Properties of machine learning and FDRs for discovery in large scale data

Megan Hollister Murray

Department of Biostatistics
Vanderbilt University

October 7th, 2020

1 Introduction

The focus of my oral exam will be on machine learning methods and false discovery rates. These two topics became of interest to me after reading a paper published in April 2018 Nature Methods Journal titled "Statistics versus machine learning" by Bzdok, Altman, and Brzywinski (Bzdok et al., 2018). The authors advocated for machine learning techniques for large-scale inference, as opposed to traditional statistical methods, which generated a great deal of discussion in the statistics community. I decided to replicate and explore their methods to determine for myself if the comparisons were fair or not. I presented my findings from this project at ENAR 2019.

During the coding and computation of these methods I discovered, the popular R function `stats::p.adjust` did not always return the desired values and did not correctly account for missing values. After researching the available options, I decided to create my own R package for false discovery rate (FDR) estimation. The package is now complete and Professor Jeffrey Blume and I have a corresponding paper that explains our methods and illustrates the package. The paper is in the process of being submitted to "The R Journal".

My orals will focus on the methodology used in the ENAR presentation and in the R package. Dr. Greevy has agreed that, in combination, these two documents can serve as my oral exam preparation. This document will provide a short introduction to these topics. Both the ENAR presentation slides and R Journal paper are included for you to review.

2 Machine learning and traditional methods for discovery in large-scale data

Professor Blume introduced me to the Nature Methods paper during the summer of 2018 and it caught my interest. The Nature Methods paper "Statistics versus machine learning" by Bzdok, Altman, and Brzywinski claims that the random forest algorithm tends to outperform traditional statistical adjustments for multiple comparisons in large-scale data (Bzdok et al., 2018). While their example focused mainly on gene expression data, they also hinted at broad claims that machine learning techniques may always outperform routine multiple comparison adjustments. Prof. Blume encouraged me to examine and compare the claims under a broader set of conditions and to use an unbiased method of comparison. We reported our findings in my 2019 ENAR presentation "An

evaluation of machine learning and traditional statistical methods for discovery in large-scale data", which was well attended.

We first explored how to use machine learning and traditional statistical methods in large-scale translational research. We focused on each type of method, how it would be used, and how accurate the procedure would be. We noticed that machine learning methods, when used in this context, required the user to state the expected number of dysregulated (non-null) genes a priori. This amounts to taking the top-ranking findings from each method, where the top number is defined a priori. Of course, this is a very different approach from traditional statistical methods, where the top number is not pre-specified, but some other statistical criterion, such as family-wise error, is controlled. We found that these procedures do not result in identical findings or interpretations. Therefore, to facilitate a fair comparison, we included the top-ranked findings for every method as a comparison criterion.

We estimated power and Type I Error rate for each procedure. We defined the power to be the proportion of truly dysregulated (non-null) genes identified as significant/dysregulated. The Type I Error rate is defined as the proportion of regulated (null) genes that are identified as significant or dysregulated. We then conducted a simulation study using the microarray gene expression data framework proposed in the Nature Methods paper. We maintained the original structure proposed by Bzdok, Altman, and Brzywinski. The structure for gene expression data includes a total of 40 genes from 20 people, in which 10 people are phenotype positive and 10 are phenotype negative. In order to find a statistical difference, 25% of the genes were set to be dysregulated across phenotype. The dysregulation forced the positive and negative phenotypes to have different mean population expressions. Additional variance was included to simulate genetic variation across the population. We also allowed for within-person correlation across genes, which was not included in the original simulations. See Algorithm 1 for the exact steps.

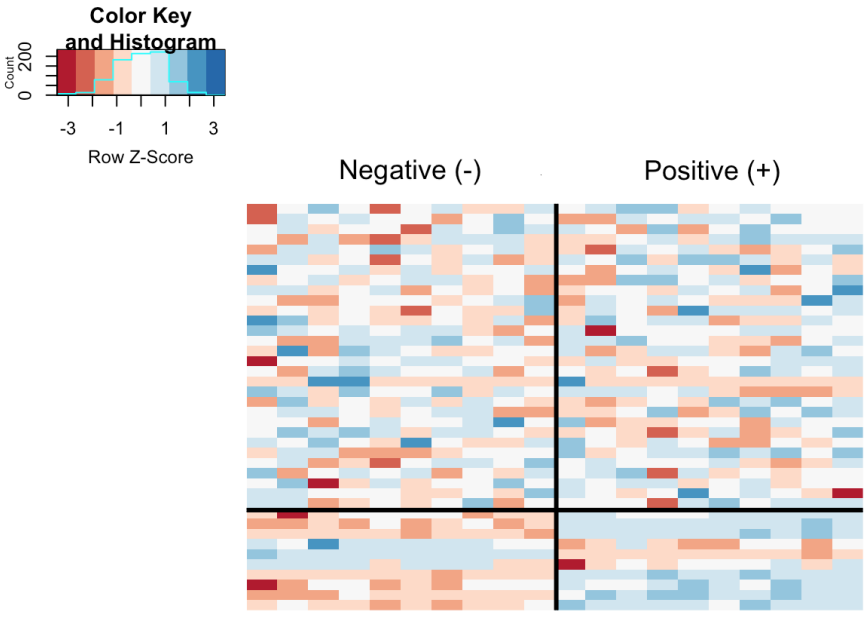

Figure 1: The simulated RNA-seq read counts for ten subjects in each phenotype generated from an over-dispersed Poisson distribution with biological variation.

Algorithm 1:

Result: Simulated RNA-seq counts

1. For all 40 genes simulate log mean expression levels from $\beta_0 \sim N(4, 2)$
   - For the positive(+) phenotype include the addition of a standard normal to each mean expression in the negative(-) phenotype $\beta_1 \sim N(0, 1)$
2. For each gene and person simulate the genetic variation across the population $\epsilon \sim N(0, 0.15)$
3. OPTIONAL: For each person simulate the within-person correlation across genes $\gamma \sim N(0, 1)$
4. Compute the mean gene expression by combining the above steps into $\lambda = \exp(\beta_0 + \beta_1 + \epsilon + \gamma)$
5. Generate the observed counts for each gene by sampling from a Poisson distribution. See Figure 1. Counts $\sim Pois(\lambda)$

We used the following methods to determine the number of dysregulated genes in a simulated data set: raw p-values, Benjamini-Hochberg empirical FDRs, Bonferroni adjusted p-values, second-generation p-values, random forest importance levels, and neural net prediction weights (Benjamini and Hochberg (1995); Bonferroni (1936); Liaw et al. (2002); Günther and Fritsch (2010)). Results varied depending on whether a pre-specified significance level was used or the top 10 ranked values were taken to be dysregulated. When all methods are given the same prior information that there are 10 dysregulated genes, the methods are almost identical in performance, with the Benjamini-Hochberg adjusted p-values and the second-generation p-values performing slightly better. Surprisingly the ranked raw p-values, the simplest method, had great performance. While adjusting for multiple comparisons does control the family-wise error rate or the false discovery rate, the rankings of the adjusted p-values vary only slightly from the ranking of the raw p-values. Ranked raw p-values require the least computation time and effort, which is desirable, of all methods considered, and they lose very little accuracy relative to the other methods. In the paper, Bzdok et. al. compared methods based on ranking to methods based on controlling a statistical criterion. This unfair comparison gives the (incorrect) impression that one method is better than another.

Machine learning methods did not yield improved statistical accuracy and they depended heavily on the a priori specified number of dysregulated genes. We were not able to validate the published finding that random forest importance levels from a machine learning algorithm outperformed classical methods. In our opinion, because their additional computation complexity, machine learning approaches do not appear preferable for identifying findings in the large-scale inference setting. The choice of an analysis method for large-scale translational data is critical to the success of any statistical investigation, and our simulations clearly highlight the various trade-offs among the available methods. It may be possible for machine learning methods to achieve the same tradeoffs as traditional statistical approaches for multiple testing, but it remains unclear what additional benefits they offer.

3 R Package: `FDRestimation`

During the course of reviewing the Nature Methods paper I noticed problems with the `stats::p.adjust` function, especially with its implementation of the Benjamini-Hochberg(BH) FDR procedure. To

provide a reliable tool for FDR estimation, I developed an R package that allows the user to implement their choice of FDR methods and decide on key assumptions for those algorithms. The new package, `FDRestimation`, includes 3 main functions: `p.fdr`, `plot.p.fdr`, and `get.pi0`. These functions compute the false discovery rates (FDRs), plot the computed values and their significance threshold(s), and estimate the null proportion, respectively. Inputs are included for six different multiple comparison adjustment methods for FDR estimation and FDR control; Benjamini-Hochberg, Benjamini-Yekutieli, Bonferroni, Holm, Hochberg, and Sidak (Benjamini and Hochberg (1995); Benjamini and Yekutieli (2001); Bonferroni (1936); Holm (1979); Hochberg (1988); Šidák (1967)).

In our paper, "False Discovery Rate Computation: Illustrations and Modifications", we explain the methodology and derive FDR adjustments. We illustrate the important difference between estimating the FDR for a particular finding and reporting the adjusted p-value that is needed to control the false discovery propensity at some level. The FDR and the adjusted p-value are sometimes, but not always, numerically identical and they are routinely confused in practice. This occurs often with the popular Benjamini-Hochberg (BH) algorithm, which is a "step-up" procedure that forces monotonicity and controls the group false discovery rate(Benjamini and Hochberg, 1995). These adjusted p-values are defined by Equation (1). The forced monotonicity is not part of the BH FDR estimates, as shown in Equation (2). In practice, we find these FDR estimates to be the most context useful when making scientific decisions about which interesting findings to pursue.

$$\tilde{p}_{(i)} \coloneqq \min_{j \geq i} \left(\frac{p_{(j)} m}{j} \right) \leq \gamma \tag{1}$$

$$FDR_i \coloneqq \frac{p_i m}{\text{rank}(p_i)} \cdot \hat{\pi}_0 \tag{2}$$

The proportion of truly null features (π_0) is an important component of the FDR estimate. While generally not identifiable, reasonable estimates of π_0 can be obtained under certain assumptions. The most common approach yields conservative estimates by setting $\pi_0 = 1$, which results in inflated FDR estimates. We propose a new method, "Last Histogram Height", for estimating the null proportion of findings. The "Last Histogram Height" method relies on the fact that under the null, a test statistic for a feature, say a Z-value, is standard normal. As such, the corresponding p-value has a uniform distribution over the unit interval. Therefore, if all the features were null, we would expect an empirical histogram of the observed p-values to be approximately flat. In Algorithm 2 we outline this null proportion estimation procedure given a vector of raw p-values.

Algorithm 2: Last Histogram Height Method

Result: Null proportion estimate

1. Plot a histogram of the raw p-values, $p_1, p_2, ...p_m$, with B number of bins, where $B < m$
   - The most consistent bin method is `scott`, according to our simulations
2. Store the histogram bin heights H_b for each bin $b = 1, 2, ..., B$
3. Call the height of last bin H_B the "null height"
4. Set the estimate of π_0 to be

$$\hat{\pi}_0 = \frac{H_B B}{m}$$

Below we show example of code for reproducing Figure 2. We simulated real data from 100 hypothesis tests and capture the 100 raw p-values. For context, 80 of these p-values were generated from a uniform distribution (and hence under the null) while the other 20 were generated from a skewed distribution representing the alternative. The raw p-values are displayed in Figure 2 as black points. The black sloped line is the BH rejection threshold, which is found in the BH derivation. Also included in the plot are the BH adjusted p-values (blue triangles), the BH FDR threshold for interesting findings (blue horizontal line), and the BH FDR estimates (red points). From the figure it should be clear that the feature-specific FDRs and the BH adjusted p-values have different values and interpretations. To conclude, we strongly encourage wider reporting of false discovery rates for observed findings and we believe this new tool will help researchers do just that.

My package can be installed using GitHub: https://github.com/murraymegan/FDRestimation

```
install.packages("devtools")
devtools::install_github("murraymegan/FDRestimation")

library(FDRestimation)
set.seed(88888)

pi0 <- 0.8
n <- 100
n.0 <- ceiling(n*pi0)
n.1 <- n-n.0

sim.data.p= c(runif(n.0),runif(n.1, min=0, max=0.01))

fdr.output = p.fdr(pvalues=sim.data.p, adjust.method="BH", threshold=0.05 )
plot(fdr.output)
```

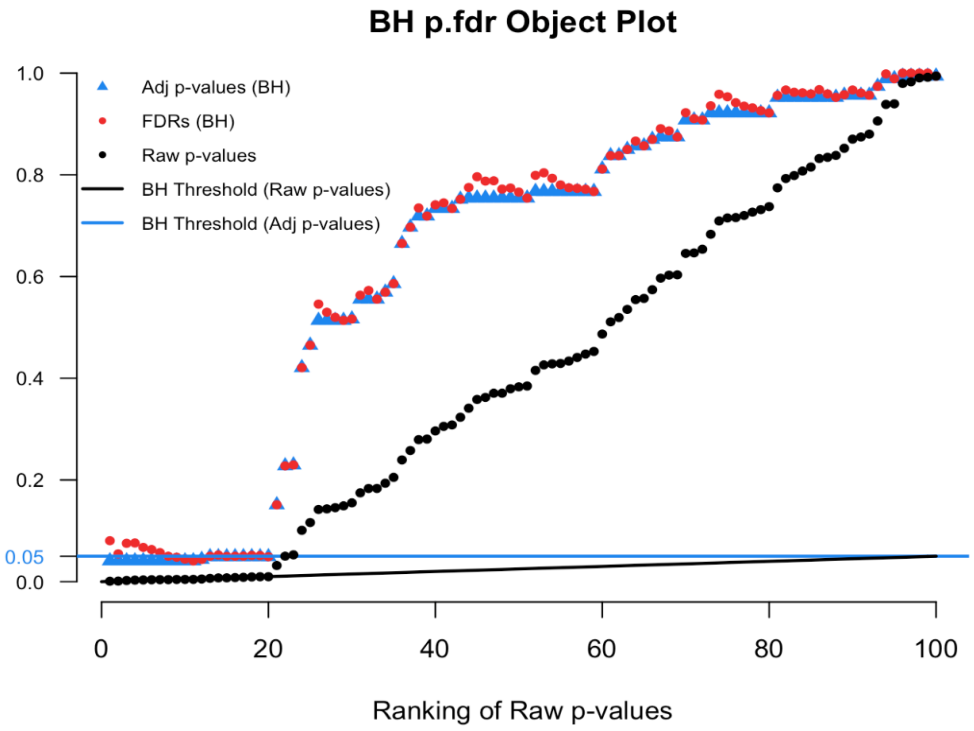

Figure 2: `plot.p.fdr` Example of 100 simulated p-values

4 Oral Exam

For my oral's presentation, I will review our methodology and our results from the machine learning project and I will introduce our false discovery rate package, `FDRestimation`, with an emphasis on why this tool is so useful.

References

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

Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annals of statistics*, pages 1165–1188, 2001.

C. Bonferroni. Teoria statistica delle classi e calcolo delle probabilita. *Pubblicazioni del R Istituto Superiore di Scienze Economiche e Commericiali di Firenze*, 8:3–62, 1936.

D. Bzdok, N. Altman, and M. Krzywinski. Points of significance: statistics versus machine learning, 2018.

F. Günther and S. Fritsch. neuralnet: Training of neural networks. *The R journal*, 2(1):30–38, 2010.

Y. Hochberg. A sharper bonferroni procedure for multiple tests of significance. *Biometrika*, 75(4): 800–802, 1988.

S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, pages 65–70, 1979.

A. Liaw, M. Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

Z. Šidák. Rectangular confidence regions for the means of multivariate normal distributions. *Journal of the American Statistical Association*, 62(318):626–633, 1967.