.1). Thus they are the basis of the lower bound estimator. We did not include this second estimator into our study for two reasons. First, we stick to the naive but fundamental setting of independent genes. Second, the approach requires the whole expression matrix as input whereas we condensed the simulated data to the level of observed scores and p-values.

Genovese Another lower bound estimator for π_1 was introduced by Genovese and Wassermann (2004). Meinshausen and Rice (2006) translated the estimator into bounding theory. Due to a constant boundary function $\delta(p) = 1$, the bounding sequence of Genovese and Wassermann (2004) is given as

$$\beta_{m,\alpha} = \sqrt{(2m)^{-1}\log(2/\alpha)}. \tag{3.58}$$

The constant bounding function is suboptimal compared to the bounding function of method Howmany, that is it only results in consistent estimates when the true amount π_1 is large. However, we kept Genovese and Wassermann's estimator for its appealing simplicity. As no original code of the method was available, we implemented it in R and set $\alpha = 0.05$ in accordance to method Howmany.

Nettleton In a genomic study, Mosig *et al.* (2001) presented a π_0 estimator that was based on an iterative comparison of frequencies observed in intervals of the p-value range. Nettleton and Hwang (2003) derived an exact formulation of the algorithm to reproduce the results of the original paper. In addition, they proved that the algorithm always converges to a certain value. An analytic solution can be given such that the iteration of Mosig *et al.* (2001) is not needed. To estimate π_0, the range of p-values [0,1] is divided into k equidistant intervals. Let m_i denote the observed p-value frequency in the ith interval. The average frequency of the intervals on the right-hand side of interval i including interval i is given as

$$\overline{m}_{i:k} = \sum_{j=i}^{k} \frac{m_j}{k-i+1}. \tag{3.59}$$

Now the procedure works as follows. Going from left to right through the intervals, each frequency count is compared to the average of the interval count and the counts of the intervals following at the right-hand side of this interval. Starting with the left-most sum of frequencies

$$M_0 = \sum_{j=1}^{k} m_j = k\,\overline{m}_{1:k}\,, \tag{3.60}$$

the authors derived a recursive formula to compute the successive sums of frequencies $M_i = \sum_{j=i+1}^{k} m_j$ for $i \geq 1$. Now one determines the first interval $i^\star$ for which

$$i^\star = \min_{1\leq i\leq k}\left\{ m_i \leq \frac{M_{i-1}}{k} = \overline{m}_{i:k} \right\}. \tag{3.61}$$

The estimated number of non-induced genes is then obtained from the count sum of the first interval $i^\star$ with a count not exceeding the average, that is

$$M_{i^\star} = k\,\overline{m}_{i^\star:k} = \widehat{\pi_0}\, m, \tag{3.62}$$

leading to the final estimate $\widehat{\pi_0} = M_{i^\star}/m$. This method is not iterative but loops only once through the intervals. The search stops when the left-most interval is found satisfying the inequality in Equation (3.61). The original implementation is available at http://www.public.iastate.edu/$\sim$dnett/ and was applied with the default value of $k = 50$ intervals.

In Chapter 5, we introduce a novel algorithm to estimate the local false discovery rate and evaluate its performance in a comprehensive simulation study including the π_0 estimators above. Prior to that, we introduce six expression data sets in the following chapter, on which some of the previously introduced methods were applied for illustration.