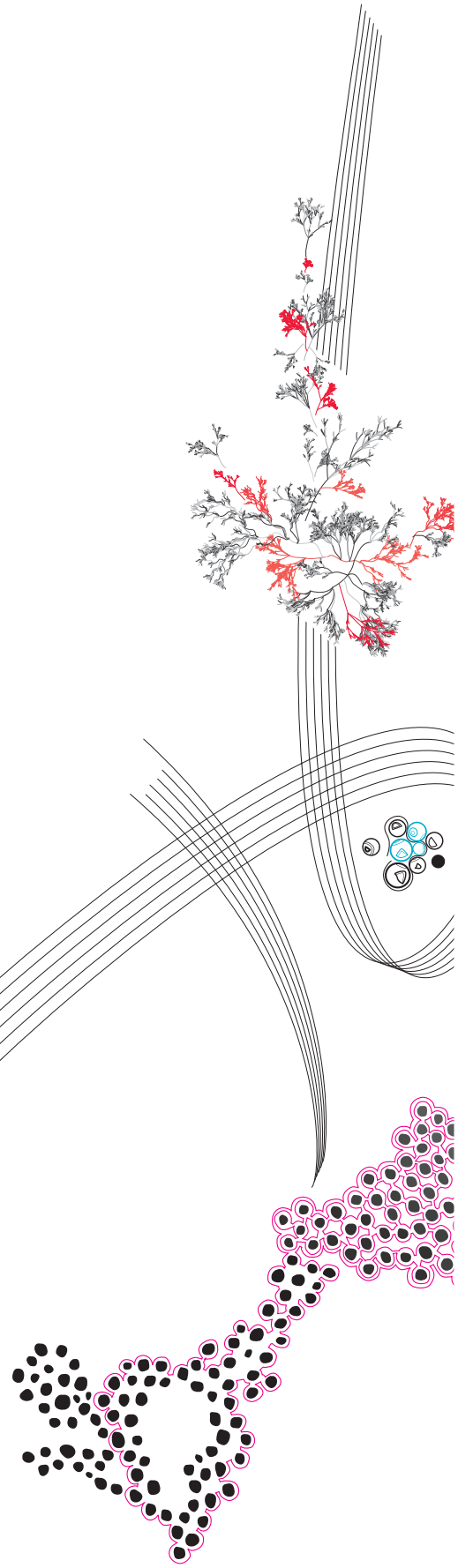

BSc Thesis Applied Mathematics

Frequentist vs. Empirical Bayes: A comparison of procedures for high-dimensional two-sample statistical inference, in particular for microRNA data

Wei-Ting Sun

Supervisors: K. Proksch and A. Marjanovic

July, 2022

Department of Applied Mathematics
Faculty of Electrical Engineering,
Mathematics and Computer Science

UNIVERSITY OF TWENTE.

Preface

This bachelor assignment, in its current form, would not exist without my supervisors Katharina Proksch and Aljosa Marjanovic. I would like to thank them for their helpful comments on my drafts and reassurance that I am not writing gibberish in the flavor of LaTeX. Extra thanks for Katharina Proksch for teaching my elective course Simultaneous Statistical Inference, as without it, I would have had to do a lot more research on the topic.

I would like to thank my fellow Applied Mathematics friends for accompanying me on our three-year journey together, and especially Anete Valnere for reading and giving me feedback on my (unfinished) draft.

As is the norm, I would like to thank my parents for their financial support. Without them, I wouldn't be here, in more ways than one.

I would also like to thank my friend Cheezy for their unwavering belief that I would cure cancer someday. I might not have cured cancer yet, but we might be one step closer to detecting lung cancer at least.

Additionally, I would like to thank my legs for helping me move forward, and my digestive system for giving me the energy to continue doing so.

In a turn of events, I would like to express my ingratitude to the authors of the manga series SPY×FAMILY, 湯神くんには友達がいない (*Yugami-kun Doesn't Have Any Friends*), 先輩はおとこのこ (*Senpai wa Otokonoko*), スキップとローファー (*Skip and Loafer*), 通りがかりにワンポイントアドバイスしていくタイプのヤンキー (*One Point Advice Yankee*), and 推しの子 (*Oshi no Ko*). Why must they write such good stories that lure me away from bachelor-assignment-land and into mangaland?

And last, but certainly least, I would like to thank myself for demonstrating first hand how to (almost) fail in turning in the bachelor assignment on time. Never again do I wish to be behind schedule on a project of this scale and importance.

Frequentist vs. Empirical Bayes: A comparison of procedures for high-dimensional two-sample statistical inference, in particular for microRNA data

Wei-Ting Sun*

July, 2022

Abstract

MicroRNAs are short strands of RNA, and nearly 2000 human microRNAs have been identified. As their levels in bodily fluids can change with the presence of cancer, they have potential to be used in non-invasive diagnosis. For this, datasets consisting of microRNA levels of healthy people and cancer patients need to be analyzed to classify which microRNAs correlate significantly with a disease—an example of a two-sample statistical inference problem. In this paper we compare two procedures to such problems: frequentist and empirical Bayes. We create simulated datasets similar to a microRNA dataset, analyze them using the two procedures, and compare the results. We find that the two procedures produce differing classifications. At the same significance level, the empirical Bayes procedure tends to classify more microRNAs as correlating, but can produce erroneous results if there are too few people per group; the frequentist procedure tends to classify microRNAs as non-correlating but is consistent. Applying the procedures on a lung cancer microRNA dataset, we find hsa.miR.10a.5p, hsa.miR.204.3p, hsa.miR.424.3p, and hsa.miR.509.3p as microRNA plausibly correlated with lung cancer. We conclude that although improvements are necessary, the empirical Bayes procedure has the potential to solve two-sample statistical inference problems and find correlating microRNAs well.

Keywords: statistical inference, microRNA, lung cancer, frequentist, false discovery rate, empirical Bayes

1 Introduction

MicroRNAs are small non-coding RNAs (around 21–25 nucleotides) that play a role in regulating gene expression [5], with 1917 known human microRNAs [6]. Some microRNAs can be found in blood serum, urine, and other bodily fluids at relatively consistent levels, even across individuals [3]. As microRNAs affect gene expression, research has shown that their levels are correlated with the presence of diseases, notably cancer [5]. They thus have the potential to be used as biomarkers—indicators of some biological condition—for diseases, and with their presence in easily extractable bodily fluids, such tests could be done quite non-invasively, which would lead to earlier cancer detection overall.

Before the implementation of microRNA-based cancer diagnostics, some research still has to be done, among which is the identification of microRNAs whose levels in bodily fluids can be used as biomarkers for certain cancers. While such microRNAs can be

*Email: w.sun-2@student.utwente.nl

identified using biomedical approaches [9], given the existence of quantitative datasets on microRNA levels, mathematical approaches are possible and have been attempted with success [8].

Mathematically, this problem of deciding which microRNAs—if any—are significantly correlated with some disease falls under the class of classification problems. However, there are some peculiarities to this particular problem that make it especially challenging. One difficulty is the high-dimensionality in the number of microRNAs, which could incur the curse of dimensionality on methods, and in addition, would increase the chances of there being false positives—microRNAs that, due to chance, appear to significantly correlate with a disease when they, in fact, do not. Another difficulty is the usually small amount of data in microRNA datasets. This is at least partly due to the platform-dependence of microRNA level measurements—different microRNA-measuring machines measure different levels of microRNAs [7]. These pecularities together suggest that a more statistical approach to the problem may be useful, as there is more of a theoretical assurance of the probabilistic properties of the classification.

In this paper, we implement and compare two statistical approaches to such high-dimensional classification problems: frequentist and empirical Bayes. We do this by constructing simulated data that resemble our microRNA dataset, applying the two procedures, and comparing their results. Finally, in Section 6.1, we apply the two statistical approaches to our microRNA data and use what we learned from the results of the analysis to obtain a set of microRNAs that plausibly correlate with lung cancer.

2 The problem

The statistical problem we concern ourselves with is the *unpaired two-sample statistical inference problem*, which asks:

> Given two *groups* of (experimental) *subjects* (e.g. people), all of which have data for a set of *variables* (e.g. height, weight, age), how do we best *classify* the set of variables into two *types*—variables with (effectively) the same distribution between the two groups, and variables with a different distribution between the two groups?

The *dataset* of such a problem can be visualized as in Table 1.

TABLE 1: The data in an unpaired two-sample statistical inference problem in tabular form.

			"Variables"			
			$j = 1$	$j = 2$	$\dots$	$j = n$
"Subjects"	Group 0	$i = 1$	$y_{0,1,1}$	$y_{0,1,2}$	$\cdots$	$y_{0,1,n}$
		$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
		$i = m_0$	$y_{0,m_0,1}$	$y_{0,m_0,2}$	$\cdots$	$y_{0,m_0,n}$
	Group 1	$i = 1$	$y_{1,1,1}$	$y_{1,1,2}$	$\cdots$	$y_{1,1,n}$
		$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
		$i = m_1$	$y_{1,m_1,1}$	$y_{1,m_1,2}$	$\cdots$	$y_{1,m_1,n}$

The number of variables in the dataset is denoted by n. We denote the two groups of subjects as *group 0* and *group 1*, with m_0 and m_1 subjects per group respectively. The goal is to classify each of the variables into one of two types. The two types of variables are denoted *type 0* (effectively the same distribution of variable values between groups) and *type 1* (different distributions of variable values between groups). All variables are to be classified as exactly one of these two types.

In the context of the microRNA data we would like to investigate, the *subjects* are the individual people whose microRNA concentrations are recorded, the *groups* are the two groups of people—healthy people (group 0) and lung cancer patients (group 1)—and the *variables* are the distinct microRNAs. The data point $y_{k,i,j}$ is then the recorded concentration of microRNA j in the urine of person i of group k (healthy or with lung cancer).

3 Data

The microRNA data we are working with consist of microRNA concentrations in the samples of urine of 30 people. 14 of them are healthy (at least, as far as we know regarding lung cancer), 2 have stage 2B lung cancer, and 14 have stage 3 or 4 lung cancer. We merge the two lung cancer groups into one with 16 people, making it into a two-sample statistical inference problem.

The raw data consist of microRNA counts—the number of microRNAs of each type counted by a PCR (polymerase chain reaction) process. Note that these counts are not the "actual counts" of the microRNAs in the urine sample—as the PCR process repeatedly replicates the microRNAs—though they do preserve the relative concentrations of the different microRNAs in the sample. The microRNA counts are systematically biased per person: for some people, their recorded microRNA counts are systematically higher (roughly by some factor) than the average, and for some others, their counts are systematically lower. As this phenomenon is quite systematic, it is likely that this is an artifact of using the PCR process to quantify microRNA levels. We thus attempt to correct for this phenomenon by normalizing the microRNA counts—for each subject, we scale all their microRNA counts such that the (arithmetic) mean microRNA count is 1.

There are 1956 microRNAs in the dataset. A proportion of microRNAs, however, have counts of 0 for all 30 subjects. We remove these microRNAs from the dataset, leaving 1421 microRNAs with data. This is then the microRNA dataset we work with and analyze in Section 6.1.

3.1 Simulated data

To be able to analyze how the two procedures compare, we need to have datasets for which we know the actual underlying distributions, and for that we create simulated data. We try to make the simulated datasets somewhat resemble the actual dataset so that results obtained from analyzing the simulated data can apply to our analysis of the actual microRNA dataset.

Looking at the microRNA dataset, we observe that most entries—84.7%—are 0. We reflect this in our simulated data; we have our simulated data follow a *zero-inflated model*—each data point has some probability of having the value of 0, and only if not does it follow some other, more varied distribution.

Regarding the other distribution, for convenience, we choose to use a log-normal distribution. While the distribution of the actual microRNA dataset looks more akin to an exponential distribution, for the simulated dataset we would like to have a measure for how much the distributions of the two groups deviate from each other, and the log-normal distribution works well for this. As a reminder, if $X \sim \mathcal{N}(\mu, \sigma^2)$ is normally distributed with mean μ and variance σ^2, then e^X is log-normally distributed with the same parameters μ and σ^2, denoted $e^X \sim \text{Lognormal}(\mu, \sigma^2)$. Note that because of this definition, μ and σ^2 are *not* the mean and variance respectively of the log-normally distributed variable, but rather of the underlying normal distribution. Note also that the log-normal distribution is always non-negative, just like microRNA data.

With log-normally distributed data, we can introduce the concept of the *deviation* d—if $Y_1 \sim \text{Lognormal}(\mu_1, \sigma^2)$ and $Y_2 \sim \text{Lognormal}(\mu_2, \sigma^2)$ (with the same variance parameter), then their deviation is

$$d := \frac{|\mu_2 - \mu_1|}{\sigma}. \tag{1}$$

This is, in fact, just the number of standard deviations away the two means of the underlying normal distribution are. The deviation also captures how easy it is to tell apart the data of two distributions—the larger the deviation, the more standard deviations away the two variables' means are, and so the easier it is to tell apart the two distributions. Furthermore, if two pairs of variables have the same deviation, then they are equally easy to tell apart—we can shift and scale the underlying normal distributions of one pair to match those of the other pair, and thus there is a bijective and probability-distribution-preserving map between the two pairs of log-normal random variables.

In the real world, when there are many variables that affect a phenomenon, usually there are only a few that have a large effect, while for most variables the effect is small. We expect this to also be true for the microRNA data: only a few microRNAs should correlate with lung cancer, while most have little correlation, if any at all. This is reflected in the simulated data—we let the deviation d_j of variable j be

$$d_j = c \cdot e^{-(j-1)/a}, \tag{2}$$

where c and a are parameters. The parameter c is the maximum deviation, and the parameter a is (approximately) proportional to the "number of variables with a big deviation". Using these deviation values, we see that only the first few variables will have a large difference in distributions between groups (if c is large enough), but the variables at the end—or at least have indices much larger than a—will have deviation values near 0, i.e. there is almost no difference in distribution between groups. Note that there are in fact no variables with a deviation of 0—no variables are, in fact, of type 0—though many are close.

Another aspect of the data is dependency. It is expected that some microRNAs have strong correlations with each other, and both positive and negative correlations are possible. However, as we will explore in detail in Section 4.1 and Figure 1, correlations between the microRNAs are on average close to 0. Hence, for simplicity, we let all data points in the simulated data be independent of each other.

With this, we can specify the model. Each data point $y_{k,i,j}$ in the two-sample statistical inference problem is a sample of a random variable $Y_{k,i,j}$, all such random variables being independent. We set the probability of the random variable being zero to 2/3. Additionally, as all pairs of log-normal variables with the same deviation have are equally hard to distinguish, the log-normal parameter σ^2 is set to 1 for all data points, $\mu = 0$ for data in group 0, and $\mu = \pm d$ for data in group 1, with a 1/2 probability of either. In summary, the data are distributed as

$$Y_{0,i,j} \sim \begin{cases} 0 & \text{with probability } 2/3 \\ \text{Lognormal}(\mu = 0, \sigma^2 = 1) & \text{otherwise} \end{cases}, \tag{3}$$

$$Y_{1,i,j} \sim \begin{cases} 0 & \text{with probability } 2/3 \\ \text{Lognormal}(\mu = \pm d_j, \sigma^2 = 1) & \text{otherwise} \end{cases}. \tag{4}$$

We generate datasets for different combinations of the parameters c and a, as well as the number of subjects per group m and the number of variables n. The parameters m and n take on values $m \in \{10, 20, 40, 80\}$ and $n \in \{500, 1000, 2000, 4000\}$, similar to the size of the microRNA dataset. For each combination of m and n, we generate one dataset with $d_j = 0$ for all variables j, as well as one dataset for each combination of $c \in \{1, 2, 3, 4\}$ and $a \in \{8, 16, 32, 64\}$. In total, the simulated data then consist of 272 datasets. Our implementation in R can be found in Appendix B.1.

4 The procedures

In this section we describe the frequentist and empirical Bayes procedures in detail. However, let us first remind ourselves what a "procedure" precisely is. A *procedure*, in the context of an unpaired two-sample statistical inference problem, is an algorithm that, given a dataset as in Table 1, outputs two complementary sets of variables—one with the variables that the algorithm classifies as type 0, and the other for variables classified as type 1.

Additionally, as this concept will be useful to us, let us also remind ourselves what exactly a "statistic" is: a statistic T of a random variable $Y \in \mathcal{Y}$ is simply[1] a function $T : \mathcal{Y} \to \mathbb{R}$. Furthermore, statistics can be "parametric" and "non-parametric". A statistic T of a random variable Y is *parametric* if the probability distribution of $T(Y)$ can only be known if we assume Y is of a distribution with a fixed set of *parameters*; otherwise, T is *non-parametric*—we can know the distribution of $T(Y)$ without assumptions on the family of distributions Y belongs to. As an example of a non-parametric statistic, let $T_1(Y_1, Y_2) := \text{sign}(Y_1 - Y_2)$. Then if Y_1 and Y_2 have the same distribution and are continuous—for all values of y, the probability that $Y_1 = y$ is 0—we know the distribution of $T_1(Y_1, Y_2)$, namely it has values -1 and 1 each with probability $1/2$. On the other hand, $T_2(Y_1, Y_2) := Y_1 + Y_2$ is a parametric distribution, since we need to know the exact distributions of Y_1 and Y_2 to know the distribution of $T_2(Y_1, Y_2)$.

4.1 Frequentist procedure

The suite of statistical methods including p-values, hypothesis testing, and confidence intervals is collectively called *frequentist statistics*. At the core of frequentist statistics are the ideas that the "probability" of an event is the limit of the relative frequency of that event happening as the number of trials goes to infinity, and that model parameters (e.g. μ and σ in a normal distribution) have fixed but (usually) unknown values. In essence, frequentist statistics encompasses what is taught in introductory statistics courses.

The standard frequentist approach to the current problem is *two-sample hypothesis testing*: what it can do is determine, for a single variable, whether the variable has a different distribution between two groups, and with a *false positive rate*—the probability of classifying a variable as type 1 when it is in fact type 0—no greater than some predetermined significance level α. Its basic steps are as follows:

1. Set a significance level α.

2. Let the *null hypothesis* $\mathcal{H}_0$—the "default" hypothesis—be that the variable has the same distribution in the two groups, and the *alternative hypothesis* $\mathcal{H}_1$ be its complement—that the variable has different distributions in the two groups.

3. Select a statistic that is sensitive to differences in distribution between the data of two groups, and that has a known distribution under the null hypothesis.

4. Apply the statistic on the observed data and from that calculate the p-value. This is the probability of observing a statistic value at least as "statistically significant" as the observed value under the null hypothesis. In this case, this means the probability that the statistic has a value corresponding to a difference in data at least as large as the observed difference.

[1] Technically, functions also have to satisfy some measurability condition to be a statistic, but that is irrelevant to the scope of this paper.

5. Reject the null hypothesis $\mathcal{H}_0$ if and only if $p \leq \alpha$.

For the choice of statistic, since we are agnostic about the distribution of microRNA concentrations, we would like to use a non-parametric statistic to test the difference in distributions of data between groups. We choose the *two-sample Kolmogorov–Smirnov test*—i.e. statistic and p-value calculation—which is a test on whether there is a difference in distribution between the data of two groups, exactly what we want. Its statistic, the Kolmogorov–Smirnov two-sample statistic, is

$$D_{m_0,m_1} := \sup_y \left| F_{0,m_0}(y) - F_{1,m_1}(y) \right|, \tag{5}$$

where $F_{0,m_0}(y)$ and $F_{1,m_1}(y)$ are the empirical cumulative mass functions of the group 0 and 1 data respectively—in other words, they are the proportion of observed data points of a variable that are less than or equal to y for groups 0 and 1 respectively. If the two groups have the same data distribution, then D_{m_0,m_1} follows a known distribution, meaning that the p-value can be calculated. The Kolmogorov–Smirnov test can be performed in R using the built-in `ks.test` function, which, given the data as input, outputs, among other things, the p-value of the test.

Multiple testing

Looking at the hypothesis testing procedure, notice that the it treats the two types of variables differently on a fundamental level—the hypothesis that a variable is of type 0 is seen as the "default", and additionally, the p-value is a probability where it is given that the null hypothesis is true. From the definition of the p-value above, we obtain that for every α, it holds that

$$\mathbb{P}(\mathcal{H}_0 \text{ rejected} \,|\, \mathcal{H}_0) = \mathbb{P}(p \leq \alpha \,|\, \mathcal{H}_0) \leq \alpha. \tag{6}$$

Thus we see that the frequentist procedure controls the false positive rate to be at most α.

However, applying hypothesis testing to multiple variables breaks the false positive rate control that the procedure aims to achieve; while the false positive rate of each variable still holds, the overall rate that at least one null hypothesis is rejected is not at most α. This is especially apparent when the number of variables is high and the number of type 1 variables is small, as in our microRNA problem. For example, if there are $n = 1000$ variables, with 990 of them of type 0 and only 10 of them of type 1, and we conduct a two-sample hypothesis test with $\alpha = 0.05$, then the expected number of false positives (classified as type 1 but are type 0) would be around $990\alpha = 49.5$. If we instead look at the probability of there being no false positives, it would be

$$\mathbb{P}(0 \text{ false positives}) = (1-\alpha)^{990} \approx 8.8 \cdot 10^{-23}$$

if all the variables are independent of each other. In short, there *will* be false positives.

In addition, in this example, the number of true positives (classified as type 1 and are type 1) would be maximally 10, much less than the expected 49.5 false positives. Hence, out of the rejected (classified as type 1) variables, we would expect most of them to be, in fact, of type 0, which mostly defeats the purpose of hypothesis testing in the first place.

One way of resolving this issue is to control not the false positive rate, but instead the *false discovery rate* (*FDR*). It is defined as

$$\text{FDR} := \mathbb{E}\left[\frac{\#\text{ of false rejections}}{\#\text{ of total rejections}}\right] \quad \text{with} \quad \frac{0}{0} := 0. \tag{7}$$

If we can ensure that FDR $\leq \alpha$ for some small α, we can be fairly sure that most of the rejections are true rejections—i.e. most variables classified as type 1 are actually of type 1.

To control the FDR—to ensure that FDR $\leq \alpha$ holds—we use the *Benjamini–Hochberg procedure*. If we have n variables, we first calculate the p-values of their respective two-sample hypothesis tests. Let $p_{(1)}, \ldots, p_{(n)}$ be the sorted (from smallest to largest) p-values, and let $\mathcal{H}_{0,(1)}, \ldots, \mathcal{H}_{0,(n)}$ be the null hypotheses corresponding to the sorted p-values. The rejection procedure is then as follows [1]:

1. Let $k := \max\left\{j : p_{(j)} \leq \frac{j}{n}\alpha\right\}$.
2. If no such j exists, no null hypotheses are rejected. Otherwise, reject null hypothesis $\mathcal{H}_{0,(j)}$ if and only if $j \leq k$.

Equivalently, the procedure can also be expressed in terms of *adjusted p-values*—instead of keeping the p-values the same and adjusting the rejection criterion, we could also *adjust* the p-values and keep the rejection criterion the same: reject the adjusted p-values that are less than or equal to α. For the jth smallest p-value $p_{(j)}$, its Benjamini–Hochberg adjusted p-value $\widetilde{p}_{(j)}$ is then

$$\widetilde{p}_{(j)} := \min\left\{1, \min_{j \leq i \leq n} \frac{p_{(i)} n}{i}\right\}, \tag{8}$$

where the outer minimum serves to constrain the adjusted p-value to within the interval [0, 1], mimicking the p-value.

One caveat is, however, that by following this procedure, the FDR is controlled to be at most α only under certain conditions. Benjamini and Hochberg proved that the FDR is controlled by their procedure if the test statistics of true null hypotheses are independent [1], and Benjamini and Yekutieli proved that the FDR is controlled even if the test statistics of true null hypotheses are "positively dependent"[2] [2].

In the context of two-sample multiple hypothesis testing, this means that for every variable, if the variable contains data with a big difference between groups, then other variables will also tend to have bigger differences between groups, as compared to if the variable had a smaller difference between groups. We check to see if this is true for our microRNA data; we do this by calculating Spearman's rank correlation coefficient ρ between each pair of variables—if there is a positive dependency in the difference in data between groups across variables, it should be reflected as a positive correlation between variable data. A histogram of the correlation coefficient values can be found in Figure 1.

We find that the mean correlation coefficient is $\rho \approx 0.0294$, which suggests a slight positive correlation, though the dependency could also actually be zero and this slight positive value be due to chance. Either way, as this is not negative, we have no reason to doubt that the difference in data between groups has a positive or no dependency across variables, and hence we conclude that we can use the Benjamini–Hochberg procedure to control the false discovery rate to our desired level α.

Our implementation of the frequentist procedure in R can be found in Appendix B.2.

[2]Technically, they proved that the FDR is controlled under the condition that there is a "positive regression dependency on the subset of statistics corresponding to true null hypotheses", which is a weaker and more complex condition that implies positive dependency.

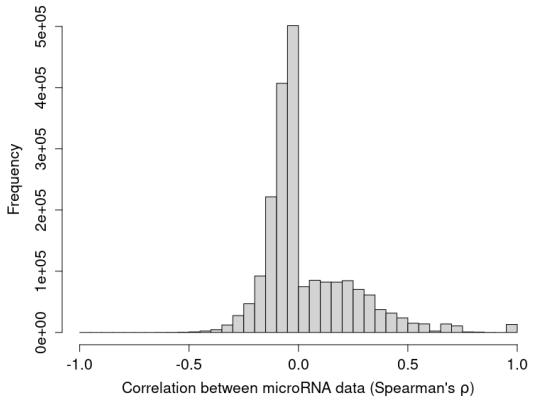

FIGURE 1: Histogram of Spearman's rank correlation coefficient between the microRNA data (both group 0 and group 1) of all pairs of microRNAs (including self-correlations).

4.2 Empirical Bayes procedure

Another approach to statistics is *Bayesian statistics*. Unlike frequentist statistics, Bayesian statistics treats the "probability" of an event as being the degree of subjective belief of the event happening, and because of that, treats model parameters as random variables. This makes it possible to assign probabilities to, for example, model parameters being in some interval.

"Empirical Bayes" is the name given to procedures that align with the Bayesian view of statistics, but that—unlike pure Bayesian procedures—obtain some of their model values through *empirical* estimation. The specific empirical Bayes procedure for two-sample inference we use is adapted from Efron et al. [4], which we describe below.

First, we assume that there are the two types of variables: those with the same distribution in the two groups (type 0), and those with a different distribution (type 1). Our goal is then, for each variable j, to obtain the probability

$$\mathbb{P}(\text{variable } j \text{ is of type } 0 \,|\, \text{the data of variable } j). \tag{9}$$

A problem, however, is that (9) is difficult to calculate—we would need to know the distribution of the data, which is high-dimensional. To resolve this, we define a statistic Z that, for each variable, reduces the high-dimensional data of the variable into a scalar. The statistic should capture the "amount of deviation" between the distributions of the data in the two groups. We denote the output of the statistic Z on the data of variable j as z_j, which we call the *score* of the variable. With this, we can estimate (9) with a much easier to calculate probability

$$\mathbb{P}(\text{variable } j \text{ is of type } 0 \,|\, Z = z_j). \tag{10}$$

We now examine the distribution of Z for variables of the two groups. Let $f_0(z)$ and $f_1(z)$ be the probability densities of the distribution of Z among variables of type 0 and type 1 respectively, and we assume that the densities are both always non-zero, i.e. $\forall z \in \mathbb{R} : f_0(z) > 0$ and $f_1(z) > 0$—this assumption prevents division by zero later on. Also, let p_0 denote the proportion of variables that are of type 0, and $1 - p_0$ then the proportion of type 1 variables. Then

$$f(z) = p_0 f_0(z) + (1 - p_0) f_1(z) \tag{11}$$

is the probability density of Z of a variable of unknown type, the *mixed distribution*. Furthermore, by Bayes' rule, we have that

$$\mathbb{P}(\text{type } 0 \,|\, Z = z_j) = \frac{\mathbb{P}(Z = z_j \,|\, \text{type } 0)\,\mathbb{P}(\text{type } 0)}{\mathbb{P}(Z = z_j)} = \frac{f_0(z_j)p_0}{f(z_j)}. \tag{12}$$

As $\mathbb{P}(\text{type } 0 \,|\, Z = z_j)$ is a probability and thus satisfies $0 \leq \mathbb{P}(\text{type } 0 \,|\, Z = z_j) \leq 1$ for all possible z, we have that

$$\forall z \in \mathbb{R} : 0 \leq \frac{f_0(z)p_0}{f(z)} \leq 1. \tag{13}$$

We can thus bound the constant p_0:

$$0 \leq p_0 \leq \inf_{z\in\mathbb{R}} \frac{f(z)}{f_0(z)} = 1 \Big/ \sup_{z\in\mathbb{R}} \frac{f_0(z)}{f(z)}. \tag{14}$$

This means that if we can estimate $f_0(z)/f(z)$, we can calculate (estimated) upper bounds for p_0 and $\mathbb{P}(\text{type } 0 \,|\, Z = z_j)$.

Type 0 scores

While we can estimate $f(z)$ from the Z statistic values of our data—$\{z_j \,|\, 1 \leq j \leq n\}$, the *mixed scores*—we currently do not have data to estimate $f_0(z)$. To resolve this, we generate scores that should behave like scores from type 0 variables and thus be distributed with density $f_0(z)$—we call these *type 0 scores*. The approach we use here is to recombine the data from the two groups such that each of the recombined groups has the same ratio of data points from the two original groups, and apply the Z statistic on the recombined groups to simulate type 0 scores.

First, within each of the two groups, randomly shuffle the order of the subjects. The two groups have m_0 and m_1 subjects respectively, meaning that both recombined groups should have a ratio of $m_0 : m_1$ for the number of subjects from the original groups. We would also like to conserve the number of subjects per group. Altogether, this means that the recombined group 0 should thus have $m_0^2/(m_0 + m_1)$ and $m_0m_1/(m_0 + m_1)$ subjects from the original groups 0 and 1 respectively, and the recombined group 1 should have $m_0m_1/(m_0+m_1)$ and $m_1^2/(m_0+m_1)$ subjects from the original groups 0 and 1 respectively. In other words, to get the recombined groups, we simply swap the first $m_0m_1/(m_0 + m_1)$ subjects between the shuffled groups.

However, $m_0m_1/(m_0 + m_1)$ is not guaranteed to be an integer if $m_0 \neq m_1$, so we instead swap $s + U$ subjects, where

$$s = \left\lfloor \frac{m_0m_1}{m_0 + m_1} \right\rfloor, \quad \text{and} \quad U = \begin{cases} 1 & \text{with probability } \frac{m_0m_1}{m_0+m_1} - s \\ 0 & \text{otherwise} \end{cases}. \tag{15}$$

Then $s + U \in \mathbb{Z}$ and $\mathbb{E}[s + U] = m_0m_1/(m_0 + m_1)$, and so on average, $m_0m_1/(m_0 + m_1)$ subjects are swapped between groups.

To finally obtain the type 0 scores, the statistic Z is then applied to the dataset with the recombined groups. Note that for each variable, the data are shuffled the same way—this is done so that systematic biases across at the subject-level are preserved—if all the data for a certain subject are systematically double that of other subjects, this doubling is preserved in the recombined groups. Note also that the original groups can be randomly shuffled and U randomly sampled multiple times to generate more type 0 scores. Mirroring Efron et al.'s procedure, we perform this 20 times to generate $20n$ type 0 scores in total [4].

Logistic regression

The function $f_0(z)/f(z)$ is hard to estimate. This is because we only have samples from distributions with $f_0(z)$ and $f(z)$ as their density functions—namely the mixed and type 0 scores. Note that we are to estimate a function from data at a set of discrete points, and the values of the mixed and type 0 scores are different. There is hence no straightforward way of estimating $f_0(z)/f(z)$.

As this method is *empirical* Bayes, we estimate $f_0(z)/f(z)$ empirically, using *logistic regression*. In regression problems, the goal is to, given some data, find a function out of a class of functions $\mathcal{F}$ that "best fits the data"—that minimize some *loss function* L involving the dependent variable(s) and the regression function values at the corresponding values of the explanatory variable(s). For example, in linear regression, the class of regression functions is the set of order 1 polynomials—$\mathcal{F} = \{f(x) = c_0 + c_1 x \,|\, c_0, c_1 \in \mathbb{R}\}$—and the loss function is usually least squares—the sum of the squared differences between the dependent variable values and the regression function values is minimized.

In (simple) logistic regression, the data consist of two sets of values of explanatory variables—a set for observed "failures" and one for observed "successes"—and the aim is to fit a curve that estimates the empirical probability of success given the value of the explanatory variable—a single explanatory variable, hence "simple". Logistic regression achieves this by maximizing the probability of observing the given data if the true probability of success is given by the regression function. With failures having explanatory variable with values $\{x_{0,i} \,|\, 1 \le i \le n_0\}$ and weights $\{w_{0,i} \,|\, 1 \le i \le n_0\}$, and successes with values $\{x_{1,j} \,|\, 1 \le j \le n_1\}$ and weights $\{w_{1,j} \,|\, 1 \le j \le n_1\}$, in addition to fitting some regression function class $\mathcal{F} = \{p_\theta(x) \,|\, \theta \in \Theta\}$ with parameter θ (exact function class not yet specified), the loss function of logistic regression is the reciprocal of the observation probability:

$$L = \left(\prod_{i=1}^{n_0} (1 - p_\theta(x_{0,i}))^{w_{0,i}} \right)^{-1} \left(\prod_{j=1}^{n_1} (p_\theta(x_{1,j}))^{w_{1,j}} \right)^{-1}. \tag{16}$$

Equivalently, we can instead minimize $\log L$,

$$\log L = -\sum_{i=1}^{n_0} w_{0,i} \log(1 - p_\theta(x_{0,i})) - \sum_{j=1}^{n_1} w_{1,j} \log(p_\theta(x_{1,j})), \tag{17}$$

as the logarithm function is strictly increasing.

The weights in logistic regression, as the name suggests, scale the magnitude of the contribution of the respective observations to the loss function—if an observation has weight $w \in \mathbb{N}$, it has the same effect as w of the same observations, each with weight 1.

Now, given this loss function, we would like to show that its minimum indeed corresponds to a fitted curve estimating the "local empirical probability of success". First, we define the *local empirical density* of failures and successes as, respectively,

$$N_{\epsilon,0}(x) := \frac{1}{\epsilon} \# \left\{ x_{0,i} \,\middle|\, |x_{0,i} - x| \le \frac{\epsilon}{2} \right\}, \quad N_{\epsilon,1}(x) := \frac{1}{\epsilon} \# \left\{ x_{1,j} \,\middle|\, |x_{1,j} - x| \le \frac{\epsilon}{2} \right\}, \tag{18}$$

with $\epsilon > 0$. This is intuitively the density of the observed explanatory variable values averaged over a window with width ϵ.

Let us now consider a situation akin to a coin toss—there are no explanatory variables, the probability of success is unknown, and we observe, from $n_0 + n_1 > 0$ trials, n_0 failures and n_1 successes, all with positive and equal weight w. The local empirical density of

successes and failures is then $N_{\epsilon,0} = n_0/\epsilon$ and $N_{\epsilon,1} = n_1/\epsilon$ respectively, as there are no explanatory variables. Then the corresponding logistic regression would be

$$\begin{aligned}
\hat{p} &\in \operatorname*{argmin}_{p\in[0,1]} \log L \\
&= \operatorname*{argmin}_{p\in[0,1]} \left\{ -\sum_{i=1}^{n_0} w\log(1-p) - \sum_{j=1}^{n_1} w\log(p) \right\} \\
&= \operatorname*{argmin}_{p\in[0,1]} \Big\{ -n_0 w\log(1-p) - n_1 w\log(p) \Big\} \\
&= \operatorname*{argmax}_{p\in[0,1]} \Big\{ n_0\log(1-p) + n_1\log(p) \Big\}.
\end{aligned}$$

Taking the derivative with respect to p, we obtain that

$$\frac{\partial}{\partial p}\big(n_0\log(1-p) + n_1\log(p)\big) = -\frac{n_0}{1-p} + \frac{n_1}{p} = 0 \iff p = \frac{n_1}{n_0+n_1}. \tag{19}$$

As $\log(1-p)$ and $\log(p)$ blow down to $-\infty$ when $p = 1$ and 0 respectively, we have that

$$\hat{p} = \frac{n_1}{n_0+n_1} = \frac{\epsilon N_{\epsilon,1}}{\epsilon N_{\epsilon,0} + \epsilon N_{\epsilon,1}} = \frac{N_{\epsilon,1}}{N_{\epsilon,0}+N_{\epsilon,1}} \tag{20}$$

is the argmax of $n_0\log(1-p) + n_1\log(p)$, and hence is the fitted value. Thus as claimed, the fitted logistic regression curve estimates the "local empirical probability of success". In addition, we note that the fitted value is independent of the weight w, from which we can conclude that scaling the weights of all observations by the same positive real number does not change the output of the logistic regression.

If we apply this to our data, where the explanatory variable is the score z, and the type 0 and mixed scores are treated as "failures" and "successes" respectively, with n_0 type 0 scores and n mixed scores, the fitted logistic regression curve will be an estimate for

$$\frac{N_{\epsilon,1}(z)}{N_{\epsilon,0}(z) + N_{\epsilon,1}(z)}. \tag{21}$$

The function we wish to estimate is, on the other hand, $f_0(z)/f(z)$. We first notice that $f_0(z)$ can be estimated as

$$f_0(z) = \lim_{\epsilon\downarrow 0} \frac{1}{\epsilon}\mathbb{P}\Big(z - \frac{\epsilon}{2} \le Z < z + \frac{\epsilon}{2} \,\Big|\, \text{type } 0\Big) \approx \frac{N_{\epsilon,0}(z)}{n_0},$$

and similarly, $f(z) \approx N_{\epsilon,1}(z)/n$, so we get that

$$\frac{f_0(z)}{f(z)} \approx \frac{N_{\epsilon,0}(z)/n_0}{N_{\epsilon,1}(z)/n}. \tag{22}$$

To transform the estimate given by the logistic regression (Equation (21)) to the desired estimate (Equation (22)), we first adjust the weights. As a data point with weight w is equivalent to w data points each with weight 1, the sum of the weights is the "effective" total number of data points. Hence if we let $w_{0,i} = 1/n_0$ for all $1 \le i \le n_0$ and $w_{1,j} = 1/n$ for all $1 \le j \le n$, then the logistic regression curve will estimate

$$\frac{N_{\epsilon,1}(z)/n}{N_{\epsilon,0}(z)/n_0 + N_{\epsilon,1}(z)/n}. \tag{23}$$

And finally, $\frac{N_{\epsilon,1}(z)/n}{N_{\epsilon,0}(z)/n_0+N_{\epsilon,1}(z)/n}$ can be transformed into $\frac{N_{\epsilon,0}(z)/n_0}{N_{\epsilon,1}(z)/n} \approx f_0(z)/f(z)$ by the mapping

$$x \mapsto \frac{1}{x} - 1. \tag{24}$$

To summarize, the function $f_0(z)/f(z)$ can be estimated from (simulated) type 0 scores $\{z_{0,i} \,|\, 1 \le i \le n_0\}$ and mixed scores $\{z_j \,|\, 1 \le j \le n\}$ using logistic regression with loss minimization

$$\hat{\theta} \in \operatorname*{argmin}_{\theta} \left\{ -\sum_{i=1}^{n_0} \frac{1}{n_0} \log(1 - p_\theta(z_{0,i})) - \sum_{j=1}^{n} \frac{1}{n} \log(p_\theta(z_j)) \right\}, \tag{25}$$

from which we obtain $p_\theta(z)$ as an estimator for Equation (23), and so an estimator of $f_0(z)/f(z)$ would be

$$\frac{1}{p_{\hat{\theta}}(z)} - 1. \tag{26}$$

Thus, plugging the estimate into Equation (14), we obtain bounds for $\widehat{p_0}$, our estimate of p_0,

$$0 \le \widehat{p_0} \le 1 \Big/ \sup_{z \in \mathbb{R}} \left\{ \frac{1}{p_{\hat{\theta}}(z)} - 1 \right\}, \tag{27}$$

as well as—for all $1 \le j \le n$—bounds for $P_0(z_j)$, our estimate for $\mathbb{P}(\text{type } 0 \,|\, Z = z_j)$:

$$0 \le P_0(z_j) \le \left(\frac{1}{p_{\hat{\theta}}(z_j)} - 1 \right) \Big/ \sup_{z \in \mathbb{R}} \left\{ \frac{1}{p_{\hat{\theta}}(z)} - 1 \right\}. \tag{28}$$

We thus use the upper bound of $P_0(z_j)$ as our estimate for

$$\mathbb{P}(\text{variable } j \text{ is of type } 0 \,|\, \text{the data of variable } j), \tag{29}$$

the probability we originally wanted to calculate.

Implementation in R

In the implementation, one decision to make is with the choice of statistic Z. In Efron et al. [4], where the data consist of paired samples, the chosen statistic is, as a function of the set of differences for the jth variable $y_j = \{y_{i,j} \,|\, 1 \le i \le m\}$,

$$Z(y_j) = \frac{\overline{y_j}}{a_0 + \mathrm{sd}(y_j)}, \tag{30}$$

where $\overline{y_j}$ is the sample mean of y_j and $\mathrm{sd}(y_j)$ the sample standard deviation. The number a_0 is the 90th percentile of sample standard deviations $\mathrm{sd}(y_j)$ across variables—90% of values of $\mathrm{sd}(y_j)$ are less than or equal to a_0; Efron et al. found that adding this constant produced better results.

Our data consist of *unpaired* samples, with $y_{0,j} = \{y_{0,i,j} \,|\, 1 \le i \le m_0\}$ and $y_{1,j} = \{y_{1,i,j} \,|\, 1 \le i \le m_1\}$ being our group 0 and group 1 data for the jth variable respectively.

We implement an analogous statistic—inspired additionally by the two-sample t-test—for our unpaired two-sample case:

$$Z(y_{0,j}, y_{1,j}) = \frac{\overline{y_{1,j}} - \overline{y_{0,j}}}{(a_0 + s_j)\sqrt{\frac{1}{m_0} + \frac{1}{m_1}}}, \quad s_j^2 = \frac{(m_0 - 1)\,\mathrm{sd}^2(y_{0,j}) + (m_1 - 1)\,\mathrm{sd}^2(y_{1,j})}{m_0 + m_1 - 2}, \tag{31}$$

where m_0 and m_1 are the sample sizes of $y_{0,j}$ and $y_{1,j}$—the sizes of group 0 and 1—respectively, s_j is the pooled sample standard deviation for variable j, and a_0 is, again, the 90th percentile of s_j across variables. The addition of the term involving the sample sizes serves to let the distribution of Z be roughly the same across different sample sizes, as in our analysis we apply this procedure to datasets with varying sample sizes.

Another implementation decision is in the choice of family of logistic regression functions $\mathcal{F} = \{p_\theta(z) \,|\, \theta \in \Theta\}$. We would like it to be continuous and furthermore differentiable, so that the fitted curve would be less prone to overfitting. The family of polynomials, while a natural choice, is not suitable, as they are locally not very "flexible". Instead, for simplicity and convenience, we use cubic splines, which are more locally flexible, and are also used in [4]. We choose to use cubic splines with 6 degrees of freedom—6 cubic polynomials connected together—6 being enough to not underfit but is otherwise arbitrary.

Additionally, since $p_\theta(z)$ represents a probability, it should stay within [0, 1] in the whole domain. We enforce this by applying the standard logistic function

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \tag{32}$$

which is always between 0 and 1, to the cubic spline. Hence our regression function class is

$$\mathcal{F} = \{f(z) = \sigma(f_3(z)) \,|\, f_3(z) \text{ is a 6-segment cubic spline}\}. \tag{33}$$

The logistic regression is thus done in R with the function

```
glm(y ~ ns(z, df = 6), family = binomial, weights = w),
```

where `z` is an array containing both $\{z_{0,j} \,|\, 1 \le j \le n_0\}$ and $\{z_j \,|\, 1 \le j \le n\}$, the array `w` is the array of corresponding weights, and the array `y` has entries either 0 or 1 depending on whether the corresponding entry of `z` is a failure or success. The function `ns` from the `splines` package returns a natural cubic spline of its input with `df` degrees of freedom, and the function `glm` fits a generalized linear model, and with the option `family = binomial`, uses the logistic loss function.

As $p_{\hat{\theta}}(z)$ is already an estimate, in the implementation, the value of the supremum $\sup_{z \in \mathbb{R}} \left\{\frac{1}{p_{\hat{\theta}}(z)} - 1\right\}$, is simply estimated. The value of $\frac{1}{p_{\hat{\theta}}(z)} - 1$ is evaluated for a dense and equidistant grid of values of z spanning the range of the statistic Z, as well as for the regression points $\{z_{0,j} \,|\, 1 \le j \le n_0\}$ and $\{z_j \,|\, 1 \le j \le n\}$, and the maximum value is taken as an estimate for $\sup_{z \in \mathbb{R}} \left\{\frac{1}{p_{\hat{\theta}}(z)} - 1\right\}$. Note that this almost always underestimates the true value of the supremum, but since the function is continuously differentiable, the estimate approaches the true value as the grid size approaches 0.

Our implementation of the empirical Bayes procedure in R can be found in Appendix B.3.

5 Results

Having performed the two procedures—frequentist and empirical Bayes—on the simulated data, we would like to compare their results.

5.1 Similarity of procedures

First, we would like to investigate how similar the outputs of the two procedures are. We have reasons to believe that the two procedures should produce similar results. Using strong FDR-control for adjusting the p-value and rejecting at some significance level α, it holds that (with the convention that $0/0 = 0$)

$$\mathbb{E}\left[\frac{\#\text{ of false rejections}}{\#\text{ of total rejections}}\right] \leq \alpha, \tag{34}$$

i.e. that out of all rejected variables (variables classified as type 1), we would expect the proportion of false rejections to be around α or less. In other words, we should expect that

$$\mathbb{P}(\text{variable } j \text{ is of type } 0 \,|\, \text{variable } j \text{ classified as type } 1) \lessapprox \alpha \tag{35}$$

to approximately hold. This matches the outcome of the empirical Bayes procedure: if we classify a variable j as type 1 if and only if $P_0(z_j)$ is less than some β, then out of all variables classified as being of type 1, we have that (approximately)

$$\mathbb{P}(\text{variable } j \text{ is of type } 0 \,|\, \text{variable } j \text{ classified as type } 1) \leq \beta. \tag{36}$$

Hence if we take $\alpha = \beta$, we would expect the variables classified as type 1 to have a similar probability of actually having a significantly different distribution between the two groups, and so we would then also expect that the two procedures output similar classifications.

If the two outputs match up well, we would expect the adjusted p-value of variable j from the frequentist method and the probability $P_0(z_j)$ that the variable is of type 0 of the empirical Bayes method to correlate well. Similarly, if the two outputs do not match up, we would expect the correlation to be low. We thus calculate the sample correlation between the adjusted p-value and $P_0(z_j)$ of all variables in the simulated data. We choose to calculate Spearman's rank correlation coefficient ρ as it captures the *monotonicity* of the relationship—$|\rho| = 1$ if the data are completely monotonically increasing or decreasing—while the usual Pearson's correlation coefficient r captures its *linearity*—if $|r| = 1$ then the data points form a straight line—which we are not that interested in.

Surprisingly though, we find that $\rho \approx -0.687 < 0$, suggesting that a larger adjusted p-value correlates more to a smaller $P_0(z_j)$ value, and vice versa (see Figure 2). Performing a hypothesis test using `cor.test`, we get that this correlation value of ρ deviates from $\rho = 0$ with p-value $< 2.2 \cdot 10^{-16}$, meaning that the negative correlation is very much real.

If we calculate the correlation within each individual dataset—at least for datasets where neither correlation variable has zero variance—we see that most of the correlation values are positive (see Figure 3 for histogram). However, most also lie near 0, and the largest correlation value found is around 0.648, which, as is apparent in its scatter plot (see Figure 4), has far from a well-defined monotonic relationship between the two probabilities—there is still a decent number of variables with a low adjusted p-value but high $P_0(z_j)$ and vice versa.

That the correlation coefficient of the combined data differs greatly from that of the individual datasets can be explained as follows: The various datasets vary by a few factors,

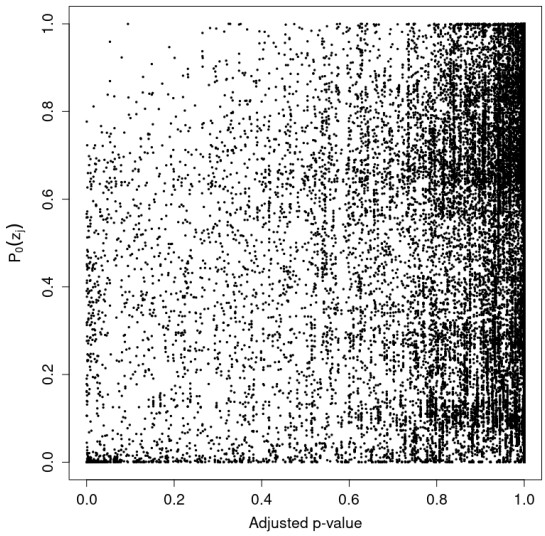

FIGURE 2: Scatter plot of the frequentist adjusted p-value and empirical Bayes $P_0(z_j)$ of all variables of all simulated datasets.

such as the number of subjects per group m and the number of variables n, and if the these factors affect the relationship between the adjusted p-value and $P_0(z_j)$, it would make combining their values across datasets not make much logical sense, and could—as in this case—obscure the correlation of the two probabilities independent of other variables. This is known as *Simpson's paradox*.

In conclusion, despite aiming to achieve the same thing, the two procedures—frequentist and empirical Bayes—are not equivalent; one cannot predict the output of one procedure well given the output of the other. Furthermore, there are additional confounding factors that further complicate the relationship between the two probabilities.

One possible explanation for the lack of congruence between the two procedures is in the difference in the choice of statistic. In our frequentist procedure, the Kolmogorov–Smirnov statistic is used, while in our empirical Bayes procedure, a modified two-sample Student's t test statistic is used, and the two are quite different. While these statistics are good choices for their respective procedures on their own, as we have seen, it makes directly comparing the two procedures difficult.

5.2 Comparison of error rates

Since we have shown that the two procedures are not equivalent, this begs the question: which one is "better"?

To answer that question, we first need to have a metric for the quality of a procedure. We would like for this metric to quantify how often the procedures categorize the variables as type 0 or type 1 "correctly". One problem is that, because of the design of the simulated data, no variable has exactly the same distribution between the two groups, so technically all variables are of type 1. Hence we instead look at the proportion of variables that are classified as type 1 as a function of the deviation d between the distribution of the two groups—we term this the type 1 classification rate curve. We would like to investigate how the type 1 classification rate curve differs between the two procedures.

To obtain type 1 classification rate curves, we use logistic regression, with the deviation values of variables classified as type 0 being the "failures" and that of variables classified as

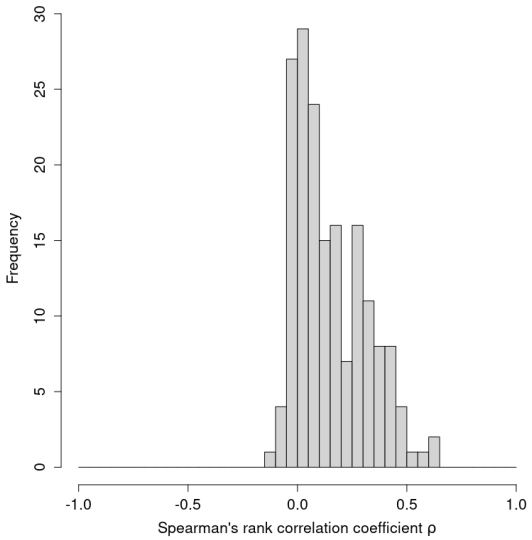

FIGURE 3: Histogram of the Spearman's rank correlation coefficient ρ between the frequentist adjusted p-value and the empirical Bayes $P_0(z_j)$ of the individual datasets of the simulated data.

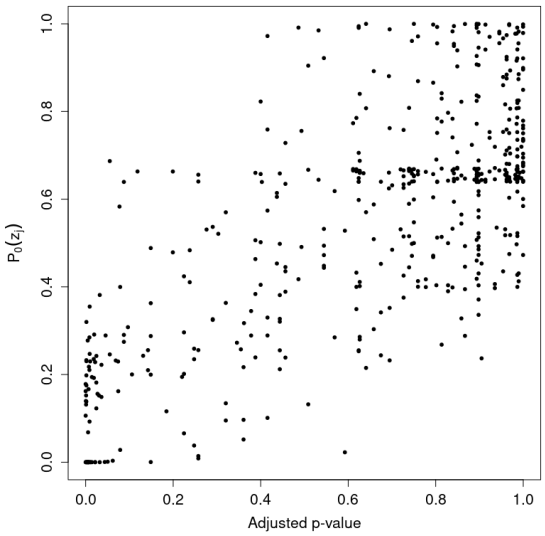

FIGURE 4: Scatter plot of the frequentist adjusted p-value and empirical Bayes $P_0(z_j)$ of the simulated dataset with $c = 4$, $a = 64$, $m = 80$, and $n = 500$. This is the simulated dataset with the largest Spearman's rank correlation coefficient between the two probabilities, with $\rho \approx 0.648$.

type 1 being the "successes". The curves should only increase, so to prevent overfitting, we constrain the class of regression functions to being the standard logistic function applied to degree 1 polynomials: $\mathcal{F} = \{f(d) = \sigma(c_0 + c_1 d) \,|\, c_0, c_1 \in \mathbb{R}\}$.

Varying the amount of data

There are five factors that we can vary—the deviation parameters c and a, the dimensions of the dataset m, n, and the significance level $\alpha = \beta$. We first look at how the type 1 classification rate curves vary with varying number of subjects per group m and number of variables n; we keep c and a constant with $c = 4$ and $a = 64$, their values that result in the most discernible differences between the two groups. The significance level is set to the orthodox value of $\alpha = \beta = 0.05$. The plots can be seen in Figure 5.

We observe a few things. First, the frequentist type 1 classification curves decrease with decreasing m. This makes sense, as with fewer subjects per group, it becomes easier to attribute the differences between the data of the two groups to chance, and so the (adjusted) p-values would be larger. On the other hand, the empirical Bayes curves do not decrease nearly as much with decreasing m as compared to the frequentist curves. This suggests that the empirical Bayes method is better at picking up on differences between the data of the two groups for small sample sizes—the frequentist procedure is quite conservative in comparison. It might be too good at picking them out though: at $m = 10$, for example, the empirical Bayes method classifies nearly all variables as type 1 when the differences are probably not significant enough to justify it.

Another observation is that as the number of variables n increases, the frequentist curve decreases. Note that, by design of the simulated data, as n increases, the variables

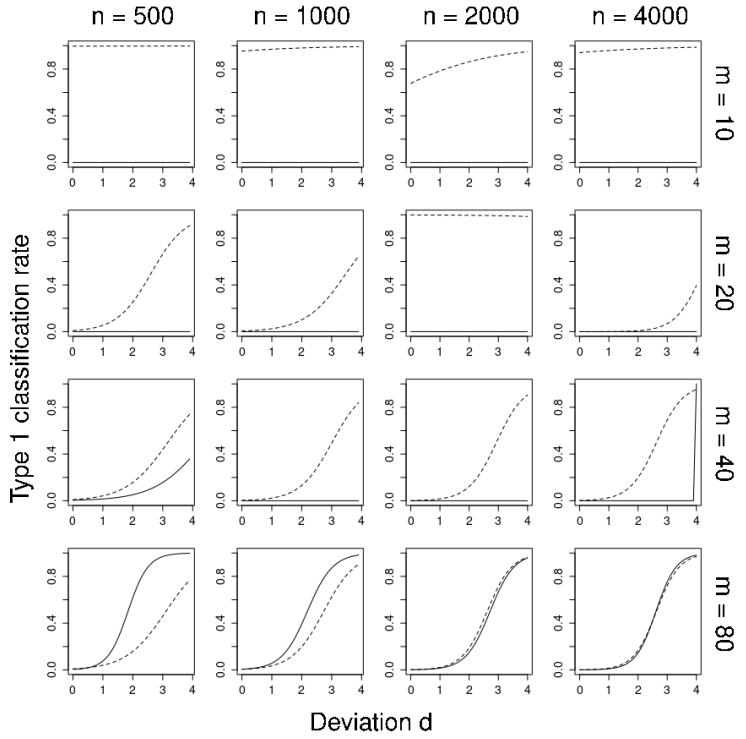

FIGURE 5: Type 1 classification rate curves (solid = frequentist, dashed = empirical Bayes) with varying number of subjects per group m and number of variables n. For all plots, $c = 4$, $a = 64$, and $\alpha = \beta = 0.05$.

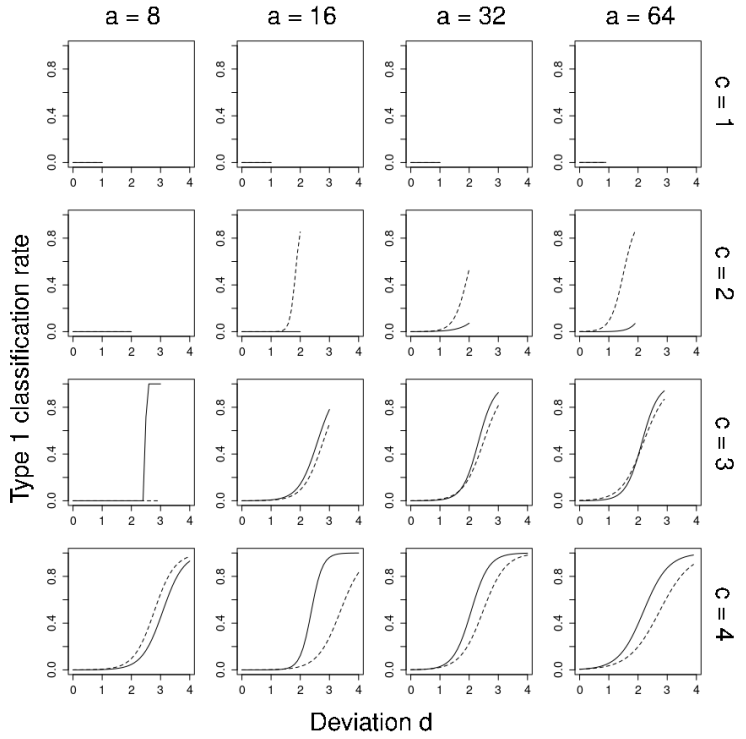

FIGURE 6: Type 1 classification rate curves (solid = frequentist, dashed = empirical Bayes) with varying c and a. For all plots, the number of subjects per group $m = 80$, the number of variables $n = 500$, and $\alpha = \beta = 0.05$.

that are added have almost no difference in distribution between the two groups. As for empirical Bayes, although it appears that its type 1 classification curves increase slightly as n increases, it is somewhat inconsistent; for $m = 20$, the opposite appears to be true. It is unclear why either should happen, as there is no explicit reference to the number of variables n in the empirical Bayes procedure.

Despite the differences, we see that the two procedures seem to produce similar type 1 classification curves for large m and n. This matches with our intuition that having more data leads to more accurate statistical results. However, at the same time, the correlation between the adjusted p-value and the empirical Bayes $P_0(z_j)$ is still unexpectedly low for large m and n: Spearman's rank correlation coefficient is $\rho \approx 0.380$ for the dataset with $c = 4$, $a = 64$, $m = 80$, $n = 4000$. If we look at the correlations between the classifications of the two procedures, we see (in Table 2) that there the correlation remains weak even if we only focus on the binary classifications at a significance level of $\alpha = \beta = 0.05$.

TABLE 2: Classification matrix of the frequentist and empirical Bayes procedures at a signifacnce level of $\alpha = \beta = 0.05$ on the dataset with parameters $c = 4$, $a = 64$, $m = 80$, and $n = 4000$.

		Empirical Bayes	
		Type 0	Type 1
Frequentist	Type 0	3951	17
	Type 1	14	18

Altogether, this suggests that the matching of the type 1 classification rate curves for large m and n as seen in Figure 5 may partially just be a coincidence.

Varying the data distributions

We now look at the type 1 classification rate curves with varying c and a (see Figure 6). As a reminder, c and a are parameters for the deviation d between the distributions of the two groups; the deviation of the jth variable, d_j, is given by

$$d_j = c \cdot e^{-(j-1)/a}. \tag{37}$$

As a result, c scales the deviation, with the largest deviation being c itself, and the proportion of variables with a large deviation is controlled by a—for all $d^* \in \mathbb{R}$, the number of variables with a deviation of at least d^* is (approximately) proportional to a. Similar to when varying m and n, we keep the other parameters constant: we choose $m = 80$ and $n = 2000$, and perform the categorization at a significance level of $\alpha = \beta = 0.05$.

Looking at the plots, we see that with decreasing c, the empirical Bayes type 1 classification rate curve appears to increase—or in other words, moves to the left. On the other hand, the frequentist curve appears to stay approximately the same with different values of c. This suggests that in the frequentist procedure, with the amount of data constant, the classification of each variable is uncorrelated with that of other variables, or at least, the correlation is lower than that of the empirical Bayes procedure. That also means that the empirical Bayes procedure is more sensitive to differences in distribution than the frequentist procedure if the differences in distribution are overall smaller. It is debatable whether this is good or bad. We would usually prefer a procedure that is more consistent, and if we would like to increase the amount of type 1 classifications, we could increase the cutoff level α to classify more variables as type 1. On the other hand, the frequentist

procedure tends to adjust the p-values too conservatively: the smallest adjusted p-values of the four datasets in the top row ($c = 1$) are (to 4 decimal places) 0.9999, 1, 0.5568, and 1 respectively. This would mean that even if we wanted to classify more variables as type 1, we couldn't if c is small enough.

Unlike c, the parameter a appears to not have much of an effect on either procedure—the curves of neither procedure clearly increase or decrease with varying a. From this, we can conclude that with the same amount of data and maximum difference in distributions, the proportion of variables with a large distribution difference has little effect on the statistical power of the procedure, and this holds for both procedures.

To conclude, it appears from our investigation that the empirical Bayes procedure performs better than the frequentist procedure at classifying variables with large deviations when the sample size m is small, though it does also not work when the sample size is too small (such as $m = 10$ here). The empirical Bayes procedure is also less consistent, both in the high "false positive" rate for small m, and in that its statistical power depends on the maximum difference in distributions. The frequentist procedure also has a drawback: it overclassifies variables with a small distribution difference as being of type 0 (having no difference), so much so that their adjusted p-values are often 1 or almost 1, making more liberal classification often difficult.

5.3 How empirical Bayes adjusts for multiplicity

One curious difference to note about the two procedures is that while the frequentist procedure explicitly controls for there being multiple comparisons, the empirical Bayes procedure does not. We do see that the empirical Bayes procedure tends to classify variables as type 1 more than the frequentist procedure—$P_0(z_j) < \widetilde{p}_j$ holds in 99.4% of all variables in all datasets—but more surprisingly, even when comparing $P_0(z_j)$ with the *unadjusted* p-value, the p-value is still usually larger—this happens in 81.2% of all variables. How, then, does the empirical Bayes procedure manage to limit the type 1 classification rate for variables with a small or even no deviation?

The answer is… it doesn't. The proportion of variables throughout the simulated data with a deviation less than 0.1 that are classified as type 1 at a significance level of 0.05 is 0.390 for the empirical Bayes method. In contrast, using the frequentist method, this proportion is $4.11 \cdot 10^{-5}$, and if we reject using the unadjusted p-value, the proportion is a reasonable 0.0319, still significantly less than that of the empirical Bayes method. However, this value can be deceiving, as it lumps together all datasets in the simulated data—from the type 1 classification curves in Figure 5, we have seen that the empirical Bayes method gives unreasonable results for small m. We calculate the same type 1 classification proportion but only for datasets with the same number of subjects per group m; the results can be found in Table 3.

Despite the questionable lumping together of multiple datasets, the empirical Bayes procedure has a high type 1 classification rate for variables with a small or no difference in distribution between groups; However, if as m increases, this rate decreases, dipping below the type 1 classification rate if classifying using the p-value for m large enough ($m = 80$ here). Nevertheless, this illustrates that the empirical Bayes procedure indeed does not control the type 1 classification rate—it does not aim to in the first place.

TABLE 3: The type 1 classification rate at significance level 0.05 of variables with deviation $d < 0.1$ using the empirical Bayes type 0 probability $P_0(z_j)$ and the unadjusted and adjusted p-values of the frequentist procedure.

	Type 1 classification rate for $d < 0.1$		
	$P_0(z_j)$	p-value	Adjusted p-value
$m = 10$	0.890	0.0226	0
$m = 20$	0.557	0.0304	$8.23 \cdot 10^{-6}$
$m = 40$	0.109	0.0357	$6.58 \cdot 10^{-5}$
$m = 80$	0.00350	0.0387	$9.05 \cdot 10^{-5}$
Overall	0.390	0.0319	$4.11 \cdot 10^{-5}$

6 Discussion

Based on our results, it appears that the empirical Bayes procedure is more statistically powerful (especially with fewer subjects per group, less than 80 but more than 20) and less sensitive to the number of variables, but is too eager to classify variables as type 1 when there are few subjects, and is affected by the number of variables with a big difference in distributions. The frequentist procedure, as it is designed to, controls the amount of false type 1 classifications, but at the cost of statistical power—it tends to classify variables as type 0 rather than as type 1. However, for large amounts of data (large m and n), the two procedures seem to produce similar classification rates, though the actual classifications do not match up well.

How can the procedures be improved? One major problem with the empirical Bayes method is that, when the number of subjects per group m is small, it overwhelmingly classifies variables as being of type 1, even those with a small difference in data between the two groups, as can be seen in Figure 5 and in Table 3. To investigate why this occurs, we plot the logistic regression curve of the empirical Bayesian procedure for the no-deviation dataset with $m = 10$, $n = 500$ (see Figure 7).

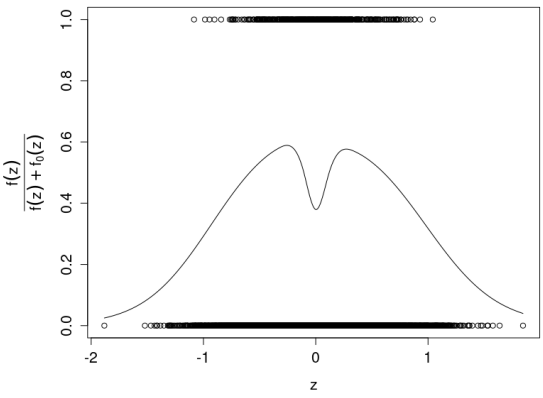

FIGURE 7: Plot of the empirical Bayes logistic regression curve for the no-deviation ($d = 0$ for all variables) dataset with $m = 10$ and $n = 500$. The points on the top are the mixed scores z_j, and the points at the bottom are the null scores $z_{0,j}$.

Looking at the plot, the problem seems to be that the type 0 scores (values plotted at the bottom) are more dispersed than the mixed scores (values plotted at the top).

This caused the regression curve to curve downwards when $|z|$ is large—this would have the interpretation that type 0 variables have a higher probability of producing a large difference between the data of the two groups, which is the opposite of what we intend.

One possible cause of this could be that too many batches of type 0 scores are produced—in this case, 20—causing the extreme values of the type 0 score to be more extreme than those of the mixed score, and hence the downwards curve of the regression curve. If this is the case, the problem can be alleviated by limiting the domain of the logistic regression to only the range of mixed score values—then, the type 0 scores will never extend past the range of mixed score values and bend the regression curve downwards. This cause could have also been compounded if the method of producing the type 0 scores tends to generate values that are more dispersed than the "actual" type 0 score distribution. More research will be needed to investigate the true cause of this phenomenon and to find a good solution to it.

As for the frequentist procedure, one weakness is its comparatively lower statistical power. While this is not a problem—a desired feature, in fact—if one wants to be sure that almost all variables classified as type 1 belong there, in some other use cases, for example if the goal of the classification is to classify almost all variables belonging to type 1 as type 1, such as in the microRNA problem, the lower statistical power may actually be a detriment. One solution to this is to forgo applying FDR control and instead classify based only on the p-values of the variables. This would resolve the problem of many adjusted p-values all being the same and/or having a value of 1, and allow the procedure to be more liberal at classifying variables as type 1. However, unlike adjusted p-values, the p-values of variables classified as type 1 do not have the same interpretation of being the probability that the variable is of type 0, as in Equation (35). Another approach could be to use another, more "liberal" multiple testing error than the FDR, and adjust the p-values according to a procedure that controls this error. Investigating this, however, is beyond the scope of this paper.

6.1 Analysis of microRNA data

Having understood the limitations of the two procedures, we now analyze the dataset we originally set out to analyze: the lung cancer microRNA dataset. The two groups consist of healthy subjects ($m_0 = 14$) and subjects with lung cancer ($m_1 = 16$) respectively. We restrict the list of microRNAs to only those with at least one non-zero microRNA count, as those with zero microRNA count for all subjects do not add anything to the analysis. This results in $n = 1421$ microRNAs in our analysis. Our goal is twofold: we would like to pinpoint microRNAs that are almost definitely correlated with lung cancer, and in addition, to obtain a larger set of microRNAs that have a tentative correlation to be considered for further research. We thus perform the procedures at significance levels of $\alpha = \beta = 0.05$ and 0.5, with the microRNAs classified as type 1 as the two sets respectively.

First, we perform the frequentist procedure on the dataset at a significance level of $\alpha = 0.05$: all microRNAs are classified as type 0. This is expected, as for example in Figure 5, the frequentist procedure classified all variables as type 0 in all $m = 10$ and $m = 20$ datasets. If we perform the empirical Bayes procedure, also at $\beta = 0.05$, we get that only one microRNA is classified as type 1: hsa.miR.10a.5p. Since the empirical Bayes procedure is known to sometimes classify variables as type 1 more than they deserve, we plot the regression curve to investigate whether this is the case here (see Figure 8).

We observe that the empirical Bayes regression curve bends up for large $|z|$, as intended, so the empirical Bayes result is likely unproblematic. Looking at the p-value of that microRNA, we see that it is quite small, at $p \approx 0.00417$, though the adjusted p-value

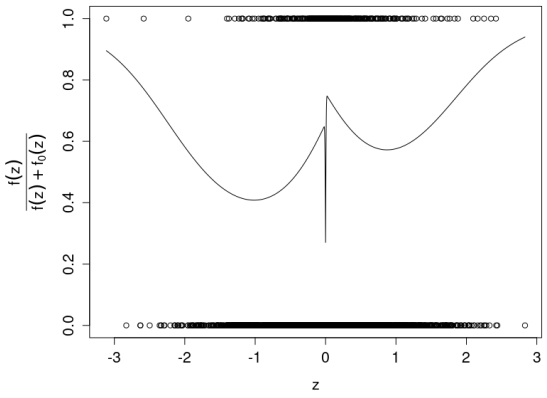

FIGURE 8: Plot of the empirical Bayes regression curve of the lung cancer microRNA dataset. The points on the top are the mixed scores z_j, and the points at the bottom are the null scores $z_{0,j}$.

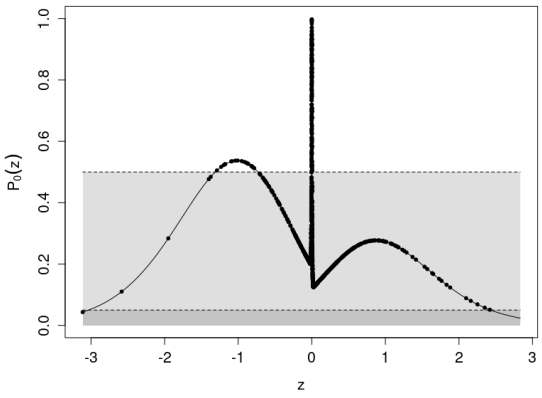

FIGURE 9: Plot of the type 0 probability curve $P_0(z)$, with variables plotted. Two classifications are also plotted—dark grey for $\beta = 0.05$, and either grey for $\beta = 0.5$—points in the shaded areas are classified as type 1.

is quite large, at $\widetilde{p} \approx 0.865$. Considering that the frequentist procedure would be too conservative in this scenario, we conclude that the microRNA hsa.miR.10a.5p is likely correlated with lung cancer.

We now perform the procedures at a significance level of $\alpha = \beta = 0.5$. The frequentist procedure classifies three microRNAs as type 1: hsa.miR.204.3p, hsa.miR.424.3p, and hsa.miR.509.3p. The empirical Bayes procedure, on the other hand, classifies 1067 microRNAs as type 1, which is clearly incorrect—looking at the $P_0(z)$ curve in Figure 9, we see that for most values of z, the corresponding variable would be classified as type 1, even those very close to $z = 0$. Hence we disregard the classification given by the empirical Bayes procedure, and conclude that the three microRNAs hsa.miR.204.3p, hsa.miR.424.3p, and hsa.miR.509.3p are candidates for microRNAs correlating with lung cancer.

The significance level of the classification can be changed to vary the ratio of microRNAs classified as the two types. It is also possible to simply sort the microRNAs by significance and present them as a list. The 20 microRNAs found to be most correlated with lung cancer for each procedure can be found in Appendix A.

Using a heterogeneous assortment of state-of-the-art classification algorithms, Lopez-Rincon et al. [8] found five microRNAs that are most informative for cancer classification: hsa-miR-378*, hsa-miR-221, hsa-miR-342-3p, hsa-miR-630, and hsa-miR-145; there is no overlap between the two sets of classified microRNAs. There is, however, the possibility that different types of cancer each have their own correlated microRNAs. In the literature reviewed by Galvão-Lima et al., breast, cervical, and prostate cancer do not share any known microRNA biomarkers [5]. In addition, two of our four microRNAs appear as microRNAs that only appear in either healthy people or lung cancer patients (but not both) in Chen et al. [3]. This thus suggests that it is inconclusive whether our set of four candidate microRNAs are truly biomarkers for lung cancer.

7 Conclusion

MicroRNAs have potential to be biomarkers for the presence of cancer, and for this to be possible, microRNAs correlating with cancer need to be identified from microRNA data. In this paper, we compared the frequentist and empirical Bayes procedures for this classification problem, the unpaired two-sample statistical inference problem. We found, in Section 5, that despite the two procedures having similar theoretical properties for the classification, the two procedures' results do not match up. We also found that the empirical Bayes procedure tends to classify variables more as type 1 than the frequentist procedure—however, it is also more inconsistent with its results.

Additionally, in Section 6.1, the two procedures were applied on a microRNA dataset to pick out microRNAs that are plausibly correlated with lung cancer: hsa.miR.10a.5p, hsa.miR.204.3p, hsa.miR.424.3p, and hsa.miR.509.3p. Further, more biomedically-oriented research could be done to investigate their biological functions and whether the correlations really exist.

The empirical Bayes procedure, despite its present shortcomings, can be promising. A natural next step would be to look into ways to fix its inconsistencies, such as the erroneous classifications made when the data only consist of type 0 variables, as described in Section 6. The current empirical Bayes procedure contains many details one can tweak, such as the function class used in logistic regression and the statistic, and the sensitivity analysis—the impact of the choice of these details on the output of the procedure—can be done to improve the procedure. Similarly, the effect of various tweaks to the frequentist procedure, such as the choice of statistic, can also be investigated.

Another path further research can take is to compare these two procedures with other two-sample statistical inference procedures. For example, a pure Bayesian approach to this problem could be implemented and a comparison with the two current procedures could be made. The computational time and/or complexity of these procedures would be an aspect of interest to investigate, as pure Bayesian procedures tend to have long computational times—would the extra time be worth the improvement (if any) it may have over a frequentist procedure? Beyond statistics, one could also make comparisons with non-statistical approaches to the problem, such as machine learning with neural networks and random forests, which have found success with microRNA classification problems [8].

References

[1] Yoav Benjamini and Yosef Hochberg. "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing." In: *Journal of the Royal Statistical Society. Series B (Methodological)* 57.1 (1995), pp. 289–300. ISSN: 00359246. URL: http://www.jstor.org/stable/2346101.

[2] Yoav Benjamini and Daniel Yekutieli. "The Control of the False Discovery Rate in Multiple Testing under Dependency." In: *The Annals of Statistics* 29.4 (2001), pp. 1165–1188. ISSN: 00905364. URL: http://www.jstor.org/stable/2674075 (visited on 06/28/2022).

[3] Xi Chen et al. "Characterization of microRNAs in serum: a novel class of biomarkers for diagnosis of cancer and other diseases." en. In: *Cell Res.* 18.10 (Oct. 2008), pp. 997–1006. DOI: 10.1038/cr.2008.282.

[4] Bradley Efron et al. "Empirical Bayes Analysis of a Microarray Experiment." In: *Journal of the American Statistical Association* 96.456 (2001), pp. 1151–1160. DOI: 10.1198/016214501753382129.

[5] Leonardo J Galvão-Lima et al. "miRNAs as biomarkers for early cancer detection and their application in the development of new diagnostic tools." en. In: *Biomed. Eng. Online* 20.1 (Feb. 2021), p. 21. DOI: 10.1186/s12938-021-00857-9.

[6] Ana Kozomara, Maria Birgaoanu, and Sam Griffiths-Jones. "miRBase: from microRNA sequences to function." In: *Nucleic Acids Research* 47.D1 (Nov. 2018), pp. D155–D162. ISSN: 0305-1048. DOI: 10.1093/nar/gky1141.

[7] Dena Leshkowitz et al. "Differences in microRNA detection levels are technology and sequence dependent." en. In: *RNA* 19.4 (Apr. 2013), pp. 527–538. DOI: 10.1261/rna.036475.112.

[8] Alejandro Lopez-Rincon et al. "Machine Learning-Based Ensemble Recursive Feature Selection of Circulating miRNAs for Cancer Tumor Classification." In: *Cancers* 12.7 (2020). ISSN: 2072-6694. DOI: 10.3390/cancers12071785. URL: https://www.mdpi.com/2072-6694/12/7/1785.

[9] Heidi Schwarzenbach, Dave S B Hoon, and Klaus Pantel. "Cell-free nucleic acids as biomarkers in cancer patients." en. In: *Nat. Rev. Cancer* 11.6 (June 2011), pp. 426–437. DOI: 10.1038/nrc3066.

A Most correlated microRNAs

TABLE 4: The 20 microRNAs most correlated with lung cancer as obtained from the lung cancer microRNA dataset using the two procedures. Probabilities are rounded to three significant figures. Bolded microRNAs appear on both lists.

(A) Frequentist procedure

microRNA	$\widetilde{p}$	p
hsa.miR.204.3p	0.226	0.000159
hsa.miR.424.3p	0.322	0.000653
hsa.miR.509.3p	0.322	0.000680
hsa.miR.574.3p	0.843	0.00237
hsa.miR.2110	0.865	0.00416
hsa.miR.10a.5p	0.865	0.00416
hsa.miR.365a.5p	0.865	0.00506
hsa.miR.888.5p	0.865	0.00506
hsa.miR.103a.3p	0.865	0.00548
hsa.miR.6758.5p	0.923	0.00650
hsa.miR.514a.3p	0.944	0.00731
hsa.miR.1285.5p	1	0.0112
hsa.miR.6124	1	0.0136
hsa.miR.6869.5p	1	0.0136
hsa.miR.125b.1.3p	1	0.0136
hsa.miR.320c	1	0.0138
hsa.miR.125a.5p	1	0.0172
hsa.miR.4298	1	0.0180
hsa.miR.3130.3p	1	0.0185
hsa.miR.1183	1	0.0208

(B) Empirical Bayes procedure

microRNA	$P_0(z)$
hsa.miR.10a.5p	0.0433
hsa.let.7b.5p	0.0510
hsa.let.7e.5p	0.0577
hsa.miR.30a.3p	0.0686
hsa.miR.423.5p	0.0800
hsa.miR.192.5p	0.0886
hsa.miR.125a.5p	0.110
hsa.miR.151a.5p	0.124
hsa.miR.5588.5p	0.125
hsa.miR.1193	0.125
hsa.miR.6131	0.125
hsa.miR.196a.5p	0.125
hsa.miR.6754.5p	0.125
hsa.miR.187.5p	0.125
hsa.miR.378g	0.125
hsa.miR.139.3p	0.125
hsa.miR.3147	0.125
hsa.miR.3138	0.125
hsa.miR.6512.5p	0.125
hsa.miR.1298.3p	0.125

B Implementation in R

B.1 Simulated data

```r
# 0 GENERATE SIMULATED DATA

# Simulate data given data parameters
simulate_data_block <- function(m, n, p, meanlog, sdlog) {
  # DISTRIBUTION NOT FINAL!
  # DISTRIBUTION: prob. 1-p of 0, prob. p of lognormal(meanlog, sdlog)
  # p, meanlog, sdlog are scalars, or vectors of size n (# of variables)
  # i.e. each obs. has the same underlying distribution
  data <- rbinom(m*n, 1, p)  # Initialize array of size n*m, prob. p of 1, prob. 1-p of 0
  data <- data * rlnorm(m * n, meanlog, sdlog)  # Prob. p of lognormal(meanlog, sdlog)

  data <- aperm(array(data, dim = c(n, m)))  # Reshape into m x n array (aperm() transposes)

  return(data)
}


calc_d_scores <- function(c, a, n) {
  return(c * exp(-(0:(n-1)) / a))
}


simulate_data_set <- function(c, a, m, n, group) {
  # Constants
  MEANLOG <- 0
  SDLOG <- 1
  P <- 1/3  # Probability of non-0

  # Initialize data array
  data <- array(0, dim = c(m, n))

  if (group == 0) {
    # Group 0
    data[,] <- simulate_data_block(m, n, P, MEANLOG, SDLOG)
  } else {
    # Group 1
    d_score <- calc_d_scores(c, a, n)
    random_sign <- 2 * rbinom(n, 1, 0.5) - 1
    data[,] <- simulate_data_block(m, n, P, MEANLOG + random_sign * d_score * SDLOG, SDLOG)
  }

  return(data)
}


simulate_data_set_list <- function(c, a, m, n) {
  result <- list(c = c, a = a, m = m, n = n)
  result$d_scores <- calc_d_scores(c, a, n)
  result$data_0 <- simulate_data_set(c, a, m, n, 0)
  result$data_1 <- simulate_data_set(c, a, m, n, 1)
  return(result)
}

```

```r

simulate_data_all <- function() {
  # All parameters
  # z-score between means = c*exp(-(j-1)/a)
  CS <- c(1, 2, 3, 4)  # z-score between means of most significant variable
  AS <- c(8, 16, 32, 64)  # The decay factor in the exponential;
                          # number of variables until z-score decays by e
  MS <- c(10, 20, 40, 80)  # Sample size per group, i.e. people per group
  NS <- c(500, 1000, 2000, 4000)  # Number of variables, i.e. miRNAs

  # Initialize data list
  data <- list()

  i <- 1
  for (m in MS) {
    for (n in NS) {
      # Null data set
      data[[i]] <- simulate_data_set_list(0, 1, m, n)
      i <- i + 1

      # Vary c and a
      for (c in CS) {
        for (a in AS) {
          data[[i]] <- simulate_data_set_list(c, a, m, n)
          i <- i + 1
        }
      }
    }
  }

  return(data)
}
```

B.2 Frequentist procedure

```r
# 1 FREQUENTIST APPROACH

# 1.1 Get p-values using Kolmogorov--Smirnov test (test for equality of
    distributions)
frequentist_p_value <- function(data_0, data_1) {
  n <- dim(data_0)[2]
  p_values <- array(1, dim = n)
  for (i in 1:n) {
    p_values[i] <- ks.test(data_0[,i], data_1[,i])$p.value
  }
  return(p_values)
}

# 1.2 Frequentist procedure: Add p-values and adjusted p-values to data set
procedure_frequentist <- function(data_set) {
  data_set$freq.p_values <- frequentist_p_value(data_set$data_0, data_set$data_
    1)
  data_set$freq.adj_p_values <- p.adjust(data_set$freq.p_values, method = "BH")
  return(data_set)
}
```

B.3 Empirical procedure

```r
# 2 EMPIRICAL BAYES APPROACH (EFRON (2001))
```

```r

# 2.1 Z STATISTIC

# Define pooled sd
pooled_sd <- function(x_0, x_1) {
  m_0 <- length(x_0)
  m_1 <- length(x_1)
  var_sum <- (m_0-1) * var(x_0) + (m_0-1) * var(x_1)
  return(sqrt(var_sum/(m_0+m_1-2)))
}

# Define Z statistic for a single variable
stat_Z <- function(x_0, x_1, s, a_0) {
  m_0 <- length(x_0)
  m_1 <- length(x_1)
  return((mean(x_1) - mean(x_0)) / (a_0 + s) / sqrt(1/m_0 + 1/m_1))
}

# Function to get Z statistic between two m x n arrays
stat_Z_array <- function(data_0, data_1) {
  n <- min(dim(data_0)[2], dim(data_1)[2])

  # Calculate pooled sd array
  s <- array(0, dim = n)
  for (i in 1:n) {
    s[i] <- pooled_sd(data_0[,i], data_1[,i])
  }

  # Calculate a_0
  a_0 <- quantile(s, probs = c(0.9))

  # Calculate Z
  result <- array(0, dim = n)
  for (i in 1:n) {
    result[i] <- stat_Z(data_0[,i], data_1[,i], s[i], a_0)
  }
  return(result)
}

sample_z_0 <- function(x_0, x_1) {
  m_0 <- length(x_0)
  m_1 <- length(x_1)

  # m_10 = number of values of x_1 to into x_0, and of x_0 to into x_1
  m_10 <- floor(m_10_raw) + rbinom(1, 1, m_10_raw - floor(m_10_raw))

  x_0_shuffled <- sample(x_0)
  x_1_shuffled <- sample(x_1)

  x_0_new <- c(x_0_shuffled[1:(m_0-m_10)], x_1_shuffled[1:m_10])
  x_1_new <- c(x_0_shuffled[(m_0-m_10+1):m_0], x_1_shuffled[(m_10+1):m_1])

  return(stat_Z(x_0_new, x_1_new))
}

sample_z_0_array <- function(data_0, data_1, k) {
  # k = number of iterations through the whole data set
  n <- min(dim(data_0)[2], dim(data_1)[2])

  m_0 <- dim(data_0)[1]
  m_1 <- dim(data_1)[1]
```

```r
  m_10_raw <- m_1 * m_0/(m_0+m_1)

  z_0_samples <- c()
  for (i in 1:k) {
    m_10 <- floor(m_10_raw) + rbinom(1, 1, m_10_raw - floor(m_10_raw))
    i_0 <- sample(1:m_0)
    i_1 <- sample(1:m_1)

    data_0_new <- rbind(data_0[i_0[1:(m_0-m_10)],], data_1[i_1[1:m_10],])
    data_1_new <- rbind(data_0[i_0[(m_0-m_10+1):m_0],], data_1[i_1[(m_10+1):m_
    1],])

    z_0_samples <- c(z_0_samples, stat_Z_array(data_0_new, data_1_new))
  }
  return(z_0_samples)
}



# 2.2 LOGISTIC REGRESSION

z_plot <- function(z_0, z) {
  x <- c(z_0, z)
  y <- c(array(0, dim = length(z_0)), array(1, dim = length(z)))
  return(data.frame(x = x, y = y))
}

# Wrapper around glm
get_logit_reg <- function(x, y, df, weights) {
  return(glm(y ~ ns(x, df = df), family = binomial, weights = weights))
}

# Generates and fits regression model f/(f + f_0)
get_domain <- function(zs, step) {
  domain <- seq(min(zs, na.rm = TRUE), max(zs, na.rm = TRUE), by = step)
  return(domain)
}

logit_reg_z <- function(z_0, z, prior_density) {
  # Set up prior
  PRIOR_BY <- 0.1  # Separation between prior data points
  if (prior_density == 0) {
    prior <- c()
    prior_weights <- c()
  } else {
    prior <- get_domain(c(z_0, z), PRIOR_BY)  # Prior data points
    prior_weights <- array(PRIOR_BY * prior_density, dim = length(prior))
  }

  frame <- z_plot(c(z_0, prior), c(z, prior))

  # Set weights of z_0 and z values,
  # so if z_0 and z have different lengths the model is still of f/(f + f_0)
  weights <- c(array(1/length(z_0), dim = length(z_0)), prior_weights,
               array(1/length(z), dim = length(z)), prior_weights)

  return(glm(y ~ ns(x, df = 6), family = binomial, data = frame, weights =
    weights))
}

# Function to evaluate a fitted logit model
```

```r
eval_reg <- function(reg, z) {
  return(plogis(predict(reg, data.frame(x = z))))
}

f_ratio_from_reg <- function(reg, z) {
  probs <- eval_reg(reg, z)
  return((1 - probs) / probs)
}

get_z_domain <- function(reg, step) {
  step_domain <- get_domain(reg$data$x, step)
  z_domain <- sort(c(step_domain, reg$data$x))
  return(z_domain)
}

# Function to estimate P(0|z) given zs and regression model of f/(f + f_0)
calc_p_0 <- function(reg) {
  z_domain <- get_z_domain(reg, 0.01)
  p_0 <- 1/max(f_ratio_from_reg(reg, z_domain))
  return(p_0)
}


# 2.3 Emp. Bayes procedure: Add z_0 (null), z (mixed), regression, and p_1 to data set
procedure_empirical_bayes <- function(data_set) {
  # print(c(data_set$k, data_set$r, data_set$m, data_set$n))
  m_0 <- dim(data_set$data_0)[1]
  m_1 <- dim(data_set$data_1)[1]
  n <- dim(data_set$data_0)[2]

  # Calculate z_0, z, and logit regression curve of f/(f + f_0)
  data_set$eb.z <- stat_Z_array(data_set$data_0, data_set$data_1)
  data_set$eb.z_0 <- sample_z_0_array(data_set$data_0, data_set$data_0, 20)
  data_set$eb.reg <- logit_reg_z(data_set$eb.z_0, data_set$eb.z, 0)
  data_set$eb.p_0 <- calc_p_0(data_set$eb.reg)

  # Calculate probabilities P(0|z)
  data_set$eb.p_0_var <- f_ratio_from_reg(data_set$eb.reg, data_set$eb.z) * data_set$eb.p_0

  return(data_set)
}



# 2.4 PLOTTING FUNCTIONS
plot_logit <- function(reg, z_0, z) {
  x_plot <- get_z_domain(reg, 0.01)
  par(mar = c(5, 6.5, 2, 2))
  plot(z_plot(z_0, z),
       xlab = TeX(r"($z$)"),
       ylab = TeX(r"($\frac{f(z)}{f(z) + f_0(z)}$)"))
  lines(x_plot, eval_reg(reg, x_plot))
  nice_par()
}

plot_empirical_bayes <- function(data_set) {
  plot_logit(data_set$eb.reg, data_set$eb.z_0, data_set$eb.z)
}

```

```r
plot_p_0_var <- function(data_set, betas) {
  z_domain <- get_z_domain(data_set$eb.reg, 0.01)
  p_0_var <- f_ratio_from_reg(data_set$eb.reg, z_domain) * data_set$eb.p_0

  plot(z_domain, p_0_var,
       ylim = c(0, 1),
       type = "l",
       xlab = TeX(r"($z$)"),
       ylab = TeX(r"($P_0(z)$)"))
  points(data_set$eb.z, data_set$eb.p_0_var)

  for (b in betas) {
    if (b >= 0) {
      lines(c(min(z_domain), max(z_domain)), c(b, b), lty = 2)
      rect(min(z_domain), 0, max(z_domain), b, col = "#00000020", border = NA)
    }
  }
}
```