Controlling the FDR through multiple competition

Author: Kristen Emery
Supervisor: Dr. Uri Keich

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy in the School of Mathematics and Statistics

June 29, 2020
Author Attribution Statement

I, Kristen EMERY, declare that this thesis titled, “Controlling the FDR through multiple competition” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.

- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.

- Where I have consulted the published work of others, this is always clearly attributed.

- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.

- I have acknowledged all main sources of help.

- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

__________________________________________________________

Date:

__________________________________________________________
The concepts explored in chapters 2 and 4 of this thesis are published as Emery et al., 2019a and Emery et al., 2019b (as a later revision). These papers were jointly written with my supervisor and Professor Bill Nobel. I designed, coded and analyzed the simulations and constructed the methods that were applied to the real data as well as jointly developed the methods that are presented in that work.

The concepts explored in chapter 5 of this thesis are published as Emery and Keich, 2019. This paper was jointly written with my supervisor. I jointly developed the methods and assisted with designing, coding and analyzing the simulations.

In addition to the statements above, in cases where I am not the corresponding author of a published item, permission to include the published material has been granted by the corresponding author.

Signed:

Date:

As supervisor for the candidature upon which this thesis is based, I can confirm that the authorship attribution statements above are correct.

Signed:

Date:
Abstract

Faculty of Science
School of Mathematics and Statistics

Doctor of Philosophy

Controlling the FDR through multiple competition

by Kristen Emery

Multiple hypotheses testing is an important area of statistical research dedicated to the simultaneous analysis of multiple hypotheses. Error control in these testing scenarios is crucial to obtaining statistical rigour when deciding which of these hypotheses should be rejected. The most popular measure of this error is the false discovery rate (FDR). While there are many established procedures for obtaining guaranteed FDR control, most require the use of p-values which in some testing settings cannot be calculated. Competition-based FDR control provides a path to solving multiple hypotheses testing problems in these settings for which the canonical procedures fail.

As their name suggests, competition-based procedures use direct head-to-head competition between the originally observed score and a randomly generated null score for each hypothesis in order to control the FDR amongst the resulting list of discoveries. In this thesis we extend this competition framework and develop multiple testing procedures that utilize competition between the original scores and multiple, rather than a single, set of null scores.

We construct these methods in the frameworks of peptide identification through target-decoy competition and the classical linear regression problem with Barber and Candès’ recent knockoff procedure. In both these cases we show through simulations and real data experiments that utilizing multiple competition properly can lead to significant power gains without losing FDR control.
Acknowledgements

The completion of this work has without a doubt been the most challenging thing I have ever attempted. I had never envisioned myself here when I started my university journey and it is only through the help of others that I have made it this far. I give my thanks to the mathematics staff at the University of Sydney, and in particular all the lecturers, such as Dr. John Ormerod, who fostered my interest in statistics and lead me here.

My greatest thanks goes to my supervisor Associate Professor Uri Keich who consistently guided and assisted me throughout this thesis. It is only through Uri’s dedication, patience, leadership and knowledge that this work was completed. I could not have asked for a better supervisor.

I would also like to thank Professor Bill Noble of the University of Washington and fellow Sydney student Syamand Hasam who assisted with the work in this thesis. Both were crucial to the success of the peptide experiment. Bill provided and prepared the data and much appreciated expertise with background knowledge of the problem and downstream analysis, while Syamand analysed this data using our newly developed R functions, allowing us to obtain the results presented here.

Lastly, I would like to thank the University of Sydney and the Australian Government for giving me this opportunity and financing me throughout it, as well as my fellow PhD candidates who provided assistance of a less material (though no less important) manner.
Contents

Author Attribution Statement iii
Author Attribution Statement v
Abstract vii
Acknowledgements ix

1 Introduction 1
  1.1 Background - Multiple Testing ................................. 1
  1.2 Problem Motivation - Failings of the Canonical Procedures ....... 2
    1.2.1 The Benjamini-Hochberg and Storey FDR controlling procedures 4
    1.2.2 Using empirical p-values .............................. 6
      Non-pooled decoys .................................. 6
      Pooled decoys ...................................... 6
  1.3 Competition-based FDR control ................................ 9
    1.3.1 Peptide Identification and Target-Decoy Competition .......... 11
      Shotgun proteomics and spectrum identification .............. 11
      False discovery rate control in spectrum identification .... 12
      The peptide detection problem .......................... 13
    1.3.2 Knockoffs ........................................ 14

2 Multiple Decoy Competition 19
  2.1 Single Decoy TDC ......................................... 19
  2.2 Multiple Decoy Competition ................................ 22
    2.2.1 The mirror method .................................. 22
    2.2.2 The max method .................................... 26
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FWER</td>
<td>Family-wise Error Rate</td>
</tr>
<tr>
<td>FDR</td>
<td>False Discovery Rate</td>
</tr>
<tr>
<td>FDP</td>
<td>False Discovery Proportion</td>
</tr>
<tr>
<td>RV</td>
<td>Random Variable</td>
</tr>
<tr>
<td>K</td>
<td>Thousand</td>
</tr>
<tr>
<td>ID</td>
<td>Identification</td>
</tr>
<tr>
<td>PSM</td>
<td>Peptide-Spectrum Match</td>
</tr>
<tr>
<td>MS/MS</td>
<td>Tandem Mass-Spectrometry</td>
</tr>
<tr>
<td>TDC</td>
<td>Target-Decoy Competition</td>
</tr>
<tr>
<td>aTDC</td>
<td>averaged Target-Decoy Competition</td>
</tr>
<tr>
<td>SSS+</td>
<td>Sequential-SeqStep+</td>
</tr>
<tr>
<td>AS</td>
<td>Adaptive-SeqStep+</td>
</tr>
<tr>
<td>FDS</td>
<td>Finite-Decoy Storey</td>
</tr>
<tr>
<td>LBM</td>
<td>Labelled Bootstrap Monitored Maximisation</td>
</tr>
<tr>
<td>LSM</td>
<td>Lasso Signed Max</td>
</tr>
<tr>
<td>LCD</td>
<td>Lasso Coefficient Difference</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Background - Multiple Testing

Hypothesis testing is a critical tool that allows statisticians and researchers to make decisions with confidence by accounting for the natural variation that occurs in nature. These tests compare a null hypothesis against an alternative and either accept or reject this null based on a numerical score statistic that summarizes how well the data fits this null hypothesis. These tests are designed to minimize the error rates, both in falsely rejecting the null hypothesis (type I) and falsely retaining the null (type II). Type II error is generally considered in terms of power, the ability of the test to correctly reject the null hypothesis, and when developing novel testing procedures we seek to maximize the power while maintaining some measure of control over the type I error.

In many modern experiments we often seek to simultaneously perform multiple hypothesis tests, obtaining a list of rejected null hypotheses typically referred to as “discoveries”. As in the single testing case, we seek to maximize our power subject to some statistical control of our error rate. In the multiple testing framework this is considered as attempting to maximize the number of rejected null hypotheses while ensuring that the discovery list does not contain too many false discoveries (a hypothesis \( H_i \) is deemed a false discovery if it is rejected but is in fact truly null).

For a long time error control in multiple testing experiments was performed by controlling the family-wise error rate (FWER), which is defined as the probability that the discovery list contains at least one false discovery. More precisely, assume we have a selection procedure that returns a list of \( R \) discoveries of which, unbeknownst to us,
Chapter 1. Introduction

We define the FWER as $P(V \geq 1)$ and there are many well established procedures that attempt to control this value at some pre-determined level $\alpha$. The simplest and probably most well known of these is the Bonferroni correction in which each individual test is performed at a modified significance level of $\alpha/m$, where $m$ is the number of simultaneous hypotheses being tested. However, as modern experimental applications develop there are limitations of the FWER that make it unwieldy and impractical. In particular, as $m$ increases to a large number (many genealogy experiments simultaneously test tens of thousands or even more hypotheses at once) it is clear that the FWER will not scale well to this setting and in order to control this measure of the error with methods such as the Bonferroni correction we will have to make significant power sacrifices. For this reason many practitioners have turned to an alternative measure of the type-I error known as the false discovery rate (FDR) and look instead to control this measure.

Pioneered by Benjamini and Hochberg (Benjamini and Hochberg, 1995), methods that control the FDR seek to bound the expected proportion of false discoveries at some desired level $\alpha \in (0, 1)$. Specifically, let $Q = V/\max\{R, 1\}$ be the unobserved false discovery proportion (FDP) and, rather than explicitly controlling this FDP, we instead look to control the expected value of $Q$ at a pre-selected level $\alpha$: $E(Q) \leq \alpha$. In fact, it is this expectation $E(Q)$ that Benjamini and Hochberg define to be the FDR as they claim it is impossible to directly control the FDP. FDR responds much better than FWER when $m$ is large, allowing for the development of significantly more powerful testing procedures, and thus it is the form of error control that we consider here.

1.2 Problem Motivation - Failings of the Canonical Procedures

FDR control in multiple testing is a well researched problem and there are many established testing procedures that provide users guaranteed control of this error rate. For example, in the paper that introduced FDR as a concept Benjamini and Hochberg also constructed a hypothesis selection procedure to control it. Precisely, provided the true null p-values are distributed as independent uniform $U(0, 1)$ random variables, their
method ensures $E(Q) \leq \alpha$ for any p-values the true alternative / false null hypotheses assume. Other, more powerful selection procedures that rely on estimating $\pi_0$, the fraction of true null hypotheses, are also available. Generally referred to as “adaptive BH” procedures, with one particularly popular variant by Storey (e.g. Benjamini and Hochberg, 2000; Benjamini, Krieger, and Yekutieli, 2006; Storey, 2002; Storey, Taylor, and Siegmund, 2004), these procedures can also achieve FDR control using a set of p-values corresponding to each of the hypotheses.

The situation we consider here is controlling the FDR when we cannot assign p-values to the hypotheses. Specifically, as normal we assume that we can compute a test statistic $Z_i$ for each hypothesis $H_i$, so that the larger $Z_i$ is, the less likely is the null. However, departing from the standard setup, we further assume that we cannot compute p-values for the observed scores. Instead, we can only generate a small sample of size $d$ of independent competing null scores for each hypothesis $H_i$: $\tilde{Z}_{ij} = 1, \ldots, d$. As elaborated below, we refer to $\tilde{Z}_{ij}$ as decoy, or alternatively knockoff scores, and we would like to utilize these to control the FDR.

This is not just an interesting theoretical problem, but in fact occurs quite frequently in real experiments. For example, this issue comes up when considering peptide identification in the field of computational mass spectrometry and we observe a closely related situation (the competing null scores in this case are not quite independent) in the variable selection problem of linear regression, settings that we will examine in more detail throughout this work. Other examples of our problem include analysing a large number of motifs reported by a motif finder (e.g., Harbison et al., 2004), where creating competing null scores can require the time consuming task of running the finder on randomized versions of the input sets (e.g., Ng and Keich, 2008), as well as controlling the FDR in selecting differentially expressed genes in microarray experiments where a small number of permutations is used to generate competing null scores (Tusher, Tibshirani, and Chu, 2001). In all these examples (barring the variable selection problem) it is conceptually simple, though computationally expensive, to generate multiple decoys.

Perhaps the most intuitive method of solving this type of problem is to use the
decoy scores to estimate an empirical null distribution for each hypothesis $H_i$ and subsequently construct empirical p-values. Then we can apply the canonical FDR controlling procedures as usual. While this works well assuming that $d$ is sufficiently large, as mentioned above, generating multiple decoy scores can be computationally intensive which often results in only being able to obtain a small $d$. As we will see this can cause significant issues with regards to both FDR control and power when trying to utilize empirical p-values, especially if the scores corresponding to each of the hypotheses are not calibrated (null scores of different hypotheses have different distributions).

1.2.1 The Benjamini-Hochberg and Storey FDR controlling procedures

We first motivate the need for new methods by examining closer the issues with the canonical FDR controlling procedures in our setting. In particular, we look at Benjamini and Hochberg’s original method (the BH procedure) and Storey’s adaptive extension and consider the issues that occur when dealing with empirical p-values in these methods.

The BH procedure was originally proposed by Benjamini and Hochberg in their seminal paper (Benjamini and Hochberg, 1995) as a simple procedure to rigorously control the FDR. This procedure determines the discovery list from a set of p-values by examining the whole list, finding the largest p-value that satisfies an index-dependent condition and rejecting all hypotheses corresponding to p-values that are less than or equal to this value. The exact procedure is as follows:

Consider the p-values $p_1, \ldots, p_m$ corresponding to hypotheses $H_1, \ldots H_m$,

1. Order the p-values $p(1) \leq p(2) \leq \cdots \leq p(m)$.

2. Calculate $\hat{k} = \max\{1 \leq k \leq m : p(k) \leq \alpha k/m\}$

3. If $\hat{k}$ exists reject all $H_i$ corresponding to p-values less than or equal to $p(\hat{k})$, or $p(1) \leq p(2) \leq \cdots \leq p(\hat{k})$. Otherwise reject nothing.

Since its conception there have been many improvements to the BH method, both to extend its applicability and improve its power (e.g. Benjamini and Yekutieli, 2001). One of the most popular of these is Storey’s adaptive BH method (Storey, 2002 and Storey, Taylor, and Siegmund, 2004) which calculates the value $\hat{\pi}_0$, an estimate of $\pi_0$.
the proportion of true nulls, and subsequently performs BH but replacing $\alpha$ with $\alpha/\hat{\pi}_0$. To estimate this value, Storey examines the ratio of p-values larger than a pre-chosen value $\lambda$ to the expected number we should see if all $m$ hypotheses were truly null. Ideally $\lambda$ is chosen such that we can be reasonably sure that all $p_i > \lambda$ correspond to truly null hypotheses. The exact formula Storey uses is:

$$\hat{\pi}_0(\lambda) = \frac{m - R(\lambda)}{(1 - \lambda)m}, \quad (1.1)$$

where $m$ is again the number of hypotheses, and $R(\lambda)$ is the number of discoveries at the threshold $\lambda$ (the number of hypotheses whose p-value is $\leq \lambda$). A $+1$ is added in the numerator if strict guaranteed FDR control is desired (rather than asymptotic).

$$\hat{\pi}_0^*(\lambda) = \frac{m - R(\lambda) + 1}{(1 - \lambda)m}, \quad (1.2)$$

In addition to his adaptive procedure Storey also introduced an alternative framework to working with FDR control in multiple testing problems. Benjamini and Hochberg developed their method by considering a fixed FDR level and calculating a variable $\hat{k}$, or p-value threshold. Storey reverses this framework and states that we can fix the threshold for some value of $t$, $0 \leq t \leq 1$ and estimate the FDR if we were to reject all $p_i \leq t$. Storey showed that we can perform the equivalent of BH in this framework by:

1. Order the p-values $p(1) \leq p(2) \leq \cdots \leq p(m)$.

2. For each $t \in (0, 1)$ calculate

$$\hat{\text{FDR}}(t) = \frac{m \cdot t}{R(t) \lor 1}. \quad (1.3)$$

3. Select $t_\alpha \left( \hat{\text{FDR}} \right) = \sup \left\{ t \in [0, 1] : \hat{\text{FDR}} \leq \alpha \right\}$ and reject all hypotheses corresponding to p-values less than or equal to $t_\alpha \left( \hat{\text{FDR}} \right)$.

We can construct Storey’s adaptive BH procedure by constructing the aforementioned estimate of $\pi_0$ and replacing (1.3) with,

$$\hat{\text{FDR}}_{\lambda}(t) = \frac{m \cdot \hat{\pi}_0(\lambda) \cdot t}{R(t) \lor 1},$$
or using $\pi_0^*$ and searching $t \leq \lambda$ if we want strict FDR control.

### 1.2.2 Using empirical p-values

As stated earlier, despite not having exact p-values we are still able to apply these methods in our setting by constructing empirical p-values. We can develop these empirical p-values by noting the rank of the originally observed score ("original score"

for short) $Z_i$, when combined with the decoy scores $\{\tilde{Z}_{ij}\}_{j=1}^d$. This can be done either by pooling all the decoy scores together and comparing the original score to $\bigcup_i \{\tilde{Z}_{ij} : i = 1, \ldots, m, j = 1, \ldots, d\}$, or keeping them separate for each hypothesis (non-pooled). However, regardless of the method that is selected there can be issues with these empirical p-values that make applying both the BH and Storey’s methods unreliable.

#### Non-pooled decoys

In the non-pooled version the empirical p-values take values of the form $i/d_1$, where $d_1 = d + 1$, and $i \in \{1, \ldots, d_1\}$ is the rank of the original score $Z_i$ in the combined list of $d_1$ scores: $\left(Z_i, \tilde{Z}_{i1}, \ldots, \tilde{Z}_{id}\right)$. Using these non-pooled p-values the BH procedure rigorously controls the FDR, and Storey’s method will asymptotically control the FDR as $m \to \infty$. However, because the p-values are rather coarse both methods can be extremely weak, especially when $d$ is small. For example, if $d = 1$, each empirical p-value is either 1/2 or 1, and therefore for many practical examples both methods will not be able to make any discoveries at usable FDR thresholds.

#### Pooled decoys

When pooling all the decoys together our empirical p-values attain values in $i/(m \cdot d + 1)$ for $i = 1, \ldots, md + 1$, and hence, particularly when $m$ is large, the p-values generally no longer suffer from being too coarse. Unfortunately however, other significant issues arise when pooling the decoys. First, the empirical p-values computed using the pooled decoys do not satisfy the assumption that the p-values of the true null hypotheses are independent: because all p-values are computed using the
same batch of pooled decoy scores, it is clear that they are dependent to some extent. While this dependency diminishes as \( m \to \infty \), there is a more serious problem that in general cannot be alleviated by considering a large enough \( m \).

In pooling the decoys we make the implicit assumption that the score is calibrated, i.e., that all true null scores are generated according to the same distribution. If this assumption is violated, as is typically the case in the computational mass spectrometry problem we will examine later for one (Keich and Noble, 2015), then the p-values of the true null hypotheses are not identically distributed, and in particular they are also not (discrete) uniform in general. This means that even the more conservative BH procedure is no longer guaranteed to control the FDR.

For example, consider just \( m = 2 \) hypotheses with \( d = 1 \) decoy, and suppose that \( P \left( Z_1^1 > \tilde{Z}_1^1 \right) = 1 \) (i.e., the support of the null distribution corresponding to \( H_1 \) is disjoint and to the right of the support of the null distribution corresponding to \( H_2 \)). Suppose further that both \( H_i \) are true nulls, so that the FDR coincides with the FWER. It is easy to see that in this case using the FDR threshold \( \alpha := 2/3 \) the event \( \{ Z_1 > \tilde{Z}_1^1, Z_2 < \tilde{Z}_2^1 \} \) will produce one discovery \( (p_1 := \text{p-value}(Z_1) = 1/3, p_2 := \text{p-value}(Z_2) = 1) \), and the disjoint event \( \{ Z_2 > \tilde{Z}_2^1 \} \) will produce two discoveries \( (p_2 = 2/3) \). However, these events have a total probability of \( 1/4 + 1/2 = 3/4 \) so the FWER=FDR is \( > \alpha \) in this case.

This effect can be much more pronounced in the case of Storey’s method. Suppose now that the null hypotheses are split into two equal sized groups, \( A \) and \( B \), where for every \( i \in A \) and \( j \in B \), \( P \left( Z_1^1 < \tilde{Z}_1^j \right) = 1 \). Suppose further that all the hypotheses in \( A \) are false nulls with scores \( Z_i \) satisfying \( P \left( Z_i > \max_{j \in A} \tilde{Z}_j^1 \right) = 1 \), and that all hypotheses in \( B \) are true nulls. The decoy-pooled p-values will be essentially no greater than \( 1/2 \). Hence, Storey’s estimate of \( \pi_0, \hat{\pi}_0(\lambda) = \frac{m-R(\lambda)}{(1-\lambda)m} \), where \( R(\lambda) \) is the number of hypotheses whose p-value is \( \leq \lambda \), will significantly underestimate \( \pi_0 \) (even if we add the +1 finite correction). For example, if \( \lambda \geq 0.5 \) then \( \hat{\pi}_0 = 0 \), which in turn implies that essentially all null hypotheses will be rejected at any FDR level \( \alpha \) and particularly for \( \alpha < 1/2 \), while the actual FDP will clearly be \( 1/2 \). Even if \( \lambda \) is chosen to better fit these p-values, e.g., \( \lambda = 0.25 \), or the set-up is changed slightly to allow some group \( A \) p-values to be null so \( \hat{\pi}_0 \neq 0 \), the procedure will still significantly underestimate \( \pi_0 \) and
thus underestimate the actual FDR.

**Example 1.** As a specific example in the above vein we constructed an experiment with $m = 300$ and $d = 5$ decoys where group $A$’s true null distribution is $N(0, 1)$, and group $B$’s true null distribution is $N(50, 1)$. We set all 150 hypotheses in group $B$ and 50 of the 150 hypotheses in group $A$ to be true null, and we generated observed scores by sampling from the appropriate null distribution above. We next generated the observed scores for the 100 false null hypotheses in group $A$ by sampling from the same, significantly shifted, $N(50, 1)$ distribution that we used to generate all observed scores of group $B$. All competing null (decoy) scores were generated using the group’s null distribution. In this setup we chose to leave a third of group $A$ as true nulls so that approximately 50 of the p-values will exceed $1/2$ ensuring that $\pi_0 > 0$.

We then computed the pooled p-values and applied Storey’s FDR controlling procedure, as presented in the package *qvalue* (Storey et al., 2019) (with $\lambda$ chosen using the bootstrap option). This experiment was repeated using 1,000 randomly drawn sets, noting each time the real FDP at FDR thresholds of $\alpha = 0.1$ and $\alpha = 0.2$. As expected in this setting, Storey’s procedure clearly failed to control the FDR: at $\alpha = 0.1$, the empirical FDR (the FDP averaged over the 1K samples) was 0.24, or over 200% of the desired significance level, and for $\alpha = 0.2$ the empirical FDR was larger than 0.5 again indicating a significant violation.

We could not find such examples, with an essentially arbitrary large $m$ and a substantial liberal bias, when using the BH procedure. However, we found a class of arbitrary large examples, similar to the above class (on which Storey similarly fails to control the FDR), where due to pooling the conservative nature of BH was amplified to the point where it was essentially powerless. Consider four groups $A$, $B$, $C$ and $D$ and suppose that for every $i \in A$, $j \in B$, $k \in C$ and $l \in D$, $P\left(\tilde{Z}_i^1 < \tilde{Z}_j^1 < \tilde{Z}_k^1 < \tilde{Z}_l^1\right) = 1$. Suppose further that all the hypotheses in groups $A$ and $B$ are false null with scores that fall in the range of values of the subsequent group, so in particular $P\left(Z_i > \tilde{Z}_k^1\right) = 0$ and similarly $P\left(Z_j > \tilde{Z}_l^1\right) = 0$. It is easy to see that using pooling in this case the p-values for the (false) null hypotheses in groups $A$ and $B$ will be $\geq 1/2$ and $1/4$ respectively, and it follows that no discoveries can be made by BH with $\alpha < 1/4$, regardless of how large $m$ and $d$ are.


Example 2. Again, we construct a specific example according to the above general outline. We set $m = 300$, so that each of the four groups has 75 hypotheses, and we use $d = 5$ decoys. The null distribution of each group is set as $\mathcal{N}(\mu, 1)$, where $\mu$ increases from $\mu_A = 0$, by 50, to $\mu_D = 150$. The observed scores corresponding to the 150 false null hypotheses of groups $A$ and $B$ were drawn from the null distributions of group $B$ and $C$ respectively, whereas the 150 observed scores of groups $C$ and $D$ were drawn from their respective null distributions. Using pooled p-values BH does not yield any discovery for any $\alpha \leq 0.65$ amongst any of our 1000 samples, and it was not until using $\alpha = 0.7$ that we finally started seeing some samples on which BH had non-zero power. Incidentally, even using non-pooled p-values is slightly better here: the first samples with non-zero BH power appear for $\alpha = 0.3$.

The above examples, which we will return to in later chapters, demonstrate that, in general, applying BH or Storey’s procedure to p-values that are estimated by pooling the competing null scores can be problematic both in terms of power, and control of the FDR. Moreover, these issues were discussed in the context of the spectrum-ID problem, where the effect of pooling on power, and on FDR control were demonstrated using both simulated and real data as can be seen in Keich and Noble, 2015 and Keich, Kertesz-Farkas, and Noble, 2015.

1.3 Competition-based FDR control

It is clear from our examination into empirical p-values and the issues that come with them that, especially when $d$ is small, we require alternative methods to obtain desirable results. The class of alternatives methods that we consider here are competition-based FDR controlling procedures that utilize direct competition between the original and decoy scores to control the FDR instead of constructing p-values.

Competition-based procedures have seen significant application in the field of computational mass spectrometry, where a procedure called target-decoy competition (TDC) (Elias and Gygi, 2007; Cerqueira et al., 2010; Jeong, Kim, and Bandeira, 2012; Elias and Gygi, 2010) has been used to solve the spectrum identification (ID) problem and the closely related peptide ID problem for over a decade. Another example of this type
of testing procedure is Barber and Candés’ knockoff procedure for variable selection and in fact, given a set of decoy or knockoff scores, these procedures proceed by an identical framework and we can use the decoy/knockoff terminology interchangeably when working with this class of methods. In this work we will use the knockoff terminology (in which we refer to the competing null scores as “knockoffs”) only when explicitly working with the variable selection case, while using the target-decoy terminology (where the “target” corresponds to the original scores and “decoys” to the competing null scores) in the simpler independent decoy cases, as there is extra difficulty in generating knockoff scores in the variable selection problem.

Both these procedures consider our problem when \( d = 1 \) and avoid the use of empirical p-values. Each decoy/knockoff score \( \tilde{Z}_i \) directly competes with its corresponding original score \( Z_i \) to determine the reported list of discoveries. Specifically, for each score threshold \( T \) we would only report the target observations that won their competition: \( Z_i > \max\{T, \tilde{Z}_i\} \). Additionally the number of decoy wins (\( \tilde{Z}_i > \max\{T, Z_i\} \)) is used to estimate the number of false discoveries in the list of target wins. Thus, the ratio between this estimate and the number of target wins yields an estimate of the FDR among the target wins. To control the FDR at level \( \alpha \) we choose the smallest score threshold \( T = T(\alpha) \) for which the estimated FDR is still \( \leq \alpha \).

Both TDC and the knockoff procedure were independently proven to control the FDR assuming that the truly null hypotheses are equally likely to be a target or decoy win, and provided that we add 1 to the number of decoy wins before calculating the FDR estimate (He et al., 2015; Levitsky et al., 2017; Barber and Candès, 2015). However these procedures can be limiting as currently they are only constructed to allow for one-on-one competition, or \( d = 1 \).

While there have been recent developments that allow the utilization of extra decoy scores these have primarily still used pairwise competition. For example, averaged Target-Decoy Competition (aTDC) as presented in Keich and Noble, 2017b and Keich, Tamura, and Noble, 2018 performs single decoy competition between the target score and each of \( d \) decoy scores individually for each hypothesis before averaging the results across the decoys, a procedure that allows Keich and Noble to decrease the variability inherent in TDC. The primary goal of this work is to develop a new competition
framework that explicitly extends the pairwise competition of TDC/knockoffs to work with $d > 1$ decoys. We will show that using this multiple competition procedure we can improve the power of the current procedures without sacrificing FDR control. In particular we will focus on both the peptide ID problem of mass spectrometry and the variable selection problem outlined in Barber and Candès, 2015.

1.3.1 Peptide Identification and Target-Decoy Competition

Tandem mass spectrometry (MS/MS) currently provides the most efficient means of studying proteins, the primary functional molecules in living cells, in a high-throughput fashion. Knowledge of these proteins for a cellular population provides insight into the functional state of the cells, allowing us to characterize cell types, differentiation states, disease states, or species-specific differences. MS/MS does not measure proteins directly however as they are difficult to separate and manipulate biochemically. Instead an MS/MS experiment first involves digesting proteins into smaller pieces called peptides and measuring these directly. An MS/MS experiment generates spectra (in the order of 18,000 per typical MS/MS experiment) as observations and canonically each observed spectrum is generated by a single peptide. Thus the first goal of the downstream analysis is to infer which peptide was responsible for generating each observed spectrum. The resulting set of detected peptides can then be used to determine what proteins are present in the sample. The hypothesis testing situation we are interested in arises in the peptide detection problem, specifically, the task of assigning (FDR) confidence estimates to peptides that have been identified by MS/MS.

Shotgun proteomics and spectrum identification

In a “shotgun proteomics” MS/MS experiment, proteins in a complex biological sample are extracted and digested into peptides, each with an associated charge. These charged peptides, called “precursors,” are measured by the mass spectrometer, and a subset of the precursors are then selected for further fragmentation into charged ions,
which are detected and recorded by a second round of mass spectrometry (P. Hernández and Appel, 2006; Noble and MacCoss, 2012). The recorded tandem fragmentation spectra, or spectra for short, are then subjected to computational analysis.

This analysis typically begins with the spectrum identification problem, which involves inferring which peptide was responsible for generating each observed fragmentation spectrum. The most common solution to this problem is peptide database search. Pioneered by SEQUEST (Eng, McCormack, and Yates, III, 1994), the search engine extracts from the peptide database all “candidate peptides,” defined by having their mass lie within a pre-specified tolerance of the measured precursor mass. The quality of the match between each one of these candidate peptides and the observed fragmentation spectrum is then evaluated using a score function. Finally, the optimal peptide-spectrum match (PSM) for the given spectrum is reported, along with its score (Nesvizhskii, 2010).

In practice, many expected fragment ions will fail to be observed for any given spectrum, and the spectrum is also likely to contain a variety of additional, unexplained peaks (Noble and MacCoss, 2012). Hence, sometimes the reported PSM is correct — the peptide assigned to the spectrum was present in the mass spectrometer when the spectrum was generated — and sometimes the PSM is incorrect. Ideally, we would report only the correct PSMs, but obviously we are not privy to this information: all we have is the score of the PSM, indicating its quality. Therefore, we instead report a thresholded list of top-scoring PSMs, together with the critical estimate of the fraction of incorrect PSMs in our reported list.

**False discovery rate control in spectrum identification**

To control the FDR in the reported list of PSMs we apply the aforementioned competition framework of target-decoy competition. TDC works by comparing searches against a target peptide database with searches against a decoy database of peptides obtained from the original database by randomly shuffling (or reversing) each peptide in the target database. More precisely, we define $Z_i$ to be the score of the optimal match (PSM) to the $i$th spectrum in the target database, and $\tilde{Z}_i$ as the corresponding
optimal match in the decoy database. We then proceed using the aforementioned competition framework, choose the smallest threshold $T = T(\alpha)$ for which the estimated FDR (given by the ratio of the decoy wins plus one to the target wins greater than this threshold) is still $\leq \alpha$ and report the discoveries as the target PSMs that won their competition: $Z_i > \max\{T, \tilde{Z}_i\}$. It was shown that this procedure obtains rigorous control of the FDR (He et al., 2015; Levitsky et al., 2017) in the spectrum ID setting.

It turns out that there are additional complications in applying the traditional FDR control methods in this setting even beyond the previously outlined issues with empirical $p$-values. The presence of both “native spectra” (those for which their generating peptide is in the target database) and “foreign spectra” (those that do not) in the shotgun proteomics dataset means that we can typically not apply FDR controlling procedures that were designed for a general multiple hypothesis testing context to control the FDR in the spectrum identification problem (Keich and Noble, 2017a).

Unfortunately, it turns out this same issue of the distinction between foreign and native spectra means that our new multiple-competition approach to control the FDR also cannot be applied to the spectrum identification problem (explicitly, it is because the null exchangeability property of Definition 2 below does not hold). Therefore we apply our procedures to the closely related peptide identification problem.

**The peptide detection problem**

The spectrum identification is largely used as the first step in addressing the peptide identification problem that we apply our new methods to. Indeed, to identify the peptides we begin, just as we do in spectrum identification, by assigning each spectrum to the unique target/decoy peptide which offers the best match in the corresponding database. We then assign to each target peptide a score $Z_i$ which is the maximum of all PSM scores of spectra that were assigned to this peptide in the first phase. Similarly, we assign to the corresponding decoy peptide a score $\tilde{Z}_i$, which again is the maximum of all PSM scores involving spectra that were assigned to that decoy peptide. The remainder of the procedure continues using the same TDC competition protocol we outlined above. (Granholm et al., 2013; Savitski et al., 2015).
1.3.2 Knockoffs

As previously mentioned, the subject of competition-based FDR control is not limited to the spectrum/peptide ID problem but has seen application in other fields as well. In particular, this topic has gained a lot of interest in the statistical and machine learning communities following the work of Barber and Candés, who demonstrated how it can be used to control the FDR in feature selection in a classical linear regression model with a procedure called knockoff+ (Barber and Candés, 2015).

The variable selection problem that the knockoff procedure attempts to solve can be described as follows: Suppose we have an observed response vector $y \in \mathbb{R}^n$ that satisfies the classical linear regression model

$$y = X\beta + \epsilon,$$  \hspace{1cm} (1.4)

where $X$ is the $n \times p$ known, real-valued, design matrix, $\beta \in \mathbb{R}^p$ is the unknown vector of coefficients, and $\epsilon \sim N(0, \sigma^2 I)$, where $I$ is the $n \times n$ identity matrix, is Gaussian noise. The variable selection question that we try to answer is which of the explanatory features (columns of $X$) should be included in the model and which should not (e.g., James et al., 2013), or more specifically which $\beta_i \neq 0$?

This model is ubiquitously utilized in many fields of science when trying to explain observed response measurements using a large number of potential explanatory features, and proper selection of these features is critical to developing an understanding of the response variable. Recently, G’Sell et al. suggested using the notion of false discovery rate (FDR) as a way of gauging and hence controlling the quality of a selected set of variables (G’Sell et al., 2016). In the model selection context FDR amounts to the expected proportion of the variables that were erroneously added to the model.

Barber and Candés’ knockoff+ variable selection procedure as outlined in their paper uses an identical competition framework to that applied in TDC to rigorously control the FDR in this finite variable selection context. Briefly, knockoff+ relies on introducing an $n \times p$ knockoff design matrix $\tilde{X}$, where each column consists of a knockoff copy of the corresponding original variable. Much like in the TDC case, these knockoff variables are constructed so that in terms of the underlying regression problem the true
null features (the ones that are not included in the model) are in some sense indistinguishable from their knockoff copies. The procedure then assigns to each null hypothesis, $H_i : \beta_i = 0$, two test statistics $Z_i$ and $\tilde{Z}_i$ which correspond to the point $\lambda$ on the Lasso path (Tibshirani, 1996) at which feature $X_i$, and respectively, its knockoff competition $\tilde{X}_i$, first enters the model when regressing the response $y$ on the augmented design matrix $[X \tilde{X}]$.

Barber and Candés then apply exactly the competition procedure outlined earlier to the original-knockoff pair $\{Z_i, \tilde{Z}_i\}$ to control the FDR amongst the features determined to be in the model. For each threshold we report the variables that both beat their knockoff copies and are larger than said threshold, with an estimate of the false discoveries given by the number of knockoff wins greater than this threshold (plus one). Barber and Candés also show that this procedure can be considered as a special case of their sequential hypothesis testing procedure Selective SeqStep+ (SSS+) which rigorously formalizes the competition framework in a much more general setting (see Supplementary Section A.1).

While controlling the FDR using knockoffs follows exactly the same procedure as in TDC, generating knockoff scores for a specific variable selection problem can be quite challenging. Certainly, when compared to other applications such as permutation tests or the spectrum ID problem — where it is conceptually simple (though computationally costly) to generate extra decoy scores — obtaining knockoff scores for a given design matrix requires much more effort. Indeed, to construct even a single knockoff for each feature demands the use of a sophisticated construction procedure specially developed by Barber and Candés.

One unfortunate side effect of this sophistication in the knockoff construction is that it is not immediately clear how to generalize Barber and Candés’ procedure to produce multiple knockoff scores in a way that still maintain our desired properties. Additionally Barber and Candés’ construction of the knockoff scores simultaneously defines the observed, or original features scores $Z_i$. Thus, presumably increasing the number of knockoffs from, say $d = 1$ to $d = 3$, would require redefining all the scores.

\footnote{The knockoff+ procedure can utilize other statistics that satisfy a certain exchangeability condition but the one presented here is the focus of Barber and Candés, 2015.}
which is a significant departure from our TDC model where the observed scores are given and we assume we can generate independent decoy scores.

Still, if one can generate $d > 1$ knockoffs then we will be able to apply our new methodology to improve the power of the variable selection procedure. Indeed, in their paper Barber and Candés consider the development of multiple knockoffs to improve power similarly to what is presented here but do not proceed on to develop a practical method to realize this. In this thesis we introduce a procedure to construct a general number $d$ of knockoffs and show that we can apply the multiple competition procedures that we develop in the peptide ID setting to this variable selection problem. Additionally, we investigate some unique issues that occur when extending the knockoff methodology to encompass multiple knockoffs leading to the development of a knockoff construction heuristic known as “batching” to alleviate some of these issues.

The remainder of this thesis proceeds as follows: Chapter 2 investigates multiple competition in a more general context focusing specifically on the example of TDC and peptide detection, developing multiple new procedures that extend the concepts of TDC to $d > 1$. Additionally, we investigate the behavior of these methods with both extensive simulations and real data experiments. Chapter 3 examines the concept of calibration between the hypotheses (a factor of the experiment that considers how similar the associated distributions for each hypothesis are) and investigates if dedicating decoy scores to manually calibrate our raw scores could improve the power of our competition-based procedures. Chapter 4 investigates how our methods behave as we take the limit of $d \to \infty$ and from this investigation we deduce a deep connection between our newly proposed competition-based methods and some well known p-value procedures. Finally, Chapter 5 investigates multiple competition using the specialized framework of Barber and Candés’ knockoff procedure. We investigate the challenges that come with extending the knockoff construction to $d > 1$ and propose solutions, including a specialized extension of our earlier competition procedures that allow us to utilize multiple knockoffs to improve the power in variable selection.

The work in Chapters 2 and 4 was the basis for an accepted submission to RECOMB 2020. Additionally, the work in Chapter 5 was the basis for a submission that is currently under review. Both these works can be found on ArXiv at Emery et al., 2019b
and Emery and Keich, 2019 respectively.
Chapter 2

Multiple Decoy Competition

2.1 Single Decoy TDC

We begin by reviewing the established process for single decoy, or \( d = 1 \), target-decoy competition which will then be extended to a general value of \( d \). First, we formulate a sufficient condition on our decoy scores called the *independent null property* in line with the construction outlined in Chapter 1. We will present this condition for a general \( d \) and for TDC set \( d = 1 \). As before we let our observed (target) score be \( Z_i = \tilde{Z}_i^0 \) and its corresponding \( j \)th decoy score be \( \tilde{Z}_i^j \). Denote \( N \subseteq \{1, \ldots, m\} \) as the set of indices corresponding to the true null hypotheses.

**Definition 1.** We say the data satisfies the *independent null property* if for each true null hypothesis, \( H_i \) with \( i \in N \), the random variables (RVs) \( \{Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^d\} \) are independent and identically distributed (iid), and they are also independent of the observed and competing scores of all other hypotheses \( \{Z_l, \tilde{Z}_l^1, \ldots, \tilde{Z}_l^k\}_{l \neq i} \). Note that even among the true null hypotheses we do not assume that the common distribution of \( \{Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^d\} \) is invariant of \( i \).

Returning to the \( d = 1 \) case, for each hypothesis \( H_i \) we construct one decoy \( \tilde{Z}_i \) such that the above condition is satisfied. To control the FDR using TDC we proceed as follows:
We assign to each hypothesis $H_i$ a score-label pair $(W_i, L_i)$ based on the competition between $Z_i$ and $\tilde{Z}_i$ scores as follows:

$$W_i = \max \left\{ Z_i, \tilde{Z}_i \right\},$$

$$L_i = \begin{cases} 
1 & Z_i > \tilde{Z}_i \\
-1 & Z_i < \tilde{Z}_i \\
0 & Z_i = \tilde{Z}_i
\end{cases}$$

(2.1)

$W_i$ simply takes the value of the best score in the $i$th target-decoy pair and $L_i$ records whether that winning score came from the original/target scores ($L_i = 1$) or the decoy scores ($L_i = -1$). In the event of a tie between the original and the decoy scores we essentially remove that hypothesis from consideration ($L_i = 0$) for the rest of the procedure. Note that for some constructions, such as Barber and Candès’ knockoff procedure (Barber and Candès, 2015), $W_i$ and $L_i$ are combined into a single statistic but here we find it is conceptually clearer to separate them.

Without loss of generality we assume that the $W$ scores are in decreasing order (recalling that larger scores represent a more significant result) such that $W_1 \geq W_2 \geq \cdots \geq W_m$. We then reject all $H_j$ with $j \leq i_\alpha$ and $L_j = 1$ where

$$i_\alpha = \max \left\{ i : 1 + \# \left\{ j \leq i : L_j = -1 \right\} \leq \alpha \right\}$$

(2.2)

The subsequent discovery list is thus $D_T(\alpha) := \{ i : i \leq i_\alpha, L_i = 1 \}$

Intuitively we see that according to the independent null property true null hypotheses are equally likely to have $L_i = 1$ or $L_i = -1$ and moreover, this is independent of the $L$ and $W$ values for all other hypotheses (in particular it is independent of the order of $W$). Therefore the numerator in (2.2) gives an estimate of the number of false discoveries in our discovery list as we expect that the true null hypotheses are evenly distributed between target and decoy wins. Hence, the fraction in (2.2) is a reasonable estimate of the FDR if we look to reject all $W_j \geq W_i$ with $L_j = 1$. Thus we look for $i_\alpha$, the largest $i$ for which this estimate is still below our desired $\alpha$, and reject all $H_i$ that have $L_i = 1$ and scores less than or equal to $W_{i_\alpha}$. 
In their knockoff paper, Barber and Candés prove that this procedure controls the FDR at a level $\alpha$, or explicitly

$$E \left(\frac{|D_T(\alpha) \cap N|}{|D_T(\alpha)| + 1}\right) \leq \alpha,$$

if the target-decoy sets satisfy the conditional null exchangeability property (Theorem 2 from Barber and Candès, 2015). This is a weaker assumption than the aforementioned independent null property and we can generalize it to our setting as follows (again presented for general $d$):

**Definition 2.** Let $S_i := \{Z_i, \tilde{Z}_{i1}, \ldots, \tilde{Z}_{id}\}$ denote the unordered set of the observed and competing null score when testing $H_i$. We say the data satisfies the conditional null exchangeability property if for each $i \in N$ (where again $N \subseteq \{1, \ldots, m\}$ is the indices of the true null hypotheses) the RVs $\{Z_i, \tilde{Z}_{i1}, \ldots, \tilde{Z}_{id}\}$ are jointly exchangeable conditional on the sets $\{S_1, \ldots, S_m\}$. In other words, the order among the $d_1 := d + 1$ observed and competing null scores of each true null hypothesis is uniformly distributed, so each permutation of the rankings of the test scores is equally likely, and moreover, this order is independent of the orders of all other true null observed scores, as well as of all the sets of observed scores.

**Remark 1.** Let $V_i := (\tilde{Z}_i^0 := Z_i, \tilde{Z}_{i1}, \ldots, \tilde{Z}_{id})$ and let $\Pi_{d_1}$ denote the set of $d_1!$ permutations on $\{1, \ldots, d_1\}$. With $\pi \in \Pi_{d_1}$ we define $V_i \circ \pi := (\tilde{Z}_{i \pi(1)}^{\pi(1)-1}, \ldots, \tilde{Z}_{i \pi(d_1)-1})$, i.e., the permutation $\pi$ is applied to the indices of $V_i$. The conditional null exchangeability property can be established if we can show that for any sequence of permutations $\pi_i \in \Pi_{d_1}$ with $\pi_i = l d$ (the identity permutation) for all $i \notin N$, the joint distribution of $V_1 \circ \pi_1, \ldots, V_m \circ \pi_m$ is invariant of the sequence of permutations.

In essence, attaining the conditional null exchangeability property ensures that for each true null hypothesis each possible ranking of the original score amongst its corresponding decoy scores (the permutation of the target-decoy set) is equally likely and thus the probability of the original score obtaining any specific rank is $1/d_1$. Moreover this ranking is independent of the ranks of all the other true null observed scores and of the complete target-decoy set of the false null hypotheses.
Chapter 2. Multiple Decoy Competition

Remark 2. • As phrased, and consistently with Barber and Candès, 2015, we effectively ignore ties (set \( L_i = 0 \), however assigning \( L_i = \pm 1 \) according to a random coin toss is both statistically valid and occasionally practiced in TDC.

• Ties between the scores \( W_i \) can be randomly broken.

2.2 Multiple Decoy Competition

When we now consider extending this competition framework to use a general number of decoys there are two natural paths we can proceed down. We term the two resultant methods the “mirror“ method and the “max“ method. For both procedures we generate \( d \) decoy scores \( \{ \tilde{Z}_{i}^{1}, \ldots, \tilde{Z}_{i}^{d} \} \), for each hypothesis \( H_i \) such that the above conditional null exchangeability property is satisfied. We will later use these two methods as a stepping stone to introduce some tuning parameters and define a general class of methods for multiple competition with \( d \) decoy scores.

2.2.1 The mirror method

We begin with the mirror method and to simplify our notation we first assume that \( d \) is odd (so \( d_1 \) is even) and that there are no ties in the target-decoy sets. Let \( r_i \in \{1, 2, \ldots, d_1\} \) denote the rank of \( Z_i \) in \( \{ \tilde{Z}_{i}^{0} := Z_i, \tilde{Z}_{i}^{1}, \ldots, \tilde{Z}_{i}^{d} \} \) (with higher ranks corresponding to better scores) and \( \tilde{Z}_{i}^{(j)} \), \( j \in \{1, \ldots, d_1\} \), correspond to the \( j \)th order statistic of the \( d_1 \) observations. We define \( W_i \) and \( L_i \) as follows:

\[
W_i = \begin{cases} 
\tilde{Z}_{i}^{(r_i)} = Z_i & r_i > d_1/2 \\
\tilde{Z}_{i}^{(d-r_i+2)} & r_i \leq d_1/2 
\end{cases} 
\]

\[
L_i = \begin{cases} 
1 & r_i > d_1/2 \\
-1 & r_i \leq d_1/2 
\end{cases} 
\]

(2.3)

It is clear to see that if \( d = 1 \) (and there are no ties) then this is equivalent to the definitions of \( W \) and \( L \) for TDC. For \( d > 1 \) (and \( d_1 \) even) we have \( L_i = 1 \) (a target win), and thus \( W_i = Z_i \), if \( Z_i \) is larger than more than half of its decoys, and otherwise this is a decoy win, or \( L_i = -1 \), and \( W_i \) is the decoy that is ranked symmetrically across the
median to $Z_i$ – which is why we refer to this procedure as the mirror method. To give
an example, if $d = 3$ we have $(W_i, L_i) = (Z_i, 1)$ if $r_i \in \{3, 4\}$ but $(W_i, L_i) = (Z_i^{(d)}, -1)$
if $r_i = 2$ and $(W_i, L_i) = (\tilde{Z}_i^{(d)}, -1)$ if $r_i = 1$.

The remainder of the procedure is the same as in TDC. We still expect that the
ture null hypotheses are evenly distributed amongst the target and decoy wins and so
the intuition of TDC still holds. The $W$ scores are sorted and, assuming they are in
descending order, the discovery list is defined as $D_M(\alpha) := \{i : i \leq i_\alpha, L_i = 1\}$, where
$i_\alpha$ is determined by (2.2).

**Claim 1.** If the data satisfies the conditional null exchangeability property (Definition 2) then the mirror method controls the FDR in the reported list of target discoveries, that is,

$$E \left( \frac{|D_M(\alpha) \cap N|}{|D_M(\alpha)| \lor 1} \right) \leq \alpha.$$ 

where the expectation is taken with respect to all true null scores (competing and ob-
erved), i.e., it holds regardless of the values of the target and decoy test scores of the
false null hypotheses.

**Proof.** The conditional exchangeability property implies that given $\tilde{Z}_1^{(d_1)}, \ldots, \tilde{Z}_m^{(d_1)}$ (the
largest observed score for each hypothesis), for each $i \in N$ (true null) the score $Z_i$ is
equally likely to be smallest ($r_i = 1$) or largest ($r_i = d_1$), and moreover these ranks are
independent of one another. Hence, conditional on $W_i = \tilde{Z}_i^{(d_1)}$ for $i \in N$, as well as
on all the other $\{W_i\}_{i \in N}$, the labels $\{L_i\}_{i \in N}$ are iid ± 1 uniform RVs. Since the same
argument applies to conditioning on any other combination of values $W_i = \tilde{Z}_i^{(j_i)}$ for
$i \in N$ and for $j_i \geq d_1/2$, this shows that conditioned on $\{W_i\}_{i=1}^m$, the variables $\{L_i\}_{i \in N}$
are iid ± 1 uniform RVs.

From here on our proof relies heavily on Theorem 3 of Barber and Candès, 2015 and
the discussion following it. Assuming still that the hypotheses are ordered in decreas-
ing $W_i$ scores, we associate with each hypothesis $H_i$ a crude p-value $p_i$ based on the
statistic $L_i = l_i$ as:

$$p_i := P(L_i \geq l_i) = \begin{cases} 1/2 & l_i = 1 \\ 1 & l_i = -1 \end{cases}.$$  

(2.4)
Chapter 2. Multiple Decoy Competition

Note that the equality above follows because for each $i \in N$, $L_i = \pm 1$ uniformly and independently of $W_i$. In particular, it follows that for any $u \in (0, 1)$ and $i \in N$ $P(p_i \leq u) \leq u$, so our null $p_i$ are valid, identically distributed p-values. Moreover, we showed that conditioned on $\{W_i\}_{i=1}^n$ the variables $\{L_i\}_{i \in N}$ are independent, hence our null p-values are independent of one another as well as of the ordering of the hypotheses. Thus, the conditions of Theorem 3 of Barber and Candès, 2015 are satisfied and applying Barber and Candés’ Selective SeqStep+ (SSS+) (Supplementary Section A.1) procedure with $c = 1/2$ controls the FDR. Finally, we note that the mirror method with our above definitions of $W_i, L_i$ (2.3) and the rejection threshold $i_\alpha$ determined by (2.2), coincides with applying SSS+ to the $p_i$ defined by (2.4) with $c = 1/2$. \hfill \Box

Remark 3. While we have presented the definitions of $W_i$ with symmetry across the median, this is not required for FDR control. Instead, we simply desire that for a truly null hypothesis, in the event of a decoy win, $W_i$ is equally likely to have come from any of the decoy scores in the winning ranks. Therefore we can formulate other definitions of $W_i$, such as setting $(W_i, L_i) = (\tilde{Z}_i^{r_i+(d-1)/2}, -1)$ if $r_i \leq d_1/2$, that would also yield procedures that control the FDR; however, such choices would typically offer less power than the mirror procedure. For example, in the above proposed variant, middling target scores are replaced by high decoy scores, whereas for the mirror method, it is the low target scores that are replaced by the same high decoy scores.

To see why this would be worse than the mirror method we note that the higher decoy scores are the ones more likely to appear in the numerator of (2.2). Combined with the fact that generally we expect the density of the target scores to monotonically decrease with the quality of the score, it follows that for any given set of scores the estimated FDR of the variant proposed above will generally be higher when compared with the mirror, and hence we can expect that the power of the variant will be weaker than that of the mirror method.

Currently we have only presented the mirror method using an odd number of decoys, however it can be easily generalized to cases using an even number. In these situations we need to consider what occurs when the observed score is the median of the set $\{\tilde{Z}_i^0 := Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^d\}$. When this occurs we treat these cases like a tie in the single decoy procedure: essentially ignoring these $H_i$ by setting $L_i = 0$. All other ranks of the
observed scores behave as previously stated: if \( r_i > d/2 + 1 \) then \((W_i, L_i) := (Z_i, 1)\), whereas if \( r_i \leq d/2 \) then \((W_i, L_i) := (\tilde{Z}_i^{(d-r_i+2)}, 1)\). We can extend (2.3) to deal with both the even and odd cases:

\[
(W_i, L_i) = \begin{cases} 
\left( \tilde{Z}_i^{(r_i)}, 1 \right) & r_i > d/2 + 1 \\
\left( \tilde{Z}_i^{(d-r_i+2)}, -1 \right) & r_i \leq d/2 \\
(\ast, 0) & r_i = d/2 + 1
\end{cases}
\]  

(2.5)

Claim 1 above still holds, although the proof needs to be adjusted slightly to allow for the case of \( L_i = 0 \). As we will observe later in Sections 2.2.3 and 2.2.4 the mirror method is a special case of a more general class of multiple competition procedures. We refer the reader to Theorem 1 in Section 2.2.4 for the proof of the mirror method in the general setting.

Lastly, we consider how to proceed using the mirror method if there are ties between two scores. There are two cases where this can occur, first between the target score \( Z_i \) and one or more of its corresponding decoy scores \( \{ \tilde{Z}_1^i, \ldots, \tilde{Z}_d^i \} \), and second between values of \( W_i \) (i.e. \( W_i = W_j \) for some \( i \neq j \)). For the latter case we proceed as we did in the single decoy case and simply break ties randomly, but for the former there are multiple ways of handling this issue. The most intuitive is to again break ties randomly, that is, randomly assign the ranks amongst the tied scores, however, we can also construct alternative methods that may be more suited for some applications. One such alternative method is to collapse ties between observed and competing test scores by removing decoys that tie with the target score. For example, if \( Z_i = \tilde{Z}_i^1 = \tilde{Z}_i^2 \) and \( Z_i \neq \tilde{Z}_j^j \) for all \( j \geq 2 \), then \( \tilde{Z}_i^1, \tilde{Z}_i^2 \) are removed and \( d \) is adjusted to \( d - 2 \) for this particular hypothesis \( H_i \). This has the same effect as assigning to \( Z_i \) the rank \( r_i \), which is the minimum over all indices \( j \) such that \( Z_i = \tilde{Z}_i^{(j)} \). In this work we randomly break ties, both in \( r_i \) and \( W_i \).

As we saw in the proof of Claim 1, we can consider the mirror procedure as a special case of Barber and Candés’ Sequential SeqStep+ ordered hypothesis testing procedure with \( c = 1/2 \) (A.1). SSS+ allows for any choice of \( c \in (0, 1) \) and thus a natural question arises of whether it is possible to formulate a procedure that uses a different value of
c. We will now show that we can construct another natural extension to TDC that corresponds to SSS+ with \( c = 1/d_1 \).

### 2.2.2 The max method

This alternative method, which we term the “max method”, also reduces to TDC in the case of \( d = 1 \) but behaves differently when \( d > 1 \) which in some cases can lead to greater power. The difference between this method and the mirror is how we define a target win, with the winning ranks reduced down to only the best rank \( d_1 \). Thus for each hypothesis we will have a target win only if the target score is the best amongst all its corresponding decoys, otherwise it will be defined as a decoy win and \( W_i \) is taken as the largest score in the target-decoy set. Hence, assuming there are no ties within the target-decoy set we can define the max method as follows:

\[
W_i = \tilde{Z}^{(d_1)}_i = \max \{ \tilde{Z}^0_i, \tilde{Z}^1_i, \ldots, \tilde{Z}^d_i \},
\]

\[
L_i = \begin{cases} 
1 & Z_i = W_i \\
-1 & Z_i < W_i 
\end{cases}. \tag{2.6}
\]

Assuming again that \( W_i \) are decreasing, we now define \( i_\alpha \) as

\[
i_\alpha = \max \left\{ i : \frac{1 + \# \{ j \leq i : L_j = -1 \}}{\# \{ j \leq i : L_j = 1 \}} \lor 1 \cdot \frac{1}{d} \leq \alpha \right\}. \tag{2.7}
\]

The intuition here again is clear: amongst the true null hypotheses we expect one target win for approximately every \( d \) decoy wins, and moreover which hypotheses correspond to the target wins is independent of the order of \( W_i \). Therefore, for any given set of \( W \) scores, the number of decoy wins divided by \( d \) gives an estimate of the number of true nulls among the target wins. Hence, we can estimate the FDR using the ratio of above estimate to the number of target wins, and seek the largest number of discoveries for which this FDR estimate is \( \leq \alpha \). The subsequent discovery list for the max method is thus \( D_X(\alpha) := \{ i : i \leq i_\alpha, L_i = 1 \} \), where \( i_\alpha \) is determined by (2.7), and we have the following analog of Claim 1:

**Claim 2.** If the data satisfies the conditional null exchangeability property then the max method controls the FDR in the reported list of discoveries, that is,
\[
E \left( \frac{|D_X(\alpha) \cap N|}{|D_X(\alpha)| \lor 1} \right) \leq \alpha.
\]

**Proof.** The proof of this claim follows in essentially the same manner as claim 1. As before, conditional exchangeability implies that given \( \tilde{Z}_1^{(d_1)}, \ldots, \tilde{Z}_m^{(d_1)} \) (the largest observed score for each hypothesis), for each \( i \in N \) (true null) the score \( Z_i \) is equally likely to be ranked as any of \( 1, \ldots, d_1 \), and moreover these ranks are independent of one another. Therefore, conditional on \( W_i = \tilde{Z}_i^{(d_1)} \) for \( i \in N \), as well as on all the other \( \{W_i\}_{i \in N} \), the labels \( \{L_i\}_{i \in N} \) are iid ±1 RVs with \( P(L_i = 1) = 1/d_1 \). Hence, in this case the p-values, based on the statistic \( L_i \), satisfy

\[
p_i := P(L_i \geq l_i) = \begin{cases} 1/d_1 & l_i = 1 \\ 1 & l_i = -1 \end{cases}.
\]

(2.8)

As in the mirror case, for any \( u \in (0,1) \) and \( i \in N \), \( P(p_i \leq u) \leq u \) so our null \( p_i \) are valid, identically distributed p-values. Thus, combined with the property of the labels outlined above, the conditions of Barber and Candés’ Theorem 3 are satisfied and applying the SSS+ procedure with \( c = 1/d_1 \) controls the FDR. Finally, analogous to the mirror case, the max method with \( W_i, L_i \) as defined in (2.6) and the rejection criterion as defined in (2.7), coincides with applying the Selective SeqStep+ procedure with \( p_i \) as defined in (2.8) and setting \( c = 1/d_1 \). \( \square \)

The benefits of using the max method over the mirror are found when \( \alpha \) is small. In these cases the +1 correction in the numerator of (2.2) can have a significant impact on the estimated FDR for any potential discovery list. For example, if \( \alpha = 0.05 \) then the minimum non-zero number of discoveries we can obtain is \( 1/\alpha = 20 \). Therefore in order to return the discovery list \( D_M(\alpha) = \{1, 2, \ldots, 20\} \), for instance, one would require that \( L_i = 1 \) for \( i = 1, \ldots, 20 \), that is, the best 20 scores all correspond to target wins. However, with the max method, where the numerator of the first term in (2.7) is multiplied by \( 1/d \), this number decreases to \( 1/(d \cdot \alpha) \) and thus we can afford to have \( d - 1 \) decoy wins \( (L_i = -1) \) among the first 20 hypotheses. Unfortunately, in other cases the max method can be far from optimal, in particular when \( \alpha \) and/or \( d \) are not
small the max method will be far too conservative in selecting target-wins, limiting the number of possible discoveries.

2.2.3 General \( c \)

We have now constructed two procedures that correspond to SSS+ using two different values of \( c \). We can these use as stepping stones to construct a general multiple competition meta-procedure that can utilize any possible value of \( c \). To develop our general scheme for a pre-determined \( c = i_c/d_1 \), for some \( i_c = 1, \ldots, d \), we consider the multiple competition framework in 3 parts: First assign the labels \( L_i \), then determine the values of \( W_i \) and finally estimate the FDR and thus obtain the discovery list. Analyzing our previously constructed methods with \( c = 1/2 \) (mirror) and \( c = 1/d_1 \) (max) we see they fit the same general scheme:

1. First, the target/decoy win label, \( L_i \), that is associated with each hypothesis, is determined based on \( r_{i} \), the rank of the target score among the \( d_1 \) scores \( \tilde{Z}^0_i = Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^{d_1} \):

\[
L_{i} = \begin{cases} 
1 & (1 - c)d_1 + 1 \leq r_{i}, \\
-1 & (1 - c)d_1 + 1 > r_{i},
\end{cases}
\]  

(2.9)

where \( c = 1/d_1 \) in the max case, and \( c = 1/2 \) in the mirror (assuming no ties and \( d_1 \) is even).

2. Each hypothesis is assigned a score \( W_i = \tilde{Z}_i^{(s_i)} \), where the “selected rank” \( s_i \) is chosen such that the labels \( L_i \) will still be independent, conditionally on the \( W \) scores. For example, \( s_i \) is always \( d_1 \) for the max method, and in the mirror case \( s_i = r_i \) for an original win \( (L_i = 1) \) and \( s_i \) is the mirror image of \( r_i \) for a decoy win \( (L_i = -1) \).

3. The hypotheses are reordered so that \( W_i \) are decreasing (ties are broken randomly) and the list of discoveries is defined as \( D(\alpha, c) := \{i : i \leq i_{ac}, L_i = 1\} \), where

\[
i_{ac} := \max \left\{ i : \frac{1 + \# \{j \leq i : L_j = -1\}}{\# \{j \leq i : L_j = 1\} \vee 1} \cdot \frac{c}{1 - c} \leq \alpha \right\},
\]  

(2.10)

and again \( c = 1/2 \) in the mirror case and \( c = 1/d_1 \) in the max case.
Assigning the labels and calculating the FDR estimate are easily generalized to allow for any value of \( c \) as shown above: for each hypothesis, we have a target win if the observed score is in the best \( c \cdot d_1 \) ranks (or the top \( 100 \cdot c \)%), otherwise it is a decoy win. Furthermore, assuming we have defined \( L_i \) and \( W_i \) sensibly, we can expect a true null hypothesis to be a target win with probability \( c \) while it will be a decoy win with probability \( 1 - c \). Therefore to obtain an estimate of the FDR we can examine the ratio of the number of decoy wins multiplied by \( c/(1-c) \) to the number of target wins in a given set of \( W \) scores greater than a given threshold as shown in \((2.10)\).

The challenge of constructing our general multiple competition method lies in how to choose the selected ranks \( s_i \) so that the labels \( L_i \) maintain the properties needed for legitimate FDR control (independence conditional on the \( W \)'s). Specifically, the issue is how to define \( W_i \) when \( r_i \in \{1, \ldots, d_1 - i_c\} \) \((L_i = -1, \text{ or a decoy win})\) because when \( r_i \in \{d_1 - i_c + 1, \ldots, d_1\} \) \((L_i = 1, \text{ or an original win})\) we naturally define \( s_i = r_i, \text{ or } W_i = Z_i \).

To solve this problem we construct a (possibly randomized) mapping function \( \varphi : \{1, \ldots, d_1 - i_c\} \mapsto \{d_1 - i_c + 1, \ldots, d_1\} \) that maps the “losing ranks” into the “winning ranks”. We can then define,

\[
s_i = \begin{cases} 
  r_i & L_i = 1 \text{ (so } W_i = Z_i \text{ in an original win)} \\
  \varphi(r_i) & L_i = -1 \text{ (so } W_i \text{ coincides with a decoy score in a decoy win)} 
\end{cases} \quad (2.11)
\]

and for a given choice of \( c \) we will need to define \( \varphi \) so that our assumptions are met and thus the FDR is controlled. For example, consider the max method where \( s_i \equiv d_1 \) (equivalently \( \varphi \equiv d_1 \)): in this case, \( L_i \) is trivially independent of \( s_i \), and thus \( W_i \), and hence by the above discussion the method controls the FDR. In contrast, assuming \( d_1 \) is even and choosing \( \varphi \equiv d_1 \) with \( c = 1/2 \) we see that the decoy winning scores \( \{W_i : i \in N, L_i = -1\} \) will generally be larger than the corresponding target winning scores \( \{W_i : i \in N, L_i = 1\} \). Indeed, when \( L_i = -1 \) we always choose the maximal score \( W_i = Z_i^{(d_1)} \), whereas \( W_i \) is one of the top half scores when \( L_i = 1 \). Hence, for \( i \in N, P(L_i = -1 \mid \text{ higher } W_i) > 1/2 \).
We now construct a sufficient condition for $s_i$ (and thus $\varphi$) such that we obtain FDR control:

**Claim 3.** If the data satisfies the conditional null exchangeability property, and if for any $i \in \mathbb{N}$ and $j \in \{d_1 - i_c + 1, \ldots, d_1\}$

$$P(s_i = j, s_i \neq r_i) = \frac{d_1 - i_c}{d_1 \cdot i_c},$$  \hspace{1cm} (2.12)

where $s_i$, like $r_i$, is also determined only from the order of the scores $\tilde{Z}_i^0 = Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^d$, then

$$E\left(\frac{|D(\alpha, c) \cap N|}{|D(\alpha, c)| \lor 1}\right) \leq \alpha.$$

**Proof.** By (2.12) for any $i \in \mathbb{N}$ and $j \in \{d_1 - i_c + 1, \ldots, d_1\}$,

$$P(L_i = 1 | s_i = j) = P(s_i = r_i | s_i = j)$$

$$= \frac{P(s_i = j, s_i = r_i)}{P(s_i = j, s_i = r_i) + P(s_i = j, s_i \neq r_i)} = \frac{1/d_1}{1/d_1 + (d_1 - i_c)/(d_1 \cdot i_c)} = \frac{i_c}{d_1}.$$

Moreover, the conditional exchangeability and the fact that $r_i$ and $s_i$ are determined from the order of the scores $\tilde{Z}_i^0, \ldots, \tilde{Z}_i^d$ imply that the above equalities hold even when conditioning on $\{W_i\}_{i \neq i}$. Hence, in this case the p-values, based again on the statistic $L_i \in \{\pm 1\}$, satisfy

$$p_i := P(L_i \geq l) = \begin{cases} 
  i_c/d_1 & l = 1 \\
  1 & l = -1 
\end{cases}.$$

(2.13)

As in the previous claims, the proof is completed by invoking Theorem 3 of Barber and Candès, 2015 with $c = i_c/d_1$. \hfill \Box

One potential $\varphi$ that satisfies the above claim is $\varphi_u$, the “random map” that simply randomly and uniformly maps $\varphi : \{1, \ldots, d_1 - i_c\} \mapsto \{d_1 - i_c + 1, \ldots, d_1\}$. Here (2.12) is clearly satisfied:

$$P(s_i = j, s_i \neq r_i) = P(r_i \in \{1, \ldots, d_1 - i_c\}) \cdot P(s_i = j | r_i \in \{1, \ldots, d_1 - i_c\}) = \frac{d_1 - i_c}{d_1} \cdot \frac{1}{i_c},$$

(2.14)
and thus, based on the last claim, multiple competition with $\varphi_u$ controls the FDR.

While we can readily construct $\varphi_u$ for any choice of $c$ and $d$, our ability to utilize other mapping schemes may depend on the value of these parameters. For example, we can reconstruct our mirror method in this framework when $c = 1/2$ and $d_1$ is even using the mapping scheme $\varphi_m$: $\varphi_m(j) := d_1 - j + 1$ (which we hence term the mirror map). Alternatively, we can construct a shift map $\varphi_s$: $\varphi_s(j) = j + d_1/2$, which will result in a third FDR-controlling variant of our meta-procedure for $c = 1/2$. When it comes to selecting the best mapping scheme for use in our multiple competition procedure, we recall the discussion in Remark 3 of Section 2.2.1: we seek that $s_i$ is anti-correlated with $r_i$ rather than non-correlated (or even positively correlated).

This desire for anti-correlation motivates the construction of a more general mapping scheme $\varphi_{md}$ which we call mirandom (“mirror” + “random”) that attempts to extend the anti-correlation mirroring structure to any combination of $c$ and $d$. The idea behind mirandom is to map the losing ranks into the winning ranks in an anti-correlated manner such that each winning rank “covers” the same number of losing ranks. However, this is not possible for every value of $c$ so we introduce some randomness into $\varphi_{md}$ by allowing the mapping of a losing rank into multiple winning ranks probabilistically ensuring that equal coverage is maintained (hence the name mirandom). Before presenting the precise mirandom procedure, we construct some examples to demonstrate its applicability.

It is easy to construct $\varphi_{md}$ when $i_c$ divides $d_1 - i_c$. For the first example, consider a simple case where $d = 5$ and $i_c = 2$ (or $c = 1/3$), we map the losing target ranks $r_i \in \{1, 2, 3, 4\}$ to the winning ranks $s_i \in \{5, 6\}$ by mirroring with respect to the midpoint between those two sets, i.e.,

$$s_i = \varphi_{md}(r_i) = \begin{cases} 6 & r_i \in \{1, 2\} \\ 5 & r_i \in \{3, 4\} \end{cases}.$$

We can observe that this scheme is very intuitive as long as $i_c | (d_1 - i_c)$, and it coincides with the mirror map $\varphi_m$ when $c = 1/2$, $d_1$ is even and there are no ties. However, as mentioned, when $i_c \nmid (d_1 - i_c)$ we need to introduce a degree of randomness into
our mapping, while still maintaining as much of the mirror principle as we can. Again, we present some examples:

• for \( d = 5 \) and \( i_c = 4 \), \( \varphi_{md} \) randomly and uniformly maps the losing rank \( r_i = 1 \) to \( s_i \in \{5, 6\} \), and it similarly maps losing rank \( r_i = 2 \) to \( s_i \in \{3, 4\} \). Figure 2.1 shows a visual representation of \( \varphi_{md} \) for \( d = 5 \) with \( i_c = 1 \) (A), 2 (B), 3 (C), 4 (D), with panel D showing the random selection.

• for \( d = 7 \) and \( i_c = 3 \), \( \varphi_{md} \) maps losing rank \( r_i = 1 \) to \( s_i = 8 \), whereas \( r_i = 2 \) is mapped to \( s_i = 8 \) with probability \( 2/3 \) and to \( s_i = 7 \) with probability \( 1/3 \), similarly, \( r_i = 3 \) is mapped to \( s_i = 7 \), and \( r_i = 4 \) is mapped to \( s_i = 7 \) with probability \( 1/3 \) and to \( s_i = 6 \) with probability \( 2/3 \), and finally losing rank \( r_i = 5 \) is mapped to \( s_i = 6 \). Notably, each rank in the range \( \{6, 7, 8\} \) has the same "coverage", that is, for each \( j \in \{6, 7, 8\} \) the sum of the probabilities \( \sum_{r \in \{1, \ldots, 5\}} P(s_i = j \mid r_i = r) = 5/3 = (d_1 - i_c) / i_c \). In particular, when choosing a losing rank \( r_i \in \{1, \ldots, 5\} \) at random the probability the corresponding \( s_i = j \) is the same \( 1/3 \) for all \( j \in \{6, 7, 8\} \).

More generally the mirandom map \( \varphi_{md} \) for a given \( c \leq \lambda \) is defined in two steps. In the first step it defines a sequence of \( d_1 - i \lambda \) distributions \( F_1, \ldots, F_{d_1 - i \lambda} \) on the range \( \{d_1 - i_c + 1, \ldots, d_1\} \) so that

- Each \( F_l \) is defined on a contiguous sequence of natural numbers.
- If \( j < l \) then \( F_j \) stochastically dominates \( F_l \) and \( \min \text{ support } \{F_j\} \geq \max \text{ support } \{F_l\} \).
- Combined, they satisfy the following equal coverage property: for each \( j \in \{d_1 - i_c + 1, \ldots, d_1\} \), \( \sum_{l=1}^{d_1 - i \lambda} F_l(j) = \frac{d_1 - i \lambda}{i_c} \).

In practice, it is straightforward to construct this sequence of distributions. In the second step, mirandom defines \( s_i \) for any \( i \) with \( r_i \in \{1, \ldots, d_1 - i_c\} \) by randomly drawing a number from \( F_{r_i} \) (independently of everything else).

As usual, in case of an original win \( r_i \in \{d_1 - i_c + 1, \ldots, d_1\} \) (\( L_i = 1 \)) we set \( s_i = r_i \). It follows from the equal coverage property that for any \( i \in N \) and
Chapter 2. Multiple Decoy Competition

### Figure 2.1: Mirandom with $d = 5$

The panels show how to select $s_i$ when using mirandom for $d = 5$ and every possible ranking of the target and decoy scores. Each row represents a target/decoy score rank from 1 (largest) to 6 (smallest) while each column represents a possible target ranking configuration with "X" corresponding to a decoy score and "O" corresponding to a target score. A green filling corresponds to the chosen rank $s_i$ and thus taking the score at that position as our value for $W_i$ while a red filling in the target scores corresponds to a target loss. Partially filled "X"s as in panel (D) mean that you must randomly select from these choices. We examine $c = 1/6$ (A), $c = 1/3$ (B), $c = 1/2$ (C) and $c = 2/3$ (D).

<table>
<thead>
<tr>
<th>A: $c = 1/6$ (max)</th>
<th>B: $c = 1/3$</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Panel A" /></td>
<td><img src="image2" alt="Panel B" /></td>
</tr>
<tr>
<td><img src="image3" alt="Panel C" /></td>
<td><img src="image4" alt="Panel D" /></td>
</tr>
<tr>
<td><img src="image5" alt="Panel E" /></td>
<td><img src="image6" alt="Panel F" /></td>
</tr>
</tbody>
</table>

**Notes:**
- **Panel A:** $c = 1/6$ (max)
- **Panel B:** $c = 1/3$
- **Panel C:** $c = 1/2$ (mirror)
- **Panel D:** $c = 2/3$
Chapter 2. Multiple Decoy Competition

\( j \in \{d_1 - i_c + 1, \ldots, d_1\} \) \((2.14)\) holds. Therefore, by Claim 3, using \( \varphi_{md} \) to define \( s_i \) (and hence \( W_i \)) and proceeding along the lines outlined at the start of this section controls the FDR with \( c = i_c/d_1 \).

Remark 4. In this construction we have restricted \( c \) to sit on the lattice \( \{1/d_1, \ldots, d/d_1\} \).

This is not explicitly required, FDR control will be obtained for any \( c \in (0,1) \), but in practice there is no benefit to selecting a value of \( c \) that is not on this lattice.

Remark 5. At the risk of stating the obvious we note that one cannot develop a general method by simply using Barber and Candès’ SSS+ selecting \( W_i = Z_i \) and using \( 1 - (r_i - 1)/d_1 \) as the corresponding p-value because in this case the order of the hypotheses (according to \( W_i \)) is not independent of the true null p-values.

2.2.4 General \( c \) and \( \lambda \)

We have now introduced the tuning parameter \( c \) to our multiple competition framework and by using the mirandom map \( \varphi_{md} \) we can control the FDR for any combination of \( d \) and \( c \). However, we can actually improve our results even further by adding another tuning parameter \( \lambda \). Recall that at their core all the procedures that have been presented so far rely on Barber and Candès’ Selective SeqStep+ (SSS+) procedure to control the FDR in sequential hypotheses testing, where the hypotheses are ordered in advance according to some outside knowledge which is independent of the null hypotheses p-values (Barber and Candès, 2015). Indeed, as shown in Claims 1, 2 and 3 the parameter \( c \) we keep referring to is the same as SSS+’ parameter \( c \).

In their 2016 paper Lei and Fithian introduced Adaptive SeqStep (AS) which relies on an additional tuning parameter \( \lambda \) to improve on SSS+ in much the same way that adaptive BH procedures, including Storey’s (Storey, Taylor, and Siegmund, 2004), improve on the Benjamini-Hochberg procedure (Lei and Fithian, 2016) \((A.2)\). We can also revise our general scheme to adopt this parameter in our multiple competition framework. We now consider two parameters \( \lambda = i_\lambda/d_1 \) to \( c = i_c/d_1 \) where \( 0 < c \leq \lambda < 1 \) and proceed as follows:
Chapter 2. Multiple Decoy Competition

1. First, $L_i$ is determined based on $r_i$, the rank of the target score among the $d_1$ scores $\tilde{Z}_i^0 = Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^d$:

$$L_i = \begin{cases} 
1 & r_i \geq d_1 - i_c + 1 \\
0 & r_i \in (d_1 - i_\lambda, d_1 - i_c + 1) \\
-1 & r_i \leq d_1 - i_\lambda 
\end{cases}$$

(2.15)

2. Each hypothesis is assigned a score $W_i = \tilde{Z}_i^{(s_i)}$, where the selected rank, $s_i$, is defined as

$$s_i = \begin{cases} 
r_i & L_i = 1 \text{ (so } W_i = Z_i \text{ in an original win)} \\
u_i & L_i = 0 \text{ (where } u_i \text{ is randomly chosen uniformly in } \{d_1 - i_c + 1, \ldots, d_1\}) \\
\varphi(r_i) & L_i = -1 \text{ (so } W_i \text{ coincides with a decoy score in a decoy win)} 
\end{cases}$$

(2.16)

3. The hypotheses are reordered so that $W_i$ are decreasing (ties are broken randomly) and the list of discoveries is defined as $D(\alpha, c, \lambda) := \{i : i \leq i_{ac\lambda}, L_i = 1\}$, where

$$i_{ac\lambda} := \max \left\{ i : \frac{1 + \#\{j \leq i : L_j = -1\}}{\#\{j \leq i : L_j = 1\} \lor 1 \cdot \frac{c}{1 - \lambda} \leq \alpha \right\}.$$  

(2.17)

From this procedure we observe that $\lambda$ acts as a partner to $c$, defining what is a decoy win just as $c$ determines what is a target win. A larger value of $\lambda$ means that we need more decoy scores to beat the target score before we classify that hypothesis as a decoy win.

As before the pertinent question is how to define $\varphi$ and thus choose $s_i$ in the event of a decoy win. The following claim generalizes Claim 3 for $c = i_c/d_1$ and $\lambda = i_\lambda/d_1$ where $0 < c \leq \lambda < 1$:

**Theorem 1.** If the data satisfies the conditional null exchangeability property, and if for any $i \in N$ and $j \in \{d_1 - i_c + 1, \ldots, d_1\}$

$$P(s_i = j, r_i \leq d_1 - i_\lambda) = P(s_i = j, L_i = -1) = \frac{d_1 - i_\lambda}{d_1 \cdot i_c},$$

(2.18)
where $s_i$, like $r_i$, is also determined only from the order of the scores $\tilde{Z}^0_i = Z_i, \tilde{Z}^1_i, \ldots, \tilde{Z}^d_i$, then

$$E\left( \frac{|D(\alpha, c, \lambda) \cap N|}{|D(\alpha, c, \lambda)| \lor 1} \right) \leq \alpha.$$  

**Proof.** By (2.18), for any $i \in N$ and $j \in \{d_1 - i_c + 1, \ldots, d_1\}$,

$$P (L_i = 1 | s_i = j) = \frac{P (s_i = j, L_i = 1)}{\sum_{l \in \{-1, 0, 1\}} P (s_i = l, L_i = l)} = \frac{1/d_1}{(d_1 - i\lambda)/(d_1 \cdot i_c) + (i\lambda - i_c)/d_1 \cdot 1/i_c + 1/d_1} = \frac{i_c}{d_1},$$  

and similarly

$$P (L_i = 0 | s_i = j) = \frac{(i\lambda - i_c)/(d_1 \cdot i_c)}{(d_1 - i\lambda)/(d_1 \cdot i_c) + (i\lambda - i_c)/d_1 \cdot 1/i_c + 1/d_1} = \frac{i\lambda - i_c}{d_1}.$$  

Again, the conditional exchangeability and the fact that $r_i$ and $s_i$ are determined from the order of the scores $\tilde{Z}^0_i, \ldots, \tilde{Z}^d_i$ imply that the above equalities hold even when conditioning on $\{W_i\}_{i \neq i}$. Hence, the p-values based on the statistic $L_i \in \{-1, 0, 1\}$ satisfy

$$p_i := P (L_i \geq l) = \begin{cases} 
\frac{i_c}{d_1} & l = 1 \\
\frac{i\lambda}{d_1} & l = 0 \\
1 & l = -1
\end{cases},$$  

(2.19)

independently of the ordering of the hypotheses, which is determined by the selected $s_i$ (and hence $W_i$). The proof is completed by invoking Theorem 1 of Lei and Fithian, 2016 with $s = c = i_c/d_1$ and $\lambda = i\lambda/d_1$.  

We can readily adjust our current mapping schemes to fit this new setting. For example the random map $\varphi_u$ now maps $\{1, \ldots, d_1 - i\lambda\}$ onto $\{d_1 - i_c + 1, \ldots, d_1\}$ randomly and uniformly. Likewise, we can reconstruct the mirandom map $\varphi_{md}$ by constructing $d_1 - i\lambda$ distributions $F_1, \ldots, F_{d_1 - i\lambda}$ as in the $c$-only case on the range $\{d_1 - i_c + 1, \ldots, d_1\}$, so that the average coverage is now $(d_1 - i\lambda)/i_c$ and (2.18) holds. In practice we do not really need to decide on $s_i$ when the outcome is neutral ($L_i = 0$) because those scores are ignored when determining $i_{a\alpha\lambda}$ in (2.17).
For example, suppose $d = 7$. Then for $i_c = 3\ (c = 3/8)$ and $i_\lambda = 4\ (\lambda = 1/2)$, $\varphi_{md}$ is defined as

$$
\varphi_{md}(j) = \begin{cases} 
8 & j = 1 \\
8\ (\text{with probability }1/3),\ or\ 7\ (\text{with probability }2/3) & j = 2 \\
7\ (\text{with probability }2/3),\ or\ 6\ (\text{with probability }1/3) & j = 3 \\
6 & j = 4 
\end{cases}
$$

As before, note the uniform coverage $(4/3)$ of each value in the range, implying that if $j$ is randomly and uniformly chosen in the domain then $\varphi_{md}(j)$ is uniformly distributed over $\{6, 7, 8\}$.

In Figure 2.2 we present the case of $d = 5$ and a range of different $c$ and $\lambda$ values visually. Particularly, we look at the mirandom map when $i_c = i_\lambda = 3\ (A)$ (the mirror method), $i_c = i_\lambda = 2\ (B)$, $i_c = 2, i_\lambda = 3\ (C)$ and $i_c = 1, i_\lambda = 3\ (D)$.

Remark 6. As stated in Section 2.2.1, in determining the ranks $r_i$ ties are broken randomly and the same applies when ordering the selected scores $W_i$.

### 2.3 Selecting Parameters

One of the most critical decisions in applying multiple decoy competition is the choice of the tuning parameters $c$ and $\lambda$. According to Theorem 1 any pre-selected choice of the parameters will allow the user to control the FDR with multiple competition. We have already seen a couple of examples: the mirror method corresponds to using $\varphi_{md}$ with $c = \lambda = 1/2$ when $d$ is odd, and when $d$ is even it corresponds to using $c = (d/2)/d_1$ and $\lambda = (d/2 + 1)/d_1$. The max method is equivalent to using $\varphi_{md}$ with $c = \lambda = 1/d_1$. However as was mentioned in Section 2.2.2 and will be seen in Section 2.5 different choices of $c$ and $\lambda$, even for the same mapping scheme, can radically affect the final power depending on the experimental factors. These factors are numerous and often unknown to us such as $\pi_0$ (the proportion of true null hypotheses), the separation between a true null and a false null score and the calibration (the similarity between the hypotheses respective null distributions). Furthermore, even when these factors
Chapter 2. Multiple Decoy Competition

Figure 2.2: Mirandom(c, λ) with d = 5. Same as Figure 2.1 but now including some values of λ. We examine c = λ = 1/2 (A), c = λ = 1/3 (B), c = 1/3, λ = 1/2 (C) and c = 1/6, λ = 1/2 (D).
are known, the optimal method still varies in a non-obvious manner. Thus, we must make a decision on how to select $c$ and $\lambda$; do we sacrifice the guarantee of FDR control by using the data to choose a pair of parameters we know are suitable? Or do we select a fixed $c$ and $\lambda$ and hope that these are good fits for the experiment? In this section we explore some potential methods of choosing $c$ and $\lambda$ that fall into one of the aforementioned classes.

### 2.3.1 LF

In their paper introducing Adaptive Seq Step+ Lei and Fithian suggest using $\lambda = 1/2$ and $c = \alpha$, implicitly assuming that $\alpha \leq 1/2$. According to Theorem 1, applying this choice of parameters with the mirandom map - a method we refer to as LF - rigorously controls the FDR. However, due to the finite nature of our problem we are often unable to choose exactly these values, for example there may not be an $i$ such that $i/d_1 = \alpha$. In these cases for $c$ we take the maximum possible $i/d_1 \leq \alpha$ or if there is no such $i$ that satisfies this we set $c = 1/d_1$. For $\lambda$ we perform the reverse and select the minimum possible $i/d_1 \geq 0.5$.

The LF method allows us to make a more adaptive decision for our parameters without sacrificing guaranteed FDR control. Generally speaking, as the FDR threshold $\alpha$ increases we can expect that $i_{\alpha c \lambda}$ will also increase, in particular we can expect that the number of targets and decoys in the FDR estimate will be larger. This means that the $+1$ correction in the numerator of (2.17) will have a smaller impact, thus motivating a choice of a larger $c$ (assuming $\lambda$ is held constant). Unfortunately, while the LF procedure is generally superior to the max and mirror methods, in some cases it can perform quite poorly in regards to power and will produce inferior results compared to one of the aforementioned methods, sometimes by a significant margin.

### 2.3.2 FDS

Even with more sophisticated fixed-parameter selection methods such as LF we often see that a pre-selected choice of $c$ and $\lambda$ is unable to provide a reasonable power across a range of experimental designs. As stated earlier, the problem of how to correctly choose these parameters in order to gain consistently good power across a range of
experiments is a complex one and depends on many experimental factors that we are not privy to. It is for this reason that we turn to data-driven methods that make use of the experimental data to attempt to select the best \( c \) and \( \lambda \) for the current experiment. In particular we will look at making use of Storey, Taylor and Seigmund’s FDR controlling procedure, (2.22) above, to inform our choice of \( c \) (Storey, Taylor, and Siegmund, 2004).

We construct this method by noting that in their paper Lei and Fithian draw similarities between the \((c, \lambda)\)\(^1\) parameters of their AS procedure and the corresponding parameters in Storey’s procedure. Specifically, AS’s \( \lambda \) is analogous to the parameter \( \lambda \) of Storey, Taylor and Siegmund that determines the interval \((\lambda, 1]\) from which \( \pi_0 \), the fraction of true null hypotheses is estimated. Recalling our summary from Section 1.2 above, in the finite sample case \( \pi_0 \) is estimated as:

\[
\hat{\pi}_0^*(\lambda) = \frac{m - R(\lambda) + 1}{(1 - \lambda)m},
\]

where \( m \) is again the number of hypotheses, and \( R(\lambda) \) is the number of discoveries at the threshold \( \lambda \) (the number of hypotheses whose p-value is \( \leq \lambda \)). Meanwhile, the \( c \) parameter is analogous to the threshold

\[
t_\alpha \left( \text{FDR}^*_\lambda \right) = \sup \left\{ t \in [0, 1] : \text{FDR}^*_\lambda \leq \alpha \right\}
\]

of Storey, Taylor, and Siegmund, 2004, where

\[
\text{FDR}^*_\lambda = \begin{cases} 
\frac{m \cdot \hat{\pi}_0^*(\lambda) \cdot t}{R(t) \vee 1} & t \leq \lambda \\
1 & t > \lambda 
\end{cases}
\]

We take this analogy one step further to create our selection procedure. We use the above method of Storey, Taylor and Siegmund, applied to the empirical p-values, to determine \( t_\alpha \) and select that value as our \( c \). Those p-values are computed from the competing decoys:

\[
\tilde{p}_i := 1 - \frac{(r_i - 1)}{d_1}.
\]

\(^1\)Lei and Fithian refer to \( c \) as \( s \).
However, in order to properly apply Storey, Taylor and Seigmund’s procedure we first need to set a value of \( \lambda \).

**Determining \( \lambda \) from the empirical p-values**

As we seek to mimic Storey’s procedure as closely as possible it stands to reason that a reasonable approach to determining \( \lambda \) is to apply Storey’s bootstrap procedure. However in practice we found that using this method (applied using the bootstrap option of the qvalue package) in our setup can often result in a failure to control the FDR. Therefore, instead we devised the following method that seems more appropriate for this discrete/lattice empirical p-value setting.

This approach is inspired by the spline based method of Storey and Tibshirani for estimating \( \pi_0 \) (Storey and Tibshirani, 2003) in that it also looks for the flattening of the tail of the p-values histogram as we approach 1. As our p-values, \( \tilde{p}_i \) of (2.23), lie on the lattice \( i/d_1 \) for \( i = 1, \ldots, d_1 \), instead of threading a spline as in Storey and Tibshirani, 2003, we ask whether the number of p-values in the first half of the considered tail interval \([i/d_1, 1]\) is larger than their number in the second half of this interval.

Specifically, starting with \( i = 2 \) we repeatedly apply the binomial test to check whether the number of p-values in \([i/d_1, i_s/d_1]\) is significantly larger than their number in \([(i_s + 1)/d_1, 1]\) where \( i_s = (i + d_1 - 1)/2 \) if \( i + d_1 \) is odd. If the test is significant at level \( \rho \) (here we used \( \rho = 0.1 \)), then we conclude that we are yet to see the flattening of the tail of the p-values histogram, and we increase \( i \) by 1. Otherwise, the first half of the interval does not have a statistically significant larger number of p-values than the second half, so we set \( \lambda = (i - 1)/d_1 \) (so that the interval \((\lambda, 1)\) from which \( \pi_0 \) is estimated in (2.20) coincides with \([i/d_1, 1]\)).

In the event \( i + d_1 \) is even we cannot split the interval equally so instead we ignore the p-values equal to \( i_s = (i + d_1)/2 \) (essentially removing the middle bin) and proceed as normal. Explicitly, we test to see if the number of p-values in \([i/d_1, i_s/d_1]\) is significantly larger than the number in \([(i_s + 2)/d_1, 1]\) where \( i_s = (i + d_1 - 1)/2 \). If the test is significant then we set \( i = i + 1 \) and continue (after adding back the removed p-values).
We present the exact algorithm as follows: Given an upper bound $\Lambda$ on $\lambda$ (we used 0.95), and a binomial test significance cutoff $\rho$,

1. Initialize: $i := 1$.

2. If $i \geq \Lambda \cdot d_1$ or $i = d$ then
   • set $i_\lambda := i$ and stop.

3. If $i + d_1$ is even then
   • set $i_s := (i + d_1)/2$,
   • $n_p^- := \# \{ \tilde{p}_i \in [(i + 1)/d_1, i_s/d_1] \}$,
   • $n_p^+ := \# \{ \tilde{p}_i \in [(i_s + 1)/d_1, 1] \}$.

4. Otherwise,
   • set $i_s := (i + d_1 + 1)/2$,
   • $n_p^- := \# \{ \tilde{p}_i \in [(i + 1)/d_1, (i_s - 1)/d_1] \}$,
   • $n_p^+ := \# \{ \tilde{p}_i \in [(i_s + 1)/d_1, 1] \}$.

5. Calculate $p_b = P(B \geq n_p^-)$ where $B \sim \text{Binomial}(n_p^+ + n_p^-, 0.5)$.

6. If $p_b > \rho$ then (the remaining tail of the p-value histogram “seems to have flattened”)
   • set $i_\lambda := i$ and stop.

7. Otherwise (we are yet to see the flattening of the tail of the p-value histogram),
   • set $i := i + 1$,
   • return to step 2.

**Finite-Decoy Storey (FDS and FDS$_1$)**

The complete “finite-decoy Storey” (FDS) procedure starts by determining $\lambda$ as above. Then we calculate $c$ by applying Storey, Taylor and Seigmund’s FDR procedure to the
empirical p-values. Given the FDR threshold $\alpha \in (0, 1)$, FDS proceeds along (2.20)-(2.22) using $R(\lambda) = |\{\tilde{p}_i : \tilde{p}_i \leq \lambda\}|$, to determine

$$t_\alpha \left( \hat{\text{FDR}}^*_\lambda \right) = \max \left\{ i \in \{0, 1, \ldots, d_1 \cdot \lambda\} : \frac{m \cdot \hat{\pi}^*_0(\lambda) \cdot i/d_1}{R(i/d_1) \lor 1} \leq \alpha \right\}.$$  

(2.24)

This in principle is our threshold $c$ except that, especially when $d$ is small, $t_\alpha \left( \hat{\text{FDR}}^*_\lambda \right)$ can often be zero which is not a valid value for $c$ in our setup. Hence FDS defines

$$c := \max \left\{ 1/d_1, t_\alpha \left( \hat{\text{FDR}}^*_\lambda \right) \right\}.$$  

(2.25)

With $(c, \lambda)$ determined, FDS continues by applying the mirandom map with the chosen parameter values.

FDS was designed to be as close as possible to Storey, Taylor and Seigmund’s recommended procedure for guaranteed FDR control in the finite $m$ setting. However, we found a variant of FDS - that we denote FDS$_1$ - that often yields better power than the original method. FDS$_1$ differs from FDS as follows:

- When computing $t_\alpha \left( \hat{\text{FDR}}^*_\lambda \right)$ (2.24) we use Storey’s asymptotic formulation which does not include the $+1$ in the estimator $\hat{\pi}^*_0(\lambda)$ (2.20), and maximizes over $i \in \{0, 1, \ldots, d_1\}$.
- Instead of defining $c$ as in (2.25), FDS$_1$ defines $c := \min \left\{ c_{\text{max}}, 1/d_1 + t_\alpha \left( \hat{\text{FDR}}^*_\lambda \right) \right\}$, where $c_{\text{max}}$ is some hard bound on $c$ (we used $c_{\text{max}} = 0.95$).
- If $c > \lambda$ (which can now occur as we search $i \in \{0, 1, \ldots, d_1\}$) then when it comes to invoking mirandom, FDS$_1$ sets $\lambda = c$ for the remainder of the procedure (the FDR controlling part) rather than continue to use the same $\lambda$ determined by (2.3.2).

2.4 Bootstrap-based selection of $c$ and $\lambda$

Even when we consider using data-informed choices of $c$ and $\lambda$, such as with the FDS procedure, it can be difficult to find a method that provides good power in a wide
range of experiments. In Section 2.5 we will see that even with data-driven procedures the optimal method of selecting \((c, \lambda)\) varies in a non-obvious way with the experimental parameters. To address this problem we next propose a resampling based procedure that attempts to select the optimal method\(^2\) for the current set of target and decoy scores, as well as the FDR threshold.

Our underlying assumptions that computing additional null scores is forbiddingly expensive, and that consequently the number of decoys scores is fairly small, imply that when we construct our bootstrap samples for each resampled observed score \(Z_i\), we also need to take with it its corresponding decoys \(\{\tilde{Z}_j^d\}_{j=1}^d\) (rather than regenerating new scores directly from \(H_i\)). We can create such a sample, for example, by independently sampling indices \(i_k \in \{1, \ldots, m\}\) for \(k = 1, \ldots, m\) and then defining the resampled set as \(\{(\tilde{Z}_0^1, \tilde{Z}_1^1, \ldots, \tilde{Z}_d^1), \ldots, (\tilde{Z}_0^m, \ldots, \tilde{Z}_d^m)\}\).

Unfortunately, attempting a direct maximization procedure with these samples will often fail to control the FDR in the resulting discovery list. To solve this problem we introduce two modifications to our procedure. The first is an attempt to capture some measure of the variability we should have obtained had we constructed the ideal sample, i.e. generated new decoy scores directly from their corresponding null distributions. This ensures that we are not maximizing over a sample that is too close to the original (a procedure that would clearly not control the FDR).

The second is to realize that, while selecting a method based purely on maximizing the number of discoveries is not generally valid, in many examples it works just fine. Hence, if we are able to predict if such a greedy approach would fail for the particular data at hand we could abort our maximization procedure and instead fall back on a pre-selected default method. Otherwise, if we expect the maximization approach to control the FDR, we can proceed to optimize with respect to the number of target discoveries.

We can implement both these modifications by labeling our samples. These labels correspond to educated guesses as to which of the hypotheses are true nulls and which are false nulls. The labels not only allow us to gauge the rate of false discoveries when we apply our methods to the bootstrap samples but also give insight into which of the

\(^2\)Note here that we are selecting the optimal selection method for \(c\) and \(\lambda\), not the optimal values.
resampled target-decoy sets are suitable to modify (in our case we permute) to inject the variability we seek into our bootstrap sample.

2.4.1 Segmented Labeling

The effectiveness of our labeling heuristic hinges on how informed our estimates are. A reasonable proposal would be to use our estimate \( \hat{\pi}_0^*(\lambda) \) described in Section 2.3.2 and simply randomly draw \( \hat{m}_1 := m \cdot (1 - \hat{\pi}_0^*(\lambda)) \) of the hypotheses to give false null labels. However, this is not an ideal method as our conjectured false nulls will produce little overlap with the correct false nulls in cases where \( \hat{\pi}_0^*(\lambda) \) is large. That in turn will impair our ability to use these labels in the method selection process. To address this issue we refined our procedure for drawing the conjectured false nulls in three major ways.

First, rather than sampling \( \hat{m}_1 \) conjectured false nulls from the set of all hypotheses we consider increasing sets of hypotheses \( \mathcal{H}_j \subset \mathcal{H}_{j+1} \) and sequentially verify that the number of conjectured false nulls we draw from each \( \mathcal{H}_j \) agrees with our estimate of the number of false nulls in \( \mathcal{H}_j \). Secondly, to determine the number of false nulls in each set, instead of estimating \( \hat{\pi}_0^*(\lambda) \), we utilize the same competition based concept that is used in the testing procedures themselves. Finally, rather than being uniform, our draws within each set \( \mathcal{H}_j \) are weighted according to the empirical p-values so that hypotheses with more significant empirical p-values are more likely to be drawn as conjectured false nulls.

The exact procedure is as follows:

1. Initialize by setting:
   - \( j := 1 \) (\( j \) is the index of the set of hypotheses we currently consider)
   - \( i_1 := 1, i_0 := 0 \) (\( i_j \) is the number of hypotheses in \( \mathcal{H}_j \))
   - \( l := 0 \) (\( l \) denotes the index of last drawn conjectured false null)
   - \( f := (0, 0, \ldots, 0) \) (\( f_i \) is the indicator of whether or not we conjecture \( H_i \) is a false null)

2. Determine \( \lambda \) as described Section 2.3.2
3. Using $c = \lambda$ from step 2 above, apply steps 2-3 of mirandom (Section 2.2.4) to define the assigned scores $W_i$ and labels $L_i$, and order the hypotheses in a decreasing order of $W_i$.

4. Estimate $a_j$, the number of false null hypotheses in $H_j = \{ H_i : i \leq i_j \}$, as

$$a_j := \left( \# \{ i \leq i_j : L_i = 1 \} - \# \{ i \leq i_j : L_i = -1 \} \cdot \frac{c}{1-c} \right) \lor 0.$$

Note that the first term is the number of target wins among the hypotheses in $H_j$ and the second is essentially the numerator of (2.10), which uses the number of decoy wins to estimate the number of false discoveries among those target wins.

5. If $a_j > \| f \|_1$ (the number of conjectured false nulls drawn so far) then draw $a_j - \| f \|_1$ additional conjectured false nulls as follows:

(a) for each $i \in \{ l + 1, l + 2, \ldots, i_j \}$ let $w_i := 1 - \tilde{p}_i$, where $\tilde{p}_i$ are the empirical p-values of (2.23)

(b) while $a_j - \| f \|_1 > 0$:

i. draw an index $i \in \{ l + 1, \ldots, i_j \}$ according to the categorical distribution with a probability mass function proportional to $w_i$

ii. set $f_i := 1$ and $w_i := 0$

6. If $i_j = m$ return the conjectured labels $f$, else continue

7. Set $\delta_{j+1} := i_j - i_{j-1} + 1$ if no new conjectured false null were drawn in step 5, otherwise set $\delta_{j+1} := i_j - i_{j-1}$

8. Set $i_{j+1} := (i_j + \delta_{j+1}) \land m$

9. Set $j := j + 1$ and go back to step 4

Note that step 7 lets the data determine the number of hypotheses in $H_{j+1} \setminus H_j$: this number grows if going from $H_{j-1}$ to $H_j$ we concluded we do not need to draw any additional conjectured false nulls. This scheme is well adapted to handle a fairly common scenario where most of the highest scoring hypotheses are false null, making sure they will be labeled as such in our resamples.
2.4.2 Generating samples with labels

Once we have estimated the labels for our current bootstrap run we are able to construct our samples. As outlined earlier, we seek to use our labels to capture some of the variability in our bootstrap samples that we would have obtained had we generated brand new decoy scores for each resampled hypothesis.

This is done by permuting the resampled score sets that have estimated labels denoting them as true null hypotheses. Specifically, consistently with our assumption that for each true null hypothesis \( H_i \) the variables \( \{ \tilde{Z}_j \}_{j=0}^d \) are exchangeable, we permute each resampled conjectured null target-decoy set. In doing so we typically swap an original score with one of its corresponding decoy scores and in fact, for the procedures considered here, we can just as well randomly sample \( j \in \{0, 1, \ldots, d\} \) and swap \( \tilde{Z}_0^i = Z_i \) with \( \tilde{Z}_j^i \).

We now present the complete algorithm for generating a single bootstrap sample:

1. Generate labels according to the algorithm in Section 2.4.1
2. Independently sample \( m \) indices \( j_1, \ldots, j_m \in \{1, 2, \ldots, m\} \)
3. for \( i = 1, \ldots, m \):
   (a) if \( f_{j_i} = 0 \) draw a permutation \( \pi_i \in \Pi_{d_i} \), else, \( f_{j_i} = 1 \) so define \( \pi_i := Id \in \Pi_{d_i} \) (the identity permutation)
   (b) apply the permutation \( \pi_i \) to \( V_i := (\tilde{Z}_{j_i}^0, \tilde{Z}_{j_i}^1, \ldots, \tilde{Z}_{j_i}^d) \) : \( V_i \circ \pi_i := (\tilde{Z}_{\pi_i(1)}^{\pi_i^{-1}(1)}, \ldots, \tilde{Z}_{\pi_i(d_i)}^{\pi_i^{-1}(d_i)}) \)
4. Return the resampled labeled data \( \{(V_i \circ \pi_i, f_{j_i}) : i = 1, \ldots, m\} \).

2.4.3 Monitored maximization approach to FDR control (LBM)

Once the samples have been generated we proceed onto choosing the selection method for the parameters \( c \) and \( \lambda \) that best matches the current experiment. Recalling the discussion above, while we would most like to select the method that gives the optimal number of discoveries, in general this will violate FDR control. Indeed, when using an analogous bootstrap approach to select the tuning parameter \( \lambda \), Storey et al. did not
seek a $\lambda$ that optimizes the number of discoveries, choosing instead to minimize the bias-variance tradeoff in estimating $\pi_0$.

Despite this, in many examples simply selecting the maximizing method works just fine. Therefore our approach is to use our labels to estimate the FDR of continuing to optimize the parameter selection method. If this FDR is smaller than our desired threshold then we continue and optimize with respect to the number of target discoveries. Otherwise, we suspect that this approach will result in a failure to control the FDR and we fall back to some pre-selected method.

To obtain this estimate we observe both the number of discoveries and the proportion of false discoveries that we obtain by applying each of our candidate procedures to the resampled data across each bootstrap run using our conjectured labels. We then select the best procedure for each run and record its FDP. Averaging these FDP estimates over each run gives us a reasonable conservative estimate for the FDR obtained by selecting the optimal method across all the bootstrap runs for use on our original data. It is this estimate that we use to decide if we can select the best overall method or if we need to default to our pre-selected fall-back method instead. Conceptually, we can consider this procedure in two steps: a testing step followed by a selection step.

In the testing phase we proceed as outlined above apply each candidate FDR control procedure to each of our $n_b$ resamples generated as in 2.4.2. For each considered $\alpha \in \Phi$ (where $\Phi$ is the list of significance levels we wish to test) we note the number of discoveries the candidate method reports at level $\alpha$, as well as the implied FDP among those discoveries – an FDP that we infer from the conjectured labels that come with each resample. In particular, for each resample we note the conjectured FDP of the candidate method that yields the largest number of discoveries (ties in the ranks are broken according to a pre-determined order). We then average those conjectured FDPs of the maximizing methods for each of the $n_b$ resamples. We use this empirical FDR as an estimate of the FDR of a meta-procedure that selects the candidate method that yields the largest number of discoveries on the resamples. Finally, the testing part concludes with reporting whether or not the latter estimated FDR is $\leq \alpha$ (the considered FDR threshold).

If the testing phase reports the FDR is not controlled then the selection phase
chooses a pre-determined fall-back method – here we consistently use FDS$_1$, though another user may specify a different method. Otherwise, the testing phase gave a green light for the maximization so we choose the candidate method that has the highest average rank in terms of the number of discoveries. Again, this average is taken over the $n_b$ resamples and the ranks are specific for the considered $\alpha$ with ties in the ranks broken according to a pre-determined order.

In terms of the candidate methods we consider we need to strike a balance between considering more methods, equivalently more choices of $(c, \lambda)$, and the increasing likelihood that the fall-back would be triggered. In practice, we found that considering the methods of FDS$_1$, mirror and FDS works well and below we refer to this resampling approach using the above three candidate methods (and in that the tie-breaking order where FDS has the highest priority) as *Labeled Bootstrap Monitored Maximization* (LBM).

We present the exact procedure for using LBM to choose our parameter selecting method as follows: Given the list of original and decoy scores, an ordered list of candidate methods $\mathcal{M}$, a fall-back method $M_f$, a set of considered FDR thresholds $\Phi$, and the number of bootstrap samples $n_b$,

1. For each bootstrap/resample run $i = 1, \ldots n_b$:

   (a) Generate a labeled resample as describe in Section 2.4.2 above.

   (b) Apply each method $M \in \mathcal{M}$ to the resample noting the number of discoveries $D^i_M(\alpha)$ for each $\alpha \in \Phi$, as well as the corresponding FDP, $F^i_M(\alpha)$ (computed based on the conjectured labels of the resample).

   (c) For each $\alpha \in \Phi$ sort the methods according to $D^i_M(\alpha)$ with ties broken according to the rank of the methods in the list $\mathcal{M}$, and

      i. record the rank $r^i_M(\alpha)$ of each method,

      ii. record $F^*_i(\alpha) := F^i_M(\alpha)$ where $M$ is the highest rank method (with the largest number of discoveries).

2. For each $\alpha \in \Phi$:

   (a) Estimate the FDR of the direct maximization approach as the simple average

   $$\hat{FDR}_* (\alpha) := \frac{1}{n_b} \sum_{i=1}^{n_b} F^*_i(\alpha).$$
Chapter 2. Multiple Decoy Competition

(b) If $\hat{FDR}_*(\alpha) > \alpha$ (see the comment below) then

- (the FDR of direct maximization seems too high so) set the selected method for this $\alpha$ to the fall-back method: $S(\alpha) := M_f$.

Otherwise,

- (direct maximization seems to work fine so) set the selected method to the one with the highest average rank: $S(\alpha) := \text{argmax}_M \sum_{i=1}^{nb} r^i_M(\alpha)$ (again, ties are broken according to the rank of the methods in the list $M$).

In the results reported below we allowed some slack in the last step of the testing phase when comparing the estimated FDR to $\alpha$. Specifically, because the empirical mean is taken over a relatively small number of resamples (we used $nb = 50$ in our applications), we allow the empirical mean to exceed $\alpha$ by no more than $4\sigma \cdot (1 - \hat{\pi}_0^*(\lambda))$. Here $\sigma$ is the standard deviation of the number of discoveries delivered by the maximal, or highest ranking method in each of the $nb$ resamples, and $\hat{\pi}_0^*(\lambda)$ is the $\pi_0$ estimate used by FDS$_1$ described in Section 2.3.2. In practice, this relaxation lead to some increase in power with no visible impact on the FDR control.

Finally, we note that our procedure reports back discovery lists for an increasing sequence of FDR thresholds $\alpha_i \in \Phi$. In order to maintain some form of monotonicity in the discovery list we added a post-processing step to the selection part. Specifically if $i > 1$, we check if the number of discoveries at $\alpha_i$ is smaller than the number we have when using $\alpha_{i-1}$ (recall that $\alpha_{i-1} < \alpha_i$), and if that is the case, then we override our resampling based selection of the optimal method for $\alpha_i$ and instead we use the same method that was previously selected for $\alpha_{i-1}$.

2.5 Simulations

In order to analyze and compare the performance of our multiple competition methods and propose an overall recommended procedure we examine simulated datasets with both calibrated and non-calibrated scores.

In the non-calibrated case we allow the distribution of the null scores to vary with the hypotheses so we sample from hypothesis-specific distributions. Specifically, for
Chapter 2. Multiple Decoy Competition

simulating using a non-calibrated score we associate with the null hypothesis $H_i$ a normal $N(\mu_i, \sigma_i^2)$ distribution from which its decoy scores are sampled. If $H_i$ is labelled a true null, this is also the distribution from which the observed score is sampled. Otherwise, $H_i$ is a false null, so the observed score is sampled from a $\gamma_i$-shifted normal $N(\mu_i + \gamma_i, \sigma_i^2)$ distribution, where $\gamma_i > 0$. The parameters $\mu_i, \sigma_i^2$, and $\gamma_i$ are themselves sampled with each newly sampled set of scores:

- $\mu_i$ is sampled from a normal $N(\mu, \sigma^2)$ distribution with the hyper-parameters $\mu = 0$ and $\sigma^2 = 1$.
- $\sigma_i^2$ is sampled from $1 + \exp(\omega)$, where $\exp(\omega)$ is the exponential distribution with rate $\omega = 1$.
- $\gamma_i$ is sampled from $1 + \exp(\nu)$, where $\nu$ is a hyper-parameter that determines the separation between the false and true null scores.

When simulating using a calibrated score the parameters $\mu_i, \sigma_i$ and $\gamma_i$ are kept constant with $\mu_i \equiv 0, \sigma_i \equiv 1$ and $\gamma_i$ dependent on the experiment.

We begin with the non-calibrated score simulations in which we drew 10K random sets of target and decoy scores (each with its own randomly drawn values of $\mu_i, \sigma_i, \gamma_i$) for each of the following 600 combinations of parameter (or hyper-parameter) values:

- The number of false null hypotheses, $k$, was set to each value in $\{1, 10, 10^2, 10^3, 10^4\}$.
- For each value of $k$, the total number of hypotheses, $m$, was set to $\min\{a \cdot k, 2 \cdot 10^4\}$ where $a$ was set to each of the following factors $\{1.25, 2, 4, 10, 20, 100, 1000\}$ subject to the constraint that $m \geq 100$.
- For each values of $k$ and $m$ above, the hyper-parameter $\nu$ that determines the separation between the false and true null scores was set to each of the values in $\{0.01, 0.05, 0.1, 0.25, 0.5, 1.0\}$.
- For each combination of values of $k, m$, and $\nu$ above, the number of decoys $d$ was set to each of the values in $\{3, 5, 9, 19, 39\}$.
We then used the 10K sampled sets from each of the 600 experiments to find the empirical FDR as well as the power of each method for each selected FDR threshold \( \alpha \in \Phi \). For a given threshold \( \alpha \), the power of a method is the average percentage of false nulls that are reported by the method at level \( \alpha \), and the empirical FDR is the average of the FDP in the discovery list (both averages are taken over the experiment’s 10K runs). We used a fairly dense set of FDR thresholds \( \Phi \): from 0.001 to 0.009 by jumps of 0.001, from 0.01 to 0.29 by jumps of 0.01, and from 0.3 to 0.95 by jumps of 0.05.

Examining first the ratio between the empirical FDR and the selected threshold we note that all the methods seem to largely control the FDR (Figure 2.3). Indeed, the maximal value of the empirical FDR over the nominal FDR across the 600 experiments is no greater than: 1.0004 (mirror), 1.0376 (max), 1.0390 (LF), 1.0414 (FDS), 1.0414 (FDS\(_1\)), 1.0444 (LBM) and the number of experiments (out of 600) in which the same ratio of the empirical to nominal FDR was > 1.02 for even a single value of \( \alpha \) was: 0 (mirror), 7 (max), 6 (LF), 2 (FDS), 2 (FDS\(_1\)), 11 (LBM). Recalling that the LF and the max methods have guaranteed finite sample FDR control it is instructive to see that their empirical violations of FDR control are roughly in line with those of LBM and of the FDS variants.

In terms of power, panel A of Figure 2.4 shows that FDS\(_1\) is almost uniformly more powerful than the mirror for the more commonly used FDR threshold values of \( \alpha \leq 0.5 \), and it is mostly more powerful than the LF and max methods for the same thresholds (panels B, and D). Similarly, FDS\(_1\) generally delivers more power than FDS (panel C), however here we find cases where FDS\(_1\) is significantly weaker than FDS. Those latter cases aside, if we had to pick a single method in this non-calibrated score case, the clear winner would have been FDS\(_1\). Unfortunately, the power picture becomes significantly murkier below once we also consider simulations using a calibrated score, as well as when we revisit below the examples of Section 1.2. Indeed, this murkier picture was the motivation for introducing our resampling-based LBM which tries to find the most appropriate method for the data at hand. Overall, still in the non-calibrated case, LBM seems to be on-par and arguably more powerful than FDS\(_1\): considering more practical FDR thresholds (\( \alpha \leq 0.3 \)) LBM enjoys a significant power advantage over FDS\(_1\) in some experiments while losing by considerably smaller margins in some others (Figure 2.4 panel E). Importantly, for practical FDR thresholds LBM
Figure 2.3: FDR control (non-calibrated data). The panels show the ratio of the empirical FDR to the selected FDR threshold, and each is made of 600 curves, each of which corresponds to one experiment involving 10K randomly drawn sets. The empirical FDR is the 10K-sample average of the FDP of each method’s discovery list at the selected FDR threshold. The 10K sets were drawn simulating a non-calibrated score and using the experiment-specific parameter combination.
A: mirror vs. FDS

B: LF vs. FDS

C: FDS vs. FDS

D: max vs. FDS

E: LBM vs. FDS

**Figure 2.4:** Power relative to FDS (non-calibrated score). Each of the panels show the difference in the average power of the two methods compared (positive values indicate the first method is more powerful). Each panel is made of 600 curves, each of which shows the difference in power averaged over the 10K sets. The sets were drawn simulating a non-calibrated score and using the experiment-specific parameter combination. The power of each method is the 10K-average percentage of false nulls that are discovered at the given FDR threshold. Note that figures’ y-axes are on different scales.
A: mirror vs. LBM  
B: LF vs. LBM  
C: FDS vs. LBM  
D: max vs. LBM  
E: TDC vs. LBM  
F: aTDC vs. LBM  

FIGURE 2.5: Power relative to LBM (non-calibrated score). Same as Figure 2.4 but comparison is with LBM (positive values indicate the first method is more powerful).
is essentially uniformly better than the mirror (Figure 2.5 panel A) and is mostly more powerful than LF, FDS, and max (panels B-D). Lastly, we compare this method to the currently accepted procedure of single decoy TDC and observe that, as expected, LBM provides superior power across the vast majority of the 600 experiments (panel E).

Our calibrated score simulation also consisted of 600 experiments, or combinations of parameter values. Specifically, we used the same values of \(k, m\) and \(d\) as in the above non-calibrated simulations and we let \(\gamma\) vary over the values in \(\{0.8, 1, 1.4, 2, 3, 4\}\). In each experiment we again draw 10K random sets of observed and competing scores using \(\mu_i \equiv 0, \sigma_i \equiv 1, \gamma_i \equiv \gamma\). Interestingly, while there are still many examples where FDS\(_1\) is significantly more powerful than the mirror in this calibrated score context, we now also find many experiments where the opposite holds for practical FDR thresholds (panel A of Figure 2.7). The same applies for comparing the power of LF and FDS\(_1\) (panel B), while FDS\(_1\) is apparently uniformly more powerful than FDS when using a calibrated score and FDR thresholds of practical interest (panel C). While in the non-calibrated simulations LBM was only arguably somewhat better than FDS\(_1\), it is clearly overall more powerful here, certainly for FDR thresholds \(\leq 0.4\) (panel E).

Thus, it is not surprising that when LBM is compared with all the other methods we present here it fares significantly more favorably than FDS\(_1\) (Figure 2.8). Specifically, for low FDR thresholds LBM significantly outperforms the other methods in many of the experiments while rarely losing more than 5% power for FDR thresholds \(\leq 0.2\), and always less than 10% power for FDR thresholds \(\leq 0.5\).

In terms of FDR control we see similar results to the non-calibrated data (Figure 2.6). The maximal value of the empirical FDR over the nominal FDR across the 600 experiments is no greater than: 1.0022 (mirror), 1.0385 (max), 1.0480 (LF), 1.0301 (FDS), 1.0301 (FDS\(_1\)), 1.0315 (LBM) and the number of experiments (out of 600) in which the same ratio of the empirical to nominal FDR was > 1.02 for even a single value of \(\alpha\) was: 0 (mirror), 2 (max), 9 (LF), 1 (FDS), 1 (FDS\(_1\)), 5 (LBM).

We also verify our comments from Sections 2.2.1 and 2.2.3 that a mapping scheme where \(s_i\) is anti-correlated with \(r_i\) will give better results than other potential mapping schemes. Specifically, we examine 100K runs for each of the parameter combinations above (for both calibrated and non-calibrated experiments) and compare the
Figure 2.6: FDR control (calibrated data). Same as Figure 2.3 only the 10K sets were drawn simulating a calibrated score.
Chapter 2. Multiple Decoy Competition

A: mirror vs. FDS

B: LF vs. FDS

C: FDS vs. FDS

D: max vs. FDS

E: LBM vs. FDS

Figure 2.7: Power relative to FDS (calibrated score). Similar to Figure 2.4 only using calibrated data (positive values indicate the first method is more powerful).
Chapter 2. Multiple Decoy Competition

Figure 2.8: Power relative to LBM (calibrated score). Same as Figure 2.4 but comparison is with LBM using calibrated scores (positive values indicate the first method is more powerful).
Figure 2.9: Power relative to the mirror (non-calibrated score). Same as Figure 2.4 but comparison is with the mirror (positive values indicate the first method is more powerful).
Chapter 2. Multiple Decoy Competition

61

Randomized uniform map ($\phi_u$) vs mirror map ($\phi_m$)

![Figure 2.10: Power difference of mapping schemes.](image)

Figure 2.10: **Power difference of mapping schemes.** The figure is made of 1200 curves, each of which shows the difference in power of the randomized uniform map ($\phi_u$) compared to the mirror map ($\phi_m$) for $c = \lambda = 1/2$ averaged over the 100K sets (positive values indicate the randomized map is more powerful). The sets were drawn simulating both calibrated and non-calibrated scores using the experiment-specific parameter combination. The power of each method is the 100K-average percentage of false nulls that are discovered at the given FDR threshold.

Examining Figure 2.10 we see strong evidence supporting our claim. The mirror map appears to be consistently beating the randomized map across the 1200 experiments, with only extremely minor losses ($< 0.5\%$ power) due to random variation. Often we can observe significant power gains, even up to 30\% extra power in some cases, providing clear evidence for the benefits of an anti-correlating mapping scheme.

Lastly, we go back to the two examples of Section 1.2 and note that all our methods essentially control the FDR with the empirical FDR (FDP averaged over the 1K samples sets) below the selected FDR threshold for all $\alpha \in \Phi$ with a single exception in Example 1 at $\alpha = 0.2$, where FDS, FDS$_1$, and LBM have an empirical FDR in the range of 0.203-0.206 or up to 3\% over the threshold (for reference, the empirical FDR of the guaranteed max method here is even higher: 20.9\%), compared with Storey using
pooled p-values that give a resultant FDR of more than double $\alpha$.

Interestingly, when comparing the power of our methods in Example 2, where BH applied to the pooled p-values made no true discoveries even at $\alpha = 0.65$, we find that both the mirror and FDS$_1$ are significantly weaker than FDS, LF and LBM, again demonstrating the utility of LBM. Specifically, at $\alpha = 0.15$ both FDS’s and LBM’s power stand at 79.6% and 79.5% respectively, and LF’s at 62.8% compared with 0% power for both the mirror and FDS$_1$. At $\alpha = 0.2$ FDS, LBM, and LF boast 100% power while the mirror power stands at 0.1% and FDS$_1$ at 0.6%.

### 2.6 Real Data Experiment - Peptide Detection

#### 2.6.1 Data

We now apply our multiple competition procedures to a real peptide spectrum identification experiment. Here we examine three datasets which we denote “human”, “yeast” and “ISB18”.

The human data set consists of a single control run (CTL_R1_1 from the data set with MassIVE identifier MSV000079437 (Zhong et al., 2016). The data was generated on an LTQ-Orbitrap Velos Pro on proteins extracted from human SH-SY5Y cells treated with 200 $\mu$M H$_2$O$_2$. The human reference proteome was downloaded from Uniprot on 28 Nov 2016.

The yeast data set consists of a single run (Yeast_In-gel_digest_2) selected at random from the data set with PRIDE identifier PXD002726 (Schittmayer et al., 2016). The data was generated on an LTQ Orbitrap Velos on proteins extracted from an in-gel digest of $S$. cerevisiae lysate. The yeast reference proteome was downloaded from Uniprot on 28 Nov 2016.

The ISB18 data set is derived from a series of experiments using an 18-protein standard protein mixture (https://regis-web.systemsbiology.net/PublicDatasets, (Klimek et al., 2008)). We use 10 runs carried out on an Orbitrap (Mix_7). The database consists of the 18 proteins from the standard mixture, augmented with the full $H$. influenzae proteome, as provided by Klimek et al.
Searches were carried out using the Tide search engine (Diament and Noble, 2011) as implemented in Crux (Park et al., 2008). The peptide database included fully tryptic peptides, with a static modification for cysteine carbamidomethylation ($\text{C+57.0214}$) and a variable modification allowing up to six oxidized methionines ($\text{6M+15.9949}$). Precursor window size was selected automatically with Param-Medic (May, Tamura, and Noble, 2017). The XCorr score function was employed for uncalibrated searches, using a fragment bin size selected by Param-Medic.

The ISB18 is a fairly unusual dataset in that it was generated using a controlled experiment, so the peptides that generated the spectra could have essentially only come from the 18 purified proteins used in the experiment. We used this to get some feedback on how well our methods control the FDR as explained next.

The spectra set was scanned against a target database that included, in addition to the 463 peptides of the 18 purified proteins, 29,379 peptides of 1,709 $H.\ influenzae$ proteins (with ID’s beginning with $\text{gi|}$). The latter foreign peptides were added in order to help us identify false positives: any foreign peptide reported is clearly a false discovery. Moreover, because the foreign peptides represent the overwhelming majority of the peptides in the target database (a ratio of 63.5 : 1), a native ISB18 peptide reported is most likely a true discovery (a randomly discovered peptide is much more likely to belong to the foreign majority). Taken together, this allows us to gauge the actual FDP for each set of $d$ drawn decoys, FDR threshold $\alpha$, and the FDR controlling procedure that generated the discovery list. Again, the FDP is averaged over the 100 drawn sets of $d$ decoys.

The 87,549 spectra of the ISB18 dataset were assembled from 10 different aliquots, so in practice it means we essentially have 10 independent replicates of the experiment. However, the last aliquot had only 325 spectra that registered any match against the combined target database, compared with an average of over 3,800 spectra for the other 9 aliquots, so we left it out when we independently applied our analysis to each of the replicates. By averaging the above decoy-drawn averaged FDP over the 9 aliquots we get a rough estimate of the FDR that we can compare to the selected FDR threshold.
2.6.2 Methods

Our peptide detection procedure here starts with a generalization of the WOTE procedure of Granholm et al., 2013, scanning each spectrum against the target peptide database \(D_0\), as well as against each of the \(d\) randomly shuffled decoy databases. Specifically, we used Tide (Diament and Noble, 2011) to find for each spectrum its best matching peptide (using the XCorr score here). We then assign to each peptide in each of the databases a score, which is the maximum of all the PSM scores that were optimally matched with that peptide during the scanning phase. If no spectrum was optimally matched to a peptide then that peptide’s score is set at \(-100\).

Next, we consider the score assigned to the \(i\)th peptide in the target database \(D_0\) as our observed score \(Z_i\), and the scores of that peptide’s \(d\) randomly shuffled copies as the corresponding decoy scores: \(\tilde{Z}_i^1, \ldots, \tilde{Z}_i^d\). We can now apply to these scores either TDC (\(d = 1\)) — representing a peptide-level analogue of the picked target-decoy strategy of Savitski et al., 2015 — or any of our more sophisticated multi-decoy procedures. Specifically, in the examples below we looked at the performance of the canonical TDC (with the +1 finite sample correction) as well as that of the mirror method (our first multi-decoy method), LBM (arguably the overall winner of the simulation studies) and aTDC (the averaging-based method of Keich, Tamura, and Noble, 2018 discussed briefly in Section 1.3 above) using \(d \in \{3, 5, 7, 9\}\).

Clearly, the competition-based control of the FDR is subject to the variability of the drawn competing scores. To ameliorate this variability here, we initially searched the spectra against 100 randomly shuffled decoy databases, and then for each \(d \in \{1, 3, 5, 7, 9\}\) we repeated our analysis drawing 100 sets, each with \(d\) of those decoy databases, while making sure that the 100 drawn sets are distinct. We can then compare the number of discoveries reported by each considered method at the selected FDR threshold \(\alpha\) (here \(\alpha \in \{0.01, 0.05, 0.1\}\)). More precisely, for each number of decoys \(d\) we average the number of discoveries over the 100 randomly drawn sets of \(d\) decoys.
2.6.3 Results

We applied our analysis to the human, yeast and ISB18\textsuperscript{3} datasets and examined both the power of the considered methods (in all three datasets) and the FDR control (only in ISB18). Panel D of Figure 2.11 suggests that when applied to the ISB18 dataset all our procedures seem to control the FDR: the empirically estimated FDR is always below the selected threshold. In terms of power, again we see that LBM is the overall winner: it typically delivers the largest number of discoveries, and even in the couple of cases where it fails to do so it is only marginally behind the top method (panels A–C). In contrast, each of the other methods has some cases where it delivers noticeably fewer discoveries. In practice, this means that scientists can extract more useful information (discoveries) from the same wet lab experiment by leveraging more computational power coupled with LBM.

More specifically, for $\alpha = 0.01$ LBM’s average of 142.0 ISB18 discoveries ($d = 3$) represents an 8.0% increase over TDC’s average of 131.5 ISB18 discoveries, and we see a 9.0% increase over TDC when using $d = 5$ (143.2 discoveries). In the human dataset and for the same $\alpha = 0.01$ we see a 3.2% increase in power going from TDC to LBM with $d = 3$ (532.4 vs. 549.4 discoveries), and a 3.9% increase when using LBM with $d = 5$ (553.1 discoveries). LBM offers the biggest gains in the yeast dataset where we see (again $\alpha = 0.01$) a 42.1% increase in power going from TDC to LBM with $d = 3$ (76.3 vs. 108.4 discoveries), and a 46.9% increase when using LBM with $d = 5$ (112.1 discoveries). Moreover, we note that for this $\alpha = 0.01$ TDC reported 0 yeast discoveries in 33 of the 100 runs (each using a different decoy database), whereas LBM reported 0 discoveries in only 1 run for $d = 3$ and in no runs for any other value of $d$ we considered.

At the higher FDR thresholds of 0.05 and 0.1 LBM offers a much smaller power advantage over TDC and is marginally behind for $\alpha = 0.1$ and $d = 3$ in the human and yeast datasets. Also, consistent with our simulations, we find that the mirror generally lags behind LBM, and in fact in these real datasets it is roughly on par with TDC.

\textsuperscript{3} In the case of the ISB18 the data consists of 9 aliquots or replicates, so our analysis was separately applied to each aliquot and then averaged over the aliquots.
Chapter 2. Multiple Decoy Competition

Figure 2.11: Peptide detection. (A: human) The number of discoveries at the given FDR threshold as the average over the 100 randomly drawn decoys sets. The numbers to the left of the markers indicate the number of runs (out of 100) in which no discovery was reported. (B: yeast) Same as (A) but for the yeast dataset. (C: ISB18, power) Similar to (A) and (B) except for the ISB18 where the data consists of 9 aliquots or replicates. Therefore, the number of discoveries is averaged over the 9 aliquots, where for each aliquot we averaged the number of discoveries over 100 randomly drawn sets of d decoys as explained above. The numbers to the left of the marker indicate the aliquots-average number of runs (out of 100) in which no discovery was reported. (D: ISB18, FDR) Similar to (C) only here we noted the putative FDP of each run (as explained in Section 2.6.1), then we averaged the FDP across the 100 runs to get an empirical FDR for each aliquot that we then averaged over the aliquots. Notably, in all cases the empirical FDR was lower than the selected threshold.
Although aTDC was designed for the spectrum identification problem, it is instructive to add aTDC\textsuperscript{4} to this comparison of the peptide detection problem. LBM consistently delivered more detected peptides than aTDC did, although in some cases, such as the yeast dataset, the difference is marginal. Still, in the human dataset for $\alpha = 0.01$ with $d = 3$ we see a 4.2\% increase in power going from aTDC to LBM (527.4 vs. 549.4 discoveries), and with $d = 5$ a 4.0\% increase when using LBM (530.7 vs. 553.1 discoveries). Similarly, in the ISB18 dataset for $\alpha = 0.01$ with $d = 3$ we see a 7.7\% increase in power going from aTDC to LBM (131.8 vs. 142.0 discoveries), and with $d = 5$ a 6.3\% increase when using LBM (134.7 vs. 143.2 discoveries).

\footnote{We used the version named aTDC\textsuperscript{1}, which was empirically shown to control the FDR even for small thresholds / datasets (Keich, Tamura, and Noble, 2018).}
Chapter 3

Calibration

3.1 Introduction - What is calibration?

In Chapter 1 we discussed the issues that occur when we pool decoy scores during the creation of empirical p-values, specifically that we can construct null p-values that are non-uniform if our hypotheses do not all have the same null distribution. Recalling the examples from Section 1.2 we can see that it is possible to observe cases in which the distribution of the null empirical p-values has significant right skew — a fact that can lead to serious failures in both FDR control and power for the canonical testing procedures.

In our multiple competition testing procedure it can be seen that we have a similar issue. Once we have performed our competition step and determined the $W$ scores we next sort them in order of significance. As the values of $W$ are directly obtained from the raw target and decoy scores, it may be the case that some $W_i$ have different measures of significance. So a highly significant score for one hypothesis may be less than a non-significant score of another simply due to an existing shift in the associated null distributions and thus always be placed ahead of it in the ordered list.

While this does not result in the catastrophic failures that occur with pooled p-values (FDR control is guaranteed for procedures such as the mirror, max and LF regardless of the corresponding distributions of the $W$ scores), we may be able to obtain more power if we introduce a procedure to baseline our scores so that the corresponding null distributions are comparable. It is for this reason we consider taking actions to “calibrate” the original raw scores through the use of a procedure known as partial calibration (Keich and Noble, 2017b).
To understand partial calibration we first introduce the notion of well-calibrated scores more rigorously. This idea was previously discussed in a paper by Keich and Noble (Keich and Noble, 2015) in the framework of spectrum identification. They define scores as well-calibrated if a score of $x$ assigned to one spectra $\sigma_i$ has the same (or very similar) significance as a score of $x$ assigned to a different spectra $\sigma_j$. We can consider the notion of calibration in our more general setting by stating that our scores are well-calibrated if a score of $Z_i = z$ for hypothesis $H_i$ has the same, or very similar, significance as a score of $Z_j = z$ for a different hypothesis $H_j$ for all $i \neq j$. We state that scores are perfectly calibrated if a score has the exact same significance for all hypotheses, or alternatively if all null distributions are equivalent (a common example of this is p-values). As mentioned above, if the scores are very poorly calibrated then we can have cases were null scores from one hypothesis can be strictly greater than any (truly null or alternative) score from another.

In their paper Keich and Noble introduce a calibrating procedure that utilizes additional newly generated decoy scores (we will refer to these as calibrating decoys and the decoy scores used for competition in TDC as competing decoys) to turn potentially uncalibrated raw scores (both the target and competing decoy scores) into baselined calibrated scores similarly to the construction of empirical p-values outlined in Section 1.2. Specifically, the new calibrated score was defined as the rank of the raw score amongst the list of corresponding calibrating decoy scores (appended with the score of interest) divided by the number of such scores, or essentially the negative of the empirical p-value that would be constructed using the calibrated decoys (so that it is maintained that larger scores are better). This procedure used a large number (10,000) of calibrating decoys to construct these new calibrated scores and by applying TDC to these values Keich and Noble were able to obtain significant power improvements. While it was shown to work well, one issue that they noted was the requirement of a large number of calibrating decoys which is often computationally infeasible. This lead to the introduction of partial calibration in a later paper (Keich and Noble, 2017b) which utilizes a variable number of calibrating decoy scores in order to calibrate the raw target and competing decoy scores.

Just as in the empirical p-value case, if there is not a large amount of calibrating
decoys then our newly calibrated scores will be very coarse and in particular this will result in many ties in both the $Z$ scores, and subsequently the $W$ scores. Keich and Noble solved this problem by constructing the partially calibrated scores in a two-tiered manner with both a primary and secondary score. This defines a new ordering for the scores in which comparisons are done using the primary score and ties are broken with the secondary score. The primary score is defined as the number of calibrating decoys corresponding to the spectra (or hypothesis in the general case) that the raw target/competing decoy score is greater than (similarly to the complete calibration above) while the secondary score is defined simply as the original raw score of the current target/decoy. Thus, we essentially calibrate as above but break ties between the calibrated scores using the original raw score.

In practice it is simpler to combine the two tiers into one single calibrated score which is calculated for each target and competing decoy score. For a general number $d_{\text{comp}}$ of competing decoys (in standard TDC $d_{\text{comp}} = 1$ and for the uncalibrated multi-decoy setting $d_{\text{comp}} = d$) and $d_{\text{cal}}$ calibrating decoys let $C^j_i$, $i = 1, \ldots, m, j = 1, \ldots, d_{\text{cal}}$ be the calibrating decoy scores for the $i$th hypothesis and let $\tilde{r}^k_i$ be the overall rank of $Z^k_i$, the $k$th decoy score ($k = 0$ corresponds to the target score) for $k = 0, \ldots, d_{\text{comp}}$, compared to all target and competing decoy scores. We thus define the calibrated score for the $k$th decoy score ($k = 0$ corresponds to the target score) of the $i$th hypothesis, $q(Z^k_i)$ as follows:

$$
q(Z^k_i) := \# \left\{ j : C^j_i < Z^k_i \right\} + \frac{\# \left\{ j : C^j_i = Z^k_i \right\}}{2} + \frac{\tilde{r}^k_i}{1 + 2m(d_{\text{comp}} + 1)}
$$

In words, we first count the number of calibrating decoys our raw score is greater than (ties correspond to a half count) then add the overall rank of the score amongst all target and competing decoy scores scaled down so that these ranks lie in $(0, 1/(2 \ast d_{\text{comp}}))$. This formulation has the same effect as the two-tiered score from Keich and Noble, 2017b in which the original raw score is used to break ties when two scores beat the same number of calibrating decoys.

It has been shown in the framework of the spectrum ID problem (Keich and Noble, 2015; Keich and Noble, 2017b; Jeong, Kim, and Bandeira, 2012) that both complete
calibration and even partial calibration can lead to large increases in power. Keich and Nobel were able to see improvements of up to 22% (with 63 calibrating decoys) through the application of partial calibration in a spectrum ID experiment. In this chapter we will examine how calibration can be applied to the multiple decoy setting and investigate if improved results can be obtained through this procedure.

### 3.2 Calibration or Competition - Assigning decoys

Partial calibration as introduced in Keich and Noble, 2017b generates new decoy scores explicitly for calibrating purposes. However, when we consider adapting this to the framework of multiple-competition there are two points we must take note of. First, according to our original assumption in Chapter 1 it is computationally impractical to generate any more decoys past the $d$ we have already constructed for competition. The second is that, even if one were to generate new decoys, it may be the case that these would be better off being used as additional competing decoys instead of for calibration. Therefore, when we consider calibration in the framework of multiple competition we should instead examine how to split a fixed number of decoys, $d$, between calibration and competition such that $d_{\text{comp}} + d_{\text{cal}} = d$, where again $d_{\text{comp}}$ and $d_{\text{cal}}$ are the number of competing and calibrating decoys respectively.

From the work of Keich and Noble, 2017b it is known that if one fixes the number of competing decoys then additional calibrating decoys will improve the power of the testing procedures (though the effect can be minor). Likewise, it is clear from the discussion in Chapter 2 that adding additional competing decoys will also improve results, primarily as we have more possible selections for our choice of $c$ and $\lambda$. Thus, our assignment scheme must balance the benefits of both potential options.

#### 3.2.1 Fixed Assignment

We first examine this problem with a focus on maintaining guaranteed FDR control. If $d_{\text{comp}}$ and $d_{\text{cal}}$ are chosen without looking at the data then the assumptions in Section 2.2.4 are still valid and FDR control will be achieved (provided that $c$ and $\lambda$ are also chosen without using the data). Thus, we seek to develop a method that splits the
decoys between competition and calibration in a way that allows for improved power across a range of experimental designs but does not use the data to inform our choice.

This is a difficult problem as the optimal choice can vary significantly based on various experimental parameters. Indeed, as the viable choices of $c$ and $\lambda$ are dependent on $d_{\text{comp}}$, it stands to reason that the optimal value of the split will be similarly complicated. As we have seen in Chapter 2, correctly choosing $c$ and $\lambda$ is critical to obtaining a good level of power, a fact that offers a starting point for developing a potential splitting procedure. Focusing on the selection methods with guaranteed FDR control where $c$ and $\lambda$ are pre-selected such as LF, max or mirror, we would prefer that our calibration procedure does not significantly impact the value that our selection methods assign to these parameters. For example, when using LF with $\alpha = 0.2$ we would like to set $c = 1/5$, so ideally our calibration procedure would make sure we have at least 4 competing decoys to ensure $c = 1/5$ is a viable threshold ($\lambda = 1/2$ is always viable with at least 1 competing decoy). Note that in practice, as the values for $(c, \lambda)$ sit on the lattice of $i/(d_{\text{comp}} + 1)$ for $i = 1, \ldots, d_{\text{comp}}$, we do not explicitly require that the exact values are maintained. Instead, our splitting algorithm assigns just enough decoys to competition such that the resulting lattice encompasses the original choices of $c$ and $\lambda$, i.e. $c \geq 1/(d_{\text{comp}} + 1)$ and $\lambda \leq d_{\text{comp}}/(d_{\text{comp}} + 1)$.

The exact procedure is as follows:

1. Select a preferred testing method and determine $c_m$ and $\lambda_m$, the parameter selections when the preferred testing method is applied with all $d$ decoys assigned to competition.

2. Set $i_c^m = \lfloor c_m \cdot d_1 \rfloor$ and $i_\lambda^m = \lceil \lambda_m \cdot d_1 \rceil$ (where again $d_1 = d + 1$).

3. If $|i_\lambda^m - d_1/2| > |i_c^m - d_1/2|$ then set $i_{\text{comp}} = i_\lambda^m$, otherwise set $i_{\text{comp}} = i_c^m$.

4. Set $d_{\text{comp}}$ to be the minimum number of decoys such that $c_m \geq 1/(d_{\text{comp}} + 1)$ and $\lambda_m \leq d_{\text{comp}}/(d_{\text{comp}} + 1)$. Specifically, if $i_{\text{comp}} \leq d_1/2$, set $d_{\text{comp}} = \min \{\lceil d_1/i_{\text{comp}} - 1 \rceil, d\}$, otherwise set $d_{\text{comp}} = \min \{\lfloor d_1/(d_1 - i_{\text{comp}}) - 1 \rfloor, d\}$.

5. Assign $d_{\text{comp}}$ decoys to competition at random.

6. Set $d_{\text{cal}} = d - d_{\text{comp}}$ and assign the remaining $d_{\text{cal}}$ decoys to calibration.
For example, if we are using the mirror method with $c = \lambda = 1/2$ then $d_{\text{comp}} = 1$ and $d_{\text{cal}} = d - 1$ (this is identical to calibrated TDC). For LF with $c = \alpha, \lambda = 1/2$ we have that $d_{\text{comp}} = \min \left\{ \lceil 1/\alpha \rceil - 1, d \right\}$ and $d_{\text{cal}} = d - d_{\text{comp}}$ (for $\alpha \leq 0.5$). This procedure is easily generalizable to any fixed choice of $c$ and $\lambda$ and allows us to ensure we are able to reach the target values for our parameters as best we can.

Applying this formulation to procedures that use data-driven selections of $c$ and $\lambda$ such as FDS is still possible by following the algorithm above. We apply our chosen parameter selection procedure to the uncalibrated data, observe the choice of $i_c$ and $i_\lambda$, then proceed with calibration as stated. However using c-calibration with these selection procedures is not recommended as there may be significant discrepancy between the values that are selected for $(c, \lambda)$ before and after calibration due to the change in $d_{\text{comp}}$, a fact that could potentially lead to suboptimal results. Furthermore, as data-driven selection procedures do not have guaranteed FDR control to begin with there is little reason to use such a restrictive calibration method. Instead, just as we did when deciding $c$ and $\lambda$, we look to utilize our data in estimating the optimal split.

### 3.2.2 Data-Driven Assignment

We expect that finding the optimal way to split the decoys between calibration and competition to be very difficult, just as it was for selecting the optimal values for $(c, \lambda)$. Indeed, as stated above, the number of competing decoys, and thus the calibration-competition split, informs the possible options that we can select for $c$ and $\lambda$, thus it stands to reason that the optimal value for this split will also vary in a complex manner with the various experimental parameters. This means that implementing a pre-selected fixed split such as c-calibration without consideration for the properties of the data can lead to poor results. As a simple example, it is clear that if we try to apply c-calibration to data that is already calibrated then all we will achieve is wastefully throwing away decoys that could have been used for competition.

For this reason we develop data-driven methods that, just as in Section 2.3, examine the data before making a decision about how to calibrate. A simple method of this type is to apply a Kruskal-Wallis test on the decoy scores to determine whether the data is already calibrated by testing if the null distributions of each hypothesis are equivalent.
If the test returns that the decoy scores are not calibrated we can proceed with a fixed calibration method, otherwise we assign all the decoys to competition.

While the Kruskal-Wallis test is indeed capable of discerning poorly calibrated data from well-calibrated data, when we consider the results from a power perspective things are more complex. The interplay between $d_{\text{comp}}$ and $d_{\text{cal}}$ and how it relates to power is not solely dependent on the level of calibration of the data and even when the data are poorly calibrated it can be the case that diverting decoys away from competition to calibration can lead to an overall loss in power. It is for this reason that we must make more use of the data than a simple calibration check and develop procedures that can find the optimal calibration split for any given experiment.

One such procedure, which we term step-calibration, proceeds by breaking the decoys into groups and sequentially testing whether each group would be better suited to be assigned to either calibration or competition. Specifically, we first assign a single decoy to competition (as we must have $d_{\text{comp}} \geq 1$) and split the remaining $d - 1$ decoys into groups with the goal to sequentially assign each group to either calibration or competition based on which explains the most variability in the discovery list. This is done by separately assigning all the current group decoys as a whole to both choices separately and performing multiple competition using these two assignments, obtaining a discovery list for each. The final group assignment choice is made by examining the symmetric difference between each of the two lists and the discovery list of the selected previous group assignment (if we are in the first group then we compare with the discovery list obtained by applying TDC using only the sole pre-assigned competing decoy) and selecting the choice that has the largest such difference. Note that we do not directly optimize the number of discoveries but instead measure the symmetric difference between the discovery lists to determine which option has the greatest impact when the group decoys are assigned to it.

In applying the step-calibration procedure there are a number of decisions that must be made regarding how to proceed. The first of these is how to construct the groups. In this work we use a grouping scheme that utilizes an initial group of size 1 (as we worked primarily with odd numbers of decoys) and then constructs groups of size $2^{(i-2)}$ for $i \geq 2$ such that size of each subsequent group exponentially increases as we...
progress. For example, if \( d = 9 \) then we have four groups \( G_i \) for \( i = 1, 2, 3, 4 \) with \( |G_1| = 1, |G_2| = 1, |G_3| = 2 \) and \( |G_4| = 4 \) (plus 1 decoy pre-assigned to competition).

The second and third choices relate to the tests that we perform to make the decision of where to best assign the current decoy group. We must select a method of choosing \( c \) and \( \lambda \) as well as a significance level \( \alpha_{\text{cal}} \) that we perform this multiple competition test at. \( \alpha_{\text{cal}} \) is best chosen equal, or at least close, to the value of \( \alpha \) that the user wishes to test at in order to prevent over or under-calibration. In the event that we test multiple \( \alpha \) simultaneously then we can choose to either prioritize a single value (here we select \( \alpha = 0.1 \)) or implement a different calibration split for each \( \alpha \), though this can be very computationally intensive and can cause issues with monotonicity of the final discovery lists. For the selection of the testing method we recommend selecting LF. While matching the users preferred testing method is best, if one is using LBM (our recommended testing procedure) it is computationally impractical to perform multiple iterations of the bootstrap procedure in the step algorithm. For that reason LF is recommended as it will not bias the method selection step in LBM (for example, if FDS is used in the calibration procedure then it is likely that calibration will improve the results for FDS more so than the mirror method which could lead to a bias in LBM’s method selection). Furthermore, in practice there is little difference between the final power of methods calibrated using step-calibration with FDS as the calibrating procedure compared to LF and thus we present it as a general recommendation.

The exact recommended procedure to obtain the list of competing decoys and calibrating decoys proceeds as follows:

1. Initializing set:
   - \( S_{\text{comp}} = \{ Z^0 = Z, Z^{(1)} \} \) - The target and decoy scores used for multiple competition.
   - \( S_{\text{cal}} = \{ \} \) - The decoy scores used for calibration.
   - Set \( n_g = \lceil \log_2(d) \rceil \) - the number of groups.

2. Perform single-decoy TDC on \( S_{\text{comp}} \) and set the discovery list as \( D_0 \).

3. For \( i \) in \( 1 : n_g \).
(a) Set $S_{G_i} = \{Z^{(2^{i+1}-1)}, \ldots, Z^{2^i}\}$.

(b) Perform multiple competition at $\alpha = \alpha_{cal}$ using $S_{comp} \cup S_{G_i}$ with $c$ and $\lambda$ defined by the LF scheme from Section 2.3 ($c = \alpha, \lambda = 0.5$) and set the discovery list as $D_{comp}$.

(c) Calculate $q(S_{comp})$, the set of calibrated scores for $S_{comp}$, by calibrating the original scores using $S_{cal} \cup S_{G_i}$.

(d) Perform multiple competition at $\alpha = \alpha_{cal}$ using $q(S_{comp})$ with the LF scheme and record the discovery list as $D_{cal}$.

(e) if $|D_{comp} \triangle D_{i-1}|$ (the symmetric difference between $D_{comp}$ and $D_{i-1}$) $\geq |D_{cal} \triangle D_{i-1}|$:

   i. Set $D_i = D_{comp}$.

   ii. Set $S_{comp} = S_{comp} \cup S_{G_i}$ (assign all group decoys to competition).

else

   i. Set $D_i = D_{cal}$.

   ii. Set $S_{cal} = S_{cal} \cup S_{G_i}$ (assign all group decoys to calibration).

4. Return $S_{comp}$ and $S_{cal}$.

In practice performing step calibration on the original data will sometimes result in an increase of the FDR and thus occasionally violate FDR control. For this reason we can construct a resampling procedure that performs the step calibration procedure over randomly generated bootstrap samples instead of the true sample. This bootstrap procedure proceeds by generating a specified number of bootstrap samples using the labeled resampling procedure from 2.4.2 (although we don’t need to retain the labels here). We then proceed through step calibration, calculating the discovery lists for each bootstrap sample with the modified condition that the assignment of each group is decided by which of competition or calibration most commonly has the largest symmetric difference between its discovery list and the previously accepted list. Unfortunately, while this procedure does obtain improved levels of FDR control (with a corresponding slight cost to power), it also requires significant computational resources. As we will see in the following sections, often calibration offers no major improvements and thus the extra work is not generally rewarded.
3.3 Simulations

We now examine how our proposed calibration methods perform on simulated data. We construct these simulations in the same manner as in Section 2.5, that is, we assign a normal distribution $N(\mu, \sigma_i^2)$ to each null hypothesis $H_i$ and simulate a target score (if $H_i$ corresponds to a true null) and $d$ decoy scores. If $H_i$ corresponds to a false null we instead sample the target score from a shifted normal $N(\mu_i + \gamma_i, \sigma_i^2)$.

As before, the parameters $\mu_i, \sigma_i^2$ and $\gamma_i$ are themselves sampled with each set of scores. $\mu_i$ is sampled from a normal distribution $N(\mu, \sigma^2)$ where $\mu = 0$ and $\sigma^2$ is set depending on the experiment. Both $\sigma_i^2$ and $\gamma_i$ are sampled exactly as in 2.5, and again we set the hyper parameter $\nu$ that controls the distribution of $\gamma_i$ differently for each experiment.

As we are testing how our calibration procedures work we examine both poorly calibrated data, corresponding to $\sigma^2 = 5$ and well (though not perfectly) calibrated data with $\sigma^2 = 0.5$. For each case we draw 3K sets of target and decoy scores for each of the following 32 combinations of $m, k, \nu$ and $d$:

- $m = 1000$,
- $k = \{100, 500\}$
- $\nu = \{0.1, 0.5\}$
- $d = \{3, 9, 17, 33\}$.

To this generated data we apply a range of different combinations of calibration and testing methods. Specifically, we test:

- Mirror with the following calibration procedures:
  - no calibration,
  - $c$-calibration (equivalent to calibrated TDC),
  - step-calibration.
- LF calibrated using:
  - no calibration,
Chapter 3. Calibration

- c-calibration
- step-calibration.

- LBM calibrated using:
  - no calibration,
  - step-calibration.

We apply these methods for each \( \alpha \) in a subset of \( \Phi \) which we denote \( \Phi_1 \). Specifically, we set \( \Phi_1 \) as each value of \( \alpha \) from 0.001 to 0.01 by jumps of 0.003, from 0.02 to 0.3 by jumps of 0.01, from 0.35 to 0.5 by jumps of 0.05 and from 0.5 to 0.9 by jumps of 0.1 with the addition of 0.95. For each experiment we obtain an estimate of the FDR and power for each significance level by averaging the FDP (the proportion of true nulls in the discovery list) and the percentage of false nulls reported respectively for each of the 3K runs.

We begin by examining the FDR control of our fixed parameter selection methods after both c-calibration and step-calibration. Figure 3.1 (panels A and B) shows that c-calibration appears to maintain control of the FDR as we expected. On the other hand, step-calibration (panels C and D) shows some violations, especially with the LF procedures. While some of the violations are quite significant (the largest violation was 10% over \( \alpha \)), we should keep in mind that these are only the average of 3000 runs, thus it is difficult to conclude if step-calibration truly violates the FDR when applied with the fixed parameter methods.

When it comes to power we see two major points from Figure 3.2. First is that, as expected, c-calibration can be sub-optimal for the fixed parameter selection procedures (panels A and C). Particularly, for the mirror method there are many cases where we can lose up to 10% power by calibrating. While this does appear to vanish when we use step-calibration (panels B and D), when we compare the calibrated fixed procedures to LBM then, even when LBM is uncalibrated, we see that they consistently provide weaker power. Thus, while calibration can improve the results of our fixed methods, it is difficult to recommend over simply using LBM.

Lastly, we consider our calibration methods applied to our general recommended procedure of LBM. From our experiments it appears that calibration does not impact
Figure 3.1: FDR control The panels show the ratio of the empirical FDR to the selected FDR threshold, and each is made of 32 curves, each of which corresponds to one experiment involving 3K randomly drawn sets. The empirical FDR is the 3K-sample average of the FDP of each method’s discovery list at the selected FDR threshold. The 3K sets were drawn simulating scores using the experiment-specific parameter combination.
Figure 3.2: Power comparison. Each of the panels show the difference in the average power of the two methods compared (positive values indicate the first method is more powerful). Each panel is made of 32 curves, each of which shows the difference in power averaged over the 3K sets. The sets were drawn simulating scores using the experiment-specific parameter combination. The power of each method is the 3K-average percentage of false nulls that are discovered at the given FDR threshold. Note that figures' y-axes are on different scales.
Figure 3.3: LBM with calibration. The panels show the performance of calibration for the general recommended procedures of LBM. Each figure is made of 32 curves, each of which corresponds to one experiment involving 3K randomly drawn sets. The empirical FDR is the 3K-sample average of the FDP of each method’s discovery list at the selected FDR threshold. The 3K sets were drawn simulating scores using the experiment-specific parameter combination. (B) FDR of step-calibrated LBM (panel (A) shows uncalibrated LBM for reference). (C) Power comparison between uncalibrated LBM and step-calibrated LBM. Note the small scale on the y-axis.

greatly on the FDR control of this procedure as shown by panel B Figure 3.3 (panel A shows the FDR of uncalibrated LBM for reference) in these cases. However, when we consider the power (panel C) we notice that not only is the calibrated procedure often losing, but generally speaking there is not much difference between the two, with a typical run only producing approximately 1–2% power difference. When we consider the large computational cost that comes with step-calibration these results are quite poor.

3.4 Real Data Experiment - Peptide Detection

Despite the disappointing simulation results we applied our method to one of the peptide identification datasets from Section 2.6.1 to observe how they perform on a real dataset. Specifically, we examine the yeast dataset and apply our methods using the same setup as in Section 2.6.1. Specifically, we examine $d = \{1, 3, 5, 7, 9\}$ and $\alpha = \{0.01, 0.05, 0.1\}$. 

A: FDR - uncalibrated LBM

B: FDR - step-calibrated LBM

C: Power - uncalibrated LBM vs step-calibrated LBM
We examine the following procedures:

- Single decoy TDC.

- Mirror with the following calibration procedures
  - no calibration
  - c-calibration (essentially calibrated TDC)
  - step-calibration.

- LBM with the following calibration procedures
  - no calibration
  - step-calibration.

Figure 3.4 shows the results from this experiment match the simulated data, that calibration offers very little over simply using all the decoys for competition with our recommended methods. As in Section 2.6.1, for $\alpha = 0.05$ and 0.1 we see there is little difference in the average number of discoveries between our any of the specific testing
methods and calibration does not appear to change that with typically a difference of under 1% in the number of discoveries between the methods.

More interestingly, for $\alpha = 0.01$ we see that while calibration can sometimes improve the mirror method at no point does it become comparable to LBM (panel A) with the uncalibrated LBM still obtaining around 40% more power than the various mirror procedures. Additionally, a wrong choice of calibration method can occasionally lead to a loss of power, and moreover which calibration method gives a loss changes with the experiment. For example, for $d = 3$ we observe that step-calibration is superior to no calibration for the mirror method with a difference of 12% in power (79.99 discoveries for step calibration vs 71.40 for the uncalibrated mirror), however for $d = 7$ we see a reversal and the uncalibrated mirror outperforms the step-calibrated version by 10% (73.92 step vs 81.98 mirror).

Meanwhile, matching with the simulations, calibration doesn’t appear to improve the results of LBM by any meaningful amount (panel B) with differences of around 2 discoveries at most, a deviance of less than 2% power. In fact, at this level of $\alpha$ we often see worse results by diverting decoys to calibrate our scores with uncalibrated LBM outperforming step-calibration in all cases except $d = 5$, in which there is a $< 0.3\%$ difference (111.57 for uncalibrated LBM vs 111.90 for the calibrated version).

### 3.5 Discussion

In this chapter we have developed two procedures for calibrating the scores before applying multiple competition. Specifically, we have investigated how to split $d$ decoys amongst calibration and competition in order to achieve the best power with our various multiple competition procedures using both a pre-selected, fixed split with a focus on maintaining FDR control and a more sophisticated data-driven split.

Unfortunately, our simulations and peptide experiment show that calibration provides very mixed results. Applying our fixed split procedures to the multiple competition methods that guarantee FDR control (mirror, LF) we observe that sometimes they perform very well, offering a route to obtaining significant gains in power without sacrificing FDR control. However, in other experiments we see the opposite and diverting
decoys to calibrate can result in significant losses of power.

Meanwhile, when we consider applying a data-driven method to select the optimal split, though we no longer see cases where calibration can lead to significant power drops with the fixed competition procedures, we lose the guarantee of FDR control. Moreover, when we compare the calibrated mirror and LF methods to the recommended testing procedure of LBM we notice that, even when it is uncalibrated, LBM generally provides superior results to these methods regardless of the calibration procedure used.

Finally, when we apply our calibration assignment procedures with LBM as a testing method we generally see no improvement in power by diverting decoys from competition to calibration. From our experiments we observe that often there is only a difference of $1 - 2\%$ power between the step-calibrated LBM and the uncalibrated version. Additionally, we occasionally see a loss in power, especially for extremely low values of $\alpha$. From these experiments it appears as though calibration offers little when used with our data-driven selections of $c$ and $\lambda$. In fact, we can observe that in our FDS-based procedures we are utilizing some level of calibration already through the use of the empirical p-values.

Despite these results, it is not to say that calibration has no use. As stated at the start of this chapter, in the single decoy case Keich and Noble, 2017b were able to show that calibration is able to provide significant power gains. In the same vein, it is intuitively clear that if, more generally, one fixes the number of competing decoys, or alternatively, has so many that diverting some has little impact on competition, then applying calibration to our scores will improve our results. As an example, consider a case in which we have infinite decoys (we will examine this setting in more detail in the next chapter), then despite essentially having access to the precise null distribution of the hypotheses, without calibration we would still order the $W$ scores of each hypothesis using the raw scores, a procedure that is clearly inferior to ordering the (perfectly) calibrated scores.

In our setting of a fixed, small $d$ however, we must conclude that it is difficult to gain consistent benefits from calibration within the framework of multiple competition. Indeed, when we consider the (possibly very large) computational cost that calibration
can bring, it is difficult to recommend diverting decoys towards calibrating our scores instead of simply proceeding with pure competition using our recommended data-driven selection procedures.
Chapter 4

Considering the limit as $d \to \infty$

Until now we have only considered a small number of decoys due to practical limits on computational power as outlined in Chapter 1. However, how our multiple competition procedures behave as $d$ becomes large is an interesting question that can allow us to better understand our methodology. In this chapter we will examine this question by investigating the behavior of our proposed multiple competition procedures as we take the limit $d \to \infty$.

Our first step is to translate our methods to this new limiting setting. We can note that with infinite decoys we can perfectly calibrate our scores with no cost and thus it makes sense to translate our scores into p-values. We should keep in mind however, that in using p-values smaller scores represent more significant results which is the opposite of how we presented the problem above.

4.1 Examining Multiple Competition as $d \to \infty$

4.1.1 Limiting Mirandom

We start by adapting our general competition framework with the mirandom mapping scheme to this p-value based infinite setting for any given $0 < c \leq \lambda < 1$. Recall that in the finite setting we had a target win ($L_i = 1$) if $Z_i$ was in the top $c \cdot d_1$ ranks of the target-decoy score set, while we had a decoy win ($L_i = -1$) in the event that $Z_i$ was in the bottom $(1 - \lambda) \cdot d_1$ ranks. Now with p-values we can encode this competition directly into our scores by translating our conditions as: $L_i = 1$ if $p_i \leq c$ and $L_i = -1$ if $p_i > \lambda$ with $L_i = 0$ otherwise. Once these labels have been determined then, as before, if $L_i = 1$ we retain the original score and set $W_i = p_i$. If $L_i = -1$ then we set $W_i$ equal.
to the decoy score chosen by the mirandom map $\varphi_{md}$ as stated in Section 2.2.4 which, in our limiting case, is equivalent to setting $W_i = (1 - p_i)(1 - \lambda)$. To summarize:

$$\begin{align*}
(W_i, L_i) &= \begin{cases} 
(p_i, 1) & p_i \leq c \\
((1 - p_i)(1 - \lambda), -1) & p_i > \lambda \\
(*) , 0 & \text{otherwise}
\end{cases} \\
(4.1)
\end{align*}$$

Sorting the scores $W_i$ in order of significance (remembering we are using p-values so smaller scores are better here) $W_{(1)} \leq W_{(2)} \leq \cdots \leq W_{(m)}$, we note that, for any $i$ with $W_{(i)} = p_{(i)} \leq c$, when we estimate the value of $i_{ac\lambda}$ (the largest rejection threshold such that $FDR_{i_{ac\lambda}} \leq \alpha$) with (2.17) the denominator term $\# \{ j \leq i : L_{(j)} = 1 \}$ is the number of target scores greater than the current threshold, or equivalently the number of p-values $p_j \leq p_{(i)}$. At the same time, for the same $i$ and $j \leq i$, $L_{(j)} = -1$ if and only if $p_{(j)} > \lambda$ and $W_{(j)} < W_{(i)} = p_{(i)} \leq c$ so we have for the numerator term

$$\# \{ j \leq i : L_{(j)} = -1 \} = \# \{ j : p_j > \lambda, W_j < p_{(i)} \} = \# \{ j : p_j > 1 - \frac{1 - \lambda}{c} p_{(i)} \}.$$  

Considering that $i_{ac\lambda} < m$ in (2.17) must be attained at a value of $i$ for which $W_i = p_i \leq c$ (original win), we can essentially rewrite (2.17) as

$$i_{ac\lambda} = \max \left\{ i : \frac{1 + \# \{ j : p_j > 1 - \frac{1 - \lambda}{c} p_{(i)} \}}{\# \{ j : p_j \leq p_{(i)} \} \lor 1} \cdot \frac{c}{1 - \lambda} \leq \alpha \right\}. \hspace{1cm} (4.2)$$

We notice that this takes a form very similar to the threshold $t_{\alpha}$ in Storey’s FDR controlling criteria. To illuminate this connection we recall the more rigorous estimate (2.20) which essentially amounts to

$$t_{\alpha} = \max \left\{ t \in [0, \lambda^*] : \frac{1 + \# \{ j : p_j > \lambda^* \}}{\# \{ j : p_j \leq t \} \lor 1} \cdot \frac{t}{1 - \lambda^*} \leq \alpha \right\},$$

\footnote{Note that $p_{(i)}$ and $L_{(i)}$ here do not refer to the $i$th order statistic of $p_i$ or $L_i$, rather $(i)$ is the index of the $i$th smallest $W_i$.}
where $\lambda^*$ is a tuning parameter that plays a similar role to our $\lambda$. Considering the cases where $t_\alpha \leq c$ and setting $\lambda^*(t) := 1 - (1 - \lambda) t/c$, Storey’s threshold $t_\alpha$ becomes

$$t_\alpha = \max \left\{ t \in [0, c] : \frac{1 + \# \{ j : p_j > 1 - \frac{1-\lambda}{c} t \}}{\# \{ j : p_j \leq t \} \lor 1} \cdot \frac{c}{1-\lambda} \leq \alpha \right\}.$$  

Since in practice $t_\alpha$ will usually be taken as equal to one of the $p(i)$ the equivalence with (4.2) becomes obvious by identifying $t$ above with $p(i)$.

Thus, for example, as $d \to \infty$ the mirror method ($\lambda = c = 1/2$) converges to Storey’s procedure using $\lambda^*(t) := 1 - t$, which coincides with the “mirroring method” of Xia et al., 2017. It is worth noting that the general setting $\lambda^*(t) := 1 - (1 - \lambda) t/c$ is not obviously supported by the finite sample theory of Storey, Taylor, and Siegmund, 2004, however it can be justified by noting the above equivalence and our results from the finite $d$ methodology.

It should be noted that an equivalence between multiple competition and Storey’s procedure is not restricted to just the random map. Other mapping schemes of ours can have similar relationships. For example, if one defines $s_i$ by the random map $\varphi_u$ (briefly proposed in Section 2.2.3) then the rejection threshold index $i_{ac\lambda}(2.17)$ will be based on the FDR estimate

$$\widehat{F}_u(t = p(i); \lambda, c) = \frac{1 + D_{\lambda,c}(p(i))}{\# \{ j : p_j \leq p(i) \} \lor 1} \cdot \frac{c}{1-\lambda},$$

where $D_{\lambda,c}(p(i))$ is the (random) number of decoy wins that are mapped into $[0, p(i)]$.

Meanwhile, examining Storey’s finite-sample procedure, we observe that the rejection threshold $t_\alpha$ is determined by the corresponding FDR estimate

$$\widehat{F}_{St}(t = p(i); \lambda^* = \lambda) = \frac{1 + \# \{ j : p_j > \lambda \}}{\# \{ j : p_j \leq p(i) \} \lor 1} \cdot \frac{p(i)}{1-\lambda}.$$  

Taking the conditional expectation of $\widehat{F}_u$ with respect to $F = \sigma \{ p_j \leq p(i) \}$, the $\sigma$-algebra generated by all $p_j \leq p(i)$, and recalling that $p(i) \leq c$, we have

$$E(\widehat{F}_u | F) = \frac{1 + \# \{ j : p_j > \lambda \} p(i)/c}{\# \{ j : p_j \leq p(i) \} \lor 1} \cdot \frac{c}{1-\lambda} \geq \frac{1 + \# \{ j : p_j > \lambda \}}{\# \{ j : p_j \leq p(i) \} \lor 1} \cdot \frac{p(i)}{1-\lambda},$$
which shows that on average the resulting FDR estimate obtained by multiple competition using the random map $\varphi_u$ is greater than or equal to the corresponding estimate of Storey. This in turn suggests that the random map will on average be weaker than Storey’s finite sample procedure for any given set of p-values, again showing that this mapping scheme is not an ideal choice for use in multiple competition.

4.1.2 Limiting FDS

We have now shown that the limit of mirandom is equivalent to Storey with a particular setting of $\lambda^\ast$. However, we can obtain an even deeper connection by selecting the values of $c$ and $\lambda$ according to the FDS procedure as $d \to \infty$. Indeed, using the same $\lambda$ as determined by the progressive interval splitting procedure described in Section 2.3.2, Storey’s finite sample procedure (2.21) would amount to setting the rejection threshold to

$$t_\alpha = \max \left\{ t \in [0, \lambda] : \frac{m \cdot \tilde{\pi}_0^*(\lambda) \cdot t}{R(t) \vee 1} \leq \alpha \right\}.$$ 

Recalling that $d_1 \to \infty$ we note that this $t_\alpha$ coincides with the value FDS assigns to $c$ via (2.24) and (2.25). Let $i_c$ be such that the above $t_\alpha = c \in [p(i_c), p(i_c+1))$ (recall we assume no ties here), then we can assume without loss of generality that $t_\alpha = c = p(i_c)$ and hence by comparing with (4.2) (note that $L_i = 1$ for all $i > i_c$ and hence, again without loss of generality, $i_{ac\lambda} \leq i_c$) we observe that applying multiple competition with the mirandom map will find

$$i_{ac\lambda} = \max \left\{ i \leq i_c : \frac{1 + \# \{ j : p_j > 1 - \frac{1 - \lambda}{c} p(i) \}}{\# \{ j : p_j \leq p(i) \} \vee 1} \cdot \frac{c}{1 - \lambda} \leq \alpha \right\}.$$ 

But

$$\frac{1 + \# \{ j : p_j > 1 - \frac{1 - \lambda}{c} p(i_c) \}}{\# \{ j : p_j \leq p(i_c) \} \vee 1} \cdot \frac{c}{1 - \lambda} = \frac{1 + \# \{ j : p_j > \lambda \}}{\# \{ j : p_j \leq c \} \vee 1} \cdot \frac{c}{1 - \lambda} = \frac{m \cdot \tilde{\pi}_0^*(\lambda) \cdot t_\alpha}{R(t_\alpha) \vee 1} \leq \alpha.$$ 

Hence, $i_c$ satisfies the required inequality and $i_{ac\lambda} = i_c$. It follows that the discovery lists of FDS and the above variant of Storey’s procedure with the same $\lambda$ coincide in the $d \to \infty$ limit.
Unfortunately FDS\(_1\) does not obtain a similarly nice relationship. The modifications in the calculation of \(c\) mean that our evaluations are no longer consistent between the \(c\)-selection and FDR estimation steps. Indeed, it is clear that even without considering the additional complications introduced by potentially varying \(\lambda\) between the two steps, by removing the +1 in the numerator of \(t_\alpha\) (by changing \(\hat{\pi}_0^*\) to \(\hat{\pi}_0\)) but not in the determination of \(i_{ac\lambda}\), we can no longer obtain the crucial equality of \(i_{ac\lambda} = i_c\) and hence the equivalence with Storey’s method.

Finally, just as FDS converges to Storey’s rigorous procedure, it is simple to propose a multiple competition variant that will converge to the more common asymptotic procedure for FDR control of Storey, Taylor, and Siegmund, 2004 through the following modifications:

- As with FDS\(_1\) we use Storey’s asymptotic estimator of the FDR to calculate \(c\).
- We allow for the case of \(c > \lambda\) but, unlike with FDS\(_1\), we do not set \(\lambda = c\) if this occurs. Instead, we retain \(\lambda\) and count some hypotheses (with \(\lambda < p_i \leq c\)) as both target and decoy wins. In this case we set \(W_i\) as if we had a target win, though for the purpose of FDR control the hypothesis is counted both in the numerator and denominator of (4.2) (i.e \(L_i = \pm 1\)).
- We do not have +1 in the numerator of the estimator of the FDR (4.2).

The third point is the reason we do not consider this method in any real depth in this work. As shown in Barber and Candès, 2015, without the +1 (corresponding to the knockoff variant) we are unable to control the FDR, instead we are only able to control a modified expression \(mFDR = E \left( \frac{V}{R + c/(1 - \lambda)\alpha^{-1}} \right) \leq \alpha\) (Barber and Candès, 2015, Lei and Fithian, 2016). One could present very similar arguments to those in Chapter 2 to show that performing multiple competition in this manner would control this mFDR (with the same requirements on \(c\) and \(\lambda\)), however as this work is focused primarily on FDR control we do not consider that here.
4.2 Simulations

In Section 2.5 we examined how our finite $d$ procedures performed on simulated data. The results from these simulations illuminated many trends amongst our various methods and thus we are interested in determining if these trends, and subsequently our recommended procedure, still hold in the limit as $d \to \infty$.

We first generated the observed scores using the same setup as in the finite decoy calibrated simulations with $\mu_i \equiv 0, \sigma_i \equiv 1, \gamma_i \equiv \gamma$. Calibrated data is used since, as we mentioned at the beginning of this chapter, we are able to perfectly calibrate any scores with no extra cost. Next, consistently with our infinite decoy formulation, instead of generating decoy scores we transformed the observed scores into p-values using their corresponding null distributions. We examined the following subset of parameters from Section 2.5:

- $k = 100$ or $1000$.
- $m = a \cdot k$ for $a \in \{1.25, 2, 5, 10, 20\}$.
- Additionally, we checked the $(m, k)$ pairs $(100, 10)$ and $(12500, 10000)$.
- $\gamma = 1$ or $2$.

We applied the following procedures to the above combinations of parameters:

- Storey’s procedure as provided in the qvalue package with $\lambda$ determined by the bootstrap option ($\lambda_b$).
- Storey’s procedure with a finite correction and $\lambda$ determined by the progressive interval splitting procedure outlined in Section 2.3.2 ($\lambda_s$). Here we chose the value of $\lambda$ by applying our procedure to the lattice defined by the points $i/M$ (we used $M = 20$). This method is identical to FDS with infinite decoys ($\text{FDS}(\infty)$).
- The infinite decoy variant of LBM, which we denote LBM($\infty$), as introduced in 2.4 which uses monitored FDR control to choose between FDS($\infty$) (formulated as above), mirror($\infty$) and FDS$_1(\infty)$ (in order of priority) with the fallback method of FDS$_1(\infty)$, the infinite decoy FDS$_1$. 
• Mirror($\infty$), the infinite decoy version of the mirror method ($c = \lambda = 0.5$).

As before, we performed 10K iterations of these experiments on $\Phi_1$ recorded the FDP and the percentage of true discoveries for each run at every significance level and averaged over all the runs to obtain values for the empirical FDR and the power for each experiment.

First examining the empirical FDR, we see that Storey’s popular bootstrap method ($\lambda_b$) often fails to control the error in these experiments (Figure 4.1, panel D). For example, when $m = 100$, $k = 10$ and $\gamma = 2$ Storey’s procedure obtained an FDR value of 0.225 at $\alpha = 0.2$, which is a violation of over 12.5%. Even at smaller, more practical levels of $\alpha$ we can see violations with this method obtaining an empirical FDR value of 0.054 when $\alpha = 0.05$, a violation of 8%. This is not an unexpected result however, as both Storey’s procedure (without the correction for finite $m$) and the accompanying bootstrap approach for $\lambda$ are claimed to only obtain asymptotic FDR control.

For the other methods we initially see that the mirror($\infty$) method (panel A) appears to control the FDR across all experiments as expected. Both LBM($\infty$) and Storey using $\lambda_s$ (panels B and C) see some cases of large violations at very small levels of $\alpha$ ($\alpha = 0.001$ precisely). These violations can be up to 55% over the threshold when $m = 20,000$, $k = 1000$, $\gamma = 1$, though practically this only corresponds to an empirical FDR of 0.00155 produced by just 15 cases of a single false discovery only (where FDP = 1) and one case of both a single false and true discovery (FDP = 1/2), in 10,000 runs (for reference, in this experiment we observe 84 cases of a single true discovery, two cases of two true discoveries and the aforementioned run with both a false and true discovery, with the remaining 99.9K runs returning 0 discoveries). Outside this small value of $\alpha$ we see that these two methods obtain much better control of the FDR than Storey’s bootstrap method and indeed have only minor or no violations for most other significance levels.

When we consider the power of our multiple competition procedures (4.2) we see the results that we expected from our finite sample simulations:

• Mirror($\infty$) is often obtaining more power than FDS($\infty$) (panel A), which matches our finite decoy simulations as we are dealing with calibrated data.

---

2Recall from Chapter 3 that we define $\Phi_1$ as each value of $\alpha$ from 0.001 to 0.01 by jumps of 0.003, from 0.02 to 0.3 by jumps of 0.01, from 0.35 to 0.5 by jumps of 0.05 and from 0.5 to 0.9 by jumps of 0.1 with the addition of 0.95
Chapter 4. Considering the limit as $d \to \infty$

**Figure 4.1: FDR control** The panels show the ratio of the empirical FDR to the selected FDR threshold, and each is made of 20 curves, each of which corresponds to one experiment involving 10K randomly drawn sets of scores. The empirical FDR is the 10K-sample average of the FDP of each method’s discovery list at the selected FDR threshold. The 10K sets were drawn simulating p-values using the experiment-specific parameter combination. Note the different scales on the y-axis for each panel.

- LBM($\infty$) consistently does better than both the mirror and FDS($\infty$) (panels B and C) reinforcing that LBM remains a good choice as $d$ increases.

Overall, Storey’s bootstrap procedure appears to give the best power, which is not an unexpected result as it has less stringent FDR control. But it should be noted that there are some cases where Storey’s method can lose to both the LBM($\infty$) and even the infinite mirror method (panels D, E and F). For example, when $m = 1250$, $k = 1000$, $\gamma = 1$, and $\alpha \in [0.01, 0.1]$ the mirror method enjoys an increase of up to 5% in total power compared with Storey with $\lambda_b$; for $\alpha = 0.05$ the mirror method gives 19.2% power compared to Storey($\lambda_b$) that attains 14.2% power (for reference, Storey($\lambda_s$)/FDS($\infty$) obtains 13.7% power in this example with a low $\pi_0$).

Overall, while the practical utility of the proposed procedures is lessened in the infinite case due to the large range of alternative methods that are explicitly designed to utilize p-values, it is reassuring that the behavior of our methods does not appear to change as $d$ increases. Additionally, if one requires stricter control of the FDR, beyond what Storey’s bootstrap procedure gives, then LBM($\infty$) is a reasonable choice of
Chapter 4. Considering the limit as $d \to \infty$

A: Storey $\lambda_s$ vs. mirror($\infty$)

B: mirror($\infty$) vs. LBM($\infty$)

C: Storey $\lambda_s$ vs. LBM($\infty$)

D: Storey $\lambda_s$ vs. Storey $\lambda_b$

E: Storey $\lambda_b$ vs. mirror($\infty$)

F: Storey $\lambda_b$ vs. LBM($\infty$)

Figure 4.2: Power comparison. Each of the panels show the difference in the average power of the two methods compared (positive values indicate the first method is more powerful). Each panel is made of 20 curves, each of which shows the difference in power averaged over the 10K sets. The sets were drawn simulating scores using the experiment-specific parameter combination which were then transformed into p-values. The power of each method is the 10K-average percentage of false nulls that are discovered at the given FDR threshold. Note that figures’ y-axes are on different scales.
method that can improve power beyond simply using Storey’s finite correction.
Chapter 5

Multiple Competition with Knockoffs

As mentioned in Chapter 1, competition-based FDR control also sees use in the setting of variable selection. In this chapter we will examine how to utilize our new multiple competition methods within the framework of the knockoff selection procedure of Barber and Candès, 2015 in order to apply our procedures in this new setting. Recalling our problem description from Section 1.3.2, suppose we have an observed response vector $y \in \mathbb{R}^n$ that satisfies the classical linear regression model:

$$y = X\beta + \epsilon,$$  \hspace{1cm} (5.1)

where $X$ is the $n \times p$ known, real-valued, design matrix, $\beta \in \mathbb{R}^p$ is the unknown vector of coefficients, and $\epsilon \sim N(0, \sigma^2 I)$ is Gaussian noise. We seek to understand which features $X_i$ (corresponding to columns of $X$) are truly in the model, or more specifically, which $\beta_i = 0$. In this setting FDR control refers to controlling the error amongst the proposed features to be included in the model.

As was discussed in Chapter 1, Barber and Candès developed a competition-based procedure that uses the same methodology as TDC to control the FDR using a set of knockoffs (the equivalent to decoys in this setting). Unfortunately, generating these knockoff scores is much more complex than it was in our peptide ID setting, and much of the work we propose in this chapter will be focused on this construction.
5.1 Knockoff+

As in the TDC case, we begin by reviewing Barber and Candès’ single-knockoff procedure. We can essentially split this procedure up into two separate parts: constructing the knockoff matrix \( \tilde{X} \), and subsequently the scores \( Z_i \) with their knockoff copies \( \tilde{Z}_i \), and controlling the FDR through the competition between \( Z_i \) and \( \tilde{Z}_i \).

The aim of Barber and Candès’ knockoff construction is to ensure that the correlation (technically, inner product) between any two distinct original features remains unchanged if we replace one or both of those with their knockoff copies. Thus, in terms of the underlying regression problem each truly null feature (\( \beta_j = 0 \)) is statistically indistinguishable from its knockoff. At the same time, their construction tries to minimize the correlation between each feature and its knockoff so that true variables (\( \beta_j \neq 0 \)) are not too similar to their copies, lest the procedure’s power would be compromised.

Specifically, given the Gram matrix \( \Sigma = X^TX \), Barber and Candès define their set of knockoff features \( \tilde{X} \) through requiring that \( \tilde{X}^T\tilde{X} = \Sigma \) and \( X^T\tilde{X} = \Sigma_0 \), where \( \Sigma_0 := \Sigma - \text{diag}(s) \), and \( s \) is a non-negative vector that will be specified below. That is, the Gram matrix of the \( n \times 2p \) dimensional augmented design matrix \( [X \tilde{X}] \) satisfies

\[
[X \tilde{X}]^T [X \tilde{X}] = \begin{bmatrix} \Sigma & \Sigma_0 \\ \Sigma_0 & \Sigma \end{bmatrix} =: G.
\]

Barber and Candès show that these latter equations can be solved if and only if the vector \( s \) is chosen so that the above defined \( G \) is a non-negative definite matrix (\( G \succeq 0 \)).

Considering a constant vector \( s = s_0 \), we can minimize \((1 - s_0)\), the correlation between each feature and its knockoff, by maximizing \( s_0 \) subject to the constraint that \( G \succeq 0 \). Barber and Candès’ equi-correlated construction shows this maximization can be achieved if we choose \( s_0 = 2\lambda_{\min}(\Sigma) \wedge 1 \), where \( \lambda_{\min}(\Sigma) \) is the minimal eigenvalue of \( \Sigma \). They then explicitly define a set of knockoff variables that satisfy the above equations (Equation (2.2) in Barber and Candès, 2015):

\[
\tilde{X} = X \left( I - \Sigma^{-1}\text{diag}(s) \right) + \tilde{U} C, \tag{5.2}
\]

\(^1\)We adopt the same convention of Barber and Candès, 2015 that the columns of \( X \) are normalized so \( \text{diag}(\Sigma) \equiv 1 \).
where $\tilde{U} \in \mathbb{R}^{n \times p}$ is an orthonormal matrix whose column space is orthogonal to that of $X$, and $C^T C = 2\text{diag}(s) - \text{diag}(s)\Sigma^{-1}\text{diag}(s)$.

Once $\tilde{X}$ has been calculated we proceed by obtaining a score for each observed variable and its knockoff copy, which we denote $Z_i$ and $\tilde{Z}_i$ respectively. These scores should satisfy the conditional exchangeability property (definition 2 in Section 2.1) in order for FDR control to be maintained. While Barber and Candès state that there are many different statistics that could be used to construct the $Z$-scores in their paper, they focus on utilizing the Lasso to generate $Z_i$ and $\tilde{Z}_i$, which we will likewise follow in this work. Specifically, $Z_i$ and $\tilde{Z}_i$ are taken to be the value of the point $\lambda$ on the Lasso path (Tibshirani, 1996) at which feature $X_i$, or respectively, its knockoff competition $\tilde{X}_i$, first enters the model when regressing the response $y$ on the augmented design matrix $[X \tilde{X}]$.

Once $Z_i$ and $\tilde{Z}_i$ are calculated such that they satisfy the conditional null exchangeability property, FDR control proceeds exactly as it did in the TDC framework of Section 2.1. $W_i$ and $L_i$ are determined based on a direct competition between $Z_i$ and $\tilde{Z}_i$ using procedure outlined in Section 2.1. That is, the discovery list (the list of variables for which we suspect $\beta_i = 0$) is $D_B(\alpha) := \{i : i \leq i_\alpha, L_i = 1\}$ where $i_\alpha$ is given by

$$i_\alpha = \max \left\{ i : \frac{1 + \# \{j \leq i : L_j = -1\}}{\# \{j \leq i : L_j = 1\}} \vee 1 \leq \alpha \right\}.$$ 

As mentioned in Chapter 2, Barber and Candès prove that this procedure rigorously controls the FDR in the finite sample case by showing that it is a special case of their Seq Step+ procedure.

### 5.2 Multiple Knockoffs

We now examine how to extend this procedure to construct and utilize $d$ knockoffs instead of 1. As before we will consider this in two parts, a knockoff construction procedure that generates $Z_i$ and $\tilde{Z}_i^k$ for $k = 1, \ldots, d$ in a way that satisfies the conditional null exchangeability property, and a FDR controlling procedure that will reiterate the

---

2In Barber and Candès’ original formulation they combined $W$ and $L$ into one statistic using the sign of $W$ as the win/loss indicator (which is feasible as $W > 0$ in this setting). Here we keep them separate to maintain consistency with Chapter 2.
work in Chapter 2 by using the $Z$ scores and their knockoffs to obtain a discovery list with guaranteed control of the FDR.

### 5.2.1 Knockoff Construction

#### Creating the Gram Matrix

We begin our construction procedure by first finding an augmented $d_1p \times d_1p$-dimensional (recall that $d_1 = d + 1$) Gram matrix $G$ and then finding an $n \times dp$-dimensional solution $\tilde{X}$ for the equation $[X \tilde{X}]^T [X \tilde{X}] = G$. Throughout this section we assume $n \geq d_1p$ (generalizing Barber and Candès’ assumption that $n \geq 2p$). We will discuss possible methods to relax this assumption below.

We first demonstrate our construction using $d = 2$ knockoffs per feature. The original matrix $G$ suggests the following $3p \times 3p$-dimensional augmented Gram matrix:

\[
G := \begin{bmatrix}
\Sigma & \Sigma_0 & \Sigma_0 \\
\Sigma_0 & \Sigma & \Sigma_0 \\
\Sigma_0 & \Sigma_0 & \Sigma
\end{bmatrix},
\]

where $\Sigma = X^T X$ and $\Sigma_0 := \Sigma - \text{diag}(s)$ as before. The idea is that now the knockoff matrix will be $\tilde{X} = \begin{bmatrix} \tilde{X}^1 \tilde{X}^2 \end{bmatrix}$, where each $\tilde{X}^i$ corresponds to one complete set of knockoff variables, so that each $\tilde{X}^i$ behaves exactly as a single set of Barber and Candès’ knockoffs. In addition, the correlations between the two sets of knockoffs are the same as between each one of them and the original design matrix $X$.

More generally, we define the $d_1p \times d_1p$-dimensional augmented Gram matrix as a $d_1 \times d_1$ block matrix, where each block is a $p \times p$ sub-matrix $B_{ij}$ with $B_{ii} = \Sigma$, and $B_{ij} = \Sigma_0$ for $i \neq j$. This corresponds to a knockoff matrix $\tilde{X} = \begin{bmatrix} \tilde{X}^1 \tilde{X}^2 \ldots \tilde{X}^d \end{bmatrix}$ that is made of $d$ blocks/copies $\tilde{X}^i$, $i = 1, \ldots, d$, with the same correlation structure as discussed for the $d = 2$ case above.

We will next show how to construct $\tilde{X}$ so that $G$ is indeed the Gram matrix of the augmented $n \times d_1p$ design matrix $[X \tilde{X}]$. However, we note that we can only do that if $G \succeq 0$, which in turn depends on $s$. Again, we consider the equi-correlated case of $s \equiv s_0$, but we can no longer use the same $s_0 = 2\lambda_{\text{min}}(\Sigma) \wedge 1$ that works for the $d = 1$
case. That said, we empirically found that setting

\[ s_0 = \frac{d_1}{d} \lambda_{\text{min}}(\Sigma) \wedge 1 \]  

(5.3)

yields the optimal result in the general case. That is, with this critical value, \( G \succeq 0 \), and if \( s_0 < 1 \) then \( G \) is also rank deficient so \( s_0 \) cannot be any larger than this value. Notably, this critical value, which generalizes Barber and Candès' expression for \( d = 1 \), decreases with \( d \)— a point we will return to below.

Creating the knockoff variables with the given Gram matrix

The original procedure (5.2) of deriving \( \tilde{X} \) from \( X \) is not clearly generalizable to our setting, so instead we offer the following alternative procedure.

We first find \( X_0 \), a \( d_1 p \times d_1 p \)-dimensional symmetric root of \( G \) so that \( X_0 X_0 = G \). Technically, we accomplish this by starting with a singular value decomposition (SVD) of \( G \): \( G = USV^T \), where \( S \) is a diagonal matrix and \( U, V \) are orthogonal matrices. Since \( G \) is symmetric, the SVD is in fact a spectral decomposition of \( G \): \( G = USU^T \), so we can define \( X_0 := U S^{1/2} U^T \).

Note that the Gram matrix of the first \( p \) columns of \( X_0 \) is the corresponding \( p \times p \) leading sub-matrix of \( G \), which is \( \Sigma \). Hence, assuming as we do that \( n \geq d_1 p \), there exists an orthogonal map \( \tilde{U} : \mathbb{R}^{d_1 p} \to \mathbb{R}^n \) that maps the first \( p \) columns of \( X_0 \) to \( X \). Specifically, we can find such a map by first doing a QR decomposition of \( X_0 \):

\[ X_0 = Q_0 R_0, \]

where \( Q_0 \) is a \( d_1 p \times d_1 p \) orthogonal matrix, and \( R_0 \) is an upper triangular matrix of the same dimension. We next find a thin QR decomposition (Golub and Van Loan, 1996) of

\[ [X A] = QR, \]  

(5.4)

where \( A \) is an arbitrary \( n \times d p \) matrix, \( Q \) is an \( n \times d_1 p \) matrix with orthonormal columns, and \( R \) is a \( d_1 p \times d_1 p \) upper triangular matrix. Subject to a sign normalization we discuss
below, the map $\tilde{U}$ we seek can be defined by the matrix

$$\tilde{U} := QQ_0^T.$$  \hfill (5.5)

Defining

$$X_1 := \tilde{U}X_0 = QQ_0^T X_0 = QR_0,$$  \hfill (5.6)

we note that $X_1$ is an $n \times d_1p$ matrix and

$$X_1^T X_1 = X_0^T Q_0 Q^T Q_0^T X_0 = X_0^T Q_0 Q_0^T X_0 = G.$$  

Moreover, because the Gram matrices of the columns of $X$ and of the first $p$ columns of $X_0$ are the same, and because the QR decomposition is essentially just the Gram-Schmidt procedure, it follows that the $p \times p$ leading minor of $R_0$ ($R_0(i,j)$ for $i,j \leq p$) agrees with $R$ up to row signs, which we can readily match by adjusting the signs of the corresponding columns of $Q$.

Thus, without loss of generality, the first $p$ columns of $X_1$ coincide with the original design matrix $X$, and the next $dp$ columns are our knockoff variables. In other words, $X_1$ is the augmented design matrix, where for each feature $i \in \{1, \ldots, p\}$ the $i$th column of $X_1$ corresponds to the original $n$ variables, and columns $i+jp$ for $j=1, \ldots, d$ are its $d$ knockoff copies.

**The knockoff scores and conditional null exchangeability**

We can now describe the (first version of) our procedure for constructing multiple-knockoff scores. Assuming $n \geq d_1p$, the procedure constructs the $n \times d_1p$ augmented design matrix $\begin{bmatrix} X \tilde{X} \end{bmatrix}$ as described above. Following Barber and Candès’ knockoff+ procedure, it then applies the Lasso procedure to $y$ and $\begin{bmatrix} X \tilde{X} \end{bmatrix}$ to generate the set of scores $\{\tilde{Z}_i^0 := Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^d\}$ for each feature $i \in \{1, \ldots, p\}$. As before, each value is the point $\lambda$ on the Lasso path at which the corresponding variable, the original $X_i$ or its $d$ knockoffs $\tilde{X}_i^j$, $j=1, \ldots, d$, first enters the model.

We next show that our procedure creates knockoff scores that satisfy the conditional null exchangeability property and hence applying our multiple competition procedure
with any pre-determined values of the tuning parameters \( (c, \lambda) \) and the mirandom map \( \varphi_{\text{md}} \) controls the FDR in the finite variable selection setting.\(^3\)

**Definition 3.** Let \( \Pi_{d_i} \) denote the set of all permutations on \( \{1, 2, \ldots, d_i\} \) and let \( N \subset \{1, 2, \ldots, p\} \) be the indices of the true null features. A sequence of permutations \( \Pi = (\pi_1, \ldots, \pi_p) \) with \( \pi_i \in \Pi_{d_i} \) is a null-only sequence if \( \pi_i = \text{Id} \) (the identity permutation) for all \( i \notin N \).

**Theorem 2.** Suppose \( y \) is generated according to the linear model (5.1) with a given \( n \times p \) design matrix \( X \) with \( n \geq d_1 p \). Let \( V_i = (\tilde{Z}^0_i, \tilde{Z}^1_i, \ldots, \tilde{Z}^d_i) \), where \( \tilde{Z}^0_i \) is the \( i \)th original feature score and \( \tilde{Z}^1_i, \ldots, \tilde{Z}^d_i \) are its corresponding \( d \) knockoff scores defined above. For \( \pi \in \Pi_{d_i} \) let \( V_i \circ \pi := (\tilde{Z}^{\pi(1)}_i, \ldots, \tilde{Z}^{\pi(d_i)}_i) \), i.e., the permutation \( \pi \) is applied to the indices of the vector \( V_i \) rearranging the order of its entries. Then for any null-only sequence of permutations \( \Pi = (\pi_1, \ldots, \pi_p) \), the joint distribution of \( V_1 \circ \pi_1, \ldots, V_p \circ \pi_p \) is invariant of \( \pi_1, \ldots, \pi_p \).

Note that (a) the conclusion of the theorem is exactly the conditional null exchangeability of Definition 2 and (b) that the joint distribution is the one induced by the Gaussian noise \( \epsilon \) in our linear model (the design matrix \( X \) is fixed).

**Proof.** The proof of the theorem uses claims analogous to Lemmas 1, 2 and 3 of Barber and Candès, 2015. Denote by \( \hat{X} = \begin{bmatrix} X & \hat{X} \end{bmatrix} \) the above \( n \times d_1 p \) augmented design matrix so that \( G = \hat{X}^T \hat{X} \), and by \( \hat{X}(i) \) its \( i \)th column, so for \( i \in \{1, \ldots, p\} \) the columns \( \hat{X}(i), \hat{X}(i + p), \ldots, \hat{X}(i + dp) \) correspond to the \( i \)th feature and its \( d \) knockoffs.

For a null-only sequence of permutations \( \Pi = (\pi_1, \ldots, \pi_p) \) let \( \hat{X} \circ \Pi \) denote the \( n \times d_1 p \) matrix whose \( i \)th column for any \( i = i_0 + i_1 \cdot p \), where \( i_0 \in \{1, \ldots, p\} \) and \( i_1 \in \{0, 1, \ldots, d\} \), is given by

\[
(\hat{X} \circ \Pi)(i) := \hat{X}(i_0 + \pi'_{i_0}(i_1) \cdot p),
\]

where \( \pi'_{i_0}(i_1) = \pi_{i_0}(i_1 + 1) - 1 \) (note that \( i_0 = (i - 1) \mod p + 1 \) and \( i_1 = (i - i_0)/p \)). In words, the permutation \( \pi_{i_0} \) is applied to reorder the columns \( i_0, i_0 + p, \ldots, i_0 + dp \) of \( \hat{X} \) so their new order is \( \pi_{i_0}(1), \ldots, \pi_{i_0}(d_1) \).

\(^3\)Recall that in this setting the number of hypotheses here is \( m = p \), the number of features considered.
The first of our claims generalizes Lemma 2 of Barber and Candès: applying as above any sequence of permutations (not necessarily null-only) \( \Pi = (\pi_1, \ldots, \pi_p) \) to the columns of the augmented design matrix does not change the correlations between its columns.

**Claim 4.** \( (\hat{X} \circ \Pi)^T (\hat{X} \circ \Pi) = \hat{X}^T \hat{X} = G \). 

**Proof.** Let \( i = i_0 + i_1 p \) and \( j = j_0 + j_1 p \), where, as above, \( i_0, j_0 \in \{1, \ldots, p\} \) and \( i_1, j_1 \in \{0,1,\ldots,d\} \). Then, with \( \Sigma = (\sigma_{ij}) \), and \( \delta_{ij} \) the Kronecker delta we have

\[
\hat{X}(i)^T \hat{X}(j) = G_{i,j} = G_{i_0 + i_1 p, j_0 + j_1 p} = \\
= \sigma_{i_0 j_0} - \delta_{i_0 j_0}(1 - \delta_{i_1 j_1}) s_0 = G_{i_0 + \pi'_{i_0}(i_1) p, j_0 + \pi'_{j_0}(j_1) p} \\
= \hat{X}(i_0 + \pi'_{i_0}(i_1) \cdot p)^T \hat{X}(j_0 + \pi'_{j_0}(j_1) \cdot p) = [(\hat{X} \circ \Pi)(i)]^T [(\hat{X} \circ \Pi)(j)].
\]

\[\square\]

The next claim generalizes Lemma 3 of Barber and Candès: applying a null-only sequence of permutations \( \Pi \) to the columns of \( \hat{X} \) has no effect on the distribution of \( \hat{X}^T y \).

**Claim 5.** \( (\hat{X} \circ \Pi)^T y \overset{d}{=} \hat{X}^T y \).

**Proof.** As noted by Barber and Candès, \( y = X\beta + \epsilon \sim N(X\beta, \sigma^2 I) \), and therefore \( \hat{X}^T y \sim N(\hat{X}^T X \beta, \sigma^2 \hat{X}^T \hat{X}) \), and \( (\hat{X} \circ \Pi)^T y \sim N((\hat{X} \circ \Pi)^T X \beta, \sigma^2((\hat{X} \circ \Pi)^T (\hat{X} \circ \Pi)).

By Claim 4, \( \hat{X}^T \hat{X} = (\hat{X} \circ \Pi)^T (\hat{X} \circ \Pi) \), therefore it suffices to show that for \( i = 1, \ldots, p \),

\[
(\hat{X}^T X)(i) \cdot \beta_i = ((\hat{X} \circ \Pi)^T X)(i) \cdot \beta_i.
\]

This, again, follows along the lines of Barber and Candès: first, clearly (5.7) holds for \( i \) for which \( \beta_i = 0 \). For \( \beta_i \neq 0 \) we need to show that the \( i \)th columns of \( \hat{X}^T X \) and of \( (\hat{X} \circ \Pi)^T X \) are identical. Consider the \( j \)th entry of that column where \( j = j_0 + j_1 p \), with \( j_0 \in \{1, \ldots, p\} \) and \( j_1 \in \{0, \ldots, d\} \). Then,
1. If $\beta_{j_0} \neq 0$ then as $\Pi$ is a null-only sequence of permutations, $\pi_{j_0} = Id$ and therefore

$$\hat{X}(j)^T X(i) = \hat{X}(j_0 + j_1 p)^T X(i) = \hat{X}(j_0 + \pi'_{j_0}(j_1) \cdot p)^T X(i) = \left((\hat{X} \circ \Pi)(j)\right)^T X(i),$$

so (5.7) holds.

2. Else, $\beta_{j_0} = 0$ so $j_0 \neq i$ and therefore

$$\hat{X}(j)^T X(i) = \hat{X}(j_0+j_1 p)^T X(i) = \sigma_{j_0,i} = \hat{X}(j_0+\pi'_{j_0}(j_1) \cdot p)^T X(i) = \left((\hat{X} \circ \Pi)(j)\right)^T X(i),$$

and again (5.7) holds.

We finally generalize Lemma 1 of Barber and Candès. Recall that $V_i = (\tilde{Z}_i^0 = Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^p)$ and $V_i \circ \pi_i = (\tilde{Z}_{\pi_i}^{(1)} - 1, \ldots, \tilde{Z}_{\pi_i}^{(d_1)} - 1)$.

Claim 6. For any null-only sequence of permutations $\Pi$, $(V_1, \ldots, V_p) \overset{d}{=} (V_1 \circ \pi_1, \ldots, V_p \circ \pi_p)$.

Proof. As explained by Barber and Candès, $\{V_i\}$ depend only on $\hat{X}^T \hat{X}$ and $\hat{X}^T y$. By Claim 4, $(\hat{X} \circ \Pi)^T(\hat{X} \circ \Pi) = \hat{X}^T \hat{X} = G$ and by Claim 5, $\hat{X}^T y \overset{d}{=} (\hat{X} \circ \Pi)^T y$.

The result now follows by observing that applying the Lasso to $(\hat{X} \circ \Pi, y)$ would produce the vectors $V_i \circ \pi_i : \hat{X} \beta = (\hat{X} \circ \Pi)(\beta \circ \Pi)$.

The last claim completes the proof showing that the joint distributions of $(V_1, \ldots, V_p)$ and $(V_1 \circ \pi_1, \ldots, V_p \circ \pi_p)$ are the same.

As defined, our construction is only applicable when $n \geq d_1 p$, which greatly limits its utility, but we can relax this restriction by using an analog of Barber and Candès’ extension to the case where $p \leq n < 2p$. Namely, as long as $n - p$ is reasonably large we can estimate $\sigma^2$ by $\hat{\sigma}^2$, the variance of the noise in (1.4), extend the design matrix $X$ with $d_1 p - n$ rows of 0s and extend the response $y$ with $d_1 p - n$ independent draws from the $N(0, \hat{\sigma}^2)$ distribution (Barber and Candès, 2015). One problem with this extension is that the guarantee of the last theorem no longer applies, although in practice as long as $n - p$ is not very small this did not seem to be a major issue.
However, the more significant problem we face, regardless of whether or not an extension is required, is that according to (5.3) $s_0$ is decreasing with $d$. Recalling Barber and Candès’ argument that a smaller $s_0$ leads to a loss of power (because of the increased correlation between a real variable and its knockoff copies), we see that as we increase the number of knockoffs, we reduce the power associated with each individual copy. In practice, the overall effect of using multiple knockoffs is therefore mixed where the introduction of additional knockoffs can often reduce power rather than increase it, an effect we will see in action in Section 5.5.

5.2.2 Controlling the FDR

Just as in the single knockoff case the remaining part of the multiple-knockoff procedure proceeds exactly as it did in the case of independent decoys from Chapter 2. Recalling the multiple competition framework of Section 2.2.4, we proceed by associating each potential feature with a null hypothesis $H_i : \beta_i = 0$ and a corresponding alternative $\beta_i \neq 0$. For a given choice of $c$ and $\lambda$ we obtain the labels $L_i$ using the formulation (2.15) above with the combined list of $d_1$ scores $\{Z_i, \tilde{Z}_1^i, \ldots, \tilde{Z}_{d_1}^i\}$ and calculate $W_i$ according to the mirandom mapping scheme $\varphi_{md}$ as described in Section 2.2.4. With the feature scores $W$ and labels $L$ defined, for a given the FDR threshold $\alpha$ we sort the selected scores $W_i$ and report $D_K(\alpha, c, \lambda) := \{i : i \leq i_{\alpha c \lambda}, L_i = 1\}$ as the accepted features, where $i_{\alpha c \lambda}$ is given by equation (2.17).

Thus, applying any one of our $(c, \lambda)$ selection procedures to the combined set of original and knockoff scores yields a multiple knockoff procedure that generalizes Barber and Candès’ original approach to controlling the FDR in the variable selection problem. When $n \geq d_1 p$ and the tuning parameters $(c, \lambda)$ are predetermined Theorems 1 and 2 guarantee that applying our multiple competition framework controls the FDR in the finite setting just as Barber and Candès’ original knockoffs do.

While our current multiple knockoff procedure has guaranteed FDR control, it is limited in both applicability and power. Indeed as we alluded to above and our simulations in Section 5.5 will show, the results compared to the standard knockoff+ are mixed. Hence, we now introduce two additional improvements to our procedure that
allow us to obtain significantly better results. The first is a heuristic that we term \textit{batching}, which attempts to solve the issue that was mentioned at the end of Section 5.2.1, in which adding additional knockoffs increases the amount of correlation between the original feature and its knockoff copies, a primary factor causing our poor results.

Secondly, just as in Chapter 2, we can introduce data-driven methods of selecting \((c, \lambda)\) to improve power. All our procedures from that chapter are able to be applied here, in particular LBM, our recommended procedure, is still able to be used as the resampling method avoids the generation of extra knockoff scores. While these methods can still work well, we can actually obtain even better results by developing an alternative, more specialized resampling procedure that selects not just the optimal values for \(c\) and \(\lambda\) but also for \(d\), the number of knockoffs. While we do lose the guarantee of FDR control with the introduction of both batching and the more sophisticated parameter selection procedures, in practice the simulations below indicate that the methods we consider here do appear to control the FDR.

5.3 Batching

We begin introducing our improvements with the batching heuristic which consists of partitioning the original set of features, or their indices \(P := \{1, \ldots, p\}\), into a disjoint union \(P = \bigcup_j I_j\) and separately creating the knockoffs for each subset of features \(I_j\). This allows us to reduce the size of the matrix \(G\) so that \(s_0\) can be made larger. Specifically, we simultaneously create \(d\) knockoff variables for each of the original features \(X_i\) for \(i \in I\), where \(I \subset P\). The \(d \cdot |I|\) knockoffs created in this batch will need to have exactly the same correlations among themselves, as well as with all the original variables, as they have when we create knockoffs for all the variables at the same time.

In order to do this, we essentially repeat our knockoff construction procedure for simultaneously creating the knockoffs for all features but omitting all uninvolved knockoff features, that is, columns \(\tilde{X}_{i+jp}\) with \(i \in P \setminus I\). Specifically, we define the augmented design covariance matrix \(G^I\) as a \((p + d|I|) \times (p + d|I|)\) dimensional block matrix made again of \(d_1 \times d_1\) blocks \(B_{ij}\) (of varying sizes), which are defined here for \(i, j \in \{1, \ldots, d_1\}\).
Chapter 5. Multiple Competition with Knockoffs

as:

\[
B_{1j} = \begin{cases} 
\Sigma & j = 1 \\
\Sigma_{0}^{j} & j > 1
\end{cases}
\quad B_{i1} = \begin{cases} 
\Sigma & i = 1 \\
\Sigma_{0}^{i} & i > 1
\end{cases}
\quad B_{ij} = \begin{cases} 
\Sigma_{I}^{j} & i = j > 1 \\
\Sigma_{0}^{j} & i \neq j, i \wedge j > 1
\end{cases},
\]

where \(A_{IJ}\) is the restriction of the matrix \(A\) to the rows specified by the set \(I\) and the columns specified by the set \(J\). For example, if \(I = \{1, 2\}\) and \(d = 3\) then

\[
G_{I} := \begin{bmatrix} 
\Sigma_{0}^{I} & \Sigma_{0}^{II} & \Sigma_{0}^{IP} \\
\Sigma_{0}^{IP} & \Sigma_{0}^{II} & \Sigma_{0}^{II} \\
\Sigma_{0}^{IP} & \Sigma_{0}^{II} & \Sigma_{0}^{II}
\end{bmatrix},
\]

where \(\Sigma_{II} = \begin{bmatrix} 
\sigma_{11} & \sigma_{12} \\
\sigma_{21} & \sigma_{22}
\end{bmatrix}, \Sigma_{0}^{I} = \begin{bmatrix} 
\sigma_{11} - s_{0} & \sigma_{12} \\
\sigma_{21} & \sigma_{22} - s_{0}
\end{bmatrix}, \Sigma_{0}^{IP} = \begin{bmatrix} 
\sigma_{11} - s_{0} & \sigma_{12} & \cdots & \sigma_{1p} \\
\sigma_{21} & \sigma_{22} - s_{0} & \cdots & \sigma_{2p}
\end{bmatrix},\) and \(\Sigma_{0}^{IP} = (\Sigma_{0}^{IP})^T\).

We want to construct the \(n \times d|I|\) dimensional knockoff matrix \(\tilde{X}_{I}\), so that the correlation (Gram) matrix of the partially-augmented design matrix \([X \tilde{X}_{I}]\) is \(G_{I}\). Again, this can be done if we can find \(s_{0}\) such that, with \(\Sigma_{0} = \Sigma - s_{0} \cdot I\), \(G_{I} \succeq 0\). Of course, with our new partial knockoff scheme the \(s_{0}\) we chose for the full matrix \(G\) in (5.3) is no longer optimal. Indeed, this was our motivation for looking at the partial knockoff scheme to begin with. Instead, we use a numerical procedure to find the value \(s_{0}\) for which the minimal eigenvalue of \(G_{I} = G_{I}(s_{0})\) is zero (or \(s_{0} = 1\) and \(G_{I} \succeq 0\)).

We proceed with constructing the matrix of knockoff variables \(\tilde{X}_{I}\) using mostly the same procedure described in 5.2.1 to generate the complete set of knockoff features with a couple of notable differences relating to the definition of the orthogonal transformation \(\tilde{U}\) that maps \(X_{0}\) to \(X_{1}\) (5.6). When generating the full set of multiple knockoffs \(\tilde{X}\) the map \(\tilde{U}\) is defined by (5.5), where \(Q\) is obtained by applying the QR factorization to an arbitrary extension \(A\) of \(X\) (5.4). We found that our batched knockoffs benefit from the following more elaborate construction of \(Q\) that aims at reducing some unwarranted correlations between the knockoff variables from different batches.
First, possibly using the same extension procedure mentioned earlier, we verify that \( n \geq d_1 p \) (again assuming that initially \( n > p \) and \( n - p \) is not too small). We then apply the same thin QR factorization as in (5.4) to create the \( n \times d_1 p \) matrix \( Q_b \) with orthonormal columns. Then, when constructing the batch of knockoffs \( \tilde{X}^I \) we define the batch-specific map \( \tilde{U} \) using a batch specific \( Q := Q^I \), where \( Q^I \) consists of the first \( p \) columns of \( Q_b \) as well as its \( d \mid I \mid \) columns corresponding to the knockoffs associated with batch \( I \). The result is that each batch of knockoffs can be expressed as a linear combination of the original features and vectors in a batch specific subspace, where these subspaces are orthogonal to one another, as well as to the original features subspace. The rest of the procedure is unchanged.

We stress that batching is a heuristic: in general the resulting knockoffs do not satisfy the conditional null exchangeability property. In particular we found that if the number of batches is too large, for example when each feature defines its own batch, the conditional null exchangeability could be violated in such a way that our competition-based FDR control can fail (see Section 5.5.2 below for such an example with \( d = 1 \)).

To address this problem we first require that the sets \( I_j \) are not too small (in practice we used an average of at least 4 or 5 features per batch). In addition, to make use of the fact that knockoffs that share the same batch are guaranteed to retain the same correlation structure as the corresponding original features we used the following clustering approach to create the partition that defines the batches: defining the leaves as the columns of the original matrix \( X \) we first construct an agglomerative hierarchical cluster tree using the averaged (Euclidean) distance between features as the distance metric (UPGMA). Then, traversing the tree from its root we determine the clusters, or our partition, based on the pre-specified number of batches. Thus, the more correlated the original features are, the more likely it is that the same correlation would be retained between its knockoffs. In Section 5.5.2 below we given an example demonstrating the potential advantage clustering defined partition can offer.

Regardless of how our partition \( P = \bigcup_j I_j \) is defined, our revised multiple knockoff construction procedure then applies the above partial knockoff procedure, using each set of indices, \( I_j \), at a time, to create a \( n \times (p + d \mid I_j \mid) \) augmented design matrix \( \left[ X\tilde{X}^{I_j} \right] \). It then applies the Lasso procedure to this design matrix (and \( y \)) to obtain the set of
scores \( \{ \tilde{Z}_i^0 := Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^d \} \) for each feature \( i \in I_j \) ignoring the other values for \( i \neq I_j \).

5.4 Model-aware resampling and parameter optimization

We now examine how to use our knockoff scores with a linear regression specific resampling procedure that provides better results than even LBM, the general recommended procedure from Chapter 2. We achieve this by utilizing our knowledge of the underlying regression model to assist with generating bootstrap samples and obtain what we call model-aware bootstrap (or simply model-bootstrap) samples. In this section we describe this new resampling technique and how we use it in a new selection procedure that we name multi-knockoff that seems much more suitable for optimally setting \((c, \lambda)\). Our last selection method described below, multi-knockoff-select, also relies on our new resampling technique but it goes one step further than the other procedures we consider by trying to determine the optimal number of knockoff copies \(d\).

Our model-aware resampling method adopts the same “labeled resampling” procedure of conjectured true/false null labels that was introduced in our generic bootstrap approach that LBM relies on (2.4.1). Here a conjectured false null label corresponds to a variable that is conjectured to be included in the model, and a conjectured true null label to a variable that is not included in the model. While the original LBM algorithm continued to resample the indices in the usual bootstrap manner and then randomly permuted the vector of original and decoy scores for each resampled index corresponding to a conjectured true null label, our new model-resampling scheme instead first regresses the response variable on the conjectured included variables and then uses the resulting linear model to generate a new sample of the response variable. The precise procedure is as follows:

1. Determine \( \lambda = \lambda_0 \) from the empirical p-values / ranks \( r_i \) of the original variable scores \( Z_i \) as described in Section 2.3.2. Note that we randomly break all ties by first transforming all observed and knockoff scores into ranks.

2. Run the first two steps of our meta-procedure (Section 2.2.4) with \( \lambda = c = \lambda_0 \) and the mirandom map \( \varphi_{md} \) to assign a score \( W_i \) and a knockoff/original win label
To each variable $i = 1, \ldots, p$. Those values of $W_i$ and $L_i$ are kept fixed when generating all subsequent bootstrap samples.

3. To generate each of the $n_b$ model-aware bootstrap samples, for $l = 1, \ldots n_b$ do:

(a) Run steps 4-9 of the algorithm described in Section 2.4.1 to sample an indicator vector $f \in \{0, 1\}^p$ where $f_i = 1$ if the $i$th variable is conjectured to be part of the model (false null) and $f_i = 0$ if the $i$th variable is conjectured to be missing from the model (true null).

(b) With $J = J_l = \{ i : f_i = 1 \}$ let $X_J = X[:, J]$ be the submatrix of $X$ consisting of the columns specified by the set $J$ and use standard least square regression to find the coefficient vector $\beta_J$ that minimizes the residual sum of squares $\|y - X_J \beta_J\|_2^2$.

(c) Randomly draw a noise vector $\varepsilon_l \in \mathbb{R}^n$ from the $N(0, I)$ distribution, where $0$ is the $n$-dimensional zero vector and $I$ is the $n \times n$ identity matrix, and define $y_l = X_j \beta_J + \hat{\sigma} \varepsilon_l$, where $\hat{\sigma}$ is the standard deviation estimated as in Section 2.1.2 of Barber and Candès, 2015 from the residual sum of squares in the original data. Note that $X_j$ here is the 0-extended matrix if $n < d_1 p$.

(d) We next apply our multiple-knockoff generating procedure to $y_l$ and $X$ to generate the model-bootstrap sample of $\left\{ \left( Z_{l,i}, \hat{Z}^0_{l,i}, \hat{Z}^1_{l,i}, \ldots, \hat{Z}^d_{l,i} \right) : i = 1, \ldots, p \right\}$. Note that the set of batched knockoff matrices $\hat{X}^b$ needs to be created only once. Scores are transformed to ranks with ties randomly broken.

4. Return the set of $n_b$ model-bootstrap samples where each sample is accompanied by the corresponding set $J_l$ of the true features.

The model-bootstrap samples are also used differently from the cruder bootstrap samples that LBM relies on. Unlike in the LBM case, we are now able to use the model-aware resamples to directly optimize the parameter selection based solely on the number of discoveries, a strategy that generally fails to control the FDR when applied to the LBM samples. Specifically, we apply the above general selection procedure (Section 2.2.4) for each pair of possible $(c, \lambda)$ values with $1/d_1 \leq c \leq \lambda \leq 1/2$ and select the pair that maximizes the average number of conjectured true discoveries. After
selecting these optimal values for \((c, \lambda)\) our so-called multi-knockoff procedure again proceeds along the general outline of our previous selection methods which all apply our meta-procedure with the mirandom map, defining the selected scores \(W_i\). We provide below empirical evidence that multi-knockoff is overall significantly better than what we achieve relying on our methods from Chapter 2.

Finally, we can take this one step further and try to optimize the power by choosing not only the optimal \((c, \lambda)\) for each fixed number of knockoffs \(d\), but also optimizing over several considered values of \(d\). We do this using the same model-bootstrap samples as described above. Specifically, we first determine for each considered number of knockoffs \(d\) its optimal setting of \(c\) and \(\lambda\), that is the values of these parameters that maximize the average number of conjectured true discoveries, and then we choose the number of knockoffs that maximizes this average. We refer to this procedure as multi-knockoff-select and below in Section 5.5 we offer some empirical evidence for its effectiveness.

### 5.4.1 How many knockoffs to construct?

Note that when applying any of our mirandom-map-based procedures using, say \(d = 3\), knockoffs, we can in principle arbitrarily select that number from a larger constructed set of, say \(d = 7\), knockoffs per feature. However, recalling that increasing \(d\) increases the similarity between an original feature and each of its individual knockoffs it is clear that to optimize the power of the competition-based FDR controlling procedure one should construct exactly as many knockoffs as they will use. In particular, when considering multiple numbers of knockoff copies \(d\), say \(d \in \{1, 3, 7\}\), we are actually constructing three different sets of knockoffs, one for each of these values of \(d\) rather than creating \(d = 7\) knockoffs and selecting either one, three or seven of those.

### 5.5 Simulations

As in Chapter 2 we performed extensive simulations to examine how our proposed multiple knockoff methods behave across a range of different experimental designs. In particular, we investigated two things:
• Whether the knockoffs created with batching still maintain the desired properties required for FDR control (Section 5.5.2).

• The performance of our proposed selection procedures in terms of empirical FDR and power (Section 5.5.3). Specifically, we give empirical evidence that our methods essentially control the FDR in the finite sample case, and we demonstrate that the proposed multi-knockoff-select procedure is overall the most powerful among all the considered methods including knockoff+.

5.5.1 Simulation setup: generating the datasets and defining the original and knockoff scores

We largely adopted the simulation setup of Barber and Candès, 2015, where we repeatedly begin with drawing an \( n \times p \) design matrix \( X \). The rows of \( X \) are independently sampled from a multivariate normal distribution with zero mean and one of the following two types of covariance matrices. The first is the same Töeplitz covariance matrix \( \Theta_\rho \) in the original setup of Barber and Candès, 2015 where for \( \rho = 0 \) the covariance matrix is \( \Theta_0 := I_p \), the \( p \)-dimensional identity matrix, corresponding to no feature correlation, and for \( \rho > 0 \), \( (\Theta_\rho)_{ij} = \rho^{|i-j|} \) which introduces some feature correlation. We also introduced a second class of covariance matrices \( \Omega_\rho \) that are constant \( \rho > 0 \) on the off-diagonal terms and with a diagonal of 1s.

We next draw \( K < p \) indices \( i_1, \ldots, i_K \in \{1, \ldots, p\} \) for which we set \( \beta_{i_j} := \pm A \), where \( A \) is a fixed amplitude, and the signs are drawn independently and uniformly. The rest of the values of the coefficient vector \( \beta \) were set to 0, corresponding to a model with the \( K \) features \( i_1, \ldots, i_K \) (so the corresponding hypotheses \( H_{i_1}, \ldots, H_{i_K} \) are false nulls). Finally, we draw the noise vector \( \varepsilon \) as iid \( N(0,1) \) variates and we define the response vector \( y \) through (1.4).

For each such randomly generated pair of a design matrix \( X \) and a response vector \( y \) we use \( b \) batches to construct the set of the original plus \( d \) knockoff scores per feature \( \{(\tilde{Z}_i^0 := Z_i, \tilde{Z}_i^1, \ldots, \tilde{Z}_i^d) : i = 1, \ldots, p\} \) as described in Section (5.3). Note that even when we construct a single knockoff set \( d = 1 \) using a single batch \( b = 1 \) it will in practice differ from the one generated by knockoff+ although the two sets are essentially equivalent.
In Supplementary Section B.1 we provide more details about the specific combination of parameter values that we used in our simulations for generating the data (design matrix and response variables) as well as for constructing the knockoffs (number of knockoffs and batches).

5.5.2 An assessment of the batched knockoffs

While we will explicitly examine the FDR control of our competition-based procedures below, we first examine our knockoffs from a different perspective. As noted above, our procedures that use a pre-determined value of \((c, \lambda)\) will control the FDR provided the knockoff scores satisfy the conditional null exchangeability. However, this exchangeability is unlikely to apply in general for our batched knockoffs and moreover it is not directly a necessary condition for FDR control.

If conditional exchangeability holds then sorting the randomly-selected scores \(W_i\) in decreasing order and applying our general selection procedure with a predetermined \(c = \lambda\), for any true null feature \(j\), \(P(L_j = 1) = c\) and \(P(L_j = -1) = 1 - c\) independently of all other features (2.2.4). Going back to the critical ratio (2.17) we see that our procedure’s control of the FDR hinges on the expected proportion of original \((L_j = 1)\) vs. knockoff wins \((L_j = -1)\). Indeed, if there are \(i_0\) true null features among the top \(i\) scores then the number of original wins among those is a binomial \((i_0, c)\) random variable (RV), and the number of knockoff wins is the complementary binomial \((i_0, 1 - c)\). Therefore, when multiplied by the \(c/(1 - c)\) factor, the expected value of the numerator of (2.17) bounds \(i_0 \cdot c\) which is the expected number of true null features among the original wins in the top \(i\) scores.

In this section we therefore evaluate the quality of our knockoffs from this perspective: considering only the true null features, are the numbers of original score wins among the top \(i_0\) null features consistent with a sequence of binomial RVs defined as the cumulative sum of iid Bernoulli\((c)\) RVs? A specific concern is when that observed sequence of true null original wins significantly exceeds the expected value of the latter theoretical sequence because it would indicate a potential liberal bias in our FDR estimation.
Note that in the case of a single batched knockoff per feature \((d = 1)\) we have a related point of reference which is to compare the same percentage of original wins among the top true null features when using our batched knockoffs with the corresponding percentage observed when using Barber and Candès’ knockoffs. The latter are guaranteed to satisfy the conditional null exchangeability so any observed deviations from the expected 50% of original wins is due to random fluctuations.

**Too many batches can be problematic**

We used the above mentioned reference point to show the potential problem with having too many batches. Specifically, we generated datasets of 60K runs as described in Section 5.5.1, each with \(p = 50\), \(n = 100\), a covariance matrix \(\Theta_\rho = I_p\) \((\rho = 0)\), \(K = 1\) feature included in the model and an amplitude that was deliberately set very high at \(A = 10.0\). For each of the 60K datasets we used Barber and Candès’ construction, as well as our batched construction with the maximal possible number of 100 batches — so each batch contained a single feature — to generate the sets of original feature scores \(Z_i\) with their corresponding knockoff scores \(\tilde{Z}_i\).

With \(c = \lambda = 1/2\) and only one knockoff copy, we counted a feature as an original win if \(Z_i > \tilde{Z}_i\) (ties were randomly broken) and sorted the winning scores \(W_i = \max\{\tilde{Z}_i, \tilde{Z}_i\}\) in decreasing order, again randomly breaking ties. We then noted the percentage of target wins among the top \(i_0\) scores corresponding to the true null features as we varied \(i_0\) from 1 to 49 (the score of the single false null feature was not considered here).

Recall that we evaluate our batched knockoffs against the assumption that the sequence of proportions we observe is consistent with that generated by a cumulative sum of iid Bernoulli\((c = 1/2)\) RVs. Under that assumption we can get some idea of whether our batched knockoffs are consistent with this model by plotting the 97.5% and 2.5% quantiles, as well as the mean, of the corresponding binomial RVs (in practice we used the normal approximation to draw the quantiles). Keep in mind that these plotted quantiles are only provided for reference: they are asymptotically only valid pointwise, so even for data that is consistent with the model the probability that the
curve will wander out of the band outlined by the quantiles is, of course, higher than 5%.

Judging by panel A of Figure 5.1 it seems that in this example where each batch consists of a single feature the resulting knockoffs exhibit a clear liberal bias: the percentage of original wins among the top true null features significantly exceeds our model-determined expected value of 1/2, as well as the variability we observed in Barber and Candès’ knockoffs. This bias further manifested itself in compromised FDR control. For example, applying our batched-knockoff+ (Section 5.5.3) we find that the empirical FDR at \( \alpha = 0.5 \) is 0.5144. This 3% overshoot of the empirical FDR might not seem that much, however our empirical FDR was computed from 60K independent samples so statistically it is a very significant deviation (8.8 standard deviations).

**Clustering the features can help**

In practice we found that with an average of five or more features per batch we avoid the significant bias observed in the example above. As mentioned in Section 5.3, we partition the features into their batches by clustering them based on the similarities of the corresponding columns of the design matrix. While this clustering based partition typically delivers only a modest improvement compared with an arbitrary uniform partition, there are cases where the difference can be significant. To see that we again consider the effect of batching on a single knockoff only now our emphasis is on the difference in percentage of target wins between these two types of partitions: uniform vs. clustering.

Specifically, we generated two sets of 50K datasets each with \( p = 200, n = 800 \), a covariance matrix \( \Omega_p \) with \( \rho = 0.7 \), \( K = 10 \) features included in the model and an amplitude \( A = 2.8 \). Each dataset’s features were partitioned into 40 batches but for the first 50K datasets we randomly and uniformly assigned 5 features to each batch while clustering was applied to define the batches of the subsequent 50K datasets.

Comparing panels B and C of Figure 5.1 we see that while clustering based batching creates knockoffs for which the target wins percentage is in line with our model (panel B, black curve), the uniformly partitioned batches exhibit an undesirable significant liberal bias at some point (panel C, black). In both cases we added for reference
Chapter 5. Multiple Competition with Knockoffs

the corresponding percentages we observe using Barber and Candès’ provably-reliable knockoffs.

Model-wise the batched multiple knockoffs behave similarly to their non-batched counterparts

In light of the above examples, and unless otherwise stated, our batched knockoffs were generated using clustering with an average of at least five features per batch. In this section we look specifically at the effect of batching on the agreement between the observed percentage of target wins among the true nulls and our model.

We begin with an example that did not require extending $X$: we generated two sets of 10K datasets, both with $p = 200$, $n = 800$, using an amplitude $A = 2.8$, $K = 10$ features included in the model and a covariance matrix $\Theta_\rho = I_p$ ($\rho = 0$). We then compared the percentage of target wins using $d = 3$ non-batched knockoffs with the same percentage when using $d = 3$ knockoffs constructed using 40 batches. Panel D of Figure 5.1 shows that in this case our batched knockoffs behave similarly to the un-batched ones. Notably, the latter are guaranteed to follow the model and indeed, in both cases the percentage of target wins does not deviate significantly from the theoretical $c = 1/2$ (using $c = 1/4$ yields qualitatively similar results).

The next example looked at $p = 200$ and $n = 600$ with $d = 3$ knockoffs as before and thus, as $n < d_1p$, required extending $X$. Again, we generated two sets of 10K datasets, one where the knockoffs were created using 40 batches per dataset and the other using a single batch per dataset. In this example all features were true null ($K = 0$) and the covariance matrix was $\Theta_\rho = I_p$. Panel E of Figure 5.1 shows that again our batched knockoffs behave similarly to the un-batched ones, and in both cases the percentage of target wins does not deviate significantly from the theoretical $c = 1/4$ (using $c = 1/2$ yields qualitatively similar results). Note that because $y$ was extended using an estimate of $\sigma$ even the un-batched knockoffs are not guaranteed to follow the model in this case but in practice it seems they still do.

In our final example we look at a more significant extension of $X$ where we compared our knockoffs constructed in three different ways. For each of the three we generated 10K datasets using our model with $p = 200$ and $n = 600$, all features are true
null ($K = 0$) and a covariance matrix $\Theta_p = I_p$. Panel F of Figure 5.1 shows that using 40 batches our $d = 11$ knockoffs (black curve) demonstrate a clear liberal bias with $c = 2/12$. Interestingly, when using a single batch to create the same number of $d = 11$ knockoffs (red curve) we observe an even larger liberal bias than the one exhibited by the batched knockoffs (same value of $c = 2/12$). This suggests that the issue lies with the fairly extreme extension we used rather than with the batching. Indeed, constructing our third set of knockoffs using the known $\sigma = 1$, rather than its estimate, to extend $y$ we note that the liberal bias has all but disappeared (green curve). Note that the three curves of Panel F were generated using the same $c = 2/12$ but the results look qualitatively similar using other values of $c = i/12$ with $i \leq 6$. Regardless of the source of the above liberal bias, we will show below that in practice it is sufficiently mild that it does not seem to obstruct our ability to control the FDR in the examples we looked at.

### 5.5.3 Assessing the knockoff selection procedures

We next investigate and compare the performance of our selection procedures by applying them to randomly drawn datasets. Specifically we considered:

- Barber and Candès’ knockoff+, that uses its own single knockoff construction, and “batched-knockoff+” which, like knockoff+, uses a single knockoff but in this case the knockoff is constructed using our batching procedure (so when the number of batches $b = 1$ the two procedures are essentially equivalent, though they can differ substantially when $b > 1$).

- The multiple competition methods of Chapter 2. Specifically the mirror, max and LBM procedures.

- The new multi-knockoff and multi-knockoff-select that use a pre-specified number of model-aware bootstrap samples, $n_b$ (Section 5.4).

In Supp. Sec. B.1 we provide more precise details of the settings that were used by these selection procedures (e.g., number of bootstrap samples).

---

4Note that we needed to extend the response $y$ from $n = 600$ to $n = 2400$ and that it is easy to find examples where any of the knockoff based procedures considered here, including Barber and Candès’ knockoff+, fails to control the FDR where one extends $X$ and $y$ when $n - p$ is fairly small.
FIGURE 5.1: Examining the batched knockoff scores. Each panel examines the percentage of original feature wins among the top scoring true null features. If the knockoffs satisfy the conditional null exchangeability then those percentages can be modeled by the percentage of cumulative successes in an iid sequence of Bernoulli(c) RVs. The cyan colored curves are the 0.975, 0.025 quantiles and the mean of the iid

A. Too many batches (d = 1)

B. Cluster-defined batches (d = 1)

C. Uniform random batches (d = 1)

D. X is not extended (d = 3)

E. X is extended (d = 3)

F. X is significantly extended

Examining the batched knockoff scores. Each panel examines the percentage of original feature wins among the top scoring true null features. If the knockoffs satisfy the conditional null exchangeability then those percentages can be modeled by the percentage of cumulative successes in an iid sequence of Bernoulli(c) RVs. The cyan colored curves are the 0.975, 0.025 quantiles and the mean of the iid
We evaluated the performance of each method by noting its empirical FDR and power as we varied the FDR threshold. Specifically, for each combination of parameter values we randomly drew (typically) 1K datasets and for each considered FDR threshold $\alpha \in \Phi^5$ we averaged the FDP in the reported list of discoveries to get the empirical FDR as well as the percentage of true features in the same list to get the average power.

We used three types of plots to visually study the selection methods we consider: power, power-difference and empirical FDR. Each plot is typically made of multiple curves, where each curve corresponds to a unique combination of parameter values. Specifically, each curve summarizes the results obtained by applying, at each considered FDR threshold, one or two of the methods to (typically) 1K datasets that were randomly drawn with the same given combination of parameter values, where:

- In a power plot ($y$-axis label indicates “Power”) each curve depicts a selection method’s average power over the randomly drawn datasets.

- In a power-difference plot ($y$-axis label indicates “Power Difference”) a curve represents the difference in average power between the first and second methods, so negative values indicate the second method is more powerful at the given FDR threshold.

- In an empirical FDR plot ($y$-axis label indicates “FDR”) the curve yields the ratio between the empirical FDR (average of the FDP) to the FDR threshold, so a value below 1 indicates a conservative bias and a value above 1 indicates a liberal bias.

Multiple-knockoff procedures that rigorously control the FDR in the finite sample case

Comparing the performance of knockoff+ with that of the multiple-knockoff procedures when all are guaranteed to control the FDR we see mixed results. Recall that such finite sample FDR control is guaranteed when the data is generated according to our model, we construct our $d \leq n/p - 1$ knockoffs using a single batch and we apply our procedure with the mirandom map and pre-determined tuning parameters (e.g.,

---

$^5$recall that $\Phi$ is the set from 0.001 to 0.009 by jumps of 0.001, from 0.01 to 0.29 by jumps of 0.01, and from 0.3 to 0.95 by jumps of 0.05.
the mirror and the max methods). Indeed, Theorem 1 from Section 2.2.4 and Theorem 2 above guarantee FDR control in this setting.

Panel A of Figure 5.2 shows that in some cases the max method delivers significantly more power than knockoff+ while in others it can deliver substantially less power, a result that is also observed, though somewhat lessened, with the mirror method (panel B). Supplementary Figure B.1 offers more insight by showing how the power of max and knockoff+ vary with the parameters of the data and the FDR threshold. Overall max tends to do better for smaller FDR thresholds, sparser models and a larger $d$ but the results are generally mixed.

Supplementary Figure B.2 shows a summary of the difference in power between max/mirror/batched-knockoff+ and knockoff+ (left column) as well as the empirical evidence of the corresponding FDR control (right column). Note that because we use a single batch in this case, batched-knockoff+ is essentially equivalent to knockoff+ and the variations in power between them are random.

The guaranteed FDR control setup considered here is rather limited. In practice we would like to apply our methods to the case where $p < n < d^1 p$. In addition, as we have seen in Chapter 2, we can gain significant power by learning $c$ and $\lambda$ from the data, as well as by using batching when creating the knockoffs. We empirically explore these extensions next.

**Batching can significantly increase the power of the knockoff procedures**

Panel C of Figure 5.2 as well as panels A-D of Supplementary Figure B.3 show examples where, as expected, the power of our procedures generally increases with the number of batches because we are able to better distinguish the original features from their knockoffs.

Similarly, Figure 5.3 shows in the context of the various datasets that make up the $n = 800, p = 200, d = 3$ set (Supp. Sec. B.1.1) that increasing the number of batches from 1 to 40 typically yields substantial power gains. This holds for all three procedures we looked at so far: max, mirror and batched-knockoff+, and for the wide range of parameter combinations described in Supp. Sec. B.1.1.
A. Max \((d = 2, 3)\) vs. knockoff+

B. Mirror \((d = 2, 3)\) vs. knockoff+ (max)

C. Varying the number of batches (max)

D. Batched-knockoff+ vs. knockoff+

**FIGURE 5.2: Assorted Figures - Methods and batching.** (A) Power difference between the max method and knockoff+ using \(d = 2\) \((n = 3000, p = 1000, \text{Supp. Sec. B.1.2})\) or \(d = 3\) \((n = 800, p = 200, \text{Supp. Sec. B.1.1})\) knockoffs. Negative values indicate that knockoff+ is more powerful. (B) Same as panel A but comparing the mirror and knockoff+. (C) Power of max for varying number of batches \(b \in [1, 40]\) \((n = 600, p = 200, d = 11, \mu = 2.8, \text{Supp. Sec. B.1.4})\) (D) Power difference between batched knockoff+ and knockoff+ using either \(b = 1\) or \(b > 1\) batches: \(b = 30\) for \(n = 3000, p = 1000, K = 30, \mu = 3.5, d \in \{1, 3\}\), and \(b = 40\) for \(n = 800, p = 200, K = 10, \mu = 3.0, d \in \{1, 3\}\), and for \(n = 600, p = 200, K = 10, \mu = 2.8, d \in \{1, 11\}\), \(\rho = 0\) in all cases, (Supp. Sec. B.1.6). Negative values indicate that knockoff+ is more powerful.
Chapter 5. Multiple Competition with Knockoffs

A. Max using $b = 1$ vs. $b = 40$ batches

B. Mirror

C. batched-knockoff+

D. knockoff+

Figure 5.3: **Batching increases the power (single vs. 40 batches).** Each of the panels shows the difference in the power of one method applied to multiple datasets using $b = 1$ ($n = 800, p = 200$, Supp. Sec. B.1.1), and $b = 40$ batches ($n = 800, p = 200$, Supp. Sec. B.1.3). The design of the experiment involved drawing a new set of 1K datasets for each value of $b$. (D) knockoff+’ does not use batching so variations are simply due to the differences in the randomly drawn datasets.
As expected, batching offers a larger increase in power as \( d \) and \( p \) increase. Some evidence of this can be seen in panel D of Figure 5.2 and the left column panels of Supplementary Figure B.5, which compare the power of max, mirror and batched-knockoff+ to the power of knockoff+ using \( b = 1 \) and \( b = 40 \) batches: the gains using \( b = 40 \) are significantly larger when \( p \) is increased from 200 to 1000 as well as when \( d \) is increased from 3 to 11.

As mentioned in Section 5.5.2, FDR control can be compromised when introducing batching, and particularly when a significant extension of \( X \) and \( y \) is involved. Thus, we should examine whether the significant power gains we see in our examples when we introduce batching are not attained at the cost of compromised FDR control. Figure 5.4 and the right panels of Supplementary Figures B.3, B.4, and B.5 confirm that the FDR seems to be properly controlled in spite of the large power gains.

**Empirically choosing the tuning parameters**

We first compare the performance of LBM, our general multiple-competition selection procedure, with that of multi-knockoff which is explicitly designed for this linear regression context. Specifically, we apply both methods to all the datasets in our combined collection of experiments, which spans the wide range of parameter values described in Supp. Sec. B.1.1. Panels A-C of Figure 5.5 show that the model-aware multi-knockoff generally offers more power than, not only the knockoff+ procedure (regardless of batching), but also the LBM procedure recommended in Chapter 2.\(^6\) More specifically, comparing panels A and B of Supplementary Figure B.6 we find that the advantage of multi-knockoff becomes evident when the number of knockoffs is larger: for \( d = 3 \) (panel B) we do not see much of a difference, which is expected given that in this case multi-knockoff considers only three possible combinations of values for \( c \) and \( \lambda \).

When we rely on data-driven methods to set the values of \( c \) and \( \lambda \) we lose the theoretical guarantee of FDR control regardless of whether or not we use batching and/or extension. Resorting to simulation studies we find that in the same extensive set of experiments both LBM and multi-knockoff seem to essentially control the FDR (panels

---

\(^6\)The one example where LBM is moderately better than multi-knockoff (cyan colored) corresponds to a realistically borderline 80% proportion of features in the model: \( K = 160 \) and \( p = 200 \) (\( n = 600, d = 11 \)).
Chapter 5. Multiple Competition with Knockoffs

Figure 5.4: Batching does not seem to compromise the finite sample FDR control. Each of the panels shows the ratio of the empirical FDR to the FDR threshold of one method applied to multiple datasets using $b = 40$ batches ($n = 800$, $p = 200$, Supp. Sec. B.1.3). The graphs show that in all the cases the methods seem to essentially control the FDR. (D) knockoff+ control of the FDR is guaranteed in this setting where $n \geq 2p$. 

A. Empirical FDR (max)  
B. Empirical FDR (mirror)  
C. Empirical FDR (batched-knockoff+)  
D. Empirical FDR (knockoff+)
Chapter 5. Multiple Competition with Knockoffs

C and D of Figure B.6), so the advantage of multi-knockoff does not seem to come at the expense of controlling the FDR.

With multi-knockoff’s optimization of the tuning parameters \((c, \lambda)\) apparently being better than LBM’s we went ahead and also compared the former’s power against all the other methods we consider here. Panels A-E (panels A, B and E are identical to panels A-C of Figure 5.5) of Supplementary Figure B.7 show that compared to the mirror and max methods multi-knockoff is a better choice: more often than not it delivers more power than each of the other methods, and moreover, when it is not optimal it is giving up only a small amount of power (certainly for the more practical FDR thresholds of \(\alpha \leq 0.3\)), while often enjoying a substantial advantage in power when it is optimal.

**Choosing the optimal number of knockoffs**

So far we examined the performance of the methods when the number of knockoff copies \(d\) is given. However, it is not clear how to choose an optimal value of \(d\) as the setup here is quite different to the independent decoys model that we examined in Chapter 2. In the latter case, the larger \(d\) is the more power the multiple-decoy procedure will generally deliver, however in our linear regression context there is a delicate balance between the increased power due to the increasing number of competing knockoffs and the reduction in power due to increased correlation between the knockoffs and the original features. Panel D of Figure 5.5 and panels A and B of Supplementary Figure B.8 demonstrate this problem: the optimal number of knockoffs varies with the method we use, the parameters of the problem, and the FDR threshold. This was the motivation behind our new multi-knockoff-select that tries to optimally select \(d\) from the choices it is given; so how well is it doing in practice?

Panels E and F of Figure 5.5 show that in the case of the experiments described in Supp. Sec. B.1.9 \((n = 600, p = 200, d \in \{1, 3, 7, 15, 31\}, b = 40)\) multi-knockoff-select seems to consistently select a nearly optimal \(d\): in the studied cases its power for any \(\alpha \leq 0.5\) was at worst 5% below the power of multi-knockoff applied with the optimal \(d\) and the power difference was even smaller for \(\alpha \leq 0.2\). At the same time, for each fixed \(d\) there are settings where multi-knockoff-select delivers significantly more power than multi-knockoff. Importantly, the overall performance of multi-knockoff-select on this
Chapter 5. Multiple Competition with Knockoffs

A. Knockoff+ vs. multi-knockoff

B. Batched-knockoff+ vs. multi-knockoff

C. LBM vs. multi-knockoff

D. The optimal $d$ varies (multi-knockoff)

E. Same as D but with multi-knockoff-select

F. multi-knockoff vs. multi-knockoff-select (combined)

FIGURE 5.5: Assorted figures - multi-knockoff and multi-knockoff-select. (A-C) Each panel shows the difference in power between the methods knockoff+, batched-knockoff+ and LBM and multi-knockoff. All the methods were applied to all the datasets that are included the combined set described in Supp. Sec. B.1.11. (D) The optimal value of $d$ varies for multi-knockoff in the example where $n = 600$, $p = 200$, $K = 10$ with $A = 3.0$ and 0 feature correlation $\Theta_0 = I_p$ of the set of experiments in Supp. Sec. B.1.9. (E) multi-knockoff-select (black “ref” curve) seems to do a good job at tracking the near-optimal value of $d$ (same data as in panel B). (F) The difference in average power of multi-knockoff vs. multi-knockoff-select using all six datasets in Supp. Sec. B.1.9.
set generated using six different combinations of parameter values (Supp. Sec. B.1.9) was uniformly better than Barber and Candès’s knockoff+ procedure for $\alpha \leq 0.5$ and often by a significant power margin (panel E, supplementary Figure B.8). At the same time, panel F of the same figure shows that this increase in power was not the result of compromised FDR control.

Moving on to our more extensive set of experiments described in Supp. Sec. B.1.11 we find that multi-knockoff-select’s flexibility of optimizing over $d$ makes it our overall preferred procedure. Indeed, Figure 5.6 shows that compared with any of the other methods we consider here multi-knockoff-select overall offers more power. In particular, panel A shows that multi-knockoff-select essentially uniformly delivers more power than knockoff+ and often significantly more. At the same time, we again find that this increase in power does not come at the expense of our ability to control the FDR (panel A of Figure 5.7).

Finally, it is instructive to take a closer look at the main example Barber and Candès considered of $n = 3000$, $p = 1000$, $K = 30$, $A = 3.5$, and 0 feature correlation $\Theta_0 = I_p$. If we use $b = 50$ batches to construct $d \in \{1, 3, 7\}$ knockoffs then even with only $n_b = 4$ bootstrap runs multi-knockoff-select is a very computationally demanding procedure (about 11 hours per run on a 3.2GHz macMini). Fortunately, this significant computational effort is rewarded as we can see when comparing the power of multi-knockoff-select to that of knockoff+ (panel B of Figure 5.7), and again FDR is well under control (panel C of same Figure).

---

7When an experiment only looks at, say $d \in \{1, 11\}$, then multi-knockoff-select essentially decides whether to use multi-knockoff with $d = 11$ or batched-knockoff+.
A. Knockoff+ vs. multi-knockoff-select

B. Batched-knockoff+ vs. multi-knockoff-select

C. Mirror vs. multi-knockoff-select

D. Max vs. multi-knockoff-select

E. LBM vs. multi-knockoff-select

F. Multi-knockoff vs. multi-knockoff-select

Figure 5.6: Power difference vs. Multi-knockoff-select. Each panel shows the difference in power between one of the methods considered in this paper and multi-knockoff-select. All the methods were applied to all the datasets that are included the combined set described in Supp. Sec. B.1.11. Note that the scale of the y-axis varies across the panels.
Chapter 5. Multiple Competition with Knockoffs

A. Empirical FDR (multi-knockoff-select)

B. Knockoff+ vs. multi-knockoff-select

C. Empirical FDR (multi-knockoff-select)

**Figure 5.7:** More on multi-knockoff-select. (A) The FDR of multi-knockoff-select is empirically controlled on the wide set of parameter values of the combined set (Supp. Sec. B.1.11). (B) Multi-knockoff-select (with $d \in \{1, 3, 7\}$ and $b = 50$ batches) is uniformly more powerful than knockoff+ on the $n = 3000$, $p = 1000$, $K = 30$, $A = 3.5$, $\Theta_0 = I_p$ example of Barber and Candès, 2015. (C) Empirical FDR of multi-knockoff-select on the same data used in panel B.
Chapter 6

Discussion

6.1 Overview

In this thesis we have explored how to extend competition-based FDR control procedures to utilize multiple decoys. Specifically, we have examined how to apply these procedures with $d > 1$ in both an independent decoy setting by looking at peptide detection and the more complex variable selection setting. While competition-based FDR control has been extensively used and studied in these settings when $d = 1$ with procedures such as target-decoy competition and Barber and Candès’ knockoff+ respectively, we are able to obtain significant power gains by extending these methods to a general $d$. Technically, all that we require is that for true null hypotheses the generated decoys/knockoffs and the original scores are exchangeable.

We motivated the need for our methods in the general $d$ case by showing that we cannot obtain desirable results (in terms of both power and FDR control) by simply applying the standard canonical procedures to empirical p-values. Indeed, when constructed, these p-values will typically be too coarse (as we assume $d$ is small) if they are computed using hypothesis-specific empirical null distributions, implying that both BH and Storey will generally make very few discoveries. Pooling the decoy scores of all hypotheses together to construct a single empirical null distribution addresses the coarseness of the p-values but other significant issues remain, especially when the null distributions vary substantially with the hypotheses. Specifically, we show that there are arbitrarily large examples on which Storey’s procedure materially fails to control the FDR and similar examples for which BH can be essentially powerless.

The core of our multiple competition methodology is the meta-procedure based on
the mirandom mapping scheme \((\varphi_{md})\) which guarantees FDR control in the finite sample case for any values of tuning parameters \(c\) and \(\lambda\) so long as they are chosen without looking at the data. The proof relies on showing that a critical component of our competition-based procedure can be identified with Barber and Candès’s SSS+ (Barber and Candès, 2015), or more generally with Lei and Fithian’s AS (Lei and Fithian, 2016) (Supplementary Section A). We stress that simply applying either SSS+, or AS to the empirical p-values will generally fail to control the FDR because these p-values are not independent of the order of the original scores.

We can construct different multiple competition variants by selecting different values of \(c\) and \(\lambda\) to apply with the mirandom map. The mirror method sets \(\lambda = c = 1/2\), the max method uses \(\lambda = c = 1/d_1\), where \(d_1 = d + 1\), and LF uses the heuristic suggested in Lei and Fithian, 2016 of setting \(\lambda = 1/2\) and \(c = \alpha\) (the selected FDR threshold). These three selections rigorously control the FDR because they set the tuning parameters without looking at the data. In contrast, our FDS and FDS\(_1\) procedures peek at the data to determine \(c\) and \(\lambda\). As such, these two methods do not fall under mirandom’s theoretical guarantees, although our extensive numerical simulations suggest that in practice they essentially control the FDR. Both methods rely on a procedure of ours that selects \(\lambda\) in a way which, similarly to the spline method of Storey and Tibshirani, 2003, seeks the point where the p-values histogram flattens out. Both then essentially apply Storey’s method to the hypothesis-specific empirical p-values to determine \(c\) before calling on mirandom.

Our simulation studies showed that which of the above methods delivers the maximal number of discoveries varies with the type/parameters of the experiment, as well as with the FDR threshold \(\alpha\). We thus introduced our LBM procedure which selects a preferred method given the current data and \(\alpha\). LBM relies on a novel resampling technique that generates resampled sets of target and decoy scores together with conjectured true/false null labels. Specifically, LBM selects its preferred method after testing whether the approach which greedily selects the method that reports the largest number of discoveries seems to control the FDR.

We examined how these methods perform through both extensive simulations and real peptide detection datasets. Our simulations suggest that LBM largely controls the
FDR, indeed, its empirical FDR seems to be in line with that of the max method which has a guaranteed control of the FDR. Importantly, while not always delivering the highest number of discoveries, LBM seems to offer the best balance among our methods for both calibrated and non-calibrated datasets. Applying our methods to the peptide detection problem suggests that, as with the simulated data, the FDR is controlled and LBM can deliver significantly more power (up to almost 50% more discoveries) than the single decoy TDC. We also note that LBM yields consistently, and at times substantially, more discoveries than the mirror method in these data sets that are derived from real experiments.

In addition to using our extra decoy scores to improve power through more advanced competition, we also examined procedures that utilize these decoys to calibrate the data. Calibration baselines the target and competing decoy scores, reducing the potential for certain hypotheses to be unfairly biased in the selection procedure due to their individual null distributions. We investigated how to split a pre-specified number of decoy scores between calibration and competition in both a pre-selected fixed manner and a data-driven optimizing procedure. While it has been well established that calibration can improve power in the single decoy case (Keich and Noble, 2015; Keich and Noble, 2017b; Jeong, Kim, and Bandeira, 2012), unfortunately when we move to $d > 1$ (though still small) we observe that calibration provides very mixed results.

Specifically, our simulations and real data experiment showed that while we can make some gains for fixed parameter methods (such as the mirror and LF), these are inconsistent and moreover, still tend to produce worse results than the variable parameter selection methods such as LBM. Additionally, when we consider the variable parameter methods that we recommend, we observe that the issue of calibration is somewhat mitigated by our multiple competition methods and thus, generally speaking, we observe little to no improvement in power when reassigning competing decoys to calibration. In general, it appears that in this multiple decoy setting with a small $d$ using the decoys for improved competition outweighs the benefits provided by calibration, especially when using the recommended data-driven parameter selection procedures. Hence, we cannot recommend diverting competing decoys to explicitly
calibrate the scores, except in cases where doing so will not compromise the competition procedures (such as when $d$ is extremely large), in which case applying canonical procedures such as Storey’s to the empirical p-values will likely return better results.

Most of our competition methods have well defined limits as $d \to \infty$. Of particular note is that FDS essentially converges to the finite sample version of Storey’s approach, albeit using our procedure for selecting $\lambda$. Interestingly, our selection of $\lambda$ seems to address Storey’s procedure’s occasional struggles to control the FDR with small samples (when using the bootstrap option), although this improvement naturally comes at the cost of some power. This is consistent with our observation in the finite-decoy case that applying our $\lambda$-selecting procedure to the empirical p-values resulted in a more reliable control of the FDR than when selecting $\lambda$ using Storey’s bootstrap applied to the same p-values.

We also applied this methodology to the variable selection framework of Barber and Candès’ knockoff+ procedure and showed that we can utilize multiple competition to improve power there as well. Unlike the independent decoys case of TDC, we need to explicitly generalize Barber and Candès’ knockoff construction procedure to generate $d$ knockoffs that maintain the conditional null exchangeability property. By doing this and applying our mirandom-based procedures we prove that under certain conditions (no extension of $X$, no batching and using a pre-determined selection of $c$ and $\lambda$) we can rigorously control the FDR using multiple knockoffs.

Our initial knockoff construction is limited both in terms of its applicability (we require $n \geq d_1 p$) and its utility (panel A of Figure 5.2). To address these issues we combined Barber and Candès’ extension procedure with our proposed batching heuristic in which we construct knockoffs simultaneously for only a subset of the variables at a time instead of all at once. We then applied our given multiple competition testing methods to these newly constructed knockoffs to obtain improved power over the standard construction.

We also constructed a specialized parameter selection technique to obtain even better results than the general recommended method of LBM. By modifying our resampling scheme to be specifically adjusted to the linear regression context we can directly
optimize our choices of not just \( c \) and \( \lambda \) but also of \( d \) the number of knockoffs. This latter case is a non-trivial optimization problem due to the inherent conflict between the advantage that increasing \( d \) offers in terms of the competition and the reduced power that comes with each knockoff due to the increased correlation between the original score and its subsequent knockoff copies (\( s_0 \), as given by (5.3) above, is decreasing as \( d \) is increasing).

Lastly, we empirically examined the performance of our knockoff methods and heuristics through a simulation study. We showed that the revised knockoffs obtained through the extension and batching heuristic still allow us to effectively control the FDR in the variable selection problem while delivering a substantial increase in power. Additionally, we showed that the multi-knockoff-select procedure consistently gives close to optimal results across a range of experiments while still maintaining good control of the FDR.

### 6.2 Related Work

In the context of the spectrum identification problem Keich and Noble, 2017b and Keich, Tamura, and Noble, 2018 recently developed aTDC, a method that averages a number of single-decoy competitions in order to reduce the variability associated with TDC. That paper only provides empirical evidence that aTDC indeed controls the FDR, but more importantly its approach differs substantially from the one offered here, which simultaneously uses all decoys rather than one at a time. Indeed, as pointed out in Keich and Noble, 2017a and above in Section 1.3.1, due to the different nature of native/foreign false discoveries, the spectrum ID problem fundamentally differs from the setup of this work. In particular, the null exchangeability property does not hold for the spectrum ID problem, but it is a much more reasonable assumption for the peptide detection problem that is studied here. Our simulation setup confirms that aTDC typically delivers less power (and often significantly so) than LBM does (panel F, Figures 2.5 and 2.8). We also observed this in our analysis of the real data sets where LBM consistently reported as many or more discovered peptides than aTDC (Figure 2.11). Note, however, that this observation does not mean that LBM can replace aTDC; indeed, for
the reasons discussed above LBM will often be too liberal in the context of the spectrum ID problem.

In the process of working on this thesis we became aware of an ArXiv submission that proposes a similar procedure for FDR control in case-control studies (He et al., 2018). Multiple competing scores are generated by permuting the case/control labels on the data, and then direct competition is utilized to control the FDR without the need to make assumptions on the distribution of the null hypotheses. While an earlier version of this paper only had an equivalent of the mirror method, it was later extended to include a choice of $c$ and selecting $W_i$ using the random map $\varphi_u$. In this work we demonstrated that our independently proposed methods (the first version of our work was submitted in 2017) are superior through the inclusion of the extra parameter $\lambda$ and the use of the mirandom mapping scheme (Figure 2.10). Finally, He et al., 2018 also introduced a data-driven technique for selecting $c$ (denoted as $r$ in their paper) which is a cross-validation style technique that is specially designed for these case-control experiments and not easily generalizable to our setting. Both the deterministic (FDS) and bootstrap-based (LBM) data-driven selection procedures that we propose here are applicable in a much more general case, though it is not immediately obvious which would be superior in the case-control setting.

When we consider the variable selection problem there are a number of alternative procedures for controlling the FDR in this setting (Miller, 2002; Miller, 1984; G’Sell et al., 2016; Meinshausen and Bühlmann, 2010; Liu, Roeder, and Wasserman, 2010). However, Barber and Candès note that those, along with their own knockoff procedure (as opposed to the knockoff+ procedure we examine here), generally only asymptotically guarantee FDR control (knockoff controls a modified version of the FDR – see Section 4.1.2). They further demonstrate that among the procedures that control the FDR in the finite setting of the variable selection problem their knockoff+ seems to be the most powerful one. As we show here, multi-knockoff-select is more powerful than knock-off+, allowing us to identify more truly associated features, while empirically we see that it maintains control of the rate of falsely discovered features even in the finite setting. It does however come at a substantial computational cost as well as of using a mathematically unproven technique.
6.3 Further Avenues of Research

A clear avenue of further research into our general methodology is to develop a proof of FDR control for the data-driven methods of FDS and LBM. While our extensive simulations provide strong empirical evidence that FDR control is maintained, a rigorous proof would guarantee this property.

When it comes to our multiple knockoff procedures specifically, as in the independent decoy case, a natural path for future investigation is to attempt to rigorously prove FDR control for the methods that we only provide empirical evidence for. This includes both data-driven methods such as multi-knockoff-select and also knockoff construction heuristics such as batching. Again, while our simulation studies give strong empirical evidence that control is maintained, the guarantee of a rigorous proof would allow for much more confidence in our methodology.

One of the major weaknesses of our multiple knockoff methodologies is the requirement for \( n \geq d_1p \), which significantly limits its applicability. In practice it is often the case that \( n \approx p \), or even that \( n < p \). Developing new knockoff construction techniques to apply in these settings would expand the practical scope of our procedures significantly. In this work we have utilized Barber and Candès’ extension to deal with case \( p < n < d_1p \), in which we extend \( X \) and \( y \) so that the required condition \( (n \geq d_1p) \) is met. In doing this we estimate \( \sigma \), the noise level, and as we saw, this can be somewhat naive resulting in poor results for large extensions. One potential path for improvement is to explore more sophisticated estimations of \( \sigma \) such as the one in Reid, Tibshirani, and Friedman, 2016.

For the latter case of \( n < p \), which we do not touch on in this thesis, one possible approach would be to combine our multi-knockoff methods with Barber and Candès’ recent extension that is based on a data-splitting technique coupled with the introduction of a two step procedure: acquiring a partial model with \( n > p \) and performing the knockoff procedure on the partial model (Barber and Candès, 2015). Alternatively we could explore a major reformulation of Barber and Candès’ knockoff work in model-X knockoffs.

Model-X knockoffs (or model-free knockoffs) is a more recent procedure introduced
in Candès et al., 2018 that constructs knockoffs probabilistically instead of geometrically, a technique that allows for the application in arbitrary conditional models (not just the linear model) and any dimensions, including the problematic case where \( n < p \). In the development of model-X knockoffs Candès et al. look at the variable selection problem from a different perspective, assuming knowledge of the distribution of the covariates \( (X_1, \ldots, X_p) \) instead of the conditional distribution \( Y|X_1, \ldots, X_p \) which is the opposite of the traditional setting. Thus, they are able to treat the \( X_{ij} \) as random instead of fixed and utilize that stochasticity instead of relying on the specific stochastic properties of the linear model of interest.

The knockoff construction techniques for model-X knockoffs are very complex outside the case where \( X \) is normal (which proceeds similarly to the original fixed-X knockoff construction). However, once we have generated these knockoffs the procedure uses the same competition-based structure that we have outlined in this work. Hence, assuming that one could generate multiple model-X knockoffs that satisfy the exchangeability conditions then we may be able to obtain superior power using the procedures developed here. One potential route of interest to constructing multiple model-X knockoffs would be to attempt to extend the second-order Gaussian approximation of Candès et al. in a way similar to what we have done here with fixed-X knockoffs.

While the model-X knockoffs are designed for a different variant of the linear regression problem where the design matrix itself is also drawn according to some known distribution, this assumption is still consistent with the setup of our simulations so we can briefly compare the performance of the model-X knockoffs to that of the other knockoff procedures in a couple of examples (Supp. Sec. B.1.12). Supp. Fig. B.9 suggests that in the context of our simulations the model-X Lasso signed max (LSM) statistic was roughly on-par or slightly weaker than the original knockoff+, and the model-X Lasso coefficient difference (LCD) statistic significantly lagged behind those two. In particular, unless the FDR threshold was relatively high and the feature correlation extremely high \( (\rho = 0.9) \), all these single-knockoff methods offered significantly less power than multi-knockoff-select. It should be noted however, that despite these results the true power of the model-X knockoff procedure is in its applicability, especially when \( n < p \),
a case in which the other knockoff procedures, including multi-knockoff-select, cannot be applied.
Appendix A

Sequential Hypothesis Testing

A.1 Seq Step +

As mentioned in the main body, the competition based framework we investigate and extend in this work has been developed in many separate fields, from mass-spectrometry to permutation testing and even in variable selection. In many of these cases independent proofs guaranteeing FDR control using such procedures have been constructed. For example He et al., 2015 and Levitsky et al., 2017 showed that single decoy TDC can control the FDR in spectrum ID experiments and Barber and Candès prove that their knockoff procedure can control the FDR in the variable selection framework.

In this work we focus on using the proof of Barber and Candès by adapting it to show FDR control in both our extended multiple knockoff+ procedure and the independent decoy setting of the peptide ID problem. The key factor of Barber and Candès’ proof lies in their sequential hypothesis testing procedure Selective SeqStep+ (Selective Sequential Step-up procedure) (SSS+) and Barber and Candès show through their knockoff+ procedure that the competition framework we consider can be thought of as a special case of SSS+. Many of the proofs in this work will rely on SSS+ so we now review the procedure as introduced in Barber and Candès, 2015.

As previously stated, SSS+ is a sequential hypothesis testing procedure that is applied to a set of ordered p-values $p_1, \ldots, p_m$ corresponding to hypotheses $H_1, \ldots H_m$. These p-values should obey that for all null $j$ and all $u \in [0, 1]$, $P(p_j \leq u) \leq u$. We fix a
value for $c \in (0, 1)$ and any subset $K$ (generally $K = [m]$) and define

$$
\hat{k}_1 = \max \left\{ k \in K : \frac{1 + \# \{ i \leq k : p_i > c \}}{\# \{ i \leq k : p_i \leq c \} \lor 1} \leq \frac{1 - c}{c} \cdot \alpha \right\}, \tag{A.1}
$$

with the convention of setting $\hat{k}_1 = 0$ if this set is empty. We then reject $H_i$ for all $i \leq \hat{k}_1$ and with $p_i \leq c$. Barber and Candès show that, assuming the true null p-values ($H_i$ for $i \in N$) are i.i.d, stochastically dominate the standard uniform distribution, $p_i \geq Unif[0, 1]$, and are independent from the false null p-values, this procedure provably controls the FDR.

### A.2 Adaptive Seq Step

In 2016 Lei and Fithian developed an extension of the Selective SeqStep procedure called Adaptive SeqStep (AS) (Lei and Fithian, 2016). The main idea of AS is to insert a tuning parameter $\lambda$ into $SSS^+$ to obtain more power, in essentially the same manner to which Storey improves on Benjamini and Hochberg’s original FDR controlling procedure. Specifically, for the same experimental setup as $SSS^+$ and for some $0 \leq c \leq \lambda \leq 1.1$ Lei and Fithian define $\hat{k}_{AS}$ as follows:

$$
\hat{k}_{AS} = \max \left\{ k \in K : \frac{1 + \# \{ i \leq k : p_i > \lambda \}}{\# \{ i \leq k : p_i \leq c \} \lor 1} \leq \frac{1 - \lambda}{c} \cdot \alpha \right\} \tag{A.2}
$$

and, as in $SSS^+$, reject all hypotheses $H_i$ with $i \leq \hat{k}_{AS}$ and $p_i \leq c$. Assuming the same restrictions on the p-values as in $SSS^+$ (independent and the distribution of the null p-values stochastically dominates the uniform distribution), Lei and Fithian prove that this procedure controls the FDR.

Just as in Storey’s procedure, the addition of this parameter allows for improved power in the resulting discovery list. In particular if $c$ is small then the numerator in (A.1) will include many false null p-values and thus be upwardly biased. In comparison for (A.2) if $\lambda > c$ then this bias will be mitigated, allowing for more discoveries to

---

1Note that in their paper Lei and Fithian use $s$ instead of $c$. 

be made. Indeed, in their paper Lei and Fithian show that AS can perform much better
than SSS+ in both simulated and real data experiments.
Appendix B

Knockoff Simulations

B.1 Knockoff Simulation Setup

Our general simulation setup is described in Section 5.5.1. In the following sections we give further details about the parameter settings we used in our experiments in generating the original design matrices (and the response variables) and the knockoff features as well as any optional settings of our selection methods. When generating the data we varied the dimension of the design matrix $X$, $n \times p$, the number of true features, $K$, the signal amplitude $A$, and the feature correlation strength $\rho$ while keeping the variance of the noise fixed at $\sigma^2 = 1$ (cf. (5.1) and Section (5.5.1)). We generally randomly sampled 1K sets of data for each of the parameter combinations we considered and constructed a set of original plus $d$ knockoff scores for each feature using the specified number of $b$ batches.

While knockoff+ and batched-knockoff+ were each applied only once to the data — each with its corresponding knockoff — the multiple knockoff procedures were applied separately for each considered value of $d$ (with the knockoffs also separately constructed for each value of $d$, cf. Section 5.4.1).

Following knockoff+ we also use the glmnet implementation of the Lasso (Qian et al., 2013). We found that the set of values the regularization parameter lambda is allowed to assume can have a non-negligible effect on our analysis. This is not surprising given that the original and the knockoff feature score corresponds to the largest value of lambda for which the coefficient of that feature is non-zero. Therefore, to make sure that the differences we observe between the methods are not due to variations in the number of lambdas, we set each method to use the same number of possible lambda
Appendix B. Knockoff Simulations

values. Specifically, this number was set to $5 \cdot (1 + d_{\text{max}}) \cdot p$ (we experimented a little with coefficients other than 5 but kept it at 5 throughout the simulations described here), where $d_{\text{max}}$ is the maximal value of $d$ considered in that experiment. Importantly, it means that knockoff+ also used that maximal number of lambdas even though it is using a single knockoff.

In the same vein, we found that our knockoffs are better behaved if all our batches use the same set of lambdas. Specifically, we use the same exponentially spaced set of lambda values that knockoff+ uses only we set the maximal value to $\max_i x_i^T y/n$ where $x_i$ varies over the columns of all augmented design matrices $\left[ X \tilde{X}^T \right]$ (Section 5.3). Note that when using a single batch this maximal value coincides with the one originally used in the knockoffs package.

Somewhat more surprising was the fact that permuting the columns of the extended design matrix before applying glmnet also occasionally had a substantial effect on the performance of the FDR controlling methods, *including* on knockoff+. Therefore, we uniformly randomly permuted all extended design matrices.

Note that when a multiple number of knockoffs are considered in the same run the data is only resampled once using the largest considered number of knockoffs to create the model-aware resamples.

B.1.1 $n = 800, p = 200, d \in \{1, 3\}, b = 1$

A set of experiments designed to study the performance of FDR control with multiple knockoffs in the setting of guaranteed finite sample control. Each drawn dataset was generated starting with randomly drawing the $n \times p$ design matrix $X$ where $n = 800$ and $p = 200$ while varying the following parameters (drawing 1K independent datasets per each setting of the parameters):

- the number of true features $K$ (model sparsity): $K = 1, 5, 10, 20, 40, 80$ with $A = 3.0$ and 0 feature correlation $\Theta_0 = I_p$.

- the signal amplitude $A = 2.6, 2.8, 3.0, 3.2, 3.4$ with $K = 10$ and 0 feature correlation $\Theta_0 = I_p$. 
• the feature correlation strength $\rho = 0, 0.3, 0.6, 0.9$ of the Töeplitz correlation matrix $\Theta_\rho$ with $K = 10, A = 3.0$.

When analyzing each of these datasets we constructed three sets of knockoffs (cf. Section 5.4.1): the first with a single knockoff per feature as part of running knockoff+, the second using our procedure to construct $d = 1$ knockoff (for use in batched-knockoff+), and the third using our construction of $d = 3$ knockoffs. Note that in this case we used a single batch so while our construction of $d = 1$ knockoff is in practice different from knockoff+’ it is mathematically equivalent to it. In particular, knockoff+ and batched-knockoff+ are essentially equivalent in this setting.

**B.1.2** $n = 3000, p = 1000, d \in \{1, 2\}, b = 1$

Similar in design and intent to the last section except that we used the same dimensions of the data as used in the introduction of knockoff+ Barber and Candès, 2015 and varied:

• the number of true features $K$ (model sparsity): $K = 0, 30, 50$ with $A = 3.5$ and 0 feature correlation $\Theta_0 = I_p$.

• the signal amplitude $A = 3.1, 3.5, 3.9$ with $K = 30$ and 0 feature correlation $\Theta_0 = I_p$.

• the feature correlation strength $\rho = 0, 0.3, 0.6, 0.9$ with $K = 30, A = 3.5$.

For each of these combinations of parameters we drew 1K datasets and constructed two sets of knockoffs, each using a single batch, one with $d = 1$ (again equivalent to knockoff+’ single knockoff) and another with $d = 2$ knockoffs per feature.

**B.1.3** $n = 800, p = 200, d \in \{1, 3\}, b = 40$

In this set of experiments we largely repeated the setup described in Supplementary Section (B.1.1) above with the major difference that we used 40 batches when constructing either $d = 1$ or $d = 3$ knockoffs per feature rather than a single batch. In particular, here batched-knockoff+ substantially differs from knockoff+.
Also, in addition to generating 1K datasets for each of the parameter combinations described in B.1 we generated additional 1K datasets per parameter combination while varying:

- the number of true features $K = 1, 5, 10, 20, 40, 80$ with $A = 2.8$ and 0 feature correlation $\Theta_0 = I_p$.
- the feature correlation strength $\rho = 0, 0.3, 0.6, 0.9$ with $K = 10$, $A = 2.8$.

Applications of multi-knockoff and multi-knockoff-select used $m_b = 32$ model-aware bootstrap samples.

B.1.4 \hspace{1em} n = 600, p = 200, d \in \{1, 11\}, b \in \{1, 5, 10, 20, 40\}

A set of experiments designed to demonstrate the effects that increasing number of batches can have. The datasets were generated using $600 \times 200$ design matrices, with $A = 2.8$ and 0 feature correlation $\Theta_0 = I_p$. For each $b \in \{1, 5, 10, 20, 40\}$ we randomly generated 1K datasets, then using $b$ batches each time we constructed two sets of knockoffs one with $d = 1$ and another with $d = 11$ knockoffs per feature. Applications of multi-knockoff and multi-knockoff-select used $m_b = 32$ model-aware bootstrap samples.

B.1.5 \hspace{1em} n = 3000, p = 1000, d \in \{1, 3\}, b \in \{1, 5, 10, 30, 50\}

Similar in design and intent to the last section except that (a) we used different parameter values, and (b) when creating these sets of knockoffs we did not use clustering when defining the batches, instead we arbitrarily divided the features into equal sized batches (which of course is irrelevant for $b = 1$).

Applications of multi-knockoff and multi-knockoff-select used $m_b = 4$ model-aware bootstrap samples.

B.1.6 “Eclectic batching example”

This set gathered together some specific experiments for demonstrating the effects of batching on larger examples. It consisted of three pairs of experiments:
• \( n = 800, p = 200, d \in \{1, 3\}, b \in \{1, 40\} \) taken from Supp. Secs. B.1.1 and B.1.3.

• \( n = 600, p = 200, d \in \{1, 11\}, b \in \{1, 40\} \) taken from Supp. Sec. B.1.4.

• \( n = 3000, p = 1000, d \in \{1, 3\}, b \in \{1, 30\} \) taken from Supp. Secs. B.1.2 and B.1.10.

**B.1.7** \( n = 800, p = 200, d \in \{1, 3, 7\}, b = 40 \)

In this set of experiments we used the same parameter combinations for generating the data as described in Supplementary Section B.1.1 above. The difference again was in the analysis stage where here we constructed for each drawn design matrix \( X \) three sets of knockoffs (in addition to the set generated by knockoff+). Each set was constructed using 40 batches: one with \( d = 1 \), another with \( d = 3 \) and the third with \( d = 7 \) knockoffs per feature. We then applied knockoff+ and batched-knockoff+ using their corresponding single knockoff set, and we applied each of the multiple-knockoff procedures twice, once to the \( d = 3 \) set and once to the \( d = 7 \) set. Applications of multi-knockoff and multi-knockoff-select used \( m_b = 32 \) model-aware bootstrap samples.

**B.1.8** \( n = 600, p = 200, d \in \{1, 11\}, b = 40 \)

In this set of experiments we used a larger number of knockoffs with \( 600 \times 200 \) design matrices. Again, we generated 1K datasets (design matrix and response variables) for each of the following combinations of parameters, varying:

- the number of true features \( K = 0, 1, 5, 10, 20, 40, 80, 160 \) with \( A = 2.8 \) and 0 feature correlation \( \Theta_0 = I_p \).

- the signal amplitude \( A = 2.4, 2.6, 2.8, 3.0, 3.2 \) with \( K = 10 \) and 0 feature correlation \( \Theta_0 = I_p \).

- the feature correlation strength \( \rho = 0, 0.3, 0.5, 0.7, 0.9 \) of the Töeplitz correlation matrix \( \Theta_\rho \), as well as using \( \rho = 0.5 \) for \( \Omega_\rho \) (constant \( \rho \) on the off-diagonal terms) with \( K = 10, A = 2.8 \).

Note that for two of the experiments we increased the number of runs to 4K from the initial 1K by adding another 3K independent runs to clarify whether the relatively high empirical FDR that was observed in a couple of the settings was significant. In
both cases ($K = 5$ and $K = 0$) the aggregated 4K runs did not show a substantial FDR violation. For each drawn dataset we used $b = 40$ batches to construct two sets of knockoffs, one with a single knockoff per feature, and another with $d = 11$.

**B.1.9 $n = 600, p = 200, d \in \{1, 3, 7, 15, 31\}, b = 40$**

This set of experiments was specifically designed to compare the performance using a varying number of knockoffs while analyzing the same data, as well as to test the ability of multi-knockoff-select to select an optimal number of knockoff. Each of the following combination of parameters was used to generate 1K datasets and for each we constructed a knockoff set for each value of $d \in \{1, 3, 7, 15, 31\}$ using $b = 40$ batches per construction (in addition to the knockoffs generated by knockoff+). To generate the data we varied:

- the number of true features $K$ (model sparsity): $K = 1, 10, 40$ with $A = 3.0$ and 0 feature correlation $\Theta_0 = I_p$.
- the signal amplitude $A = 2.6, 3.0, 3.4$ with $K = 10$ and 0 feature correlation $\Theta_0 = I_p$.
- the feature correlation strength $\rho = 0, 0.5$ of $\Theta_p$ with $K = 10, A = 3.0$.

Applications of multi-knockoff and multi-knockoff-select used $m_b = 32$ model-aware bootstrap samples.

**B.1.10 $n = 3000, p = 1000, d \in \{1, 3\}, b = 30$**

The data for this set of experiments was generated using the same general parameter settings as those used in the simulation part of Barber and Candès, 2015. Specifically, we drew $3000 \times 1000$ design matrices and response variables by varying:

- the number of true features $K$ (model sparsity): $K = 0, 5, 10, 30, 50, 75, 100, 200$ with $A = 3.5$ and 0 feature correlation $\Theta_0 = I_p$.
- the signal amplitude $A = 2.7, 3.1, 3.5, 3.9, 4.3$ with $K = 30$ and 0 feature correlation $\Theta_0 = I_p$. 
Appendix B. Knockoff Simulations

• the feature correlation strength $\rho = 0, 0.3, 0.5, 0.7, 0.9$ of the Toeplitz correlation matrix $\Theta_\rho$, as well as using $\rho = 0.5$ for $\Omega_\rho$ (constant $\rho$ on the off-diagonal terms) with $K = 30, A = 3.5$.

For each of these combinations of parameters we drew 1K datasets and constructed two sets of knockoffs, one with $d = 1$ and another with $d = 3$ knockoffs per feature, using $b = 30$ batches each time.

Applications of multi-knockoff and multi-knockoff-select used $m_b = 4$ model-aware bootstrap samples.

B.1.11 Combined dataset

The “combined dataset” was created by merging together the $n = 3000, p = 1000$ sets (Supp. Sec. B.1.10), the $n = 800, p = 200$ sets (Supp. Secs. B.1.3 and B.1.7 but keeping only one copy of each duplicated $d = 3$ set), and the $n = 600, p = 200$ sets (Supp. Secs. B.1.8 and B.1.9).

B.1.12 The model-X dataset

We used this data to compare against the newer model-X knockoffs of Candès et al., 2018. The design matrix was $n = 600$ by $p = 200$, the model had $K = 10$ features with $A = 2.8$ and we only varied the feature correlation $\rho = 0.0, 0.5, 0.9$ while creating $d = 11$ knockoffs using $b = 40$ batches. Applications of multi-knockoff and multi-knockoff-select used $m_b = 32$ model-aware bootstrap samples. The model-X knockoffs were created as follows:

• The knockoff features were created using the create function of the Matlab knockoffs package (Candès et al., 2018) with the model defined as gaussian coupled with the mean and covariance estimated from the randomly drawn design matrix $X$ using the Matlab functions mean and cov. We used the “equicorrelated” construction because our knockoff construction also uses the same constant $s_0$ construction.
• We generated the model-X Lasso signed max (LSM) statistic using a slightly modified version of the function `lassoLambdaSignedMax` from the knockoffs package that enabled us to randomly permute the order of the columns of the extended design matrix (see Supp. Sec. B.1). We set the function’s `nlambda` parameter to the same value that the other methods were using (see Supp. Sec. B.1).

• We generated the model-X Lasso coefficient difference (LCD) statistic using a similarly modified version of the `lassoCoefDiff` function from the knockoffs package that allowed us to randomly permute the extended design matrix columns but other than that we used all the default settings of the original function.

B.1.13 The set $\Phi$ of FDR thresholds

For computational efficiency we evaluated the power and empirical FDR of each of the considered procedures on a pre-determined set of possible FDR thresholds. Specifically we used the set of FDR thresholds $\Phi$: from 0.001 to 0.009 by jumps of 0.001, from 0.01 to 0.29 by jumps of 0.01, and from 0.3 to 0.95 by jumps of 0.05. Our figures however only extend to an FDR threshold of 0.5 since in practice FDR thresholds higher than 50% are typically of little importance.

B.2 Knockoff Figures
Appendix B. Knockoff Simulations

A. $n = 3000, p = 1000$; varying $A$ (amplitude)

B. $n = 800, p = 200$; varying $A$

C. varying $\rho$ (feature correlation)

D. varying $\rho$

E. varying $K$ (# of features in model)

F. varying $K$

Figure B.1: Power plots of knockoff+ (KO) and max in specific cases of guaranteed FDR control. Each left column panel gives the power of Barber and Candès’ single knockoff juxtaposed with the max procedure using two knockoffs ($n = 3000$, $p = 1000$ and unless explicitly varied, $k = 30, \rho = 0, A = 3.5$, Supp. Sec. B.1.2). The right column panels show knockoff+ and the max using three knockoffs ($n = 800, p = 200$ and unless explicitly varied, $k = 10, \rho = 0, A = 3.0$, Supp. Sec. B.1.1). (A) varying the amplitude: $A \in \{3.1, 3.5, 3.9\}$. (B) varying the feature correlation strength of $\Theta_p$: $\rho \in \{0.0, 0.3, 0.6, 0.9\}$. (C) varying the number of features, sparsity: $K \in \{10, 30, 50\}$. (D) varying the amplitude: $A \in \{2.6, 3.0, 3.4\}$. (E) varying $\Theta_p$’s $\rho$: $\rho \in \{0.0, 0.3, 0.6, 0.9\}$. (F) varying $K \in \{5, 20, 40\}$. Overall max tends to do better for smaller FDR thresholds, sparser models and a larger $d$ but the results are mixed.
Figure B.2: Comparison with knockoff+ and FDR control in guaranteed settings. The left column panels show the difference between the power of the considered method and knockoff+’s power (negative values indicate knockoff+ is more powerful at that threshold). The right column panels examine the ratio of the empirical FDR of the considered method (averaged over 1K runs) to the FDR threshold. The data for both columns consists of experiments in which all methods have guaranteed FDR control: \( n = 3000, p = 1000, d = 2, b = 1 \) (Supp. Sec. B.1.2) and \( n = 800, p = 200, d = 3, b = 1 \) (Supp. Sec. B.1.1). (B) There is a single (essentially random) spike at the FDR threshold of 0.001 (0.1%) where for that particular set of parameters from Supp. Sec. B.1.2 the empirical FDR of max is just below 0.2% so it is almost 20% over the threshold. (E) Batched-knockoff+ vs. knockoff+. As \( b = 1 \) here the two methods are essentially equivalent so variations in power are essentially random. (F) Batched-knockoff+ (knockoff+’s FDR plot is naturally quite similar to this one).
Appendix B. Knockoff Simulations

A. Power of mirror ($n = 600, p = 200$)

B. FDR of mirror

C. Power of batched-knockoff+

D. FDR of batched-knockoff+

E. Power of knockoff+ (variations are random)

F. FDR of max

Figure B.3: Power increases with the number of batches (I). Each of the left column panels shows the power of one method applied using a different number of batches $b \in \{1, 5, 10, 20, 40\}$ to construct the knockoffs. The design of the experiment involved randomly drawing a new set of 1K datasets with for each value of $b$ ($n = 600, p = 200$, Supp. Sec. B.1.4). Each of the right column panels shows the ratio of the empirical FDR to the FDR threshold. (E) knockoff+ is not affected by the number of batches hence the observed power variations give us some idea on the magnitude of the differences due to the randomly generated datasets.
Appendix B. Knockoff Simulations

A. Power of max ($n = 3000$, $p = 1000$)

B. FDR of max

C. Power of batched-knockoff+

D. FDR of batched-knockoff+

E. Power of knockoff+ (variations are random)

F. FDR of mirror

Figure B.4: Power increases with the number of batches (II). Same as Supp. Fig. B.3 except $b \in \{1, 5, 10, 30, 50\}$ and $n = 3000$, $p = 1000$ (Supp. Sec. B.1.5).
A. Max vs. knockoff+

B. Empirical FDR (max)

C. Mirror vs. knockoff+

D. Empirical FDR (mirror)

E. Batched-knockoff+ vs. knockoff+

F. Empirical FDR (batched-knockoff+)

**Figure B.5:** When the batching effect is more pronounced. Each of the left column panels shows the difference in the power of one method vs. knockoff+ applied using either $b = 1$ or $b > 1$ batches: $b = 30$ for $n = 3000$, $p = 1000$, $K = 30$, $A = 3.5$, $d \in \{1, 3\}$, and $b = 40$ for $n = 800$, $p = 200$, $K = 10$, $A = 3.0$, $d \in \{1, 3\}$, and for $n = 600$, $p = 200$, $K = 10$, $A = 2.8$, $d \in \{1, 11\}$; $p = 0$ in all cases (Supp. Sec. B.1.6). The design of the experiment involved drawing a new set of 1K datasets for each value of $b$ but knockoff+ and the method to which it is compared were applied to the same dataset each time. Negative values indicate knockoff+ is more powerful. Each right column panel uses the same datasets as the panel to its left to show the ratio of the empirical FDR of the considered method to the FDR threshold. The ratios are all below 1 indicating the methods seem to control the FDR in all these cases.
Appendix B. Knockoff Simulations

A. LBM vs. multi-knockoff ($d \geq 7$)

B. LBM vs. multi-knockoff ($d \leq 3$)

C. Empirical FDR (multi-knockoff)

D. Empirical FDR (LBM)

Figure B.6: Multi-knockoff vs. LBM. Comparison of the two resampling-based methods for selecting the ($c, \lambda$) tuning parameters. Both methods were applied to all the datasets in our combined collection of experiments, which spans a wide range of parameter values and is described in Supp. Sec. B.1.11. (A) Power difference between LBM and multi-knockoff (negative numbers mean multi-knockoff is better). Only experiments with $d \geq 7$ knockoffs are shown. The one example where LBM is moderately better than multi-knockoff (cyan colored) corresponds to a realistically borderline 80% proportion of features in the model: $K = 160$ and $p = 200$ ($n = 600$, $d = 11$). (B) Same as A but with $d \leq 3$ (same as $d < 7$ in this case). (C-D) Empirical FDR on the entire “combined” set.
A. Knockoff+ vs. multi-knockoff

B. Batched-knockoff+ vs. multi-knockoff

C. Mirror vs. multi-knockoff

D. Max vs. multi-knockoff

E. LBM vs. multi-knockoff

F. Multi-knockoff-select vs. multi-knockoff

Figure B.7: Power difference vs. multi-knockoff. Each panel shows the difference in power between one of the methods considered in this paper and multi-knockoff. All the methods were applied to all the datasets that are included the combined set described in Supp. Sec. B.1.11. Note that the scale of the y-axis varies across the panels.
A. The optimal # of knockoffs, $d$, varies (max)

B. The optimal $d$ varies (multi-knockoff)

C. Same as B but with multi-knockoff-select

D. multi-knockoff vs. multi-knockoff-select

E. knockoff+ vs. multi-knockoff-select

F. Empirical FDR (multi-knockoff-select)

FIGURE B.8: Varying the number of knockoffs. All the plots were created using the set of experiments defined in Supp. Sec. B.1.9. When applying any multiple-knockoff selection method to a randomly drawn dataset we used the increasing sequence of $d = 3, 7, 15, 31$ knockoffs so we can examine how the method’s power varies with $d$. (A-B) The optimal value of $d$ varies for the max method and multi-knockoff in this example where $n = 600$, $p = 200$, $K = 10$ with $A = 3.0$ and 0 feature correlation $\Theta_0 = I_p$. (C) multi-knockoff-select (black “ref” curve) seems to do a good job at tracking the near-optimal value of $d$ (same data as in panel B). (D) The difference in average power of multi-knockoff vs. multi-knockoff-select using all six datasets in Supp. Sec. B.1.9. (E) Same as in panel D but now the comparison is with knockoff+, which is evidently uniformly weaker than multi-knockoff-select. (F) Empirical evidence that multi-knockoff-select controls the FDR when applied to these datasets (same legend as panel E).
Appendix B. Knockoff Simulations

A. $\rho = 0$

B. $\rho = 0.5$

C. $\rho = 0.9$

D. Empirical FDR ($\rho = 0$)

Figure B.9: Comparison with model-X knockoffs. In this figure we examine the performance of the model-X knockoffs. The data consisted of $n = 600$, $p = 200$, $K = 10$, $A = 2.8$ and a varied feature correlation strength as described in Supp. Sec. B.1.12. (A-C) The model-X LSM is on-par or below knockoff+ while the model-X LCD is generally significantly behind both. At the same time multi-knockoff and multi-knockoff-select dominate the single knockoff methods except when the FDR threshold is $\geq 0.35$ and the feature correlation is very high ($\rho = 0.9$). (D) For this case of $\rho = 0$ the empirical FDR of LCD is always below 70% of the threshold hence it does not appear in the plot. Similarly, for $\rho = 0.5, 0.9$ the empirical FDR of all the methods was consistently significantly below the threshold so we omitted those figures.
Bibliography


