Removing Unwanted Variation from Microarray Data with Negative Controls

by

Johann Andreas Gagnon-Bartsch

A dissertation submitted in partial satisfaction of the
requirements for the degree of
Doctor of Philosophy

in

Statistics

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Terence P. Speed, Co-chair
Professor Philip B. Stark, Co-chair
Professor Sandrine Dudoit
Professor John Ngai

Fall 2012

Removing Unwanted Variation from Microarray Data with Negative Controls

Abstract

Removing Unwanted Variation from Microarray Data with Negative Controls

by

Johann Andreas Gagnon-Bartsch

Doctor of Philosophy in Statistics

University of California, Berkeley

Professor Terence P. Speed, Co-chair

Professor Philip B. Stark, Co-chair

Microarray expression studies suffer from the problem of batch effects and other unwanted variation. Unwanted variation complicates the analysis of microarray data, leading to high rates of false discoveries, high rates of missed discoveries, or both. Many methods have been proposed to adjust microarray data to mitigate the problems of unwanted variation. Because the factors causing the unwanted variation are frequently unknown, several of these methods rely on factor analysis to infer the unwanted factors from the data. A central problem with this approach is the difficulty in discerning the unwanted variation from the biological variation that is of interest to the researcher. To overcome this problem, we present novel methods that use *negative controls* to help identify the unwanted factors and separate the unwanted variation from the variation that is of interest. Negative control genes are genes known *a priori* not to be differentially expressed with respect to the biological factor of interest.

The first method we present is a simple two-step procedure that we name RUV-2. In the first step RUV-2 estimates the unwanted factors by performing factor analysis on the negative control genes. Here, RUV-2 exploits the fact that any variation in the expression levels of negative control genes can be assumed to be unwanted variation. In the second step, RUV-2 regresses the expression data on the factor of interest, including the estimated unwanted factors as covariates in the regression model. The principal difficulty with RUV-2 is choosing the number of unwanted factors to include in the model.

The second method we present is a more complicated four-step procedure that we name RUV-4. Compared to RUV-2, RUV-4 is relatively insensitive to the number of unwanted factors included in the model; this makes estimating the number of factors less critical. We also present a novel method for estimating the genes' variances that may be used even when a large number of unwanted factors are included in the model and the design matrix is full rank. We name this method the "inverse method for estimating variances." By combining RUV-4 with the inverse method, it is no longer necessary to estimate the number of unwanted factors at all.

We discuss various techniques for assessing the performance of an adjustment method, and compare the performance of RUV-2, RUV-4, and their variants with the performance of other commonly used adjustment methods such as Combat, SVA, LEAPP, and ICE. We present several example studies, each concerning genes differentially expressed with respect to gender in the brain. We find that our methods performs as well or better than other methods.

Contents

List of Figures

List of Tables

Acknowledgments

I would like to thank my advisors, Terry Speed and Philip Stark, for their encouragement, support, enthusiasm, and guidance. I am very thankful to have had such wonderful advisors. I would also like to thank the other members of my dissertation committee, Sandrine Dudoit and John Ngai, along with the other members of my qualifying exam committee, Haiyan Huang and Barbara Romanowicz, for their time and support, and for their helpful comments. In addition, I would like to thank Yuval Benjamini, Anne Biton, Julia Brettschneider, Dongseok Choi, Prabhakara Choudary, Darya Chudova, Francois Collin, Noureddine El Karoui, Jun Li, Winston Lin, Luke Miratrix, Philip Musk, Pierre Neuvial, Moshe Olshansky, Elizabeth Purdom, Matthew Ritchie, Mark Robinson, Keith Satterley, Jas Sekhon, Hui Shen, Leming Shi, Gordon Smyth, Tim Triche, Mark Vawter, Roel Verhaak, Juergen von Frese, Victoria Wang, Sue Wilson, Di Wu, and Ying Xu for helpful discussions, comments, and logistical support in the course of my research. Laurent Jacob was a close collaborator for much of my research, and deserves a special thanks.

Chapter 1

Introduction

Microarray expression studies are plagued by the problem of unwanted variation. In addition to the biological factor(s) that are of interest to the researcher, other factors, both technical and biological, influence observed gene expression levels. A typical example is a *batch effect*, which can occur when some samples are processed differently than others. For example, significant batch effects may arise when some samples are processed in a different laboratory, by a different technician, or even just on a different day (Leek et al., 2010; Scherer, 2009). Though infamous, batch effects are not the only source of unwanted variation. Other sources of unwanted technical variation can occur *within* batches and be just as problematic. Moreover, unwanted biological variation can be a problem as well.

Unwanted variation complicates the analysis of microarray data. Unwanted variation may lead to high rates of false discoveries, high rates of missed discoveries, or both. Consider a typical differential expression (DE) analysis in which a researcher wishes to learn which genes are associated with a particular factor of interest. For example, a researcher may wish to learn which genes are associated with a particular tumor type. If there are unwanted factors (e.g. batch) that are correlated with the factor of interest (tumor type), the confounding of batch and tumor type may lead to false discoveries. Conversely, if there are unwanted factors that are uncorrelated with tumor type, the unwanted variation may simply obscure any true association between tumor type and gene expression levels, and thus lead to missed discoveries.

The causes of unwanted variation are often partially or entirely unknown. In some cases, factors that cause unwanted variation are known, but cannot be easily or precisely measured. For example, the level of atmospheric ozone has been shown to affect microarray quality (Fare et al., 2003). In other cases, only proxies of the true unwanted factors may be known, and these proxies may give only a partial hint of the underlying unwanted variation. Consider batch effects. It is not "batch" itself that causes "batch effects," but rather some other physical variable that is correlated with batch. As a simple example, suppose changes in the operating temperature of the scanner lead to unwanted variation, and consider a study in which some samples are processed at lab A and the rest at lab B. If the scanner at lab A generally runs cooler than the scanner at lab B, this will lead to a "batch effect." However, if the operating temperature at lab A is itself quite variable, this will lead to additional within-batch unwanted variation. In this example batch is only a proxy for temperature, and it may be a poor one. Section A.1 in the appendix provides a brief example from the MAQC study of substantial within-batch unwanted variation. Of course, in many other cases, even proxy variables are unavailable, and the causes of unwanted variation are a complete mystery.

This complicates the removal of unwanted variation. If the factors causing the unwanted variation are unknown, or even just poorly measured, it becomes difficult to discern what variation may be attributed to the unwanted factors. It becomes correspondingly difficult to discern what variation may actually be attributed to the factor of interest. A researcher trying to remove unwanted variation may fail to remove all of the unwanted variation, may accidentally remove the variation of interest, or both.

Adjusting for unwanted variation when the causes of the unwanted variation are unknown is the central topic of this thesis. Our strategy is to use control genes. Negative control genes are genes whose expression levels are known *a priori* to be truly unassociated with the biological factor of interest. Conversely, positive control genes are genes whose expression levels are known *a priori* to be truly associated with the factor of interest. For example, if the factor of interest is the presence or absence of ovarian cancer, CyclinE would be a positive control. Negative control genes are in general harder to identify with certainty. So-called housekeeping genes are often good candidates, but not always. Another example of negative controls are the spike-in controls found on many microarray platforms.[1]

Negative control genes are used routinely to detect the presence of unwanted variation. However, as we will see, they can also be used to *adjust* for unwanted variation. We are not the first to make this observation. In particular, Lucas et al. (2006) have used negative control genes to adjust for unwanted variation. However, we do not believe the importance of control genes in adjusting for unwanted variation is widely recognized. Indeed, we believe the role of control genes is of central importance.

The structure of this thesis is as follows. In what remains of the introduction we present a brief summary of existing methods to adjust for unwanted variation. In Chapter 2 we present a simple, two-step procedure that uses control genes to adjust for unwanted variation. We also discuss techniques to evaluate the performance of methods that adjust for unwanted variation. In Chapter 3 we present more complex methods that build on the simple method introduced in Chapter 2. Chapter 4 concludes.

Methods to adjust for unwanted variation can be divided into two broad categories. In the first category are methods that can be used quite generally, and provide a *global adjustment*. A global adjustment produces a modified (adjusted) dataset that is essentially identical to the original dataset but — hopefully — with the unwanted variation removed. An example of a global adjustment would be quantile normalization, which is commonly

[1] Two notes on terminology: 1) When we refer generally to "negative control genes" or simply "control genes" we often use the term to include the spike-in control probesets as well, despite the fact they are not genes. 2) In other contexts (e.g. Illumina) the term "negative controls" is used to denote probes that should never be expressed in any sample. This usage of the term "negative controls" is different than ours, and should not be confused.

used in the preprocessing of microarray data. Quantile normalization is generally regarded as a self-contained step, and plays no role in the downstream analysis of the data. In the second category of adjustment methods are *application specific* methods. Application specific methods integrate the adjustment for unwanted variation directly into the main analysis of interest. For example, in a differential expression study, batch effects may be handled by explicitly adding batch terms to a linear model. A modified (adjusted) dataset is not created in the process. Thus, this method is application specific in the sense that it is only useful in the context of a differential expression analysis. It is not necessarily clear how — or whether — the method may be altered in order to adjust for unwanted variation in other types of analyses, such as classification or clustering analyses.

Much of the progress that has been made in removing unwanted variation from microarray data has been with application specific methods intended for use in differential expression analyses. Most of these methods make use of linear models. In some of these methods it is assumed that the factors causing the unwanted variation are known; in this case the effects of the known batches can be directly modeled. Combat is one such successful and well-known method; in particular Combat has been shown to work well with small datasets (Johnson et al., 2007). While Combat and other similar methods can be quite successful, their use is limited by the assumption that the unwanted factors are known. As discussed above, the unwanted factors are often only partially known, or entirely unknown.

Other linear-model based methods presume the sources of the unwanted variation to be unknown. These methods attempt to infer the unwanted variation from the data, and then adjust for it. Often this is accomplished via some form of factor analysis; several factors believed to capture the unwanted variation are computed and then incorporated into the model in just the same way known confounders are incorporated. In the simplest approach, factors are computed directly from the observed expression matrix by means of a singular value decomposition or some other factor analysis technique. This is often successful in practice, but can be dangerous — if the biological effect of interest is large, it too will be picked up by the factor analysis, and removed along with the unwanted variation. In other words, if one adjusts for unwanted variation by removing the first several principal components (PCs) from the data, one may very well throw out the baby with the bathwater. This problem has been acknowledged, and several attempts have been made to avoid it. One of the most well-known methods that directly address this problem is SVA (Leek and Storey, 2007, 2008). Other methods of potential interest include Kang et al. (2010), Kang et al. (2008a), Kang et al. (2008b), Listgarten et al. (2010), Mecham et al. (2010), Price et al. (2006), Stegle et al. (2008), Yu et al. (2005). Some of the first uses of factor analysis to adjust for unwanted variation can be found in Alter et al. (2000) and Nielsen et al. (2002), although in these examples there is no explicit linear model.

Chapter 2

RUV-2

2.1 Introduction

In this chapter we present RUV-2. RUV-2 is a simple, two-step procedure that uses control genes to detect and adjust for unwanted variation. The two steps of RUV-2 are: 1) perform factor analysis on the negative control genes to estimate the unwanted factors, and 2) incorporate the estimated unwanted factors into a linear regression model (along with any known covariates, and, of course, the factor of interest). RUV-2 exploits the fact that the negative controls are not truly associated with the factor of interest; any observed variation in the negative controls can be assumed to be unwanted variation. By constraining the factor analysis to the control genes, there is no danger in picking up any of the relevant biology in the factor analysis step, and thus no danger of throwing out the baby with the bathwater.

RUV-2 is application specific and meant for use in differential expression studies. However, there is an obvious interest in adapting RUV-2 for use in other applications, such as classification or clustering. Thus, a secondary goal of this chapter is to explore the possibility that RUV-2 might be adapted for use in other applications. We find that there may be some hope of success, but substantial challenges remain.

A third goal of this chapter is to present some techniques we have found to be useful for comparing the performance of different adjustment methods. Many methods have been proposed to adjust microarray data for unwanted variation, and several of these methods have been quite successful. However, there is no "silver bullet" and probably never will be. As such, there is a crucial need for techniques to evaluate the relative strengths and weakness of the various adjustment methods that are available.

The structure of this chapter is as follows. In Section 2.2 we discuss techniques to evaluate the performance of adjustment methods. In Section 2.3 we apply RUV-2 to some real datasets. We compare the performance of RUV-2 to that of other adjustment methods. In Section 2.4 we present the methodology of RUV-2 in more detail. Section 2.5 concludes.

2.2 Criteria for a Good Adjustment

Techniques to evaluate the quality of an adjustment are in many ways as important as the adjustment method itself. The statistical models on which adjustment methods are based are artificial. The models are most useful as sources of inspiration for improved methods; they are substantially less useful in proving the worth of a method. In the end, an adjustment method must prove its value by working in practice.

The question thus arises of how to know whether an adjustment is helping or hurting. This is not trivial. In many cases, evidence that seems to suggest an adjustment method is helping (or hurting) is actually ambiguous. As an example, consider a differential expression study, and consider assessing the quality of the study by counting the number of genes "discovered" at a certain FDR threshold. If the unwanted variation is roughly orthogonal to the factor of interest, the unwanted variation will manifest itself as additional "noise" that obscures any true association between the factor of interest and gene expression levels. An effective adjustment method would therefore increase the number of discovered genes. On the other hand, if the unwanted variation is correlated with the factor of interest, this will introduce spurious associations between the factor of interest and gene expression levels. An effective adjustment method would therefore decrease the number of discovered genes. As a second example, consider a classification study in which a researcher wants to classify tumor samples into one of several tumor sub-types. Suppose the researcher wants to test her classification algorithm on a set of tumor samples in which the sub-type is known, and she does her test once in combination with a method that adjusts for unwanted variation, and once without. Suppose the rate of misclassification is higher when the adjustment method is used. This would seem to suggest the adjustment method is hurting. However, it is equally possible that the adjustment method is working — if the tumor sub-types were processed in batches, the resulting batch effects could artificially help the classifier.

In the following few sections we present some techniques that we have found to be useful in evaluating the quality of an adjustment.[1] Only the first technique provides a (nearly) unambiguous assessment of the quality of an adjustment. However, its applicability is limited. The other two techniques are also very informative, if not entirely definitive, and can be used in a wider variety of situations.

[1] One fairly common technique that we do *not* discuss is clustering. In some circumstances, performing a cluster analysis both before and after adjustment and observing whether the adjustment causes samples to cluster more strongly by biology (or less strongly by batch) can be a highly effective way to assess the quality of the adjustment. Indeed, we use this technique ourselves in Section 2.3.4. However, there are also many circumstances in which clustering can be deceptive (the classification example we provide above applies just as well to clustering). We feel a full discussion is beyond the scope of this thesis. In general, we feel that clustering is more helpful as an exploratory tool, and less helpful as a test of the quality of an adjustment.

2.2.1 Control Genes / Gene Rankings

Positive control genes can be used for quality assessment in differential expression studies. After computing p-values for each gene, we can rank the genes in order of increasing p-value. Positive controls should be towards the top of this list. We can therefore use the number of positive controls ranked in (for example) the top 50 genes as a quality metric. If an adjustment method substantially increases the number of top-ranked positive controls, we have reason to believe the method is effective.[2]

Note that we use the ranks of the p-values and not the p-values themselves. This is for reasons discussed above; a good adjustment may increase or decrease the positive controls' p-values depending on the nature of the unwanted variation. Ranking helps to resolve the ambiguity.

While ranking p-values is generally preferred, there remain some situations in which it may be better to look at the p-values themselves. An example might be when only a very small number of positive controls are available, and their rankings do not change substantially after the adjustment. In this case, one might wish to examine the p-values of both positive and negative controls. If the p-values of the positive controls substantially decrease — and the p-values of the negative controls do not — this would suggest the adjustment helps. On the other hand, if the p-values of the negative controls decrease as well, this may simply suggest an artifact of the adjustment method. Likewise, if the p-values of the positive and negative controls both increase, the result is ambiguous, but the technique decribed in 2.2.2 might help clarify matters.

Some caution is required when using negative controls to assess the quality of an adjustment. After all, if the method of adjustment is to fit and remove variation characteristic of a set of negative controls, then observing that the adjustment diminishes the association between the factor of interest and the negative controls is simply to be expected. A better strategy would be to use two different sets of negative controls — one to make the adjustment, and one to use in assessing the quality of the adjustment. Preferably, the two sets of negative controls will be different from each other in some important way. For example, we might use spike-in controls to make an adjustment, and housekeeping genes to assess the quality of the adjustment, or vice versa.

2.2.2 The p-value Distribution

Consider a differential expression study in which the factor of interest is assumed to be associated with the expression level of only a fraction of genes. The distribution of the p-values for the genes that are unassociated with the factor of interest would ideally be

[2] Even here, however, the evidence, strictly speaking, is not entirely conclusive. For example, consider a simple hypothetical adjustment method that simply shinks gene-specific variances to a common average variance. This would have the effect of systematically increasing the magnitude of the t statistics for highly variable genes. If the positive controls all happened to be highly variable, the end result would be that the rankings of the positive controls improve, despite the fact that the "adjustment" is not really correcting for unwanted variation at all.

uniformly distributed over the unit interval, whereas the p-values for the genes that are associated with the factor of the interest will ideally be nearly 0. Thus, a histogram of the p-values will ideally be nearly uniform, with a spike near 0. In practice however, this is uncommon, as unwanted variation tends to introduce dependence across measured gene expression levels. Since adjusting for unwanted variation should remove this dependence, we might expect a good adjustment to result in p-value histograms closer to the "ideal."

2.2.3 RLE Plots

RLE (relative log expression) plots are boxplots that can be used to determine the overall quality of a dataset, and, in particular, identify bad chips. Consider a set of m chips, each with n genes, and denote the log expression level of the j^{th} gene on the i^{th} chip by y_{ij}. Denote the j^{th} column of the matrix (y_{ij}) by $y_{\star j}$. For each of the n genes we can calculate $\text{median}(y_{\star j})$, the median (over the m chips) log expression level. For each gene on each chip, we can then calculate $y_{ij} - \text{median}(y_{\star j})$, the deviation from the median gene expression level. For each chip, we can then produce a box plot of its n deviations. In most cases, if the chip is of good quality, this boxplot will be centered around zero and its width (IQR) will be around 0.2 or less. Examples of RLE plots can be found in Figure 2.1 (to be discussed more fully later), and more information about RLE plots can be found in Bolstad et al. (2005), Brettschneider et al. (2008).

2.3 Examples

We present four examples. In the first three we discover genes that are differentially expressed in the brain with respect to gender. The fourth example involves clustering tumors of known types. We chose these examples because "truth" is in some sense known.

We chose differential expression with respect to gender because it provides us with a clear set of potential positive controls — in this case, genes that are located on the X and Y chromosomes. Treating X and Y genes as positive controls and using our technique of counting the number of top-ranked positive controls (Section 2.2.1) allows us to compare the performance of various adjustment methods. We chose the brain because of its comparatively complex biology — a very large fraction of genes are expressed in the brain — and the availability of several interesting datasets. Indeed, our lead example is ideal, as the study was originally intended to discover differentially expressed genes in the brain,[3] and the data exhibit profound batch effects.

In all of our examples, a few practical issues must be considered. The first is what preprocessing should be done. Microarray data routinely go through three stages of preprocessing — background correction, normalization, and summarization. Several algorithms have been

[3] In this thesis we are primarily concerned with methodology, not biology, and we do not discuss the specific genes we find to be differentially expressed. Readers interested in the particular genes that are differentially expressed can find tables in Sections A.3, A.4, and A.5 of the appendix.

proposed for each of these steps. For simplicity, we limit ourselves to the "standard" sequence of algorithms used in RMA (Bolstad et al., 2003), (Irizarry et al., 2003a), (Irizarry et al., 2003b). The preprocessing steps — particularly the quantile normalization — are nonlinear, and it is not clear how they might interact with the adjustment methods. Thus, we repeat many of our analyses omitting one or more stages of preprocessing in order to see what happens.

The second issue to consider is which negative controls to use. To effectively adjust for batch effects, our negative controls must both 1) be uninfluenced by the factor(s) of interest and 2) be influenced by the unwanted factors. In other words, they must actually be negative controls, and their expression levels must accurately reflect the unwanted variation. We focus on two classes of possible negative controls — housekeeping genes and spike-in controls. The housekeeping genes we use are those discovered in Eisenberg and Levanon (2003).[4] Information regarding the spike-in controls can be found in the appendix of the Affymetrix GeneChip Expression Analysis Technical Manual (Affymetrix, 2005-2009). A good discussion of both can be found in Lippa et al. (2010).

2.3.1 Gender Study

Vawter et al. (2004) conducted a study to find genes differentially expressed in the brain with respect to gender. Samples were taken post-mortem from the brains of 10 individuals, 5 men and 5 women. Three samples were taken from each individual — one from the anterior cingulate cortex, one from the dorsolateral prefrontal cortex, and one from cortex of the cerebellar hemisphere. One aliquot of each sample was sent to each of three laboratories for analysis. The analyses were done using either Affymetrix HG-U95A or Affymetrix HG-U95Av2. We are unaware of how the decision was made to use which platform for which analysis. One of the laboratories used only HG-U95Av2. Note that there should have been $10 \times 3 \times 3 = 90$ chips total. However, six of the combinations were missing, leaving us with 84 chips.[5] Data are available on GEO (GSE2164).

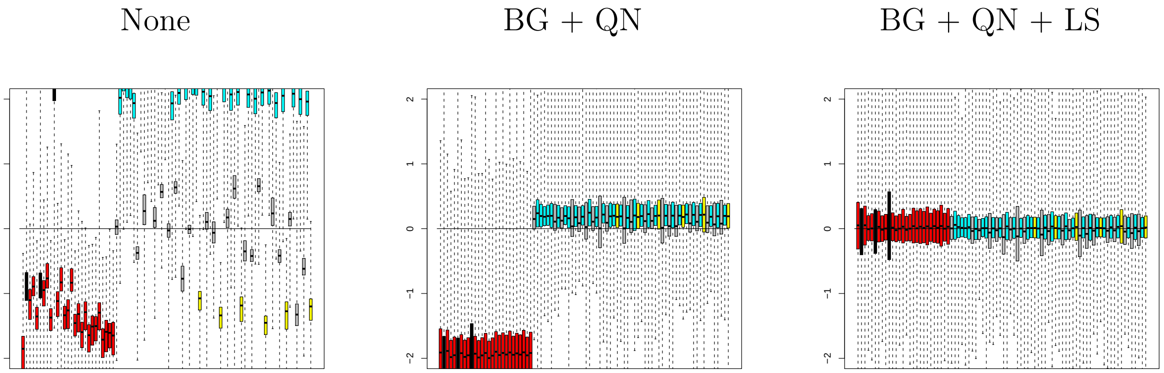
The HG-U95A platform has 12626 probesets, and the HGu95Av2 platform has 12625 probesets. We identified 12600 probesets that were shared between the two platforms. We did not, however, attempt to map individual probes from one platform to the other. Since preprocessing (background correction, quantile normalization, summarization) requires probe level data, we did these preprocessing steps for each platform separately. As a result, large differences remained between the different platform types even after the standard preprocessing.[6] The raw HG-U95Av2 expression values are generally larger than their HG-U95A

[4] Housekeeping genes are genes that are essential to basic cellular activities, such as metabolism. These genes are therefore expressed in all cells. The strategy used in Eisenberg and Levanon (2003) to discover housekeeping genes was to examine gene expression levels in many different tissues, and see which genes were expressed in all of the tissues.

[5] Additionally, 3 of the combinations were replicated, so in fact there are 87 chips available on GEO. We omitted the replicates in our analysis.

[6] In this respect, the situation is similar to the one found in Nielsen et al. (2002), one of the first uses

analogs by about a factor of 4, so the $\log_2$ expression values for the HG-U95Av2 expression values are generally greater than those of the HG-U95A by about 2. We therefore added an additional preprocessing step after summarization; we performed a location / scale adjustment in which we linearly re-scaled the data so that each chip had the same mean and standard deviation. See Figure 2.1 for RLE plots at different stages of preprocessing — no preprocessing; background correction and quantile normalization only (BG + QN); background correction, quantile normalization, and location / scale adjustment (BG + QN + LS). It is important to note that the vertical scale of the RLE plots in Figure 2.1 is substantially different than that of all other RLE plots in this thesis.



None

BG + QN

BG + QN + LS

Figure 2.1: Gender study RLE plots at different stages of preprocessing. From left to right: No preprocessing; background correction / quantile normalization done separately for each platform type; background correction / quantile normalization followed by a final location/scale adjustment across all chips. Coloring: red – site A, HG-U95A; yellow – site A, HG-U95Av2; black – site B, HG-U95A; gray – site B, HG-U95Av2; cyan – site C, HG-U95Av2. NOTE: The scale on the y-axis is different for these RLE plots than for all other RLE plots in this thesis.

The unwanted variation apparent in Figure 2.1 is striking. Even ignoring the differences between chip types, observed expression levels differ by up to an order of magnitude between laboratories. There is substantial within-laboratory unwanted variation as well. The average observed expression level varies from chip to chip by roughly a factor of two within laboratories. As a result, discovering genes that are differentially expressed with respect to gender is nearly impossible without adjusting for the unwanted variation. On un-preprocessed, unadjusted data, every gene has an FDR-adjusted p-value of approximately 1, and only 7 of the top 60 genes come from the X or Y chromosome. The preprocessing helps; after preprocessing (but no other adjustments), 15 of the top 60 genes are from the X or Y chromosome,

of factor analysis to remove unwanted variation, where the primary source of unwanted variation was chip type.

and 8 of these have FDR-adjusted p-values that are significant at the 0.05 level. Even after preprocessing, however, substantial unwanted variation persists, as can be seen in RLE, p-value, and scree plots.

A critical step in RUV-2 is determining the number k of factors to remove. In general, this is difficult, and there is no clear way to determine k. We recommend pursuing several approaches and exercising judgment. We have found RLE plots and p-value histograms to be helpful.[7] In addition, if any positive controls are known, these should be used as well. To make use of RLE plots and p-value plots it is necessary to complete the analysis for several values of k and then examine the plots to evaluate the quality of the results. Several such plots can be found in the appendix (Figures A.3 and A.4); based on these plots, we choose a k of 10 — see Figure 2.2.

Several questions remain: Which factor analysis method is best? To what extent does performance depend on k? Does RUV-2 work well in combination with Limma? Do we achieve a better adjustment using housekeeping genes or spike-in controls? How does RUV-2 compare to other adjustment methods such as standard linear regression, Combat, or SVA? How does preprocessing affect performance? We address each of these questions in turn.

Given the central role of factor analysis in RUV-2, one might expect that the choice of factor analysis method is quite important. In our examples this turns out not to be the case. We repeated our analysis with three different factor analysis methods. The first is the singular value decomposition (SVD). The second is an EM algorithm based on a relatively simple probabilistic model that allows for gene-specific variances in the error term. The model and the algorithm are described in Section 12.2.4 of Bishop (2006). The implementation of the algorithm is our own. The third method is a "robust" method described in Hubert et al. (2005) and implemented in the `PcaHubert` fuction of the R package `rrcov`. RLE plots after adjustment looked nearly identical for all three factor analysis methods. P-value histograms were also remarkably similar. More convincingly, the number of top-ranked X/Y genes is roughly the same for all three methods, as we discuss below.

The choice of k turns out to be critical to the performance of RUV-2. This can be seen in RLE plots and p-value histograms (figures are in the appendix), but can be seen most dramatically by counting the number of top-ranked X/Y genes at different values of k. We repeated the analysis for all possible values of k. At first, the number of top-ranked X/Y genes increases with increasing k, as additional unwanted variation is removed. After a certain point, however, increasing k only degrades performance, as adding additional factors to the model simply increases variance.[8] See Figure 2.3. Note from Figure 2.3 that although the performance of RUV-2 depends critically on the choice of k, the region over which RUV-2 performs well is fairly large. This is important because it implies, in this example at least, that while RLE plots and p-value histograms may not lead us to the single best choice of k,

[7] Many people find scree plots to be helpful as well (Venables and Ripley, 2002). In our examples, we found scree plots to be of questionable value. However, we provide them in the appendix.

[8] Note that as $k \to m$, we keep adjusting away additional dimensions, until we finally remove everything. At this point the OLS estimator becomes undefined. Thus, performance will certainly degrade for large k. The only question is when.

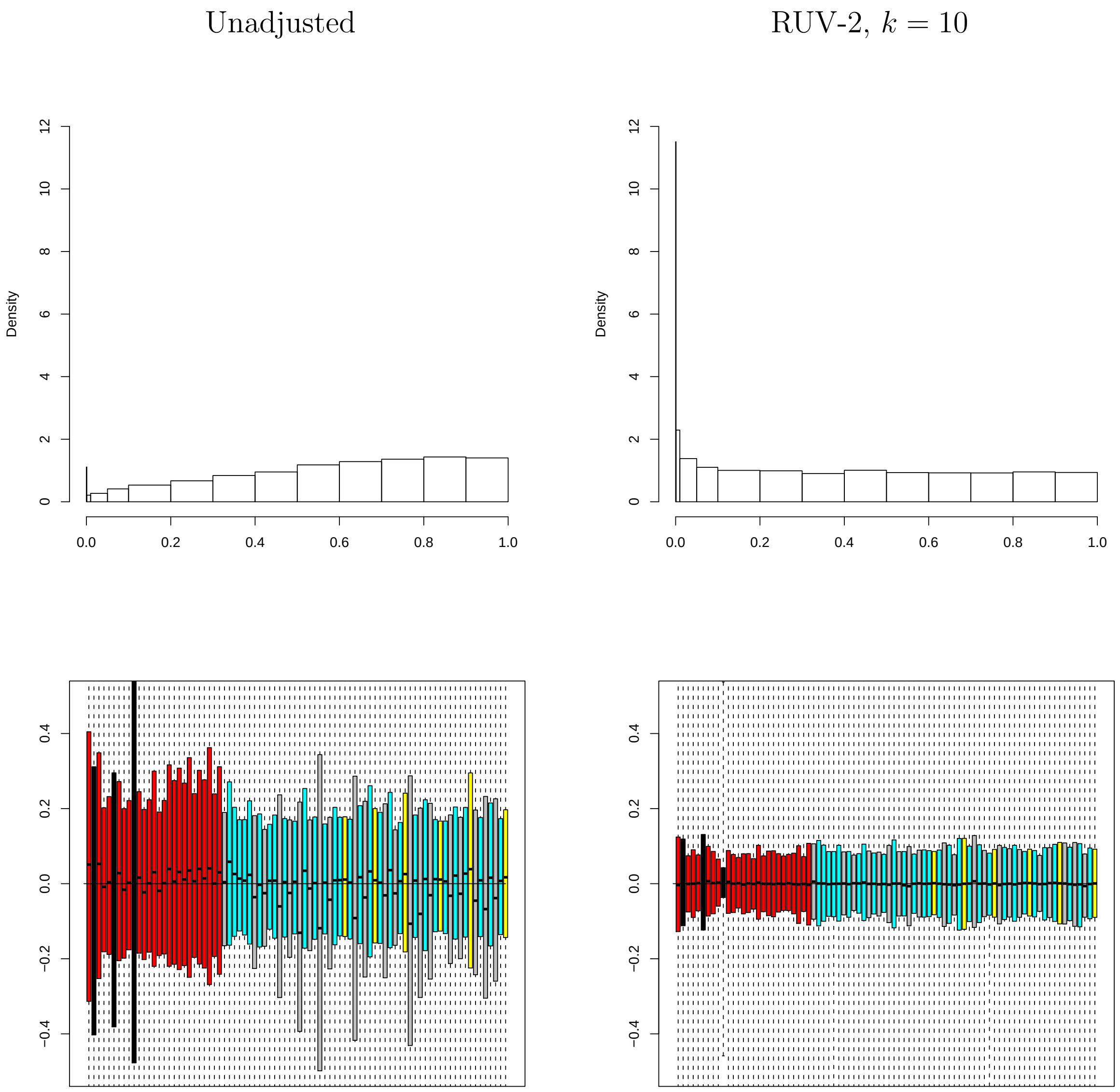


Figure 2.2: Gender study p-value histograms and RLE plots before and after adjustment. Histogram breakpoints are at 0.001, 0.01, 0.05, and 0.1, 0.2, 0.3, etc. Data were fully preprocessed (BG+QN+LS). The factors were computed by SVD on the housekeeping genes. P-values were computed using Limma.

they can at least lead us to a good one. Finally, note also from Figure 2.3 that the choice of factor analysis method does not greatly impact the results.

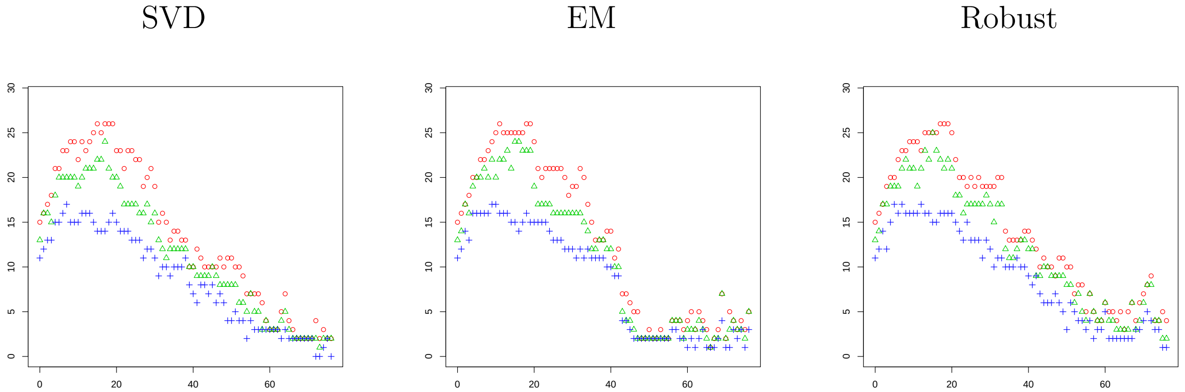


Figure 2.3: Comparison of performance of different factor analysis methods in the gender study. The number of X / Y genes discovered is plotted as a function of k. Genes were ranked by p-value; results are shown for the number of X / Y genes ranked in the top 20 (plus), top 40 (triangle), and top 60 (circle). $k = 0$ corresponds to no adjustment. Data were preprocessed (BG + NM + LS). PCs were computed using the housekeeping genes. P-values were computed using Limma.

It is common to analyze microarray data using the Limma package in Bioconductor (Smyth, 2004). Limma uses empirical Bayes methods to improve the estimates of the variances of individual genes. Since adjusting with RUV-2 or any other method can substantially affect the general structure of the residuals, we checked to ensure that RUV-2 and Limma work well together. To accomplish this we completed our analysis once using Limma and once using "ordinary" regression. We did not notice any substantial difference in performance (as assessed using RLE plots, p-value histograms, and counts of top-ranked X/Y genes) between the two methods. Details can be found in the appendix.

The performance of RUV-2 depends greatly on the choice of negative controls. Adjustments based on both the housekeeping genes and the Affymetrix spike-in controls improved performance relative to unadjusted data, however the performance increase using housekeeping genes was substantially better. See Figure 2.4. We believe housekeeping genes may outperform the spike-ins in this example for three reasons. Firstly, housekeeping genes may be able to capture unwanted biological variation, whereas spike-in controls can only capture unwanted technical variation. Secondly, even with regards to technical variation, housekeeping genes may be more "representative" than spike-ins. For example, housekeeping genes may capture unwanted variation introduced during sample collection, whereas spike-ins would not. Conversely, spike-ins may exhibit some unique variation related to their own administration. Lastly, there are far more housekeeping genes than there are spike-in controls. There are 799 probesets that correspond to one of the housekeeping genes in Eisenberg and Levanon (2003). However, there are only 33 probesets corresponding to spike-in controls.

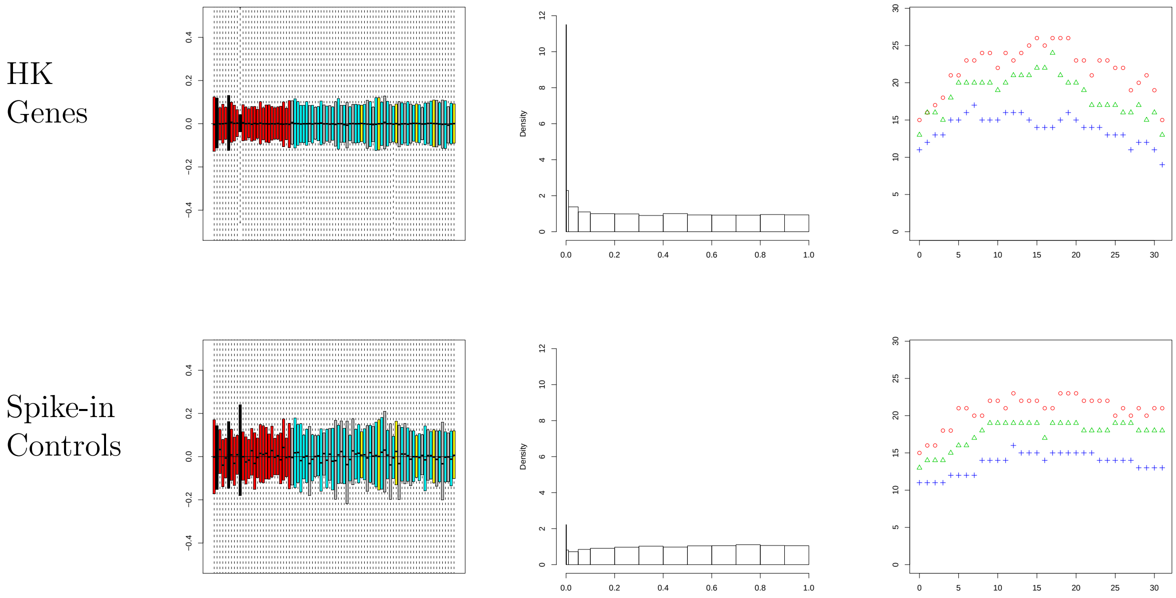


Figure 2.4: Comparison of results for housekeeping genes and Affymentrix spike-in controls in the gender study. For the RLE plots and p-value histograms, $k = 10$. Factors were computed by SVD. P-values were computed using Limma. Note that there are only 33 spike-in controls, so adjustments with $k > 33$ are undefined for the spike-in case. We truncate results in the housekeeping case as well for easy comparison.

Finally, we wish to compare the performance of RUV-2 to that of other adjustment methods. Additionally we would like to investigate the effects of preprocessing. We use the number of top-ranked X/Y genes as our basis for comparison,[9] and present the results in Table 2.1. Several observations merit mention. RUV-2 outperforms Combat and ordinary regression in all cases, and SVA outperforms them in several cases as well. This is despite the fact that Combat and ordinary regression explicitly model known batches (lab / chip type), whereas RUV-2 and SVA infer the unwanted variation from the data. Moreover, it seems that even when we do explicitly model known batches with RUV-2 by including a "Z" term in the model (see Section 2.4) there is no substantial increase in performance.[10] Another observation is that the level of preprocessing does not seem to matter for the SVD and EM

[9] Additionally, RLE plots and p-value histograms for Combat and SVA are given in the appendix.

[10] It is true that more X / Y genes are found when we include a Z term in the "Top 40" and "Top 60" cases, but this should be interpreted with caution; including a Z term results in adjusting for three additional factors; similar increases in performance can be seen by dropping the Z term and increasing k to 13 — see Figure 2.3.

(but not robust) variants of RUV-2, matters slightly for Combat, and matters greatly for the other methods. This seems to suggest that, at least in some cases, RUV-2 can obviate the preprocessing. While this is of little immediate value in the current example, it may be useful in situations where there is concern that the nonlinearities introduced by preprocessing are problematic. For example, if a very large number of genes are differentially expressed with respect to the factor of interest, the nonlinearities in the quantile normalization may induce an artificial correlation between the negative control genes and the factor of interest. This would violate the assumptions of RUV-2, and create problems. In such a situation it may be best to skip the quantile normalization. In any case, we regard the fact that RUV-2 performs well even without preprocessing as quite encouraging; after all, the un-preprocessed data are extremely noisy. Finally, we observe that in general, the best performing methods are the SVD and EM variants of RUV-2.

		No Preproc.		BG + QN		BG, QN, LS	
		Std	Lim	Std	Lim	Std	Lim
Top 20	No Adjustment	7	6	9	9	11	11
	Regression (Z)	5	5	14	13	12	12
	SVA (IRW)	5	6	12	11	14	14
	SVA (Two-Step)	NA	NA	NA	NA	16	16
	Combat	11	11	13	13	12	13
	RUV-2 — SVD ($k = 10$)	15	15	15	15	15	15
	RUV-2 — SVD ($k = 10$), w/Z	14	14	15	15	16	15
	RUV-2 — EM ($k = 10$)	**17**	**17**	**17**	**17**	**17**	**17**
	RUV-2 — EM ($k = 10$), w/Z	16	16	16	16	**17**	16
	RUV-2 — Robust ($k = 10$)	8	7	**17**	16	**17**	16
		No Preproc.		BG + QN		BG, QN, LS	
		Std	Lim	Std	Lim	Std	Lim
Top 40	No Adjustment	7	7	12	12	13	13
	Regression (Z)	6	6	15	15	16	15
	SVA (IRW)	6	7	13	14	17	16
	SVA (Two-Step)	NA	NA	NA	NA	21	20
	Combat	14	12	17	17	17	17
	RUV-2 — SVD ($k = 10$)	22	22	21	19	20	19
	RUV-2 — SVD ($k = 10$), w/Z	**23**	**23**	22	**22**	**22**	**22**
	RUV-2 — EM ($k = 10$)	22	22	**23**	**22**	**22**	20
	RUV-2 — EM ($k = 10$), w/Z	22	22	22	**22**	**22**	**22**
	RUV-2 — Robust ($k = 10$)	11	10	21	21	**22**	21
		No Preproc.		BG + QN		BG, QN, LS	
		Std	Lim	Std	Lim	Std	Lim
Top 60	No Adjustment	7	7	14	13	15	15
	Regression (Z)	7	7	16	16	16	17
	SVA (IRW)	7	8	15	14	19	19
	SVA (Two-Step)	NA	NA	NA	NA	23	22
	Combat	16	15	18	18	19	19
	RUV-2 — SVD ($k = 10$)	24	23	22	21	22	22
	RUV-2 — SVD ($k = 10$), w/Z	**25**	**25**	24	24	24	**25**
	RUV-2 — EM ($k = 10$)	24	24	**25**	**25**	25	24
	RUV-2 — EM ($k = 10$), w/Z	24	23	**25**	24	**26**	**25**
	RUV-2 — Robust ($k = 10$)	12	11	**25**	**25**	24	24

Table 2.1: Summary of performance of different methods in the gender study. The number of X / Y genes found in the top 20 / 40 / 60 is shown for different levels of preprocessing (None; BG + QN; BG + QN + LS), different methods of computing p-values (standard and Limma), and different methods of adjustment (none, standard regression, SVA, Combat, and RUV-2). In the case of RUV-2, results are given for different methods of factor analysis (SVD, EM, robust). Additionally, we present results for models that include an explicit Z term, where Z is a matrix of dummy variables corresponding to site (A, B, or C) and chip type (HGU-95A or HGU-95Av2). For all RUV-2 methods, $k = 10$ and housekeeping genes were used as negative controls. Results for the two-step variant of SVA are not available in some cases, because the function exited with an error. The highest number in each column is shown in bold.

2.3.2 Alzheimer's Study

Blalock et al. (2004) conducted a microarray study to investigate patterns of gene expression in Alzheimer's patients. We learned of this dataset from a colleague soon after we completed our initial analysis of the gender study, and decided to see if we could use it to replicate our findings. We continued to use gender as the factor of interest (instead of Alzheimer's disease state) so that we had clear positive controls. Note that our choice of this dataset was thus rather arbitrary — many other brain studies would have been just as suitable — but this one had our attention, was publicly available, and seemed to exhibit a fair degree of unwanted variation. Data are available on GEO (GSE1297).

Samples were taken post-mortem from the hippocampus of 35 individuals suffering from various stages of Alzheimer's disease. Four samples were discarded by the authors of the study because the severity of the patients' disease was not clear, leaving 31 samples available for analysis. These samples were assayed using Affymetrix HG-U133A microarrays. We were unable to obtain clinical gender data for the samples, but found we could infer the gender using the expression levels of XIST and DDX3Y.[11]

Our analysis — and conclusions — closely parallel those of the gender study. Most of the details are provided in the appendix; here we simply highlight key results. One important difference between this study and the gender study is that in this study there are no known batch effects. Therefore standard regression and Combat are inapplicable. Moreover, a final location / scale pre-processing step is unnecessary, since there is only one chip type. A summary of the performance of SVA and RUV-2 is provided in Table 2.2.

In contrast with the gender study, we find that preprocessing improves performance in all cases. In addition, Limma provides a clear performance enhancement as well, particularly when preprocessing is omitted. This is as we might expect, since Limma is most helpful when the sample size is small, and here we have only 31 samples compared to the gender study's 84. Again, the best performance is attained with RUV-2. Note that in some cases, SVA actually decreases performance.

Finally, we note that, unlike in the gender study example, RUV-2 does not perform well with spike-in controls. A comparison between housekeeping genes and spike-in controls of RLE plots, p-value histograms, and the number of top-ranked X / Y genes can be found in Figure 2.5. Note that on the HG-U133A platform there are 45 probesets for the spike-in controls, and 1112 corresponding to the housekeeping genes in Eisenberg and Levanon (2003). We cannot say with confidence why the spike-in controls fail in this case, but we note that two possible explanations would be that most of the unwanted variation is biological in nature, or due to sample quality.

[11] Details are provided in the appendix. Note in particular that we found 19 women and 12 men, and this contradicts the statement in Blalock et al. (2004) that, of the original 35 subjects, 16 were female and 19 were male.

	Top 20				Top 40				Top 60			
	No Preproc.		BG + QN		No Preproc.		BG + QN		No Preproc.		BG + QN	
	Std	Lim	Std	Lim	Std	Lim	Std	Lim	Std	Lim	Std	Lim
No Adjustment	8	9	15	16	9	9	19	21	9	9	23	24
SVA (IRW)	8	9	16	16	10	13	17	19	12	13	18	19
SVA (Two-Step)	NA	NA	16	16	NA	NA	18	18	NA	NA	19	19
RUV-2 — SVD ($k = 10$)	**16**	**19**	**19**	**20**	**22**	**23**	**25**	**26**	**23**	**26**	26	26
RUV-2 — EM ($k = 10$)	15	17	**19**	**20**	18	22	24	25	**23**	25	25	**27**
RUV-2 — Robust ($k = 10$)	11	13	**19**	**20**	15	17	**25**	**26**	19	20	**27**	**27**

Table 2.2: Summary of performance of different methods in the Alzheimer's study. The number of X / Y genes found in the top 20 / 40 / 60 is shown for different levels of preprocessing (None, BG + QN), different methods of computing p-values (standard and Limma), and different methods of adjustment (none, SVA, and RUV-2). In the case of RUV-2, results are given for different methods of factor analysis (SVD, EM, robust). For all RUV-2 methods, $k = 10$ and housekeeping genes were used as negative controls. Results for the two-step variant of SVA are not available in some cases, because the function exited with an error. The highest number in each column is shown in bold.

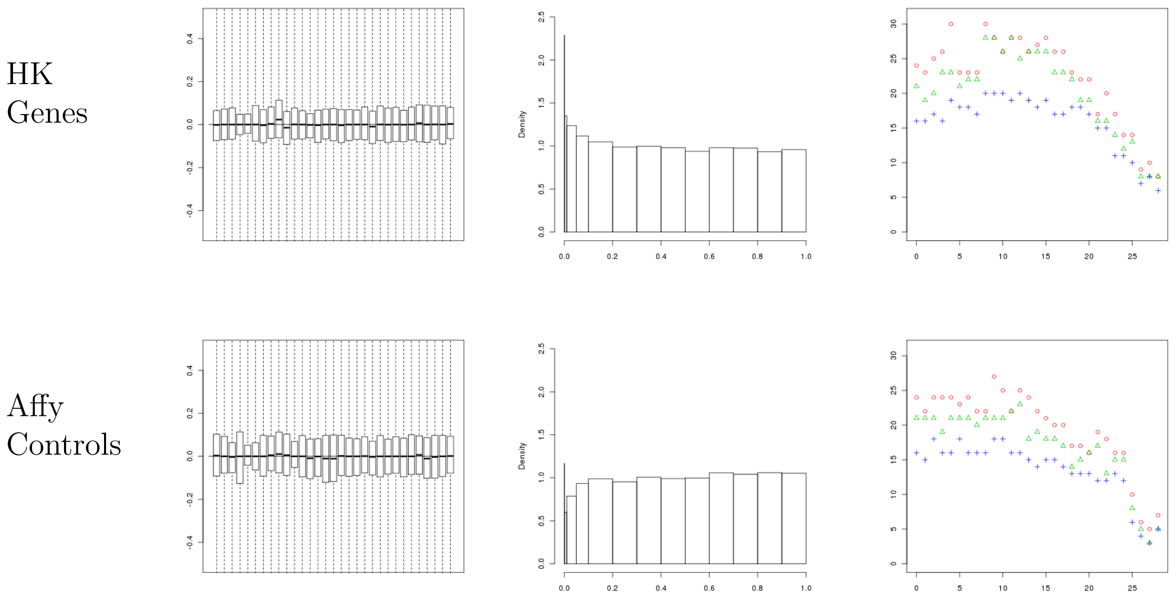


Figure 2.5: Comparison of results for housekeeping genes and Affymentrix spike-in controls in the Alzheimer's study. HK genes clearly perform better. For the RLE plots and p-value histograms, $k = 10$. Factors were computed by SVD. P-values were computed using Limma.

2.3.3 TCGA

The Cancer Genome Atlas (TCGA) is a large collaborative project established by the National Institutes of Health and managed by the National Cancer Institute and the National Human Genome Research Institute with the aim of collecting many types of data (e.g. expression, sequence, methylation) on a large number of samples from many different types of cancers. In particular, TCGA has already collected a few hundred glioblastoma multiforme (brain tumor) samples, and measured gene expression levels in these samples using (among others) the Affymetrix GeneChip Human Exon 1.0 ST array and the Affymetrix HT HG-U133A array. Because of the importance and size of this dataset, we decided it would be a good choice with which we could further replicate our findings from the gender study.

As with the Alzheimer's study, our analysis and conclusions closely parallel those of the gender study, and we limit our remarks here to highlight key results. Details can be found in the appendix. We discuss the exon array data first, followed by the HT HG-U133A data.

We downloaded exon array data for 386 samples from TCGA. Clinical gender data were available for 316 of the samples. We processed the data using aroma.affymetrix (Bengtsson et al., 2008) using a custom CDF[12] provided by colleagues at Lawrence Livermore National Laboratory. The custom CDF has 18632 probesets, corresponding to genes (not exons). The custom CDF does not include any probesets for spike-in controls, so we were only able to study adjustments using housekeeping genes. We identified 518 probesets as housekeeping genes.

Substantial unwanted variation is evident. The RLE plot of the un-preprocessed data (Figure A.21 in the appendix) suggests the presence of batch effects as well as additional within-batch variation. RLE plots and p-value histograms using various values of k (Figures A.22 and A.23 in the appendix) do not suggest a clear choice for k. The RLE plots suggest that k should be at least 30. The p-value histograms suggest that the unwanted variation skews p-values downwards, but a k of 100 is sufficient to solve this problem. Since a wide variety of values of k may be appropriate, we include results for both $k = 50$ and $k = 100$ in Table 2.3. Indeed, it turns out that results do not vary much over this wide range in k (see Figure A.26 in the appendix).

One of the more striking features of Table 2.3 is the relatively small gain achieved by adjustment. Only an additional 5 or so genes are discovered, despite the substantial unwanted variation. This may be because the large sample size compensates for the unwanted variation, allowing most of the differentially expressed genes to be found even without adjustment. Nonetheless, adjustment does help. RUV-2 and the two-step variant of SVA perform the best. A final interesting observation is that in this example, unlike in the gender study and Alzheimer's study examples, the robust variant of RUV-2 performs well even on un-preprocessed data.

We now turn to the HT HG-U133A data. We downloaded data for 414 samples from

[12] A CDF is a file that maps probes to probesets. A custom CDF is necessary because Affymetrix does not provide a CDF with their exon arrays. They provide a different file format, which is incompatible with aroma.affymetrix.

	Top 20				Top 40				Top 60			
	No Preproc.		BG + QN		No Preproc.		BG + QN		No Preproc.		BG + QN	
	Std	Lim	Std	Lim	Std	Lim	Std	Lim	Std	Lim	Std	Lim
No Adjustment	20	20	20	20	25	25	30	30	28	28	33	33
SVA (IRW)	20	20	20	20	28	29	30	30	30	30	32	32
SVA (Two-Step)	NA	NA	20	20	NA	NA	34	34	NA	NA	36	36
RUV2 — SVD ($k = 50$)	20	20	20	20	34	34	35	35	36	36	36	36
RUV2 — EM ($k = 50$)	20	20	20	20	35	35	**36**	**36**	37	37	**37**	**37**
RUV2 — Robust ($k = 50$)	20	20	20	20	33	33	35	35	37	36	36	36
RUV2 — SVD ($k = 100$)	20	20	20	20	**36**	**36**	35	35	**39**	**39**	**37**	**37**
RUV2 — EM ($k = 100$)	20	20	20	20	**36**	**36**	**36**	**36**	**39**	**39**	**37**	**37**
RUV2 — Robust ($k = 100$)	20	20	20	20	**36**	**36**	**36**	**36**	37	37	**37**	**37**

Table 2.3: Summary of performance of different methods for TCGA exon array data. The number of X / Y genes found in the top 20 / 40 / 60 is shown for different levels of preprocessing (None, BG + QN), different methods of computing p-values (standard and Limma), and different methods of adjustment (none, SVA, and RUV-2). In the case of RUV-2, results are given for different methods of factor analysis (SVD, EM, robust), and different values of k (50 and 100). Housekeeping genes were used as negative controls. Results for the two-step variant of SVA are not available in some cases, because the function exited with an error. The highest number in each column is shown in bold.

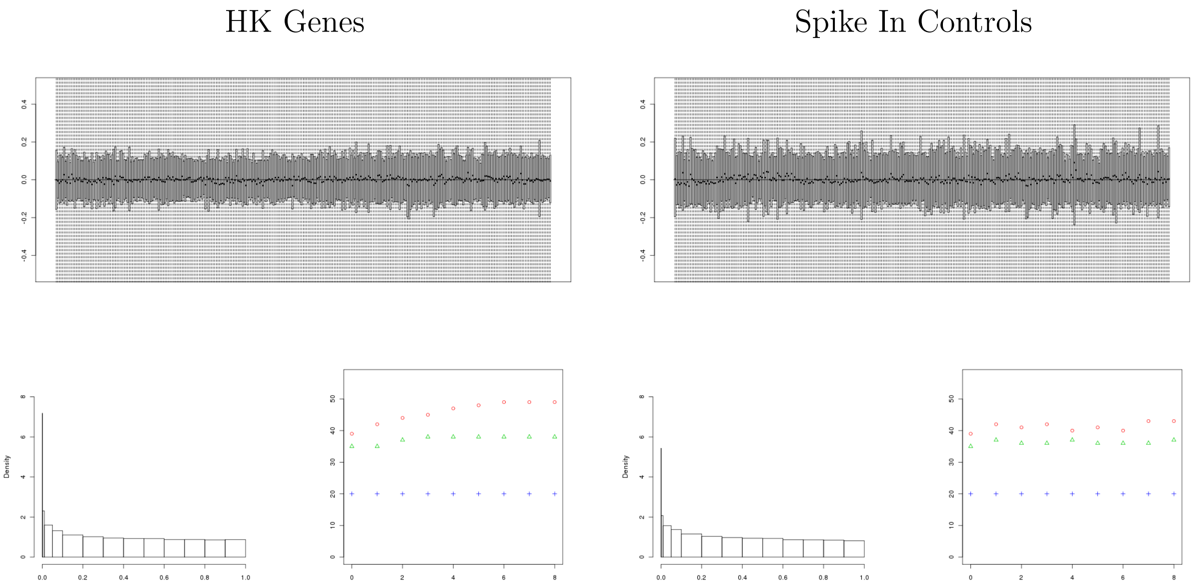
TCGA. Clinical gender data were available for 319 of these. Unlike with the exon array data, spike-in control data are available, but the HT HG-U133A only has 9 spike-in probesets. We identified 1045 probesets as housekeeping genes.

As with the exon array data, substantial unwanted variation is evident. Indeed, the RLE plot of the un-preprocessed data (Figure A.30 in the appendix) suggests the presence of very substantial batch effects. RLE plots and p-value histograms (Figures A.31 and A.32 in the appendix) suggest using a k of 30 is appropriate. Table 2.4 summarizes the results. Once again, RUV-2 and the two-step variant of SVA perform the best. Unlike with the exon array data, the robust variant of RUV-2 does not perform well on un-preprocessed data.

A comparison of the performance of spike-in controls and housekeeping genes is provided in Figure 2.6. As one might expect, considering there are only 9 spike-in controls, the housekeeping genes perform better.

	Top 20				Top 40				Top 60			
	No Preproc.		BG + QN		No Preproc.		BG + QN		No Preproc.		BG + QN	
	Std	Lim	Std	Lim	Std	Lim	Std	Lim	Std	Lim	Std	Lim
No Adjustment	14	14	20	20	21	22	35	35	27	27	39	39
SVA (IRW)	**20**	**20**	20	20	36	36	37	37	38	38	44	44
SVA (Two-Step)	NA	NA	20	20	NA	NA	**39**	**39**	NA	NA	**53**	**53**
RUV-2 — SVD ($k = 30$)	**20**	**20**	20	20	**38**	**38**	38	38	**47**	**47**	50	50
RUV-2 — EM ($k = 30$)	**20**	**20**	20	20	**38**	**38**	**39**	**39**	46	46	51	50
RUV-2 — Robust ($k = 30$)	**20**	**20**	20	20	31	30	**39**	**39**	37	35	51	51

Table 2.4: Summary of performance of different methods for the TCGA HT HG-U133A data. The number of X / Y genes found in the top 20 / 40 / 60 is shown for different levels of preprocessing (None, BG + QN), different methods of computing p-values (standard and Limma), and different methods of adjustment (none, SVA, and RUV-2). In the case of RUV-2, results are given for different methods of factor analysis (SVD, EM, robust). For all RUV-2 methods, $k = 30$ and housekeeping genes were used as negative controls. Results for the two-step variant of SVA are not available in some cases, because the function exited with an error. The highest number in each column is shown in bold.



HK Genes

Spike In Controls

Figure 2.6: Comparison of results for housekeeping genes and Affymentrix spike-in controls in the TCGA HT HG-U133A data. Housekeeping genes clearly perform better. For the RLE plots and p-value histograms, $k = 30$. Factors were computed by SVD. P-values were computed using Limma. Note that there are only 9 spike-in controls, so adjustments with $k > 9$ are undefined for the spike-in case. We truncate results in the housekeeping case as well for easy comparison.

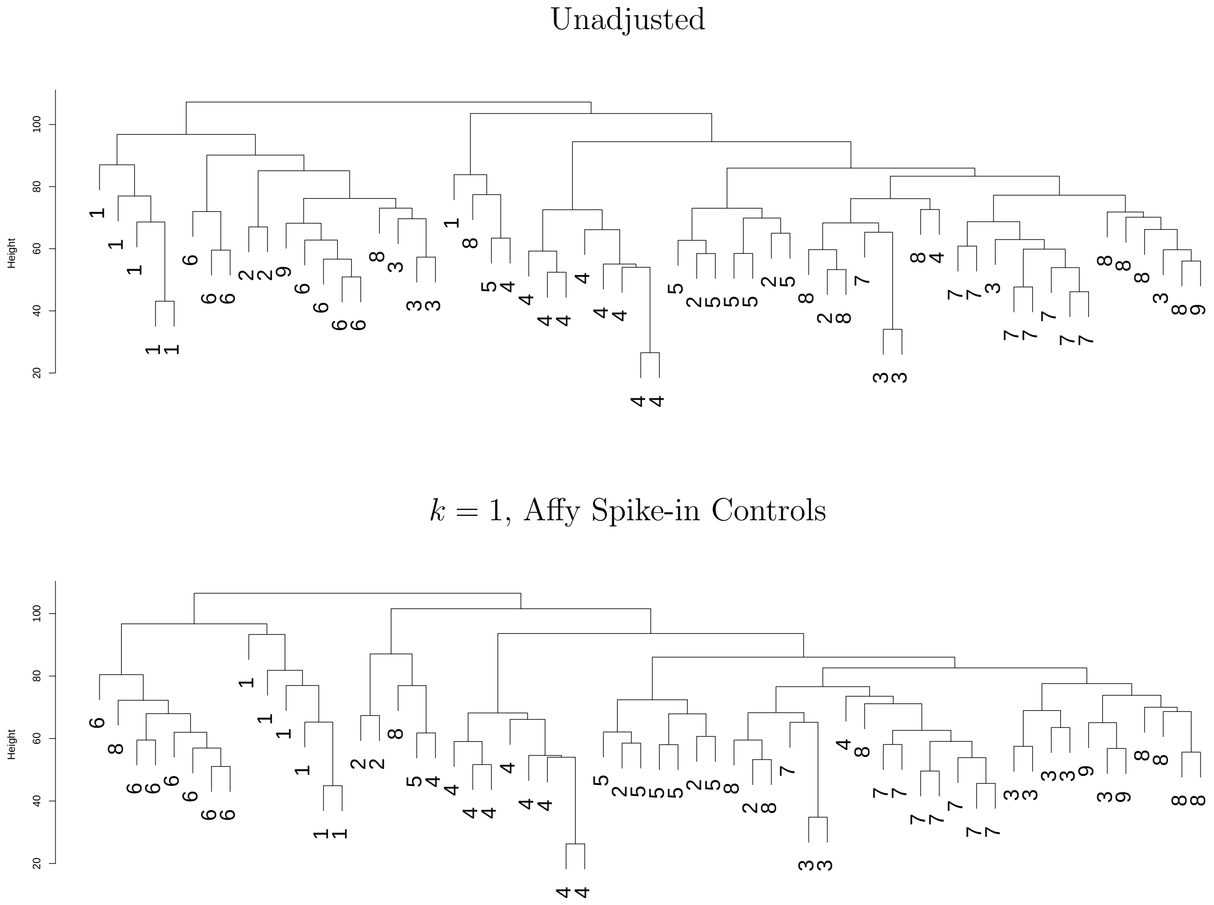
2.3.4 NCI-60

As discussed in Section 2.1, RUV-2 is an *application specific* adjustment method intended for use in differential expression studies. A natural question is whether the method can be adapted to provide a global adjustment, in which the entire dataset is adjusted to provide a modified dataset that can then be used just like the original dataset in any subsequent analysis. The advantages of such a method are obvious — not only would it allow one to use the method for applications other than differential expression (e.g. for classification or clustering), but the self-contained nature of the method would allow one to easily insert the algorithm into pre-existing code, and thus easily re-visit past analyses. Unfortunately, adapting RUV-2 to provide a global adjustment is not trivial. In this section we consider a naive adaptation of RUV-2 that provides a global adjustment and demonstrate that there may be some promise in the approach, but that there are many potential pitfalls.

Our naive method is simple — perform factor analysis on the control genes as before, regress the original dataset onto the factors, and use the residuals of the regression as the new dataset. In other words, subtract off (from the entire dataset) the components of variation that are seen to exist in the control genes. This method is "naive" because it implicitly assumes that all of the factors corresponding to the biology of interest are orthogonal to the unwanted factors. For applications such as clustering, in which the goal is to discover the biology of interest from the data, the assumption that the biology of interest is orthogonal to the unwanted variation is at best unverifiable. If the assumption is false, the problem of throwing out the baby with the bathwater returns, and the "adjustment" may very well hurt more than it helps.

Our example is the NCI-60 dataset. The National Cancer Institute maintains 60 cell lines derived from the tissues of 9 different types of human cancers (brain, blood, breast, colon, kidney, lung, ovary, prostate, skin). These cell lines have been well studied, and a great deal of public data are available. More information (including data) can be found at `http://discover.nci.nih.gov/`. We obtained expression data from a study that analyzed the NCI-60 cell lines using using both the Affymetrix HG-U95A and HG-U133A platforms Shankavaram et al. (2007). We wanted to see whether the naive adjustment described above would result in a better clustering of the data into the 9 tissue types. To perform the clustering, we used the R functions `dist` and `hclust` with their default settings ("euclidian" and "complete linkage").

Figure 2.7 provides the clustering results for the HG-U95A data before and after an adjustment using the spike-in controls and $k = 1$. The adjustment helps. Before adjustment, the colorectal cancers were grouped into one clade of size 4 and another clade of size 3. After adjustment, they were all grouped into a single clade. This clade also included one lung cancer, however. Other clusterings improved as well. Before adjustment, 5 of the Leukemias grouped into a single clade, and 1 grouped by itself. After adjustment, all 6 Leukemias grouped into a single clade. Ovarian went from four clades of sizes 3, 2, 1 and 1 to three clades of size 4, 2, and 1. Renal went from 3 clades of size 5, 2 and 1 to two clades of size 7 and 1. Lung went from nine clades of size 1 to 6 clades — one of size 4 and five of size 1. The


Unadjusted

$k = 1$, Affy Spike-in Controls

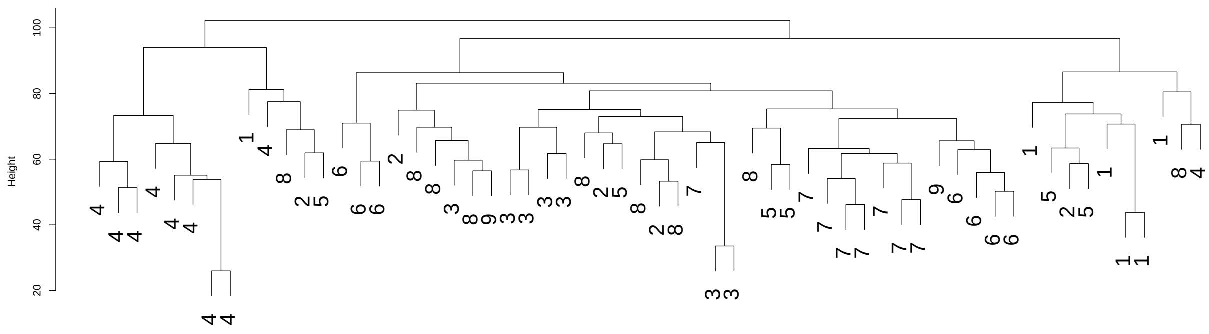
Figure 2.7: Dendrograms of NCI-60 HG-U95A dataset before and after adjustment. The data were preprocessed (BG + QN). The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.

prostate cancers moved closer together, but still did not form their own clade. There was no substantial change in the quality of the clusterings of the other cancers (brain, breast, skin).

Despite the success in Figure 2.7, the naive global adjustment cannot be relied upon in general. Increasing k from 1 to 2 does not further improve the performance (some cancers cluster slightly better while others cluster slightly worse), and increasing k past 2 quickly leads to a *decrease* in the quality of the clustering. See Figures A.38, A.39, and A.40 in the appendix for dendrograms at various values of k. In this particular example, we only knew to set $k = 1$ because we knew the "correct answer." RLE plots provide poor guidance in choosing k, and p-value plots / positive controls are not even applicable since this is not a differential expression study. In a more realistic example, choosing an appropriate k would be very difficult.

Moreover, the method does not work with the HG-U133A data for any value of k. This may be because the unwanted variation is correlated with the biology of interest. We discuss this possibility in more detail in Section 2.3.5. It is also possible that the spike-in controls are simply too noisy, or that the variation characteristic of the spike-in controls is not sufficiently representative of the unwanted variation affecting the majority of the probesets.[13] Indeed, recall from previous examples that spike-in controls generally did not perform as well as housekeeping genes.

Lastly, the method does not work if we use housekeeping genes instead of the spike-in controls. The housekeeping genes are highly correlated with cancer type. (Again, see Section 2.3.5 for a more complete discussion of this topic.) This is presumably due to the fact that the tissue types are sufficiently different that even the expression levels of the housekeeping genes actually do vary from one tissue type to the next, violating of the control gene assumption. See Figure 2.8 for a dendrogram of the HG-U95A data after an adjustment using housekeeping genes.



$k = 1$, Housekeeping Genes

Figure 2.8: Dendrograms of NCI-60 HG-U95A dataset after adjustment using housekeeping genes. The data were preprocessed (BG + QN). The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.

[13] A final logical possibility is that there is simply no unwanted variation to remove. In that case, we would see no improvement after adjustment. However, we do not believe this is the case, as we have evidence there is unwanted variation (although not necessarily batch effects). Indeed, as we discuss in Section 2.3.5, it seems there is unwanted variation that is correlated with biology.

2.3.5 Extended NCI-60 Discussion

This section provides further discussion of the NCI-60 example. The discussion is something of a digression, and may be safely skipped.

In our discussion of the NCI-60 data in Section 2.3.4 we raise the possibility that, in the case of the HG-U133A data, unwanted variation is correlated with biology. As discussed in Section 2.4, the "naive" global adjustment method can fail when the unwanted variation is correlated with biology. Thus, this correlation may explain why the adjustment fails in the case of the HG-U133A data. We also asserted that the (true, wanted) biological variation in the housekeeping genes is highly correlated with biology, and therefore the housekeeping genes cannot be regarded as negative controls in this example. In this section, we present the methods and results by which we arrived at these (tentative) conclusions.

In Section 2.4 we observe that the "naive" global adjustment method will fail if X and $\hat{W}$ are substantially correlated. Recall that this is because the method will only work when $R_{\hat{W}}X = X$, and $R_{\hat{W}}X = X$ only when X and $\hat{W}$ are orthogonal. Now, since $\hat{W}$ is calculated from data, X and $\hat{W}$ will never be perfectly orthogonal in practice. We would therefore like some measure of just how correlated X and $\hat{W}$ actually are. We choose to use canonical correlations as a measure. Since canonical correlations may be unfamiliar to some readers, we present here a brief review, beginning with some motivation. More information can be found in Venables and Ripley (2002).

Consider two full-rank matrices $U_{n\times p}$ and $V_{n\times q}$. Assume that each column of U and each column of V has a mean of 0. Note that U and V have the same number of rows. If we choose one column of U and one column of V it is possible to compute the correlation coefficient between these two columns. We could do this for all pq pairs of columns, and this would, in some sense, tell us about the correlation between U and V. This is called the "cross correlation" of U and V. For our purposes, however, the cross correlation presents some difficulties. Firstly, it consists of pq individual numbers, and it is not clear how to interpret all of these numbers simultaneously to get a clear sense of the extent to which U and V are correlated. Perhaps more importantly, if we reparameterize either U or V, so that its individual columns are changed but its column space is not, the cross correlation of U and V will, in general, change. This makes the cross correlation undesirable as a metric for the "correlation" of X and $\hat{W}$, since the quantity $R_{\hat{W}}X$ is unaffected by reparameterizations of $\hat{W}$; we would like our metric of "correlation" between X and $\hat{W}$ to be likewise unaffected by reparameterizations of $\hat{W}$. Canonical correlations achieve this.

Let u be an element of the column space of U, and let v be an element of the column space of V. Let $r(u, v)$ be the correlation of u and v. We can define the first canonical correlation as

$$r_1 \equiv \max_{u,v} r(u, v).$$

Note that this number is unaffected by reparameterizations of U and V. It also has the advantage of being a single number. We consider it to be a good metric by which to measure the extent to which X and $\hat{W}$ are correlated, and the extent to which $R_{\hat{W}}X \neq X$.

(Note that we could also define the second canonical correlation — and the third, fourth, etc., up to $\min(p, q)$ — but these are not needed for our purposes. For sake of completeness, however, we remark that the second canonical correlation can be defined similarly to how we defined the first canonical correlation, but constraining the maximization over u and v to include only u and v that are orthogonal to u_1 and v_1, where $r_1 = r(u_1, v_1)$.)

It can be shown that the first canonical correlation is equal to the square root of the largest eigenvalue of

$$(V'V)^{-\frac{1}{2}}V'U(U'U)^{-1}U'V(V'V)^{-\frac{1}{2}}. \tag{2.1}$$

Note that in the case that $q = 1$ (i.e. the case that V is a single column),

$$\begin{aligned}
(V'V)^{-\frac{1}{2}}V'U(U'U)^{-1}U'V(V'V)^{-\frac{1}{2}} &= \frac{V'U(U'U)^{-1}U'V}{V'V} && (2.2)\\
&= \frac{V'U(U'U)^{-1}U'U(U'U)^{-1}U'V}{V'V} && (2.3)\\
&= \frac{\left[U(U'U)^{-1}U'V\right]'\left[U(U'U)^{-1}U'V\right]}{V'V} && (2.4)\\
&= \frac{\hat{V}'\hat{V}}{V'V} && (2.5)\\
&= R^2 && (2.6)
\end{aligned}$$

where $\hat{V} \equiv U(U'U)^{-1}U'V$ and R^2 is the coefficient of determination in a regression of V on U. Thus, in the case that $q = 1$, the first canonical correlation between U and V is equal to the familiar quantity $\sqrt{R^2}$ in a regression of V on U. This completes our review of canonical correlations.

We now return to the NCI-60 example. The biology of interest in our example is the cancer tissue type (blood, brain, etc.). Since tissue type is known (despite the fact we treat it as unknown in the example), we can construct X as a 9-column matrix of dummy variables. We can then calculate the first canonical correlation between X and $\hat{W}$.

If we are interested in whether or not the variation characteristic of the control genes is "systematically" correlated with biology, we might want to know whether the observed canonical correlation is "statistically significant." Strictly speaking, this question does not make sense unless the samples were processed at random. Nonetheless, we can generate an interesting "null distribution" by randomly permuting tissue type labels (i.e. randomly permuting the rows of X) and recalculating the canonical correlation each time. We can then calculate the fraction of these re-calculated canonical correlations that are greater than the observed canonical correlation to get an approximate "p-value." If this p-value is small, we might conclude that the unwanted variation is systematically correlated with biology (tissue type). Alternatively, we might conclude that the "control genes" are not control genes after all, and their expression levels are in fact influenced by the tissue type.

We calculated the canonical correlation between X and $\hat{W}$ and an associated p-value for both the HG-U95A and HG-U133A datasets, both with preprocessing and without preprocessing, for both the spike-in controls and the housekeeping genes, with a k of 1 and a k of 5. We present the results in Table 2.5.

	Spike-in Controls				Housekeeping Genes			
	No Preprocessing		BG + QN		No Preprocessing		BG + QN	
	$k = 1$	$k = 5$	$k = 1$	$k = 5$	$k = 1$	$k = 5$	$k = 1$	$k = 5$
HG-U95A	$r_1 = 0.33$ $p \approx 0.61$	$r_1 = 0.55$ $p \approx 0.42$	$r_1 = 0.29$ $p \approx 0.78$	$r_1 = 0.56$ $p \approx 0.38$	$r_1 = 0.37$ $p \approx 0.46$	$\mathbf{r_1 = 0.93}$ $\mathbf{p \approx 0}$	$\mathbf{r_1 = 0.82}$ $\mathbf{p \approx 0}$	$\mathbf{r_1 = 0.92}$ $\mathbf{p \approx 0}$
HG-U133A	$\mathbf{r_1 = 0.55}$ $\mathbf{p \approx 0.015}$	$r_1 = 0.65$ $p \approx 0.06$	$r_1 = 0.44$ $p \approx 0.18$	$r_1 = 0.56$ $p \approx 0.40$	$\mathbf{r_1 = 0.56}$ $\mathbf{p \approx 0.009}$	$\mathbf{r_1 = 0.93}$ $\mathbf{p \approx 0}$	$\mathbf{r_1 = 0.83}$ $\mathbf{p \approx 0}$	$\mathbf{r_1 = 0.93}$ $\mathbf{p \approx 0}$

Table 2.5: First canonical correlation between X and $\hat{W}$ (and an associated p-value) in various cases of the NCI-60 data. Statistically significant entries are shown in bold.

Table 2.5 is a bit complicated, and difficult to interpret. We begin with the HG-U95A dataset; it is a bit simpler. The spike-in controls are not significantly correlated with the biology in any of the cases shown. This is as we might expect. The housekeeping genes, however, are significantly correlated with biology in some cases. This seems to suggest either that unwanted variation is correlated with biology, or that the housekeeping genes are not actually negative controls. Given that the source of the biological signal is quite strong (completely different tissue types) and that the spike-in data suggest that unwanted variation is not correlated with biology, we conclude that the housekeeping genes are not actually negative controls. This mostly explains the HG-U95A results, but one puzzle remains — why are the housekeeping genes not significantly correlated with biology in the case of no preprocessing and $k = 1$? Our interpretation of this result is that without preprocessing, the unwanted variation dominates the biological signal, and thus the first PC is unwanted variation, not biology. This is supported by the fact that the correlation of the first PC of the housekeeping genes (without preprocessing) and the first PC of the spike-in controls (again, without preprocessing) is 0.77. (By contrast, after preprocessing, this correlation is -0.01.)

We now turn to the HG-U133A results. Here we see some confirmation that the housekeeping genes are not negative controls. Indeed, it is interesting to note that the canonical correlations in the last three columns of Table 2.5 are nearly identical between the HG-U95A case and the HG-U133A case. However, the spike-in controls, without preprocessing, are also significantly correlated with biology. This seems to suggest that the unwanted variation in this example is in fact significantly correlated with biology. The puzzle is now why the spike-in controls are *not* significantly correlated with biology after preprocessing. Our interpretation is that the preprocessing — in particular, the quantile normalization — removes much of the component of the unwanted variation that is correlated with biology. As an aside, we note that once again the correlation between the first PC of the housekeeping genes (without preprocessing) and the first PC of the spike-in controls (without preprocessing) is quite strong — 0.88. Thus it seems that the significant correlation between the housekeeping genes and biology in the case of no preprocessing and $k = 1$ is actually driven by unwanted variation, not the biological signal. In other words, it seems that in this particular case, we get the "right answer for the wrong reason."

To summarize, Table 2.5 suggests that housekeeping genes are not negative controls, that the HG-U133A samples were processed in such a manner as to partially confound tissue type with technical artifacts, and that preprocessing at least partially removes these artifacts.

An Unsolved Mystery

We were interested to learn the source of the unwanted variation that is partially confounded with tissue type in the HG-U133A data. The CEL files indicated that all samples were scanned on a single day. We arranged the samples in order of scan time and made box plots of the log perfect-match probe intensities. No temporal patterns were evident. There were no obvious batch effects / clusters. We therefore contacted Gene Logic, where the samples were assayed. Gene Logic was able to provide us with data on the scanner, hybridization station / position, chip lot, and fluidics station used for each of the samples.[14] There were 12 scanners, 8 hybridization stations, 4 fluidics stations, and 2 chip lots employed in the processing of the arrays. Each of the 8 hybridization stations had 4 positions, for a total of 32 station / position combinations. Only 27 of these combinations were actually used. For each of the four main factors (scanner, hybridization station, fluidics station, chip lot), we created a matrix of dummy variables for the factor. In other words, we created a 12-column matrix of dummy variables for the scanners, a 2-column matrix of dummy variables for the chip lots, etc. We also created a 27-column matrix of dummy variables for each of the hybridization station / position combinations. We then took each of these five matrices and calculated the first canonical correlation between it and a matrix of dummy variables representing tissue types. We also computed a p-value for this canonical correlation using a permutation test. Results are presented in Table 2.6.

Chip lot stands out; its correlation with tumor type is highly significant. Forty-six chips came from the first chip lot and 11 came from the second. Of the 11 chips from the second chip lot, 8 were used to assay melanoma samples, 2 were used to assay prostate samples, and one was used to assay an ovarian sample. Conversely, only 2 of the melanoma samples were assayed on chips from the first lot, none of the prostate samples, five of the ovarian samples, and all of the other remaining samples.

Despite the very strong confounding of chip lot and tissue type, it does not seem that this confounding explains the correlation between the first PC of the spike-in controls and

[14] There was missing / mislabeled / mismatched data for two of the samples. All of the CEL files we downloaded from `http://discover.nci.nih.gov/` were timestamped on April 30, 2002. In the data provided by Gene Logic, all samples were reported as having been processed on April 30, 2002, with the exception of two that were reported to have been processed on May 16. These two samples did not stand out in any obvious way. For example, their log perfect-match box plots seemed normal when compared to the rest of the samples. We re-calculated the canonical correlation between the first PC of the spike-in controls and a dummy matrix representing tumor tissue type, this time omitting the two suspicious samples. The results were essentially unchanged. We therefore determined that it was not these two samples that were driving the correlation between tissue type and the first PC of the spike-in controls. In the discussion / analysis that follows, we omit these two samples.

	Scanner	Hyb. Station	Hyb. Stat. / Pos.	Fluidics Stat.	Chip Lot
Correlation w/ Tissue Type	0.69	0.66	0.86	0.48	0.85
p-value	0.5	0.19	0.68	0.52	0

Table 2.6: First canonical correlation between known potential confounders (scanner, hybridization station, hybridization station / position combination, fluidics station, chip lot) and tumor tissue type. P-values are approximated using a permuation test (10,000 iterations).

tissue type. We calculated the canonical correlation between chip lot and the first PC of the spike in controls (un-preprocessed data). It is only 0.03 ($p \approx 0.8$). Chip lot is not a major source of unwanted variation.

Interestingly, it seems that in fact none of these factors are major sources of unwanted variation. In Table 2.7 we present canonical correlations (and p-values) between each of the factors and the first PC of the spike-in controls.

	Scanner	Hyb. Station	Hyb. Stat. / Pos.	Fluidics Stat.	Chip Lot
Correlation w/ first PC	0.47	0.40	0.69	0.33	0.03
p-value	0.35	0.25	0.44	0.11	0.80

Table 2.7: First canonical correlation between known potential confounders (scanner, hybridization station, hybridization station / position combination, fluidics station, chip lot) and the first PC of the spike-in controls (unpreprocessed data). P-values are approximated using a permutation test (10,000 iterations).

We also regressed the first PC of the spike-in controls onto scanner, hybridization station, fluidics station and chip lot simultaneously. The design matrix included a column of 1s for the intercept, 11 columns for the scanners, 7 columns for the hybridization stations, 3 columns for the fluidics stations, and one for chip lot. The value of R^2 is only 0.44, with a permutation test p-value of 0.31. (Equivalently, the canonical correlation between the design matrix and the first PC is 0.66.) Again, the conclusion seems to be that these four factors are not major sources of unwanted variation.

Finally, we computed the residuals of this regression. The canonical correlation between the residuals and tissue type is 0.5, with a p-value of 0.06. This seems to support the assertion that there is some source of unwanted variation partially confounded with tissue type, but the source of this unwanted variation is associated neither with scanner, hybrid station, fluidics station, nor chip lot. Since this unwanted variation is evident in the spike-in controls, it seems it must be technical variation that enters somewhere in the final stages of the assay. What it is, however, remains a mystery.

2.4 Methods

Assume we have m arrays each with n genes (or probes or probesets). Let y_{ij} denote the observed log expression level of the j^{th} gene on the i^{th} array, and let Y denote the $m \times n$ matrix (y_{ij}). We model Y as:

$$Y_{m\times n} = X_{m\times p}\beta_{p\times n} + Z_{m\times q}\gamma_{q\times n} + W_{m\times k}\alpha_{k\times n} + \epsilon_{m\times n} \tag{2.7}$$

Here, X is a matrix whose columns are the factors of interest (e.g. disease state, treatment / control), Z is a matrix whose columns are observed covariates (e.g. batch, ethnicity), and W is a matrix whose columns are unobserved covariates (e.g. sample quality). Note that k is unobserved. The matrices β, γ, and α are all unobserved coefficients that determine the influence of a particular factor on a particular gene. The $Z\gamma$ term is optional; a researcher may not be aware of any observed covariates, or may wish to treat observed covariates as if they were unobserved. To complete our specification of the model, we make the following assumptions:

$$\text{Rank}\left[(X \mid Z \mid W)\right] = p + q + k < m \tag{2.8}$$

$$\mathbb{E}\left[\epsilon \mid X, Z, W\right] = 0 \tag{2.9}$$

$$\text{Var}\left[\epsilon_{ij} \mid X, Z, W\right] = \sigma_j^2 \tag{2.10}$$

$$\epsilon_{ij} \perp\!\!\!\perp \epsilon_{i'j'} \text{ if } (i, j) \neq (i', j') \tag{2.11}$$

Note that this model is essentially identical to the model used in SVA (Leek and Storey, 2007, 2008); where RUV-2 and SVA differ are in the methods used to estimate the unknowns.

In a differential expression study the goal is to estimate β. This is almost a standard linear regression problem. The important difference is that W is unobserved. To address this difficulty, we propose to estimate W from the data using negative control genes.[15] As stated previously, negative control genes are those genes whose expression levels are known *a priori* not to be "truly associated" with X. More formally, the j^{th} gene is a negative control if we can assume that $\beta_{\star j} = 0$, where $\beta_{\star j}$ denotes the j^{th} column of β.

Let Y_c, α_c, β_c, γ_c and ϵ_c be reduced versions of Y, α, β, γ and ϵ that only contain the columns of control genes. Then by (2.7)

$$Y_c = X\beta_c + Z\gamma_c + W\alpha_c + \epsilon_c. \tag{2.12}$$

The "control gene assumption" however is that $\beta_c = 0$, so (2.12) becomes

$$Y_c = Z\gamma_c + W\alpha_c + \epsilon_c. \tag{2.13}$$

For simplicity, first consider the case that there is no $Z\gamma$ term in the model. Then (2.13) becomes

$$Y_c = W\alpha_c + \epsilon_c. \tag{2.14}$$

[15] Recall that we use the term "negative control genes" loosely, and allow it to refer to spike-in controls as well.

This is a typical model in factor analysis, and many methods exist to estimate W (e.g. SVD). Note that, no matter what factor analysis method is used, W can only be estimated if $\operatorname{rank}(\alpha_c) \geq k$. In practice, this means that the control genes must not only be unassociated with the factor of interest, but must indeed be associated with the unwanted factors — the negative controls must exhibit the unwanted variation so that the factor analysis can detect it!

If there is a $Z\gamma$ term in the model, we can multiply both sides of (2.13) by the residual operator

$$R_Z \equiv I - Z\left(Z'Z\right)^{-1} Z'$$

to obtain

$$R_Z Y_c = R_Z W \alpha_c + \epsilon_c. \tag{2.15}$$

We can then use factor analysis to estimate $R_Z W$. Now, in general we cannot assume that $R_Z W = W$, but in practice we can safely treat our estimate of $R_Z W$ as if it were in fact an estimate of W. This is because the standard OLS estimator for β depends only on the column space of $(Z|W)$ — not on Z and W themselves — and the column space of $(Z|W)$ is equal to the column space of $(Z|R_Z W)$. Thus we see that whether or not a $Z\gamma$ term is included in the model, a de-facto estimate $\hat{W}$ of W can be easily produced using factor analysis. We can then calculate $\hat{\beta}$ by OLS, substituting $\hat{W}$ for W. Note that in a trivial sense we include a Z term in all of our analyses, since we include a constant (intercept) term in our models.

We now consider the adaptation of this method introduced in Section 2.3.4. Now the goal is not an estimate $\hat{\beta}$ of β, but rather an adjusted expression matrix $Y^\star$. Additionally, we now regard X as unknown. Regarding X as unknown makes any effort to adapt RUV-2 highly non-trivial. For simplicity, consider the case in which there is no $Z\gamma$ term; extending to the case in which there is a $Z\gamma$ term is trivial. An ideal globally adjusted expression matrix would simply equal the original expression matrix Y minus the unwanted variation $W\alpha$; i.e. $Y^\star$ would equal $X\beta + \epsilon$. In the method of Section 2.3.4, we fall short of this ideal. We define:

$$Y^\star \equiv \left[I - \hat{W}\left(\hat{W}'\hat{W}\right)^{-1}\hat{W}'\right] Y \tag{2.16}$$

where $\hat{W}$ is found by factor analysis as in RUV-2. In words, we project onto the orthogonal complement of the column space of $\hat{W}$. This should effectively remove the unwanted variation. The problem is that it may remove some of the signal as well. Note that

$$Y^\star = R_{\hat{W}}\left(X\beta + W\alpha + \epsilon\right) \tag{2.17}$$

where $R_{\hat{W}}$ is the projector onto the orthogonal complement of the column space of $\hat{W}$. Thus if $\hat{W} \approx W$,

$$Y^\star \approx R_{\hat{W}} X\beta + R_{\hat{W}}\epsilon. \tag{2.18}$$

If $\hat{W}$ is orthogonal to X, then $R_{\hat{W}} X = X$ and $Y^\star$ is nearly our ideal globally adjusted expression matrix. If the unwanted factors are primarily technical factors (not biological), and samples are processed randomly, X and $\hat{W}$ may very well be nearly orthogonal. On the other hand, if $\hat{W}$ and X are not orthogonal, $Y^\star$ may be far from the ideal, and if X is regarded as unknown, it is not even possible to check whether $\hat{W}$ and X are orthogonal — let alone correct for the bias non-orthogonality would introduce. Thus, this naive adaptation of RUV-2 should be used with extreme caution, if at all.

2.5 Discussion

Methodologically, RUV-2 is extremely simple. Its two steps — factor analysis and regression — are well studied and well understood. Despite this simplicity, RUV-2 is highly effective. RUV-2 derives its strength not from any deep new statistical theory but from some powerful biological assumptions. RUV-2 is only as good as these assumptions on which it is based, and it is therefore worthwhile to reiterate these assumptions. The control genes must satisfy two key conditions — they must be 1) uninfluenced by the factor of interest, and they must be 2) influenced by the unwanted factors. Different situations will call for different sets of control genes. The choice of an appropriate set of control genes is central to RUV-2.

We have discussed two possible sets of control genes — housekeeping genes and spike-in controls. In the differential expression examples (gender, Alzheimer's, and TCGA) the housekeeping genes are the better choice, presumably because they are more "representative," and because there are more of them. In the NCI-60 example the spike-in controls are the better choice because the housekeeping genes are not negative controls with respect to tissue type. The NCI-60 example highlights the important fact that housekeeping genes are influenced by biology and cannot be casually assumed to be negative controls in every situation. Housekeeping genes are effective negative controls in the first several examples because they are unaffected by *gender*, not because they are unaffected by biology in general. In short, housekeeping genes are a good place to begin a search for negative controls, but cannot be relied upon in all cases — the factor of interest matters.

We need not restrict our search for control genes to housekeeping genes and spike-in controls. In other studies, still other sets of genes might be the best choice. For example, a researcher might wish to use genes known to be stably expressed within a particular tissue type (Stamova et al., 2009), or under certain experimental conditions; choosing genes specially suited to the study at hand may improve performance. In other situations, a researcher might wish to include additional control genes with the intent of adjusting for specific types of unwanted variation. For example, if a researcher suspects that the cause of death is an important source of unwanted variation, it might be wise to include control genes that could possibly capture this information — e.g. genes associated with cellular stress, apoptosis, etc.

There may be a temptation to "discover" negative control genes. For example, a researcher may wish to find genes whose expression levels are not highly correlated with the factor of interest, label these genes as negative controls, and then adjust via RUV-2. The allure of this approach is clear — finding a set of negative controls would be much easier, and could in fact be automated. However, we feel this approach is misguided. If there are unwanted factors that are correlated with the factor of interest, then the expression levels of the true negative controls should in fact be correlated with the factor of interest. Excluding genes correlated with the factor of interest would bias our estimate of the unwanted factors.

Just as the researcher must exercise judgment when choosing a set of control genes, the researcher must also exercise judgment when choosing k. This can be difficult. We have seen that RLE plots and p-value histograms can be quite helpful. Positive controls, when available,[16] can be even more helpful. Some readers may question why we encourage choosing k based on these quality assessments when more "objective" and automated methods exist. For example, it is possible to choose k via a series of hypothesis tests, in which one keeps increasing k until no more "statistically significant" factors can be found. This is the approach taken in SVA. However, we feel there are problems with this approach. One reason is that including an additional term in a linear regression model may lead to a decrease in bias, but it can also lead to an increase in variance. Thus, it is possible that we might get a better estimate of β by leaving some of the unwanted factors out of the model. Using hypothesis testing to find k does not account for this bias-variance trade-off; instead the goal is simply to include all factors. A second reason is a bit more philosophical. Whenever a researcher calculates a small p-value, three logical possibilities are on the table — the null hypothesis is true and we have observed an unlikely event; the null hypothesis is false; the model is wrong. We know already the model is wrong. We don't know "how wrong," or in precisely which way. Nor do we know exactly how the model misspecification affects the results of any particular hypothesis test. Thus, we feel it is unwise to rely too heavily on a hypothesis test to give us a "good" answer, especially when the choice of k is so important.[17] We feel there is a role here for human judgment, and that quality assessments based on positive controls, p-value histograms, etc., are useful tools in guiding this judgment.

The simplicity of RUV-2 makes it relatively flexible, and an excellent starting point for

[16] Some readers may question the availability of positive controls. Positive controls are genes known to be associated with the factor of interest. The goal of differential expression studies is to find genes associated with the factor of interest. Thus it may seem that if we have positive controls, we already have "the answer." To be sure, positive controls will not always be available, but the situation is not hopeless. In some cases, we might have a set of genes known to be highly enriched with differentially expressed genes, although we don't know exactly which of the genes are the differentially expressed ones. In this case, it might be possible to treat the entire set of genes as if they were all positive controls. For example, the method of ranking genes and counting the number of top-ranked positive controls can still be used. This is exactly the approach taken in our gender examples. Not all X and Y-linked genes are differentially expressed, but many of them are. Alternatively, we may begin with only a handful of known positive controls, when in fact many genes are differentially expressed. An example would have been if we had found any differentially expressed autosomal genes in our gender examples. (In actuality, we did not find any autosomal genes that appeared to be consistently differentially expressed with respect to gender across multiple datasets.)

[17] Our objections here are not "purely philosophical." We experimented with various "objective" methods for determining k, and found that many worked very well on simulated data, but very poorly on real data. Presumably, this was due to model misspecification. Some methods worked well on real data as well — but of course, we could only verify this on the relatively small number of datasets that we had available. Other datasets may suffer other forms of model misspecification, and the methods may no longer work.

new, more advanced methods. In addition, some of the basic ideas of RUV-2 can be useful in exploratory data analysis — the extended NCI-60 discussion in the SI is a good example. In these final paragraphs, we discuss some ways in which RUV-2 might be improved, and suggest possibilities for future development.

One possible direction for improvement is in the regression step. While we considered three different methods of factor analysis (and found that a simple SVD seems to work as well as anything else), with the exception of Limma, we did not consider any elaborations to the regression step of RUV-2. There may be room for improvement. Combat, for example, is essentially an advanced form of regression, and, as we saw in the gender example, it does in fact perform better than standard regression. Incorporating some of the techniques from Combat into RUV-2 may lead to a still better method.

A second avenue for future development concerns combining multiple datasets. It is an open question whether RUV-2 is effective enough to allow a researcher to combine multiple datasets from completely separate studies without introducing excessive unwanted variation. It is also an open question as to whether RUV-2 can be used when combining data from different platforms. We saw in the gender study example that we were able to effectively combine data from the HG-U95A and HG-U95Av2 platforms, but these platforms are relatively similar. It is not clear that such a procedure would also work when combining data from, for example, the HG-U95A and HG-U133A platforms, or when combining Affymetrix and Agilent chips.

Yet another open question is whether RUV-2 can be adapted to entirely different technologies. While microarrays are still used, high throughput sequencing technologies are becoming increasingly popular for use in gene expression studies. Adapting RUV-2 for use with these sequencing technologies could be very helpful. Other technologies, such as qrt-PCR, may benefit from an adaptation of RUV-2 as well.

Finally, we feel it may be possible to apply some of the ideas of RUV-2 to applications other than differential expression. As we have stated, we feel that unwanted variation is most effectively dealt with when it is considered in the context of the goal of the analysis at hand. RUV-2 deals effectively with unwanted variation in the context of differential expression studies. However, microarrays are also commonly used for classification and for clustering. We do not yet know how best to handle unwanted variation in these types of studies, but we believe control genes will play an important role.

Chapter 3

RUV-4

3.1 Introduction

In this chapter we build upon the work of the previous chapter. The contributions of this chapter are several. To begin, we present a new method, RUV-4. RUV-4 superficially resembles RUV-2. RUV-4 is intended for use in differential expression studies, uses control genes to identify unwanted factors, and includes the estimated unwanted factors as covariates in a regression model. However, the exact method by which the unwanted factors are estimated is different, and this difference has important statistical implications. Compared to RUV-2, the performance of RUV-4 is much less sensitive to the number K of unwanted factors that we include in the regression model. Compared to RUV-2, RUV-4 is also much less sensitive to violations of the control genes assumption, i.e. situations in which the designated "negative controls" are in fact truly associated with the factor of interest.

RUV-4 is also of theoretical interest. It is possible to view RUV-4 as a method in which unwanted factors are inferred from the data and then included in the design matrix of a regression model. In this way, RUV-4 is similar to RUV-2, SVA (Leek and Storey, 2007, 2008), LEAPP (Sun et al., 2011), and other related methods. However, we show that it is also possible to view RUV-4 as form of generalized least squares (GLS). In this way, RUV-4 is similar to ICE (Kang et al., 2008a), LMM-EH (Listgarten et al., 2010), and other related methods. RUV-4 provides an interesting theoretical link between these two classes of methods. Perhaps more importantly, however, RUV-4 may also be viewed as an exercise in prediction, or function estimation. This view of RUV-4 is important for two reasons. Firstly, it allows for a deeper understanding of the assumptions of RUV-4, giving researchers more insight into when RUV-4 is likely to succeed, when it may fail, and why. Secondly, viewing RUV-4 as a prediction problem leads quickly and naturally to ideas for more advanced methods.

Another important contribution of this chapter is a novel method for estimating variances, which we name the "inverse method." This method, which uses random "factors of interest," allows us to estimate gene-wise variances even when all available degrees of freedom have been used up adjusting for unwanted variation, i.e. when K is so large that the design matrix is full rank. By using the inverse method, we may simply set K to be very large by default, and in most cases suffer no performance penalty. Thus, the inverse method *eliminates any need to estimate the number of unwanted factors*. Nonetheless, methods somewhat related to the inverse method may also be used to estimate the number of unwanted factors when such an estimate is needed. We present one such method that we have found to perform reasonably well in practice.

Additional contributions of this chapter include 1) a "ridged" variant of RUV-4, useful in situations where only a small number of negative controls are available; 2) methods to empirically adjust estimates of variances, in order to achieve better control of the type 1 error rate; 3) a discussion on how negative controls might be discovered "empirically;" and 4) "projection plots," a novel diagnostic plot that allows a researcher to visualize the adjustment being made by RUV-2 or RUV-4, and assess whether this adjustment seems appropriate.

The structure of this chapter is as follows. In Section 3.2 we present the datasets we will use to evaluate the performance of our methods. In Section 3.3 we present our methods. In Section 3.4 we evaluate the performance of our methods using simulated data, and in Section 3.5 we evaluate the performance of our methods on the real datasets of Section 3.2.

3.2 Data

In order to compare the relative performance of the methods discussed in this chapter, we will want to apply the methods to a few real datasets. As in Chapter 2, we will investigate differential expression with respect to gender in the brain. The primary reason we investigate differential expression with respect to gender is because the answer is in some sense "known." It is sensible to assume *a priori* that most, if not all, of the genes differentially expressed with respect to gender in the brain will come from either the X or Y chromosomes. This observation provides us with some very straight-forward metrics with which we can compare the performance of various methods intended to find differentially expressed genes. We can, for example, simply take the 100 top-ranked genes (in terms of p-values) and count the number of these top-ranked genes that come from the X or Y chromosomes. With only a few minor caveats, we may regard the better method to be the one that finds more X/Y genes. See Chapter 2 for further discussion on the use of gene rankings as a quality metric, and discussion on quality metrics more generally.

We focus on the brain partly because the relatively complex biology of the brain makes finding differentially expressed genes more challenging, and partly because of the availability of several suitable datasets. As in Chapter 2, we will use data from three different studies: a "gender study" in which the original scientific goal was to find genes in the brain differentially expressed with respect to gender; an "Alzheimer's study" in which the original scientific goal was to find genes differentially expressed with respect to the severity of Alzheimer's disease, and the The Cancer Genome Atlas's (TCGA) glioblastoma multiforme study. To be clear, despite the fact that these studies were originally conducted with different scientific goals in mind, we will use each of these datasets in exactly the same way — to find genes differentially expressed with respect to gender. Further information on the gender and Alzheimer's studies can be found in Vawter et al. (2004) and Blalock et al. (2004), respectively.

In total we will examine 11 distinct datasets. We will examine two variants of the gender dataset. One has been fully preprocessed and the other has not. These two datasets are exactly those described in Chapter 2. The point of using data that has not been preprocessed is that it is much "noiser" than preprocessed data, and therefore more challenging. It is of interest to see how the various methods discussed in this chapter handle this added challenge. Likewise, we examine both preprocessed and non-preprocessed versions of the Alzheimer's data. Again, these two datasets are exactly those described in Chapter 2.

The remaining seven datasets come from TCGA data. TCGA glioblastoma multiforme expression data is available from three different microarray platforms: the Affymetrix GeneChip Human Exon 1.0 ST array, the Affymetrix HT HG-U133A array, and the Agilent custom 244K array. In Chapter 2 we examined data from the two Affymetrix arrays. Here we examine datasets from all three arrays. Note however that the Affymetrix datasets that we examine in this chapter are not identical to those in Chapter 2. TCGA has continued to assay additional samples, and the datasets we examine here have been "updated" to include many of the newer samples. Moreover, note that TCGA provides both raw ("Level 1") and preprocessed ("Level 3") data. In Chapter 2 we began with the raw data and performed the preprocessing themselves. Here however we simply downloaded TCGA's preprocessed datasets.

Three of the TCGA datasets that we examine are simply the three "full" datasets, i.e. one "full" dataset for each of the three platforms (Exon, U133A, and Agilent). We created a fourth TCGA dataset by combining data from all three platforms. Because the full datasets are all individually quite large, we included only a subset of the data from each of the three platforms. We included the first 100 arrays from the Exon dataset, the second 100 arrays from the U133A dataset, and the third 100 arrays from the Agilent dataset. The "combined" dataset therefore includes 300 samples, none of which are technical replicates. Only genes that are common to all three platforms are included in the "combined" dataset. The final three TCGA datasets are simply the three subsets of the "combined" dataset corresponding to each of the three platforms, i.e. the 100 Exon samples, the 100 U133A samples, and the 100 Agilent samples. Note that these "subset" datasets are subsets of the "full" datasets not only in the sense that they include only a subset of samples, but also in the sense that they include only a subset of the genes (specifically, those common to all the platforms). The point of examining these three "subset" datasets individually is so that we have a valid basis for comparison when we discuss the advantages and disadvantages of combining data from different platforms.

In Table 3.1 we report the number of samples m, the number of genes n, and the number of available control genes n_c in each of the 11 datasets. The control genes we use are the housekeeping genes discovered by Eisenberg and Levanon (2003). This is the same set of housekeeping genes used in Chapter 2. Unlike in Chapter 2 however, we do not discuss the use of spike-in controls in this chapter.

	m (# of arrays)	n (# of genes)	n_c (# of control genes)
Gender (preprocessed)	84	12600	799
Gender (non-preprocessed)	84	12600	799
Alzheimer's (preprocessed)	31	22283	1112
Alzheimer's (non-preprocessed)	31	22283	1112
TCGA – Exon (Full)	420	18632	518
TCGA – U133A (Full)	490	12042	520
TCGA – Agilent (Full)	466	17472	521
TCGA – Combined	300	11750	509
TCGA – Exon (Subset)	100	11750	509
TCGA – U133A (Subset)	100	11750	509
TCGA – Agilent (Subset)	100	11750	509

Table 3.1: The number of arrays, genes, and control genes in each dataset.

3.3 Methods

In this section we formally present the methods of this chapter. In Section 3.3.1 we present a model and some notation. In Section 3.3.2 we review RUV-2 and discuss its relationship to IVLS. In Section 3.3.3 we present RUV-4, one of the main contributions of this chapter. We then compare RUV-4 to RUV-2 in Section 3.3.4. In Section 3.3.5 we investigate the basic statistical properties of RUV-4, and in Section 3.3.6 we consider additional statistical properties of RUV-4 that are particularly relevant to the use of RUV-4 with real data. In Section 3.3.7 we present the inverse method for estimating variances, a second important contribution of this chapter. In Section 3.3.8 we introduce the functional approach. This section provides an alternative framework by which to understand the methods of this chapter, and, just as importantly, suggests directions for future research. In Section 3.3.9 we present a few variations and extensions of the main methods of this chapter.

3.3.1 Background and Model

First we present the model. Then we define some additional notation. Finally, we provide a brief general discussion of the statistical challenges we face when fitting the model.

3.3.1.1 The Model

Assume we have m arrays each with n genes (or probes or probesets). Let y_{ij} denote the observed log expression level of the j^{th} gene on the i^{th} array, and let Y denote the $m \times n$

matrix (y_{ij}). We model Y as:

$$Y_{m\times n} = X_{m\times p}\beta_{p\times n} + Z_{m\times q}\gamma_{q\times n} + W_{m\times k}\alpha_{k\times n} + \epsilon_{m\times n} \tag{3.1}$$

where

$$\begin{aligned}
\text{Rank}\left[(X \mid Z \mid W)\right] &= p + q + k < m && (3.2)\\
\epsilon_{ij} &\sim N(0, \sigma_j^2) && (3.3)\\
\epsilon_{ij} &\perp\!\!\!\perp \epsilon_{i'j'} \text{ if } (i,j) \neq (i',j') && (3.4)
\end{aligned}$$

Here, X is an observed matrix whose columns are the factors of interest (e.g. disease state, treatment / control), Z is a matrix whose columns are observed covariates (e.g. batch, ethnicity), and W is a matrix whose columns are unobserved covariates (e.g. sample quality). Note that k is unobserved. The matrices β, γ, and α are all unobserved coefficients that determine the influence of a particular factor on a particular gene. We regard X, Z, W, β, γ, and σ_j to be fixed. As for α, in some sections we will regard it as fixed; in other sections we will regard it to be random. We begin by assuming it is fixed. When we do regard α as random we assume:

$$\begin{aligned}
\alpha &\perp\!\!\!\perp \epsilon && (3.5)\\
\alpha_{ij} &\perp\!\!\!\perp \alpha_{i'j'} \text{ if } (i,j) \neq (i',j') && (3.6)
\end{aligned}$$

The $Z\gamma$ term is optional; a researcher may not be aware of any observed covariates, or may wish to treat observed covariates as if they were unobserved. Unless we specifically state otherwise, we will assume for simplicity that there are no observed covariates, and only one factor of interest. The model simplifies to

$$Y_{m\times n} = X_{m\times 1}\beta_{1\times n} + W_{m\times k}\alpha_{k\times n} + \epsilon_{m\times n} \tag{3.7}$$

3.3.1.2 Additional Notation

- Let σ^2 denote the n-dimensional vector of gene variances (σ_j^2).
- Let n_c denote the number of control genes.
- Let Y_c, α_c, β_c, and ϵ_c be reduced versions of Y, α, β, and ϵ that only contain the columns of the negative control genes. Thus Y_c is an $m \times n_c$ matrix, α_c is $k \times n_c$, etc. Recall that negative control genes are genes that we assume are uninfluenced by the factor of interest. Thus we assume $\beta_c = 0$. We refer to this as the "control gene assumption."
- Let j_c index control genes and $j_{\bar{c}}$ index non-control genes.
- If A is a matrix, denote the i^{th} row of of A by $A_{i\star}$ and the j^{th} column of A by $A_{\star j}$.

- If A is a matrix with a single column, let $A_i \equiv A_{i1}$. If A is a matrix with a single row, let $A_j \equiv A_{1j}$.
- Denote the range (column space) of a matrix A by $\mathfrak{R}(A)$.
- Denote the projection operator of a matrix A by P_A; i.e. $P_A \equiv A(A'A)^{-1}A'$ projects onto $\mathfrak{R}(A)$.
- Denote the residual operator of a matrix A by R_A; i.e. $R_A \equiv I - A(A'A)^{-1}A'$ projects onto the orthogonal complement of $\mathfrak{R}(A)$.
- Denote the particularly important quantity $R_X W$ by W_0.
- If A is some matrix with N rows and rank $M < N$, let $A_\perp$ denote some (possibly arbitrary) rank $N - M$ matrix whose columns are unit length, mutually orthogonal, and such that $A'A_\perp = 0$.
- Denote the partial regression coefficient of A on B as b_{AB} and the partial regression coefficient of A on B adjusted for C as $b_{AB.C}$. Let β_{AB} and $\beta_{AB.C}$ denote the associated parameters. Alternatively, readers may simply choose to regard the following as definitions:

$$
\begin{aligned}
b_{YX} &\equiv (X'X)^{-1}X'Y && (3.8)\\
b_{Y_cX} &\equiv (X'X)^{-1}X'Y_c && (3.9)\\
b_{WX} &\equiv (X'X)^{-1}X'W && (3.10)\\
b_{Y_cX.W} &\equiv (X'R_WX)^{-1}X'R_WY_c && (3.11)\\
b_{Y_cW.X} &\equiv (W'R_XW)^{-1}W'R_XY_c && (3.12)\\
\beta_{Y_cX.W} &\equiv \beta_c && (3.13)\\
\beta_{Y_cW.X} &\equiv \alpha_c && (3.14)
\end{aligned}
$$

Note: We will assume throughout this thesis without loss of generality that the columns of W_0 are mutually orthogonal and have unit length, and that $||X|| = 1$ as well.

3.3.1.3 Statistical Challenges

Our model (3.7) closely resembles a standard linear regression model. The difference is that in our model W is unobserved. A natural strategy to fit our model would therefore be to find some way to estimate W and then proceed with standard regression. The difficulty with this approach is that W is not identifiable without additional assumptions.

Let A be any invertible $k \times k$ matrix. Then

$$(WA)(A^{-1}\alpha) = W\alpha, \tag{3.15}$$

so neither W nor α are identifiable. However, this particular form of unidentifiability is not a problem. Our ultimate goal is to estimate β. For this, knowledge of the column-space of W will suffice. To see this, note that knowledge of $\mathfrak{R}(W)$ would allow us to compute R_W, the residual operator of W. We could therefore calculate $(X'R_WX)^{-1}X'R_W$, the OLS estimate of β.

The real problem is that even $\mathfrak{R}(W)$ is unidentifiable. To illustrate the unidentifiability, let a be an arbitrary $1 \times k$ row-matrix. Then

$$X(\beta + a\alpha) + (W - Xa)\alpha = X\beta + W\alpha. \tag{3.16}$$

In words, since we don't know the correlation of X and W, we are unable to separate β from α. This is the fundamental problem that needs solving. Our solution is to use control genes. As we will see, the assumption that $\beta_c = 0$, along with a few other technical assumptions, is enough to make $\mathfrak{R}(W)$, and thus β, identifiable.

3.3.2 The Two-Step Method (RUV-2)

First we present the method. Then we provide a brief discussion.

3.3.2.1 The Method

Consider only the expression values for the negative control genes. By (3.7),

$$Y_c = X\beta_c + W\alpha_c + \epsilon_c. \tag{3.17}$$

The "control gene assumption" however is that $\beta_c = 0$, so (3.17) becomes

$$Y_c = W\alpha_c + \epsilon_c. \tag{3.18}$$

This is a typical model in factor analysis, and many methods exist to estimate W (e.g. SVD). The two-step method therefore is to

- **Step 1:** Estimate W by factor analysis on Y_c.
- **Step 2:** Estimate β by regressing Y on X and the estimate of W.

More explicitly, we may denote the estimate of W as $\hat{W}^{(\mathrm{RUV-2})}$ and define

$$\hat{\beta}^{(\mathrm{RUV-2})} \equiv \left(X'R_{\hat{W}^{(\mathrm{RUV-2})}}X\right)^{-1} X'R_{\hat{W}^{(\mathrm{RUV-2})}}Y. \tag{3.19}$$

In the future, we will drop the RUV-2 superscript when it is clear from context.

3.3.2.2 Relationship to IVLS and Further Discussion

RUV-2 is discussed in depth in Chapter 2. In this paragraph we merely summarize some important points of Chapter 2. The first is that no matter what factor analysis method is used, W can only be estimated if $\text{rank}(\alpha_c) = k$. This means that in addition to requiring that the control genes be unassociated with the factor of interest, we must also require that they be associated with the unwanted factors. Choosing an appropriate set of control genes is essential to the success of RUV-2. A second point is that we must estimate k. This can be very difficult. Again, see Chapter 2 for more on this matter.

One point not discussed in Chapter 2 is the relationship of RUV-2 to instrumental variables least squares (IVLS). Some readers may find RUV-2 reminiscent of IVLS. Indeed, there are some similarities. Let V be a full rank $m \times r$ matrix of instruments such that $m > r \geq p$, such that $V'W \approx 0$, and such that $V'X$ is full rank. An IVLS estimator of β would be $[X'V(V'V)^{-1}V'X]^{-1}X'V(V'V)^{-1}V'Y$. Alternatively, we may write this IVLS estimator as $(X'P_VX)^{-1}X'P_VY$. Compare the IVLS estimator to the RUV-2 estimator $(X'R_{\hat{W}}X)^{-1}\, X'R_{\hat{W}}Y$. With both the IVLS estimator and the RUV-2 estimator, we "avoid" the unwanted variation by projecting the data into a "safe" subspace that is (approximately) orthogonal to $\mathfrak{R}(W)$. In the case of IVLS the "safe" subspace is $\mathfrak{R}(V)$. This subspace is orthogonal to $\mathfrak{R}(W)$ by assumption. In practice, the assumption that $\mathfrak{R}(V)$ is orthogonal to $\mathfrak{R}(W)$ usually derives from the assumption that W and V are stochastically independent. In the case of RUV-2 the "safe" subspace is $\mathfrak{R}(\hat{W}_\perp)$. This subspace is orthogonal to $\mathfrak{R}(W)$ if $\mathfrak{R}(W) \subseteq \mathfrak{R}(\hat{W})$. In practice, the assumption that $\mathfrak{R}(W) \subseteq \mathfrak{R}(\hat{W})$ derives from the assumptions that $\text{rank}(\alpha_c) = k$, that $\hat{k} \geq k$, and that the factor analysis "works."

We might choose to view IVLS and RUV-2 as complementary. With IVLS we identify a "safe" subspace using instruments. Instruments are variables that we assume lie within the "safe" subspace. With RUV-2 we identify a "safe" subspace using negative controls. Negative controls are variables that we assume lie within the "dangerous" subspace that is the orthogonal complement of the "safe" subspace. With both IVLS and RUV-2 there is a caveat. The caveat is that X must not be orthogonal to the "safe" subspace. In the case of IVLS, this means that V must be reasonably correlated with X; we want to avoid weak instruments. In the case of RUV-2, this means that X must lie outside $\mathfrak{R}(\hat{W})$; the control genes must not be influenced by X.

3.3.3 The Four-Step Method (RUV-4)

We now present a new method to estimate β. As we did with RUV-2, we first present the method and then provide a brief discussion.

3.3.3.1 The Method

We estimate β in four steps.

- **Step 1: Fit and Remove** X

Multiply both sides of (3.7) by R_X to obtain

$$\begin{aligned} R_X Y &= R_X X\beta + R_X W\alpha + R_X\epsilon && (3.20)\\ &= W_0\alpha + R_X\epsilon && (3.21)\end{aligned}$$

- **Step 2: Factor Analysis**

  Use some variant of factor analysis to produce an estimate $\widehat{W_0\alpha}$ of $W_0\alpha$. In addition, define individual estimates $\hat{W}_0$ and $\hat{\alpha}$ such that $\hat{W}_0\hat{\alpha} = \widehat{W_0\alpha}$. Although we need not commit to a specific method of factor analysis, we impose two requirements:

  $$P_X\hat{W}_0 = 0 \tag{3.22}$$

  and

  $$\hat{\alpha} = \left(\hat{W}_0'\hat{W}_0\right)^{-1}\hat{W}_0'Y. \tag{3.23}$$

  Note that W_0 and α are not identifiable, so the factorization of $\widehat{W_0\alpha}$ into $\hat{W}_0$ and $\hat{\alpha}$ will be somewhat arbitrary; this turns out not to matter.

- **Step 3: Estimate** W

  We begin with the observation that

  $$\begin{aligned} W &= W - X(X'X)^{-1}X'W + X(X'X)^{-1}X'W \\ &= (I - X(X'X)^{-1}X')W + X\left[(X'X)^{-1}X'W\right] \\ &= W_0 + Xb_{WX} && (3.24)\end{aligned}$$

  We know X and we have an estimate of W_0. We therefore would like to estimate b_{WX}. To do so we make use of the identity

  $$b_{Y_cX} = b_{Y_cX.W} + b_{WX}b_{Y_cW.X}. \tag{3.25}$$

  Assuming $\left(b_{Y_cW.X}b'_{Y_cW.X}\right)^{-1}$ exists, solving for b_{WX} yields

  $$b_{WX} = \left(b_{Y_cX} - b_{Y_cX.W}\right)b'_{Y_cW.X}\left(b_{Y_cW.X}b'_{Y_cW.X}\right)^{-1}. \tag{3.26}$$

  Note that

  $$b_{Y_cX.W} \approx \beta_{Y_cX.W} = \beta_c = 0 \tag{3.27}$$

  and

  $$b_{Y_cW.X} \approx \beta_{Y_cW.X} = \alpha_c \approx \hat{\alpha}_c \tag{3.28}$$

  so

  $$b_{WX} \approx b_{Y_cX}\hat{\alpha}_c'\left(\hat{\alpha}_c\hat{\alpha}_c'\right)^{-1}. \tag{3.29}$$

  We therefore define our estimate of W as

  $$\hat{W} \equiv \hat{W}_0 + Xb_{Y_cX}\hat{\alpha}_c'(\hat{\alpha}_c\hat{\alpha}_c')^{-1}. \tag{3.30}$$

- **Step 4: Regress Y onto X and $\hat{W}$ to estimate β**

  Just as in the two-step method, we plug in our estimate of W as a covariate in a regression model. More explicitly, we may denote $\hat{W}$ as $\hat{W}^{(\mathrm{RUV-4})}$ and define

  $$\hat{\beta}^{(\mathrm{RUV-4})} \equiv \left(X' R_{\hat{W}^{(\mathrm{RUV-4})}} X\right)^{-1} X' R_{\hat{W}^{(\mathrm{RUV-4})}} Y. \tag{3.31}$$

  Again, we will drop the superscript when it is clear from context.

3.3.3.2 Discussion

RUV-4 may be roughly regarded as a hybrid between RUV-2 and SVA (Leek and Storey, 2007, 2008). Recall that a central problem when using factor analysis to discover unwanted factors is that the factor analysis might also pick up signal from the biological factor of interest. RUV-2 addressed this problem by limiting the factor analysis to negative control genes. Leek and Storey proposed a different solution: first remove the signal of interest by projecting the data onto the orthogonal complement of $\mathfrak{R}(X)$, and only then do the factor analysis. This effectively solves the problem of picking up the factor of interest in the factor analysis, and we borrow the technique in steps 1 and 2 of RUV-4.

However, this technique introduces a problem of its own. A factor analysis on $R_X Y$ will not accurately estimate the unwanted factors W. Instead, it will estimate W_0. Therefore, we need to recover the bit of W that was projected away in step 1. This is complicated by the unidentifiability of $\mathfrak{R}(W)$ highlighted in (3.16). We address this problem by borrowing the main idea of RUV-2 — that negative control genes can be used to make $\mathfrak{R}(W)$ identifiable.

Step 3 of RUV-4 is where we attempt to recover the component of W lost in step 1. Step 3 is the most important and complicated step, so we clarify a few points. The control gene assumption enters in (3.27), when we assume $b_{Y_c X.W} \approx 0$. The exact interpretation of this assumption is a bit subtle. Recall that $b_{Y_c X.W}$ is the partial regression coefficient of Y_c on X adjusted for W. In other words, it is the OLS estimate of β_c that we would get in a regression of Y_c on X and W, if in fact we had the true W available to us. Since we assume $\beta_c = 0$, and since $b_{Y_c X.W}$ is the hypothetical OLS estimate of β_c, we conclude $b_{Y_c X.W} \approx 0$. To be even more precise, note that

$$\begin{aligned} b_{Y_c X.W} &= (X' R_W X)^{-1} X' R_W Y_c && (3.32)\\ &= (X' R_W X)^{-1} X' R_W \left(X\beta_c + W\alpha_c + \epsilon_c\right) && (3.33)\\ &= \beta_c + (X' R_W X)^{-1} X' R_W \epsilon_c && (3.34)\\ &= (X' R_W X)^{-1} X' R_W \epsilon_c && (3.35) \end{aligned}$$

and that $\mathbb{E}\left[(X' R_W X)^{-1} X' R_W \epsilon_c\right] = 0$.

The exact interpretation of (3.28) is subtle as well. We assume that $b_{Y_c W.X} \approx \alpha_c$, that $\hat{\alpha}_c \approx \alpha_c$, and thus that $b_{Y_c W.X} \approx \hat{\alpha}_c$. The first of these assumptions is analogous to our assumption that $b_{Y_c X.W} \approx \beta_c$. The second assumption is simply that our estimate of α_c from step 2 (factor analysis) is in fact a good estimate of α_c. The composite assumption that

$b_{Y_cW.X} \approx \hat{\alpha}_c$ is therefore an assumption that two different estimates of α_c — one hypothetical, and one obtainable — are approximately equal to one another. A different way to view this assumption is as follows: $b_{Y_cX.W} = (W_0'W_0')^{-1}W_0'Y_c$ and $\hat{\alpha}_c = (\hat{W}_0'\hat{W}_0)^{-1}\hat{W}_0'Y_c$. Therefore, if $\hat{W}_0 \approx W_0$, $b_{Y_cX.W} \approx \hat{\alpha}_c$. Indeed, if $\hat{W}_0$ is a particularly good estimate of W_0, it may turn out that $b_{Y_cX.W}$ and $\hat{\alpha}_c$ are closer to one another than either is to α_c. In fact, since $\hat{W}_0$ is in general a better estimate of W_0 than $\hat{\alpha}$ is of α (a consequence of the fact that $n \gg m$), it may often be the case in practice that $b_{Y_cX.W}$ and $\hat{\alpha}_c$ are closer to one another than either is to α_c. Finally, a minor technical point: A careful reader might object that the unidentifiability of W and α expressed in (3.15) implies that $\hat{W}_0$ and W_0 are not necessarily approximately equal to one another, and likewise for $\hat{\alpha}$ and α. This is true — recall that the factorization of $\widehat{W_0\alpha}$ into $\hat{W}_0$ and $\hat{\alpha}$ is arbitrary. However, it can be shown that $\hat{\beta}^{(\mathrm{RUV-4})}$ is independent of the choice of factorization. See Section (B.1.1) in the appendix.

A reformulation of $\hat{\beta}^{(\mathrm{RUV-4})}$ that will prove useful later is:

$$\hat{\beta}^{(\mathrm{RUV-4})} = (X'X)^{-1} X'(Y - \hat{W}\hat{\alpha}). \tag{3.36}$$

To see that this is true, note that we have defined $\hat{\beta}^{(\mathrm{RUV-4})}$ in (3.31) as the OLS coefficient of X in a regression of Y on X and $\hat{W}$. Note further that as a consequence of (3.22) and (3.23), $\hat{\alpha}$ is the OLS coefficient of $\hat{W}$ in a regression of Y on X and $\hat{W}$, and that therefore $(X'X)^{-1} X'(Y - \hat{W}\hat{\alpha})$ is also the OLS coefficient of X in a regression of Y on X and $\hat{W}$. Finally, note that (3.36) can itself be reformulated as

$$\hat{\beta}^{(\mathrm{RUV-4})} = b_{YX} - \hat{b}_{WX}\hat{\alpha} \tag{3.37}$$

where $\hat{b}_{WX} \equiv b_{Y_cX}\hat{\alpha}_c' \left(\hat{\alpha}_c\hat{\alpha}_c'\right)^{-1}$.

Finally, some additional notation. Let $\hat{k}$ denote an estimator of k. In Section 3.3.6.6 we will explicitly define $\hat{k}$; until then we may think of $\hat{k}$ as representing some arbitrary estimator. Let $\hat{W}_0^{(K)}$ denote the estimate of W_0 that we would get in Step 2 if we instructed our factor analysis routine to produce an estimate of W that was of dimension $m \times K$. Similarly denote $\hat{\alpha}^{(K)}$, $\hat{b}_{WX}^{(K)}$, $\hat{W}^{(K)}$, etc. as the estimates of α, b_{WX}, W, etc. that we would subsequently produce if we were to use $\hat{W}_0^{(K)}$ as our estimate of W_0. Note that there are three "k"s: k is the true parameter in the model; $\hat{k}$ is an estimate of k; and K may be viewed either as a parameter in an algorithm, or simply as an index variable.

3.3.4 Comparison of RUV-2 and RUV-4

In both RUV-2 and RUV-4, we estimate $\hat{\beta}$ by first estimating W, and then regressing Y on X and $\hat{W}$. The difference between the methods is only in how we estimate W. In both methods, however, we use factor analysis to estimate W, making special use of control genes to make $\mathfrak{R}(W)$ identifiable. It is therefore natural to ask whether there are any substantive differences between the methods. Indeed there are, and in this section we attempt to develop some intuition for these differences using a few simple examples and illustrations.

Consider a very simple example in which $m = 2$ and $k = 1$. Note that in this specific example, $W_0 = X_\perp$. Note in particular that we do not need any factor analysis to "discover" W_0. Step 2 of RUV-4 is therefore irrelevant in this example, but this does not seriously detract from the intuition we develop in the discussion that follows.

Now, note that because $m = 2$, for any gene j the vector of expression levels across samples (i.e. $Y_{\star j}$) is a two-dimensional vector and can be plotted as a point on a standard coordinate plane. X and W_0 form a natural set of basis vectors against which we can plot $Y_{\star j}$. Such plots of Y against X and W_0 are very helpful for developing an intuitive understanding of RUV-4; as we will see later, they are also a very useful diagnostic tool.

Figure 3.1 decomposes such a plot for a single $Y_{\star j}$. A few simplifications are made for visual clarity. We suppress the subscripts on $Y_{\star j}$, β_j, etc. Far more importantly, we do not distinguish between W and $\hat{W}$ (or between b_{WX} and $\hat{b}_{WX}$). Although the distinction between W and $\hat{W}$ is an important one, including both W and $\hat{W}$ (and b_{WX} and $\hat{b}_{WX}$) in the figure introduces too much clutter. We leave the distinction to the reader's imagination.

In Figure 3.1 we see that $Y_{\star j}$ is the vector sum of $X\beta_j$, $W\alpha_j$, and $\epsilon_{\star j}$. We decompose $Y_{\star j}$ into a horizontal component and a vertical component. The magnitude of the horizontal (W_0) component gives us an estimate of how much of the unwanted factor is present; i.e. it is our $\hat{\alpha}_j$. This estimate is unbiased, and accurate up to the error introduced by the horizontal component of $\epsilon_{\star j}$. The vertical component of $Y_{\star j}$ (i.e. $b_{Y_{\star j}X}$) may be regarded as the "observed X-signal." The magnitude of the "observed X-signal" is the sum of three terms — the "true X-signal" β_j, the magnitude of the vertical component of $W\alpha_j$, and the magnitude of the vertical component of $\epsilon_{\star j}$. There is not much we can do about the vertical component of $\epsilon_{\star j}$, but we can try to adjust for the vertical component of $W\alpha_j$. We can estimate the magnitude of the vertical component of $W\alpha_j$ by $\hat{b}_{WX}\hat{\alpha}_j$. We can thus subtract $\hat{b}_{WX}\hat{\alpha}_j$ from the "observed X-signal" $b_{Y_{\star j}X}$ to produce our estimate $\hat{\beta}_j$ of β_j. Note that in this way, we have just graphically re-derived formula (3.37) — however, we have left out the important detail of where $\hat{b}_{WX}$ comes from!

Figure 3.2 shows where $\hat{b}_{WX}$ comes from. Instead of plotting $Y_{\star j}$ for a single gene j, we plot all $n = 1000$ genes. Control genes are plotted as green. Genes for which $\beta_j \neq 0$ are plotted as purple. All other genes (i.e. genes for which $\beta_j = 0$ but which are not specifically designated as control genes) are plotted as gray. W is plotted as black. $\hat{W}$ is plotted as orange. For one arbitrary example gene (plotted as solid purple) we show that $\hat{\beta}_j$ is the vertical distance from $Y_{\star j}$ to the dotted orange line spanned by $\hat{W}$.

Now, $\hat{b}_{WX}$ is simply the slope of the dotted orange line. Estimating b_{WX} (and thus W) therefore amounts to choosing the slope of this "baseline" from which we measure $\hat{\beta}_j$. The RUV-4 strategy for choosing this slope is to draw the regression line (without an intercept) through the control genes (green). That is why $\hat{b}_{WX} = b_{Y_c X}\hat{\alpha}'_c\left(\hat{\alpha}_c\hat{\alpha}'_c\right)^{-1}$.

We are now in a position to compare RUV-4 with RUV-2. Figure 3.3A compares the RUV-4 and RUV-2 estimators using the same example data as in Figure 3.2. To reduce clutter, we have removed the vector representation of $\hat{W}^{(\mathrm{RUV-4})}$, leaving only the orange dotted line to show its span. The brown dotted line is the span of $\hat{W}^{(\mathrm{RUV-2})}$. Just as

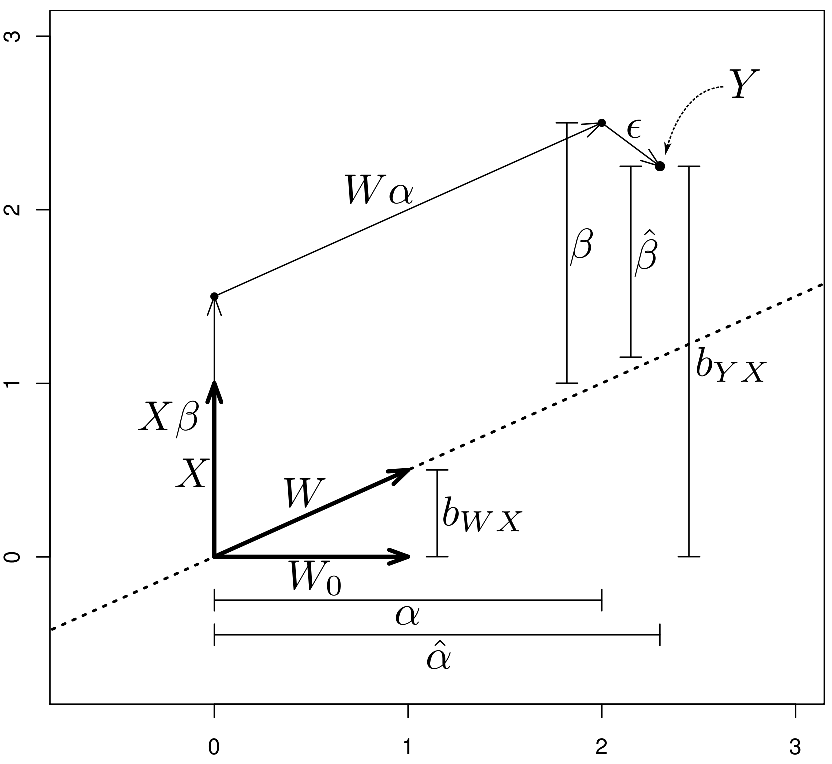


Figure 3.1: A graphical depiction of RUV-4. See main text for commentary.

$\hat{\beta}_j^{(\mathrm{RUV-4})}$ is the vertical distance from $Y_{\star j}$ to the orange dotted line, $\hat{\beta}_j^{(\mathrm{RUV-2})}$ is the vertical distance from $Y_{\star j}$ to the brown dotted line. The difference between RUV-4 and RUV-2 is essentially the difference between these two lines. We have seen that the orange dotted line is the regression line through the control genes. To see where the brown dotted line comes from, recall that in RUV-2 we estimate W by taking the first k eigenvectors of Y_cY_c'. Since $k = 1$ in this example, $\hat{W}^{(\mathrm{RUV-2})}$ is simply the principal eigenvector of Y_cY_c'. To assist in visualizing this, we have included in Figure 3.3A a green ellipse that "summarizes" the structure of Y_cY_c'. The major and minor axes of this ellipse are aligned with the first and second eigenvectors of Y_cY_c', and the lengths of the axes are proportional to the square roots of the associated eigenvalues. As we see in the plot, the brown dotted line goes directly along the major axis of this ellipse. Roughly speaking then, we may summarize the difference between RUV-2 and RUV-4 in the terminology of Freedman et al. (2007) as this: where RUV-2 uses the SD

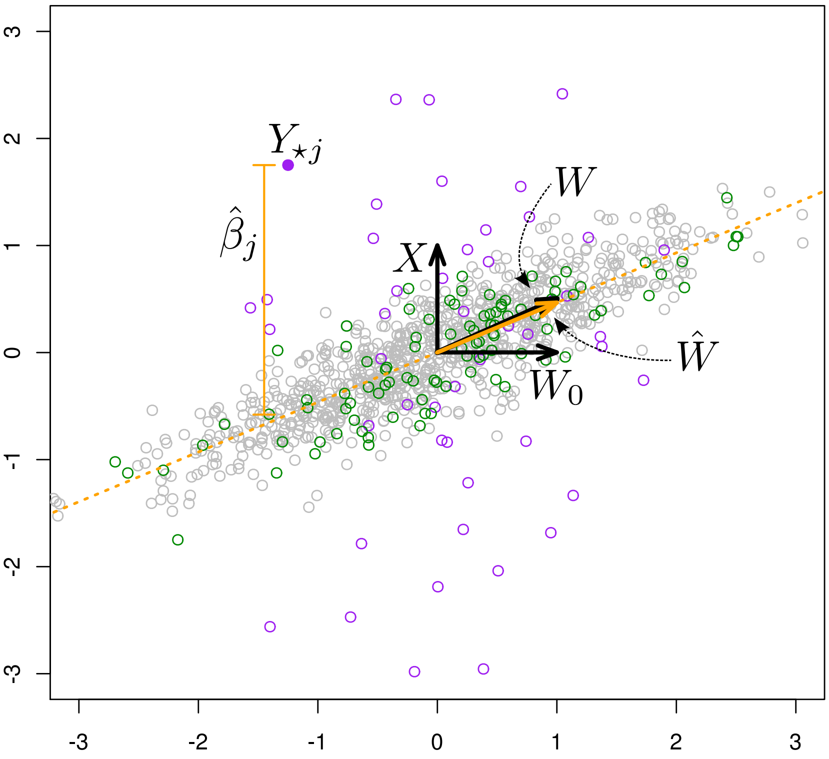


Figure 3.2: A graphical depiction of the estimation of b_{WX}. See main text for commentary. The simulated data were generated as follows: $X = (0, 1)'$; $W = (1, 0.5)'$; $\alpha_j \sim \mathrm{N}(0, 1)$; $\epsilon_{ij} \sim \mathrm{N}(0, \frac{1}{16})$; $\beta_j \sim \mathrm{N}(0, \frac{9}{4})$ for $1 \leq j \leq 50$; $\beta_j = 0$ for $51 \leq j \leq 1000$.

line, RUV-4 uses the regression line.[1]

In Figure 3.3A, there is little difference between $\hat{\beta}_j^{(\mathrm{RUV-2})}$ and $\hat{\beta}_j^{(\mathrm{RUV-4})}$. Figure 3.3B, however, provides an example in which the difference between $\hat{\beta}_j^{(\mathrm{RUV-2})}$ and $\hat{\beta}_j^{(\mathrm{RUV-4})}$ is quite substantial. The reason for the difference between the two estimators in Figure 3.3B is that the unwanted variation is less pronounced. While some unwanted variation from $W\alpha$ is clearly present, most of the scatter in the plot is purely random, i.e. comes from ϵ. Graphically, this can also be seen in the fact that the green ellipse is much more circular.

RUV-2 and RUV-4 are clearly different, but which is better? As we can see in Figure

[1] Strictly speaking this is not true, since the regression line and the SD line as defined in Freedman et al. (2007) both pass through the point of averages. We, however, force our lines to pass through the origin.

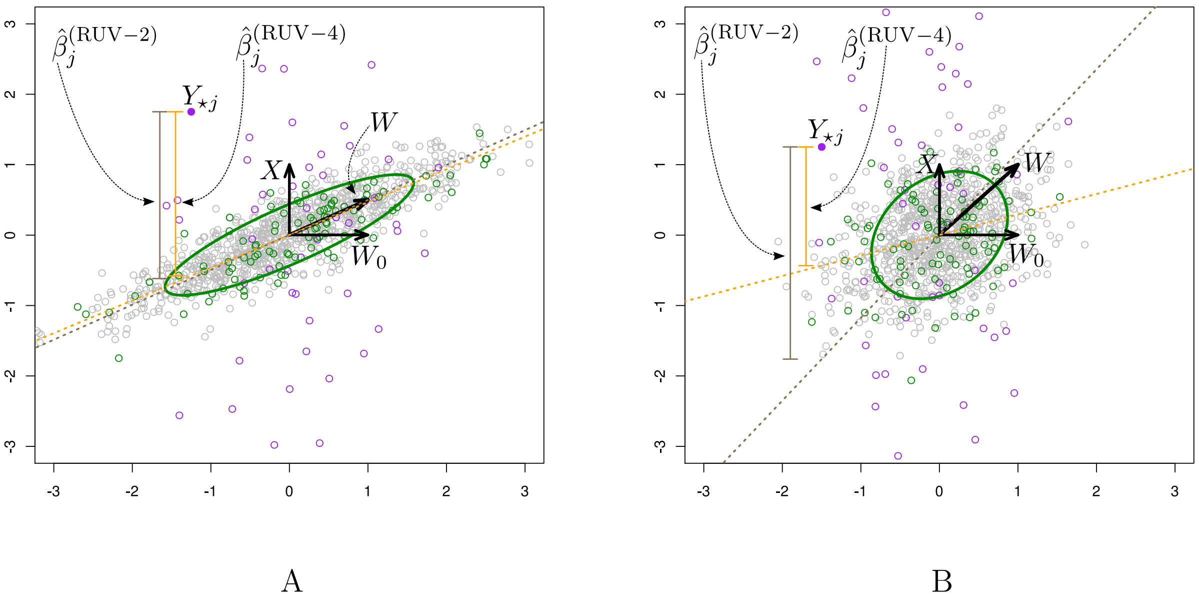


Figure 3.3: A comparison of RUV-4 and RUV-2. See main text for commentary. The simulated data in B were generated as follows: $X = (0,1)'$; $W = (1,1)'$; $\alpha_j \sim \mathrm{N}(0, \frac{9}{64})$; $\epsilon_{ij} \sim \mathrm{N}(0, \frac{1}{4})$; $\beta_j \sim \mathrm{N}(0, \frac{9}{4})$ for $1 \leq j \leq 50$; $\beta_j = 0$ for $51 \leq j \leq 1000$.

3.3B, RUV-2 seems to do a better job at accurately estimating W. On the other hand, it also appears that RUV-4 provides what our intuition might suggest to be the better estimate of β_j. This leads to a curious conclusion — that to get a better estimate of β_j, it may actually be a good idea to mis-estimate W. This peculiar observation is our first hint that the RUV-4 estimator may be more naturally formulated in the context of a different (or at least more fully specified) model. Indeed, we will see later that the RUV-4 estimator does arise more naturally in the context of a mixed effects model, in which α is explicitly modeled as random.

The differences between RUV-2 and RUV-4 are especially interesting when certain assumptions break down. In particular, we will now provide some intuition to suggest that RUV-4 may outperform RUV-2 when the control genes are not properly specified, or when k is overestimated. We begin with an example in which k is overestimated. The example is essentially the same as those in Figure 3.3, except that now in truth $k = 0$. Nonetheless, we still proceed to estimate W as if we had estimated $\hat{k} = 1$. In other words, $\hat{W} = \hat{W}^{(1)}$. Since in truth there is no unwanted variation $(W\alpha)$ term to adjust for, we would ideally like that no adjustment is made; we would like $\hat{b}_{WX}$ to equal 0 and our dotted line to be horizontal. Figure 3.4 shows three different simulations of this example. As we can see, RUV-4 does a much better job of providing no adjustment when none is needed; the orange dotted lines are relatively horizontal, while the brown dotted lines flap around wildly.

This example is particularly interesting in that it suggests that when performing RUV-4,

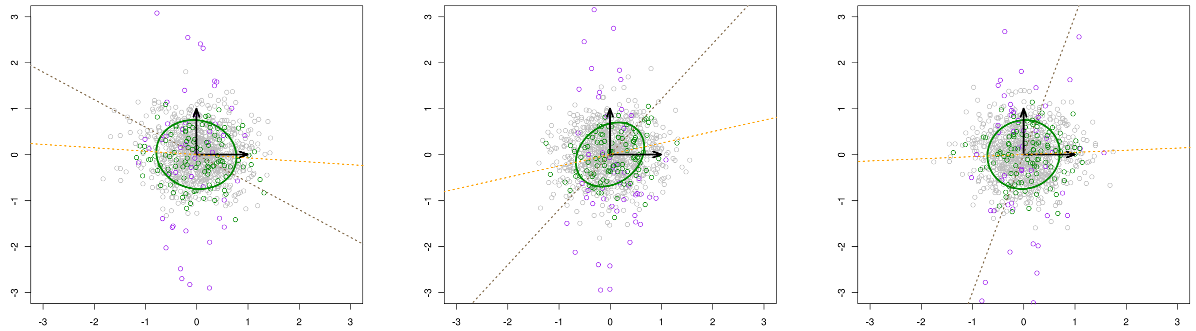


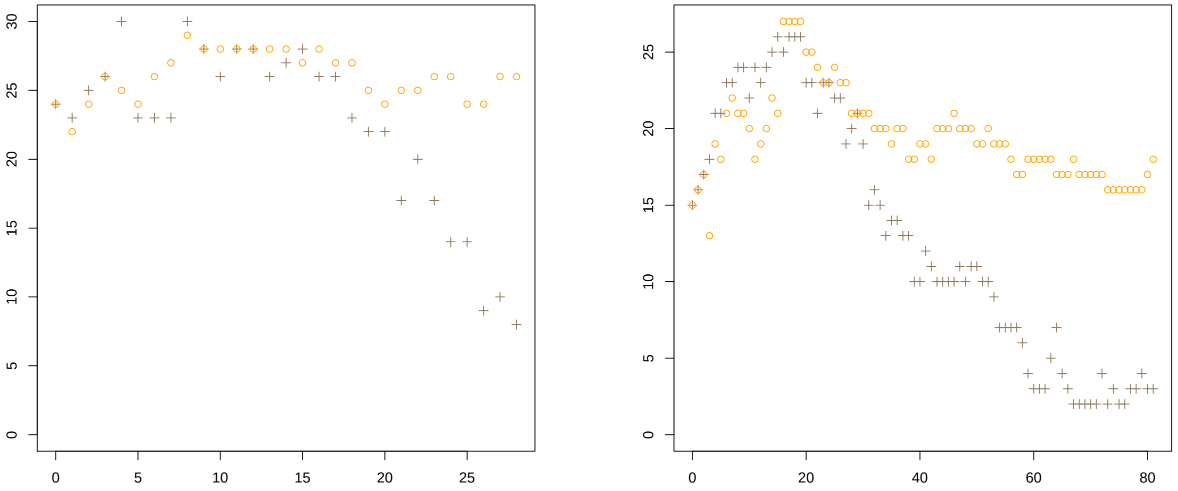
Figure 3.4: A comparison of RUV-4 and RUV-2 when k has been overestimated. See main text for commentary. The simulated data were generated as follows: $X = (0, 1)'$; $\epsilon_{ij} \sim \mathrm{N}(0, \frac{1}{4})$; $\beta_j \sim \mathrm{N}(0, \frac{9}{4})$ for $1 \leq j \leq 50$; $\beta_j = 0$ for $51 \leq j \leq 1000$.

we may in fact pay only a small price in performance if we overestimate k. Indeed, in this particularly simple example it is easy to see that $\mathbb{E}\left[\hat{b}_{WX}^{(\mathrm{RUV-4})}\right] = 0$, and that $\mathrm{Var}\left[\hat{b}_{WX}^{(\mathrm{RUV-4})}\right]$ approaches 0 as the number of control genes grows large. With a large enough set of control genes, there is effectively no harm at all in overestimating k. Figure 3.5 makes this point using real data. In both the Alzheimer's and Gender datasets, the performance of RUV-2 drops substantially if K is too high. The performance of RUV-4 however remains good even for very high K. Indeed RUV-4 performs at least as well as the unadjusted case ($K = 0$) for nearly every possible K.

Finally, note that although the performance of RUV-4 does decrease somewhat if K is set too large, this is not necessarily an indication that the performance of $\hat{\beta}^{(\mathrm{RUV-4})}$ is decreasing with large K. Estimates of the variances (i.e. the $\hat{\sigma}_j^2$) may instead be to blame. Indeed, if we use standard methods instead of Limma (Smyth, 2004) to estimate $\hat{\sigma}^2$, the performance of RUV-4 is nearly as bad as the performance of RUV-2 for large K (data not shown). The performance of RUV-4 as a whole depends on getting good estimates of both β and σ^2. We have demonstrated there is reason to believe that $\hat{\beta}^{(\mathrm{RUV-4})}$ performs well even for very high K. Getting good estimates of σ^2 is a separate challenge. We will return to this point in Section 3.3.7.

We now consider an example in which the control genes are misspecified. Figure 3.6A shows an example similar to those in Figure 3.3. 100 genes are (correctly) designated as control genes, and colored green. Figure 3.6B shows the same example dataset, but now 110 genes are designated as control genes — the same 100 as in Figure 3.6A, plus an additional 10 that have been incorrectly designated as control genes. These 10 misspecified control genes are plotted as purple circles filled in with a green dot.

As we can see, the inclusion of these 10 misspecified control genes does not substantially



Alzheimer's

Gender

Figure 3.5: Comparison of the performance of RUV-2 (brown cross) and RUV-4 (orange circle) as a function of K in the Alzheimer's and Gender studies. The horizontal axis is K. $K = 0$ corresponds to no adjustment. For each K, genes were ranked by p-value; the vertical axis is the number of X / Y genes ranked in the top 60. Data were preprocessed. Housekeeping genes were used as control genes. p-values were computed using Limma.

affect the RUV-4 estimate of $\hat{W}$, but does affect the RUV-2 estimate. It is easy to see why the RUV-2 estimate is affected. The additional scatter in the vertical direction introduced by the 10 misspecified control genes "pulls" the principal eigenvector into a more vertical position. Why is the RUV-4 estimate not also affected? The reason is that there is not a strong correlation between α and β. The misspecified control genes in this example introduce additional vertical scatter, but they are not, for example, systematically too high on the right (where α is positive) and too low on the left (where α is negative). The slope of the regression line is therefore largely unaffected.

We see then that RUV-4 is relatively robust to misspecification of the control genes. As we show in Section 3.3.5, RUV-4 does not require that $\beta_c = 0$ but only that $\beta_c \alpha_c' = 0$. This is clearly a much weaker requirement of the control genes. It is also a fairly mysterious requirement that is hard to interpret and nearly impossible to verify. Thus, the extent to which we can exploit this weaker requirement is limited. For example, we may be tempted to reason loosely that W contains mainly technical factors (e.g. temperature of the scanner, etc.), and that the effects of the technical factors (i.e. α) should not relate in any systematic way to the effects of biological factors (i.e. β), and thus that $\beta_c \alpha_c' \approx 0$. Why, we might reason, should the genes that are up-regulated by cancer also be the genes whose observations are biased upwards by an overly warm scanner? And yet, this could very well be. For example, if the genes that are up-regulated by cancer tend to be genes that are highly expressed, and genes whose observations are biased upwards by an overly warm scanner also tend to be genes that are highly expressed, we would get just that scenario. Instead, we view the relatively weak requirements placed on the control genes by RUV-4 not as something to exploit per se, but rather as a comforting reassurance that even if we accidentally misspecify some control genes, we may still get lucky and end up OK — or at least not as badly off as if we had used RUV-2.

Figures 3.6C and 3.6D provide an even starker example of the difference between RUV-2 and RUV-4 when control genes are misspecified. As in Figure 3.6A, in Figure 3.6C there are 100 genes correctly designated as control genes. Figure 3.6D shows the same example dataset as Figure 3.6C, but with an additional 10 misspecified control genes. The difference between Figures 3.6C and 3.6D and Figures 3.6A and 3.6B is that in Figures 3.6C and 3.6D, k has been overestimated. As in Figure 3.5, $k = 0$ but $K = 1$. The combination of misspecified control genes and an overestimated k is particularly damaging to RUV-2. Since there is no more unwanted variation, the only remaining systematic variation in the control genes comes from the misspecified control genes — and is in the X direction. As we can see, the slope of the brown dotted line in Figure 3.6D is much steeper than in Figure 3.6C. In this example, RUV-2 clearly detects — and adjusts away — the biological factor of interest. On the other hand, the slope of the orange dotted line is largely unaffected. RUV-4 is relatively robust to misspecification of control genes and overestimation of k, even in combination.

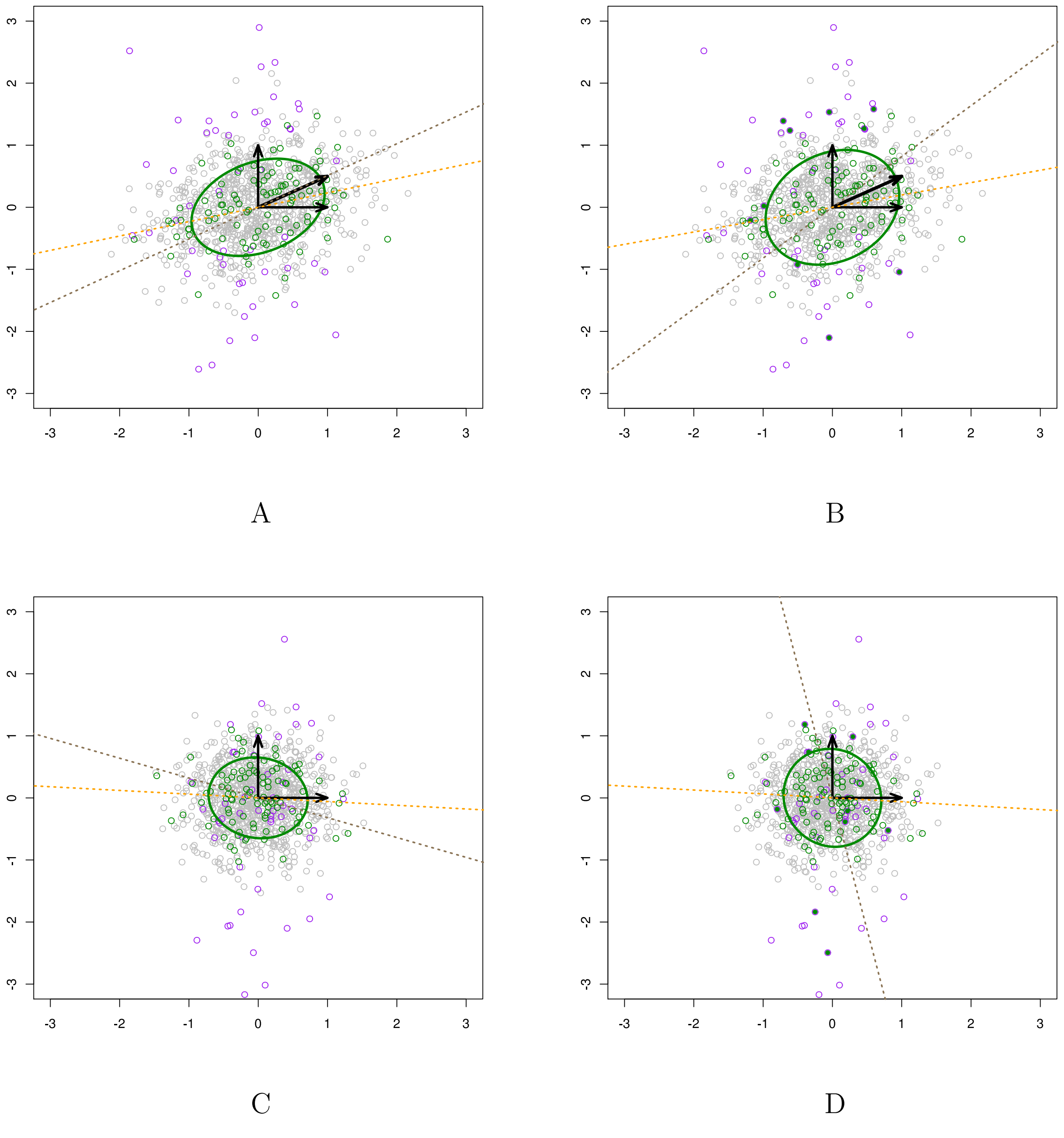


Figure 3.6: A comparison of RUV-4 and RUV-2 when control genes have been misspecified. See main text for commentary. The simulated data in A and B were generated as follows: $X = (0, 1)'$; $W = (1, 0.5)'$; $\alpha_j \sim \mathrm{N}(0, \frac{1}{4})$; $\epsilon_{ij} \sim \mathrm{N}(0, \frac{1}{4})$; $\beta_j \sim \mathrm{N}(0, \frac{9}{4})$ for $1 \leq j \leq 50$; $\beta_j = 0$ for $51 \leq j \leq 1000$. The simulated data in C and D were generated as follows: $X = (0, 1)'$; $\epsilon_{ij} \sim \mathrm{N}(0, \frac{1}{4})$; $\beta_j \sim \mathrm{N}(0, \frac{9}{4})$ for $1 \leq j \leq 50$; $\beta_j = 0$ for $51 \leq j \leq 1000$.

3.3.5 Statistical Properties of RUV-4

We now explore the bias and variance of $\hat{\beta}$. Calculating the bias and variance requires that we specify whether α is to be regarded as fixed or random. Ideally we would like to analyze the bias and variance of $\hat{\beta}$ under the assumption that α is fixed. However, this is difficult. Therefore we will occasionally regard α as random. To make the discussion consistent, we will always formally treat α as if it is random, but condition on α when appropriate.

Unfortunately, exact calculations of the bias and variance are difficult if not impossible for a variety of reasons, so instead we focus on simplifications and approximations that are both illuminating and reasonably accurate. One complication is that the statistical properties of $\hat{\beta}_j$ depend on whether the j^{th} gene is a control gene or not. In what follows we limit our discussion to a gene $j_{\bar{c}}$ that is not a control gene.

Another complication when calculating the bias and variance of $\hat{\beta}$ is that we must know the statistical properties of $\hat{W}_0$ and $\hat{\alpha}$. These properties depend on the choice of factor analysis method — and are difficult to calculate in any case. We therefore simplify the situation by making use of a hypothetical idealized factor analysis method. Specifically, we suppose that $\hat{W}_0 = W_0$ and thus that $\hat{\alpha} = \left(W_0'W_0\right)^{-1} W_0'Y$. In other words, we imagine we have access to an idealized factor analysis method that perfectly estimates W_0; with this we then additionally estimate α by OLS as specified in (3.23). This simplification is not entirely unrealistic; with tens of thousands of genes, estimates of W_0 can be quite good.

3.3.5.1 Results for fixed α

We begin by defining

$$\begin{aligned}\zeta &\equiv (X'X)^{-1}X'\epsilon \\ \xi &\equiv \left(W_0'W_0\right)^{-1} W_0'\epsilon\end{aligned}$$

and noting that

$$\begin{aligned}\zeta_{j_{\bar{c}}} &\sim \mathrm{N}(0, \sigma^2_{j_{\bar{c}}}) \\ \xi_{\star j_{\bar{c}}} &\sim \mathrm{N}(0, \sigma^2_{j_{\bar{c}}} I) \\ \hat{b}_{WX} &\perp\!\!\!\perp \zeta_{j_{\bar{c}}} \\ \hat{b}_{WX} &\perp\!\!\!\perp \xi_{\star j_{\bar{c}}} \\ \zeta_{j_{\bar{c}}} &\perp\!\!\!\perp \xi_{\star j_{\bar{c}}}.\end{aligned}$$

It follows that

$$\begin{aligned}\hat{\beta} &= \left(X'X\right)^{-1} X' \left(Y - \hat{W}\hat{\alpha}\right) \\ &= \left(X'X\right)^{-1} X' \left[X\beta + \left(W_0 + Xb_{WX}\right)\alpha + \epsilon - \left(W_0 + X\hat{b}_{WX}\right)\hat{\alpha}\right] \\ &= \beta + b_{WX}\alpha - \hat{b}_{WX}\hat{\alpha} + \zeta \\ &= \beta + b_{WX}\alpha - \hat{b}_{WX}\left[\alpha + \left(W_0'W_0\right)^{-1} W_0'\epsilon\right] + \zeta \\ &= \beta + \left(b_{WX} - \hat{b}_{WX}\right)\alpha + \zeta - \hat{b}_{WX}\xi\end{aligned}$$

and that

$$\begin{aligned}\mathbb{E}\left[\hat{\beta}_{j_{\bar{c}}}\right] &= \beta_{j_{\bar{c}}} + \mathbb{E}\left[b_{WX} - \hat{b}_{WX}\right]\mathbb{E}\left[\alpha_{\star j_{\bar{c}}}\right] \\ \mathrm{Var}\left[\hat{\beta}_{j_{\bar{c}}}\right] &= \mathrm{Var}\left[\zeta_{j_{\bar{c}}}\right] + \mathrm{Var}\left[\hat{b}_{WX}\left(\alpha_{\star j_{\bar{c}}} + \xi_{\star j_{\bar{c}}}\right)\right] \\ &= \sigma^2_{j_{\bar{c}}} + \mathbb{E}\left[\alpha'_{\star j_{\bar{c}}}\right]\mathrm{Var}\left[\hat{b}_{WX}\right]\mathbb{E}\left[\alpha_{\star j_{\bar{c}}}\right] + \mathbb{E}\left[\hat{b}_{WX}\right]\mathrm{Var}\left[\alpha_{\star j_{\bar{c}}} + \xi_{\star j_{\bar{c}}}\right]\mathbb{E}\left[\hat{b}'_{WX}\right] + \\ &\quad \mathrm{tr}\left(\mathrm{Var}\left[\hat{b}_{WX}\right]^{\frac{1}{2}}\mathrm{Var}\left[\alpha_{\star j_{\bar{c}}} + \xi_{\star j_{\bar{c}}}\right]\mathrm{Var}\left[\hat{b}_{WX}\right]^{\frac{1}{2}}\right).\end{aligned}$$

If we now condition on $\alpha_{j_{\bar{c}}}$ we find that

$$\begin{aligned}\mathbb{E}\left[\hat{\beta}_{j_{\bar{c}}}\middle|\alpha_{\star j_{\bar{c}}}\right] &= \beta_{j_{\bar{c}}} + \mathbb{E}\left[b_{WX} - \hat{b}_{WX}\right]\alpha_{\star j_{\bar{c}}} \\ \mathrm{Var}\left[\hat{\beta}_{j_{\bar{c}}}\middle|\alpha_{\star j_{\bar{c}}}\right] &= \sigma^2_{j_{\bar{c}}} + \alpha'_{\star j_{\bar{c}}}\mathrm{Var}\left[\hat{b}_{WX}\right]\alpha_{\star j_{\bar{c}}} + \sigma^2_{j_{\bar{c}}}\mathbb{E}\left[\hat{b}_{WX}\right]\mathbb{E}\left[\hat{b}'_{WX}\right] + \sigma^2_{j_{\bar{c}}}\mathrm{tr}\left(\mathrm{Var}\left[\hat{b}_{WX}\right]\right)\end{aligned}$$

and conclude

$$\mathrm{Bias}\left[\hat{\beta}_{j_{\bar{c}}}\middle|\alpha_{\star j_{\bar{c}}}\right] = \mathbb{E}\left[b_{WX} - \hat{b}_{WX}\right]\alpha_{\star j_{\bar{c}}} \tag{3.38}$$

$$\begin{aligned}\mathrm{Var}\left[\hat{\beta}_{j_{\bar{c}}}\middle|\alpha_{\star j_{\bar{c}}}\right] &= \sigma^2_{j_{\bar{c}}}\left\{1 + \mathbb{E}\left[\hat{b}_{WX}\right]\mathbb{E}\left[\hat{b}'_{WX}\right]\right\} \\ &\quad + \sigma^2_{j_{\bar{c}}}\mathrm{tr}\left(\mathrm{Var}\left[\hat{b}_{WX}\right]\right) + \alpha'_{\star j_{\bar{c}}}\mathrm{Var}\left[\hat{b}_{WX}\right]\alpha_{\star j_{\bar{c}}}.\end{aligned} \tag{3.39}$$

Recall that these expressions do not take into account any bias or variance introduced by the estimation of W_0, but are otherwise exact.

3.3.5.2 Discussion: $\mathbb{E}\left[\hat{b}_{WX}\right]$

We now consider $\mathbb{E}\left[\hat{b}_{WX}\right]$. As we will soon see, this leads us to an error-in-variables regression problem. Begin by noting

$$\begin{aligned}
\hat{b}_{WX} &= b_{Y_c X}\hat{\alpha}_c'\left(\hat{\alpha}_c\hat{\alpha}_c'\right)^{-1} && (3.40)\\
&= (X'X)^{-1}X'Y_c\left(\alpha_c+\xi_c\right)'\left[\left(\alpha_c+\xi_c\right)\left(\alpha_c+\xi_c\right)'\right]^{-1} && (3.41)\\
&= \left(\beta_c+b_{WX}\alpha_c+\zeta_c\right)\left(\alpha_c+\xi_c\right)'\left[\left(\alpha_c+\xi_c\right)\left(\alpha_c+\xi_c\right)'\right]^{-1} && (3.42)\\
&= \left(b_{WX}\alpha_c+\zeta_c\right)\left(\alpha_c+\xi_c\right)'\left[\left(\alpha_c+\xi_c\right)\left(\alpha_c+\xi_c\right)'\right]^{-1}\\
&\quad +\beta_c\left(\alpha_c+\xi_c\right)'\left[\left(\alpha_c+\xi_c\right)\left(\alpha_c+\xi_c\right)'\right]^{-1}. && (3.43)
\end{aligned}$$

If the control gene assumption $\beta_c = 0$ holds, the second term vanishes, and we are left with

$$\hat{b}_{WX} = \left(b_{WX}\alpha_c+\zeta_c\right)\left(\alpha_c+\xi_c\right)'\left[\left(\alpha_c+\xi_c\right)\left(\alpha_c+\xi_c\right)'\right]^{-1}.$$

This would be the OLS estimator for the parameter b_{WX} in a regression of the "response variable" $b_{WX}\alpha_c+\zeta_c$ on the "explanatory variable" α_c, were it not for the fact that the "explanatory variable" α_c has been corrupted by the error term ξ_c.

3.3.5.3 Results for random α

Error-in-variables regression problems are difficult to analyze, and general expressions for the bias and variance are not known. We are therefore unable to analyze $\mathbb{E}\left[\hat{b}_{WX}\right]$ in general. Instead, we attempt to gain some insight into $\mathbb{E}\left[\hat{b}_{WX}\right]$ by considering a simple, specific example. Assume that for all control genes

$$\begin{aligned}
\alpha_{\star j_c} &\sim \mathrm{N}(0,\Psi^2) && (3.44)\\
\sigma^2_{j_c} &= \sigma^2_0 && (3.45)
\end{aligned}$$

where σ_0^2 is some fixed constant. Assume without a loss in generality that Ψ^2 is diagonal. Assume also that n_c is large. Then

$$\begin{aligned}
\hat{b}_{WX} &= \left(b_{WX}\alpha_c+\zeta_c\right)\left(\alpha_c+\xi_c\right)'\left[\left(\alpha_c+\xi_c\right)\left(\alpha_c+\xi_c\right)'\right]^{-1}\\
&= \left(\frac{b_{WX}\alpha_c\alpha_c'}{n_c}+\frac{\zeta_c\alpha_c'}{n_c}+\frac{b_{WX}\alpha_c\xi_c'}{n_c}+\frac{\zeta_c\xi_c'}{n_c}\right)\left(\frac{\alpha_c\alpha_c'}{n_c}+\frac{\xi_c\alpha_c'}{n_c}+\frac{\alpha_c\xi_c'}{n_c}+\frac{\xi_c\xi_c'}{n_c}\right)^{-1}\\
&\approx b_{WX}\Psi^2\left(\Psi^2+\sigma_0^2 I\right)^{-1}.
\end{aligned}$$

We therefore see that the entries of $\hat{b}_{WX}$ are asymptotically biased towards 0. In particular, the $(1,l)^{\text{th}}$ entry of $\hat{b}_{WX}$ (which we denote $(\hat{b}_{WX})_l$) is asymptotically biased by a factor of $\frac{\psi_l^2}{\psi_l^2+\sigma_0^2}$, where ψ_l^2 is the l^{th} diagonal entry of Ψ^2. This observation agrees with the intuition we developed in Section (3.3.4) — $\hat{b}_{WX}$ is biased towards 0, and the bias grows stronger as the unwanted variation becomes weaker.

Under assumptions (3.44) and (3.45) and the assumption that n_c is large, we may simplify (3.38) as

$$\text{Bias}\left[\hat{\beta}_{j\bar{c}}\middle|\alpha_{\star j\bar{c}}\right] \approx b_{WX}\left(\Psi^2/\sigma_0^2+I\right)^{-1}\alpha_{\star j\bar{c}}. \tag{3.46}$$

Moreover, under the same assumptions $\text{Var}\left[\hat{b}_{WX}\right]\approx 0$, so we may we may simplify (3.39) as

$$\begin{aligned}\text{Var}\left[\hat{\beta}_{j\bar{c}}\middle|\alpha_{\star j\bar{c}}\right] &\approx \sigma_{j\bar{c}}^2\left\{1+\mathbb{E}\left[\hat{b}_{WX}\right]\mathbb{E}\left[\hat{b}'_{WX}\right]\right\} &(3.47)\\ &\approx \sigma_{j\bar{c}}^2\left(1+b_{WX}\Psi^4\left(\Psi^2+\sigma_0^2I\right)^{-2}b'_{WX}\right). &(3.48)\end{aligned}$$

From this we conclude

$$\begin{aligned}\text{MSE}\left[\hat{\beta}_{j\bar{c}}\middle|\alpha_{\star j\bar{c}}\right] \approx\; &\sigma_{j\bar{c}}^2+b_{WX}\left[\left(\Psi^2/\sigma_0^2+I\right)^{-1}\alpha_{\star j\bar{c}}\alpha'_{\star j\bar{c}}\left(\Psi^2/\sigma_0^2+I\right)^{-1}\right]b'_{WX}\\ &+\sigma_{j\bar{c}}^2 b_{WX}\left[\Psi^4\left(\Psi^2+\sigma_0^2I\right)^{-2}\right]b'_{WX}. \qquad (3.49)\end{aligned}$$

This is a complicated expression and somewhat difficult to interpret. It is clear that, unless $b_{WX}=0$, the MSE can be arbitrarily large depending on the value of $\alpha_{\star j\bar{c}}$. However, it is not clear how large the MSE might be in a "typical" case. To investigate this question, we will now assume

$$\begin{aligned}\alpha_{\star j\bar{c}} &\sim \text{N}(0,\Psi^2) &(3.50)\\ \sigma_{j\bar{c}}^2 &= \sigma_0^2 &(3.51)\end{aligned}$$

just as we did for the control genes. Under these assumptions

$$\begin{aligned}\text{MSE}\left[\hat{\beta}_j\right] &\approx \sigma_0^2+b_{WX}\left[\left(\Psi^2/\sigma_0^2+I\right)^{-1}\Psi^2\left(\Psi^2/\sigma_0^2+I\right)^{-1}\right]b'_{WX}\\ &\quad +\sigma_0^2b_{WX}\left[\Psi^4\left(\Psi^2+\sigma_0^2I\right)^{-2}\right]b'_{WX} &(3.52)\\ &= \sigma_0^2+b_{WX}\left(\Psi^2/\sigma_0^2+I\right)^{-2}\left(\Psi^2+\Psi^4/\sigma_0^2\right)b'_{WX} &(3.53)\\ &= \sigma_0^2+b_{WX}\Psi^2\left(\Psi^2/\sigma_0^2+I\right)^{-1}b'_{WX} &(3.54)\\ &= \sigma_0^2\left[1+b_{WX}\Psi^2\left(\Psi^2+\sigma_0^2I\right)^{-1}b'_{WX}\right]. &(3.55)\end{aligned}$$

It may not be immediately obvious whether (3.55) is "good" or "bad." For sake of comparison, suppose we knew W, and could simply estimate β by OLS. Designate this hypothetical estimator by $\hat{\beta}^{(\text{OLS})}$. Now, $\hat{\beta}^{(\text{OLS})}$ is unbiased, so the MSE is simply the variance:

$$\text{MSE}\left[\hat{\beta}_j^{(\text{OLS})}\right] = \sigma_0^2\left(1+b_{WX}b'_{WX}\right). \tag{3.56}$$

(See Section B.1.2 for a proof.) Thus, under assumptions (3.44), (3.45), (3.50), (3.51) and the assumption that n_c is large, we see that $\mathrm{MSE}\left[\hat{\beta}_j^{(\mathrm{RUV-4})}\right] < \mathrm{MSE}\left[\hat{\beta}_j^{(\mathrm{OLS})}\right]$, at least up to approximation:

$$\begin{aligned}\mathrm{MSE}\left[\hat{\beta}_j^{(\mathrm{OLS})}\right] - \mathrm{MSE}\left[\hat{\beta}_j^{(\mathrm{RUV-4})}\right] &\approx \sigma_0^2 b_{WX}\left[I - \Psi^2\left(\Psi^2 + \sigma_0^2 I\right)^{-1}\right] b'_{WX} &(3.57)\\ &= \sigma_0^2 \sum_{l=1}^{k} \frac{\sigma_0^2}{\psi_l^2 + \sigma_0^2} (b_{WX})_{1l}^2. &(3.58)\end{aligned}$$

3.3.5.4 Discussion: Random or Fixed?

In the previous section we analyzed the RUV-4 estimator under the assumption that α is random. This was motivated by the fact that it is very difficult to analyze the RUV-4 estimator under the assumption that α is fixed. Regarding α as random allowed us to develop some intuition about the behavior of the RUV-4 estimator that we might not have been able to develop otherwise. We found that under our assumptions the RUV-4 estimator actually has a smaller MSE than the (hypothetical) OLS estimator. This result may be surprising at first, but can be easily understood as RUV-4 exploiting the assumption that the $\alpha_{\star j_{\bar{c}}}$ and the $\alpha_{\star j_c}$ are drawn from the same normal distribution. The conclusion is that RUV-4 has the potential to outperform OLS. For it to do so, however, the control genes must satisfy an additional criterion. The control genes must not only be uninfluenced by X yet influenced by W, but they must also be influenced by W in much the same way as all of the other genes are. The $\alpha_{\star j_c}$ must be "representative" of the $\alpha_{\star j_{\bar{c}}}$.

In Section 3.3.7 we will reformulate the RUV-4 estimator. Under this reformulation, we will see that the RUV-4 estimator arises very naturally from a model in which α is random. In Section 3.5 we will observe that RUV-4 works very well when applied to real datasets.

A natural question to ask, then, is whether we should regard α as random. This question does not have an easy answer. The RUV model is highly artificial. We regard the model primarily as a source of inspiration for new methods; the value of these methods must then be established independently, by testing how well they perform on real data. In this context, it may be wise to regard α as random. RUV-4 is an effective method that arises naturally from a model in which α is random. Regarding α as random may ultimately inspire ideas for even more effective methods.

On the other hand, neither the effectiveness nor the "naturalness" of an estimator can ultimately justify the model from which the estimator arose. Moreover, despite the fact the RUV model is artificial, there is an obvious interest in keeping the model as realistic as feasible. In this context, it seems wise to regard α as fixed. Random effects seem implausible; we are unaware of any physical argument that would justify regarding α as random, let alone the distributional assumptions we have imposed on α. For example, if $\alpha_{4,25}$ is the effect of temperature on the observed expression level of the 25th probe, should we not expect $\alpha_{4,25}$ to be dictated by the physical and chemical properties of the 25th probe, and thus to be constant from one experiment to the next?

Instead of regarding α as random, it may be better to regard α as fixed with some (non-random) distribution that can often be reasonably approximated in practice by a normal distribution. The distinction may seem pedantic, but we believe regarding α in this manner is useful. For example, it serves as a reminder that some α_{ij} may be serious outliers, and that these outliers may reveal interesting information upon further investigation. It also serves as a reminder that different genes may have different biases. It may be that $\hat{\beta}$ is consistently biased one way or another for genes with high GC content, or for genes that are highly expressed. If these biases are consistent from one experiment to the next, they may lead to inaccurate conclusions, despite the replication. We believe it is helpful to keep such considerations in mind.

To summarize, there is no clear answer as to whether α should be regarded as fixed or random. Both points of view are useful in their own way. Moreover, a decisive answer may not be necessary. In Section 3.3.8 we will provide still another framework for understanding RUV-4 that partly sidesteps the issue. Finally, and most importantly, we reiterate that we have seen that the superior performance of RUV-4 relies on the $\alpha_{\star j_c}$ being "representative" of the $\alpha_{\star j_{\bar{c}}}$. This clearly has important practical implications for choosing a set of control genes.

3.3.6 Practical considerations for RUV-4

We now consider several topics including the estimation of σ^2, $\text{Var}\left[\hat{\beta}_{j_{\bar{c}}}\middle|\alpha_{\star j_{\bar{c}}}\right]$ and k; the handling of covariates; the consequences of misspecified control genes; and the consequences of under- or over-estimating k. We continue to formally treat α as if it is random, conditioning when appropriate.

3.3.6.1 Estimating σ^2 and $\text{Var}\left[\hat{\beta}_{j_{\bar{c}}}\middle|\alpha_{\star j_{\bar{c}}}\right]$

Suppose we want to estimate σ_j^2. If we disregard the fact that $\hat{W}$ is an estimate and simply treat it as W, the "standard" estimate for σ_j^2 is:

$$\begin{aligned}\hat{\sigma}_j^2 &\equiv \frac{1}{m-\hat{k}-1}\sum_{i=1}^{m}\left(Y_{ij}-X\hat{\beta}_j-\hat{W}\hat{\alpha}_{\star j}\right)^2 &(3.59)\\ &= \frac{1}{m-\hat{k}-1}\left(R_{(X|\hat{W})}Y_{\star j}\right)'\left(R_{(X|\hat{W})}Y_{\star j}\right). &(3.60)\end{aligned}$$

$R_{(X|\hat{W})}$ depends only on the column space of $(X|\hat{W})$, and the columns of X and $\hat{W}_0$ together form a basis of $\mathfrak{R}\left[(X|\hat{W})\right]$, so $R_{(X|\hat{W})} = R_{(X|\hat{W}_0)}$. Thus $\hat{\sigma}_j^2$ is independent of our estimate of b_{WX}. Indeed, if $\hat{W}_0 = W_0$ then $R_{(X|\hat{W})} = R_{(X|W)}$ and $\hat{\sigma}_j^2$ is identical to the "standard" estimate of σ_j^2 that we would get if W were known.

Now suppose we want to estimate $\mathrm{Var}\left[\hat{\beta}_{j\bar{c}}\middle|\alpha_{\star j\bar{c}}\right]$. Recall that when n_c is large,

$$\mathrm{Var}\left[\hat{\beta}_{j\bar{c}}\middle|\alpha_{\star j\bar{c}}\right] \approx \sigma^2_{j\bar{c}}\left\{1+\mathbb{E}\left[\hat{b}_{WX}\right]\mathbb{E}\left[\hat{b}'_{WX}\right]\right\} \tag{3.61}$$

$$\approx \sigma^2_{j\bar{c}}\left(1+\hat{b}_{WX}\hat{b}'_{WX}\right). \tag{3.62}$$

Given our estimate $\hat{\sigma}^2_{j\bar{c}}$ of $\sigma^2_{j\bar{c}}$, we may therefore estimate $\mathrm{Var}\left[\hat{\beta}_{j\bar{c}}\middle|\alpha_{\star j\bar{c}}\right]$ as

$$\widehat{\mathrm{Var}}\left[\hat{\beta}_{j\bar{c}}\middle|\alpha_{\star j\bar{c}}\right] \equiv \hat{\sigma}^2_{j\bar{c}}\left(1+\hat{b}_{WX}\hat{b}'_{WX}\right). \tag{3.63}$$

This is a particularly appealing result since (3.63) is equivalent to the standard estimate of $\mathrm{Var}\left[\hat{\beta}_{j\bar{c}}\middle|\alpha_{\star j\bar{c}}\right]$ that we would calculate if we simply treated $\hat{W}$ as W and ran a standard regression (see Section B.1.2 of the appendix for proof).

To summarize, if n_c is large and $\hat{W}_0 \approx W_0$ then plugging in $\hat{W}$ for W and and estimating σ^2 and $\mathrm{Var}\left[\hat{\beta}_{j\bar{c}}\middle|\alpha_{\star j\bar{c}}\right]$ in the standard way will provide satisfactory results. Thus, once we have computed $\hat{W}$ using RUV-4, standard methods and software can be used to estimate variances, calculate t statistics and p-values, etc. However, we should note that even under these "ideal" conditions, p-values computed in the standard way are not necessarily reliable, since $\hat{\beta}_{j\bar{c}}$ is biased (conditional on $\alpha_{\star j\bar{c}}$). Even more importantly, we note that a major implicit assumption in our analysis is that we have properly estimated k. Serious problems may occur if we mis-estimate k. See Section 3.3.6.5. Finally, note that strictly speaking (3.63) is only appropriate for non-control genes. In practice however, we apply it to the control genes as well.

3.3.6.2 Handling Covariates, and the Case $p > 1$

In (3.7) we made the simplifying assumptions that there is no Z term and that $p = 1$. We now consider what to do when this is not true. We will first consider the case that $p = 1$ but that observed covariates Z are available. We will then consider the case that $p > 1$. We find that both cases can be reduced to the case that there is no Z term and $p = 1$. However, there is more than one way to reduce to the case that there is no Z term and $p = 1$, and it is not always clear which way is best.

Suppose observed covariates Z are available. There are two options. The first is to simply ignore them. RUV-4 can then estimate these unwanted factors and incorporate them into $\hat{W}$ just as it would any other unwanted factors. This is often the best option. In Chapter 2 we argue that the observed covariates that one has available are often only proxies for the "true" unwanted factors. For example, "batch" itself does not cause "batch effects." Batch effects are the result of other factors (e.g. temperature) that are correlated with batch. Compared to the proxy variables in Z, factor analysis may provide a better estimate of the "true" unwanted factors.

The second option is to explicitly adjust for Z. Multiplying both sides of (3.1) by $Z_\perp'$ yields

$$Z_\perp' Y = (Z_\perp' X)\beta + (Z_\perp' W)\alpha + Z_\perp' \epsilon. \tag{3.64}$$

Note that

$$Z_\perp' \epsilon_{\star j} \sim N(0, \sigma_j^2 I_{(m-q)\times(m-q)}).$$

We may therefore simply use $Z_\perp' Y$ and $Z_\perp' X$ instead of Y and X, and proceed as we would in the case that there is no Z term. Note that we have effectively "projected" Z away. However, strictly speaking this is not a "projection" in the technical sense, since we premultiply by $Z_\perp$ instead of by R_Z; instead of mapping the data from $\mathbb{R}^m$ to a $m-q$ dimensional subspace of $\mathbb{R}^m$, we map the data from $\mathbb{R}^m$ to $\mathbb{R}^{m-q}$.

When should we explicitly adjust for Z and when should we just ignore it? A few observations are relevant. One such observation is that a poor proxy variable can create more problems than it solves. Assume that in truth

$$Y = X\beta + Z\gamma + \epsilon \tag{3.65}$$

but that we regress Y on X and $\tilde{Z}$, where $\tilde{Z}$ is correlated with but not equal to Z. It is possible that the bias of $\hat{\beta}$ in this case is even larger than if we had simply regressed Y on X and ignored $\tilde{Z}$. This is of particular concern if $\tilde{Z}$ is highly correlated with X. Another observation is that by explicitly including a Z term, we are effectively treating γ as fixed. By letting RUV-4 incorporate Z into W and γ into α, we are effectively treating γ as random. By explicitly including a Z term, we may therefore lose some of the performance advantage offered by RUV-4. A final observation is that explicitly adjusting for Z may hinder our ability to estimate W and α. If W and Z are correlated, projecting away Z will also project away some of W. This will make the estimation of W and α more difficult. If the factors contained in Z are less important than those contained in W (i.e $Z\gamma$ is "smaller" than $W\alpha$), it may be better to ignore Z for sake of a better estimate of $W\alpha$. A rather extreme example may help make this point more clear. Suppose there are $m-1$ known covariates. Suppose that none of these covariates has a strong influence on Y, i.e. γ is "small." Adjusting for these $m-1$ covariates may remove the relatively minor bias of the $Z\gamma$ term, but it will also leave us with only one dimension for $\hat{W}$! Taken together, these observations tend to suggest that Z should generally be ignored.

However, there is an important exception. Consider the case that γ is sparse. Suppose that we leave Z out of the model. Since only a few genes exhibit the effects of Z, the factor analysis routine may not properly estimate Z and incorporate it into W. This may cause problems. The problems are somewhat different depending on whether or not X is correlated with Z. We discuss both cases in turn.

Suppose first that Z is strongly correlated with X. Suppose that gene j is one of the few genes such that $\gamma_j \neq 0$. If we do not explicitly adjust for Z, $\hat{\beta}_j$ may be strongly biased. If in truth $\beta_j = 0$, we may falsely conclude that $\beta_j \neq 0$. In other words, we may be led to false discoveries. On the other hand, suppose that in truth $\beta_j \neq 0$. We would likely correctly conclude that $\beta_j \neq 0$ (barring a very unfortunate cancellation of the $X\beta_j$ and $Z\gamma_j$ terms). However, $\hat{\beta}_j$ would still be biased, and perhaps even the sign would be wrong. This would be quite unfortunate. Since gene j is in fact differentially expressed with respect to X, gene j — and the actual value of β_j — is presumably of substantial scientific interest.

Suppose now that Z is not strongly correlated with X. We do not need to worry that omitting Z from the model will seriously bias β_j. However, the estimate $\hat{\sigma}_j^2$ of σ_j^2 may be inflated, as the $Z\gamma$ will still be present in the residuals. If in truth $\beta_j = 0$, an inflated $\hat{\sigma}_j^2$ is not of much concern. However, if in truth $\beta_j \neq 0$, an inflated $\hat{\sigma}_j^2$ will lead to a drop in power. We might fail to discover that gene j is differentially expressed with respect to X.

Nonetheless, our conclusion is not simply "if γ is sparse, include Z." A better motto might be "if γ is sparse, proceed with caution." If Z suffers from measurement error, or is simply a proxy for some other variable, it may still be best to leave Z out. Moreover, our discussion so far has been far from complete. Additional complications arise, for example, if Z is correlated with W. A complete discussion is beyond the scope of this thesis. On balance, if Z is important, if Z is well-measured, if q (the number of columns of Z) is small, and if γ is sparse, it is probably best to include Z. This is especially true if we believe that both β and γ are sparse, and that the few genes for which β is non-zero are also the few genes for which γ is non-zero. Still, this is only a rule of thumb. Repeating the analysis both with and without Z and inspecting the results seems reasonable.

We now consider the case that $p > 1$. There are three ways to handle this case. However, two of the three ways turn out to be equivalent, so effectively there are only two ways to handle the case $p > 1$. We now describe the three possibilities. The first strategy to handle the case $p > 1$ is to proceed with the RUV-4 algorithm exactly as described in Section 3.3.3.1; nothing in the procedure requires that $p = 1$. Of course, we do require that $p < m$. The second strategy to handle the case $p > 1$ is to run RUV-4 p times. Each time we redefine X to be just a single column of the original X, and ignore the other columns. The third strategy to handle the case $p > 1$ is to run RUV-4 p times. Each time we redefine X to be just a single column of the original X, and move the remaining columns of X to Z; we then explicitly adjust for this "Z."

It turns out that the first and the third strategies are equivalent; $\hat{\beta}$, $\hat{\sigma}^2$, p-values, etc. all are identical. We may therefore limit our attention to the second and third strategies. In both strategies, we run RUV-4 p times. Each time, we select a single column of X to be the factor of interest. Denote this column of X by $\tilde{X}$. The remaining columns of X play the role of observed covariates. Denote these columns by $\tilde{Z}$ (no relation to the $\tilde{Z}$ mentioned above). The difference between strategies two and three is whether or not we explicitly adjust for $\tilde{Z}$.

Which strategy is better? Once again, there is no clear answer. All of the considerations regarding whether or not to include Z continue to apply. However, there are additional complications as well. For example, we have assumed that $\beta_c = 0$. It follows that $\tilde{\gamma} = 0$. Thus, even if $\tilde{\gamma}$ is not sparse in general, it is "sparse" for the control genes. If we leave $\tilde{Z}$ out of the model, RUV-4 may not properly incorporate $\tilde{Z}$ into W. This is a strong argument for including $\tilde{Z}$. A second complication is that, in cases in which $p > 1$, it is common in practice that the columns of X are in some way "related." For example, several columns of X may simply be dummy variables representing different levels of a single factor. As a result, if β_{lj} is non-zero for one value of l, we might expect that $\beta_{l'j}$ is non-zero for several other values of l' as well — even when β as a whole is sparse. Thus, the genes for which $\tilde{\beta}$ is non-zero will also tend to be the genes for which $\tilde{\gamma}$ is non-zero. This too may be an argument for including $\tilde{Z}$. On balance, if p is small, it is probably best to include $\tilde{Z}$. Once again, however, the most prudent strategy may simply be to run the analysis both with and without $\tilde{Z}$, and carefully inspect the results.

3.3.6.3 Violation of the control gene assumption

We now consider what happens when certain assumptions are violated. In particular, we will first consider what happens when $\beta_c \neq 0$. Later we will consider what happens when $K \neq k$. Suppose $\beta_c \neq 0$. By (3.43) we see that this will result in additional conditional bias to $\hat{b}_{WX}$. The amount of this bias is $\mathbb{E}\left[\beta_c\left(\alpha_c+\xi_c\right)'\left[\left(\alpha_c+\xi_c\right)\left(\alpha_c+\xi_c\right)'\right]^{-1}\middle|\alpha_c\right]$. In general, depending on β_c and α_c, this bias may be arbitrarily bad. However, we observe that if β_c' is approximately orthogonal to $(\alpha_c+\xi_c)'$ then the bias will be approximately 0. With high probability ξ_c is approximately orthogonal to β_c. Therefore, we should not expect violations of the control gene assumption to be problematic unless $\beta_c\alpha_c'$ is appreciably non-zero.

More formally, suppose $\beta_c\alpha_c' = 0$. Then

$$\begin{aligned}
&\beta_c\left(\alpha_c+\xi_c\right)'\left[\left(\alpha_c+\xi_c\right)\left(\alpha_c+\xi_c\right)'\right]^{-1} \\
&\qquad = \beta_c P_{\beta_c'}\xi_c'\left[\left(\alpha_c+\xi_c P_{\beta_c'}+\xi_c R_{\beta_c'}\right)\left(\alpha_c'+P_{\beta_c'}\xi_c'+R_{\beta_c'}\xi_c'\right)\right]^{-1} &(3.66)\\
&\qquad = \beta_c P_{\beta_c'}\xi_c'\left[\xi_c P_{\beta_c'}P_{\beta_c'}\xi_c'+\left(\alpha_c+\xi_c R_{\beta_c'}\right)\left(\alpha_c'+R_{\beta_c'}\xi_c'\right)\right]^{-1}. &(3.67)
\end{aligned}$$

Now, the joint distribution of $\left(\xi_c P_{\beta_c'}, \xi_c R_{\beta_c'}\right)$ is equal to the joint distribution of $\left(-\xi_c P_{\beta_c'}, \xi_c R_{\beta_c'}\right)$, so

$$\begin{aligned}
&\mathbb{E}\left\{\beta_c P_{\beta_c'}\xi_c'\left[\xi_c P_{\beta_c'}P_{\beta_c'}\xi_c'+\left(\alpha_c+\xi_c R_{\beta_c'}\right)\left(\alpha_c'+R_{\beta_c'}\xi_c'\right)\right]^{-1}\middle|\alpha_c\right\} \\
&\quad = \mathbb{E}\left\{\beta_c\left(-P_{\beta_c'}\xi_c'\right)\left[\left(-\xi_c P_{\beta_c'}\right)\left(-P_{\beta_c'}\xi_c'\right)+\left(\alpha_c+\xi_c R_{\beta_c'}\right)\left(\alpha_c'+R_{\beta_c'}\xi_c'\right)\right]^{-1}\middle|\alpha_c\right\} &(3.68)\\
&\quad = -\mathbb{E}\left\{\beta_c P_{\beta_c'}\xi_c'\left[\xi_c P_{\beta_c'}P_{\beta_c'}\xi_c'+\left(\alpha_c+\xi_c R_{\beta_c'}\right)\left(\alpha_c'+R_{\beta_c'}\xi_c'\right)\right]^{-1}\middle|\alpha_c\right\} &(3.69)
\end{aligned}$$

and therefore $\mathbb{E}\left\{\beta_c\left(\alpha_c+\xi_c\right)'\left[\left(\alpha_c+\xi_c\right)\left(\alpha_c+\xi_c\right)'\right]^{-1}\middle|\alpha_c\right\} = 0$. No bias is introduced by a breakdown of the control gene assumption in which $\beta_c \neq 0$ but $\beta_c\alpha_c' = 0$.

3.3.6.4 Misspecification of k: Consequences for $\hat{\beta}$

We now consider what happens when $K \neq k$. There are two possible cases: $K < k$ and $K > k$. First consider the case that $K < k$. We may be tempted to regard $\hat{W}^{(K)}$ as simply a "reduced" version of $\hat{W}^{(k)}$ from which we have dropped $k - K$ columns. In other words, we might guess that $\Re\left(\hat{W}^{(K)}\right) \subset \Re\left(\hat{W}^{(k)}\right)$. Since omitting terms from a regression model generally leads to biased estimates, we might therefore reason that setting $K < k$ will lead to (additional) bias in our estimate of β. This is only partially correct. Setting $K < k$ does bias $\hat{\beta}$. However, it is not generally true that $\Re\left(\hat{W}^{(K)}\right) \subset \Re\left(\hat{W}^{(k)}\right)$. One trivial reason for this is that, depending on our choice of factor analysis routine, it may not even be the case that $\Re\left(\hat{W_0}^{(K)}\right) \subset \Re\left(\hat{W_0}^{(k)}\right)$. However, suppose that indeed $\Re\left(\hat{W_0}^{(K)}\right) \subset \Re\left(\hat{W_0}^{(k)}\right)$. It does not follow that $\Re\left(\hat{W}^{(K)}\right) \subset \Re\left(\hat{W}^{(k)}\right)$. Recall that we estimate b_{WX} by regressing b_{Y_cX} on $\hat{\alpha}_c$. However, the rows of $\hat{\alpha}_c$ are not in general orthogonal. Dropping some rows from $\hat{\alpha}_c$ therefore leads to a different estimate b_{WX}, even for the rows that remain. The conclusion then is that setting $K < k$ biases $\hat{\beta}$, but quantifying the bias is difficult. Nonetheless, by considering the limiting case that $K = 0$ (i.e. no adjustment), we may reason that the bias is potentially substantial. The simulations of Section 3.4 support this conclusion.

We now consider the case that $K > k$. We focus on the "worst case" in which $K = m-1$, but our results are clearly relevant whenever $k < K < m-1$. Note that when $K = m-1$ there is no role for the factor analysis in Step 2; $\hat{W}_0^{(m-1)}$ is simply equal to $X_\perp$. Let $W_1 \equiv (X|W_0)_\perp$. Assume without loss of generality that $\hat{W}_0^{(m-1)} = (W_0|W_1)$. Now define

$$\tilde{W} \equiv (W|W_1) \tag{3.70}$$

$$\tilde{\alpha} \equiv \begin{pmatrix} \alpha \\ 0_{m-k-1\times n} \end{pmatrix}. \tag{3.71}$$

Observe that $b_{\tilde{W}X} = (b_{WX}|0)$ and $\tilde{W}\tilde{\alpha} = W\alpha$. We may therefore regard $\tilde{W}\tilde{\alpha}$ as a reparameterization of $W\alpha$. Under this reparameterization, $\hat{W}_0^{(m-1)}$ is a perfect estimator of $\tilde{W}_0$. Therefore, expressions (3.38) and (3.39) apply exactly. To analyze the bias and variance under assumptions (3.44), (3.45), and the assumption that n_c is large, first define

$$\tilde{\Psi}^2 \equiv \begin{pmatrix} \Psi^2 & 0_{k\times m-k-1} \\ 0_{m-k-1\times k} & 0_{m-k-1\times m-k-1} \end{pmatrix}. \tag{3.72}$$

By (3.46)

$$\begin{aligned} \text{Bias}\left[\hat{\beta}_{j_{\bar{c}}}^{(m-1)}\middle|\tilde{\alpha}_{\star j_{\bar{c}}}\right] &\approx b_{\tilde{W}X}\left(\tilde{\Psi}^2/\sigma_0^2 + I\right)^{-1}\tilde{\alpha}_{\star j_{\bar{c}}} && (3.73)\\ &= b_{WX}\left(\Psi^2/\sigma_0^2 + I\right)^{-1}\alpha_{\star j_{\bar{c}}} && (3.74)\\ &\approx \text{Bias}\left[\hat{\beta}_{j_{\bar{c}}}^{(k)}\middle|\alpha_{\star j_{\bar{c}}}\right] && (3.75)\end{aligned}$$

and by (3.48)

$$\begin{aligned} \text{Var}\left[\hat{\beta}_{j_{\bar{c}}}^{(m-1)}\middle|\tilde{\alpha}_{\star j_{\bar{c}}}\right] &\approx \sigma_{j_{\bar{c}}}^2\left(1 + b_{\tilde{W}X}\tilde{\Psi}^4\left(\tilde{\Psi}^2 + \sigma_0^2 I\right)^{-2} b'_{\tilde{W}X}\right) && (3.76)\\ &= \sigma_{j_{\bar{c}}}^2\left(1 + b_{WX}\Psi^4\left(\Psi^2 + \sigma_0^2 I\right)^{-2} b'_{WX}\right) && (3.77)\\ &\approx \text{Var}\left[\hat{\beta}_{j_{\bar{c}}}^{(k)}\middle|\alpha_{\star j_{\bar{c}}}\right]. && (3.78)\end{aligned}$$

Thus, up to approximation, the bias and variance of $\hat{\beta}_{j\bar{c}}^{(m-1)}$ and $\hat{\beta}_{j\bar{c}}^{(k)}$ are the same; nothing is lost by over-estimating k. It is useful to recall at this point what approximations are being made that justify this conclusion. The approximations are based on the assumption that n_c is large. The approximations may be made arbitrarily good by a sufficiently large n_c. If n_c is not sufficiently large, we may in fact pay a substantial price by overestimating k. In particular, $\text{Var}\left[\hat{b}_{WX}\right]$ may no longer be negligible, and the $\sigma_{j\bar{c}}^2 \text{tr}\left(\text{Var}\left[\hat{b}_{WX}\right]\right)$ and $\alpha_{\star j\bar{c}}' \text{Var}\left[\hat{b}_{WX}\right] \alpha_{\star j\bar{c}}$ terms of (3.39) may become important. We return to this point in Section 3.3.9.1.

3.3.6.5 Misspecification of k: Consequences for $\hat{\sigma}^2$

In practice, estimating σ^2 is much more complicated than Section 3.3.6.1 suggests. The difficulty arises in the estimation of k. The performance of $\hat{\sigma}_j^2$ depends critically on a proper estimation of k. Unlike $\hat{\beta}_j$, $\hat{\sigma}_j^2$ performs poorly both when k has been underestimated and when k has been overestimated. Therefore we cannot simply dodge the issue by systematically over- or underestimating k as we can with $\hat{\beta}$.

We will not analyze the statistical properties of $\hat{\sigma}_j^2$ in any detail. Roughly speaking, however, we may summarize the main issues as follows: Firstly, overestimating k increases the variance of $\hat{\sigma}^2$. This is simply because σ^2 must be estimated using fewer degrees of freedom. Moreover, overestimating k biases $\hat{\sigma}^2$ downwards on average. This is because the factor analysis routine in Step 2 will presumably allocate the extra $K - k$ dimensions of $\hat{W}_0$ to the (random) dimensions in which ϵ shows the greatest variation. The residuals will therefore be too small. Finally, underestimating k biases $\hat{\sigma}^2$ upwards on average, possibly substantially. This is because some terms of $W\alpha$ are not effectively adjusted away. Some unwanted variation remains in the residuals, and this inflates $\hat{\sigma}^2$.

To see that underestimating k biases $\hat{\sigma}^2$ upwards on average and overestimating k biases $\hat{\sigma}^2$ downwards on average it is helpful to consider the specific case in which we use the singular value decomposition (SVD) as our factor analysis method. Let UDV' be the singular value decomposition of R_XY, i.e. $R_XY = UDV'$ where U and V are orthonormal matrices and D is a diagonal matrix with decreasing diagonal entries denoted by d_i. $\hat{W}_0^{(K)}$ is defined to be the first K columns of U. Let $\bar{\sigma}^2 \equiv \sum_{j=1}^n \sigma_j^2$ denote the average gene variance. Let $\dot{\sigma}^2 \equiv \sum_{j=1}^n \hat{\sigma}_j^2$ denote the estimated average variance. The estimated average variance as a function of K is therefore

$$
\begin{aligned}
\left(\dot{\sigma}^2\right)^{(K)} &= \frac{1}{n}\sum_{j=1}^{n}\left(\hat{\sigma}_j^2\right)^{(K)} && (3.79)\\
&= \frac{1}{n}\sum_{j=1}^{n}\frac{1}{m-K-1}\left(R_{\left(X|\hat{W}_0^{(K)}\right)}Y_{\star j}\right)'\left(R_{\left(X|\hat{W}_0^{(K)}\right)}Y_{\star j}\right) && (3.80)\\
&= \frac{1}{n(m-K-1)}\sum_{j=1}^{n}\left(R_{\hat{W}_0^{(K)}}R_XY_{\star j}\right)'\left(R_{\hat{W}_0^{(K)}}R_XY_{\star j}\right) && (3.81)\\
&= \frac{1}{n(m-K-1)}\sum_{j=1}^{n}V_{j\star}DU'R_{\hat{W}_0^{(K)}}UDV_{j\star}' && (3.82)\\
&= \frac{1}{n(m-K-1)}\sum_{j=1}^{n}\sum_{i=K+1}^{m}V_{ji}^2d_i^2 && (3.83)\\
&= \frac{1}{n(m-K-1)}\sum_{i=K+1}^{m}d_i^2. && (3.84)
\end{aligned}
$$

Thus $\left(\dot{\sigma}^2\right)^{(K)}$ is decreasing in K, since the d_i are decreasing in i. Since $\dot{\sigma}^2$ is an unbiased estimator of $\bar{\sigma}^2$ when $\hat{W}_0 = W_0$, it follows that if $\hat{W}_0^{(k)} = W_0$, $\left(\dot{\sigma}^2\right)^{(K)}$ will be biased upwards when $K < k$ and biased downwards when $K > k$. Of course, in practice, it is not true that $\hat{W}_0^{(k)} = W_0$ but rather that $\hat{W}_0^{(k)} \approx W_0$, so these results hold only approximately. A full discussion is beyond the scope of this thesis. Nonetheless, we do feel that the above argument is useful for the intuition it provides. Moreover, in both simulation experiments and the analysis of real data, we have found that the conclusions tend to hold — when K is too large, $\hat{\sigma}^2$ is too small; when K is too small, $\hat{\sigma}^2$ is too large.

3.3.6.6 Estimating k

As Sections 3.3.6.4 and 3.3.6.4 suggest, a good choice of K is critical to the performance of RUV-4. Unfortunately, selecting an appropriate K is difficult. It is not even clear that the optimal K for $\hat{\beta}$ is the same as the optimal K for $\hat{\sigma}^2$. (Indeed, we have had success estimating $\hat{\beta}$ and $\hat{\sigma}^2$ using different values of K, but we do not discuss this approach in this thesis.)

As for RUV-2, we feel the best way to select K for RUV-4 is to run the analysis for several values of K and choose the "best" one based on p-value histograms, RLE plots, the rankings of positive controls, and other quality assessments. We are unaware of any good algorithmic way to estimate k, and we feel there is an important role for human judgment.

Nonetheless, this hands-on approach is not always feasible or desirable. For example, in Section 3.4 we present the results of simulation experiments in which RUV-4 was run thousands of times. Estimating k "by hand" thousands of times is not feasible. Therefore we will now present a method to estimate k that we have found to perform moderately well in many situations.

Our method for estimating k relies on control genes. The key insight is that if RUV-4 works as intended

$$\mathbb{E}\left[\hat{\beta}_c \middle| \alpha\right] \approx 0 \tag{3.85}$$

and thus

$$\hat{\beta}_{j_c}^2 \quad \dot{\sim} \quad \text{Var}\left[\hat{\beta}_{j_c} \middle| \alpha\right] \chi_1^2. \tag{3.86}$$

The symbol $\dot{\sim}$ means "is approximately distributed as." Unfortunately, the quantity Var $\left[\hat{\beta}_{j_c} \middle| \alpha\right]$ is difficult to analyze. We therefore begin by considering a gene j_0 that is not a designated control gene but nonetheless such that $\beta_{j_0} = 0$.

Assume (3.44), (3.45), and that n_c is large. Assume that RUV-4 works as intended so that $\mathbb{E}\left[\hat{\beta}_{j_0} \middle| \alpha\right] \approx 0$. By (3.62)

$$\text{Var}\left[\hat{\beta}_{j_0} \middle| \alpha\right] \quad \approx \quad \sigma_{j_0}^2 \left(1 + \hat{b}_{WX}\hat{b}_{WX}'\right) \tag{3.87}$$

and therefore

$$s_{j_0}^2 \equiv \frac{\hat{\beta}_{j_0}^2}{1 + \hat{b}_{WX}\hat{b}_{WX}'} \quad \dot{\sim} \quad \sigma_{j_0}^2 \chi_1^2. \tag{3.88}$$

As in Section (3.3.6.4), suppose that $\hat{W}_0^{(m-1)} = (W_0|W_1)$. Assume that W_1, while otherwise arbitrary, is fixed. Then for all i such that $k < i < m$ and all j such that $1 \leq j \leq n$,

$$s_{ij}^2 \equiv \left(\hat{\alpha}_{ij}^{(m-1)}\right)^2 \quad \sim \quad \sigma_j^2 \chi_1^2. \tag{3.89}$$

Now, for $1 \leq i < m$ consider the quantity

$$r_i^{(0)} \equiv \text{median}_{j_0} \sqrt{s_{ij_0}^2 / s_{j_0}^2}. \tag{3.90}$$

This quantity gives some measure of the scale of the i^{th} row of $\hat{\alpha}^{(m-1)}$ relative to the scale of ϵ. In light of (3.88) and (3.89), we would expect that $r_i^{(0)} \approx 1$ for $k < i < m$. We can exploit this fact to estimate k. For example, we could estimate k by $\#\left\{r_i^{(0)} > C\right\}$, where $C > 1$ is some cutoff value.

In practice, we will want to designate every gene known to be uninfluenced by X as a control gene. Thus we should not expect to have available any genes j_0 as described above. Instead, we will just use the control genes. However, the statistical properties of the control genes are different than the statistical properties of the other genes. In particular, since we estimate b_{WX} by regressing b_{Y_cX} on $\hat{\alpha}_c$, it is not true that $\hat{b}_{WX}$ is independent of ζ_{j_c} and

$\xi_{\star j_c}$. The practical consequence of this is that $\hat{b}_{WX}$ is overfitted to the control genes, and as a result the variance of $\hat{\beta}_{j_c}$ tends to be somewhat less than $\sigma^2_{j_c}\left(1+\hat{b}_{WX}\hat{b}'_{WX}\right)$. Note that $\hat{\beta}_c$ may be regarded as the residuals of a regression of b_{Y_cX} on $\hat{\alpha}_c$. In a "standard" regression, the residuals are "too small" by a factor of $\sqrt{(N-P)/N}$, where N is the number of observations and P is the number of regressors. This inspires us to define:

$$r_i \equiv \text{median}_{j_c}\sqrt{\left(\frac{n_c}{n_c-K_i}\right)\frac{s^2_{ij_c}}{s^2_{j_c}}} \tag{3.91}$$

where K_i denotes the value of K we use in the calculation of $s^2_{j_c}$ (in the analyses of Sections 3.4 and 3.5 we set $K_i = i$, but other strategies are possible). This is somewhat ad hoc. The regression of b_{Y_cX} on $\hat{\alpha}_c$ is not a "standard" regression. In particular, we have ignored the fact that different genes have different variances. Nonetheless, we now define our estimate of k as

$$\hat{k}(C) \equiv \#\left\{r_i > C\right\}. \tag{3.92}$$

We must choose a value for C. In choosing a value for C we consider the fact that it is not actually true that $\hat{W}_0^{(m-1)} = (W_0|W_1)$ where W_1 is some arbitrary but fixed matrix. In particular, we may not take W_1 to be fixed. Its parameterization is random, and determined by the factor analysis routine in Step 2 of RUV-4. This did not matter in Section 3.3.6.4 because the parameterization of $\hat{W}_0$ was irrelevant. Here, however, the parameterization of $\hat{W}_0$ does matter; r_i is defined in terms of the individual columns of $\hat{W}_0$. Assume that we use the SVD as the method of factor analysis in Step 2 of RUV-4. In this case, we might expect that $\hat{W}_0^{(k)} \approx W_0$. Again, this is only approximately correct, and a full discussion is beyond the scope of this thesis. For sake of argument, however, simply assume that $\hat{W}_0^{(k)} = W_0$. We may then write $\hat{W}_0^{(m-1)} = (W_0|\tilde{W}_1)$, implicitly defining $\tilde{W}_1$. The problem is that $\tilde{W}_1$ is not fixed; it has been rotated so that the most variation is captured by first column. As a result, we should not expect that $r_{k+1} \approx 1$ but rather that $r_{k+1} > 1$. To fix this problem, we set $C > 1$. We choose to set $C = \mathbb{E}(\eta)$ where η is the principal singular value of an $m \times n$ matrix of independent $\mathrm{N}(0, \frac{1}{n})$ random variables. In our analyses we simply approximate C by simulation. A far more computationally efficient approach would be to approximate C using the fact that the distribution of η^2 is approximately Tracy-Widom. This approximation can be very good. See, for example, Ma (2012).

3.3.7 The Inverse Method

In Section 3.3.6.4 we saw that overestimating k does not seriously degrade the performance of $\hat{\beta}$ as long as a sufficient number of control genes are available. In Section 3.3.6.5 we saw that overestimating (or underestimating) k does seriously degrade the performance of $\hat{\sigma}^2$. We are left with a dilemma. A good estimate of β is readily available — just set K as high as it can go, to $m-1$ — but we are unable to estimate σ^2. To solve this dilemma we will introduce a novel method for estimating σ^2, which we name the "inverse method." We introduce the abstract method in Section 3.3.7.1. In Section 3.3.7.2 we apply the inverse method to RUV-4. In Sections 3.3.7.3 and 3.3.7.5 we reformulate our estimators. These reformulations are of both theoretical and practical interest. We provide discussions in Sections 3.3.7.4 and 3.3.7.6.

3.3.7.1 The Inverse Method

We now present the inverse method in the abstract, followed by a simple example. The method is so simple as to seem trivial. However, as we will eventually see, properly applied it can be quite powerful. Note that all of the notation used in this section is specific to this section only.

Let $\hat{\theta}_U$ be a family of estimators indexed by U. What $\hat{\theta}_U$ estimates need not be relevant. Assume that there exist some (possibly random) values of U, denoted $U_1, U_2, ..., U_i, ...$, such that

$$\mathbb{E}\left[\hat{\theta}_{U_i}\middle|U_i\right] = 0.$$

Assume also that

$$\mathrm{Var}\left[\hat{\theta}_{U_i}\middle|U_i\right] = f_{U_i}(\sigma^2)$$

where σ^2 is some unknown parameter. If the f_{U_i} are linear functions of σ^2, then

$$\mathbb{E}\left[\hat{\theta}^2_{U_i}\middle|U_i\right] = f_{U_i}(\sigma^2) \tag{3.93}$$

$$f^{-1}_{U_i}\left(\mathbb{E}\left[\hat{\theta}^2_{U_i}\middle|U_i\right]\right) = \sigma^2 \tag{3.94}$$

$$\mathbb{E}\left[f^{-1}_{U_i}\left(\hat{\theta}^2_{U_i}\right)\middle|U_i\right] = \sigma^2 \tag{3.95}$$

where $f^{-1}_{U_i}$ is the functional inverse of f_{U_i}.

If such U_i are available, if the functions f_{U_i} are known, and if we have the data necessary to compute $\hat{\theta}_{U_i}$, we are able to compute $f^{-1}_{U_i}\left(\hat{\theta}^2_{U_i}\right)$. We may regard each of these $f^{-1}_{U_i}\left(\hat{\theta}^2_{U_i}\right)$ as estimates of σ^2. We may then combine the $f^{-1}_{U_i}\left(\hat{\theta}^2_{U_i}\right)$ in some way, e.g. take their average, to produce a final "inverse" estimate of σ^2.

For concreteness, we will now present a very simple (but contrived) example. Suppose we have a standard linear regression model

$$Y_{n\times 1} = X_{n\times p}\beta_{p\times 1} + \epsilon_{n\times 1} \tag{3.96}$$

where Y is observed, X is fixed and observed, β is fixed and unknown, $p < n$, and the elements of ϵ are IID with expectation 0 and variance σ^2. To estimate σ^2 using the inverse method, first note that we may model Y as

$$Y = X\beta + U\theta + \epsilon \tag{3.97}$$

where U may be any $n \times 1$ matrix and $\theta = 0$. If U is not a linear combination of X, then we may estimate θ by OLS and let

$$\hat{\theta}_U \equiv \left(U' R_X U\right)^{-1} U' R_X Y.$$

Let $U_1, U_2, ..., U_i, ...$ be random matrices whose entries are IID standard normal and independent of ϵ. Then with probability 1, U_i is not a linear combination of the columns of X and $\hat{\theta}_{U_i}$ exists. Moreover,

$$\begin{aligned} \mathbb{E}\left[\hat{\theta}_{U_i} \middle| U_i\right] &= 0 && (3.98) \\ \mathrm{Var}\left[\hat{\theta}_{U_i} \middle| U_i\right] &= \sigma^2 \left(U_i' R_X U_i\right)^{-1} && (3.99) \end{aligned}$$

and therefore

$$\mathbb{E}\left[\frac{\hat{\theta}_{U_i}^2}{\left(U_i' R_X U_i\right)^{-1}} \middle| U_i\right] = \sigma^2. \qquad (3.100)$$

Thus, we might imagine generating many U_i, and for each U_i calculating $\hat{\theta}_{U_i}^2 / \left(U_i' R_X U_i\right)^{-1}$, and then averaging these values to produce an estimate of σ^2. Or, by taking the limit of this process, we may define the inverse estimator of σ^2 to be

$$\hat{\sigma}_{\text{inv}}^2 \equiv \mathbb{E}_{U_1}\left[\frac{\hat{\theta}_{U_1}^2}{\left(U_1' R_X U_1\right)^{-1}}\right]. \qquad (3.101)$$

Note that, defined this way, it is actually the case that

$$\hat{\sigma}_{\text{inv}}^2 = \frac{1}{n-p}(Y - X\hat{\beta})'(Y - X\hat{\beta}) \qquad (3.102)$$

and $\hat{\sigma}_{\text{inv}}^2$ is thus equivalent to the standard OLS estimate of σ^2 (proof omitted).

3.3.7.2 The Inverse Method for RUV-4

We now apply the inverse method to RUV-4. Let $X^\star$ be a random "factor of interest" chosen uniformly at random from the unit $(m-1)$-sphere. Since $X^\star$ is random, it should not be "truly associated" with the expression levels of any of the genes. We may model Y as

$$Y = X\beta + W\alpha + X^\star \beta^\star + \epsilon \qquad (3.103)$$

where $\beta^\star = 0$. We will estimate $\beta^\star$ by RUV-4. In this context, X is now an unwanted factor and plays the role of Z. In Section 3.3.6.2 we discussed two ways to handle a known covariate: ignore it and let RUV-4 incorporate it into W, or explicitly adjust for it. In this case, it is almost certainly better to explicitly adjust for X. By assumption, the control genes are uninfluenced by X, and therefore RUV-4 will not properly incorporate X into W. $\hat{\beta}^\star$ may be biased, violating the key assumption of the inverse method (see (3.110), below).

Therefore, we explicitly adjust for X. Define U to be the matrix whose columns are the first $m-1$ eigenvectors of $R_X Y_c Y_c' R_X$. Note that U is a specific parameterization of $X_\perp$. This parameterization will prove convenient in Section 3.3.7.5. Now define:

$$\begin{aligned} \mathsf{Y} &\equiv \mathsf{U}'Y && (3.104)\\ \mathsf{X} &\equiv \mathsf{U}'X^\star && (3.105)\\ \mathsf{W} &\equiv \mathsf{U}'W && (3.106)\\ \varepsilon &\equiv \mathsf{U}'\epsilon && (3.107)\\ \mathsf{m} &\equiv m-1. && (3.108)\end{aligned}$$

Note that:

$$\mathsf{Y}_{\mathsf{m}\times n} = \mathsf{X}_{\mathsf{m}\times 1}\beta^\star_{n\times 1} + \mathsf{W}_{\mathsf{m}\times k}\alpha_{k\times n} + \varepsilon_{\mathsf{m}\times n}. \qquad (3.109)$$

Let $\hat{\beta}^\star$ denote the RUV-4 estimator of $\beta^\star$ for some fixed K. Typically we set $K = \mathsf{m}-1$.

We now state the key assumption of the inverse method applied to RUV-4. We assume that with high probability

$$\mathbb{E}\left[\hat{\beta}^\star \middle| \mathsf{X}\right] \approx 0. \qquad (3.110)$$

The real assumption here is that RUV-4 “works” — that the unwanted variation is effectively adjusted for, and that the regression coefficients corresponding to a random “factor of interest” that is not truly associated with the expression levels of any genes will in fact be estimated to be about 0.

If n_c is large, we have by (3.62) that

$$\mathrm{Var}\left[\hat{\beta}^\star_{j\bar{c}} \middle| \mathsf{X}\right] \approx \sigma^2_{j\bar{c}}\left(1 + \hat{b}_{\mathsf{WX}}\hat{b}'_{\mathsf{WX}}\right). \qquad (3.111)$$

Following the example of the previous section, we define

$$\left(\hat{\sigma}^2_{j\bar{c}}\right)^{(K,\mathrm{inv})} \equiv \mathbb{E}_{\mathsf{X}}\left[\frac{\left(\hat{\beta}^\star_{j\bar{c}}\right)^2}{1 + \hat{b}_{\mathsf{WX}}\hat{b}'_{\mathsf{WX}}}\right]. \qquad (3.112)$$

In Section 3.3.7.5 we will develop an exact analytic expression for $\left(\hat{\sigma}^2_{j\bar{c}}\right)^{(K,\mathrm{inv})}$ in the special case that $K = \mathsf{m}-1$. However, it is also very straight-forward to approximate $\left(\hat{\sigma}^2_{j\bar{c}}\right)^{(K,\mathrm{inv})}$ more generally by repeatedly generating random $\mathsf{X} = \mathsf{U}'X^\star$, fitting with RUV-4, calculating $\left(\hat{\beta}^\star_{j\bar{c}}\right)^2 / \left(1 + \hat{b}_{\mathsf{WX}}\hat{b}'_{\mathsf{WX}}\right)$, and averaging the results.

We conclude this section by reiterating the importance of explicitly adjusting for X when calculating $\hat{\beta}^\star$. Suppose we do not explicitly adjust for X and RUV-4 does not properly incorporate X into W; X is thus simply unadjusted for. As a result, $\hat{\beta}^\star_{j\bar{c}}$ will be biased.

$\left(\hat{\beta}^{\star}_{j\bar{c}}\right)^2$ will be too large and $\left(\hat{\sigma}^2_{j\bar{c}}\right)^{(K,\text{inv})}$ will be inflated. To see this more clearly, we present an analogy. In the simple example of Section 3.3.7.1 we estimate θ by $\left(U'R_XU\right)^{-1}U'R_XY$; we "explicitly adjust for X." The resulting inverse-method estimate $\hat{\sigma}^2$ is $[1/(n-p)](Y - X\hat{\beta})'(Y - X\hat{\beta})$, and $\mathbb{E}[\hat{\sigma}^2] = \sigma^2$. Now suppose that we had not explicitly adjusted for X, but instead had estimated θ by $\left(U'U\right)^{-1}U'Y$. The resulting inverse-method estimate $\hat{\sigma}^2$ would be $(1/n)Y'Y$. The expected value of $\hat{\sigma}^2$ would be

$$\beta'\left(\frac{X'X}{n}\right)\beta + \sigma^2.$$

Thus, by not explicitly adjusting for X, we would inflate our estimate of σ^2 by a factor of roughly

$$1 + \frac{\beta'X'X\beta}{n\sigma^2}.$$

3.3.7.3 A Closed-Form Solution for $\hat{\beta}^{(\text{RUV}-\text{inv})}$

Define $\hat{\beta}^{(\text{RUV}-\text{inv})}$ to be the RUV-4 estimator when $K = m - 1$. Note that the notation is slightly misleading, since strictly speaking the inverse method is a method for estimating variances, and can be applied to a wide class of estimators, including RUV-4 for any K. However, the RUV-4 estimator with $K = m-1$ is the preferred estimator to which we apply the inverse method — and indeed the estimator which most requires and initially inspired the method — and hence we denote it by $\hat{\beta}^{(\text{RUV}-\text{inv})}$. The goal of this section is to produce a closed-form expression for $\hat{\beta}^{(\text{RUV}-\text{inv})}$. We begin by reformulating the four-step estimator:

$$\begin{aligned}
\hat{\beta}^{(\text{RUV}-4)} &= X'\left(Y - \hat{W}\hat{\alpha}\right) \\
&= X'\left[Y - \left(\hat{W}_0 + Xb_{Y_cX}\hat{\alpha}_c'\left(\hat{\alpha}_c\hat{\alpha}_c'\right)^{-1}\right)\hat{\alpha}\right] \\
&= X'Y - b_{Y_cX}\hat{\alpha}_c'\left(\hat{\alpha}_c\hat{\alpha}_c'\right)^{-1}\hat{\alpha} \\
&= X'Y - X'Y_cY_c'\hat{W}_0\left(\hat{W}_0'Y_cY_c'\hat{W}_0\right)^{-1}\hat{W}_0'Y \\
&= X'\left[I - Y_cY_c'\hat{W}_0\left(\hat{W}_0'Y_cY_c'\hat{W}_0\right)^{-1}\hat{W}_0'\right]Y \\
&= X'\left(Y_cY_c'\right)^{\frac{1}{2}}\left[I - \left(Y_cY_c'\right)^{\frac{1}{2}}\hat{W}_0\left(\hat{W}_0'Y_cY_c'\hat{W}_0\right)^{-1}\hat{W}_0'\left(Y_cY_c'\right)^{\frac{1}{2}}\right]\left(Y_cY_c'\right)^{-\frac{1}{2}}Y \\
&= X'\left(Y_cY_c'\right)^{\frac{1}{2}}R_{\left(Y_cY_c'\right)^{\frac{1}{2}}\hat{W}_0}\left(Y_cY_c'\right)^{-\frac{1}{2}}Y \\
&= X'\left(Y_cY_c'\right)^{\frac{1}{2}}P_{\left(Y_cY_c'\right)^{-\frac{1}{2}}\hat{W}_{0\perp}}\left(Y_cY_c'\right)^{-\frac{1}{2}}Y \\
&= X'\hat{W}_{0\perp}\left(\hat{W}_{0\perp}'\left(Y_cY_c'\right)^{-1}\hat{W}_{0\perp}\right)^{-1}\hat{W}_{0\perp}'\left(Y_cY_c'\right)^{-1}Y. \qquad (3.113)
\end{aligned}$$

If we now set $K = m - 1$ so that $\hat{W}_0 = X_\perp$ and thus $\hat{W}_{0\perp} = X$, then (3.113) becomes

$$\hat{\beta}^{(\mathrm{RUV-inv})} = \left(X' \left(Y_c Y_c' \right)^{-1} X \right)^{-1} X' \left(Y_c Y_c' \right)^{-1} Y. \tag{3.114}$$

Again note that although we give the inverse estimator a special name and are able to define it using a relatively simple, closed form expression, it is still exactly equivalent to the RUV-4 estimator with $K = m - 1$.

3.3.7.4 Discussion: $\hat{\beta}^{(\mathrm{RUV-inv})}$

$\hat{\beta}^{(\mathrm{RUV-inv})}$ is of considerable theoretical interest. It has the form of a GLS estimator. Consider the case in which α is random. Let $\Sigma \equiv \mathrm{Cov}\left[W\alpha_{\star j} \right]$ and let $\delta \equiv W\alpha + \epsilon$. We then have

$$Y_{\star j} = X\beta_{1j} + \delta_{\star j} \tag{3.115}$$

with

$$\Sigma_j \equiv \mathrm{Cov}\left[\delta_{\star j} \right] = \Sigma + \sigma_j^2 I. \tag{3.116}$$

We may therefore wish to regard $\hat{\beta}^{(\mathrm{RUV-inv})}$ as a GLS estimator of β in which Σ_j has been (implicitly) approximated by $\frac{1}{n_c}\left(Y_c Y_c' \right)$. Now,

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{n_c} \left(Y_c Y_c' \right) \right] &= \frac{1}{n_c} \sum_{j_c} \mathbb{E}\left[Y_{\star j_c} Y_{\star j_c}' \right] && (3.117) \\
&= \frac{1}{n_c} \sum_{j_c} \mathbb{E}\left[\delta_{\star j_c} \delta_{\star j_c}' \right] && (3.118) \\
&= \frac{1}{n_c} \sum_{j_c} \Sigma + \sigma_{j_c}^2 I && (3.119) \\
&= \Sigma + \bar{\sigma}_c^2 I && (3.120)
\end{aligned}$$

where

$$\bar{\sigma}_c^2 \equiv \frac{1}{n_c} \sum_{j_c} \sigma_{j_c}^2. \tag{3.121}$$

Therefore

$$\Sigma_j = \mathbb{E}\left[\frac{1}{n_c} \left(Y_c Y_c' \right) \right] + \left(\sigma_j^2 - \bar{\sigma}_c^2 \right) I \tag{3.122}$$

and $\frac{1}{n_c}\left(Y_c Y_c' \right)$ is a biased estimate of Σ_j, with bias $\left(\bar{\sigma}_c^2 - \sigma_j^2 \right) I$. To some extent, this complicates the interpretation of $\hat{\beta}^{(\mathrm{RUV-inv})}$ as a GLS estimator. $\hat{\beta}^{(\mathrm{RUV-inv})}$ is not the "best" estimator for any specific gene. However, if the values of σ_j^2 do not vary wildly from gene to gene and $\bar{\sigma}_c^2 \approx \bar{\sigma}^2$ then we may wish to regard $\hat{\beta}^{(\mathrm{RUV-inv})}$ as a GLS-like estimator that is "reasonably good on average."

In light of (3.122) we may be tempted to refine our estimator of β on a gene-by-gene basis. For example, if initial estimates $\hat{\sigma}_j^2$ of σ_j^2 and $\dot{\sigma}_c^2$ of $\bar{\sigma}_c^2$ are available, we may be tempted to estimate β_j by

$$\left\{X'\left[Y_cY_c' + n_c\left(\hat{\sigma}_j^2 - \dot{\sigma}_c^2\right)I\right]^{-1}X\right\}^{-1}X'\left[Y_cY_c' + n_c\left(\hat{\sigma}_j^2 - \dot{\sigma}_c^2\right)I\right]^{-1}Y_{\star j}.$$

This is not necessarily a good idea, and may very well prove disastrous. In Section B.1.3 of the appendix we discuss this issue a bit further. To summarize Section B.1.3, it may be possible to make use of initial estimates of σ_j^2 and $\bar{\sigma}_c^2$ to refine our estimator of β on a gene-by-gene basis, but to do so requires considerable care, and is beyond the scope of this thesis. In any case, any gain in performance is likely to be minor, and the performance of the "unrefined" estimator $\hat{\beta}^{(\mathrm{RUV-inv})}$ is generally adequate.

Finally, we note that the GLS interpretation is not unique to $\hat{\beta}^{(\mathrm{RUV-inv})}$. From (3.113) we see that, after an appropriate transformation of the data and some additional algebra (omitted), the RUV-4 estimator may be viewed as a GLS-like estimator for any K, not just $K = m - 1$. More specifically, if we let $\tilde{Y} \equiv P_{\left(X\middle|\hat{W}_0\right)}Y$ it can be shown that

$$\hat{\beta}^{(\mathrm{RUV-4})} = \left(X'\left(\tilde{Y}_c\tilde{Y}_c'\right)^{+}X\right)^{-1}X'\left(\tilde{Y}_c\tilde{Y}_c'\right)^{+}\tilde{Y}. \tag{3.123}$$

Note that we must use the generalized inverse $\left(\tilde{Y}_c\tilde{Y}_c'\right)^{+}$ because $\tilde{Y}_c\tilde{Y}_c'$ is only rank $K+1$. Alternatively, we may redefine $\tilde{Y}$ as $\left(X\middle|\hat{W}_0\right)'Y$ and let $\tilde{X} \equiv \left(X\middle|\hat{W}_0\right)'X$. The RUV-4 estimator can then be expressed as

$$\hat{\beta}^{(\mathrm{RUV-4})} = \left(\tilde{X}'\left(\tilde{Y}_c\tilde{Y}_c'\right)^{-1}\tilde{X}\right)^{-1}\tilde{X}'\left(\tilde{Y}_c\tilde{Y}_c'\right)^{-1}\tilde{Y}. \tag{3.124}$$

In either case, we may informally describe the approach as throwing away the dimensions spanned by $\left(X\middle|\hat{W}_0\right)_{\perp}$ and fitting by GLS in the remaining $K+1$ dimensions. If the dimensions spanned by $\left(X\middle|\hat{W}_0\right)_{\perp}$ only contain noise, removing them should reduce the variance of $\hat{\beta}$. Thus we may view RUV-4 as fitting by GLS with an additional noise-reducing dimensionality reduction step. It is interesting to note just how different this view of RUV-4 is from the one initially presented in Section (3.3.3).

3.3.7.5 An Analytic Solution for $\left(\hat{\sigma}_j^2\right)^{(\mathrm{RUV-inv})}$

Define

$$\left(\hat{\sigma}_{j\bar{c}}^2\right)^{(\mathrm{RUV-inv})} \equiv \left(\hat{\sigma}_{j\bar{c}}^2\right)^{(\mathsf{m}-1,\mathrm{inv})}.$$

When it is clear from context, we will drop the superscript and the $\bar{c}$ subscript on the j and refer to this quantity simply as $\hat{\sigma}_j^2$. The goal of this section is to produce an analytic expression for $\hat{\sigma}_j^2$.

Combining (3.112) and (3.114) we have

$$\hat{\sigma}_j^2 \equiv \mathbb{E}_{\mathsf{X}}\left[\frac{\left[\left(\mathsf{X}'\left(\mathsf{Y}_c\mathsf{Y}_c'\right)^{-1}\mathsf{X}\right)^{-1}\mathsf{X}'\left(\mathsf{Y}_c\mathsf{Y}_c'\right)^{-1}\mathsf{Y}_{\star j}\right]^2}{1+\hat{b}_{\mathsf{WX}}\hat{b}_{\mathsf{WX}}'}\right] \tag{3.125}$$

Let D denote the diagonal matrix whose diagonal entries are the first $m-1$ eigenvalues of $R_X Y_c Y_c' R_X$ and note that $\mathsf{Y}_c\mathsf{Y}_c' = \mathsf{D}$. Further note that

$$\operatorname{Var}\left[\hat{\beta}^{\star}_{j\bar{c}}\middle|\mathsf{X},\mathsf{Y}_c\right] = \left(1+\hat{b}_{\mathsf{WX}}\hat{b}_{\mathsf{WX}}'\right)\sigma^2_{j\bar{c}} \tag{3.126}$$

and

$$\operatorname{Var}\left[\hat{\beta}^{\star}_{j\bar{c}}\middle|\mathsf{X},\mathsf{Y}_c\right] = \left|\left|\left(\mathsf{X}'\left(\mathsf{Y}_c\mathsf{Y}_c'\right)^{-1}\mathsf{X}\right)^{-1}\mathsf{X}'\left(\mathsf{Y}_c\mathsf{Y}_c'\right)^{-1}\right|\right|^2\sigma^2_{j\bar{c}} \tag{3.127}$$

and thus

$$1+\hat{b}_{\mathsf{WX}}\hat{b}_{\mathsf{WX}}' = \left|\left|\left(\mathsf{X}'\left(\mathsf{Y}_c\mathsf{Y}_c'\right)^{-1}\mathsf{X}\right)^{-1}\mathsf{X}'\left(\mathsf{Y}_c\mathsf{Y}_c'\right)^{-1}\right|\right|^2. \tag{3.128}$$

We may now simplify (3.125) as

$$\begin{aligned}
\hat{\sigma}_j^2 &= \mathbb{E}_{\mathsf{X}}\left[\frac{\left[\left(\mathsf{X}'\mathsf{D}^{-1}\mathsf{X}\right)^{-1}\mathsf{X}'\mathsf{D}^{-1}\mathsf{Y}\right]^2}{\left|\left|\left(\mathsf{X}'\mathsf{D}^{-1}\mathsf{X}\right)^{-1}\mathsf{X}'\mathsf{D}^{-1}\right|\right|^2}\right] && (3.129)\\
&= \mathbb{E}_{\mathsf{X}}\left[\frac{\mathsf{Y}_{\star j}'\mathsf{D}^{-1}\mathsf{X}\left(\mathsf{X}'\mathsf{D}^{-1}\mathsf{X}\right)^{-2}\mathsf{X}'\mathsf{D}^{-1}\mathsf{Y}_{\star j}}{\left(\mathsf{X}'\mathsf{D}^{-1}\mathsf{X}\right)^{-1}\mathsf{X}'\mathsf{D}^{-2}\mathsf{X}\left(\mathsf{X}'\mathsf{D}^{-1}\mathsf{X}\right)^{-1}}\right] && (3.130)\\
&= \mathsf{Y}_{\star j}'\mathbb{E}_{\mathsf{X}}\left[\frac{\mathsf{D}^{-1}\mathsf{X}\mathsf{X}'\mathsf{D}^{-1}}{\mathsf{X}'\mathsf{D}^{-2}\mathsf{X}}\right]\mathsf{Y}_{\star j}. && (3.131)
\end{aligned}$$

To calculate the expectation, first note that the distribution of X is uniform on the unit $(\mathsf{m}-1)$-sphere. Let

$$\begin{aligned}
\tilde{\mathsf{X}} &\sim \mathrm{N}\left(0, I_{\mathsf{m}\times\mathsf{m}}\right) && (3.132)\\
\rho &\sim \sqrt{\chi^2_{\mathsf{m}}} && (3.133)
\end{aligned}$$

and note that if ρ and X are independent then $\tilde{\mathsf{X}}$ is equal in distribution $\rho\mathsf{X}$. Then

$$\begin{aligned}
\mathsf{E} &\equiv \mathbb{E}_{\mathsf{X}}\left[\frac{\mathsf{D}^{-1}\mathsf{X}\mathsf{X}'\mathsf{D}^{-1}}{\mathsf{X}'\mathsf{D}^{-2}\mathsf{X}}\right] && (3.134)\\
&= \mathbb{E}_{\tilde{\mathsf{X}}}\left[\frac{\mathsf{D}^{-1}\tilde{\mathsf{X}}\tilde{\mathsf{X}}'\mathsf{D}^{-1}}{\tilde{\mathsf{X}}'\mathsf{D}^{-2}\tilde{\mathsf{X}}}\right]. && (3.135)
\end{aligned}$$

For the off-diagonal entries of E we have

$$\begin{aligned}\mathsf{E}_{ij} &= \mathbb{E}_{\tilde{\mathsf{X}}}\left[\frac{\mathsf{d}_i^{-1}\mathsf{d}_j^{-1}\tilde{\mathsf{X}}_i\tilde{\mathsf{X}}_j}{\sum_{l=1}^{\mathsf{m}}\mathsf{d}_l^{-2}\tilde{\mathsf{X}}_l^2}\right] &(3.136)\\ &= 0 &(3.137)\end{aligned}$$

where $\mathsf{d}_i \equiv \mathsf{D}_{ii}$. For the diagonal entries $\mathsf{e}_i \equiv \mathsf{E}_{ii}$ of E we have

$$\mathsf{E}_{ii} = \mathbb{E}_{\tilde{\mathsf{X}}}\left[\frac{\mathsf{d}_i^{-2}\tilde{\mathsf{X}}_i^2}{\sum_{l=1}^{\mathsf{m}}\mathsf{d}_l^{-2}\tilde{\mathsf{X}}_l^2}\right] \quad (3.138)$$

which can be shown to be equal to

$$\int_0^\infty \frac{dt}{\mathsf{d}_i^2\left(1+2t/\mathsf{d}_i^2\right)\prod_{l=1}^{\mathsf{m}}\sqrt{1+2t/\mathsf{d}_l^2}} \quad (3.139)$$

using the results of Magnus (1986). To summarize,

$$\begin{aligned}\hat{\sigma}_j^2 &= \mathsf{Y}_{\star j}'\mathsf{E}\mathsf{Y}_{\star j} &(3.140)\\ &= \sum_{i=1}^{\mathsf{m}}\frac{\mathsf{Y}_{ij}^2}{\mathsf{d}_i^2}\int_0^\infty \frac{dt}{\left(1+2t/\mathsf{d}_i^2\right)\prod_{l=1}^{\mathsf{m}}\sqrt{1+2t/\mathsf{d}_l^2}}. &(3.141)\end{aligned}$$

3.3.7.6 Discussion: $\left(\hat{\sigma}_{j_{\bar{c}}}^2\right)^{(\mathrm{RUV-inv})}$

We wish to develop some intuition for $\hat{\sigma}_j^2$. Write $\hat{\sigma}_j^2$ as

$$\hat{\sigma}_j^2 = \sum_{i=1}^{\mathsf{m}}\mathsf{e}_i\mathsf{Y}_{\star j}^2. \quad (3.142)$$

It can be shown (see below) that $\sum_{i=1}^{\mathsf{m}}\mathsf{e}_i = 1$. $\hat{\sigma}_j^2$ is therefore a weighted average of $\mathsf{Y}_{\star j}^2$. To interpret this result, recall that U is the first $m-1$ left singular vectors of $R_X Y_c$. If we use the SVD as the method of factor analysis in Step 2 of RUV-4, $\mathsf{U} = \hat{W}_0^{(m-1)}$. We therefore assume that

$$\mathsf{U} \approx (W_0|W_1)$$

for an appropriate parameterization of W_0 and W_1. Then

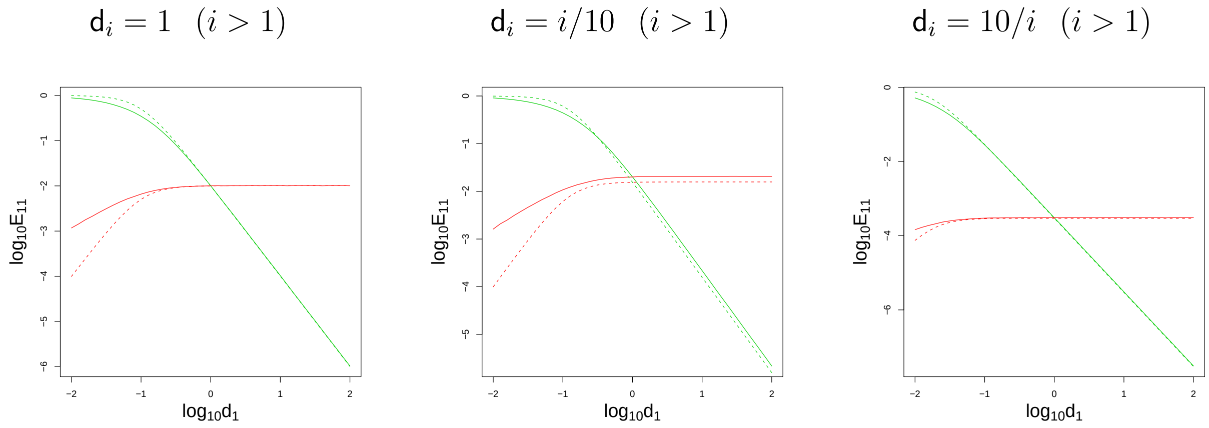
$$\begin{aligned}\mathsf{Y} &\approx (W_0|W_1)'Y &(3.143)\\ &= \begin{pmatrix} W_0'Y \\ W_1'Y\end{pmatrix} &(3.144)\\ &= \begin{pmatrix}\alpha \\ 0\end{pmatrix} + \varepsilon. &(3.145)\end{aligned}$$

A weighted average of $\mathsf{Y}_{\star j}^2$ is therefore an appropriate estimator of σ_j^2. The weights e_i should be small (ideally 0) for $i \leq k$ and large (ideally $\frac{1}{\mathsf{m}-k}$) for $i > k$.

With the inverse method, this is indeed what happens. The weights e_i are functions of the d_i. Now,

$$\mathsf{d}_i = \sum_{j_c} \mathsf{Y}_{\mathsf{ij_c}}{}^2. \tag{3.146}$$

For $i > k$ the d_i will all be approximately equal to one another and relatively small. For $i \leq k$ the d_i will be relatively large. We therefore want e_i to be small when d_i is large and vice versa. Indeed, if we consider a single d_i and hold all other $\mathsf{d}_{i'}$ constant, then e_i approaches 1 as d_i approaches 0. Conversely, e_i is asymptotically proportional to $1/\mathsf{d}_i^2$ as d_i^2 grows large. See Figure 3.7.



$\mathsf{d}_i = 1 \quad (i > 1)$

$\mathsf{d}_i = i/10 \quad (i > 1)$

$\mathsf{d}_i = 10/i \quad (i > 1)$

Figure 3.7: Plots of e_i as a function of d_i. In each plot $\mathsf{m} = 100$, d_1 is varied from 10^{-2} to 10^2, and d_i is kept fixed for $i > 1$. The solid green line is a plot of e_1. The solid red line is a plot of e_{10} (note that in each of the three plots, $\mathsf{d}_{10} = 1$). For purpose of comparison, the green and red dotted lines are plots of $(1/\mathsf{d}_1^2)/(\sum_{l=1}^{\mathsf{m}} 1/\mathsf{d}_l^2)$ and $(1/\mathsf{d}_{10}^2)/(\sum_{l=1}^{\mathsf{m}} 1/\mathsf{d}_l^2)$.

We now verify the claim $\sum_{i=1}^{\mathsf{m}} \mathsf{e}_i = 1$. Begin by noting

$$\mathsf{E} \;=\; \mathbb{E}_{\mathsf{X}}\left[\mathsf{V}\mathsf{V}'\right] \tag{3.147}$$

where

$$\mathsf{V} \equiv \frac{\mathsf{D}^{-1}\mathsf{X}}{\sqrt{\mathsf{X}'\mathsf{D}^{-2}\mathsf{X}}}. \tag{3.148}$$

Note that $||\mathsf{V}|| = 1$. Now,

$$\begin{aligned}\sum_{i=1}^{\mathsf{m}} \mathsf{e}_i &= \mathrm{tr}\left(\mathsf{E}\right) & (3.149)\\ &= \mathrm{tr}\left(\mathbb{E}_{\mathsf{X}}\left[\mathsf{V}\mathsf{V}'\right]\right) & (3.150)\\ &= \mathbb{E}_{\mathsf{X}}\left[\mathrm{tr}\left(\mathsf{V}\mathsf{V}'\right)\right] & (3.151)\\ &= \mathbb{E}_{\mathsf{X}}\left[\mathrm{tr}\left(\mathsf{V}'\mathsf{V}\right)\right] & (3.152)\\ &= 1 & (3.153)\end{aligned}$$

Next we investigate the distribution of $\hat{\sigma}_j^2$. It is easy to show that

$$\mathsf{V}'\mathsf{Y}_{\star j} = \frac{\mathsf{X}'\mathsf{D}^{-1}\mathsf{X}}{\sqrt{\mathsf{X}'\mathsf{D}^{-2}\mathsf{X}}}\mathbb{E}\left[\hat{\beta}_j^\star\middle|\mathsf{X}\right] + \mathsf{V}'\varepsilon_{\star j}. \qquad (3.154)$$

If we assume

$$\mathbb{E}\left[\hat{\beta}_j^\star\middle|\mathsf{X}\right] \approx 0$$

it follows that

$$\mathsf{V}'\mathsf{Y}_{\star j} \approx \mathsf{V}'\varepsilon_{\star j} \qquad (3.155)$$

and therefore that

$$\begin{aligned}\hat{\sigma}_j^2 &= \mathbb{E}_{\mathsf{X}}\left[\mathsf{Y}_{\star j}'\mathsf{V}\mathsf{V}'\mathsf{Y}_{\star j}\right] & (3.156)\\ &\approx \mathbb{E}_{\mathsf{X}}\left[\varepsilon_{\star j}'\mathsf{V}\mathsf{V}'\varepsilon_{\star j}\right] & (3.157)\\ &= \varepsilon_{\star j}'\mathsf{E}\varepsilon_{\star j}. & (3.158)\end{aligned}$$

Since $\varepsilon_{\star j} \sim N(0, \sigma_j^2 I)$ we conclude that

$$\hat{\sigma}_j^2 \mathrel{\dot{\sim}} \sigma_j^2 \sum_{i=1}^{\mathsf{m}} \mathsf{e}_i \chi_{1,i}^2 \qquad (3.159)$$

where the $\chi_{1,i}^2$ are IID χ_1^2.

We pause to interpret our analysis. One way to think of the inverse method is as follows: (1) Generate a random X. (2) Transform X into V. (3) Calculate $\mathsf{s}_j^2 \equiv (\mathsf{V}'\mathsf{Y}_{\star j})^2$. (4) Repeat (1-3) many times and average the resulting s_j^2. The key step is (2). Ideally, V will be a random unit vector in $\mathfrak{R}(\mathsf{W}_\perp)$. Then $(\mathsf{V}'\mathsf{Y}_{\star j})^2 \sim \sigma_j^2\chi_1^2$. However, we cannot sample from $\mathfrak{R}(\mathsf{W}_\perp)$ because W is unknown. We therefore sample from $\mathbb{R}^{\mathsf{m}}$ and attempt to orthogonalize to W. In step (2) we "effectively orthogonalize" X to W by multiplying X by D^{-1}. V is just $\mathsf{D}^{-1}\mathsf{X}$ rescaled to have unit length. V, therefore, is a random unit vector more or less confined to $\mathfrak{R}(\mathsf{W}_\perp)$.

Note that we can imagine generating an infinite number of s_j^2, each of which is approximately distributed as $\sigma_j^2\chi_1^2$. However, in (3.159) we see that the (approximate) distribution of $\hat{\sigma}_j^2$ can be expressed as the sum of a finite number of rescaled χ_1^2 random variables. The reason is that the s_j^2 are dependent. They are all calculated using the same observation of $\varepsilon_{\star j}$. The only difference between the s_j^2 is the V that is used. The dependence between the s_j^2 is therefore a consequence of the fact that the V are sampled from a finite dimensional space.

Now suppose we want to calculate the t statistic $t_j \equiv \hat{\beta}_j / \sqrt{\hat{\sigma}_j^2}$. If we want to use this t statistic to calculate p-values, etc., we must know its distribution. t_j does not follow the t distribution because $\hat{\sigma}_j^2$ does not follow a rescaled χ^2 distribution. Therefore, we cannot calculate exact p-values using standard methodology and software. However, we might wish to approximate the distribution of t_j by some t distribution. In this case it is necessary to approximate the distribution of $\hat{\sigma}_j^2/\sigma_j^2$ by some rescaled χ^2 distribution. In particular, we must come up with an "effective degrees of freedom." To do so we note that

$$\mathbb{E}\left[\sum_{i=1}^{\mathsf{m}} \mathsf{e}_i \chi_{1,i}^2\right] = 1 \tag{3.160}$$

$$\mathrm{Var}\left[\sum_{i=1}^{\mathsf{m}} \mathsf{e}_i \chi_{1,i}^2\right] = 2\sum_{i=1}^{\mathsf{m}} \mathsf{e}_i^2. \tag{3.161}$$

Let

$$\hat{r} \equiv 1/\sum_{i=1}^{\mathsf{m}} \mathsf{e}_i^2.$$

Then

$$\mathbb{E}\left[\frac{\chi_{\hat{r}}^2}{\hat{r}}\right] = 1 \tag{3.162}$$

$$\mathrm{Var}\left[\frac{\chi_{\hat{r}}^2}{\hat{r}}\right] = 2\sum_{i=1}^{\mathsf{m}} \mathsf{e}_i^2. \tag{3.163}$$

We therefore approximate the distribution of t_j by the t distribution with $\hat{r}$ degrees of freedom.

An interesting observation is that $\hat{r}$ may be useful for more than just specifying which t distribution to use when calculating p-values. As previously noted, V is a random unit vector more or less confined to $\mathfrak{R}(\mathsf{W}_\perp)$. If it were in fact the case that V was a unit vector distributed uniformly on the unit $\mathsf{m} - k - 1$ sphere in $\mathfrak{R}(\mathsf{W}_\perp)$, then $\hat{\sigma}_j^2$ would have a rescaled $\chi_{\mathsf{m}-k}^2$ distribution. Therefore, if in reality $\hat{\sigma}_j^2$ approximately follows a rescaled $\chi_{\hat{r}}^2$ distribution, it may be reasonable to regard $\hat{r}$ as a measure of the "effective dimension" of $\mathsf{W}_\perp$. We may therefore choose to estimate k as

$$\hat{k}^{(\mathrm{inv})} \equiv \mathsf{m} - \hat{r}. \tag{3.164}$$

Although we find this idea quite interesting, we have not found $\hat{k}^{(\mathrm{inv})}$ to perform any better in practice than the $\hat{k}$ described in Section 3.3.6.6. In some cases it performs notably worse. For example, $\hat{k}^{(\mathrm{inv})}$ may perform poorly when n_c is only marginally larger than m and the smaller eigenvalues of $\mathsf{Y}_c\mathsf{Y}_c'$ are noisy.

3.3.7.7 A Brief Note on Preprocessing

In Section 3.3.7.6 we noted that e_i approaches 1 as d_i approaches 0. This fact has important implications for data preprocessing. Several common preprocessing steps wholly or partially remove one or more degrees of freedom from the data. Consider a simple example. It is common practice to subtract away gene averages. This is equivalent to setting $Z = 1_{m\times 1}$ and multiplying Y by R_Z. This reduces the rank of Y to $m-1$. The smallest singular value of Y will be 0 and the inverse method will fail. One should not multiply Y by R_Z as a preprocessing step. Instead, one should multiply X and Y by $Z'_\perp$ as suggested in Section 3.3.6.2.

Subtracting off gene means is just one example of a preprocessing step that wholly or partially removes a degree of freedom from the data. We are in no position to discuss all such examples. Moreover, it is common for a researcher to be given preprocessed data without knowing the exact preprocessing methods that were used. Therefore, we need a general strategy for dealing with preprocessed data. One possible strategy is as follows: First, a researcher makes a scree plot of the (preprocessed) data. The researcher then notes whether there are any abnormally small singular values, and if so, how many. Suppose the researcher observes n_s abnormally small singular values. The researcher then takes the final n_s left singular vectors of Y and includes these vectors as columns of Z. This effectively removes the n_s troublesome dimensions of Y by transforming Y into a lower dimensional space.

3.3.8 The Functional Approach

In this section we introduce a new framework for understanding RUV-4 and for developing new, more general methods. In this framework, we transform the problem of estimating β into a standard prediction or function estimation problem. Control genes play the role of a training set.

The justification of the approach is rather informal. We find that the "technical assumptions" are less important than the "practical assumptions." In particular, the question of whether α is random or fixed is of secondary importance. By contrast, the assumption that the $\alpha_{\star j_c}$ are "representative" of the $\alpha_{\star j_{\bar{c}}}$ takes center stage.

3.3.8.1 Motivating Example

We begin with a motivating example. The example will demonstrate a weakness of RUV-4 and the need for a new approach. Recall that if we model $\alpha_{\star j}$ as random with expectation 0 and variance Σ, then $\hat{\beta}^{(\mathrm{RUV-inv})}$ is (approximately) the minimum variance unbiased linear estimator of β. This is a reasonable estimator to use if $\alpha_{\star j}$ follows a (multivariate) normal distribution. However, if α is not normally distributed, other estimators may be preferable. Our example will illustrate this fact.

Recall the examples of Section 3.3.4 in which $m = 2$ and $k = 1$. Recall that $\hat{\beta} = b_{YX} - \hat{b}_{WX}\hat{\alpha}$, and that we may interpret $\hat{\beta}_j$ as the vertical distance from the point $(W_0'Y_{\star j}, X'Y_{\star j})$

to the line that passes through the origin and has slope $\hat{b}_{WX}$ (see Figures 3.1 and 3.2). In the examples of Section 3.3.4, α was normally distributed. Here we consider an example in which α_j equals either -1 or 1, each with probability 0.5. This example is shown in Figure 3.8. As before, $\hat{\beta}^{(\mathrm{RUV-4})}$ is given by the vertical distance to the orange line. However, simple visual inspection suffices to convince us that other estimators may be preferable. For example, we have drawn horizontal red lines through the vertical mean of the control genes of each of the two clouds. Instead of using the vertical distance to the orange line, we may prefer to estimate β by the vertical distance to the horizontal red lines.

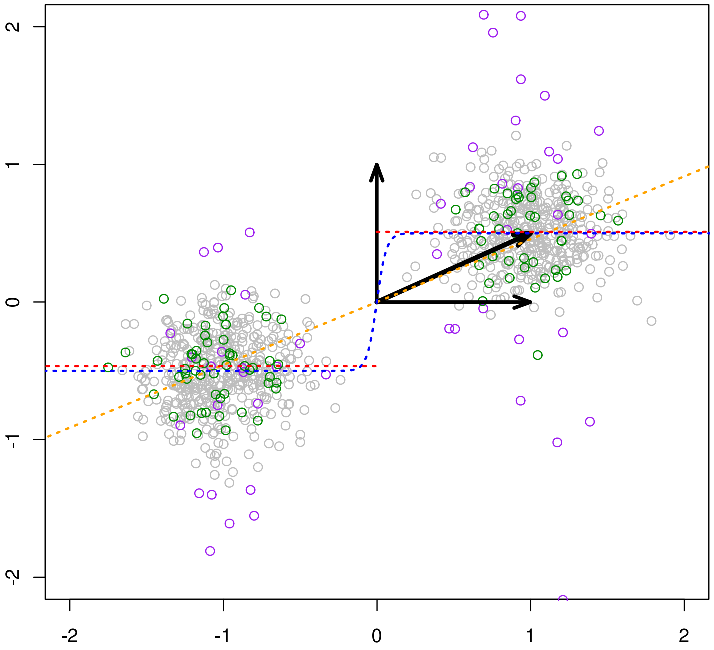


Figure 3.8: An example in which α is not normally distributed. See main text for commentary. The simulated data were generated as follows: $X = (0, 1)'$; $W = (1, 0.5)'$; α_j equals either -1 or 1, each with probability 0.5; $\epsilon_{ij} \sim \mathrm{N}(0, \frac{1}{16})$; $\beta_j \sim \mathrm{N}(0, 1)$ for $1 \leq j \leq 50$; $\beta_j = 0$ for $51 \leq j \leq 1000$.

We will now try to formalize the intuition provided by Figure 3.8. We begin by writing

$$b_{YX} \quad = \quad \beta + b_{WX}\alpha + \zeta. \tag{3.165}$$

We may rewrite this as

$$b_{YX} \quad = \quad \beta + B(\alpha) + \zeta \tag{3.166}$$

where B denotes the conditional bias of b_{YX} as a function of α, i.e.

$$\begin{aligned} B(\alpha) &\equiv \mathbb{E}\left[b_{YX} - \beta|\alpha\right] && (3.167)\\ &= b_{WX}\alpha. && (3.168)\end{aligned}$$

In this context, we may think of $\hat{\beta}^{(\mathrm{RUV-4})}$ as an "approximately de-biased" version of b_{YX}, i.e.

$$\begin{aligned} \hat{\beta} &= b_{YX} - \hat{b}_{WX}\hat{\alpha} && (3.169)\\ &= b_{YX} - \hat{B}(\hat{\alpha}) && (3.170)\end{aligned}$$

where $\hat{B}(\hat{\alpha}) \equiv \hat{b}_{WX}\hat{\alpha}$.

In light of (3.170), we see that the quality of $\hat{\beta}$ as an estimator of β is directly determined by the quality of $\hat{B}(\hat{\alpha})$ as an estimator of $B(\alpha)$. We see intuitively that $\hat{B}(\hat{\alpha})$ is a good estimator of $B(\alpha)$ in Figure 3.2 but not in Figure 3.8. What is the difference? Consider the quantity $\mathbb{E}\left[B(\alpha)|\hat{\alpha}\right]$. We may think of $\mathbb{E}\left[B(\alpha)|\hat{\alpha}\right]$ as the "best guess" of the unobserved quantity $B(\alpha)$ given the observed quantity $\hat{\alpha}$. Note that $\mathbb{E}\left[B(\alpha)|\hat{\alpha}\right]$ is a function of $\hat{\alpha}$. Indeed,

$$\begin{aligned} \mathbb{E}\left[B(\alpha)|\hat{\alpha}\right] &= \mathbb{E}\left\{\mathbb{E}\left[b_{YX} - \beta|\alpha\right]|\hat{\alpha}\right\} && (3.171)\\ &= \mathbb{E}\left\{\mathbb{E}\left[b_{YX} - \beta|\alpha, \hat{\alpha}\right]|\hat{\alpha}\right\} && (3.172)\\ &= \mathbb{E}\left[b_{YX} - \beta|\hat{\alpha}\right] && (3.173)\\ &= \mathbb{B}(\hat{\alpha}) && (3.174)\end{aligned}$$

where

$$\mathbb{B}(\hat{\alpha}) \equiv \mathbb{E}\left[b_{YX} - \beta|\hat{\alpha}\right]. \qquad (3.175)$$

We may think of $\mathbb{B}(\hat{\alpha})$ as the "ideal" estimator of $B(\alpha)$. We cannot calculate $\mathbb{B}(\hat{\alpha})$ itself because we do not know the function B, the distribution of α, or σ^2. However, it is nonetheless the case that a good estimator of $B(\alpha)$ will be some function of $\hat{\alpha}$ that closely approximates $\mathbb{B}(\hat{\alpha})$.

It turns out that $\hat{B}_j(\hat{\alpha})$ closely approximates $\mathbb{B}_j(\hat{\alpha})$ if α_j is normally distributed with expectation 0. To see this, assume

$$\alpha_j \sim N(0, \psi^2). \qquad (3.176)$$

It follows that

$$\begin{aligned}
\mathbb{B}_j(\hat{\alpha}) &= \mathbb{E}\left[B_j(\alpha)|\hat{\alpha}\right] && (3.177)\\
&= \mathbb{E}\left[b_{WX}\alpha_j|\hat{\alpha}_j\right] && (3.178)\\
&= b_{WX}\mathbb{E}\left[\alpha_j|\hat{\alpha}_j\right] && (3.179)\\
&= b_{WX}\left(\frac{\psi^2}{\psi^2+\sigma_j^2}\right)\hat{\alpha}_j && (3.180)\\
&= \mathbb{E}\left[\hat{b}_{WX}\right]\hat{\alpha}_j && (3.181)\\
&\approx \hat{b}_{WX}\hat{\alpha}_j && (3.182)\\
&= \hat{B}_j(\hat{\alpha}). && (3.183)
\end{aligned}$$

Thus, $\hat{B}_j(\hat{\alpha}) \approx \mathbb{B}_j(\hat{\alpha})$ and we consider $\hat{B}_j(\hat{\alpha})$ to be a good estimator of $B_j(\alpha)$.

In the example of Figure 3.8, however, α is not normally distributed and $\mathbb{B}_j(\hat{\alpha})$ is no longer a linear function of $\hat{\alpha}_j$. Indeed, it can be shown that in this particular example

$$\mathbb{B}_j(\hat{\alpha}) = \frac{1}{2}\left(\frac{e^{8(\hat{\alpha}_j+1)^2} - e^{8(\hat{\alpha}_j-1)^2}}{e^{8(\hat{\alpha}_j+1)^2} + e^{8(\hat{\alpha}_j-1)^2}}\right). \qquad (3.184)$$

This function is plotted in Figure 3.8 in blue. It is no longer true that $\hat{B}_j(\hat{\alpha}) \approx \mathbb{B}_j(\hat{\alpha})$; the orange line does not approximate the blue curve. $\hat{B}(\hat{\alpha})$ is no longer a good estimator of $B(\alpha)$.

3.3.8.2 The Functional Approach, Part I

To summarize our discussion so far: $b_{YX} = \beta + b_{WX}\alpha + \zeta$. In RUV-4, we estimate $B(\alpha) = b_{WX}\alpha$ by $\hat{B}(\hat{\alpha}) = \hat{b}_{WX}\hat{\alpha}$ and set $\hat{\beta} = b_{YX} - \hat{B}(\hat{\alpha})$. This works well if α is normally distributed, but does not necessarily work well otherwise. The important point to notice is that when we estimate $B(\alpha)$ in RUV-4 we effectively do so in two parts: we estimate the linear function B by the linear function $\hat{B}$, and we estimate α by $\hat{\alpha}$. We then combine these and use $\hat{B}(\hat{\alpha})$ as our estimate of $B(\alpha)$. This seams reasonable at first, but as we have seen, the "best" estimate of $B(\alpha)$ is not necessarily linear in $\hat{\alpha}$.

These considerations inspire a new approach. We do not attempt to estimate W and then estimate β by linear regression. We do not focus our attention on the estimation of b_{WX}. We do not estimate $B(\alpha)$ by $\hat{B}(\hat{\alpha})$. *Rather, our goal is to directly estimate the function* $\mathbb{B}$. Equipped with an estimate $\hat{\mathbb{B}}$ of $\mathbb{B}$ we then estimate β by $b_{YX} - \hat{\mathbb{B}}(\hat{\alpha})$. We call this the functional approach, or RUV-fun. Note that unlike RUV-2 or RUV-4, RUV-fun does not refer to a specific algorithm but rather to a general strategy; we may estimate $\mathbb{B}$ by any method we see fit. Indeed, we may view RUV-4 as a special case of RUV-fun.

How might we estimate $\mathbb{B}$? As always, the key is the control genes. Recall that

$$\begin{aligned}
b_{YX} &= \beta + B(\alpha) + \zeta && (3.185)\\
&\approx \beta + \mathbb{B}(\hat{\alpha}) + \zeta && (3.186)
\end{aligned}$$

and thus for the control genes we have

$$(b_{YX})_c \quad \approx \quad \mathbb{B}_c(\hat{\alpha}) + \zeta_c. \tag{3.187}$$

We can use $(b_{YX})_c$ and $\hat{\alpha}_c$ to help us estimate $\mathbb{B}$. However, we also need an additional assumption. We need an additional assumption because $\mathbb{B}$ is a vector of n functions:

$$\mathbb{B}(\hat{\alpha}) \quad = \quad (\mathbb{B}_1(\hat{\alpha}_{\star 1}), \mathbb{B}_2(\hat{\alpha}_{\star 2}), ..., \mathbb{B}_n(\hat{\alpha}_{\star n})) \tag{3.188}$$

In principle, each of these $\mathbb{B}_j$ may be different functions. However, we only have one observation of each $\hat{\alpha}_{\star j}$ and each $(b_{YX})_j$. It is not feasible to estimate the entire function $\mathbb{B}_j$ from a single observation. Moreover, the fundamental idea of the functional approach is to use the biases of the control genes to help us estimate the biases of the non-control genes. This is not possible if we treat each $\mathbb{B}_j$ as its own distinct entity. We need an assumption that will relate the $\mathbb{B}_j$ to one another in some way, and allow us to estimate the $\mathbb{B}_j$ jointly. We are in no position to estimate all n $\mathbb{B}_j$ separately.

Faced with this dilemma, we make a very strong technical assumption. We assume that the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ pairs are IID. This is the key technical assumption of the functional approach. From this assumption it follows that

$$\mathbb{B}_1 = \mathbb{B}_2 = ... = \mathbb{B}_n.$$

We can now define

$$\mathbb{B}_0 \equiv \mathbb{B}_j.$$

$\mathbb{B}_0$ is the function that we will estimate.

Each $(\hat{\alpha}_{\star j_c}, (b_{YX})_{j_c})$ pair provides an estimate of $\mathbb{B}_0$ evaluated at a specific point. In other words, control genes play the role of a training set in a prediction problem. The $\hat{\alpha}_{\star j_c}$ are the predictors, and the $(b_{YX})_{j_c}$ are the response variables. We may choose to estimate $\mathbb{B}_0$ by any of a number of methods. We need not restrict ourselves to linear functions, or even parametric functions. We have at our disposal the numerous methods available in the prediction, function estimation, and machine learning literature.

3.3.8.3 The Assumptions of RUV-fun

In this section we discuss the assumptions of the functional approach. We distinguish between "technical assumptions" and "practical assumptions." The technical assumptions include the modeling assumptions of Section 3.3.1 and the assumption that the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ pairs are IID. We find the technical assumptions implausible. We also argue that violations of the technical assumptions do not necessarily lead the functional approach to perform poorly. The practical assumptions are less rigorous than the technical assumptions. We find these assumptions plausible. We also believe these assumptions must be satisfied to ensure the functional approach performs well. We argue that that the practical assumptions may also be used as an informal justification of the functional approach.

We begin with a more careful look at the main technical assumption of RUV-fun. The main technical assumption of RUV-fun is that the pairs $(\hat{\alpha}_{\star j}, B_j(\alpha))$ are IID. This assumption does not follow as a necessary consequence of the modeling assumptions presented in Section 3.3.1. For example, suppose we model α as fixed. Then the $\hat{\alpha}_{\star j}$ will not be IID. Suppose instead we model the $\alpha_{\star j}$ as random and IID. Even then, the $\hat{\alpha}_{\star j}$ need not be IID. Recall that $\hat{\alpha}_{\star j} = \alpha_{\star j} + \xi_{\star j}$. The $\xi_{\star j}$ are not IID unless $\sigma_j^2 = \sigma_0^2$ for all j. However, if we do assume that the $\alpha_{\star j}$ are IID and that $\sigma_j^2 = \sigma_0^2$ for all j, it does follow that the pairs $(\hat{\alpha}_{\star j}, B_j(\alpha))$ are IID.

To satisfy the main technical assumption of the functional approach we assume that the $\alpha_{\star j}$ are IID and $\sigma_j^2 = \sigma_0^2$ for all j. We find the assumption that $\sigma_j^2 = \sigma_0^2$ for all j to be implausible. However, we also believe this assumption to be relatively unimportant. Recall the discussion of Section 3.3.7.4. We noted that $\frac{1}{n_c}(Y_c Y_c')$ is a biased estimate of Σ_j unless $\sigma_j^2 = \bar{\sigma}_c^2$. Under the assumption that $\sigma_j^2 = \sigma_0^2$ for all j, $\sigma_j^2 = \bar{\sigma}_c^2 = \sigma_0^2$ and $\frac{1}{n_c}(Y_c Y_c')$ is an unbiased estimate of Σ_j. We may view the consequences of a violation of the assumption that $\sigma_j^2 = \sigma_0^2$ as analagous to the consequences of using a biased estimate of Σ_j in Section 3.3.7.4. We argue in Section B.1.3 of the appendix that the consequences of using a biased estimate of Σ_j are not severe. Likewise, we do not believe that the consequences of a violation of the assumption that $\sigma_j^2 = \sigma_0^2$ are very severe.

We have just argued that the main technical assumption of the functional approach is false but that in practice this doesn't matter. This is somewhat comforting. However, the falsity of the technical assumptions does raise an unsettling conceptual problem. It is no longer clear what we are estimating! If the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ pairs are not IID, it is false that $\mathbb{B}_1 = \mathbb{B}_2 = ... = \mathbb{B}_n$. If it is false that $\mathbb{B}_1 = \mathbb{B}_2 = ... = \mathbb{B}_n$, it follows that $\mathbb{B}_0$, as we have defined it, does not exist. This is disturbing, since the goal of the functional approach is to estimate $\mathbb{B}_0$.

Is there a better way to define $\mathbb{B}_0$? Ideally, $\mathbb{B}_0$ would satisfy three criteria. Firstly, we would like that $\mathbb{B}_0(\alpha_{\star j})$ approximately equal $\mathbb{B}_j(\alpha_{\star j})$ with high probability. Secondly, we want $\mathbb{B}_0$ to exist even when $\mathbb{B}_j \neq \mathbb{B}_{j'}$. Finally, we would like $\mathbb{B}_0$ to have some relatively simple, real-world interpretation. Unfortunately, we are unable to provide any such definition of $\mathbb{B}_0$. For example, suppose we try defining $\mathbb{B}_0$ as the average of the $\mathbb{B}_j$. This definition of $\mathbb{B}_0$ would satisfy the second criterion, but not the first. To see this, suppose that α is fixed. In this case, each $\mathbb{B}_j$ is simply a constant function. Specifically, $\mathbb{B}_j(\hat{\alpha}) = W\alpha_{\star j}$. Thus, the average of the $\mathbb{B}_j$ is also just a constant function. Alternatively, suppose we try defining $\mathbb{B}_0$ as $\mathbb{E}\left[B_J(\alpha_{\star J}) | \hat{\alpha}_{\star J}\right]$ where J is a random variable distributed uniformly on $\{1, 2, ..., n\}$. This may satisfy the first two criteria, but it is not clear to us what the real-world interpretation of this function might be.

Fortunately, from a purely practical point of view, it does not necessarily matter if $\sigma_j^2 = \sigma_0^2$, or if the $\alpha_{\star j}$ are IID, or even if $\mathbb{B}_0$ exists. What matters is whether the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ are well described by the function $\hat{\mathbb{B}}_0$ that we have fit to the $(\hat{\alpha}_{\star j_c}, (b_{YX})_{j_c})$. This is still possible even if the technical assumptions are violated.

When can we expect the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ to be well described by $\hat{\mathbb{B}}_0$? There is no simple answer. However, for guidance, we now present two important "practical assumptions" of the functional approach. The practical assumptions are relatively informal, and we do not attempt to treat them rigorously. Nonetheless, these assumptions warrant serious consideration, and are critical to the success of the functional approach. The first assumption is that genes with similar $\hat{\alpha}_{\star j}$ experience similar biases. In other words,

$$\hat{\alpha}_{\star j} \approx \hat{\alpha}_{\star j'} \quad \rightarrow \quad B_j(\alpha) \approx B_{j'}(\alpha). \tag{3.189}$$

The second is that the control genes are representative of the other genes. In particular, the (realized) distribution of the $\hat{\alpha}_{\star j_c}$ must be roughly the same as the distribution of the $\hat{\alpha}_{\star j_{\bar{c}}}$. Moreover, the biases of the control genes must not differ systematically from those of the non-control genes.

These are strong assumptions, and have important implications for selecting an appropriate set of control genes. As a concrete example, consider spike-in controls (see Chapter 2 for a discussion of spike-in controls). These controls likely exhibit unwanted variation related to their own preparation. Other genes do not. The biases of the spike-in controls are therefore likely to differ systematically from the biases of non-control genes. Spike-in controls may be a poor choice for the functional approach.

The importance of the "practical assumptions" is not limited to the selection of control genes. These two assumptions may also be used as an informal — but relatively realistic — justification of the functional approach. If we assume (3.189) and that the control genes are representative, it follows that we may very roughly approximate the bias of gene j by the $(b_{YX})_{j_c}$ of the nearest control gene. In other words, we may very roughly approximate $B_j(\alpha)$ by $(b_{YX})_{c(j)}$ where

$$c(j) = \operatorname*{argmin}_{j_c} d(\hat{\alpha}_{\star j}, \hat{\alpha}_{\star j_c})$$

and where d is some appropriate distance measure. Of course, we may get a better estimate of $B_j(\alpha)$ by taking the average of the $(b_{YX})_{j_c}$ of several nearby control genes instead of just the single closest, and may get a still better estimate by fitting some curve to the $(b_{YX})_{j_c}$ of all the control genes. In this way, we are led back to the functional approach. Although informal, we might consider this to be the most appropriate justification for the functional approach. For better or worse, the approach is intrinsically ad hoc.

To conclude this section, we consider the utility of the technical assumptions. We have argued at several points that the technical assumptions are neither plausible nor necessary for the success of the functional approach. However, we do not wish to imply that the technical assumptions are useless. Consideration of the technical assumptions can be very helpful in alerting us to potential problems. If the technical assumptions were true, they would justify the use of the functional approach. Thus, by considering the ways in which the assumptions are false, we are led to consider ways in which the functional approach might fail. An example is our discussion regarding the violation of the assumption that $\sigma_j^2 = \sigma_0^2$. We ultimately concluded that the consequences of a violation of this assumption were not severe. However, we arrived at this conclusion only after investigating the matter in B.1.3 of the appendix. Finally, note that most off-the-shelf prediction algorithms effectively assume the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ to be IID. Some algorithms may be more sensitive to violations of this assumption than others.

3.3.8.4 RUV-4 as RUV-fun

We noted in Section 3.3.8.2 that RUV-4 may be interpreted as a special case of RUV-fun. More specifically, we may interpret RUV-4 as a parametric version of RUV-fun in which we constrain $\hat{\mathbb{B}}_0$ to be a linear function that passes through the origin, and in which we fit $\hat{\mathbb{B}}_0$ by least squares. In this section we explore more fully the interpretation of RUV-4 as a special case of RUV-fun.

We consider first the rationale for constraining $\hat{\mathbb{B}}_0$ to be a linear function that passes through the origin. In the derivation of RUV-4 in Section 3.3.3 we do not address this issue explicitly. Implicitly, however, we rationalize constraining $\hat{\mathbb{B}}_0$ to be a linear function that passes through the origin on the grounds that B is a linear function that passes through the origin. As the example of Section 3.3.8.1 shows, however, linearity of B does not imply linearity of $\hat{\mathbb{B}}_0$. If we assume that $\alpha_{\star j} \sim N(0, \Sigma)$ for all j, it follows that $\mathbb{B}_0$ is a linear function that passes through the origin. This would justify the constraint on $\hat{\mathbb{B}}_0$. However, as noted in Section 3.3.5.4, we find this assumption implausible.

In the RUV-fun framework, the rationale for constraining $\hat{\mathbb{B}}_0$ to be a linear function that passes through the origin is primarily empirical. It is of secondary importance whether the $\alpha_{\star j}$ are actually distributed as $N(0, \Sigma)$, or whether α is even random. Of primary importance is whether the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ pairs are well described by $\hat{\mathbb{B}}_0$. The best justification for constraining $\hat{\mathbb{B}}_0$ to be a linear function that passes through the origin is that, in practice, it does seem that the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ pairs are well described by such a function. See Section 3.5 for evidence.

We now revisit the discussion of Section 3.3.4. In Section 3.3.4 we observed that RUV-2 seems to provide a better estimate of W than RUV-4, but RUV-4 nonetheless provides a better estimate of β. By viewing RUV-4 as a special case of RUV-fun we may gain perspective on this curious fact. In the RUV-fun framework, what matters is that the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ are well described by a linear function $\hat{\mathbb{B}}_0$ that passes through the origin. Unlike with RUV-4, with RUV-fun we do not care whether the coefficients of the linear function $\hat{\mathbb{B}}_0$ provide a good approximation of b_{WX}. If we view RUV-4 simply as RUV-fun in disguise, we really don't care about $\hat{W}$ at all. $\hat{W}$ is at best a means to an ends, at worst a distraction. Said another way, $\hat{W}$ does not determine $\hat{\beta}$; $\hat{\beta}$ determines $\hat{W}$.

An example of this "abuse of $\hat{W}$" can be seen in (3.179) – (3.181). In (3.180) we write $\mathbb{B}_j(\hat{\alpha})$ as the product of three terms: b_{WX}, a shrinkage factor $\psi^2 / \left(\psi^2 + \sigma_j^2\right)$, and $\hat{\alpha}_j$. Conceptually, the shrinkage factor is best understood as something having to do with α_j and $\hat{\alpha}_j$. In particular,

$$\mathbb{E}\left[\alpha_j | \hat{\alpha}_j\right] \;=\; \left(\frac{\psi^2}{\psi^2 + \sigma_j^2}\right) \hat{\alpha}_j. \tag{3.190}$$

However, in RUV-4, we effectively incorporate the shrinkage factor into our estimate of b_{WX}. Recall that

$$\mathbb{E}\left[\hat{b}_{WX}\right] \;=\; b_{WX}\left(\frac{\psi^2}{\psi^2+\sigma_j^2}\right). \tag{3.191}$$

If we wanted to more faithfully follow the general strategy of RUV-4 as it was presented in Section 3.3.3, we would first want to find an unbiased, or nearly unbiased, estimate of b_{WX}. We could then construct a more proper estimate of W. Finally, instead of estimating α_j by $\hat{\alpha}_j$, we would instead estimate α by some estimate of $\mathbb{E}\left[\alpha_j|\hat{\alpha}_j\right]$, e.g. $\left[\hat{\psi}^2/\left(\hat{\psi}^2+\hat{\sigma}_j^2\right)\right]\hat{\alpha}_j$, where $\hat{\psi}^2$ is some estimate of ψ^2.

3.3.8.5 The Functional Approach, Part II

Until now we have retained the modeling assumptions of Section 3.3.1. In Section 3.3.8.1 we considered an "unusual" distribution of α, but we did not depart from the model of Section 3.3.1. In Section 3.3.8.2 we introduced the assumption that the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ pairs are IID. This assumption added to, but did not modify or replace, the assumptions of Section 3.3.1.

We have retained the modeling assumptions of Section 3.3.1 until now so that we could discuss the functional approach in a familiar setting. In Section 3.3.8.1 we demonstrated that the functional approach could be used to develop better methods to estimate β. In Section 3.3.8.3 we discussed the nature of the assumptions of the functional approach. In Section 3.3.8.4 we demonstrated that the functional approach could be used to better understand RUV-4. All of this was possible in the familiar setting of the model of Section 3.3.1.

However, the functional approach is most powerful when we abandon the model of Section 3.3.1. We now present a new model. Define m, n, X, and Y as we have previously. Let $\mathcal{P}$ be a $K \times n$ matrix of observed predictors. Let $\mathcal{S} \equiv X'Y = b_{YX}$ denote the observed "signal of interest." Let β be an unobserved $1 \times n$ parameter of interest. Let f denote some unknown function. We model $\mathcal{S}_j$ as

$$\mathcal{S}_j = \beta_j + f(\mathcal{P}_{\star j}) + \delta_j. \tag{3.192}$$

We assume that $\mathcal{P} \perp\!\!\!\perp \delta$ and that the $(\mathcal{P}_{\star j}, \delta_j)$ are IID. We do not assume here any constraints on the function f, nor do we assume here anything about the distribution of δ. However, in any given application of the functional approach, we will need to make assumptions regarding the form of f and the distribution of δ. Note that the interpretation of β here is similar, though not identical, to its interpretation in Section 3.3.1. By abuse of notation, we use the same symbol. Likewise, the interpretation of K here is similar, but not identical, to its former interpretation. Note that δ here is not related in any way to the δ of Section 3.3.7.4. We assume that $\beta_c = 0$. To estimate f we note that $\mathcal{S}_{jc} = f(\mathcal{P}_{\star j_c}) + \delta_{j_c}$ and fit $\hat{f}$ using any method of our choosing. We estimate β as the difference between the observed signal and the predicted signal:

$$\hat{\beta}_j^{(\mathrm{RUV-fun})} \;\equiv\; \mathcal{S}_j - \hat{f}(\mathcal{P}_{\star j}). \tag{3.193}$$

The most important feature of our new model is that we place no restrictions on what we may include in $\mathcal{P}$. To be sure, we will often wish to fill $m-1$ rows of $\mathcal{P}$ with $X_\perp' Y$. (Note that by setting $\mathcal{P} = X_\perp' Y$, constraining $\hat{f}$ to be a linear function that passes through the origin, and fitting $\hat{f}$ by least squares, we recover $\hat{\beta}^{(\mathrm{RUV-inv})}$). However, we may also include any other variables of our choosing. We may include non-linear features of the data. For example, we may include an initial estimate of σ^2. We may also include "outside" sources of information. For example, we may wish to include the GC content of the genes. Including information on GC content may be particularly useful when applying RUV-fun to RNA-seq data.

3.3.9 Variations and Extensions

In this section we consider three unrelated enhancements to the basic RUV methods. None of these enhancements are technically sophisticated, but all are quite useful. The first, the ridged inverse method, is useful when n_c is small. The second, empirical controls, is useful when no control genes are available but β is known to be sparse. The third, rescaled and empirical variances, improves control of the type 1 error rate.

3.3.9.1 The Ridged Inverse Method (RUV-rinv)

In Section 3.3.6.4 we noted that setting K too large may substantially increase the variance of $\hat{\beta}$ if n_c is not large. This is a problem, since $\hat{\beta}^{(\mathrm{RUV-inv})} = \hat{\beta}^{(\mathrm{RUV-4})}$ with $K = m - 1$. If n_c is only slightly larger than m, $\hat{\beta}^{(\mathrm{RUV-inv})}$ will not be a good estimator β. This is particularly easy to see in the context of the functional approach. Let $\mathcal{P} = X_\perp' Y$, constrain $\hat{f}$ to be a linear function that passes through the origin, and fit $\hat{f}$ by least squares; i.e. let $\hat{f}(u) = \mathcal{S}_c \mathcal{P}_c' (\mathcal{P}_c \mathcal{P}_c')^{-1} u$. The resulting estimate of β will be identical to $\hat{\beta}^{(\mathrm{RUV-inv})}$. Now, the quality of $\hat{\beta}^{(\mathrm{RUV-inv})}$ as an estimator of β depends on the quality of $\hat{f}$ as an estimator of f. If n_c is only slightly larger than m, $\hat{f}$ will be noisy, and $\hat{\beta}^{(\mathrm{RUV-inv})}$ will be too.

One possible solution is to use ridge regression. (Readers unfamiliar with ridge regression may wish to consult, e.g. Friedman et al. (2009).) We can estimate $f(u)$ as

$$\hat{f}(u) \quad = \quad \mathcal{S}_c \mathcal{P}_c' (\mathcal{P}_c \mathcal{P}_c' + \lambda I)^{-1} u \tag{3.194}$$

where $\lambda \geq 0$ is a tuning parameter. We therefore define

$$\hat{\beta}^{(\mathrm{RUV-rinv})}(\lambda) \quad = \quad \mathcal{S} - \mathcal{S}_c \mathcal{P}_c' (\mathcal{P}_c \mathcal{P}_c' + \lambda I)^{-1} \mathcal{P}. \tag{3.195}$$

We drop the superscript when it is clear from context. We also drop the explicit dependence on λ. Note that in the notation of Section 3.3.7,

$$\hat{\beta} \quad = \quad \left[X' - X' Y_c Y_c' X_\perp (X_\perp' Y_c Y_c' X_\perp + \lambda I)^{-1} X_\perp'\right] Y. \tag{3.196}$$

We are unable to substantially simplify this expression.

To estimate σ^2 we use the inverse method. Define

$$\mathsf{V}^{(\mathrm{rinv})} \equiv \frac{\mathsf{X} - \mathsf{X}_\perp(\mathsf{X}_\perp'\mathsf{D}\mathsf{X}_\perp + \lambda I)^{-1}\mathsf{X}_\perp'\mathsf{D}\mathsf{X}}{||\mathsf{X} - \mathsf{X}_\perp(\mathsf{X}_\perp'\mathsf{D}\mathsf{X}_\perp + \lambda I)^{-1}\mathsf{X}_\perp'\mathsf{D}\mathsf{X}||}, \tag{3.197}$$

drop the superscript, and define

$$\left(\hat{\sigma}_j^2\right)^{(\mathrm{RUV-rinv})} \equiv \mathsf{Y}_{\star j}'\mathbb{E}_{\mathsf{X}}\left[\mathsf{V}\mathsf{V}'\right]\mathsf{Y}_{\star j}. \tag{3.198}$$

Again, drop the superscript when it is clear from context. We are unable to provide an analytic expression for $\hat{\sigma}_j^2$. Instead, we estimate $\mathbb{E}_{\mathsf{X}}\left[\mathsf{V}\mathsf{V}'\right]$ via simulation.

We need to find a good value for λ. Note that

$$\begin{aligned}\mathbb{E}\left[(\mathcal{P}_c\mathcal{P}_c')\right] &= \mathbb{E}\left[X_\perp' Y_c Y_c' X_\perp\right] &(3.199)\\ &= n_c\left(X_\perp'\Sigma X_\perp + \bar{\sigma}_c^2 I\right). &(3.200)\end{aligned}$$

The smallest eigenvalue of $\mathbb{E}\left[(\mathcal{P}_c\mathcal{P}_c')\right]$ is $n_c\bar{\sigma}_c^2$, but the smallest eigenvalue of $\mathcal{P}_c\mathcal{P}_c'$ may be considerably smaller. We might therefore wish to set λ equal to

$$\begin{aligned}\lambda_0 &= n_c\bar{\sigma}_c^2 &(3.201)\\ &= \sum_{j_c}\sigma_{j_c}^2. &(3.202)\end{aligned}$$

Of course, σ^2 is unknown and so is λ_0. However, we can estimate λ_0 if we have an estimate of σ^2. This raises a tricky problem: we can estimate σ^2 once we have a value for λ, but our desired value of λ requires an estimate of σ^2. Our solution to this problem is to estimate σ^2 using some other version of RUV. Which version of RUV is best for this purpose? RUV-inv is a poor choice. As mentioned in Section 3.3.6.6, when n_c is only slightly larger than m, $\hat{b}_{WX}$ is overfitted to the control genes, and the variance of $\hat{\beta}_{j_c}$ is less than $\sigma_{j_c}^2\left(1 + \hat{b}_{WX}\hat{b}_{WX}'\right)$. As a result, the RUV-inv estimate of σ_c^2 tends to be too small. Instead, we use the RUV-4 estimate, with $K = \hat{k}$. This estimate tends to be of the right size. We define

$$\hat{\lambda}_0 = \sum_{j_c}\left(\hat{\sigma}_{j_c}^2\right)^{(\hat{k})}. \tag{3.203}$$

This is the value of λ we use in all of the applications of RUV-rinv in this thesis.

Note that $\hat{\lambda}_0$ is not necessarily the "best" value of λ. Other values of λ may provide better results. For example, it may be better to use cross validation to find an optimal value for λ. We do not pursue alternative strategies for selecting λ in this thesis. We find that $\hat{\lambda}_0$ generally provides satisfactory results (see Sections 3.4 and 3.5). Moreover, computing $\hat{\lambda}_0$ is relatively computationally efficient, particularly when compared to methods such as cross validation.

Note also that ridge regression is not the only potential solution when n_c is only slightly larger than m. Other dimensionality reduction strategies are possible as well. Note in particular that principal components regression (PCR), a common alternative to ridge regression, is simply equivalent to RUV-4. We may prefer ridge regression to PCR for two reasons. The first is that the dimensionality reduction of ridge regression is "softer" than that of PCR. See, e.g. Friedman et al. (2009) for further discussion. The second reason is that the inverse method is somewhat more computationally efficient when used with ridge regression than when used with PCR. Applying the inverse method to PCR would require computing a new SVD for each X.

3.3.9.2 Empirical Controls

In many cases, a researcher will know *a priori* that $\beta_j = 0$ for many j, but not know the specific j for which $\beta_j = 0$. The researcher would like to discover the j for which $\beta_j = 0$. She may then use these genes as "empirical" control genes. Discovering empirical controls is often feasible, but may require some care. In this section we comment briefly on the strategy of empirical controls.

There is no single method for discovering empirical controls. This is why we refer simply to the "strategy" of empirical controls. Nonetheless, we might describe a typical application of the strategy of empirical controls as follows: First, a researcher designates an initial set of control genes. She then applies an initial analysis, such as RUV-4. Finally, she notes which genes are found to be significantly associated with X at some false discovery rate (FDR) and designates all other (insignificant) genes as empirical negative controls. The initial set of negative controls and the initial method of analysis are left to the discretion of the researcher. However, even this "typical application" is not set in stone. For example, a researcher may fear that her p-values are biased (either systematically inflated or deflated) and not trust the FDR. However, if she believes, for example, that no more than 100 entries of β are non-zero, she may simply rank the genes by p-value and designate all but the top 100 as empirical negative controls.

The strategy of empirical controls can be iterated. For example, a researcher may begin with an initial set of control genes, generate a set of empirical controls, and then use this set of empirical controls to produce a refined set of empirical controls. A related point is that the initial (non-empirical) set of control genes need not be a "perfect" set of control genes. RUV-4 is relatively insensitive to violations of the control gene assumption, and it is often OK to include a few genes j such that $\beta_j \neq 0$ in the initial set of control genes. For example, if β is known to be sparse, it is often satisfactory to use *all* genes as an initial set of control genes. This fact is particularly useful when it is known that β is sparse, but nothing at all is known about which entries of β are non-zero.

Note that the strategy of empirical controls is much "safer" with RUV-4 than it is with RUV-2. The reason is that RUV-4 is less sensitive to violations of the control gene assumption. Consider a typical application of the strategy of empirical controls as described above. It is very unlikely that the initial analysis will properly identify all differentially expressed genes. The resulting set of empirical controls will very likely contain genes j such that $\beta_j \neq 0$. With RUV-2, this may be a serious problem. With RUV-4 it often is not.

Nonetheless, we recommend that a researcher who chooses to pursue the strategy of empirical controls do so with particular care. We are unaware of any argument that would guarantee that the strategy of empirical controls will always lead to better results. We therefore recommend that a researcher pause to inspect her set of empirical controls, and ensure that they seem reasonable. If a researcher chooses to iterate the strategy of empirical controls, we recommend that she pause and inspect her empirical controls after each iteration. Presumably only a very small number if iterations will be required; if the set of empirical controls keeps changing substantially with each iteration, this may be a sign that something is wrong. Similarly, we recommend that the researcher evaluate the quality of the initial analysis itself. Gene rankings, p-value histograms, and projection plots (see Section 3.5) are all helpful.

The initial analysis does not need to be perfect. The goal of the initial analysis is only to produce a set of empirical controls that is better than the set of initial controls. This has implications for the choice of method (and tuning parameters) in the initial analysis. Consider a simple example. Suppose that β is known to be sparse, but nothing is known about which elements of β are non-zero. A researcher may choose to include all genes in the set of initial controls. Suppose that some of the non-zero elements of β are quite large. Even though RUV-4 is relatively insensitive to violations of the control gene assumption, including genes with large β_j in the set of control genes may cause serious problems, particularly when K is large. The researcher may therefore choose to avoid RUV-inv, stick to RUV-4, and choose an appropriate, relatively small K "by hand," i.e. choose K based on p-value histograms, projection plots, etc. Using this approach, the very large entries of β can be reliably identified and discarded from the set of control genes. It may then be safe to pursue a second iteration of the strategy of empirical controls. In the second iteration, a larger value of K, or even RUV-inv may be appropriate.

Finally, note that the "typical" application of the strategy of empirical controls described above relies on an accurate estimation of the FDR. For this reason, it is important that the method used for the initial analysis exhibit good control of the type 1 error rate. If the method is too conservative, some genes that are actually differentially expressed will not appear to be differentially expressed. These genes will be improperly included in the set of empirical controls. If the method is anti-conservative, many genes that are not actually differentially expressed will appear to be differentially expressed. These genes will be improperly excluded from the set of empirical controls. The ability of a method to properly control the type 1 error rate is therefore an important consideration to keep in mind when selecting the method to be used in the initial analysis. In Sections 3.4 and 3.5 we see that RUV-rinv is often a suitable choice. Another possibility is to use either the method of rescaled variances or the method of empirical variances. These methods are the subject of the next section.

3.3.9.3 Rescaled and Empirical Variances

In Section 3.3.6.1 we define

$$\widehat{\mathrm{Var}}\left[\hat{\beta}_{j_{\bar{c}}}\middle|\alpha_{\star j_{\bar{c}}}\right] \equiv \left(1+\hat{b}_{WX}\hat{b}'_{WX}\right)\hat{\sigma}^2_{j_{\bar{c}}}. \tag{3.204}$$

In other words, conditional on $\hat{b}_{WX}$, we estimate the conditional variance of $\hat{\beta}_{j_{\bar{c}}}$ by multiplying $\hat{\sigma}^2_{j_{\bar{c}}}$ by a fixed constant. In Section 3.3.6.5 we note that, given a poor choice of K, the $\hat{\sigma}^2_j$ may be either systematically too large or too small. As a result, the type 1 error rate may be wrong. We now propose a different estimate of $\mathrm{Var}\left[\hat{\beta}_{j_{\bar{c}}} \middle| \alpha_{\star j_{\bar{c}}}\right]$. Let

$$\widehat{\mathrm{Var}}^{(\mathrm{rs})}\left[\hat{\beta}_{j_{\bar{c}}} \middle| \alpha_{\star j_{\bar{c}}}\right] \equiv \left(\frac{1}{n_c}\sum_{j_c}\frac{\hat{\beta}^2_{j_c}}{\hat{\sigma}^2_{j_c}}\right)\hat{\sigma}^2_{j_{\bar{c}}}. \tag{3.205}$$

We name this the "rescaled" estimate of the variance. Just as in (3.204), we estimate the conditional variance of $\hat{\beta}_{j_{\bar{c}}}$ by multiplying $\hat{\sigma}^2_{j_{\bar{c}}}$ by a fixed constant. However, in (3.205) we ignore the theory and simply use the control genes to figure out what the constant "should" be.

Note that $\widehat{\mathrm{Var}}\left[\hat{\beta}_{j_{\bar{c}}}\right]$ and $\widehat{\mathrm{Var}}^{(\mathrm{rs})}\left[\hat{\beta}_{j_{\bar{c}}} \middle| \alpha_{\star j_{\bar{c}}}\right]$ differ only by a fixed constant factor. Thus, t-statistics calculated using rescaled variances will also differ from standard t-statistics only by a fixed constant factor. In particular, the ordering of the t-statistics will be unaffected, as will the ordering of the p-values.

More generally, we might consider estimates of the form

$$\widehat{\mathrm{Var}}^{(\mathrm{emp})}\left[\hat{\beta}_{j_{\bar{c}}} \middle| \alpha_{\star j_{\bar{c}}}\right] \equiv \hat{g}\left(\hat{\sigma}^2_{j_{\bar{c}}}\right) \tag{3.206}$$

for some function $\hat{g}$. If we set $\hat{g}(u) = \left(1 + \hat{b}_{WX}\hat{b}'_{WX}\right)u$ we recover (3.204); if we set $\hat{g}(u) = \left(\frac{1}{n_c}\sum_{j_c}\hat{\beta}^2_{j_c}/\hat{\sigma}^2_{j_c}\right)u$ we recover (3.205). In general, however, we need not restrict $\hat{g}$ to be a linear, or even parametric, function. If a large proportion of genes are control genes (e.g. empirical controls) it may be possible to fit a nonparametric function $\hat{g}$ to the $\left(\hat{\sigma}^2_{j_c}, \hat{\beta}^2_{j_c}\right)$ pairs. Alternatively, if we do not have many control genes but believe that β is sparse, it may be possible to fit a nonparametric function $\hat{g}$ to the pairs $\left(\hat{\sigma}^2_j, \hat{\beta}^2_j\right)$ using some form of robust regression that ignores outliers. We refer to such methods generically as "the method of empirical variances." We will now discuss one such method in particular.

We begin by re-indexing the genes. We re-order the genes in order of increasing $\hat{\sigma}^2$ and then bin the genes into B bins of size S (the final bin may be smaller than S; we ignore this minor complication and assume $n = B \times S$). We then index genes by bin and number within bin, so that $\hat{\sigma}^2_{b,s}$ is the s^{th} gene in bin b. Note that $\hat{\sigma}^2_{b,s} \leq \hat{\sigma}^2_{b',s'}$ if $b < b'$ and that $\hat{\sigma}^2_{b,s} \leq \hat{\sigma}^2_{b,s'}$ if $s < s'$.

Let

$$s^\star(b) \equiv \operatorname*{argmax}_s \hat{\beta}^2_{b,s} \tag{3.207}$$

For each b, remove the $(b, s^\star(b))^{\mathrm{th}}$ gene from the dataset. We may view the removal of these genes as the removal of potential outliers. Alternatively, we may think of the remaining

$B(S-1)$ genes as a set of empirical controls. Now use some form of non-parametric regression to fit a function $\hat{g}_0$ to the $(\hat{\sigma}^2_{b,s}, \hat{\beta}^2_{b,s})$ pairs of the remaining $B(S-1)$ genes.

We do not want to set $\hat{g} = \hat{g}_0$. $\hat{g}_0$ is too small, because we have systematically removed from the dataset the genes with the largest values of $\hat{\beta}^2_{b,s}$. To fix this problem we set

$$\hat{g} \;=\; \nu \hat{g}_0 \tag{3.208}$$

for some value of ν. We choose to set

$$\nu^{-1} \;=\; \mathbb{E}\left[\frac{1}{S-1}\sum_{t \neq t^\star} \chi^2_t\right] \tag{3.209}$$

where t ranges from 1 to S, the χ^2_t are IID and follow a χ^2 distribution with 1 degree of freedom, and $t^\star = \operatorname{argmax} \chi^2_t$.

In all of the examples of this thesis we set $S = 10$. This is arbitrary. Other values of S may perform better. In particular, one may wish to choose S based on the degree of sparsity of β. For the non-parametric regression, we choose to use the minimum lower sets algorithm (Wright, 1978; Barlow et al., 1972). This method restricts $\hat{g}_0$ to be a non-decreasing function, but otherwise imposes few constraints on $\hat{g}_0$. We choose this method for its relative simplicity; one nice feature of the minimum lower sets algorithm is that it does not require us to set a bandwidth parameter. Other non-parametric regression methods may perform better.

3.4 Simulation Results

In this section we use simulated data to explore the performance of the various RUV methods. In Section 3.4.1 we outline the process we use to simulate the data. In Section 3.4.2 we compare the performance of RUV-2, RUV-4, and “vanilla” RUV-inv. We find that RUV-4 generally outperforms RUV-2, and that RUV-inv generally performs as well as RUV-4 at the optimal value of K. In Section 3.4.3 we compare the several variants of RUV-inv, both to one another and to SVA, LEAPP, and ICE.[2]

3.4.1 The Simulated Data

In all simulations we set $m = 50$ and $n = 10000$. We designate n_c genes as control genes. The value of n_c is specified separately for each simulation. In some simulations, the control genes are true negative controls, i.e. $\beta_c = 0$. In others, the “control genes” have been misspecified and $\beta_c \neq 0$. Note that when we refer to “control genes,” we refer to these genes that have

[2] Note that we do not include comparisons with LMM-EH (Listgarten et al., 2010), since no R package is currently available. However, we suspect that the performance of LMM-EH would be similar to that of ICE, although LMM-EH may exhibit somewhat better control of the type 1 error rate. Note Figure 2 of Listgarten et al. (2010); the ROC curve of LMM-EH and ICE are nearly identical.

been designated as negative controls, whether or not $\beta_c = 0$. Conversely, we refer to a gene j as a "true negative control" if $\beta_j = 0$, whether or not we have designated gene j to be used as control gene.

We generate the simulation data as follows:

- X is chosen uniformly at random from the unit $m - 1$ sphere.
- Each column of W_0 is chosen uniformly at random from the unit $m - 2$ sphere lying in the orthogonal complement of $\mathfrak{R}(X)$. Each column of W_0 is chosen independently of the others, and thus the columns of W_0 are not exactly orthogonal. W is then set equal to $W_0 + Xb_{WX}$. b_{WX} is specified separately for each simulation.
- Some entries of β are set equal to 0. Which entries of β are set equal to 0 is specified separately for each simulation. Non-zero entries of β are IID standard normal.
- The entries of α are independent and normally distributed with mean 0. The variance of α_{ij} depends only on the row i. Denote the variance of row i by $\sigma^2_{\alpha,i}$ and denote $\sigma_\alpha \equiv (\sigma_{\alpha,1}, ..., \sigma_{\alpha,k})$. Note that σ_α specifies the square roots of the variances, not the variances themselves.
- The individual gene variances σ^2_j (not to be confused with the $\sigma^2_{\alpha,i}$) are IID and distributed as $(0.025S + .025)^2$, where $S \sim \text{Exp}(1)$. This distribution roughly approximates empirical distributions of σ^2_j that we have observed in real data.
- The ϵ_{ij} are independent and normally distributed with mean 0 and variance σ^2_j.
- Finally, we set $Y = X\beta + W\alpha + \epsilon$.

Note that the key parameters that vary from one simulation to the next are: k, b_{WX}, σ_α, which entries of β equal 0, and which genes are designated as controls.

3.4.2 RUV-2 vs. RUV-4 vs. RUV-inv

In this section we run 12 simulations and compare the relative performance of RUV-2, RUV-4, and RUV-inv. First we discuss the details of the simulations. Then we discuss the results of one of the 12 simulations in detail. Finally we discuss briefly the results of the remaining 11 simulations.

3.4.2.1 Simulation Details

In each simulation we set $n_c = 1000$. In six of the simulations ("good controls"), $\beta_{1j} = 0$ for every control gene. In the other six simulations ("bad controls"), $\beta_{1j} = 0$ for only 900 of the 1000 control genes.

In the first four simulations $k = 20$ and

$$\sigma_\alpha = (1.1, 1.0, 0.8, 0.5, 0.4, 0.4, 0.3, 0.3, 0.2, 0.2, .16, .16, .15, .15, .14, .13, .13, .12, .12, .11). \tag{3.210}$$

This value of σ_α is similar to what we observe empirically in the gender dataset. In the first two simulations ("lightly correlated") $b_{WX} = (.1, ..., .1)$ and in the second two simulations ("moderately correlated") $b_{WX} = (.4, .4, .4, .2, ..., .2)$. Note that in the "moderately correlated" case the columns of W that are most highly correlated with X correspond to the rows of α with the largest $\sigma_{\alpha,i}$. In other words, the biggest unwanted factors are also the most correlated with X.

The next four simulations are "harder." In these simulations ("moderate decay") $k = 70$ and $\sigma_\alpha = (1, 1/2, ..., 1/70)$. Note in particular that $k = 70 > m = 50$. In two of the simulations ("lightly correlated") $b_{WX} = (.1, ..., .1)$ as before. In the other two simulations ("highly correlated") the elements of b_{WX} are chosen uniformly at random from (-1, 1). The final four simulations ("slow decay") are "harder" still. These simulations are identical to the previous four, but now $\sigma_\alpha = (1, 1/\sqrt{2}, ..., 1/\sqrt{70})$.

For all 12 simulations we generate 1000 datasets. We fit each dataset by RUV-2, RUV-4, and RUV-inv. In the case of RUV-2 and RUV-4, we fit with each value of K from 1 to 47. We also fit each dataset using a standard linear model that contained only an X term (the "unadjusted" case). All of our models include an intercept, i.e. a $Z = 1_{m\times 1}$ term. We fit all models both with and without Limma (Smyth, 2004).

For every model fit we record the following six quality metrics: (1) the fraction of the 100 genes with the largest values of β_{1j}^2/σ_j^2 that end up being ranked as one of the top 100 most significantly DE genes ("top ranked fraction"), (2) the fraction of genes with $\beta_{1j} = 0$ to have a p-value less than 0.05 ("type 1 error rate"), (3) the fraction of genes with $\beta_{1j} \neq 0$ to have a p-value less than 0.05 ("average power"), (4) the RMSE of $\hat{\beta}$ ("beta hat RMSE"), (5) the log of the mean value of $\hat{\sigma}_j^2/\sigma_j^2$ ("sigma hat scale"), (6) the IQR of $\log\left(\hat{\sigma}_j^2/\sigma_j^2\right)$ ("rescaled sigma hat IQR"). Of these six quality metrics, the first is arguably the most important. In practice, the goal of a DE study is usually to produce a list of genes that are "most interesting" and warrant further study. We will refer to the ability of a method to properly rank top genes as "discriminative power."

3.4.2.2 Results of "$k = 20$, moderately correlated, good controls"

We plotted the average value (over the 1000 datasets) of each of the six quality metrics. See Figure 3.9 and Figures B.2-B.13 in the appendix. The results for RUV-2 (brown) and RUV-4 (orange) are shown as a function of K. The results for RUV-inv (blue) and the unadjusted case (black) are shown by horizontal lines. Solid lines are for "standard" estimates of σ^2 and dashed lines are for estimates using Limma. The light dotted lines show 95% nominal confidence intervals; these are not always visible as the confidence intervals are quite small.

In this section we focus on Figure 3.9 ("$k = 20$, moderately correlated, good controls"). The first thing to notice is that RUV-inv performs very well. In terms of discriminative

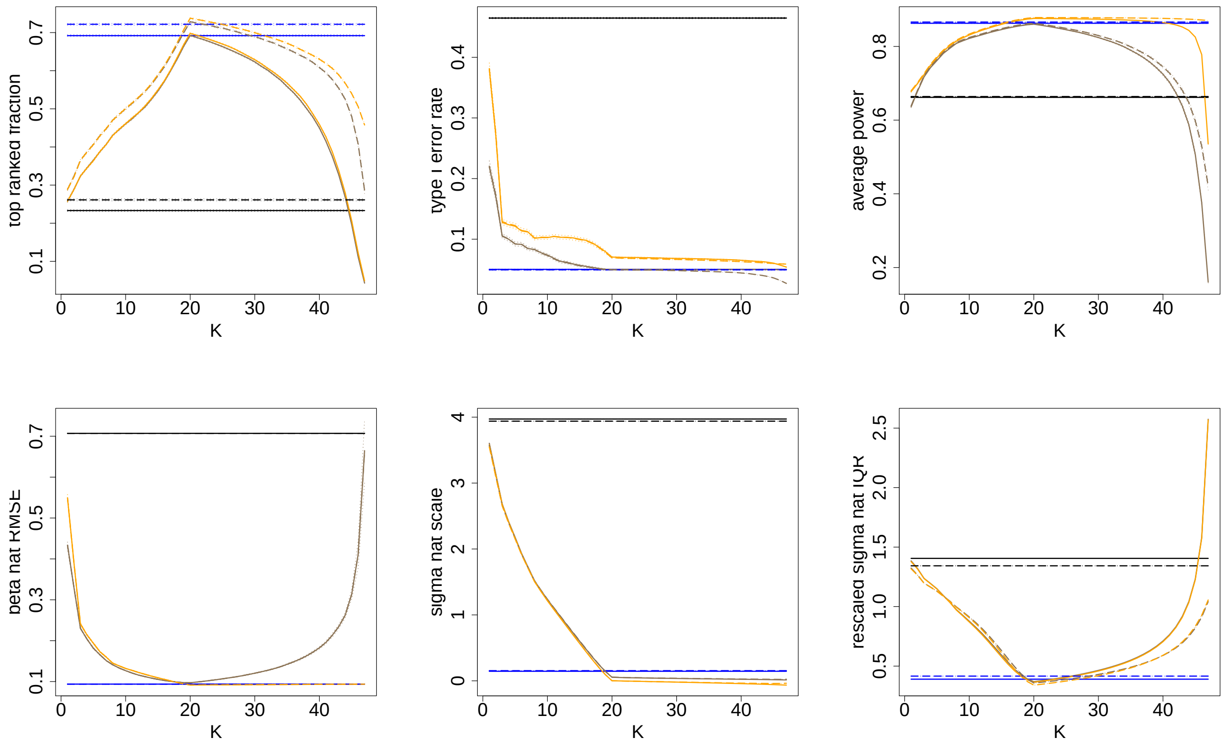


Figure 3.9: Moderately correlated, good controls.

power, RUV-inv performs about as well as RUV-2 and RUV-4 at the optimal value of K. The type 1 error rate for RUV-inv is very close to 0.05 (see also Tables B.1, B.2, B.3 in the appendix). By comparison, RUV-2 is anti-conservative when $K < 20$ and RUV-4 is anti-conservative for all K.

RUV-inv also performs well in terms of its estimation of β and σ^2. As expected, the RMSE of $\hat{\beta}$ is effectively nonincreasing in K for RUV-4 but not RUV-2. RUV-inv essentially achieves the minimum RMSE. Also as expected, both RUV-2 and RUV-4 have seriously inflated estimates of σ^2 when $K < 20$. When $K > 20$ RUV-4 has slightly deflated estimates of σ^2, but RUV-2 is nearly spot-on. RUV-inv has slightly inflated estimates of σ^2. The average value of $\hat{\sigma}_j^2/\sigma_j^2$ is about 1.16 (see also Tables B.1, B.2, B.3 in the appendix). Nonetheless, these slightly inflated estimates of σ^2 do not cause an unreasonable loss in power.

Indeed, the slightly inflated RUV-inv estimates of σ^2 are both expected and desirable. Recall that $\hat{\beta}$ is slightly biased. Recall also that σ^2 is estimated under the assumption that $\hat{\beta}^\star$ is unbiased. Therefore it is reasonable to expect that $\hat{\sigma}^2$ will be slightly inflated due to the biases of $\hat{\beta}^\star$. The end result is that we essentially fold the small biases of $\hat{\beta}^\star$ into the estimate $\hat{\sigma}^2$, and thereby keep the type 1 error rate in check. Of course, there is no guarantee that the type 1 error rate will equal 0.05 in all situations. Nonetheless, the inflated estimates of σ^2

are often a useful feature in practice. Finally, note that RUV-inv achieves a nearly optimal value of IQR $\left[\log\left(\hat{\sigma}_j^2/\sigma_j^2\right)\right]$, suggesting that RUV-inv makes good use of all the degrees of freedom that are available to estimate σ^2.

An interesting conclusion of Figure 3.9 seems to be that the primary determinant of the discriminative power is the quality of the estimate of σ^2, and not the quality of the estimate of β. We arrive at this conclusion by noting several facts. First, note that although the RMSE($\hat{\beta}$) curves for RUV-2 and RUV-4 diverge substantially for $K > 20$, the curves for the discriminative power of RUV-2 and RUV-4 without Limma (solid lines) follow each other very closely. This suggests that the quality of $\hat{\beta}$ is not the main determinant of the discriminative power. Secondly, note that the discriminative power is substantially higher when we use Limma (dashed lines). This suggests that the quality of $\hat{\sigma}^2$ is important. Moreover, the improvement in discriminative power offered by Limma is largest when K is large, where the quality of $\hat{\sigma}^2$ is poorest. Even more tellingly, the curves for the discriminative power of RUV-2 and RUV-4 *with* Limma *do* diverge for large K; once the problem of the poor estimates of σ^2 at large K has been "solved" by Limma, it becomes possible to see the difference between the performance of $\hat{\beta}^{(\mathrm{RUV-2})}$ and $\hat{\beta}^{(\mathrm{RUV-4})}$. Finally, we observe that the the "kinks" in the discriminative power curves at $K = 20$ are very similar to the kinks in the curves of the IQR of $\log\left(\hat{\sigma}_j^2/\sigma_j^2\right)$. Indeed, the entire discriminative power curve is visually similar to the IQR $\left[\log\left(\hat{\sigma}_j^2/\sigma_j^2\right)\right]$ curve, just upside down.

3.4.2.3 Results of the Remaining Simulations

Turning now to the other 11 simulations (see Figures B.2-B.13 in the appendix), we see that many, but not all, of the conclusions of Figure 3.9 are true more generally. In terms of discriminative power, RUV-4 performs almost uniformly better than RUV-2, and RUV-inv performs about as well as RUV-4 at the optimal value for K. The RMSE of $\hat{\beta}$ falls then rises again for RUV-2, but is essentially nonincreasing for RUV-4. RUV-4 is often anti-conservative, sometimes substantially. With RUV-2, the type 1 error rate is good for large K, but this is a moot point because the discriminative power of RUV-2 is poor at large K. For small K, the type 1 error rate can exhibit strange behavior in both RUV-2 and RUV-4. RUV-inv generally exhibits better control of the type 1 error rate than RUV-4, but is not perfect. RUV-inv tends to be anti-conservative when the unwanted factors are strongly correlated with factor of interest, and, to a lesser extent, when the control genes are misspecified. RUV-inv does a fairly good job of estimating σ^2 when $k = 20$. Not surprisingly, no method does a particularly good job of estimating σ^2 when $k = 70 > m$.

We also see in the other simulations the effects of misspecified control genes. To a large extent, the "bad controls" do not affect RUV-4 or RUV-inv. RUV-2, by contrast, is very sensitive to the control gene assumption. The performance of $\hat{\beta}^{(\mathrm{RUV-2})}$ deteriorates considerably when the control genes are misspecified. This has serious consequences for both the discriminative power and type 1 error rate. Moreover, the consequences are not entirely predictable. Both the discriminative power and type 1 error rate may be complicated functions of K. See, for example, Figures B.3 and B.5.

3.4.3 A Comparison of Methods

In this Section we run 24 simulations to compare the relative performance of SVA, LEAPP, ICE, RUV-4, and RUV-(r)inv and their variants (empirical controls, rescaled variances, empirical variances). First we discuss the details of the simulations. Then we discuss the results of one of the 24 simulations in detail. Finally we discuss briefly the results of the remaining 23 simulations.

3.4.3.1 Simulation Details

The simulations of this section are similar to those of Section 3.4.2. In 12 of the simulations, $k = 20$ and σ_α is set as in (3.210). In the other 12 simulations $k = 70$ and σ_α is set as in "moderate decay." In 12 of the simulations b_{WX} is set as in "lightly correlated" and in the other 12 as in "highly correlated." In all of the simulations all of the control genes are good controls, but in 12 of the simulations $n_c = 1000$ and in the other 12 simulations $n_c = 60$.

Unlike Section 3.4.2, here we also vary the sparsity of β. In eight of the simulations ("very sparse"), only 100 elements of β are non-zero. In another eight simulations ("sparse") only 500 elements of β are non-zero. In the remaining eight simulations ("not sparse") 5000 elements of β are non-zero. There are a few other minor differences as well. We generated only 100 datasets per simulation instead of 1000. We did not fit with Limma. We report the mean value of $\hat{\sigma}_j^2/\sigma_j^2$ directly (instead of the log). We report only the results of RUV-4 for $\hat{k}$ and not all values of K. In some of the methods we make use of empirical controls. We define empirical controls to be all genes whose RUV-rinv FDR-adjusted p-values is greater that 0.5.

Note that we do not report the results of RMSE($\hat{\beta}$), AVG($\hat{\sigma}_j^2/\sigma_j^2$), or IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)] for the rescaled variances or empirical variances methods, since these results are identical to those of the standard method. We also do not report AVG($\hat{\sigma}_j^2/\sigma_j^2$) or IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)] for ICE, since ICE, which is based on a random effects model, does not return an estimate of the equivalent of the σ^2 that exists in our model.

3.4.3.2 $k = 20$, moderately correlated, $n_c = 60$, sparse

As in the Section 3.4.2, we discuss the results of just one of the simulations in detail. We present the results of the other simulations in the Supplementary Material. The simulation we discuss in detail is "$k = 20$, moderately correlated, $n_c = 60$, sparse". The results of this simulation are given in Table 3.2.

First we discuss RMSE($\hat{\beta}$). All of the methods show a substantial improvement over "unadjusted," but the best performance comes from ICE and RUV-(r)inv with empirical controls. Without empirical controls, RUV-inv performs considerably worse. This is as expected, since $n_c = 60$ is only a little larger than $m = 50$, and $\hat{b}_{WX}$ suffers from over-fitting. Compared to RUV-inv without empirical controls, RUV-rinv without empirical controls performs much better — nearly as well as RUV-(r)inv with empirical controls. The ridging

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.52	(4×10^{-3})	0.47	(5×10^{-4})	0.66	(2×10^{-3})	0.707	(5×10^{-4})	53.02	(5×10^{-2})	1.41	(2×10^{-3})
SVA (IRW)	0.72	(4×10^{-3})	0.10	(5×10^{-3})	0.78	(2×10^{-3})	0.170	(3×10^{-3})	6.08	(7×10^{-2})	1.06	(3×10^{-3})
SVA (2-step)	0.72	(4×10^{-3})	0.11	(5×10^{-3})	0.80	(2×10^{-3})	0.163	(3×10^{-3})	6.03	(7×10^{-2})	1.05	(4×10^{-3})
LEAPP	0.72	(4×10^{-3})	0.26	(8×10^{-3})	0.86	(2×10^{-3})	0.133	(2×10^{-3})	5.22	(7×10^{-2})	1.06	(3×10^{-3})
ICE	0.80	(2×10^{-3})	0.00	(4×10^{-5})	0.72	(2×10^{-3})	0.091	(6×10^{-4})				
RUV-4	0.78	(1×10^{-2})	0.12	(3×10^{-3})	0.80	(7×10^{-3})	0.151	(3×10^{-3})	0.96	(1×10^{-3})	0.73	(3×10^{-2})
RUV-inv	0.72	(4×10^{-3})	0.02	(1×10^{-3})	0.62	(1×10^{-2})	0.209	(5×10^{-3})	1.57	(1×10^{-2})	1.01	(1×10^{-2})
RUV-rinv	0.82	(3×10^{-3})	0.06	(2×10^{-3})	0.84	(2×10^{-3})	0.110	(1×10^{-3})	1.98	(1×10^{-2})	0.62	(3×10^{-3})
RUV-inv (Ectl)	0.90	(2×10^{-3})	0.05	(8×10^{-4})	0.87	(2×10^{-3})	0.090	(6×10^{-4})	1.14	(5×10^{-4})	0.35	(4×10^{-4})
RUV-rinv (Ectl)	0.89	(2×10^{-3})	0.06	(1×10^{-3})	0.87	(2×10^{-3})	0.092	(7×10^{-4})	1.35	(1×10^{-3})	0.39	(5×10^{-4})
RUV-inv-rsvar (Ectl)	0.90	(2×10^{-3})	0.05	(2×10^{-3})	0.87	(2×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.89	(2×10^{-3})	0.05	(2×10^{-3})	0.86	(3×10^{-3})						
RUV-inv-evar (Ectl)	0.83	(4×10^{-3})	0.01	(2×10^{-4})	0.82	(2×10^{-3})						
RUV-rinv-evar (Ectl)	0.83	(3×10^{-3})	0.01	(3×10^{-4})	0.82	(2×10^{-3})						

Table 3.2: $k = 20$, moderately correlated, $n_c = 60$, sparse.

clearly helps. The remaining methods, all of which rely on an estimate of k, perform moderately well.

Next we discuss $\hat{\sigma}^2$. All of the methods show improvement over "unadjusted." The best overall performance comes from RUV-inv with empirical controls. RUV-rinv with empirical controls also performs quite well. RUV-4 performs exceptionally well in terms of AVG($\hat{\sigma}_j^2/\sigma_j^2$), but less so in terms of IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]; RUV-4 gets the overall "scale" of σ^2 right, but does not do as good of a job at estimating the individual σ_j^2. Both RUV-inv and RUV-rinv without empirical controls suffer from the fact that n_c is only 60, but in different ways. RUV-rinv performs worse than RUV-inv in terms of AVG($\hat{\sigma}_j^2/\sigma_j^2$) but better in terms of IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]. The remaining methods, which rely on an estimate of k, do not perform as well as the inverse method and its variants.

Now we discuss the discriminative power. Given our discussions of RMSE($\hat{\beta}$) and $\hat{\sigma}^2$, our findings regarding discriminative power are no surprise. All of the methods offer an improvement over "unadjusted." RUV-4 performs moderately well. RUV-inv suffers from the fact that n_c is small and performs worse than RUV-4. RUV-rinv overcomes the problem of a small n_c and outperforms both RUV-4 and RUV-inv. However, RUV-(r)inv with empirical controls performs the best. Empirical controls are the best way to handle the small n_c. Finally, note that out of SVA, LEAPP, and ICE, ICE performs the best.

There is considerable variation between the methods in terms of their control of the type 1 error rate. SVA, LEAPP, and RUV-4 are all notably anti-conservative. ICE is excessively conservative. RUV-inv is also too conservative. RUV-rinv, and both RUV-inv and RUV-rinv with empirical controls, demonstrate quite good control of the type 1 error rate. Using rescaled variances in this case works, but is not needed. On the other hand, using empirical variances actually makes things worse. The problem is that even though only 500 of the 10,000 elements of β are non-zero, this is still not sparse enough for the method of empirical variances. As a result, both the type 1 error rate and the discriminative power are adversely affected.

3.4.3.3 Results of the Remaining Simulations

We turn now to the other simulations. We begin with a comparison of RUV-4, RUV-inv, and RUV-rinv. The relative performance of these methods depends on whether $n_c = 1000$ or $n_c = 60$. When $n_c = 1000$, RUV-inv performs best. When $n_c = 60$, RUV-rinv performs best.

When $n_c = 1000$, all three methods generally perform well in terms of discriminative power. RUV-inv performs the best. RUV-rinv is a close second. RUV-4 performs a little worse, particularly when X is highly correlated with W. The differences between the methods are more pronounced in terms of the type 1 error rate. In terms of the type 1 error rate, RUV-inv clearly performs the best. RUV-inv exhibits good control of the type 1 error rate in most cases, but is notably anti-conservative when $k = 70$ and X is highly correlated with W. RUV-rinv is more anti-conservative than RUV-inv in all cases, and RUV-4 is even more anti-conservative than RUV-rinv. The differences between RUV-inv, RUV-rinv, and RUV-4 are most pronounced when $k = 70$ or when X is highly correlated with W.

When $n_c = 60$, the story is much different. RUV-rinv performs the best by far, and RUV-inv performs the worst. RUV-inv is overly conservative and exhibits poor discriminative power. RUV-rinv is anti-conservative, but exhibits far more discriminative power. RUV-4 is more anti-conservative than RUV-rinv. The discriminative power of RUV-4 is generally somewhere between that of RUV-inv and RUV-rinv.

We now consider the use of empirical controls. The empirical controls work as expected. Performance of RUV-(r)inv with empirical controls is roughly the same as the performance of RUV-(r)inv without empirical controls but with $n_c = 1000$. RUV-inv does slightly better than RUV-rinv. Importantly, note that the performance of the empirical controls is essentially the same whether the initial n_c is equal to 1000 or 60. In other words, if we begin with just 60 control genes, use RUV-rinv to generate empirical controls, and then apply RUV-(r)inv, we get results just as good as if we had 1000 control genes to begin with.

Next we consider the use of rescaled and empirical variances. The rescaled variances work as expected. The use of rescaled variances does not affect discriminative power in any way. The type 1 error rate is much better with rescaled variances than without. Across all 24 simulations, the average type 1 error rate with rescaled variances is never less than 0.04 nor greater than 0.08. Without rescaled variances, the range is 0.05 to 0.24. The empirical variances also work as expected. In all 8 “very sparse” simulations, the average type 1 error rate with empirical variances is 0.05. The discriminative power with empirical variances is equal to or slightly better than the discriminative power without empirical variances. However, the usefulness of the empirical variances is limited to the case that β is very sparse. In the 16 other simulations (“sparse” and “not sparse”) the use of empirical controls leads to a serious decrease in performance.

Finally, we consider the performance of SVA, LEAPP, and ICE. Of these methods, ICE performs the best in all cases except those in which $k = 70$ and β is not sparse. Note that ICE tends to be extremely conservative, while SVA and LEAPP tend to be extremely anti-conservative. Finally, note that RUV-inv with empirical controls and rescaled variances performs at least well as any other method. When β is very sparse, ICE performs as well as RUV-inv with empirical controls and rescaled variances, but only in terms of discriminative power (not type 1 error rate). SVA and LEAPP also perform reasonably well when β is very sparse.

3.5 Data Results

In this section we apply the RUV methods to the datasets of Section 3.2. We analyze all 11 datasets the same way. In Section 3.5.1 we describe the details of the analyses. In Sections 3.5.2 and 3.5.3 we describe two types of plots we use to visualize the results of our analyses. In Section 3.5.4 we discuss the results of our analyses.

3.5.1 Analysis Details

Let κ, κ_1, and κ_2 be index variables that range over the symbols $\{0, 1, 2, ..., m-2, \mathrm{k}, \mathrm{i}, \mathrm{r}\}$. Note that m, in italics, is a variable that stands for the number of arrays; k, i, and r are not variables, but just letters. Define estimates $\hat{\beta}^{(\kappa)}$ as follows: when $\kappa = 0$, $\hat{\beta}^{(\kappa)}$ is the OLS estimate of β in a regression of Y on X; when $\kappa \in \{1, ..., m-2\}$, $\hat{\beta}^{(\kappa)}$ is the RUV-4 estimate of β with $K = \kappa$; when $\kappa = \mathrm{k}$, $\hat{\beta}^{(\kappa)}$ is the RUV-4 estimate of β with $K = \hat{k}$; when $\kappa = \mathrm{i}$, $\hat{\beta}^{(\kappa)}$ is the RUV-inv estimate of β; when $\kappa = \mathrm{r}$, $\hat{\beta}^{(\kappa)}$ is the RUV-rinv estimate of β. Define $(\hat{\sigma}^2)^{(\kappa)}$, $\hat{W}^{(\kappa)}$, $\hat{b}_{WX}^{(\kappa)}$, etc. similarly; note that $\hat{W}^{(\kappa)}$ and $\hat{b}_{WX}^{(\kappa)}$ are undefined when $\kappa = 0$.

Let $v_j^{(\mathrm{s},\kappa_1,\kappa_2)}$ denote the "standard" estimate of the variance of $\hat{\beta}_j^{(\kappa_1)}$, given an estimate $(\hat{\sigma}_j^2)^{(\kappa_2)}$ of σ_j^2. For example,

$$v_j^{(\mathrm{s},2,r)} = \left(\hat{\sigma}_j^2\right)^{(\mathrm{r})} \left[1 + \left(\hat{b}_{WX}^{(2)}\right)' \hat{b}_{WX}^{(2)}\right] \tag{3.211}$$

and

$$v_j^{(\mathrm{s},i,4)} = \left(\hat{\sigma}_j^2\right)^{(4)} \left[1 + \left(\hat{b}_{WX}^{(\mathrm{i})}\right)' \hat{b}_{WX}^{(\mathrm{i})}\right]. \tag{3.212}$$

Let $v_j^{(\mathrm{e},\kappa_1,\kappa_2)}$ denote the empirical estimate of the variance of $\hat{\beta}_j^{(\kappa_1)}$, given estimates $(\hat{\sigma}_j^2)^{(\kappa_2)}$ of σ_j^2.

Define the t statistic

$$t_j^{(\mathrm{s},\kappa_1,\kappa_2)} \equiv \frac{\hat{\beta}_j^{(\kappa_1)}}{\sqrt{v_j^{(\mathrm{s},\kappa_1,\kappa_2)}}} \tag{3.213}$$

and define $t_j^{(\mathrm{e},\kappa_1,\kappa_2)}$ similarly. Define the p-value

$$p_j^{(\mathrm{s},\kappa_1,\kappa_2)} \equiv \mathbb{P}\left[|t| > \left|t_j^{(\mathrm{s},\kappa_1,\kappa_2)}\right| \middle| t_j^{(\mathrm{s},\kappa_1,\kappa_2)}\right] \tag{3.214}$$

where t follows a t distribution with an appropriate number of degrees of freedom (e.g. $m-1$ degrees of freedom if $\kappa_2 = 0$, $m-6$ degrees of freedom if $\kappa_2 = 5$, or $\hat{r}$ degrees of freedom if $\kappa_2 = \mathrm{i}$). Define $p_j^{(\mathrm{e},\kappa_1,\kappa_2)}$ similarly.

For fixed values of κ_1 and κ_2, consider the $N \leq n$ genes with the N smallest p-values $p_j^{(\mathrm{s},\kappa_1,\kappa_2)}$. Let $C^{(\mathrm{s},\kappa_1,\kappa_2,N)}$ denote the number of these top-ranked genes that are located on the X or Y chromosomes (we choose "C" for "top rank Count"). $C^{(\mathrm{s},\kappa_1,\kappa_2,N)}$ is one of the most important statistics we use to compare the effectiveness of the various methods. Next define the statistic

$$T^{(\mathrm{s},\kappa_1,\kappa_2)} \quad \equiv \quad \log\left(\frac{\mathrm{median}_j \left|t_j^{(\mathrm{s},\kappa_1,\kappa_2)}\right|}{T_0}\right). \tag{3.215}$$

Here T_0 is the 50th percentile of $|t|$, where t follows a t distribution with an appropriate number of degrees of freedom. Assuming β is sparse, $T^{(\mathrm{s},\kappa_1,\kappa_2)}$ is a good measure of whether the t-statistics $t_j^{(\mathrm{s},\kappa_1,\kappa_2)}$ are too big or too small. Now, consider all genes that do not come from the X or Y chromosomes. Let $E^{(\mathrm{s},\kappa_1,\kappa_2)}$ denote the fraction of these genes whose p-value $p_j^{(\mathrm{s},\kappa_1,\kappa_2)}$ is less than 0.05. We use $E^{(\mathrm{s},\kappa_1,\kappa_2)}$ as an effective type 1 error rate.[3] Finally, define $C^{(\mathrm{e},\kappa_1,\kappa_2,N)}$, $T^{(\mathrm{e},\kappa_1,\kappa_2)}$, and $E^{(\mathrm{e},\kappa_1,\kappa_2)}$ similarly to $C^{(\mathrm{s},\kappa_1,\kappa_2,N)}$, $T^{(\mathrm{s},\kappa_1,\kappa_2)}$, and $E^{(\mathrm{s},\kappa_1,\kappa_2)}$.

Our analyses proceed as follows. First we define three sets of control genes. The first set is the set of housekeeping genes. The second set includes all genes. The third set is a set of empirical controls. Then, for each set of control genes, for each possible pair (κ_1, κ_2), and for each value of N in $\{20, 40, 60, 80, 100\}$, we calculate $C^{(\mathrm{s},\kappa_1,\kappa_2,N)}$, $T^{(\mathrm{s},\kappa_1,\kappa_2)}$, $C^{(\mathrm{e},\kappa_1,\kappa_2,N)}$, and $T^{(\mathrm{e},\kappa_1,\kappa_2)}$. We also calculate $E^{(\mathrm{s},\kappa,\kappa)}$ and $E^{(\mathrm{e},\kappa,\kappa)}$ for $\kappa \in \{\mathrm{k}, \mathrm{i}, \mathrm{r}\}$.

Some notes: All of our models include a $Z = 1_{m\times 1}$ term. This is not reflected in the notation above. All of our estimates of σ^2 are unadjusted; we do not use Limma. To generate the empirical controls, we simply regress Y on X, compute FDR-adjusted p-values, and designate all genes with FDR-adjusted p-values greater than 0.5 as empirical controls. This is a very crude application of the strategy of empirical controls. The rationale for this choice of empirical controls is to demonstrate that even a crude application of the strategy will often be effective. Still, we encourage researchers to be cautious in their own applications of the strategy.

In addition to the analysis just described, we also analyze each dataset by SVA, LEAPP, and ICE. For each of these methods, we rank genes by p-value, and count the number of genes in the top 20 / 40 / 60 / 80 / 100 that are on the X or Y chromosomes. We also calculate an "effective type 1 error rate" analogous to the one described above.

[3] It is worth noting that in the course of our analyses we have been unable to produce any convincing evidence that there are any autosomal genes that are differentially expressed between the brains of men and women. To be sure, in each of the datasets we examine, there are a few autosomal genes with small FDR-adjusted p-values. However, none of these genes are consistently "significant" across multiple datasets.

3.5.2 Summary Plots

We need a way to visualize our results. We accomplish this with “summary plots.” An example summary plot is given in Figure 3.10. Summary plots are rather complex. The purpose of this section is to describe them.

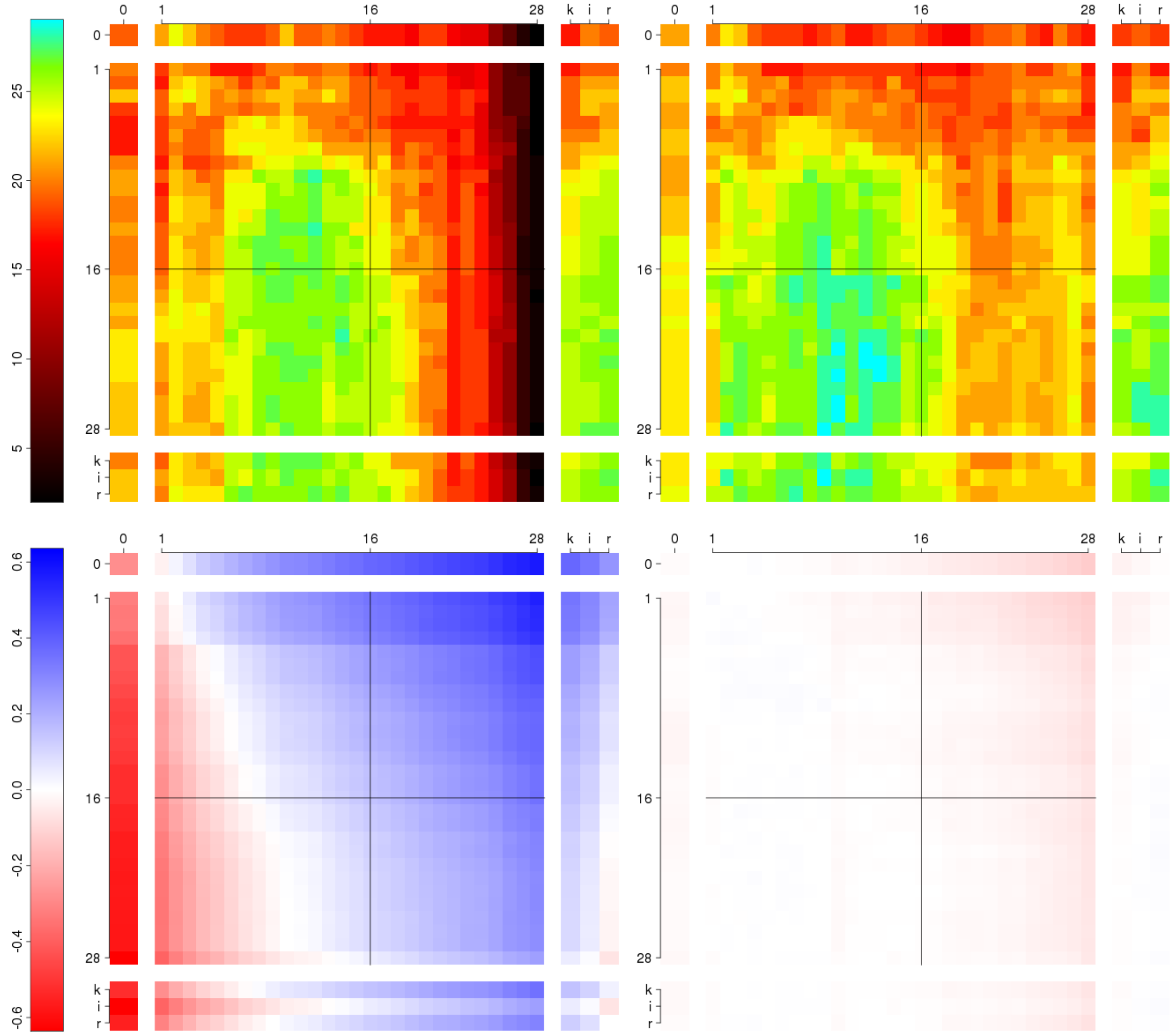


Figure 3.10: Example Summary Plots. Alzheimer’s (Preprocessed) dataset, HK controls. X/Y gene counts are out of the top 40 genes.

Ignoring the color scales on the far left, each summary plot can be divided into 4 identically “shaped” divisions. On the top are two divisions with colors ranging from black to red to green to blue. On the bottom are two divisions with colors ranging from red to white to blue. Each division can be further divided into nine subdivisions. These subdivisions do not all have the same shape. The top left subdivision is a single colored square; the middle subdivision is a giant multi-colored square with many rows and columns; etc.

We now describe the top left division. The top left division is a plot of $C^{(\mathrm{s},\kappa_1,\kappa_2,N)}$ for all values of (κ_1, κ_2). Each row of the plot represents a different value of κ_1; each column of the plot represents a different value of κ_2. The value of $C^{(\mathrm{s},\kappa_1,\kappa_2,N)}$ is given by the color. An example: The top left corner of the bottom right subdivision is a light greenish yellow color. Referring to the color scale on the far left, we see that this color corresponds to a value of 24. Thus $C^{(\mathrm{s,k,k},40)} = 24$. In other words, if we run RUV-4 with $K = \hat{k}$ and rank genes by p-value, we will find that 24 of the top 40 genes are on the X or Y chromosomes. A second example: In the middle column of the middle-right subdivision, the 11th square from the top is a light green color. Referring to the color scale on the far left, we see that this color corresponds to a value of 25. Thus $C^{(\mathrm{s},11,\mathrm{i},40)} = 25$. In other words, if we run RUV-4 with $K = 11$ to get our estimate of β, but estimate σ^2 using RUV-inv, we will find that 25 of the resulting top 40 genes are on the X or Y chromosomes. Note the black lines in the 16th row and 16th column of the middle subdivision. These black lines represent $\hat{k}$; $\hat{k}$ in this example is equal to 16. Note that the color at the intersection of these lines is the same as the color in the top left square of the bottom right subdivision.

The other divisions are analogous to the top left division. The top right division is a plot of $C^{(\mathrm{e},\kappa_1,\kappa_2,N)}$ instead of $C^{(\mathrm{s},\kappa_1,\kappa_2,N)}$. The bottom left division is a plot of $T^{(\mathrm{s},\kappa_1,\kappa_2)}$. The bottom right division is a plot of $T^{(\mathrm{e},\kappa_1,\kappa_2)}$. Note that in the case of the bottom divisions, shades of red correspond to t-values that are generally “too small.” p-vales are therefore “too big” and the method is conservative. Conversely, shades of blue correspond to t-values that are generally “too big,” p-vales that are “too small,” and the method is anti-conservative. White is just right.

We have described the mechanics of reading the summary plots. We now give a few brief examples of how the summary plots can be used to learn something. First note that the upper right division is “greener” than the upper left division. From this we learn that using empirical variances increases the discriminative power (on this dataset). Next observe that the lower right division is “whiter” than the lower left division. From this we learn that using empirical variances leads to better control of the type 1 error rate. Now consider just the middle subdivision of the upper left division. Note that as we move downwards through the rows, the rows are roughly non-decreasing in green-ness. This tells us that the quality of $\hat{\beta}$ as an estimator of β is roughly non-decreasing in K. Conversely, moving from left to right, the columns first get greener, and then fall to black. This tells us that $\hat{\sigma}^2$ is only a good estimator of σ^2 when K is somewhere between 6 and 17.

3.5.3 Projection Plots

In Sections 3.3.4 and 3.3.8 we considered examples in which $m = 2$. We were able to plot each gene as a point in 2-dimensional space. These plots were very helpful for visualizing and understanding RUV-4. We would like to produce similar plots for $m > 2$. Such plots would be a very useful diagnostic tool when applying RUV-4 to real data.

When $m = 2$ and $K = 1$ the column space of $\hat{W}$ can be represented as a line passing through origin. The slope of the line is $\hat{b}_{WX}$. When $m > 2$ and $k > 1$ this is no longer possible. The column space of $\hat{W}$ is a hyperplane and cannot be graphed in two dimensions. To solve this problem, we project the data into a 2-dimensional space.

Assume throughout this section that we use the SVD as our method of factor analysis. In particular, assume that the columns of $\hat{W}_0$ are orthonormal. Now let

$$\tilde{P} \equiv \left(\frac{\hat{W}_0 \hat{b}'_{WX}}{\left|\left|\hat{W}_0 \hat{b}'_{WX}\right|\right|} \middle| X \right)' \tag{3.216}$$

$$= \left(\frac{\hat{W}_0 \hat{b}'_{WX}}{\left|\left|\hat{b}'_{WX}\right|\right|} \middle| X \right)' \tag{3.217}$$

and let

$$\tilde{Y} \equiv \tilde{P} Y. \tag{3.218}$$

Note that

$$\hat{\beta}_j = X' Y_{\star j} - \hat{b}_{WX} \hat{W}'_0 Y_{\star j} \tag{3.219}$$

$$= \tilde{Y}_{2,j} - \left|\left|\hat{b}'_{WX}\right|\right| \tilde{Y}_{1,j}. \tag{3.220}$$

If we plot the points $(\tilde{Y}_{1,j}, \tilde{Y}_{2,j})$ on standard coordinate axes, $\hat{\beta}_j$ is the vertical distance between the point $(\tilde{Y}_{1,j}, \tilde{Y}_{2,j})$ and the line passing through the origin with slope $\left|\left|\hat{b}'_{WX}\right|\right|$. We name such plots projection plots. Examples are given in Figure 3.11.

What is the interpretation of a projection plot? Consider RUV-4 in the context of the functional approach.

$$\hat{\beta}_j = y_j - \hat{f}(x_{\star j}) \tag{3.221}$$

where

$$x = \hat{W}'_0 Y \tag{3.222}$$

and

$$\hat{f}(u) = \hat{b}_{WX} u. \tag{3.223}$$

The gradient of the function $\hat{f}$ is $\hat{b}'_{WX}$. Any component of u orthogonal to $\hat{b}'_{WX}$ plays no role in the determination of $\hat{f}(u)$. We name

$$\frac{\hat{W}_0 \hat{b}'_{WX}}{\left|\left|\hat{b}'_{WX}\right|\right|}$$

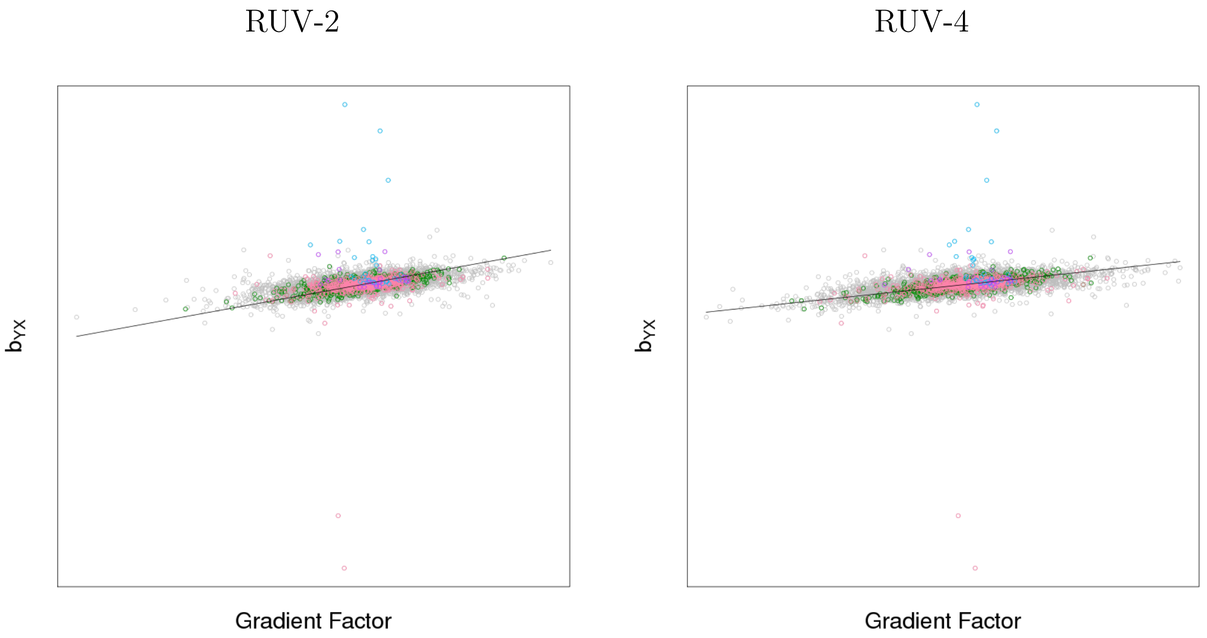


Figure 3.11: Projection Plots for RUV-2 and RUV4. Alzheimer's dataset. HK controls. Coloring scheme: negative controls, green; X-genes, pink; Y-genes, blue; X/Y-genes, purple; all other genes, gray. Note that we plot the gray dots first, followed by the green, the pink, the blue, and the purple. Thus many of the green and gray points are hidden behind the pink.

the "gradient factor" of $\hat{W}_0$. The components of $Y_{\star j}$ orthogonal to both X and the gradient factor play no role in the determination of $\hat{\beta}_j$. The 2-dimensional subspace spanned by X and the gradient factor is all that matters. This is the subspace that we plot.

3.5.4 Results

We now discuss the results of our analyses. A complete set of summary plots is provided in Section B.3 of the appendix. Section B.3 also provides projection plots for RUV-2, RUV-4, RUV-inv, and RUV-rinv. Section B.4 provides a complete set of tables listing the values of top-ranked gender gene counts and effective type 1 error rates for SVA, LEAPP, ICE, RUV-4, RUV-inv, RUV-rinv.

3.5.4.1 The Practical Assumptions

We begin our discussion by inspecting the projection plots. In Section 3.3.8 we noted that the success of RUV-4 depends critically on two "practical assumptions." The first is that the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ pairs are well described by a linear function passing through the origin. The second is that the control genes are "representative" of the other genes. We would like to check that these assumptions are plausible.

Now, $B_j(\alpha)$ is unobservable. However, in our model, $y_j = \beta_j + B(\alpha_{\star j}) + \xi$. Thus, if $\beta_j = 0$ it follows that $y_j \approx B(\alpha_{\star j})$. Moreover, we do not expect gender to affect the expression levels of more than a handful of genes, and we therefore expect that $\beta_j = 0$ for the vast majority of the genes; we believe that β is very sparse. Thus, if it is true that the $(\hat{\alpha}_{\star j}, B_j(\alpha))$ pairs are well described by a linear function passing through the origin, it should also be the case that the vast majority of the $(\hat{\alpha}_{\star j}, y_j)$ pairs are well described by a linear function passing through the origin. An examination of the projection plots suggests that this is indeed the case. See, for example, Figures 3.11 and 3.12.

An examination of the projection plots also suggests that the control genes are "representative" of the rest of the genes. See, for example, Figure 3.11. The green dots are more or less representative of the gray dots. The housekeeping genes in the TCGA datasets appear to be an exception. See, for example, the top left plot of Figure 3.12. However, the problem evident in Figure 3.12 is less an issue of the housekeeping genes being "unrepresentative" than it is an issue of RUV-inv overfitting to the housekeeping genes. The problem goes away when we use RUV-rinv instead.

3.5.4.2 RUV-2 vs RUV-4

We very briefly compare the performance of RUV-2 to that of RUV-4. Our analysis of RUV-2 is limited to projection plots. In many cases RUV-2 appears to perform fairly well. See, for example the RUV-2 projection plots of the TCGA datasets with housekeeping or empirical controls (Section B.3 of the appendix). In many other cases, however, RUV-2 suffers from the problems outlined in Section 3.3.4. See, for example, Figure 3.11; RUV-4 is clearly preferable.

Many of the examples in Section B.3 of the appendix are far more dramatic than Figure 3.11. This is particularly true when the control genes are misspecified. In the case of the Alzheimer's and Gender datasets, RUV-4 performs just fine even when all genes are used as control genes. RUV-2, however, performs horribly. In the case of the TCGA data, both RUV-2 and RUV-4 are adversely affected by misspecification of the control genes. However, RUV-2 performs far worse.

Finally, note that the comparison here between RUV-2 and RUV-4 is not entirely fair. RUV-2 is more sensitive to the choice of K than RUV-4. In Chapter 2 we emphasize the importance of exercising judgment when selecting K, and using quality measures such as RLE plots, p-value histograms, and gene rankings to guide the choice of K. We did not do that here. Had we been more careful in our selection of K, the performance of RUV-2 may have been considerably better.

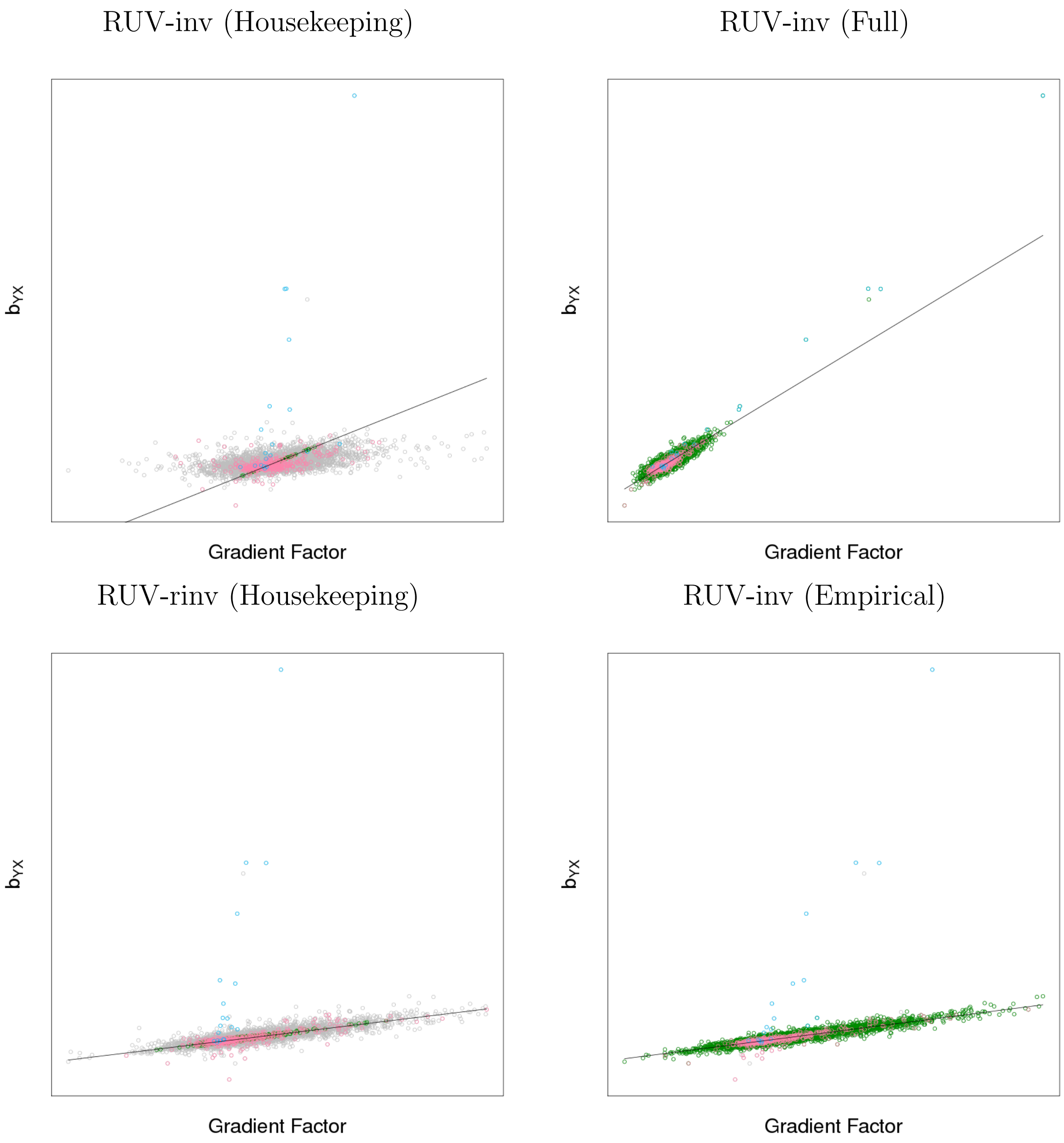


Figure 3.12: Projection Plots of TCGA HG-U133A.

3.5.4.3 Choice of K

We now examine the choice of K on the performance of RUV-4. Our first observation is that, in many cases, setting K to be very large does not notably hurt the performance of $\hat{\beta}$. See, for example, Figure 3.10. However, there are exceptions. One exception is when

K is large relative to n_c. This occurs in the TCGA examples when we use housekeeping genes as controls. See Figures B.18, B.19, and B.20 in the appendix. The summary plots show that the quality of $\hat{\beta}$ decreases for large K; the projection plots (RUV-inv) show that the reason is overfitting to the control genes. A second exception is when the control genes are misspecified. Misspecification of the control genes is not necessarily a problem in and of itself, but becomes a problem when K is very large. This can be seen in the TCGA examples. When we use all genes as control genes, the quality of $\hat{\beta}$ is poor when K is greater than 100 or so. See Figures B.18, B.19, and B.20 in the appendix. The summary plots show that the quality of $\hat{\beta}$ decreases for large K; the projection plots (RUV-4 and especially RUV-inv) show that the reason is misspecification of the control genes.

Our second observation is that $\hat{\sigma}^2$ performs poorly both when K is too small and when K is too large. See, for example, Figure 3.10. The discriminative power is poor both when K is small and when K is large. Moreover, the plot of $T^{(\mathrm{s},\kappa_1,\kappa_2)}$ suggests that $\hat{\sigma}^2$ is generally too large when K is small and $\hat{\sigma}^2$ is too small when K is large. However, there is good news as well. In many cases, the discriminative power is only hurt by a poor estimate of $\hat{\sigma}^2$ when the value of K is relatively extreme. See, for example, Figures B.18, B.19, and B.20 in the appendix. As long as K is not so small that $\hat{\sigma}^2$ is severely biased by unwanted variation that has not been properly adjusted for, and as long as K is not so large that $\hat{\sigma}^2$ must be estimated using only a few degrees of freedom, $\hat{\sigma}^2$ is "good enough" from the point of view of discriminative power. Of course, the overall scale of $\hat{\sigma}^2$ (i.e. $\dot{\sigma}^2$) remains an issue. The plots of $T^{(\mathrm{s},\kappa_1,\kappa_2)}$ suggest that $\dot{\sigma}^2$ — and thus the type 1 error rate — is fairly sensitive to the choice of K. Fortunately, we can solve this problem by using empirical variances.

Our final observation is that $\hat{k}$ is a decent, but not great, choice of K. Although $\hat{k}$ may be informative, we do not advise relying solely on $\hat{k}$ when selecting K in practice. Gene rankings, p-value histograms, etc. should be considered as well. Perhaps more importantly, we observe that in most of the examples, there is no single "best" choice for K. A value of K that is good for $\hat{\beta}$ is not necessarily good for $\hat{\sigma}^2$, and vice versa. It may be better to select two values of K; one value, K_1, could be used when estimating β, and another, K_2, could be used when estimating $\hat{\sigma}^2$. Note, however, that even if we allow ourselves to use two separate Ks, it is still not necessarily the case that a choice of (K_1, K_2) that provides good discriminative power will also provide a good type 1 error rate. The problem of selecting K is indeed very difficult.

3.5.4.4 Choice of Control Genes

We now consider the choice of control genes. As we noted in the previous section, all three sets of control genes work fairly well when m (and therefore K) is relatively small. When K is fairly small, there is no problem of overfitting to the control genes, nor are there any problems due to misspecification of the control genes. Indeed, all three sets of control genes work fairly well for the Alzheimer's and Gender datasets.

The story is very different with the TCGA datasets. With housekeeping genes, n_c is too small. This is not necessarily a problem for all methods. RUV-rinv continues to perform well. RUV-4 with $K = \hat{k}$ performs moderately well. RUV-inv, however, overfits to the control genes and performs horribly. Both $\hat{\beta}$ and $\hat{\sigma}^2$ are adversely affected. With all genes as control genes the situation is even worse. RUV-inv continues to perform horribly. RUV-4 and RUV-rinv now perform poorly as well. Unlike with the housekeeping genes, however, only $\hat{\beta}$ is adversely affected. See the summary plots for evidence. With the empirical controls, the situation is much better. With empirical controls, RUV-4, RUV-inv, and RUV-rinv all perform very well. Moreover, RUV-4 performs well for a very wide range of K. In the examples of this thesis, empirical controls are an unequivocal success.

3.5.4.5 Use of Empirical Variances

We now consider the use of empirical variances. In the simulations of Section 3.4.3 we found that empirical variances are only effective when β is very sparse. Fortunately, we believe that β is in fact very sparse in the examples of this section. Indeed, we find empirical variances to be very helpful. Comparing plots of $T^{(\mathrm{s},\kappa_1,\kappa_2)}$ to plots of $T^{(\mathrm{e},\kappa_1,\kappa_2)}$ suggests that the use of empirical variances helps control the type 1 error rate. The tables in Section B.4 in the appendix confirm this directly.

The benefits of using empirical variances are not limited to better control of the type 1 error rate. In the case of the Alzheimer's dataset, the use of empirical variances also increases discriminative power. Presumably, the reason the use of empirical variances improves discriminative power in the Alzheimer's dataset is that m is fairly small, and estimates of σ^2 are therefore somewhat noisy. The method of empirical variances shrinks estimates of the variances to the mean. In this sense, the method of empirical variances plays a role similar to the role more commonly played by Limma. Note that in the other datasets, the use of empirical variances neither notably increases nor notably decreases discriminative power. The use of empirical variances seems suitable for general use whenever β is known to be very sparse.

3.5.4.6 A Comparison of Methods

We now compare the performance of SVA, LEAPP, ICE, and the variants of RUV. We begin with the variants of RUV. A quick glance at the summary plots of Section B.3 in the appendix confirms that RUV-inv works largely as intended. By setting $K = m - 1$ and estimating σ^2 with the inverse method we avoid the problem of estimating k but get results nearly as good as if we had chosen an optimal value of K. The usual caveats apply — RUV-inv will fail if m is large and n_c is small or the control genes are misspecified. However, these issues can be overcome by using either RUV-rinv, empirical controls, or both. Indeed, somewhat closer inspection reveals that RUV-rinv is generally preferable to RUV-inv. There are several examples in which RUV-rinv clearly outperforms RUV-inv. However, there are no examples in which RUV-inv substantially outperforms RUV-rinv. Thus, when in doubt, we find it is generally advisable to use RUV-rinv. We also advise using empirical controls whenever feasible, and empirical variances whenever β is known to be very sparse.

We now consider the results of SVA, LEAPP, and ICE as well. In the simulations of Section 3.4.3 we found that all of these methods perform reasonably well when β is sparse and X is not strongly correlated with W. See Tables B.1 and B.13 in the appendix. In our example datasets, β is sparse and X does not appear to be strongly correlated with W (see, for example, the projection plots; the slopes are not steep). We therefore expect all of the methods to do reasonably well. Indeed they do. Tables 3.3–3.5 provide a brief summary of results for SVA, LEAPP, ICE, RUV-4, RUV-inv, RUV-rinv. Section B.4 in the appendix provides a more complete set of results. As we see in Tables 3.3–3.5, the performance of the RUV methods compares well with that of SVA, LEAPP, and ICE.

Particularly encouraging is the ability of these methods to effectively combine the TCGA datasets. Consider the TCGA Combined dataset. Without any adjustment, results are poor. Only 16 of the top 60 genes are on the X or Y chromosome. Now consider any of the three datasets that make up the TCGA Combined dataset (i.e. the three "subset" datasets). In each case, without any adjustment, 18 of the top 60 genes are on the X or Y chromosome. Thus, despite the fact we triple our sample size, combining these three datasets into one without performing any adjustment actually hurts performance. Now suppose we adjust. If we stick to a single "subset" dataset, we might find up to 23 genes out of the top 60 are on the X or Y chromosome. However, if we combine the three subset datasets to make the Combined dataset, we might find up to 27 genes out of the top 60 that are on the X or Y chromosome. Thus, when we adjust, it is no longer the case that combining datasets hurts us; it now helps us.

Finally, we draw attention to the special cases of the Alzheimer's and Gender datasets without preprocessing. Without preprocessing, these datasets are extremely noisy. Nonetheless, the RUV methods perform exceedingly well. Indeed, the results without preprocessing are about as good as the results with preprocessing. This may not be enormously helpful to the world of microarrays — microarray data is routinely preprocessed. Nonetheless, we find these results very encouraging. We feel these results suggest that the RUV methods are relatively robust. We are therefore hopeful that the basic RUV methodology will prove useful to many different types of high dimensional data.

Alzheimer's (Preprocessed)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	15	19	23	25	26	0.02
SVA-IRW	18	19	21	24	24	0.04
SVA-TS	16	18	19	19	21	0.09
LEAPP	16	24	24	26	27	0.13
ICE	20	27	29	31	31	0.04
RUV-4 (HK)	18	24	27	29	31	0.1
RUV-rinv (HK)	20	26	30	32	33	0.05
RUV-rinv-ev (E)	20	26	29	32	34	0.06

Alzheimer's (Not Preprocessed)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	8	9	9	13	15	0.3
SVA-IRW	8	9	12	13	14	0.26
SVA-TS	NA	NA	NA	NA	NA	NA
LEAPP	17	23	24	26	26	0.13
ICE	13	16	17	17	21	0.23
RUV-4 (HK)	14	19	23	24	26	0.1
RUV-rinv (HK)	18	21	26	28	31	0.05
RUV-rinv-ev (E)	18	25	28	28	30	0.06

Gender (Preprocessed)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	11	13	15	17	19	0.01
SVA-IRW	14	17	19	19	19	0.08
SVA-TS	16	21	23	25	27	0.09
LEAPP	18	20	22	25	26	0.12
ICE	16	23	26	27	28	0.04
RUV-4 (HK)	14	19	21	24	28	0.1
RUV-rinv (HK)	16	20	22	25	28	0.08
RUV-rinv-ev (E)	15	22	26	27	28	0.06

Gender (Not Preprocessed)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	7	7	7	8	10	0
SVA-IRW	4	6	8	9	12	0
SVA-TS	NA	NA	NA	NA	NA	NA
LEAPP	11	16	18	19	19	0.01
ICE	8	11	13	14	17	0
RUV-4 (HK)	13	20	22	26	29	0.12
RUV-rinv (HK)	14	22	24	25	28	0.08
RUV-rinv-ev (E)	17	24	26	30	30	0.06

Table 3.3: Comparison of the number of top-ranked X/Y genes and the effective type 1 error rates for SVA, LEAPP, ICE, and RUV. Note that for two datasets (Alzheimer's and Gender without preprocessing) the two-step variant of SVA exits with an error.

TCGA (Exon)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	17	30	33	35	35	0.08
SVA-IRW	17	31	32	33	34	0.12
SVA-TS	17	33	34	37	40	0.08
LEAPP	17	33	34	35	36	0.12
ICE	17	33	35	35	36	0.01
RUV-4 (HK)	17	33	34	38	39	0.13
RUV-rinv (HK)	17	34	37	39	40	0.07
RUV-rinv-ev (E)	17	33	36	39	41	0.05

TCGA (U133A)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	16	22	22	24	25	0.12
SVA-IRW	17	24	25	26	26	0.04
SVA-TS	17	27	31	31	32	0.08
LEAPP	17	29	32	32	34	0.11
ICE	17	28	31	32	32	0.02
RUV-4 (HK)	17	24	26	29	32	0.14
RUV-rinv (HK)	17	29	32	32	32	0.08
RUV-rinv-ev (E)	17	29	33	36	36	0.06

TCGA (Agilent)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	17	30	36	38	40	0.07
SVA-IRW	17	33	37	41	42	0.08
SVA-TS	17	33	37	38	42	0.08
LEAPP	17	33	37	38	43	0.12
ICE	17	33	34	37	41	0.01
RUV-4 (HK)	17	33	34	37	40	0.13
RUV-rinv (HK)	17	33	38	39	41	0.08
RUV-rinv-ev (E)	17	33	39	43	45	0.05

TCGA (Combined)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	12	14	16	17	17	0.02
SVA-IRW	17	21	25	25	26	0.07
SVA-TS	17	22	24	26	26	0.09
LEAPP	17	22	23	23	25	0.1
ICE	17	24	25	27	27	0.01
RUV-4 (HK)	17	22	24	25	25	0.16
RUV-rinv (HK)	17	24	27	28	28	0.06
RUV-rinv-ev (E)	17	25	27	29	31	0.05

Table 3.4: Comparison of the number of top-ranked X/Y genes and the effective type 1 error rates for SVA, LEAPP, ICE, and RUV. This continues Table 3.3.

TCGA (Exon) — Subset

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	13	17	18	18	18	0.08
SVA-IRW	15	18	19	24	25	0.05
SVA-TS	16	18	19	19	21	0.06
LEAPP	16	20	24	25	26	0.12
ICE	17	22	22	24	24	0.03
RUV-4 (HK)	15	18	21	24	25	0.13
RUV-rinv (HK)	17	21	22	23	24	0.07
RUV-rinv-ev (E)	16	22	22	23	23	0.05

TCGA (U133A) — Subset

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	15	17	18	19	19	0.05
SVA-IRW	14	16	16	19	20	0.07
SVA-TS	14	19	20	21	22	0.06
LEAPP	15	18	19	22	22	0.12
ICE	17	21	21	22	22	0.03
RUV-4 (HK)	16	19	20	22	26	0.11
RUV-rinv (HK)	16	19	21	23	23	0.05
RUV-rinv-ev (E)	16	21	23	23	23	0.06

TCGA (Agilent) — Subset

	Top 20	Top 40	Top 60	Top 80	Top 100	Type 1
unadjusted	11	14	18	18	19	0.05
SVA-IRW	13	16	16	17	18	0.06
SVA-TS	14	17	20	21	22	0.05
LEAPP	13	16	18	19	21	0.1
ICE	17	22	22	23	23	0.03
RUV-4 (HK)	15	17	19	19	19	0.11
RUV-rinv (HK)	16	19	22	22	23	0.04
RUV-rinv-ev (E)	16	21	23	23	23	0.05

Table 3.5: Comparison of the number of top-ranked X/Y genes and the effective type 1 error rates for SVA, LEAPP, ICE, and RUV. This continues Table 3.3.

Chapter 4

Conclusion

We now provide some final commentary. We begin by reconsidering the differences between RUV-2 and RUV-4. In Section 3.3.4 we discussed the differences between RUV-2 and RUV-4 extensively. For one, we found that RUV-4 is less sensitive than RUV-2 to the choice of K. For another, we found that RUV-4 is less sensitive to violations of the control gene assumption. This second observation, however, is only part of a larger point, which is that RUV-2 and RUV-4 use control genes differently. In some sense, both RUV-2 and RUV-4 use control genes as a reference point for a comparison. The variation present in the control genes is assumed to be unwanted variation. The negative control genes tell us what the unwanted variation "looks like." When we look at variation in non-control genes, the question we must answer is, "is this variation of interest?" To answer this question, we compare the variation we observe in the non-control genes to the variation we observe in the control genes. If the variation we observe in a non-control gene looks like the variation we observe in the negative controls, we conclude that there is no interesting variation present in that gene. However, if the variation in the non-control gene does not look like the variation present in the negative controls, we conclude that that there is indeed some interesting variation in that gene, i.e. that the gene is differentially expressed with respect to the factor of interest.

The difference between RUV-2 and RUV-4 is that the comparison between the non-control genes and the control genes is more "direct" with RUV-4 than it is with RUV-2. In Section 3.3.5.3, we found that the RUV-4 estimate of β might outperform even the (hypothetical) OLS estimate of β (if W were somehow known). In the discussion of Section 3.3.5.4, we noted that this enhanced performance of RUV-4 relies on the assumption that the $\alpha_{\star j_c}$ are representative of the $\alpha_{\star j_{\bar{c}}}$. In our discussion of the functional approach, we took this a step further. The assumption that the control genes are representative takes on a central role. From the functional point of view, the role of control genes in RUV-4 is to provide an estimate of the background signal, against which the signal of the non-control genes may be compared. To determine whether a non-control gene $j_{\bar{c}}$ is differentially expressed, we first compute its observed signal $(b_{YX})_{j_{\bar{c}}}$. We then calculate $\hat{\alpha}_{\star j_{\bar{c}}}$ to see how this gene has been affected by the unwanted factors. Next we check to see how much signal we observe from control genes that have been similarly affected by the unwanted factors, i.e. have similar values of $\hat{\alpha}$. Finally, we compare the observed signal of gene $j_{\bar{c}}$ to the observed signal of the comparable control genes (those that have similar values of $\hat{\alpha}$). If the observed signal of gene $j_{\bar{c}}$ is about the same as that of the comparable control genes, we conclude that the observed signal of gene $j_{\bar{c}}$ is just due to unwanted variation and that gene $j_{\bar{c}}$ is not differentially expressed. If, however, the observed signal of gene $j_{\bar{c}}$ is substantially different from the observed signal of the comparable control genes, we conclude that gene $j_{\bar{c}}$ is differentially expressed.

With RUV-2, the comparison between the non-control genes and control genes is less direct. The comparison is more strongly intermediated by the linear model. With RUV-2 it is not necessary that the values of α for the control genes are in any way representative of the values of α for the non-control genes. All that matters is that the control genes are affected by the same unwanted factors as the non-control genes, and that the linear model holds. With RUV-2 we use the control genes simply to identify the linear subspace in which the unwanted variation resides (Recall the RUV-2 estimate of W is better than the RUV-4 estimate of W, even if the RUV-4 estimate of β is better than the RUV-2 estimate of β. Indeed, unlike the RUV-4 estimate of W, it can be shown that under suitable conditions the RUV-2 estimate of W is consistent.). We then completely remove any and all variation within this subspace. Whether the patterns of variation within this subspace are similar between the control genes and the non-control genes is a moot point; all of the variation in this subspace is removed.

This difference between RUV-2 and RUV-4 has important practical implications. We have made a strong case in this thesis for the use of RUV-4. However, RUV-4 is not necessarily preferable to RUV-2 in all circumstances. Consider a case in which only a small number of genes are differentially expressed with respect to the factor of interest X. Suppose that these genes are also strongly affected by an unknown, unwanted factor W. Suppose also, however, that the unwanted factor W affects only a small number of the control genes. Then the values of α for the control genes will not be representative of the values of α for the genes that are differentially expressed. The values of α for the genes that are differentially expressed will be large (these genes are strongly affected by W) but the values of the α for the control genes will be mostly 0 (most control genes are unaffected by W).

As a concrete example, suppose there is a genetic disease and it is known that the disease is somehow caused by a gene or genes on the X chromosome. A researcher wants to perform a differential expression analysis to find the gene(s) associated with the disease. Suppose that both men and women are in the study. Gender will be an important source of unwanted biological variation. Suppose, however, that information on the gender of the people in the study is missing. The researcher therefore decides to use genes from the Y chromosome as negative controls. Since the researcher would also like to get a good estimate of any unwanted technical factors, and since there are not many genes on the Y chromosome, the researcher includes housekeeping genes in the set of negative controls as well. Now, most of the control genes will be housekeeping genes and unaffected by gender. Thus, the $\alpha_{\star j_c}$ will *not* be representative of the $\alpha_{\star j_{\bar{c}}}$ of the genes on the X chromosome. RUV-4 may fail to properly adjust for gender. On the other hand, as long as K is chosen large enough, gender should find its way into RUV-2's estimate of W, and RUV-2 may succeed in adjusting for gender.

Of course, with RUV-2, a relatively large K may lead to problems of its own. In practice, the best option may sometimes be a hybrid of RUV-2 and RUV-4. For example, the researcher might choose to perform factor analysis on just the genes of the Y chromosome and keep only the first few factors. Assuming that gender is captured in these first few factors, the researcher might then choose to include these factors as covariates in RUV-4. Although these factors are estimated from negative controls (specifically, Y-chromosome genes), they could be incorporated into RUV-4 in exactly the same way known covariates are incorporated into RUV-4. In other words, the W from RUV-2 (with Y-chromosome genes as controls) would now play the role of Z in RUV-4 (with housekeeping genes as controls).

Of course, RUV-2 is not the only other method with which RUV-4 shares an interesting connection. As noted in the introduction, RUV-4 is of theoretical interest precisely because it shares similarities both with methods such as SVA and LEAPP, in which unwanted factors are estimated from the data and then included in the design matrix of a regression model, and with methods such as ICE and LMM-EH, in which unwanted variation is modeled as part of a complicated error term.

Consider first a comparison of RUV-4, SVA, and LEAPP. All model unwanted variation as arising from unobserved latent variables (our W), and assume that the number of latent variables (our k) is less than the number of samples (our m). Each of these methods requires an estimate of k. We estimate k using the method of Section 3.3.6.6, while both SVA and LEAPP estimate k using the method of Buja and Eyuboglu (1992). Each of the methods allows every gene to have its own variance (our σ_j^2), and estimates these variances in the "standard" way (from the residuals). Where these methods differ is in the exact method by which they estimate the latent factors. Even here, however, there are similarities — each of the methods begins by projecting away the factor of interest (with LEAPP, this is formulated as a rotation) in order to make sure that the factor of interest is not accidentally picked up with the unwanted factors. After this first step, though, the methods differ. RUV-4 relies on control genes to estimate b_{WX}. LEAPP proceeds somewhat similarly, and also estimates b_{WX} as an intermediate step. Instead of relying on control genes, however, LEAPP assumes that β is sparse, and then applies the outlier-detection algorithm Θ-IPOD (She and Owen, 2011). SVA attempts to isolate genes that are primarily influenced by the unwanted factors but not influenced by the factor of interest, and then proceeds to estimate the unwanted factors by focusing the factor analysis on just those genes.

Consider now a comparison of RUV-inv (which is simply RUV-4 with $K = m - p - q$) and the mixed model methods ICE and LMM-EH. All model the unwanted variation as part of a random error term with a complicated covariance structure. In each of these methods, the covariance matrix is assumed to be of the form $\tau_j^2\Sigma + \sigma_j^2 I$, where Σ is the same for all genes. For the purposes of estimating β, RUV-inv effectively assumes that τ_j^2 and σ_j^2 are constant across genes (see Section 3.3.7.4 and the discussion in Section B.1.3), but ICE and LMM-EH allow τ_j^2 and σ_j^2 to vary by gene. For the purposes of estimating $\text{Var}(\hat{\beta})$, however, all three methods allow σ_j^2 to vary by gene. The real difference between the methods is in the way the model is fit. Both ICE and LMM-EH fit all of the parameters of the model simultaneously using maximum likelihood methods. With RUV-inv, however, the covariance matrix is estimated using the control genes, β is estimated using GLS, and σ_j^2 is estimated using the inverse method. It is the inverse method that is arguably the most important difference between RUV-inv and other mixed model methods.

The inverse method for estimating variances is quite different from the methods that ICE and LMM-EH use to estimate variances. Indeed, the inverse method is unlike any other method of which we are aware.

Some readers may find the inverse method reminiscent of randomization tests. Both randomization tests and the inverse method make use of "random factors of interest." We too see similarities, and we have found the analogy with randomization tests to be helpful in developing intuition for what the inverse method is doing. Of course, there are also serious differences, perhaps the most obvious of which is that randomization tests are generally used with non-parametric models, whereas the inverse method is used in the context of a parametric model.

On an intuitive level, the inverse method may perhaps be best understood as a hybrid between traditional randomization tests and traditional parametric methods. This hybrid loses some of the nicer conceptual properties of randomization tests, but retains some of the practical benefits. The p-values produced by randomization tests (properly applied) have a very clear, believable interpretation. The p-values produced by the inverse method do not; inverse method p-values are computed using artificial modeling assumptions. On the other hand, like randomization tests, the inverse method does appear to maintain fairly good control of the type-1 error rate.

We conclude this thesis with some suggestions for future research. One direction for improvement would be to allow σ^2 to vary not just by gene, but by sample. This would have immediate applications when combining data from different microarray platforms (e.g. Affymetrix arrays and Agilent arrays); the variance of the measured expression level of a gene can be quite different from one platform to another. By allowing σ^2 to vary from one batch to another, it may be possible to achieve an increase in power.

The method of empirical variances presents a second possibility for future improvements. Our development of the method of empirical variances in this thesis has been mainly proof of principle. It seems very likely that improvements could be made. For example, other methods of non-linear regression may out-perform the minimum lower sets algorithm. Moreover, our implementation of the method of empirical variances relies on the assumption that β is sparse, and ignoring outliers. Instead of assuming sparsity and ignoring outliers, however, it may be better to simply limit the non-linear regression to control genes. We did not pursue this approach because the function fit by the minimum lower sets algorithm is very flexible, and some of our datasets contained only a few hundred negative controls. The estimated regression function would be far too noisy. This problem could be solved, however, by either increasing the number of controls (e.g. empirical controls), or replacing the minimum lower sets algorithm with a less flexible alternative.

Additional possibilities for future research lay in the estimation of k. Our method for estimating k relies on comparing the scale of the variation seen in $\hat{\beta}_c$ to the scale of the variation seen in $\hat{\alpha}_c$. Thus, since $\hat{\beta}$ is a function of X, our estimate of k is also a function of X. However, factor analysis has many applications outside the removal of unwanted variation in differential expression analyses, and in many applications of factor analysis there is no factor of interest to play the role of our X. However, there may still be a need to know the number of factors. In such situations, a researcher could simply choose many $X^\star$ at random and produce many different estimates of k. A final estimate of k could then be produced, for example, by taking the median. Note, moreover, that with a random $X^\star$, we might wish to regard every gene (or more generally, "feature") as a control gene, on the grounds that no gene should be "truly" differentially expressed with respect to a random $X^\star$. Thus, if we estimate k via random $X^\star$, it is not even necessary to have any negative controls. Such a method may therefore present a novel and widely applicable solution to the number-of-factors problem.

Finally, we would like to reiterate our belief that significant advances may be made by fully exploiting the functional approach. In particular, the possibility of incorporating "outside" information, such as a gene's GC content, seems to hold great promise.

Bibliography

Affymetrix. *GeneChip Expression Analysis Technical Manual.* Affymetrix, Inc., 2005-2009. Available at http://www.affymetrix.com /support/downloads/manuals/expression_analysis_technical_manual.pdf.

O. Alter, P.O. Brown, and D. Botstein. Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 97(18):10101–10106, 2000.

R.E. Barlow, D.J. Bartholomew, J.M. Bremner, and H.D. Brunk. *Statistical Inference under Order Restrictions; The Theory and Application of Isotonic Regression.* Wiley, 1972.

H. Bengtsson, K. Simpson, J. Bullard, and K. Hansen. aroma.affymetrix: A generic framework in R for analyzing small to very large affymetrix data sets in bounded memory. Technical Report #745, Department of Statistics, University of California, Berkeley, February 2008.

C.M. Bishop. *Pattern recognition and machine learning.* Springer, 2006.

E.M. Blalock, J.W. Geddes, K.C. Chen, N.M. Porter, W.R. Markesbery, and P.W. Landfield. Incipient Alzheimer's disease: microarray correlation analyses reveal major transcriptional and tumor suppressor responses. *Proceedings of the National Academy of Sciences of the United States of America*, 101(7):2173, 2004.

B. Bolstad, F. Collin, J. Brettschneider, K. Simpson, L. Cope, R. Irizarry, and T.P. Speed. Quality assessment of Affymetrix GeneChip data. *Bioinformatics and computational biology solutions using R and bioconductor*, pages 33–47, 2005.

B.M. Bolstad, R.A. Irizarry, M. Astrand, and T.P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185, 2003. ISSN 1367-4803.

J. Brettschneider, F. Collin, B.M. Bolstad, and T.P. Speed. Quality assessment for short oligonucleotide microarray data. *Technometrics*, 50(3):241–264, 2008. ISSN 0040-1706.

A. Buja and N. Eyuboglu. Remarks on parallel analysis. *Multivariate Behavioral Research*, 27(4):509–540, 1992.

E. Eisenberg and E.Y. Levanon. Human housekeeping genes are compact. *TRENDS in Genetics*, 19(7):362–365, 2003. ISSN 0168-9525.

T.L. Fare, E.M. Coffey, H. Dai, Y.D. He, D.A. Kessler, K.A. Kilian, J.E. Koch, E. LeProust, M.J. Marton, M.R. Meyer, et al. Effects of atmospheric ozone on microarray data quality. *Analytical chemistry*, 75(17):4672–4675, 2003.

D.A. Freedman, R. Pisani, and R. Purves. *Statistics*. W.W. Noton & Company, Inc., fourth edition, 2007. ISBN 0-393-92972-8.

J. Friedman, T. Hastie, and R. Tibshirani. *The elements of statistical learning*, volume 1. Springer Series in Statistics, 2 edition, 2009.

M. Hubert, P.J. Rousseeuw, and K. Vanden Branden. ROBPCA: a new approach to robust principal component analysis. *Technometrics*, 47(1):64–79, 2005. ISSN 0040-1706.

R.A. Irizarry, B.M. Bolstad, F. Collin, L.M. Cope, B. Hobbs, and T.P. Speed. Summaries of Affymetrix GeneChip probe level data. *Nucleic acids research*, 31(4):e15, 2003a. ISSN 0305-1048.

R.A. Irizarry, B. Hobbs, F. Collin, Y.D. Beazer-Barclay, K.J. Antonellis, U. Scherf, and T.P. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249, 2003b. ISSN 1465-4644.

W.E. Johnson, C. Li, and A. Rabinovic. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, 8(1):118–127, 2007.

H.M. Kang, C. Ye, and E. Eskin. Accurate discovery of expression quantitative trait loci under confounding from spurious and genuine regulatory hotspots. *Genetics*, 180(4):1909–1925, 2008a.

H.M. Kang, N.A. Zaitlen, C.M. Wade, A. Kirby, D. Heckerman, M.J. Daly, and E. Eskin. Efficient control of population structure in model organism association mapping. *Genetics*, 178(3):1709, 2008b.

H.M. Kang, J.H. Sul, S.K. Service, N.A. Zaitlen, S.Y. Kong, N.B. Freimer, C. Sabatti, and E. Eskin. Variance component model to account for sample structure in genome-wide association studies. *Nature Genetics*, 42(4):348–354, 2010. ISSN 1061-4036.

J.T. Leek and J.D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3:e161, 2007.

J.T. Leek and J.D. Storey. A general framework for multiple testing dependence. *Proceedings of the National Academy of Sciences*, 105(48):18718–18723, 2008. ISSN 0027-8424.

J.T. Leek, R.B. Scharpf, H.C. Bravo, D. Simcha, B. Langmead, W.E. Johnson, D. Geman, K. Baggerly, and R.A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11:733–739, 2010.

K.A. Lippa, D.L. Duewer, M.L. Salit, L. Game, and H.C. Causton. Exploring the use of internal and external controls for assessing microarray technical performance. *BMC Research Notes*, 3(349), 2010. ISSN 1756-0500.

J. Listgarten, C. Kadie, E.E. Schadt, and D. Heckerman. Correction for hidden confounders in the genetic analysis of gene expression. *Proceedings of the National Academy of Sciences*, 107(38):16465, 2010. ISSN 0027-8424.

J. Lucas, C. Carvalho, Q. Wang, A. Bild, J. Nevins, and M. West. Sparse statistical modelling in gene expression genomics. In K. Do, P. Muller, and M. Vannucci, editors, *Bayesian Inference for Gene Expression and Proteomics*, pages 156–176. Cambridge University Press, 2006.

Z. Ma. Accuracy of the tracy–widom limits for the extreme eigenvalues in white wishart matrices. *Bernoulli*, 18(1):322–359, 2012.

J.R. Magnus. The exact moments of a ratio of quadratic forms in normal variables. *Annales d'Economie et de Statistique*, pages 95–109, 1986.

B.H. Mecham, P.S. Nelson, and J.D. Storey. Supervised normalization of microarrays. *Bioinformatics*, 26(10):1308–1315, 2010. ISSN 1367-4803.

T.O. Nielsen, R.B. West, S.C. Linn, O. Alter, M.A. Knowling, J.X. O'Connell, S. Zhu, M. Fero, G. Sherlock, J.R. Pollack, P.O. Brown, D. Botstein, and M. van de Rijn. Molecular characterisation of soft tissue tumours: a gene expression study. *The Lancet*, 359(9314): 1301–1307, 2002. ISSN 0140-6736.

A.L. Price, N.J. Patterson, R.M. Plenge, M.E. Weinblatt, N.A. Shadick, and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38(8):904–909, 2006.

A. Scherer. *Batch Effects and Noise in Microarray Experiments: Sources and Solutions.* Wiley, 2009. ISBN 0470741384.

U.T. Shankavaram, W.C. Reinhold, S. Nishizuka, S. Major, D. Morita, K.K. Chary, M.A. Reimers, U. Scherf, A. Kahn, D. Dolginow, J. Cossman, Kaldjian E.P., D.A. Scudiero, E. Petricoin, L. Liotta, J.K. Lee, and J.N. Weinstein. Transcript and protein expression profiles of the NCI-60 cancer cell panel: an integromic microarray study. *Molecular Cancer Therapeutics*, 6(3):820–832, 2007. ISSN 1535-7163.

Y. She and A.B. Owen. Outlier detection using nonconvex penalized regression. *Journal of the American Statistical Association*, 106(494):626–639, 2011.

L. Shi, L.H. Reid, W.D. Jones, R. Shippy, J.A. Warrington, S.C. Baker, P.J. Collins, F. de Longueville, E.S. Kawasaki, K.Y. Lee, et al. The MicroArray Quality Control (MAQC) project shows inter-and intraplatform reproducibility of gene expression measurements. *Nature Biotechnology*, 24(9):1151–1161, 2006. ISSN 1087-0156.

G.K. Smyth. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Statistical applications in genetics and molecular biology*, 3(1), 2004.

B.S. Stamova, M. Apperson, W.L. Walker, Y. Tian, H. Xu, P. Adamczy, X. Zhan, D.Z. Liu, B.P. Ander, I.H. Liao, J.P. Gregg, R.J. Turner, G. Jickling, L. Lit, and F.R. Sharp. Identification and validation of suitable endogenous reference genes for gene expression studies in human peripheral blood. *BMC Medical Genomics*, 2(49), 2009. ISSN 1755-8794.

O. Stegle, A. Kannan, R. Durbin, and J. Winn. Accounting for non-genetic factors improves the power of eQTL studies. In *Proceedings of the 12th annual international conference on Research in computational molecular biology*, pages 411–422. Springer-Verlag, 2008. ISBN 3540788387.

Y. Sun, N. Zhang, and A.B. Owen. Multiple hypothesis testing, adjusting for latent variables. 2011. Available at http://www-stat.stanford.edu/~owen/reports/leapp.pdf.

M.P. Vawter, S. Evans, P. Choudary, H. Tomita, J. Meador-Woodruff, M. Molnar, J. Li, J.F. Lopez, R. Myers, D. Cox, S.J. Watson, H. Akil, E.G. Jones, and W.E. Bunney. Gender-specific gene expression in post-mortem human brain: localization to sex chromosomes. *Neuropsychopharmacology*, 29(2):373–384, 2004.

W.N. Venables and B.D. Ripley. *Modern applied statistics with S*. Springer-Verlag, 2002. ISBN 0387954570.

F.T. Wright. Estimating strictly increasing regression functions. *Journal of the American Statistical Association*, 73(363):636–639, 1978.

J. Yu, G. Pressoir, W.H. Briggs, I.V. Bi, M. Yamasaki, J.F. Doebley, M.D. McMullen, B.S. Gaut, D.M. Nielsen, J.B. Holland, S. Kresovich, and E.S. Buckler. A unified mixed-model method for association mapping that accounts for multiple levels of relatedness. *Nature Genetics*, 38(2):203–208, 2005.

Appendix A

RUV-2 Supplementary Material

A.1 MAQC Example

The Microarray Quality Control (MAQC) project (Shi et al., 2006) sought to investigate the replicability of microarray results by processing multiple technical replicates of a few biological samples at multiple laboratories on multiple microarray platforms. In particular, they processed 5 technical replicates of Stratagene Universal Human Reference RNA ("Sample A") and 5 technical replicates of Ambion Human Brain Reference RNA ("Sample B") at each of 6 laboratories (for a total of 60 chips) using Affymetrix HG-U133 Plus 2 arrays. To investigate unwanted variation in this dataset, we first subtracted off sample means gene by gene (to remove the biological signal), and then computed the SVD of the resulting matrix. Substantial variation between laboratories is evident, but so is substantial within-laboratory variation.

We contacted MAQC to enquire if there was any known explanation for the relatively large (though still minor in absolute terms) unwanted variation at the fourth laboratory (the E.P.A., shown in red). We were informed that at this laboratory, the person who ran the experiment did not get enough training and hands-on experience.

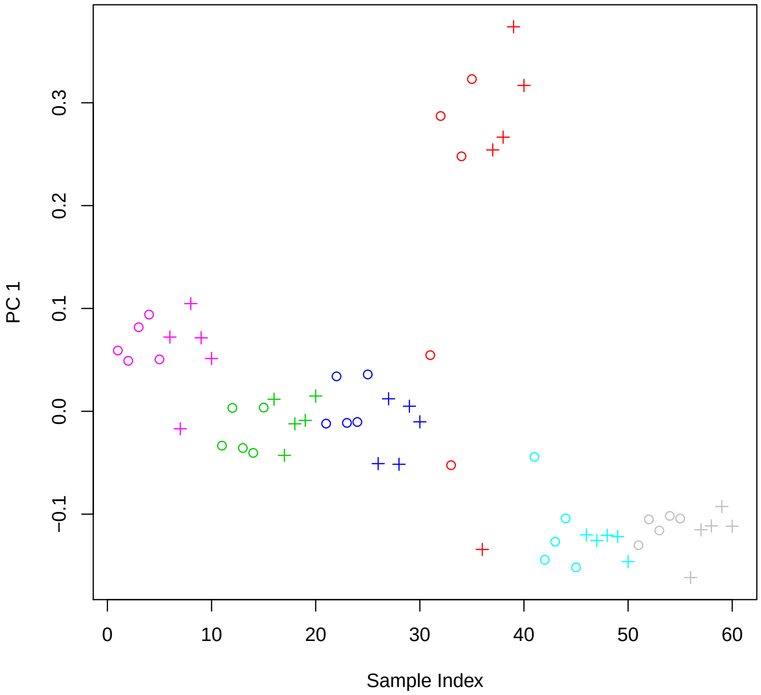


Figure A.1: The first PC of the MAQC HG-U133 Plus 2 data after removal of the biological signal. Different colors represent different laboratories. Circles represent Sample A; pluses represent Sample B. Unwanted variation occurs both between and within batches. The data was not preprocessed.

A.2 Example Code

RUV-2 is very simple to implement and does not warrant an R package. Instead, we include here some sample code implementing RUV-2 using SVD and Limma. Note that the expression matrix in this example is $n \times m$ instead of $m \times n$, since this is how expression data is often stored in practice. The variable `ctl` must index the genes to be used as control genes.

```
RUV2 = function(Y, X, ctl, k, Z=matrix(rep(1, ncol(Y))))
{
  library(limma)

  # Project onto the orthogonal complement of Z
  RZY = Y - Y%*%Z%*%solve(t(Z)%*%Z)%*%t(Z)

  # Perform SVD
  W = svd(RZY[ctl,])$v

  # Keep the first k factors
  W = W[,1:k]

  # Fit using Limma and return
  fit = lmFit(Y,cbind(X,Z,W))
  fit = eBayes(fit)
  return(fit)
}
```

A.3 Additional Gender Study Figures

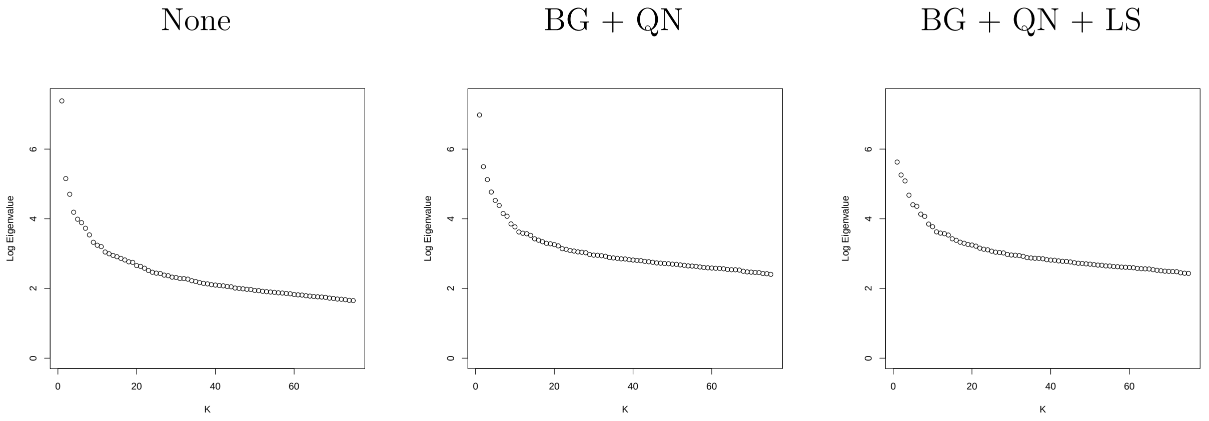


Figure A.2: Gender study scree plots on a log scale at different stages of preprocessing. From left to right: No preprocessing; background correction / quantile normalization done separately for each platform type; background correction / quantile normalization followed by a final location/scale adjustment across all chips. All genes were included in the eigenanalysis.

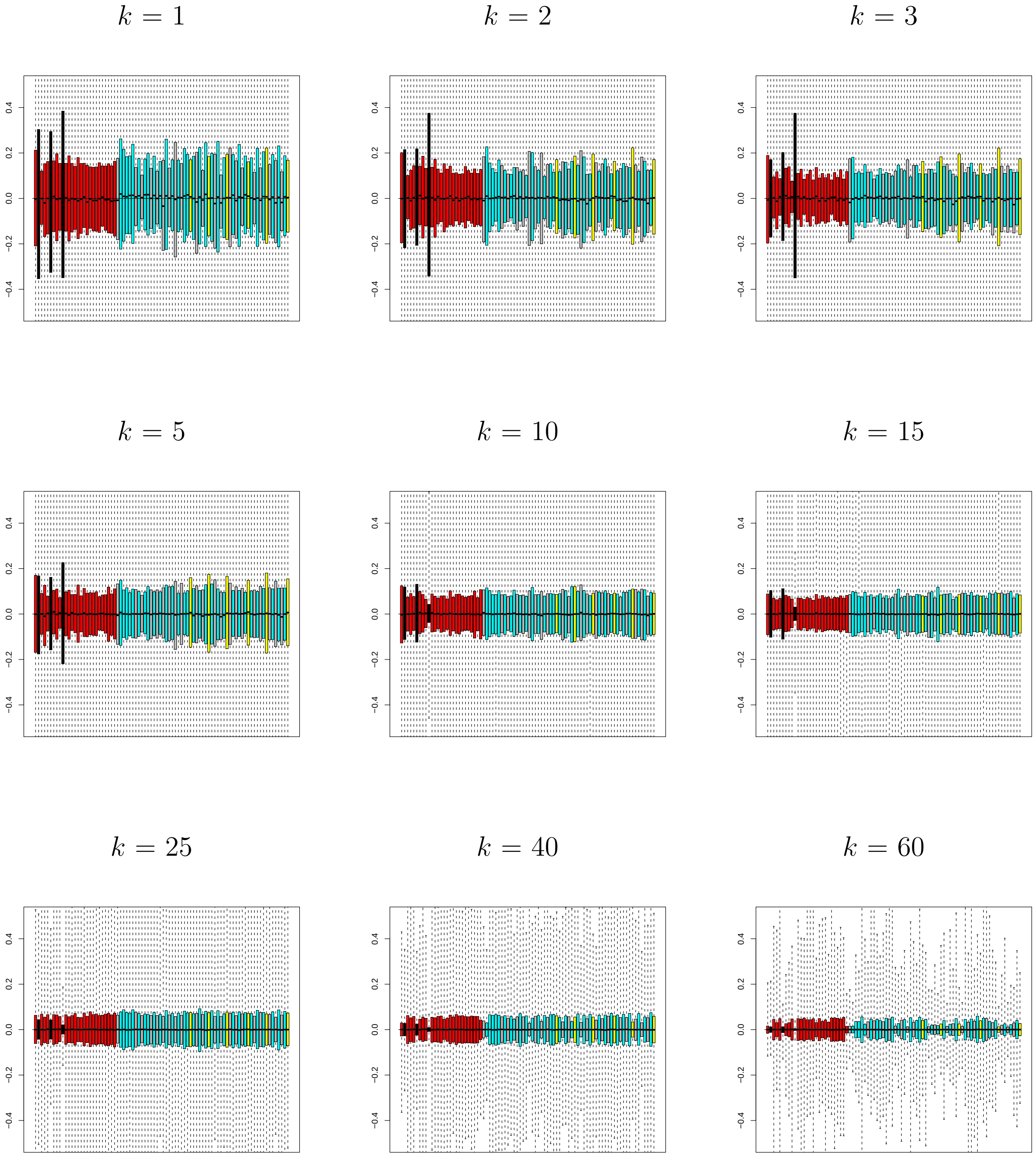
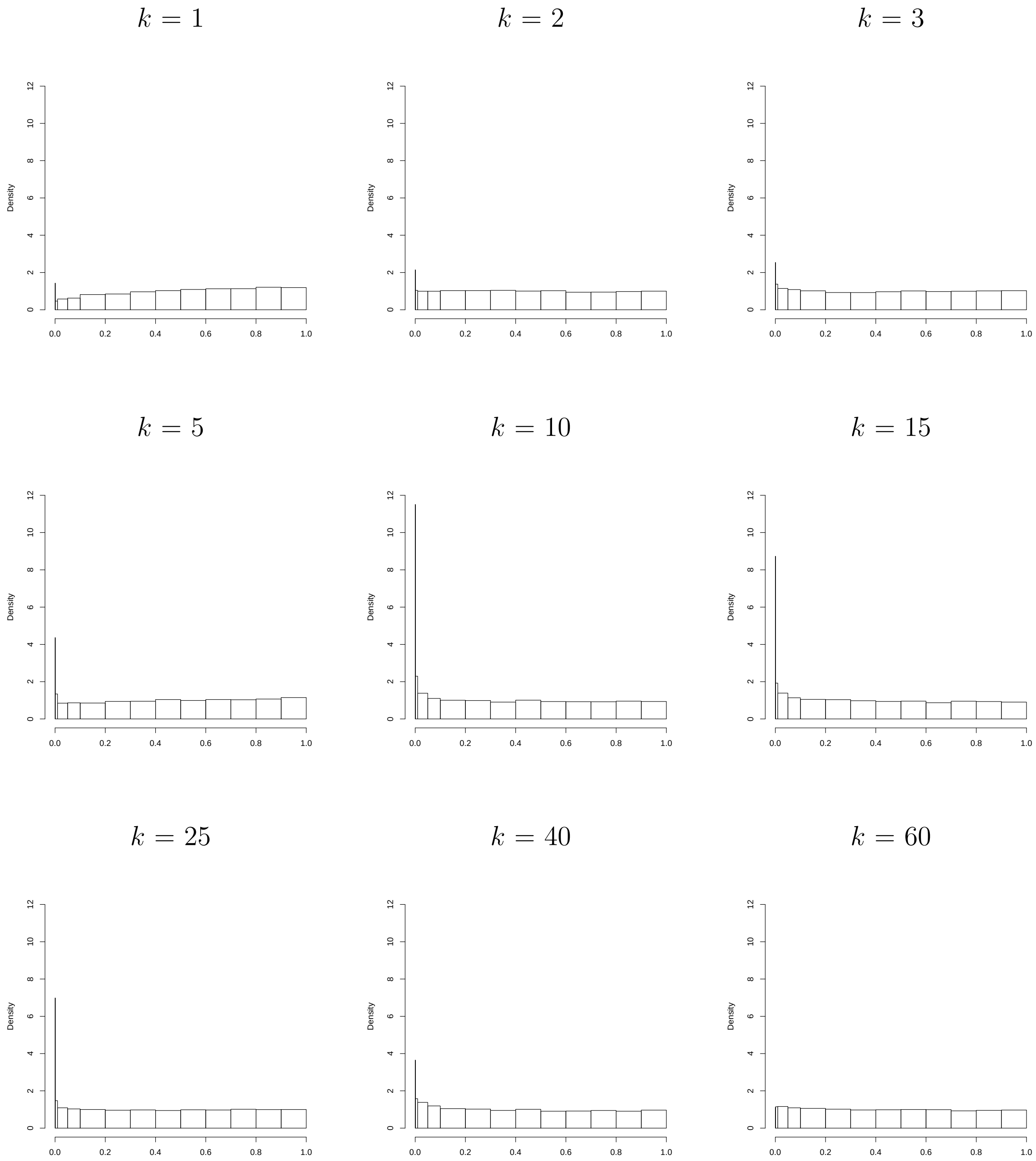


Figure A.3: Gender study RLE plots after adjustment, using various values of k. Quality improves with increasing k at first, then decreases as too many factors are removed. Data was fully preprocessed (BG+QN+LS). The factors were computed by SVD on the housekeeping genes.



$k = 1$

$k = 2$

$k = 3$

$k = 5$

$k = 10$

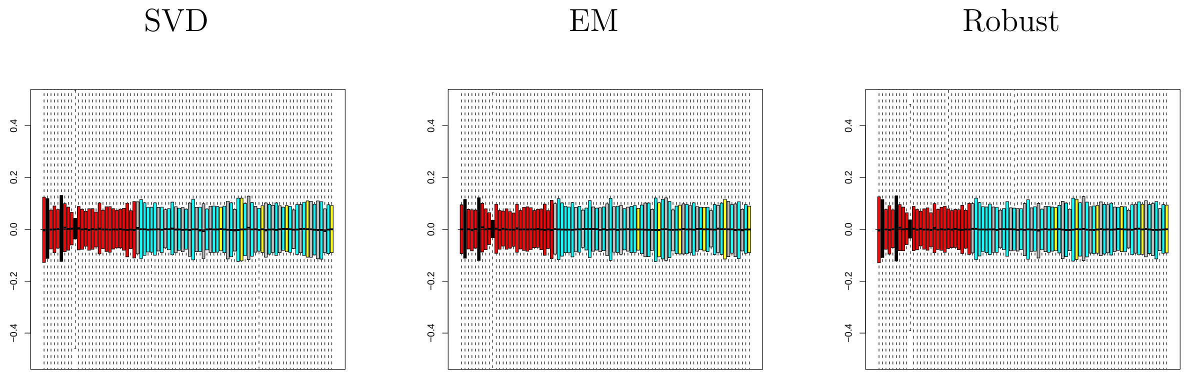
$k = 15$

$k = 25$

$k = 40$

$k = 60$

Figure A.4: Gender study p-value histograms after adjustment, using various values of k. Histogram breakpoints are at 0.001, 0.01, 0.05, and 0.1, 0.2, 0.3, etc. Data was fully preprocessed (BG+QN+LS). The factors were computed by SVD on the housekeeping genes. P-values were computed using Limma.


SVD

EM

Robust

Figure A.5: Gender study RLE plots after adjustment ($k = 10$), using different methods of factor analysis. The results are remarkably similar. The data was preprocessed (BG + QN + LS). The factors were computed using the housekeeping genes.

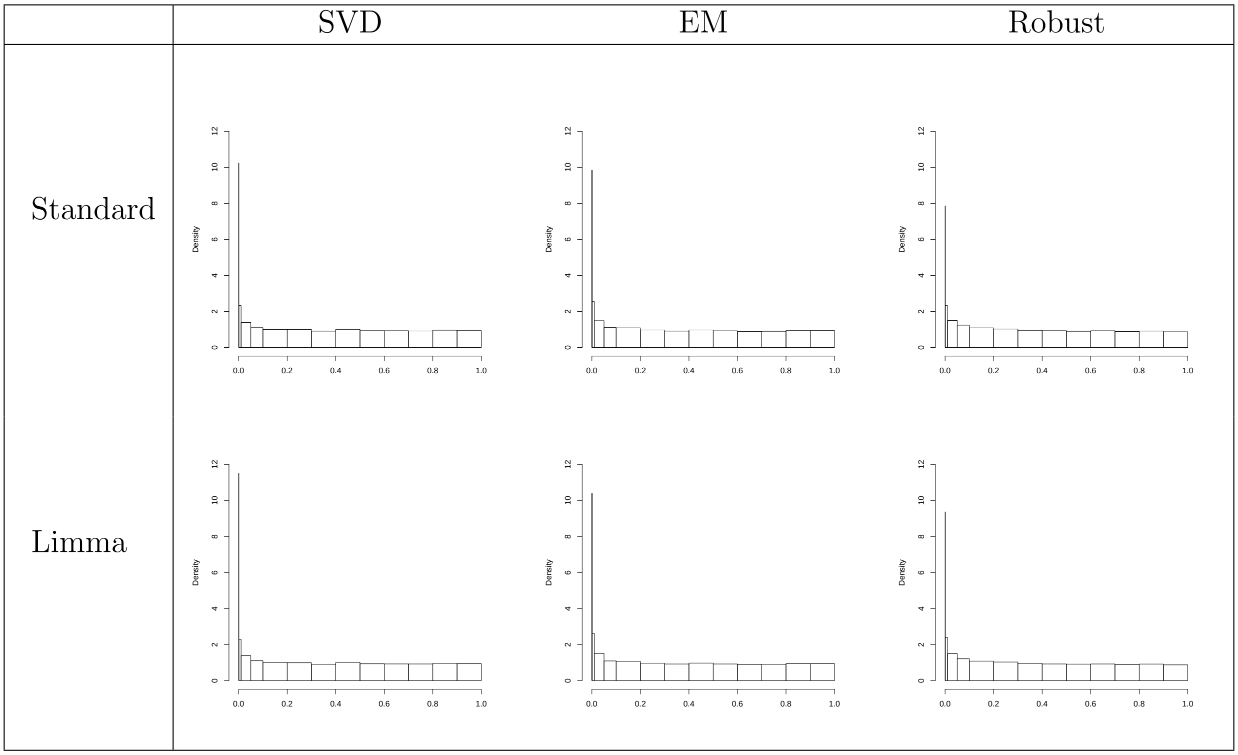


Figure A.6: Gender study p-value histograms after adjustment ($k = 10$), using different methods of factor analysis, and different methods of computing p-values. The data was preprocessed (BG + QN + LS). The factors were computed using the housekeeping genes.

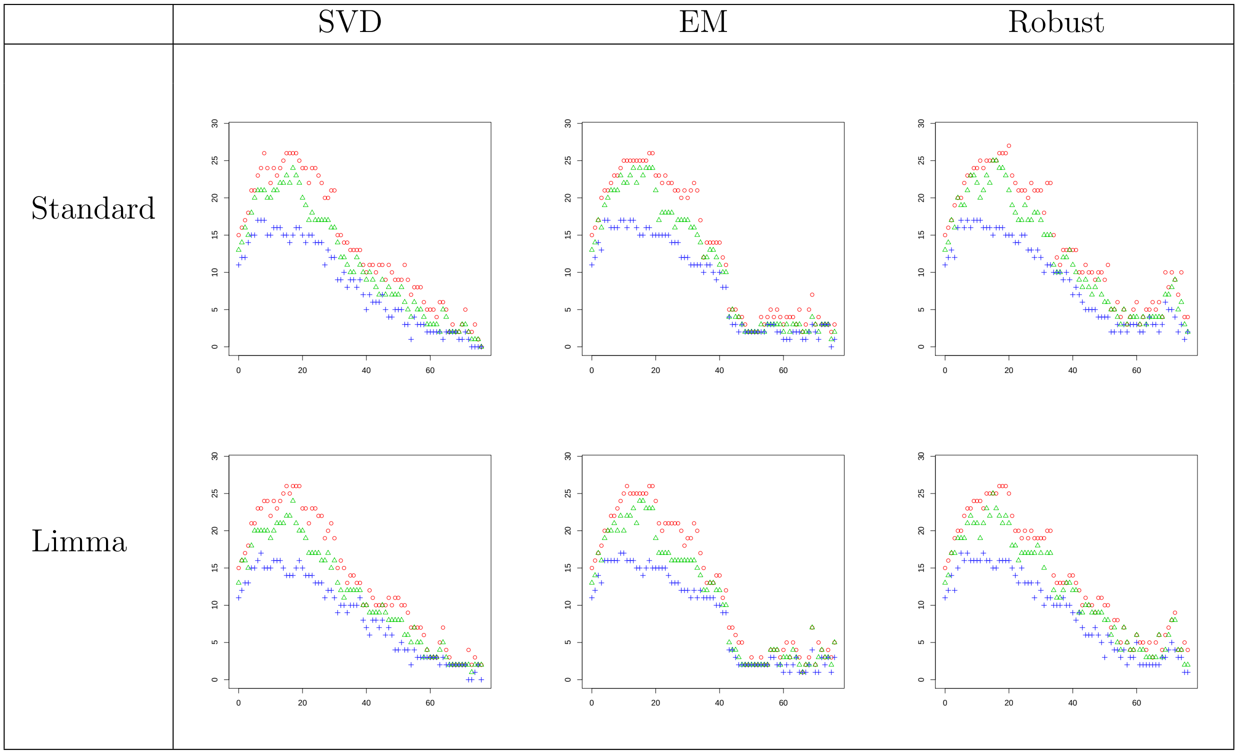


Figure A.7: Comparison of the performance of variants of RUV-2 in the gender study. The number of X / Y genes discovered is plotted as a function of k. Genes were ranked by p-value; results are shown for the number of X / Y genes ranked in the top 20 (plus), top 40 (triangle), and top 60 (circle). $k = 0$ corresponds to no adjustment. Data was preprocessed (BG + NM + LS). PCs were computed using the housekeeping genes.

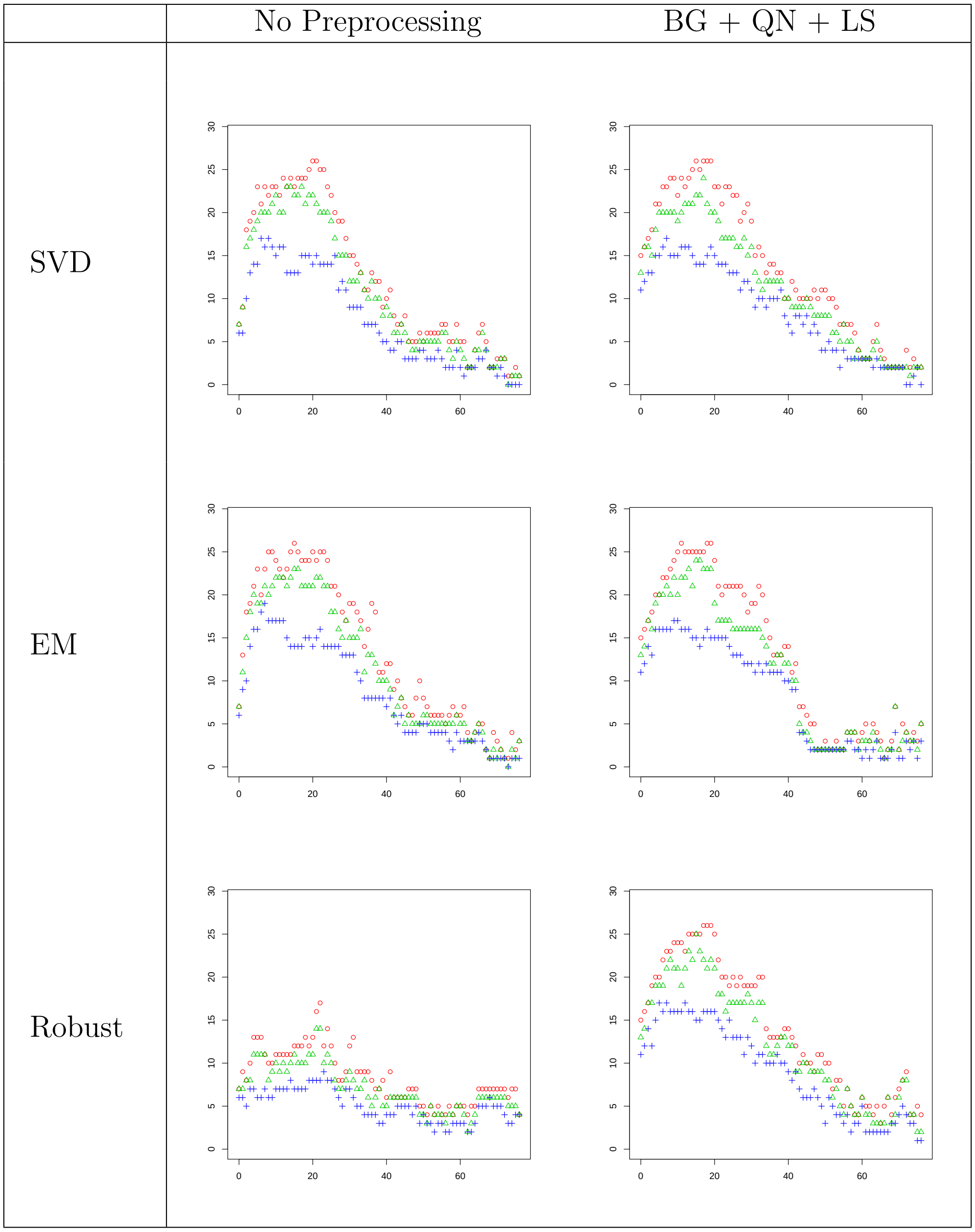


Figure A.8: Comparison of the performance of different factor analysis methods with and without preprocessing in the gender study. The number of X / Y genes discovered is plotted as a function of k. Genes were ranked by p-value; results are shown for the number of X / Y genes ranked in the top 20 (plus), top 40 (triangle), and top 60 (circle). $k = 0$ corresponds to no adjustment. PCs were computed using the housekeeping genes. P-values were computed using Limma.

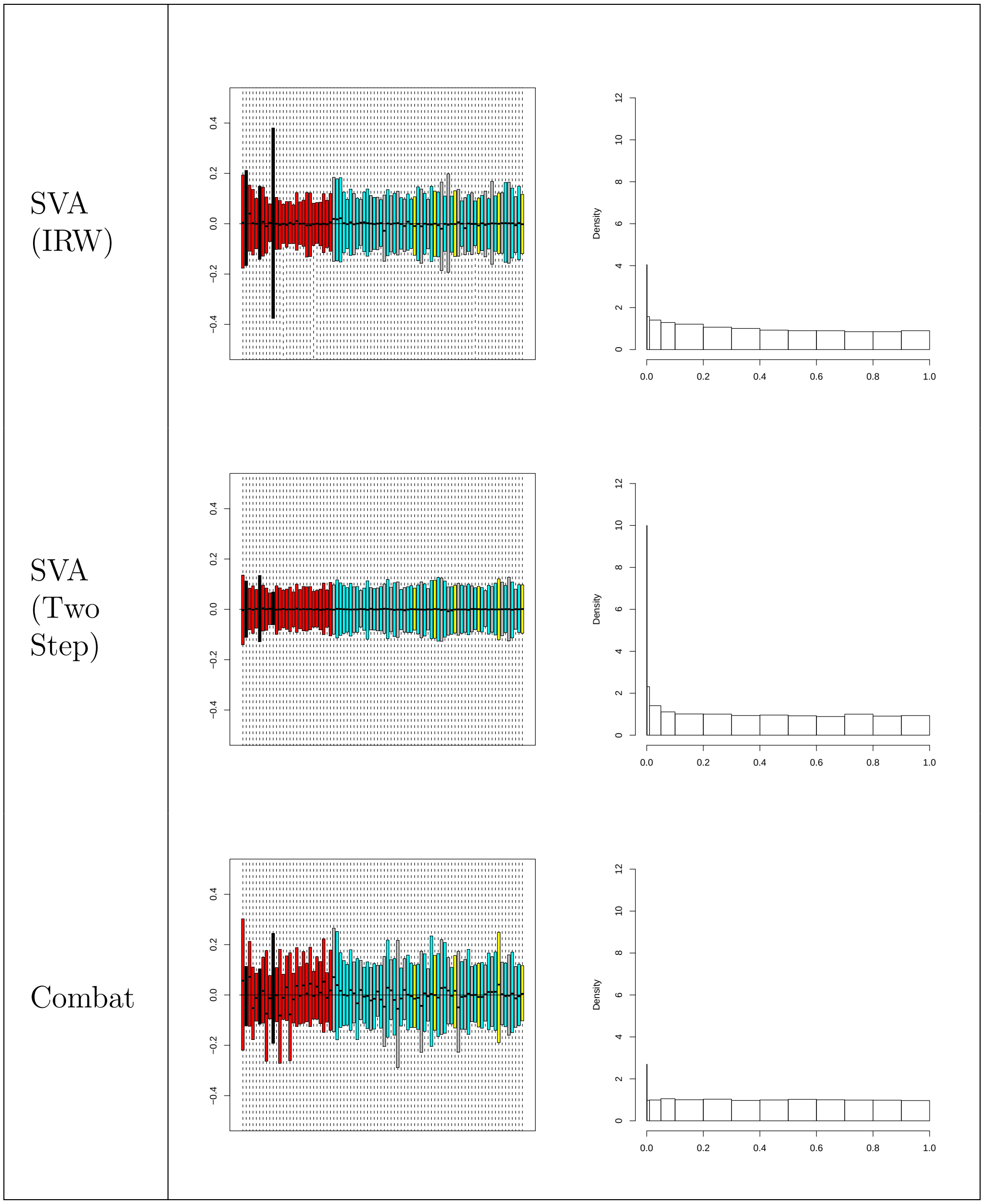


Figure A.9: Gender study RLE plots and p-value histograms after adjustments by SVA / Combat. The "two-step" variant of SVA appears to do fairly well. P-values were computed using Limma.

Unadjusted

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	RPS4Y1	Y	6×10^{-39}	11	IL2RA	10	0.4	21	PTPN20B	10	1
2	KDM5D	Y	2×10^{-19}	12	ASMTL	X Y	0.5	22	SLC25A6	X Y	1
3	DDX3Y	Y	3×10^{-19}	13	IFITM1	11	0.7	23	CDC42	1	1
4	XIST	X	2×10^{-13}	14	USP9X	X	0.9	24	COASY	17	1
5	USP9Y	Y	4×10^{-11}	15	CD24	6	1	25	VWF	12	1
6	TTTY15	Y	2×10^{-08}	16	PECAM1	17	1	26	DBC1	9	1
7	UTY	Y	2×10^{-07}	17	HBA1	16	1	27	CD59	11	1
8	EIF1AY	Y	3×10^{-05}	18	PCDH11X	X	1	28	IL10RB	21	1
9	HBB	11	5×10^{-02}	19	HBG1	11	1	29	DNAJB1	19	1
10	CIRBP	19	0.3	20	HBB	11	1	30	ANKRD26	10	1

RUV-2 (SVD), $k = 10$, housekeeping genes

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	RPS4Y1	Y	1×10^{-43}	11	DDX3X	X	5×10^{-06}	21	NOMO1	16	3×10^{-03}
2	KDM5D	Y	2×10^{-26}	12	LITAF	16	4×10^{-05}	22	TF	3	3×10^{-03}
3	DDX3Y	Y	6×10^{-24}	13	PCDH11X	X	6×10^{-05}	23	HBA1	16	4×10^{-03}
4	XIST	X	1×10^{-19}	14	HBG1	11	3×10^{-04}	24	GTPBP6	X Y	4×10^{-03}
5	USP9Y	Y	2×10^{-16}	15	USP9X	X	6×10^{-04}	25	ENPP2	8	4×10^{-03}
6	UTY	Y	7×10^{-14}	16	SLC25A6	X Y	6×10^{-04}	26	TUBA1B	12	5×10^{-03}
7	CD99	X Y	7×10^{-12}	17	FAM153A	5	1×10^{-03}	27	KLK6	19	5×10^{-03}
8	CYorf15B	Y	5×10^{-10}	18	RPS4X	X	1×10^{-03}	28	IFITM1	11	5×10^{-03}
9	TTTY15	Y	2×10^{-08}	19	PPP2CB	8	2×10^{-03}	29	HBB	11	5×10^{-03}
10	EIF1AY	Y	4×10^{-07}	20	H2AFY	5	2×10^{-03}	30	PRKY	Y	7×10^{-03}

Table A.1: Comparison of gender study gene rankings before and after adjustment. The data has been fully preprocessed (BG + NM + LS). The p-values were calculated using Limma.

Unadjusted

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	RPS4Y1	Y	1	11	CRYM	16	1	21	GAP43	3	1
2	DDX3Y	Y	1	12	XIST	X	1	22	GNAS	20	1
3	HBB	11	1	13	CALM1	14	1	23	SPP1	4	1
4	HBA1	16	1	14	ACTB	7	1	24	SNAP25	20	1
5	HBB	11	1	15	PFN2	3	1	25	UTY	Y	1
6	CIRBP	19	1	16	PRKAR1A	17	1	26	SLC17A7	19	1
7	KDM5D	Y	1	17	ACTB	7	1	27	RPS23	5	1
8	GAPDH	12	1	18	SLC25A6	X Y	1	28	RPS21	20	1
9	GAD1	2	1	19	GAPDH	12	1	29	HSP90AB1	6	1
10	USP9Y	Y	1	20	GNAS	20	1	30	ATP5A1	18	1

RUV-2 (SVD), $k = 10$, housekeeping genes

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	RPS4Y1	Y	8×10^{-30}	11	HBB	11	2×10^{-05}	21	PDE4DIP	1	1×10^{-03}
2	KDM5D	Y	4×10^{-21}	12	HBA1	16	4×10^{-05}	22	DDX3X	X	1×10^{-03}
3	DDX3Y	Y	4×10^{-14}	13	SLC25A6	X Y	1×10^{-04}	23	EIF1AY	Y	3×10^{-03}
4	XIST	X	1×10^{-11}	14	PCDH11X	X	1×10^{-04}	24	NCL	2	3×10^{-03}
5	CD99	X Y	4×10^{-11}	15	HBB	11	2×10^{-04}	25	HDHD1A	X	3×10^{-03}
6	UTY	Y	5×10^{-10}	16	SLC25A6	X Y	5×10^{-04}	26	GTPBP6	X Y	3×10^{-03}
7	USP9Y	Y	3×10^{-09}	17	USP9X	X	6×10^{-04}	27	ASMTL	X Y	3×10^{-03}
8	RPS4X	X	5×10^{-07}	18	FDPS	1	6×10^{-04}	28	IFITM1	11	4×10^{-03}
9	TTTY15	Y	7×10^{-07}	19	HBG1	11	6×10^{-04}	29	PIN1	19	5×10^{-03}
10	CYorf15B	Y	1×10^{-05}	20	NLGN4Y	Y	6×10^{-04}	30	POLD2	7	6×10^{-03}

Table A.2: Comparison of gender gene rankings before and after adjustment. The data has not been preprocessed. The p-values were calculated using Limma.

A.4 Additional Alzheimer's Study Figures

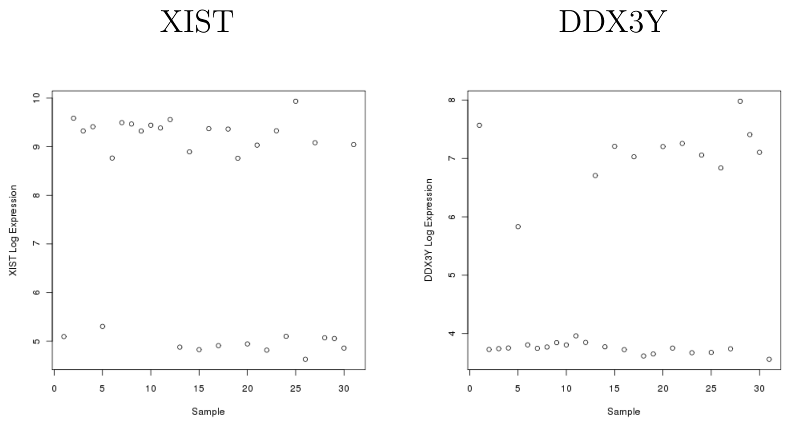


Figure A.10: Plots of XIST and DDX3Y expression levels in the Alzheimer's study. The horizontal axis is just sample index. It is clear which samples are male and which are female. Data was preprocessed.

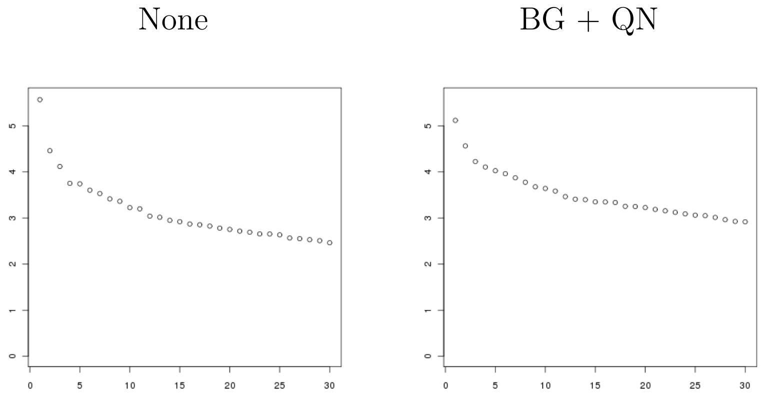


Figure A.11: Alzheimer's study scree plots with and without preprocessing. Left: No preprocessing; Right: Background correction / quantile normalization. All genes were included in the eigenanalysis.

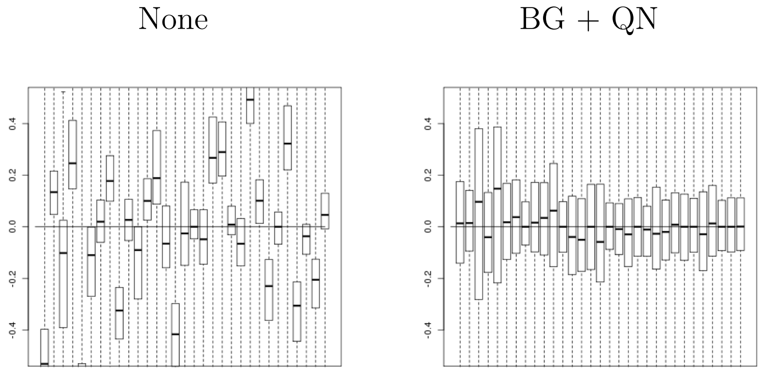


Figure A.12: Alzheimer's study RLE plots with and without preprocessing. Left: No preprocessing; Right: Background correction / quantile normalization.

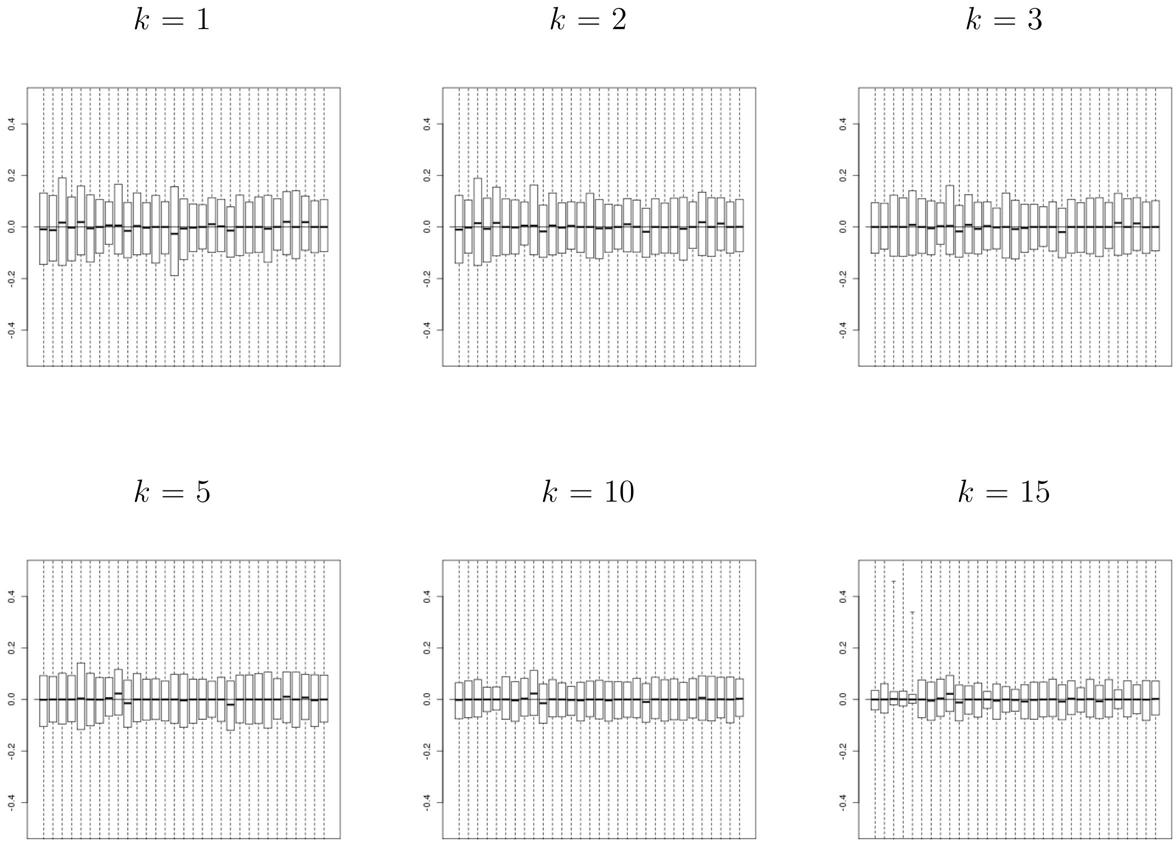
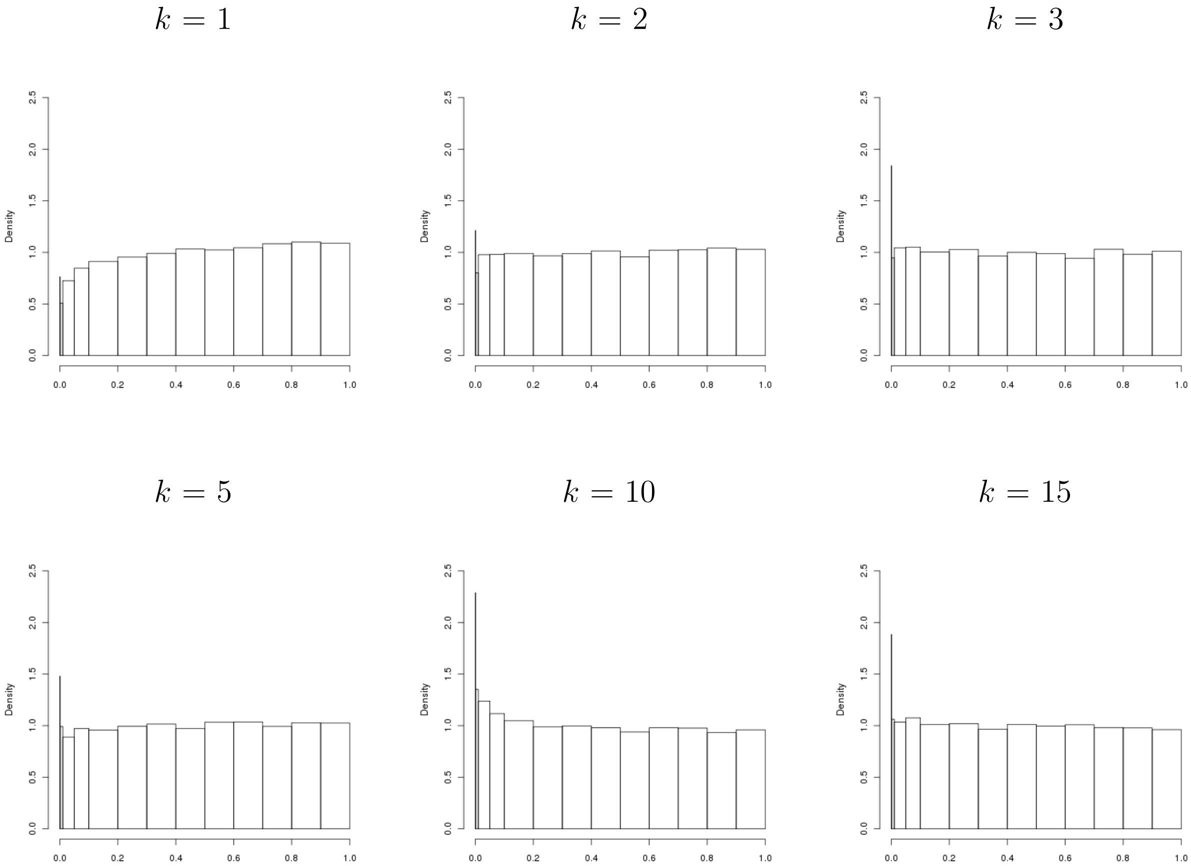


Figure A.13: Alzheimer's study RLE plots after adjustment, using various values of k. Data was preprocessed (BG+QN). The factors were computed by SVD on the housekeeping genes.



$k = 1$ $k = 2$ $k = 3$

$k = 5$ $k = 10$ $k = 15$

Figure A.14: Alzheimer's study P-value histograms after adjustment, using various values of k. Histogram breakpoints are at 0.001, 0.01, 0.05, and 0.1, 0.2, 0.3, etc. Data was preprocessed (BG+QN). The factors were computed by SVD on the housekeeping genes. P-values were computed using Limma.

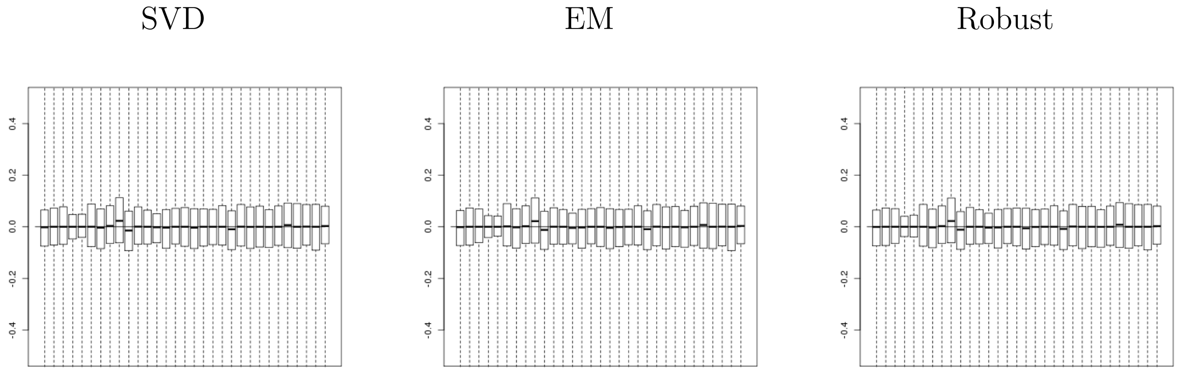


Figure A.15: Alzheimer's study RLE plots after adjustment ($k = 10$), using different methods of factor analysis. The data was preprocessed (BG + QN). Factors were computed using the housekeeping genes.

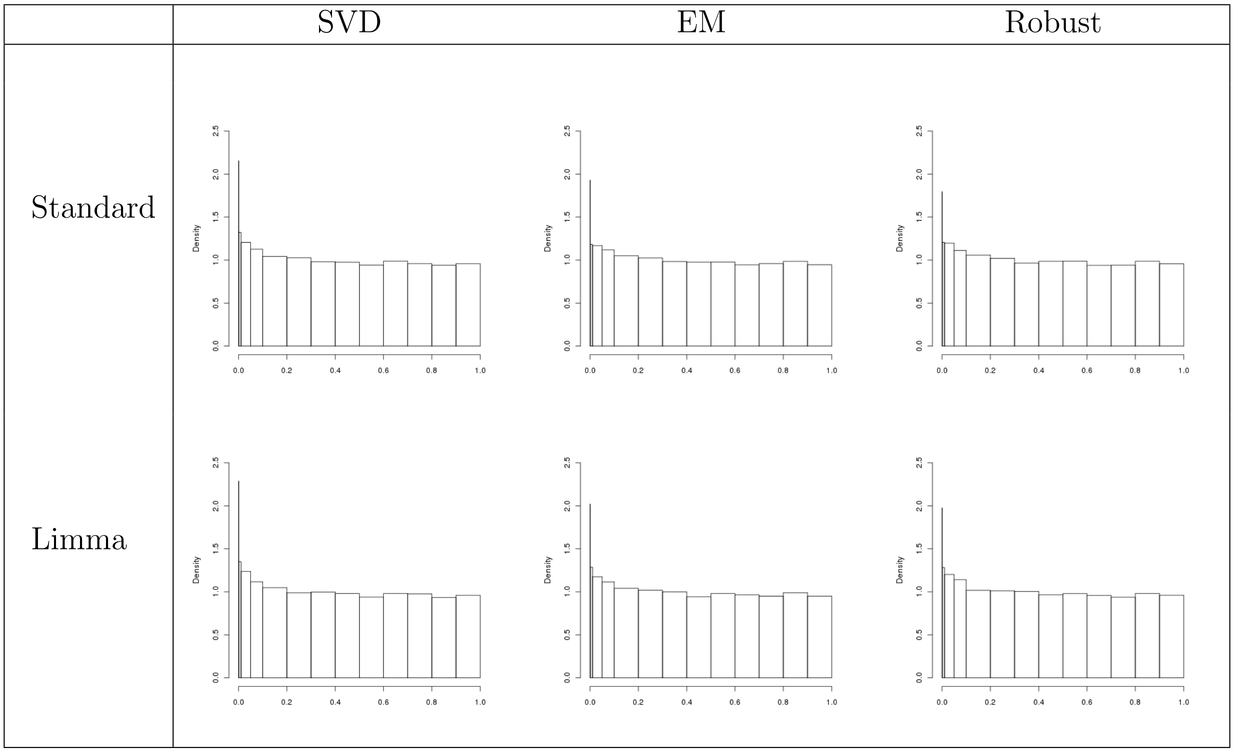


Figure A.16: Alzheimer's study p-value histograms after adjustment ($k = 10$), using different methods of factor analysis, and different methods of computing p-values. The data was preprocessed (BG + QN). Factors were computed using the housekeeping genes.

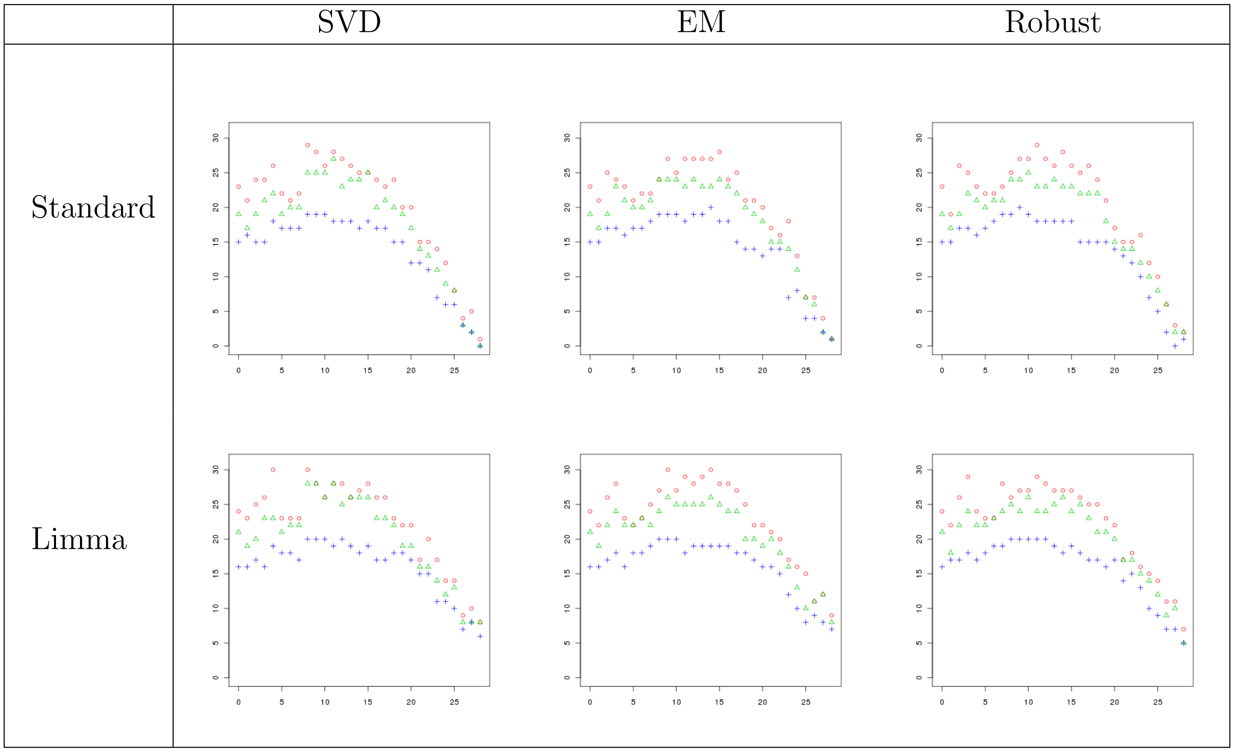


Figure A.17: Comparison of the performance of variants of RUV-2 in the Alzheimer's study. The number of X / Y genes discovered is plotted as a function of k. Genes were ranked by p-value; results are shown for the number of X / Y genes ranked in the top 20 (plus), top 40 (triangle), and top 60 (circle). $k = 0$ corresponds to no adjustment. Data was preprocessed (BG + QN). PCs were computed using the housekeeping genes.

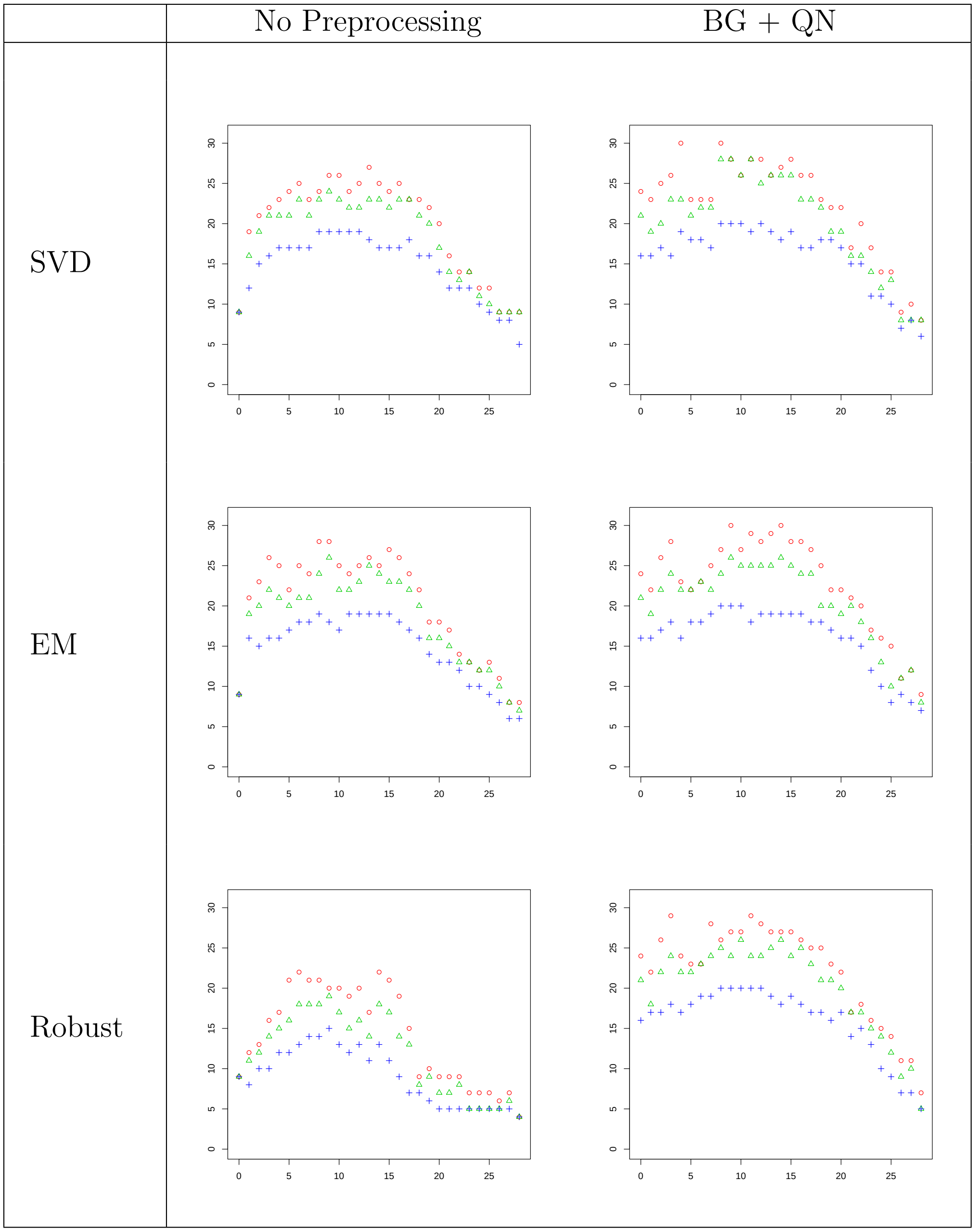
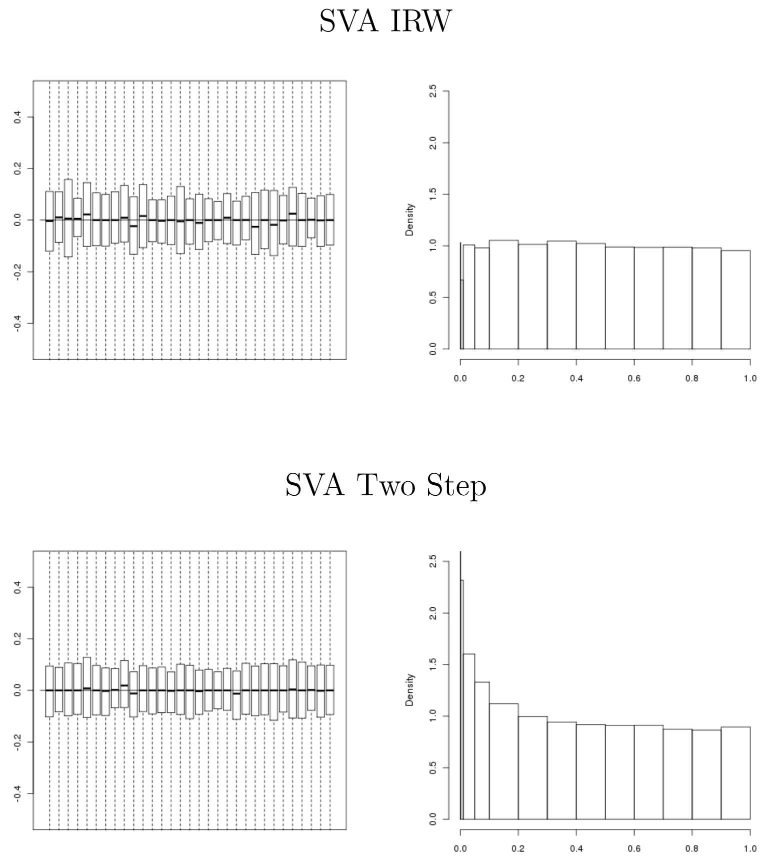


Figure A.18: Comparison of the performance of different factor analysis methods with and without preprocessing in the Alzheimer's study. The number of X / Y genes discovered is plotted as a function of k. Genes were ranked by p-value; results are shown for the number of X / Y genes ranked in the top 20 (plus), top 40 (triangle), and top 60 (circle). $k = 0$ corresponds to no adjustment. PCs were computed using the housekeeping genes. P-values were computed using Limma.



SVA IRW

SVA Two Step

Figure A.19: Alzheimer's study RLE plots and p-value histograms after adjustments by SVA. P-values were computed using Limma. Data was preprocessed (BG + QN).

Unadjusted

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	XIST	X	3×10^{-28}	11	TTTY15	Y	1×10^{-03}	21	KLHL22	22	1
2	XIST	X	8×10^{-27}	12	UTY	Y	1×10^{-03}	22	TAPBPL	12	1
3	DDX3Y	Y	2×10^{-20}	13	CYorf15B	Y	2×10^{-03}	23	LPAR6	13	1
4	RPS4Y1	Y	3×10^{-15}	14	DDX3Y	Y	8×10^{-02}	24	PRKAB1	12	1
5	KDM5D	Y	2×10^{-11}	15	ZBED1	X Y	1	25	GPX3	5	1
6	EIF1AY	Y	1×10^{-07}	16	PLCXD1	X Y	1	26	HDHD1A	X	1
7	USP9Y	Y	8×10^{-07}	17	CDK10	16	1	27	KDM6A	X	1
8	EIF1AY	Y	1×10^{-06}	18	GPX3	5	1	28	CD99	X Y	1
9	NLGN4Y	Y	5×10^{-06}	19	NCKAP1	2	1	29	PRKAR1B	7	1
10	NCRNA00185	Y	1×10^{-05}	20	UBAP2L	1	1	30	MARCH1	4	1

RUV-2 (SVD), $k = 10$, housekeeping genes

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	XIST	X	1×10^{-21}	11	TTTY15	Y	1×10^{-05}	21	COL4A5	X	7×10^{-02}
2	XIST	X	1×10^{-21}	12	CYorf15B	Y	1×10^{-05}	22	FBXO9	6	7×10^{-02}
3	DDX3Y	Y	4×10^{-16}	13	CD99	X Y	5×10^{-04}	23	PLCL1	2	7×10^{-02}
4	RPS4Y1	Y	1×10^{-15}	14	NA	X	3×10^{-03}	24	DDX3X	X	0.1
5	KDM5D	Y	1×10^{-11}	15	CD99	X Y	3×10^{-03}	25	KDM6A	X	0.1
6	USP9Y	Y	8×10^{-09}	16	UTY	Y	1×10^{-02}	26	DOPEY1	6	0.1
7	EIF1AY	Y	4×10^{-08}	17	DDX3Y	Y	3×10^{-02}	27	DDX3X	X	0.1
8	EIF1AY	Y	2×10^{-07}	18	RPS4X	X	3×10^{-02}	28	ZBED1	X Y	0.1
9	NLGN4Y	Y	4×10^{-07}	19	KDM6A	X	4×10^{-02}	29	RARS	5	0.2
10	NCRNA00185	Y	1×10^{-06}	20	PLCXD1	X Y	6×10^{-02}	30	WNT8B	10	0.2

Table A.3: Comparison of gene rankings before and after adjustment (SVD, $k = 10$) in the Alzheimer's study. The data has been preprocessed (BG + NM). The p-values were computed using Limma.

A.5 Additional TCGA Figures

A.5.0.7 Exon Array Data

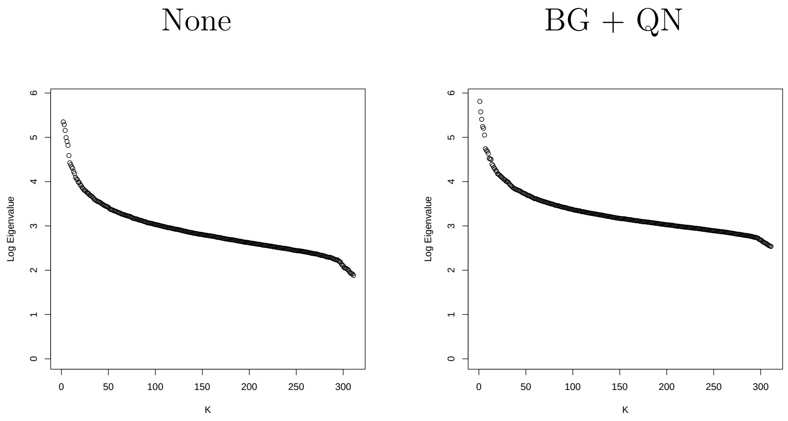


Figure A.20: TCGA exon array scree plots with and without preprocessing. Left: No preprocessing; Right: Background correction / quantile normalization. All genes were included in the eigenanalysis.

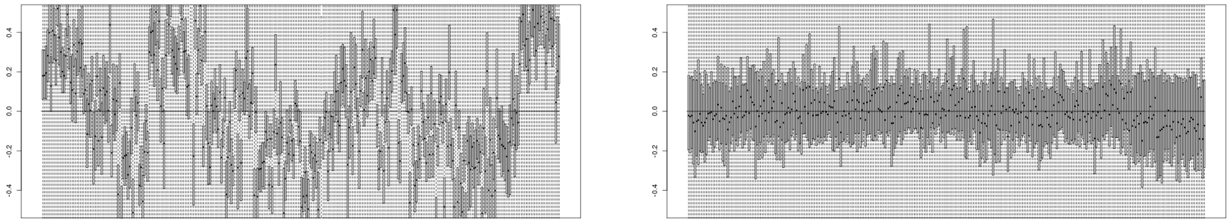
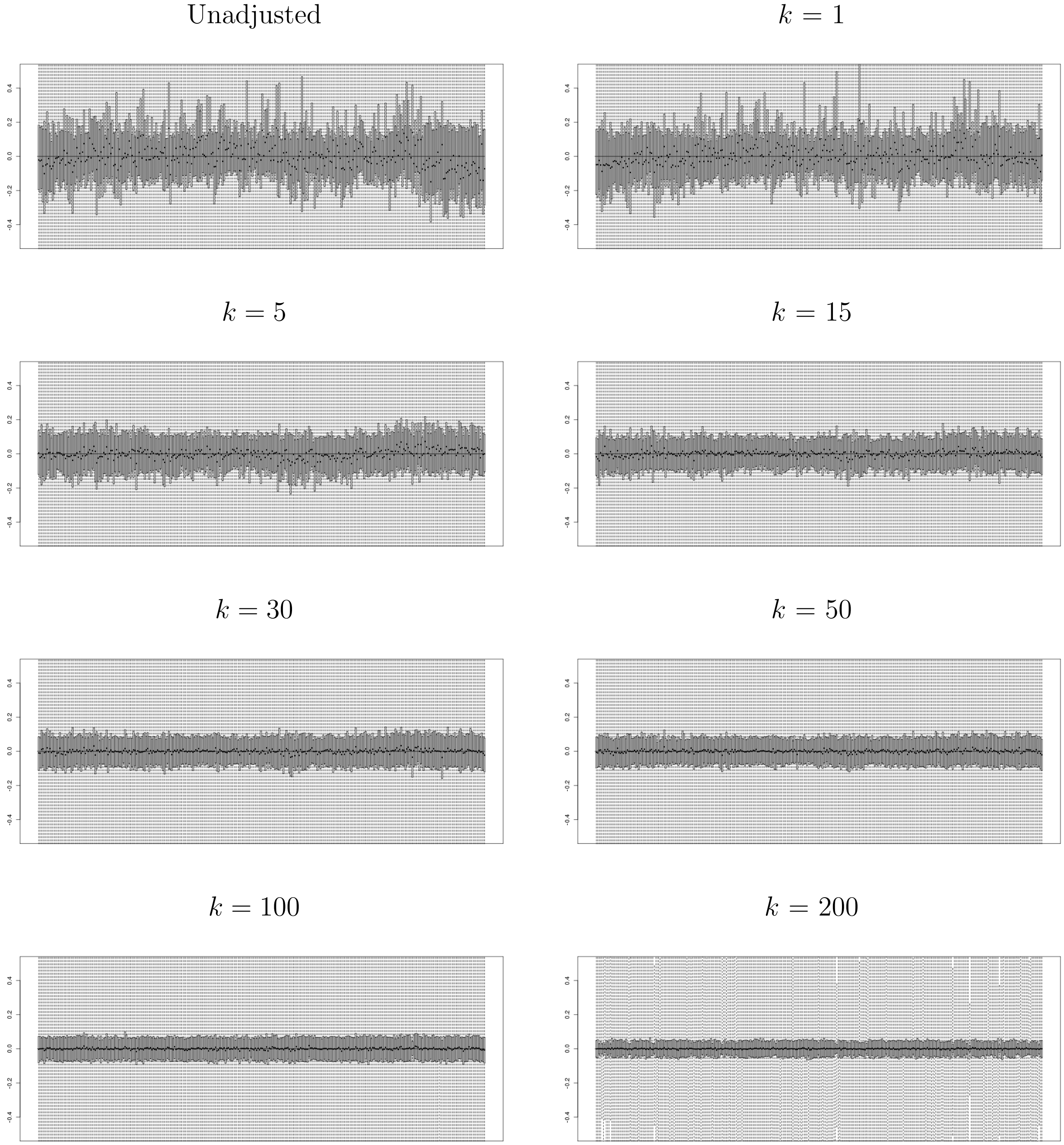


Figure A.21: TCGA exon array RLE plots with and without preprocessing. Left: No preprocessing; Right: Background correction / quantile normalization.



Unadjusted

$k = 1$

$k = 5$

$k = 15$

$k = 30$

$k = 50$

$k = 100$

$k = 200$

Figure A.22: TCGA exon array RLE plots after adjustment, using various values of k. Data was preprocessed (BG+QN). The factors were computed by SVD on the housekeeping genes.

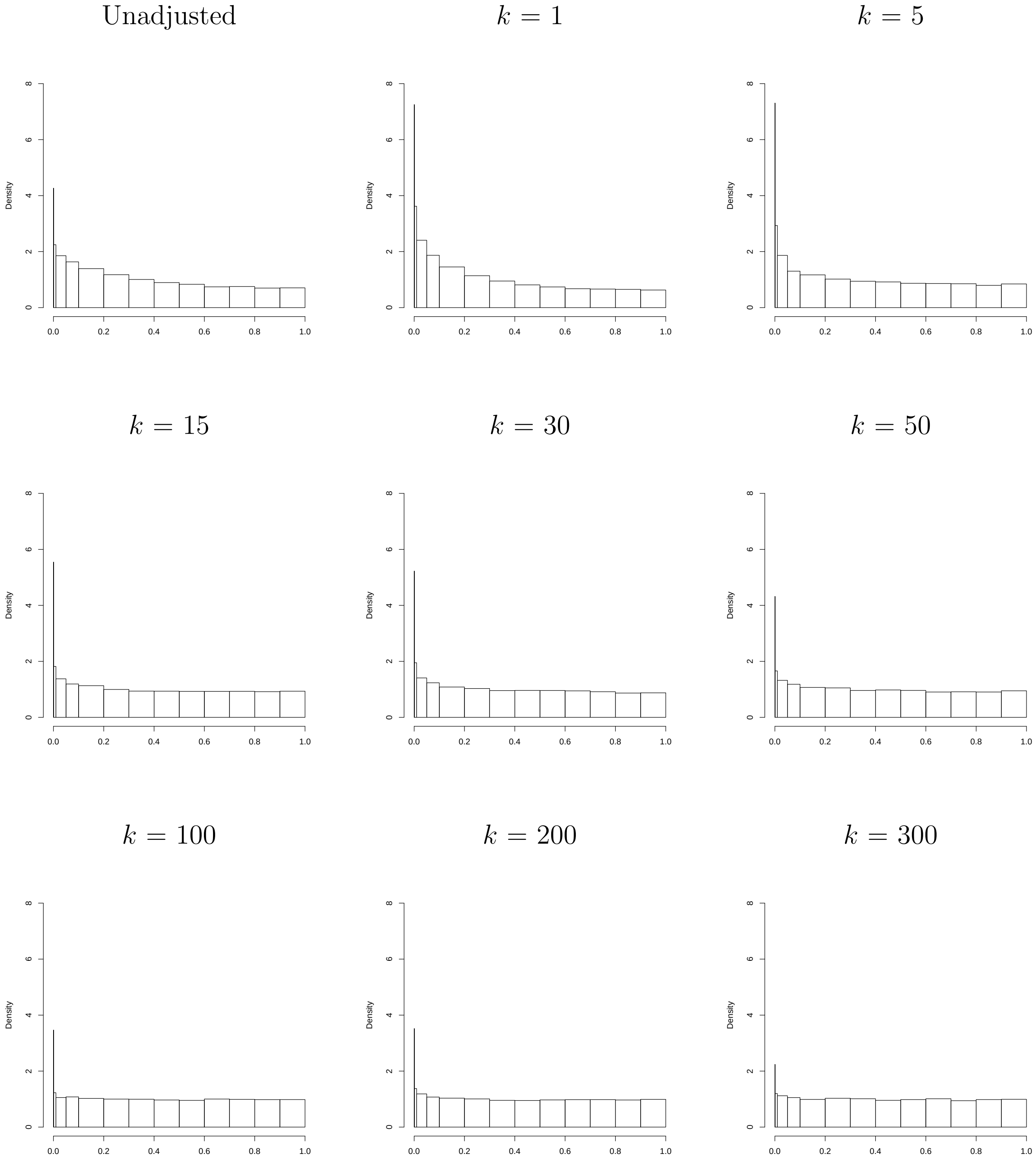


Figure A.23: TCGA exon array p-value histograms after adjustment, using various values of k. Histogram breakpoints are at 0.001, 0.01, 0.05, and 0.1, 0.2, 0.3, etc. Data was preprocessed (BG+QN). The factors were computed by SVD on the housekeeping genes. P-values were computed using Limma.

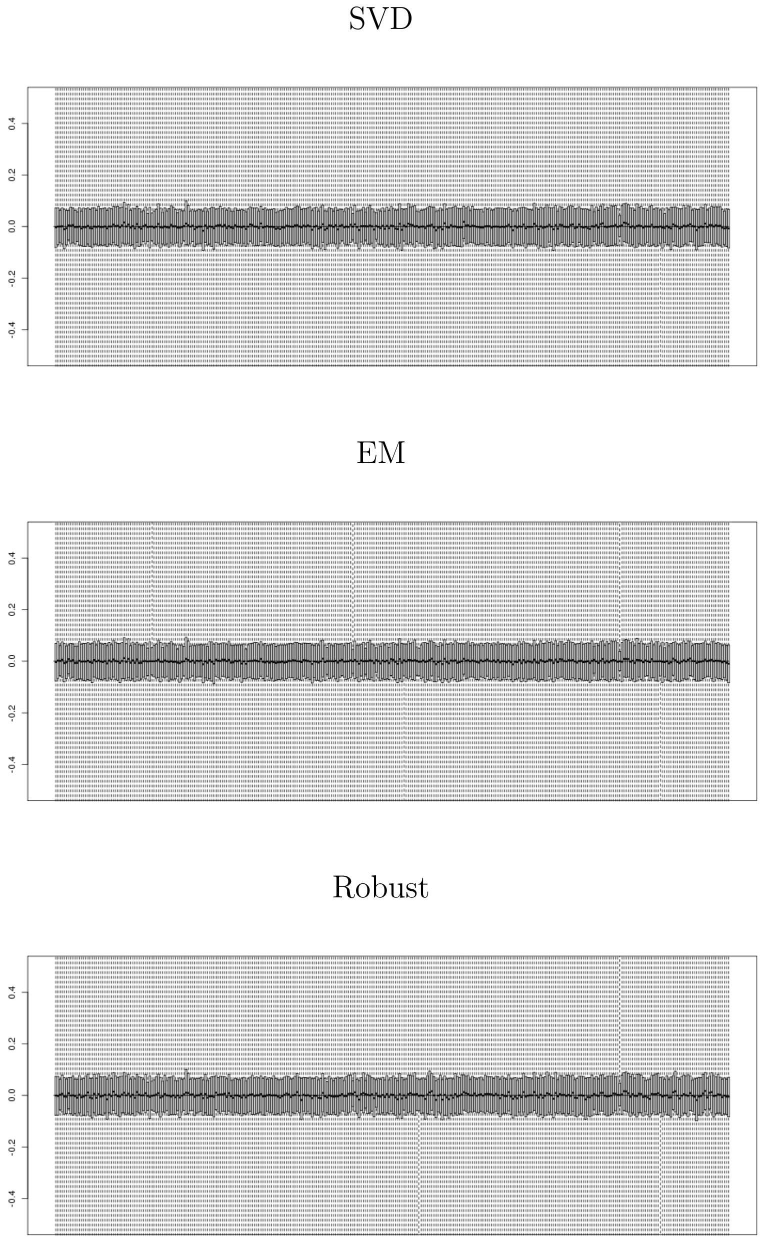


Figure A.24: TCGA exon array RLE plots after adjustment ($k = 100$), using different methods of factor analysis. The data was preprocessed (BG + QN). The factors were computed using housekeeping genes.

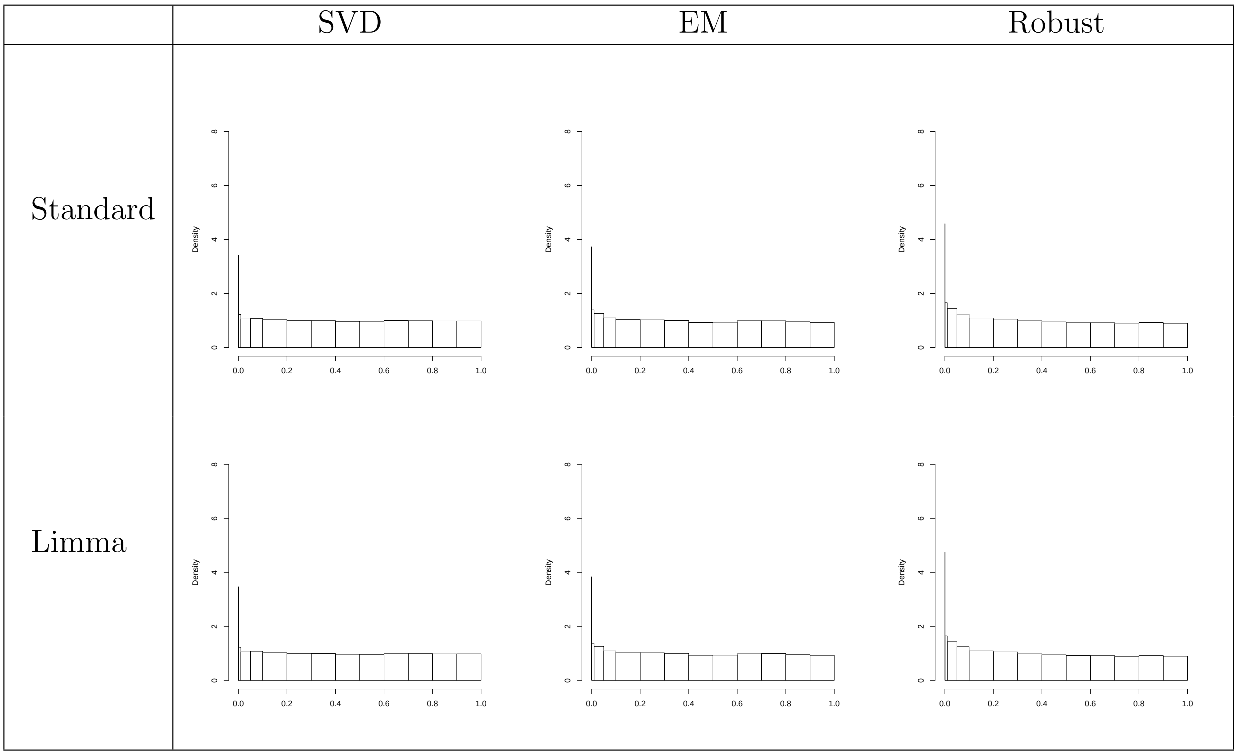


Figure A.25: TCGA exon array p-value histograms after adjustment ($k = 100$), using different methods of factor analysis, and different methods of computing p-values. The data was preprocessed (BG + QN). The factors were computed using housekeeping genes.

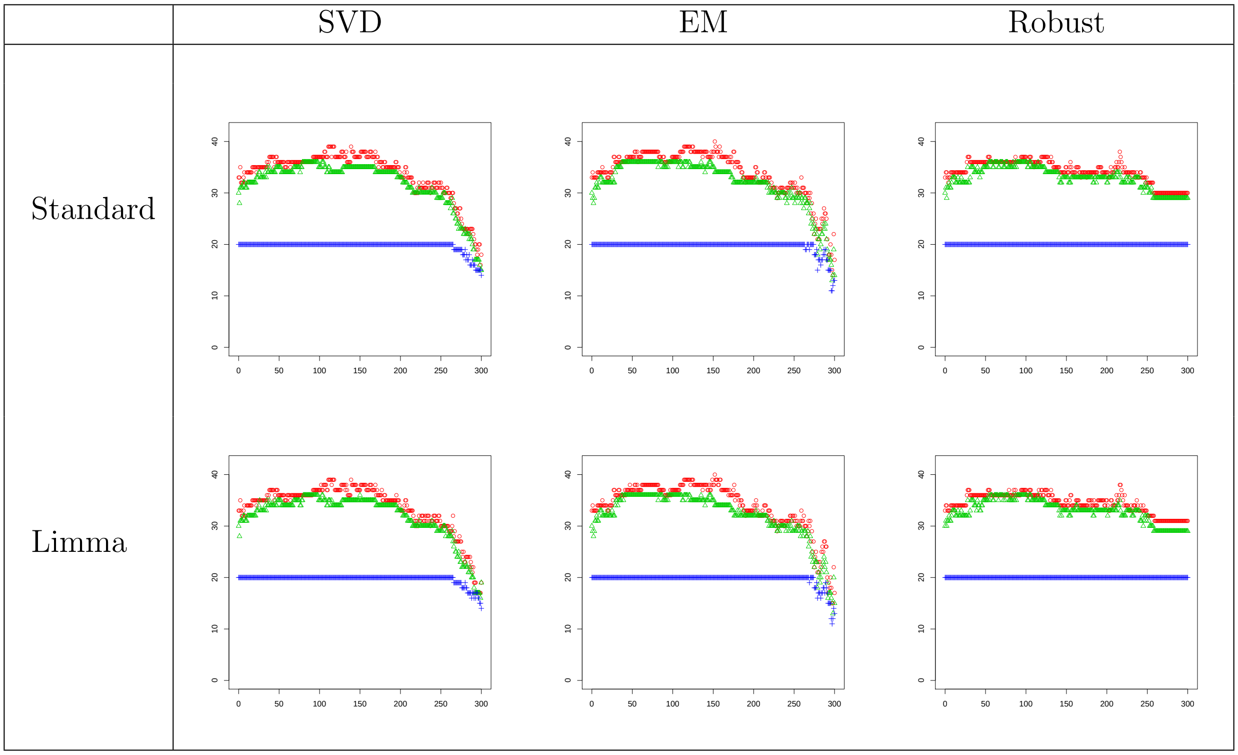


Figure A.26: Comparison of the performance of variants of RUV-2 in the TCGA exon array data. The number of X / Y genes discovered is plotted as a function of k. Genes were ranked by p-value; results are shown for the number of X / Y genes ranked in the top 20 (plus), top 40 (triangle), and top 60 (circle). $k = 0$ corresponds to no adjustment. Data was preprocessed (BG + NM). PCs were computed using the housekeeping genes.

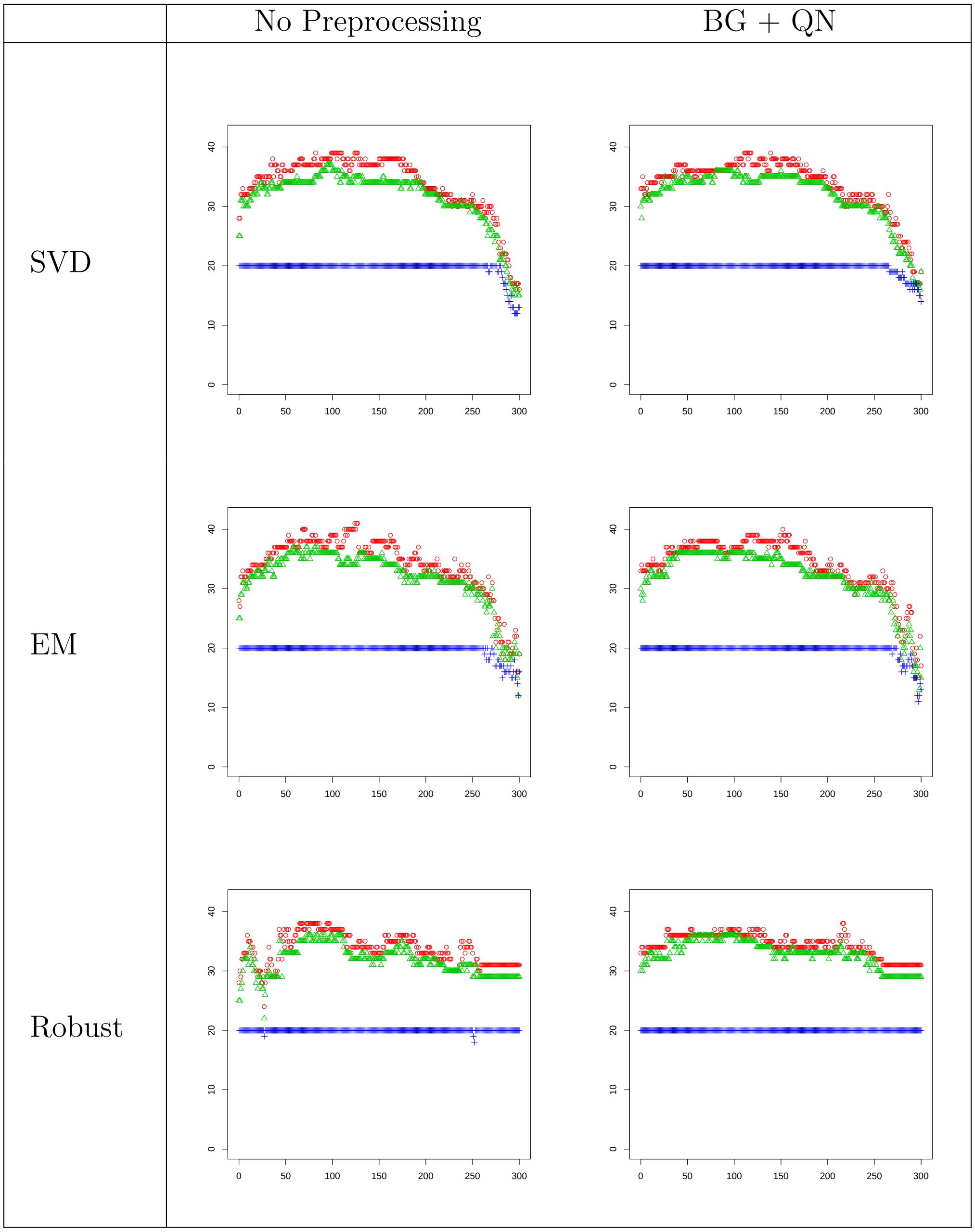


Figure A.27: Comparison of the performance of different factor analysis methods with and without preprocessing in the TCGA exon array data. The number of X / Y genes discovered is plotted as a function of k. Genes were ranked by p-value; results are shown for the number of X / Y genes ranked in the top 20 (plus), top 40 (triangle), and top 60 (circle). $k = 0$ corresponds to no adjustment. PCs were computed using the housekeeping genes. P-values were computed using Limma.

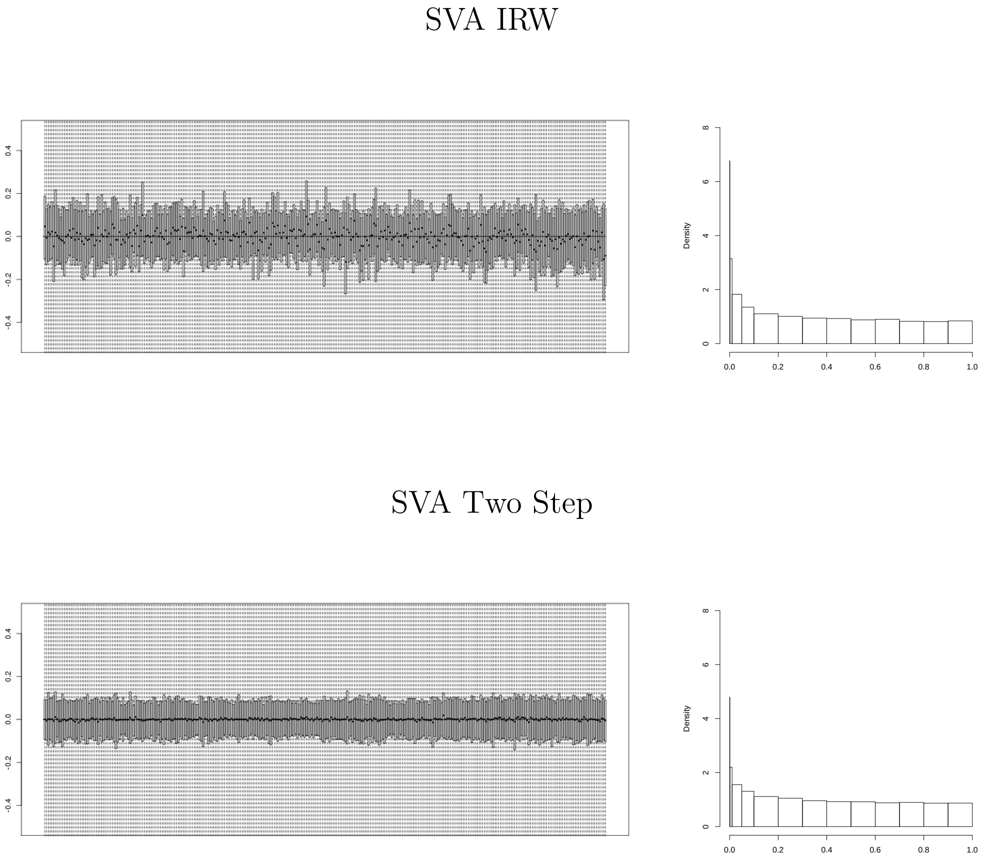


Figure A.28: TCGA exon array RLE plots and p-value histograms after adjustments by SVA. The data was preprocessed. The p-values were computed using Limma.

Unadjusted

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	RPS4Y1	Y	1×10^{-159}	16	JARID1C	X	7×10^{-21}	31	LRRC47	1	3×10^{-2}
2	DDX3Y	Y	6×10^{-137}	17	ZFX	X	3×10^{-18}	32	SYAP1	X	3×10^{-2}
3	EIF1AY	Y	2×10^{-131}	18	UTX	X	1×10^{-13}	33	MYLK	3	3×10^{-2}
4	UTY	Y	3×10^{-125}	19	LOC554203	X	1×10^{-12}	34	IHPK1	3	5×10^{-2}
5	USP9Y	Y	3×10^{-120}	20	HDHD1A	X	1×10^{-12}	35	HTATIP2	11	5×10^{-2}
6	CYorf15A	Y	3×10^{-117}	21	CXorf15	X	5×10^{-8}	36	NLGN4X	X	6×10^{-2}
7	ZFY	Y	4×10^{-113}	22	DDX3X	X	9×10^{-7}	37	ZRSR2	X	6×10^{-2}
8	JARID1D	Y	2×10^{-112}	23	EIF1AX	X	2×10^{-6}	38	PRKCH	14	6×10^{-2}
9	RPS4Y2	Y	3×10^{-94}	24	RPS4X	X	1×10^{-5}	39	EDG3	na	7×10^{-2}
10	NLGN4Y	Y	1×10^{-92}	25	SRY	Y	2×10^{-4}	40	NUPL2	7	7×10^{-2}
11	CYorf15B	Y	4×10^{-81}	26	STIM2	4	4×10^{-3}	41	SH3PXD2A	10	7×10^{-2}
12	TMSB4Y	Y	7×10^{-48}	27	EIF2S3	X	5×10^{-3}	42	STS	X	7×10^{-2}
13	TTTY10	Y	1×10^{-47}	28	ZFP2	5	5×10^{-3}	43	PAPSS2	10	7×10^{-2}
14	PRKY	Y	5×10^{-31}	29	GEMIN8	X	7×10^{-3}	44	ABCA11	4	7×10^{-2}
15	TTTY14	Y	3×10^{-26}	30	MICAL2	11	1×10^{-2}	45	GGTLA1	22	7×10^{-2}

RUV-2 (SVD), $k = 100$, housekeeping genes

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	RPS4Y1	Y	1×10^{-104}	16	TTTY14	Y	3×10^{-30}	31	CA5B	X	4×10^{-5}
2	DDX3Y	Y	2×10^{-88}	17	ZFX	X	3×10^{-28}	32	GEMIN8	X	4×10^{-5}
3	UTY	Y	2×10^{-83}	18	PRKY	Y	9×10^{-21}	33	ORC4L	2	2×10^{-3}
4	EIF1AY	Y	5×10^{-81}	19	HDHD1A	X	1×10^{-19}	34	CHM	X	3×10^{-3}
5	USP9Y	Y	5×10^{-81}	20	DDX3X	X	4×10^{-19}	35	GPR88	1	4×10^{-3}
6	CYorf15A	Y	2×10^{-73}	21	RPS4X	X	4×10^{-17}	36	FUNDC1	X	5×10^{-3}
7	ZFY	Y	3×10^{-73}	22	EIF2S3	X	9×10^{-17}	37	SRY	Y	1×10^{-2}
8	JARID1D	Y	5×10^{-71}	23	EIF1AX	X	1×10^{-16}	38	CENPA	2	2×10^{-2}
9	RPS4Y2	Y	8×10^{-71}	24	SYAP1	X	6×10^{-16}	39	INSR	19	3×10^{-2}
10	NLGN4Y	Y	3×10^{-60}	25	LOC554203	X	1×10^{-15}	40	MYL3	3	8×10^{-2}
11	CYorf15B	Y	2×10^{-57}	26	CXorf15	X	3×10^{-15}	41	FMNL2	2	8×10^{-2}
12	JARID1C	X	2×10^{-38}	27	SMC1A	X	4×10^{-15}	42	RNF213	17	8×10^{-2}
13	TMSB4Y	Y	1×10^{-34}	28	STS	X	6×10^{-13}	43	GIP	17	9×10^{-2}
14	UTX	X	1×10^{-31}	29	USP9X	X	1×10^{-5}	44	UBE1	X	9×10^{-2}
15	TTTY10	Y	3×10^{-30}	30	ZRSR2	X	2×10^{-5}	45	TRIM23	5	0.1

Table A.4: Comparison of gene rankings before and after adjustment (SVD, $k = 100$) using the TCGA exon array data. The data has been preprocessed (BG + NM). The p-values were computed using Limma.

A.5.0.8 HG-U133a Data

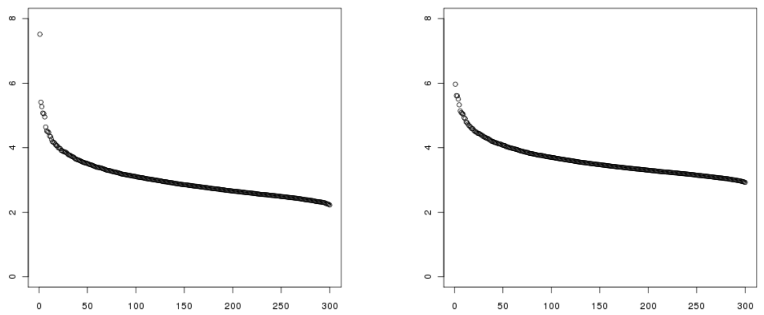
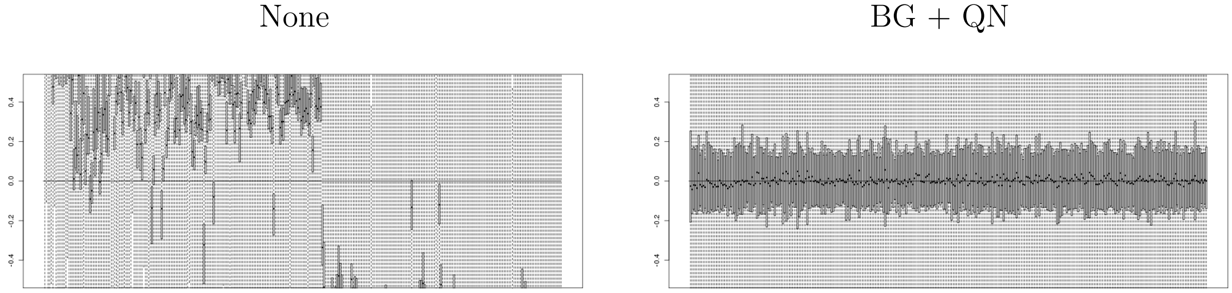


Figure A.29: TCGA HT HG-133A scree plots with and without preprocessing. Left: No preprocessing; Right: Background correction / quantile normalization. All genes were included in the eigenanalysis.


None

BG + QN

Figure A.30: TCGA HT HG-133A RLE plots with and without preprocessing. Left: No preprocessing; Right: Background correction / quantile normalization.

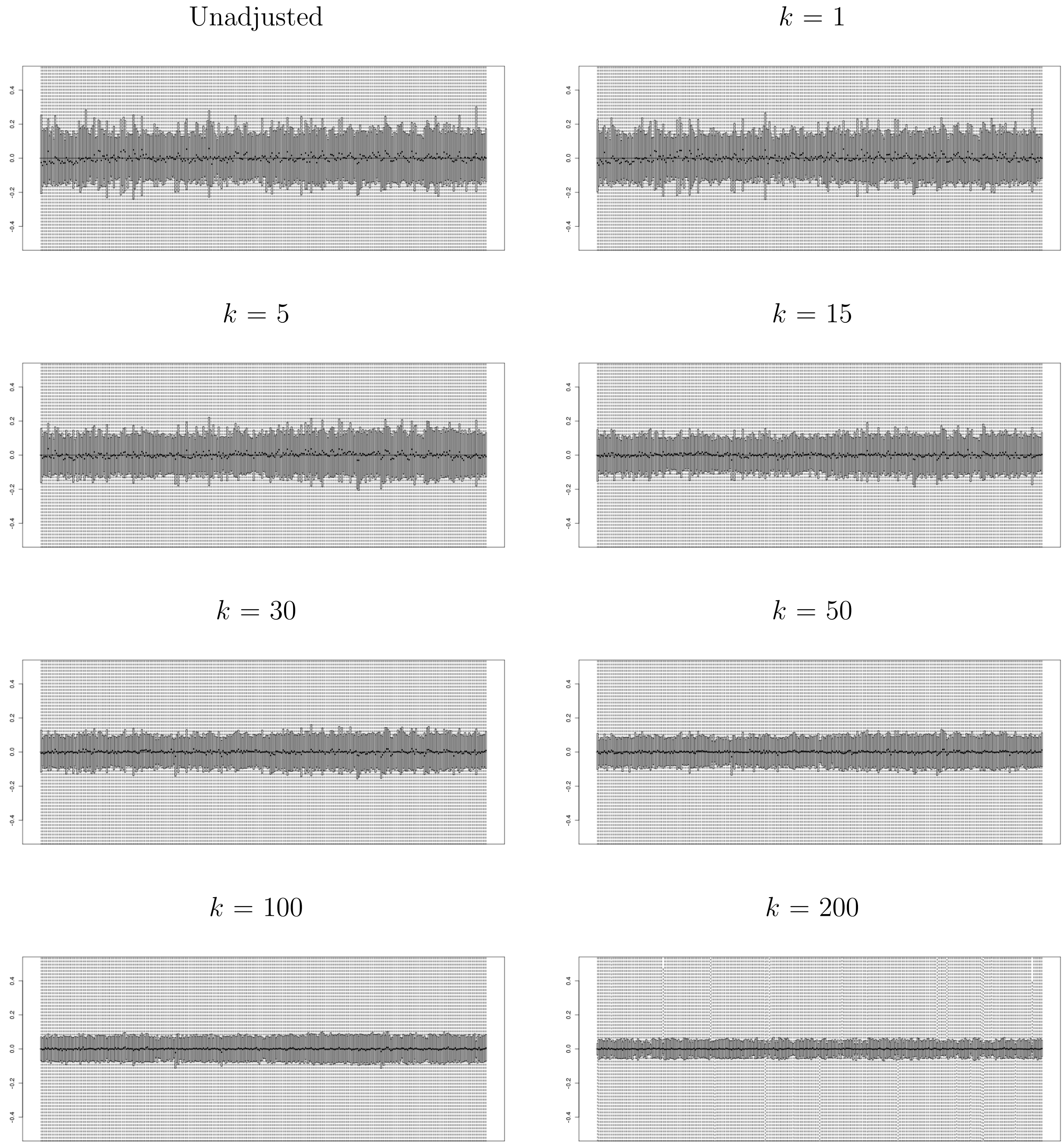
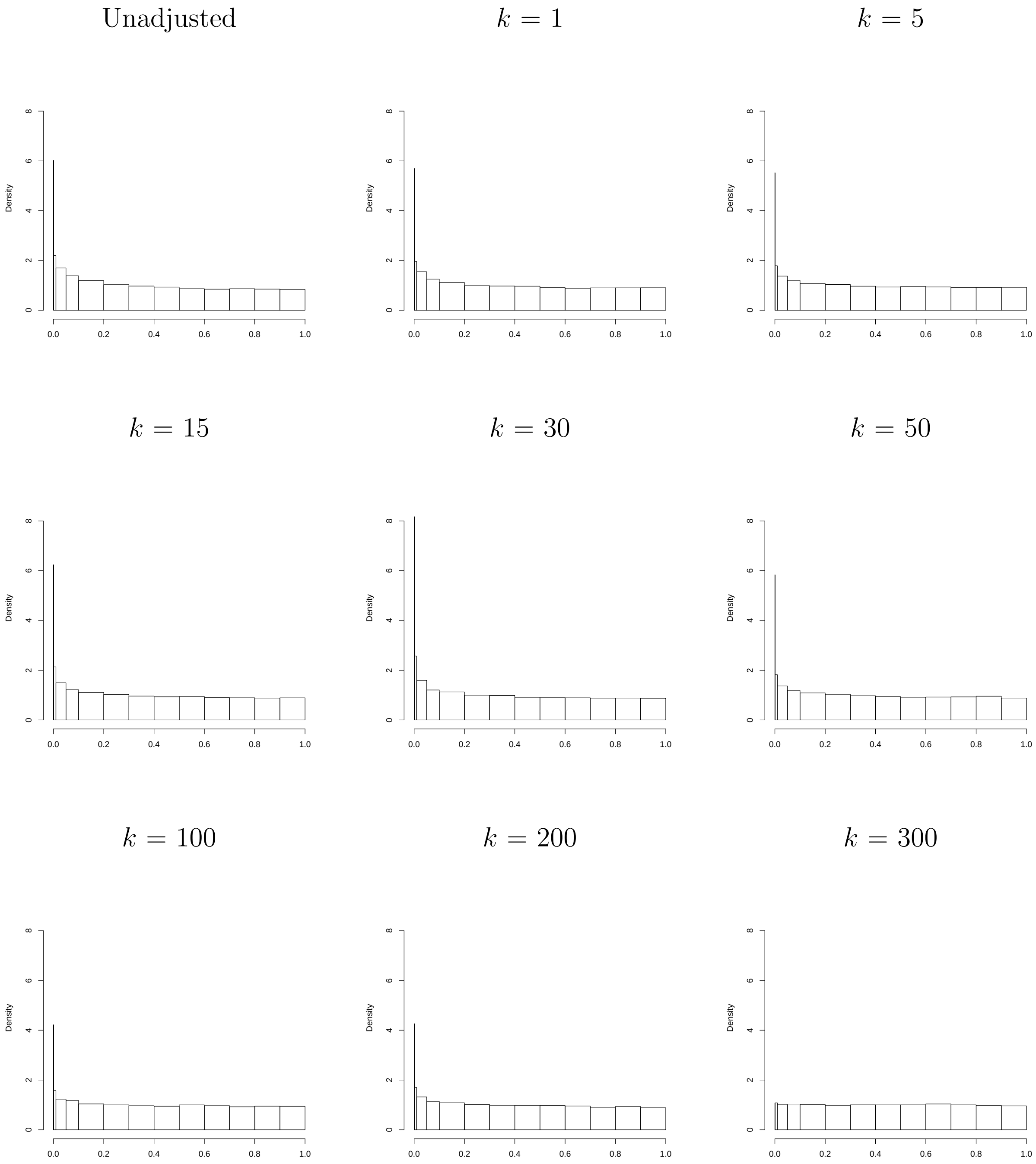


Figure A.31: TCGA HT HG-133A RLE plots after adjustment, using various values of k. Data was preprocessed (BG+QN). The factors were computed by SVD on the housekeeping genes.


Unadjusted

$k = 1$

$k = 5$

$k = 15$

$k = 30$

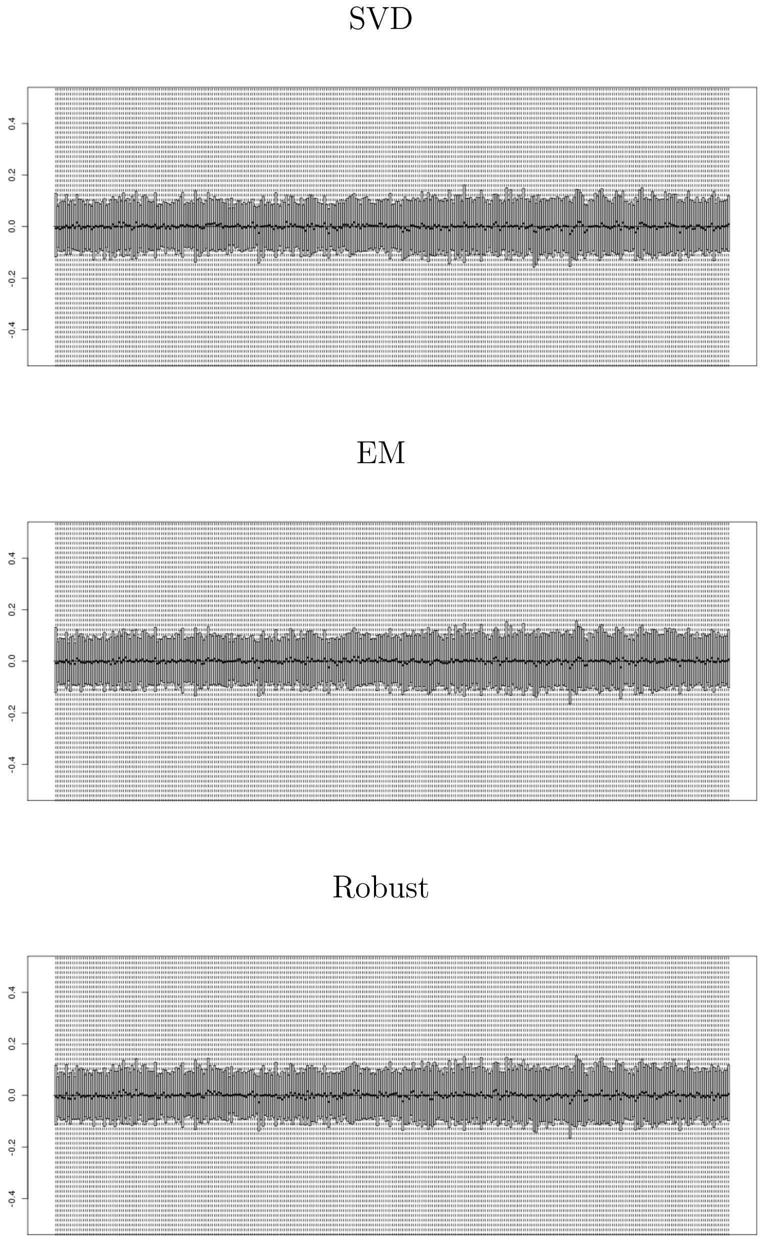
$k = 50$

$k = 100$

$k = 200$

$k = 300$

Figure A.32: TCGA HT HG-133A p-value histograms after adjustment, using various values of k. Histogram breakpoints are at 0.001, 0.01, 0.05, and 0.1, 0.2, 0.3, etc. Data was preprocessed (BG+QN). The factors were computed by SVD on the housekeeping genes. P-values were computed using Limma.



SVD

EM

Robust

Figure A.33: TCGA HT HG-133A RLE plots after adjustment ($k = 30$), using different methods of factor analysis. The data was preprocessed (BG + QN). The factors were computed using housekeeping genes.

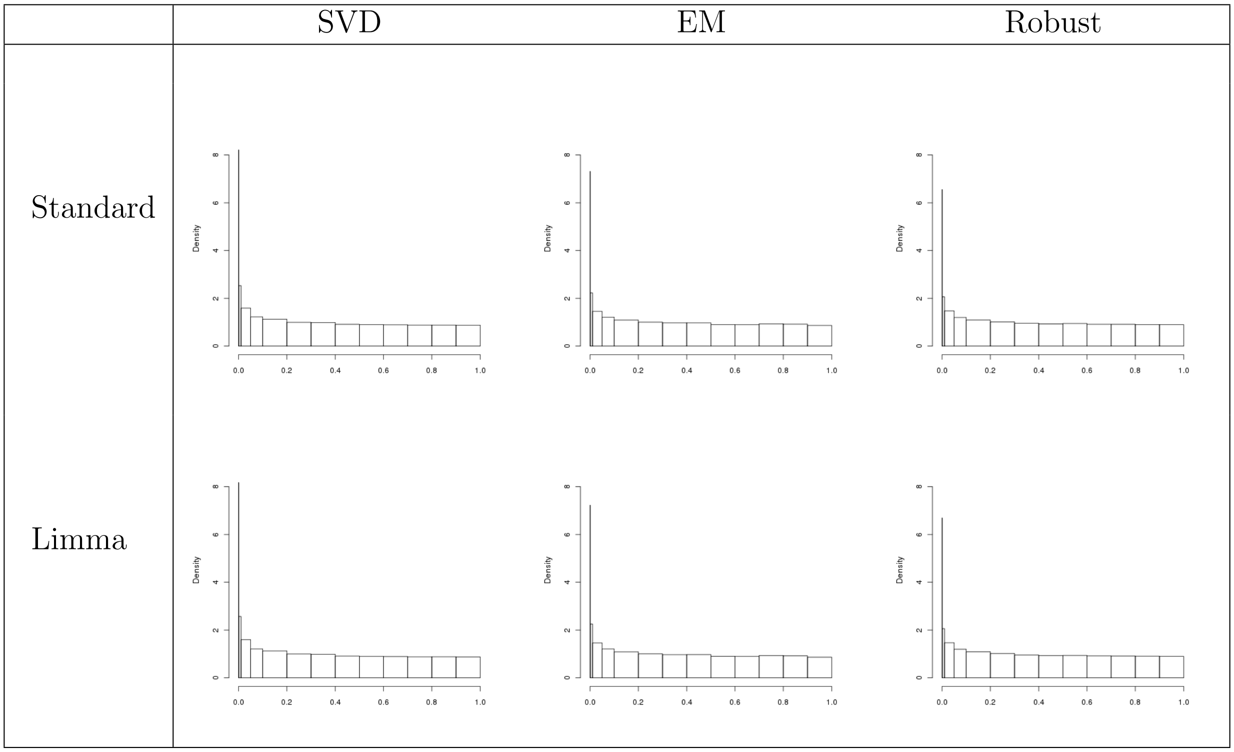


Figure A.34: TCGA HT HG-133A p-value histograms after adjustment ($k = 30$), using different methods of factor analysis, and different methods of computing p-values. The data was preprocessed (BG + QN). The factors were computed using housekeeping genes.

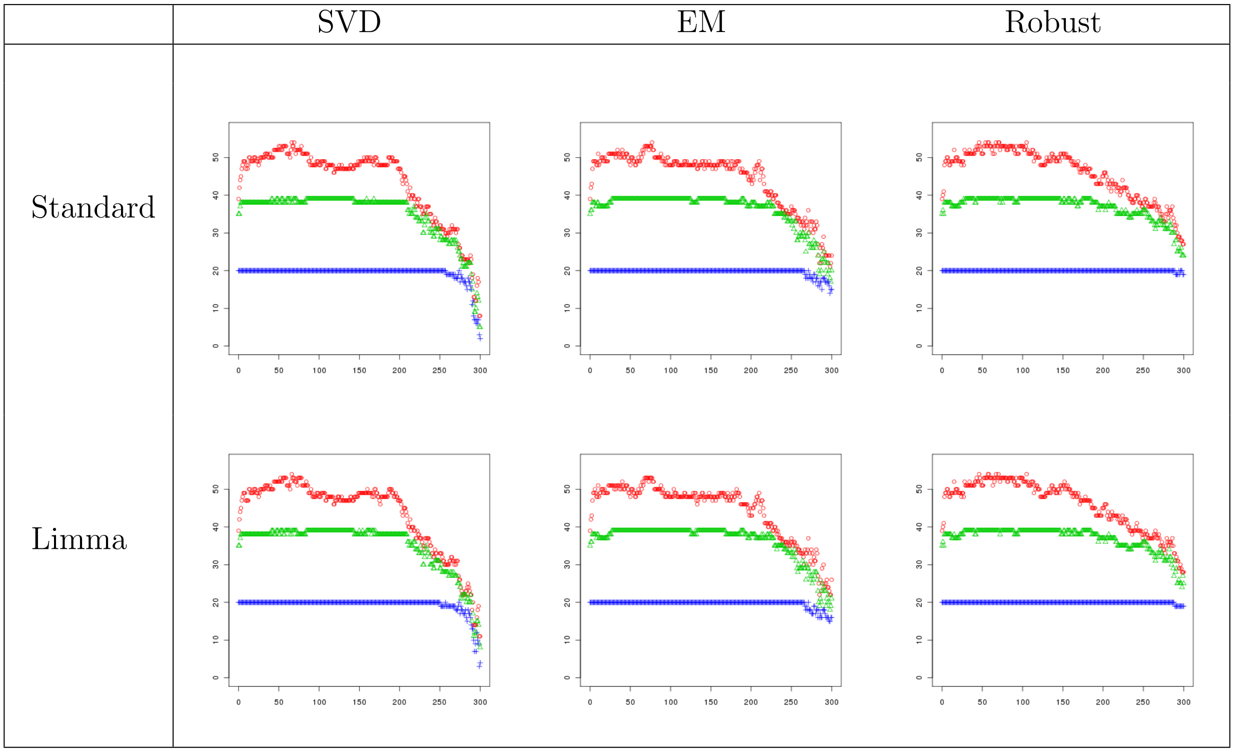


Figure A.35: Comparison of the performance of variants of RUV-2 in the TCGA HT HG-U133A data. The number of X / Y genes discovered is plotted as a function of k. Genes were ranked by p-value; results are shown for the number of X / Y genes ranked in the top 20 (plus), top 40 (triangle), and top 60 (circle). $k = 0$ corresponds to no adjustment. Data was preprocessed (BG + NM). PCs were computed using the housekeeping genes.

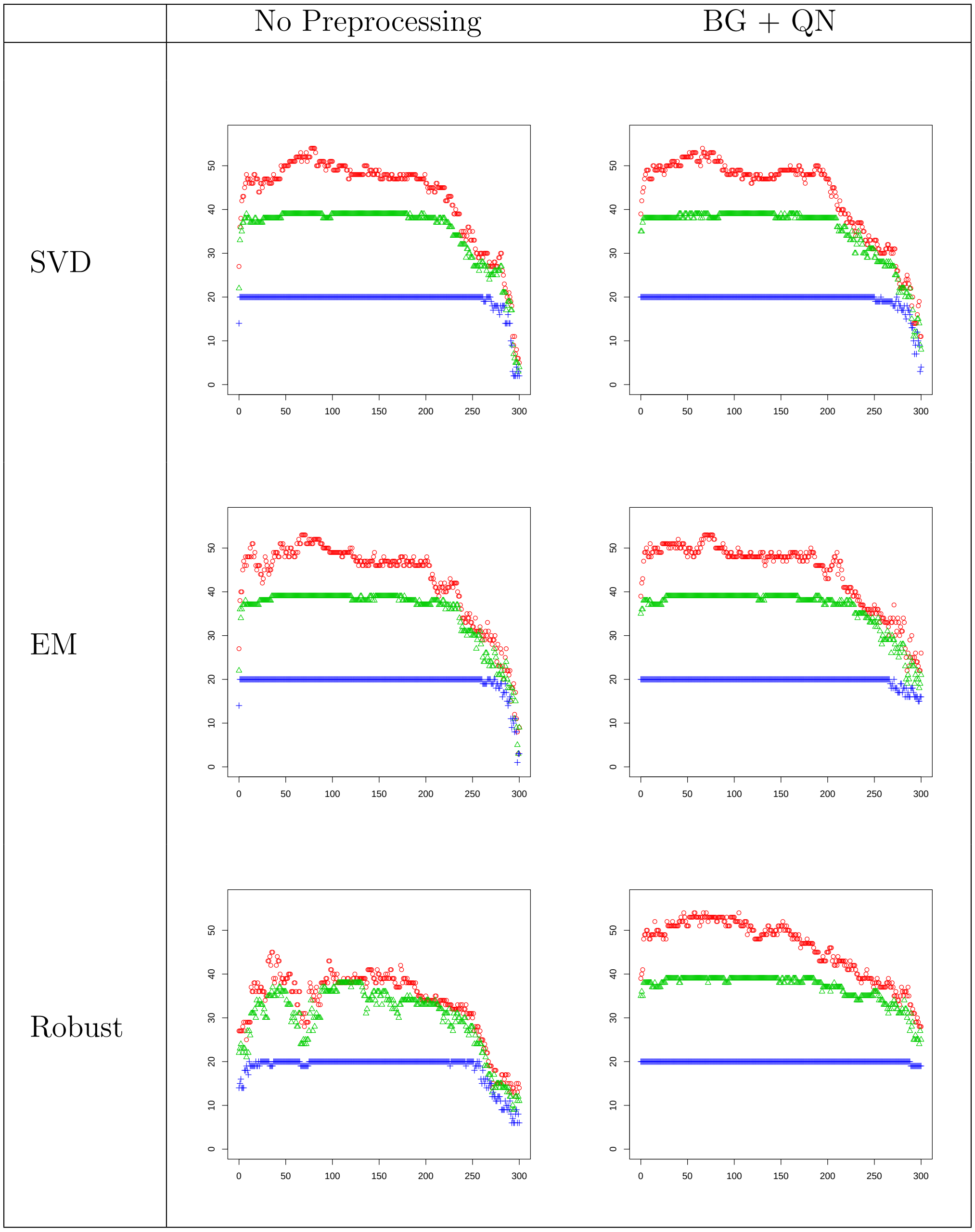
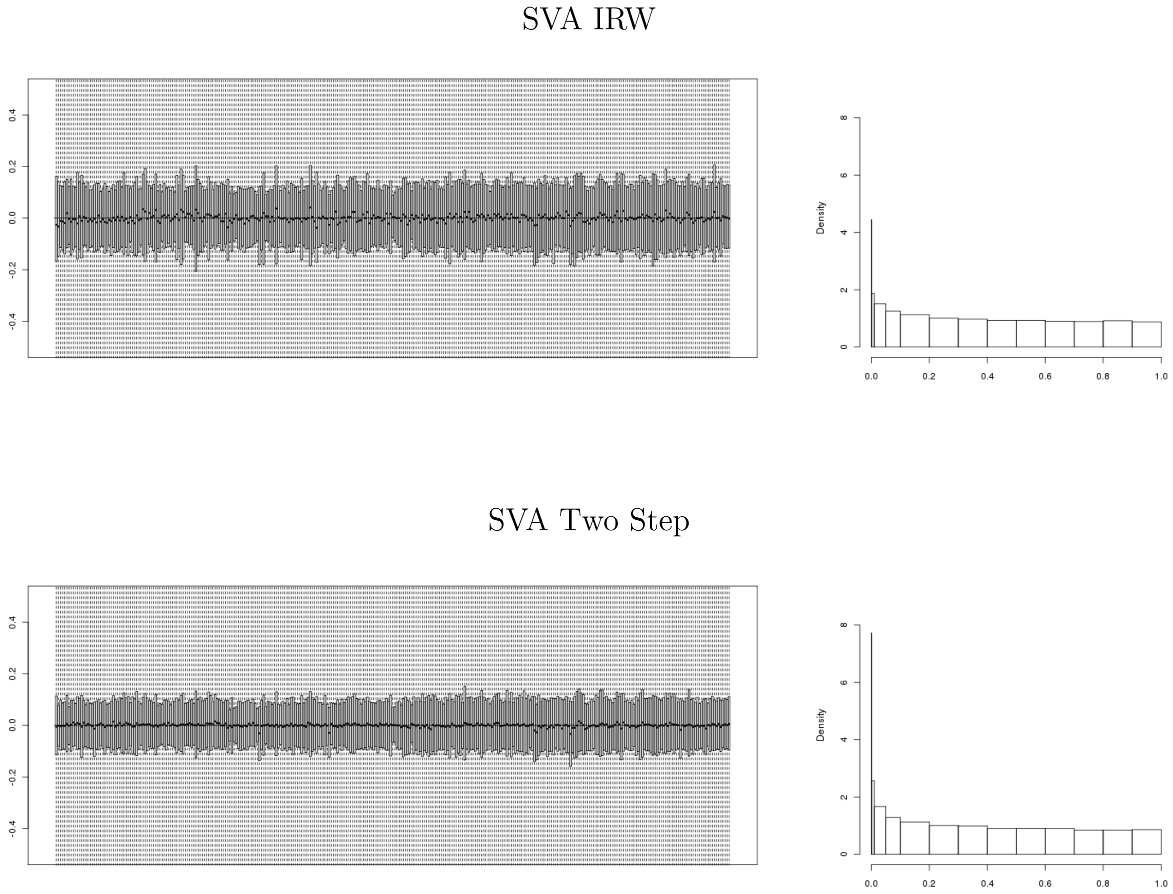


Figure A.36: Comparison of the performance of different factor analysis methods with and without preprocessing in the TCGA HT HG-U133A data. The number of X / Y genes discovered is plotted as a function of k. Genes were ranked by p-value; results are shown for the number of X / Y genes ranked in the top 20 (plus), top 40 (triangle), and top 60 (circle). $k = 0$ corresponds to no adjustment. PCs were computed using the housekeeping genes. P-values were computed using Limma.


SVA IRW

SVA Two Step

Figure A.37: TCGA HT HG-U133A RLE plots and p-value histograms after adjustments by SVA. The data was preprocessed (BG + QN). P-values were computed using Limma.

Unadjusted

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	RPS4Y1	Y	2×10^{-157}	21	KDM6A	X	4×10^{-12}	41	MICAL2	11	3×10^{-03}
2	DDX3Y	Y	5×10^{-139}	22	TMSB4Y	Y	2×10^{-11}	42	GPR88	1	4×10^{-03}
3	XIST	X	3×10^{-113}	23	KDM6A	X	2×10^{-09}	43	ZRSR2	X	4×10^{-03}
4	EIF1AY	Y	5×10^{-111}	24	KDM5C	X	9×10^{-09}	44	CCDC71	3	5×10^{-03}
5	XIST	X	1×10^{-109}	25	RPS4X	X	1×10^{-08}	45	FILIP1L	3	6×10^{-03}
6	KDM5D	Y	5×10^{-94}	26	CD99	X Y	5×10^{-08}	46	GEMIN8	X	7×10^{-03}
7	EIF1AY	Y	7×10^{-93}	27	KDM6A	X	1×10^{-07}	47	CADM4	19	1×10^{-02}
8	DDX3Y	Y	6×10^{-89}	28	CD99	X Y	2×10^{-07}	48	LASS4	19	2×10^{-02}
9	UTY	Y	1×10^{-77}	29	SRY	Y	2×10^{-07}	49	CA5BP	X	2×10^{-02}
10	TTTY15	Y	4×10^{-77}	30	RPS4X	X	1×10^{-06}	50	NA	NA	2×10^{-02}
11	NLGN4Y	Y	1×10^{-64}	31	PRKY	Y	3×10^{-06}	51	TMEM147	19	3×10^{-02}
12	NCRNA00185	Y	4×10^{-64}	32	NA	NA	4×10^{-06}	52	DDX3X	X	3×10^{-02}
13	CYorf15B	Y	3×10^{-41}	33	EIF1AX	X	4×10^{-06}	53	ZFYVE9	1	4×10^{-02}
14	USP9Y	Y	2×10^{-39}	34	TRIM31	6	1×10^{-04}	54	HTATIP2	11	4×10^{-02}
15	HDHD1A	X	2×10^{-23}	35	RPS4X	X	2×10^{-04}	55	NA	NA	4×10^{-02}
16	UTY	Y	1×10^{-22}	36	ZRSR2	X	4×10^{-04}	56	RTCD1	1	5×10^{-02}
17	ZFY	Y	5×10^{-21}	37	USP19	3	1×10^{-03}	57	THY1	11	5×10^{-02}
18	UTY	Y	9×10^{-18}	38	USF2	19	1×10^{-03}	58	HTATIP2	11	5×10^{-02}
19	DDX3X	X	1×10^{-14}	39	USF2	19	2×10^{-03}	59	DYRK1B	19	5×10^{-02}
20	ZFX	X	2×10^{-13}	40	EIF1AX	X	2×10^{-03}	60	TUSC2	3	5×10^{-02}

RUV-2 (SVD), $k = 30$, housekeeping genes

Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P	Rank	Gene	Chrom	Adj. P
1	RPS4Y1	Y	2×10^{-141}	21	TMSB4Y	Y	5×10^{-19}	41	ZRSR2	X	5×10^{-06}
2	DDX3Y	Y	8×10^{-126}	22	ZFX	X	3×10^{-17}	42	EIF1AX	X	7×10^{-06}
3	XIST	X	8×10^{-105}	23	KDM6A	X	3×10^{-16}	43	USP9X	X	3×10^{-05}
4	XIST	X	1×10^{-101}	24	DDX3X	X	2×10^{-15}	44	EIF1AX	X	5×10^{-05}
5	EIF1AY	Y	7×10^{-99}	25	KDM6A	X	2×10^{-14}	45	TMEM147	19	3×10^{-04}
6	EIF1AY	Y	4×10^{-93}	26	RPS4X	X	2×10^{-13}	46	EIF2S3	X	3×10^{-04}
7	KDM5D	Y	5×10^{-91}	27	KDM5C	X	5×10^{-12}	47	DDX3X	X	3×10^{-04}
8	DDX3Y	Y	1×10^{-84}	28	NA	NA	1×10^{-11}	48	ATP10A	15	3×10^{-04}
9	TTTY15	Y	9×10^{-81}	29	RPS4X	X	5×10^{-11}	49	STIM1	11	5×10^{-04}
10	UTY	Y	6×10^{-79}	30	STS	X	7×10^{-11}	50	USP9X	X	7×10^{-04}
11	NLGN4Y	Y	4×10^{-71}	31	CD99	X Y	1×10^{-10}	51	TMEM204	16	8×10^{-04}
12	NCRNA00185	Y	6×10^{-67}	32	CD99	X Y	4×10^{-10}	52	NA	NA	1×10^{-03}
13	USP9Y	Y	6×10^{-49}	33	PRKY	Y	1×10^{-09}	53	ZFX	X	2×10^{-03}
14	CYorf15B	Y	3×10^{-46}	34	RPS4X	X	1×10^{-09}	54	GPR88	1	2×10^{-03}
15	UTY	Y	2×10^{-33}	35	EIF1AX	X	5×10^{-09}	55	DOCK9	13	3×10^{-03}
16	KDM6A	X	2×10^{-28}	36	ZRSR2	X	3×10^{-07}	56	UBA1	X	3×10^{-03}
17	HDHD1A	X	6×10^{-26}	37	DDX3X	X	4×10^{-07}	57	CA5BP	X	5×10^{-03}
18	DDX3X	X	2×10^{-25}	38	TRIM31	6	7×10^{-07}	58	STS	X	6×10^{-03}
19	UTY	Y	4×10^{-23}	39	SRY	Y	8×10^{-07}	59	GEMIN8	X	6×10^{-03}
20	ZFY	Y	6×10^{-23}	40	STS	X	2×10^{-06}	60	PDE4DIP	1	7×10^{-03}

Table A.5: Comparison of gene rankings before and after adjustment (SVD, $k = 30$) using the TCGA HT HG-U133A data. The data has been preprocessed (BG + NM). P-values were calculated using Limma.

A.6 Additional NCI-60 Figures

The following figures are dendrograms of the NCI-60 datasets (HG-U95A and HG-U133A) before and after preprocessing, before and after adjustment using various values of k, and using different sets of negative controls (housekeeping genes and spike-in controls). There are a lot of dendrograms, and it would take a long time to examine all of them. We therefore verbally summarize the dendrograms below. For most readers, the summary should suffice; the dendrograms themselves are included only for the especially curious.

Figures A.38, A.39 and A.40: HG-U95A dataset, preprocessed, adjusted by spike-in controls, $k = 0$ to 8.

The quality improves going from unadjusted to $k = 1$. Increasing k does not lead to further improvement, and the quality soon decreases.

Figure A.41: HG-U95A dataset, preprocessed, adjusted by housekeeping genes, $k = 0$ to 2.

Quality decreases going from unadjusted to $k = 1$, and decreases further when k is increased to 2.

Figure A.42: HG-U95A dataset, not preprocessed, adjusted by spike-in controls, $k = 0$ to 2.

Quality increases somewhat going from unadjusted to $k = 1$, then decreases somewhat going from $k = 1$ to $k = 2$.

Figure A.43: HG-U95A dataset, not preprocessed, adjusted by housekeeping genes, $k = 0$ to 2.

Quality increases going from unadjusted to $k = 1$, and does not change much going from $k = 1$ to $k = 2$.

Figure A.44: HG-U133A dataset, preprocessed, adjusted by spike-in controls, $k = 0$ to 2.

Quality stays about the same going from unadjusted to $k = 1$, and decreases somewhat going from $k = 1$ to $k = 2$.

Figure A.45: HG-U133A dataset, preprocessed, adjusted by housekeeping genes, $k = 0$ to 2.

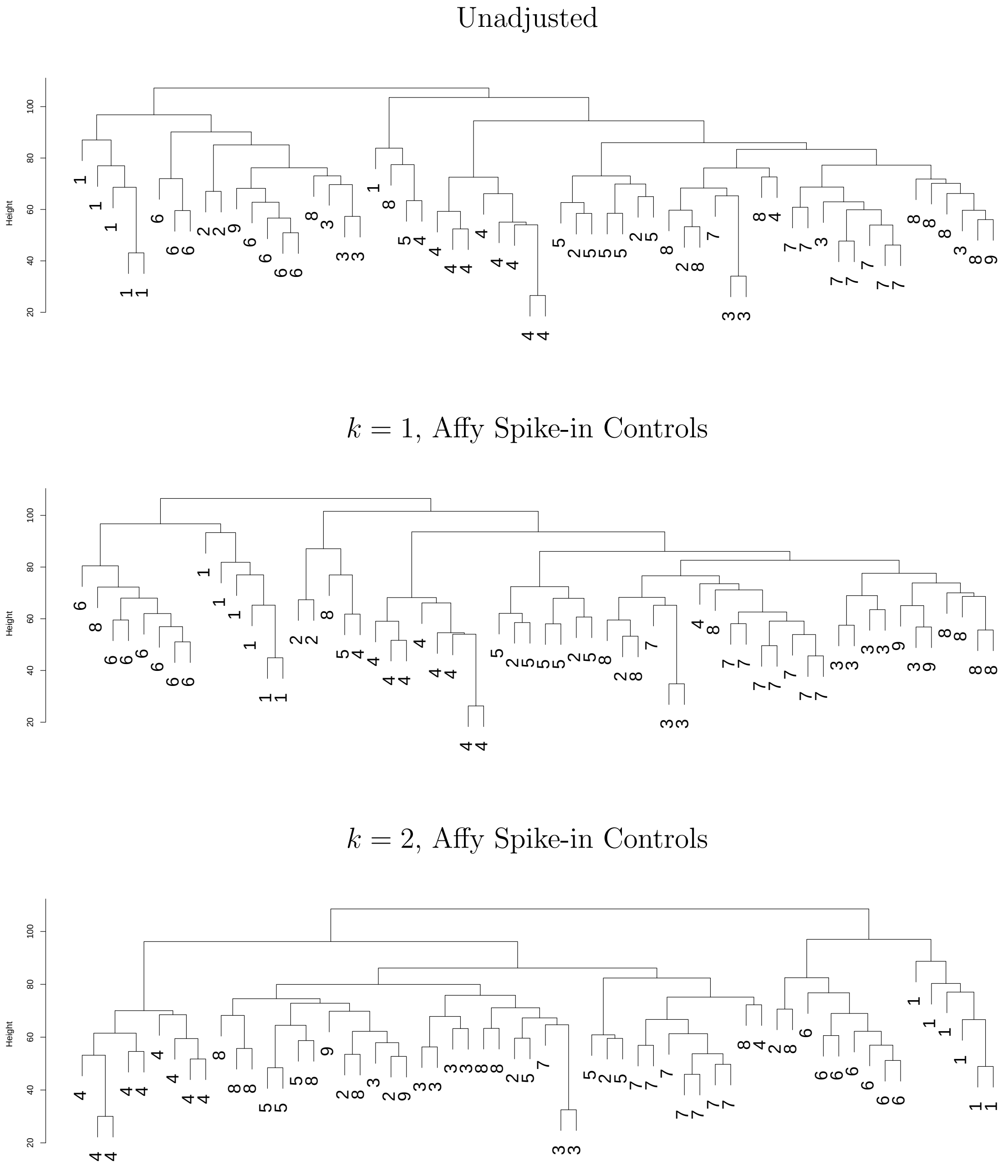
Quality decreases going from unadjusted to $k = 1$, and decreases further going from $k = 1$ to $k = 2$.

Figure A.46: HG-U133A dataset, not preprocessed, adjusted by spike-in controls, $k = 0$ to 2.

Quality increases going from unadjusted to $k = 1$, and increases slightly more going from $k = 1$ to $k = 2$.

Figure A.47: HG-U133A dataset, not preprocessed, adjusted by housekeeping genes, $k = 0$ to 2.

Quality increases going from unadjusted to $k = 1$, and stays about the same going from $k = 1$ to $k = 2$.

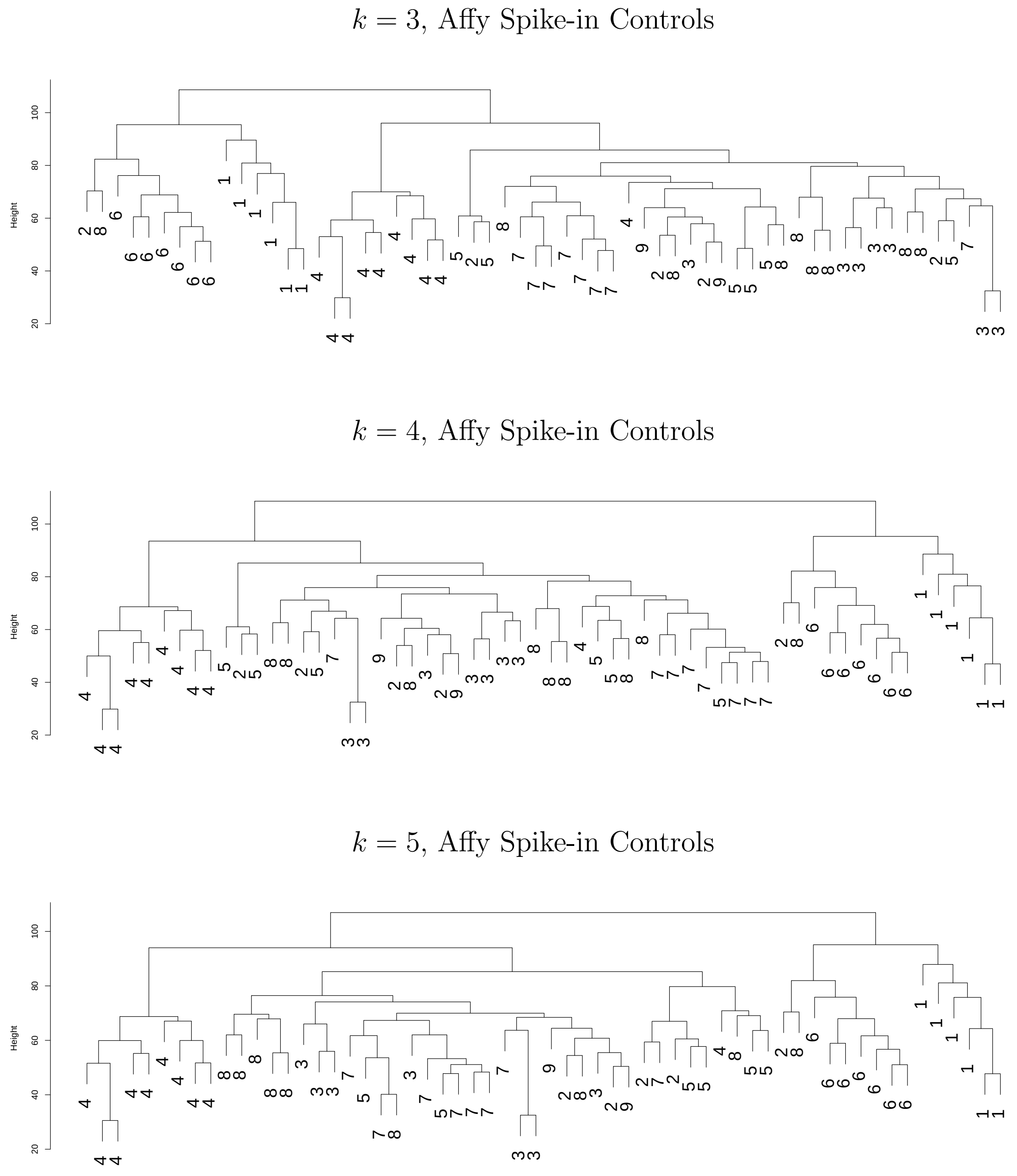


Unadjusted

$k = 1$, Affy Spike-in Controls

$k = 2$, Affy Spike-in Controls

Figure A.38: Dendrograms of NCI-60 HG-U95A dataset before and after adjustment. The data was preprocessed (BG + QN). The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.

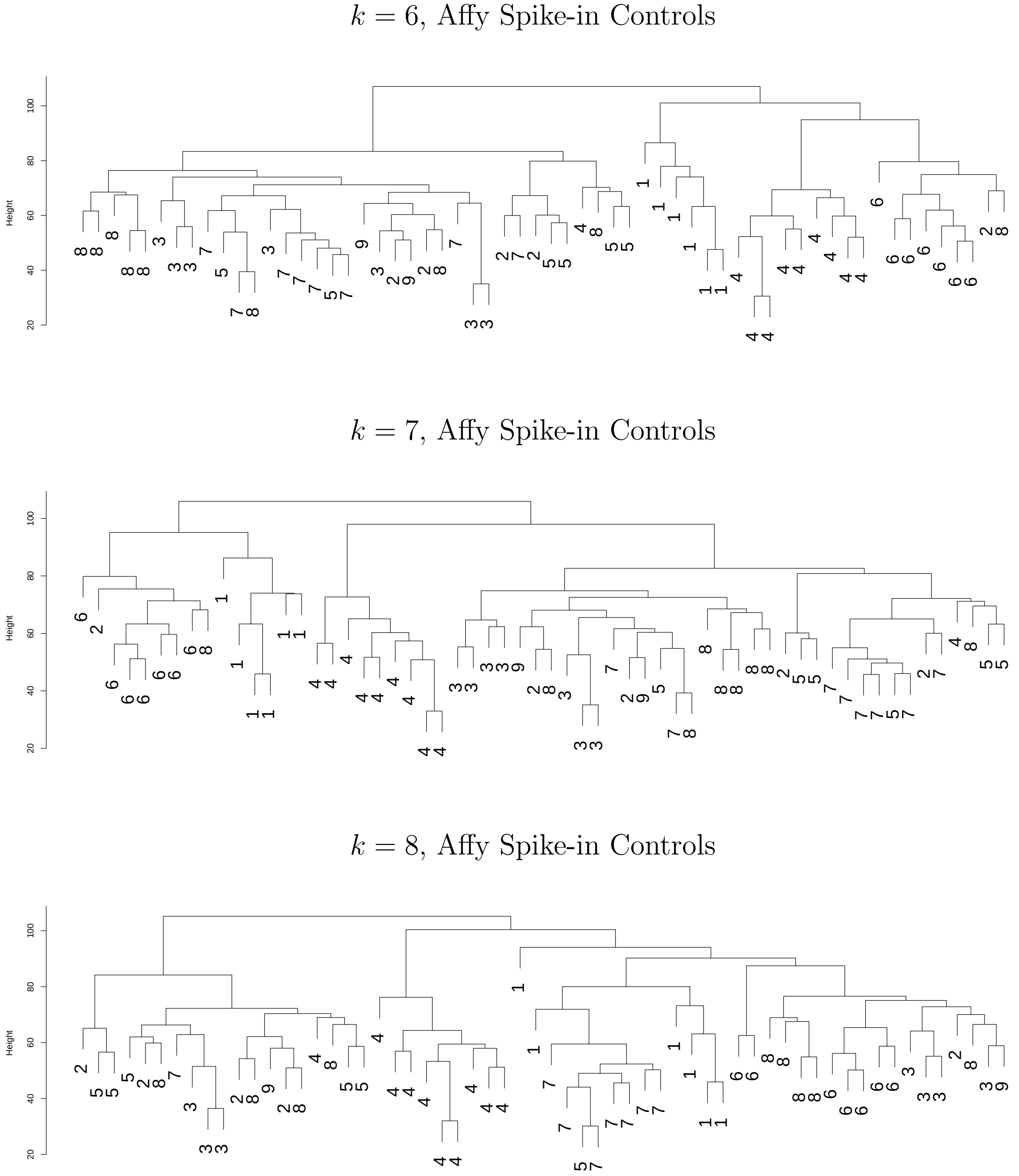


$k = 3$, Affy Spike-in Controls

$k = 4$, Affy Spike-in Controls

$k = 5$, Affy Spike-in Controls

Figure A.39: This is a continuation of Figure A.38

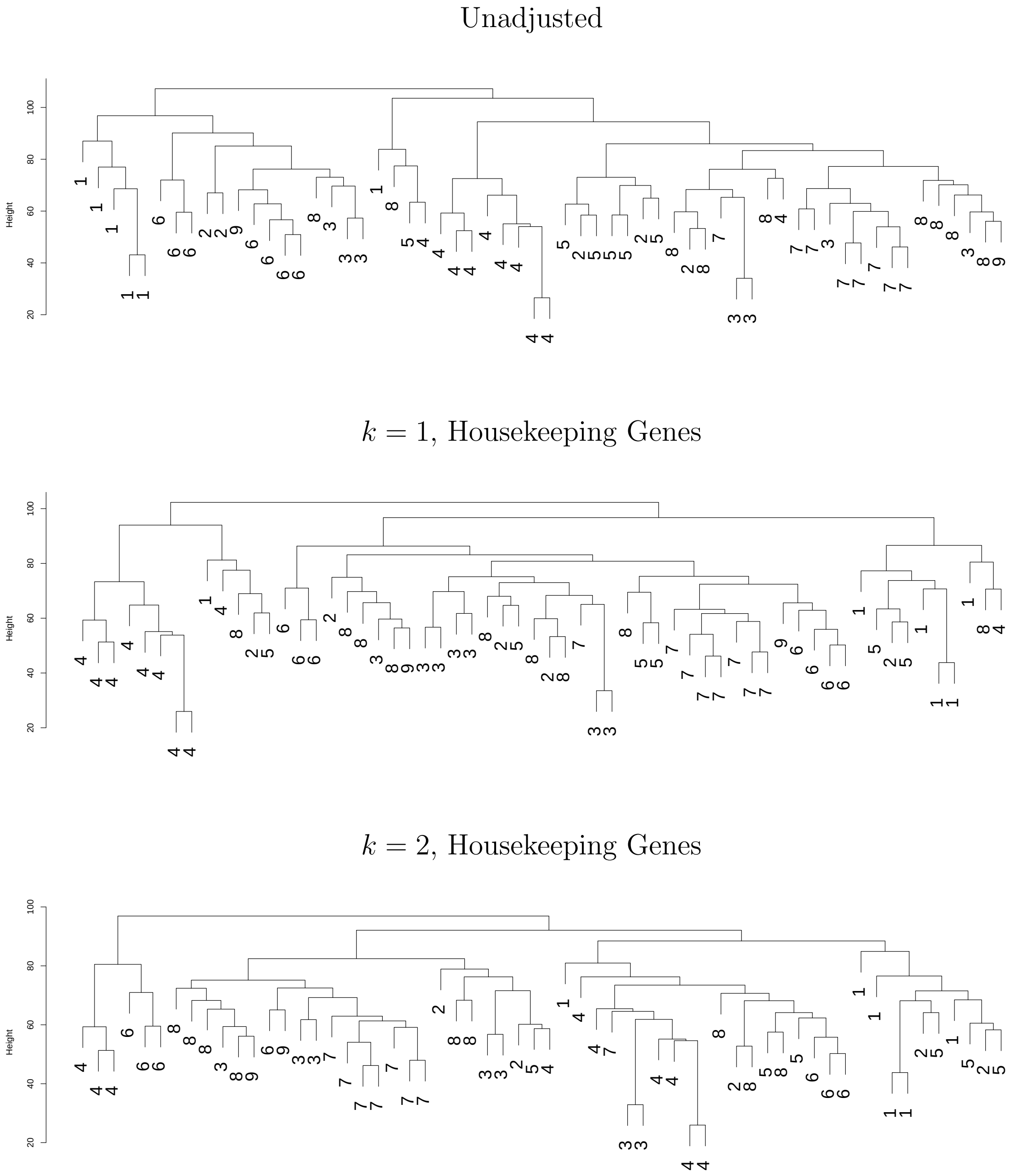


$k = 6$, Affy Spike-in Controls

$k = 7$, Affy Spike-in Controls

$k = 8$, Affy Spike-in Controls

Figure A.40: This is a continuation of Figure A.38



Unadjusted

$k = 1$, Housekeeping Genes

$k = 2$, Housekeeping Genes

Figure A.41: Dendrograms of NCI-60 HG-U95A dataset before and after adjustment. The data was preprocessed (BG + QN). The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.

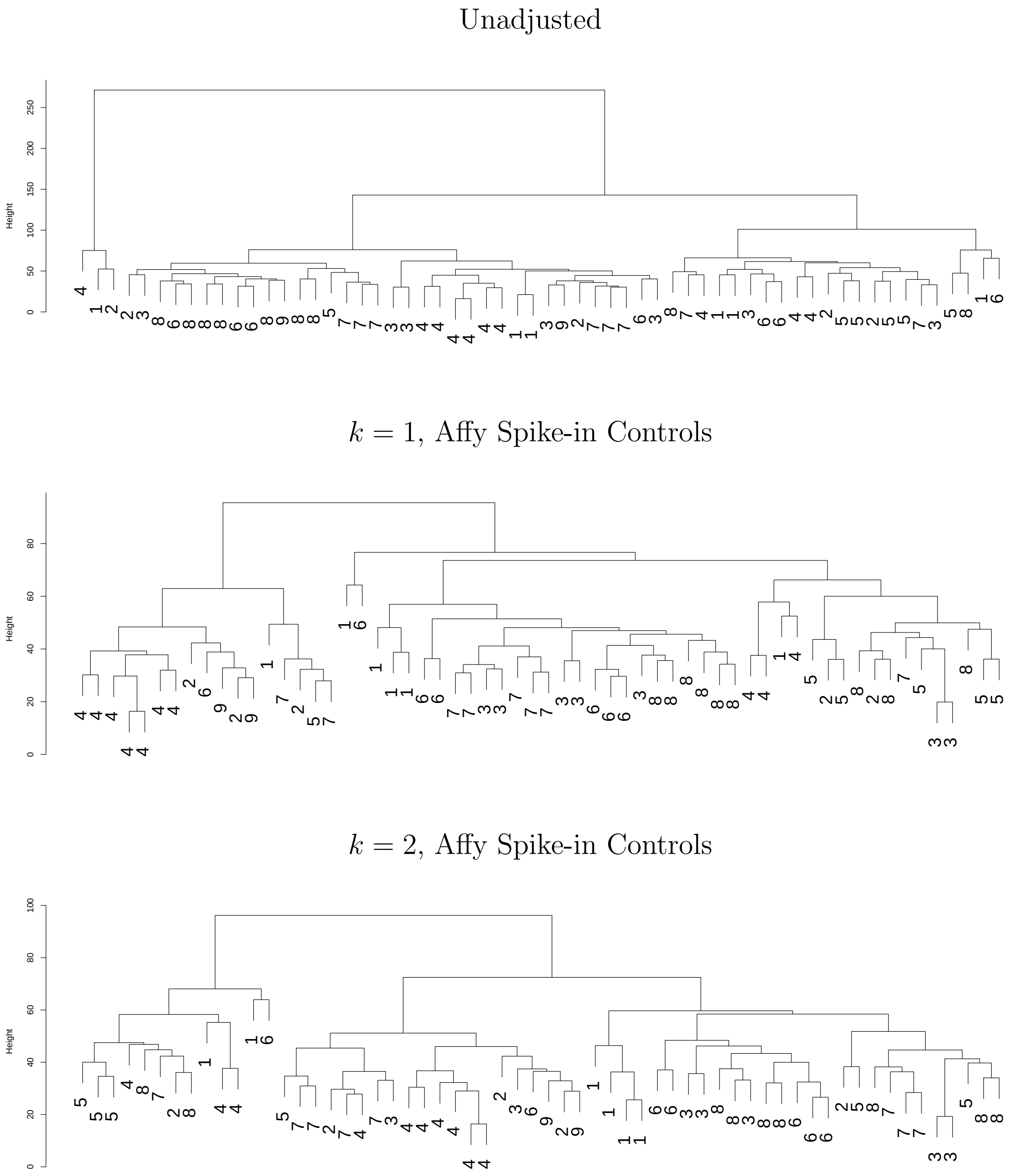
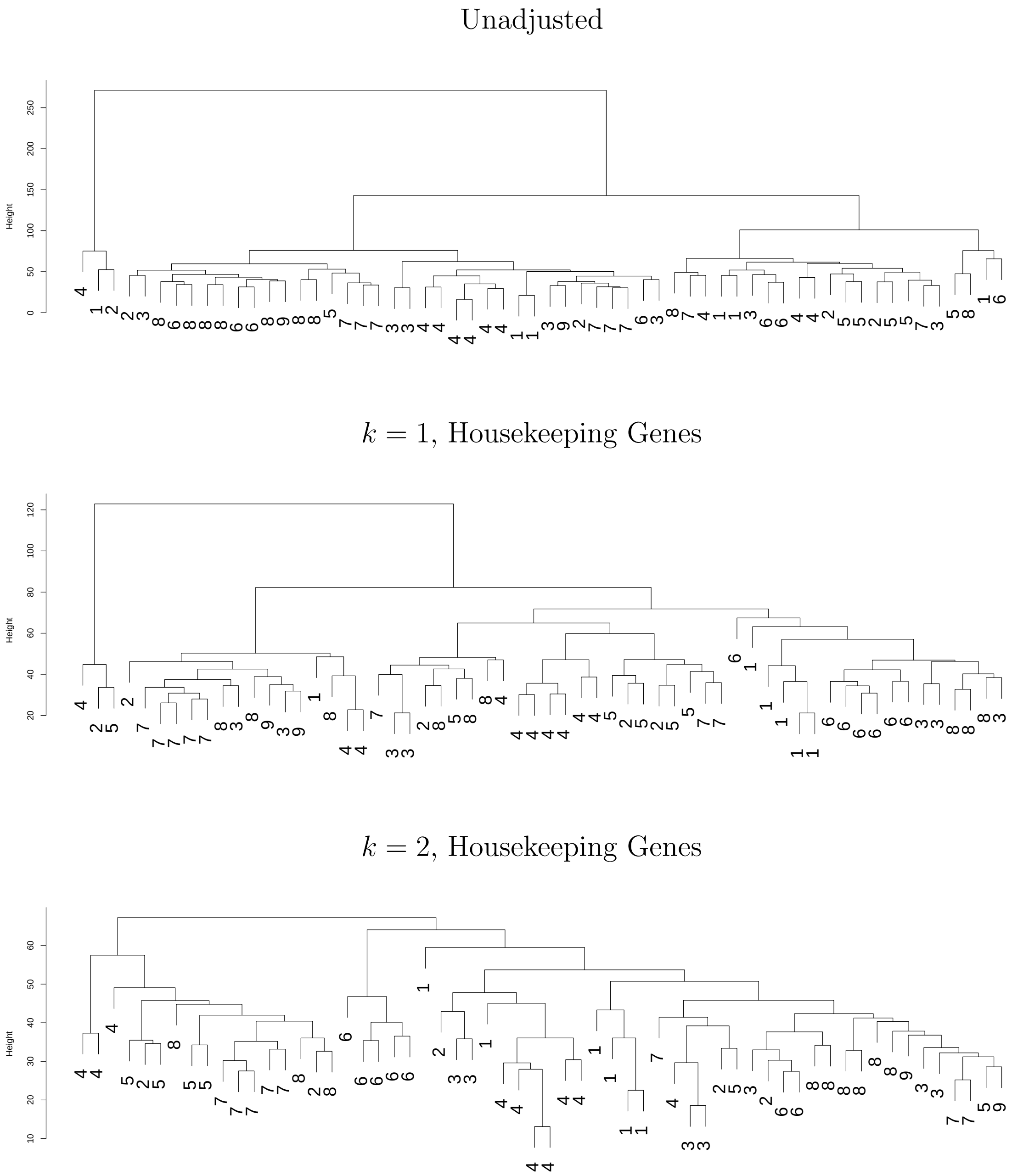


Figure A.42: Dendrograms of NCI-60 HG-U95A dataset before and after adjustment. The data was not preprocessed. The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.

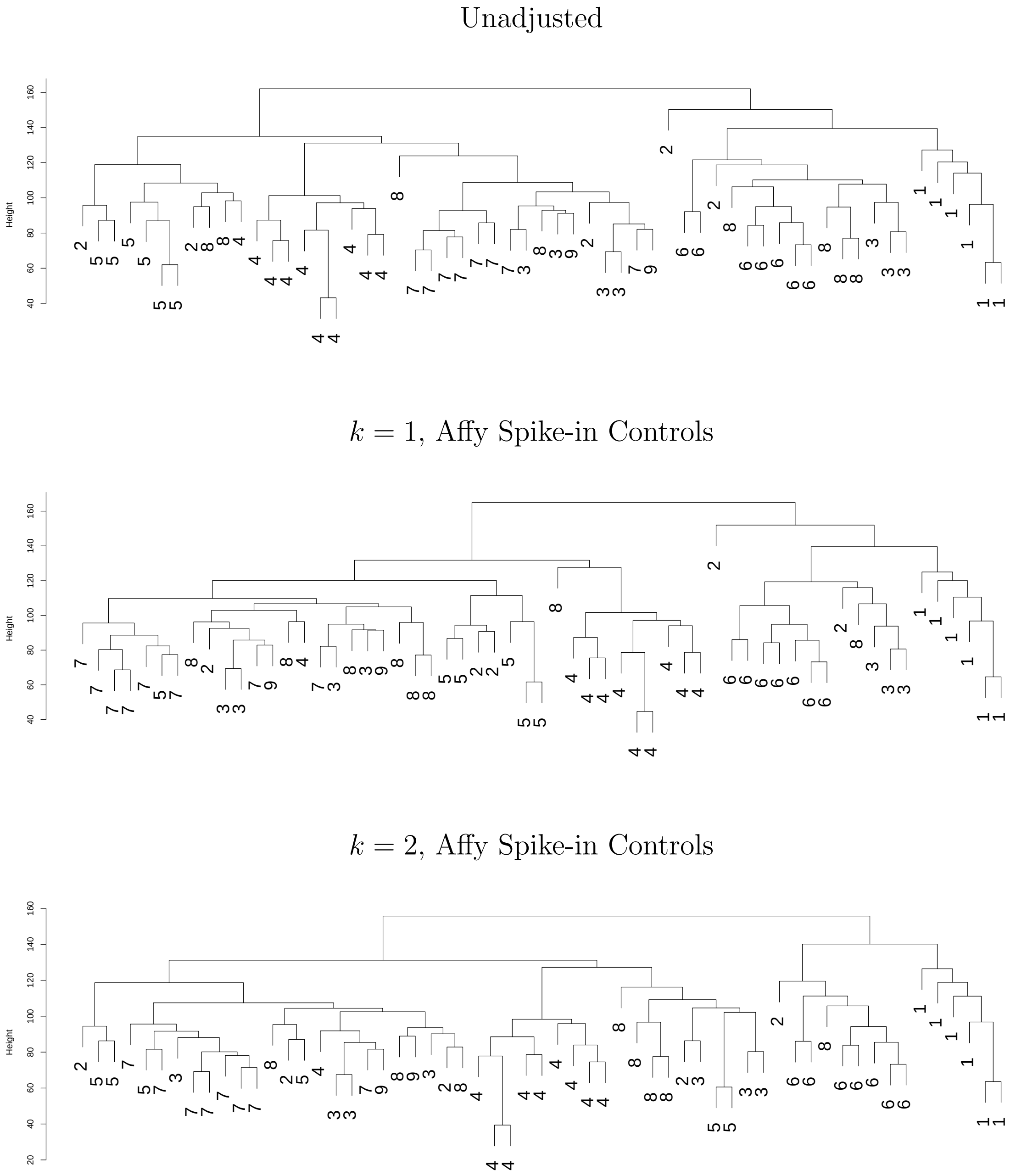


Unadjusted

$k = 1$, Housekeeping Genes

$k = 2$, Housekeeping Genes

Figure A.43: Dendrograms of NCI-60 HG-U95A dataset before and after adjustment. The data was not preprocessed. The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.

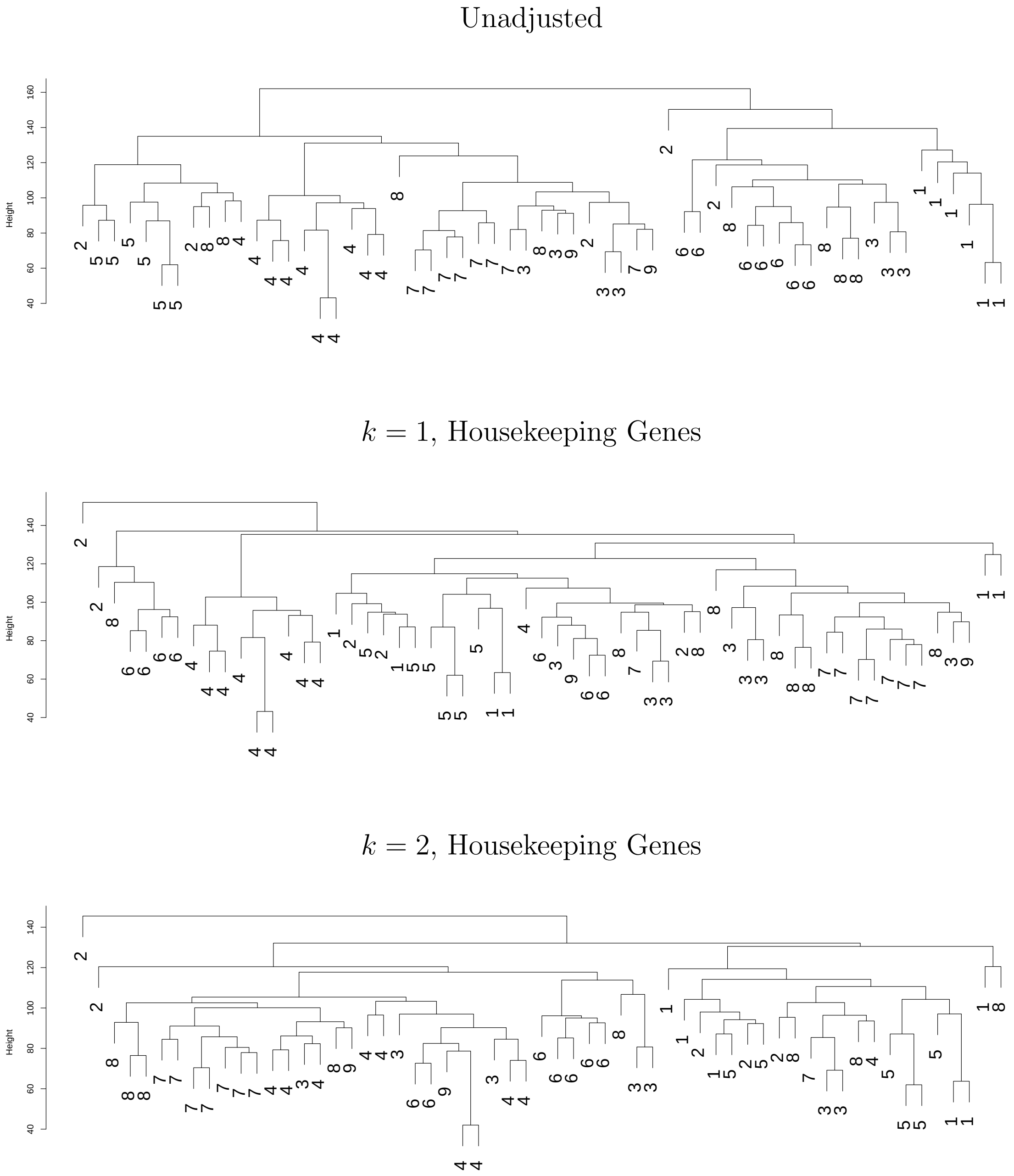


Unadjusted

$k = 1$, Affy Spike-in Controls

$k = 2$, Affy Spike-in Controls

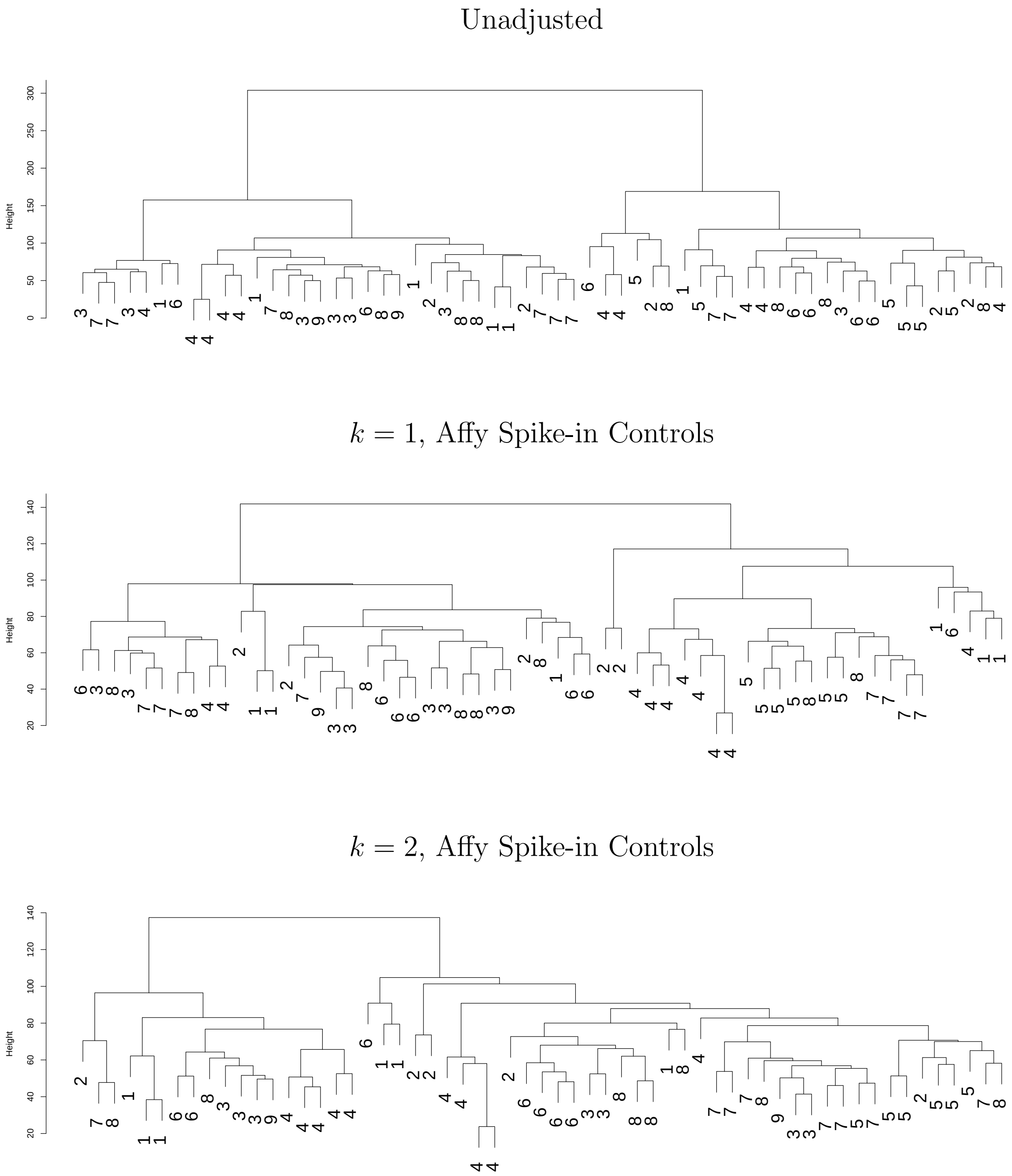
Figure A.44: Dendrograms of NCI-60 HG-U133A dataset before and after adjustment. The data was preprocessed (BG + QN). The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.


Unadjusted

$k = 1$, Housekeeping Genes

$k = 2$, Housekeeping Genes

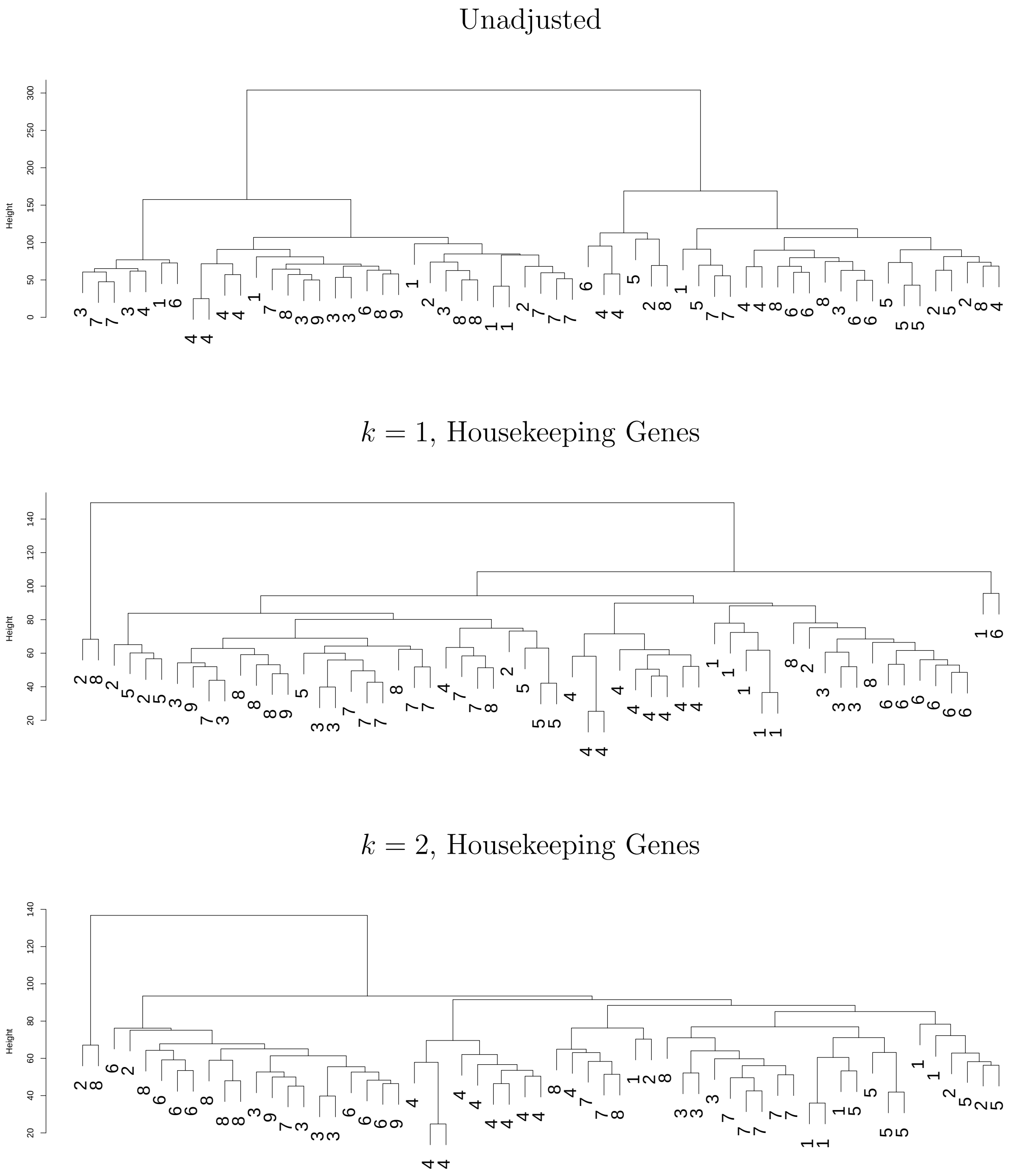
Figure A.45: Dendrograms of NCI-60 HG-U133A dataset before and after adjustment. The data was preprocessed (BG + QN). The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.


Unadjusted

$k = 1$, Affy Spike-in Controls

$k = 2$, Affy Spike-in Controls

Figure A.46: Dendrograms of NCI-60 HG-U133A dataset before and after adjustment. The data was not preprocessed. The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.



Unadjusted

$k = 1$, Housekeeping Genes

$k = 2$, Housekeeping Genes

Figure A.47: Dendrograms of NCI-60 HG-U133A dataset before and after adjustment. The data was not preprocessed. The factors were computed by SVD. Key: 1 – Blood; 2 – Breast; 3 – Ovarian; 4 – Melanoma; 5 – Brain; 6 – Colorectal; 7 – Renal; 8 – Lung; 9 – Prostate.

Appendix B

RUV-4 Supplementary Material

B.1 Miscellaneous Derivations and Discussions

B.1.1 The Parameterization of $\hat{W}_0$ Does Not Matter

Recall that the parameterization of $\hat{W}$ does not matter. If $\tilde{W}$ is a reparameterization of $\hat{W}$ in the sense that $\mathfrak{R}(\tilde{W}) = \mathfrak{R}(\hat{W})$, then the resulting $\hat{\beta}$ will be the same whether we use $\tilde{W}$ or $\hat{W}$. The point of this section is to show that the parameterization of $\hat{W}_0$ does not matter either. More formally, we wish to show that if $\mathfrak{R}(\tilde{W}_0) = \mathfrak{R}(\hat{W}_0)$ and if $\tilde{W}$ and $\hat{W}$ are the corresponding estimates of W computed using Step 3 of RUV-4, then $\mathfrak{R}(\tilde{W}) = \mathfrak{R}(\hat{W})$.

Let $\tilde{W}_0 = \hat{W}_0 Q$ where Q is any $k \times k$ invertible matrix. Then

$$\begin{aligned} \tilde{\alpha}_c &\equiv \left[(\hat{W}_0 Q)' \hat{W}_0 Q\right]^{-1} Q' \hat{W}_0' Y_c && \text{(B.1)} \\ &= Q^{-1} \left(\hat{W}_0' \hat{W}_0\right)^{-1} \hat{W}_0' Y_c && \text{(B.2)} \\ &= Q^{-1} \hat{\alpha}_c && \text{(B.3)} \end{aligned}$$

and

$$\begin{aligned} \tilde{b}_{WX} &\equiv b_{Y_c X} \tilde{\alpha}_c' \left(\tilde{\alpha}_c \tilde{\alpha}_c'\right)^{-1} && \text{(B.4)} \\ &= b_{Y_c X} \hat{\alpha}_c' (Q^{-1})' \left[Q^{-1} \hat{\alpha}_c \hat{\alpha}_c' (Q^{-1})'\right]^{-1} && \text{(B.5)} \\ &= b_{Y_c X} \hat{\alpha}_c' \left(\hat{\alpha}_c \hat{\alpha}_c'\right)^{-1} Q && \text{(B.6)} \\ &= \hat{b}_{WX} Q && \text{(B.7)} \end{aligned}$$

so

$$\begin{aligned} \tilde{W} &\equiv \tilde{W}_0 + X \tilde{b}_{WX} && \text{(B.8)} \\ &= \hat{W}_0 Q + X \hat{b}_{WX} Q && \text{(B.9)} \\ &= \hat{W} Q. && \text{(B.10)} \end{aligned}$$

B.1.2 Reformulation of the OLS Variance

Assume that X has unit length and that the columns of W_0 are orthogonal and have unit length. Let $\hat{\beta}^{(\mathrm{OLS})}$ be the OLS estimate of β (ignore for the moment that W is unknown). The variance of $\hat{\beta}_j^{(\mathrm{OLS})}$ is the $(1,1)$ entry of the matrix $\sigma_j^2\left[(X|W)'(X|W)\right]^{-1}$. The goal of this section is to show that this is equal to $\sigma_j^2\left(1+b_{WX}b_{WX}'\right)$.

Begin with the observation that

$$\left[(X|W)'(X|W)\right]^{-1} = \begin{pmatrix} X'X & X'W \\ W'X & W'W \end{pmatrix}^{-1}. \tag{B.11}$$

We can invert this matrix block-wise. The $(1,1)$ entry is equal to

$$\left[X'X - X'W\left(W'W\right)^{-1}W'X\right]^{-1}.$$

Now, $X'X = 1$, $X'W = b_{WX}$, $W'X = b_{WX}'$, and $W'W = I + b_{WX}'b_{WX}$, so

$$\begin{aligned}
\mathrm{Var}\left[\hat{\beta}_j^{(\mathrm{OLS})}\right] &= \sigma_j^2\left[1 - b_{WX}\left(I + b_{WX}'b_{WX}\right)^{-1}b_{WX}'\right]^{-1} && \text{(B.12)}\\
&= \sigma_j^2\left\{1 - b_{WX}\left[I - b_{WX}'\left(I + b_{WX}b_{WX}'\right)^{-1}b_{WX}\right]b_{WX}'\right\}^{-1} && \text{(B.13)}\\
&= \sigma_j^2\left[1 - b_{WX}b_{WX}' + b_{WX}b_{WX}'\left(1 + b_{WX}b_{WX}'\right)^{-1}b_{WX}b_{WX}'\right]^{-1} && \text{(B.14)}\\
&= \sigma_j^2\left[1 - x + x\left(1+x\right)^{-1}x\right]^{-1} && \text{(B.15)}\\
&= \sigma_j^2\left(\frac{1-x^2}{1+x} + \frac{x^2}{1+x}\right)^{-1} && \text{(B.16)}\\
&= \sigma_j^2(1+x) && \text{(B.17)}\\
&= \sigma_j^2(1+b_{WX}b_{WX}') && \text{(B.18)}
\end{aligned}$$

where $x \equiv b_{WX}b_{WX}'$.

B.1.3 Estimating Σ_j as $\frac{1}{n_c}\left(Y_c Y_c'\right) + \left(\hat{\sigma}_j^2 - \dot{\sigma}_c^2\right) I$

In Section 3.3.7.4 we made the observation that

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{n_c}\left(Y_c Y_c'\right)\right] &= \Sigma + \bar{\sigma}_c^2 I && \text{(B.19)}\\
&\neq \Sigma + \bar{\sigma}_j^2 I && \text{(B.20)}\\
&= \Sigma_j. && \text{(B.21)}
\end{aligned}$$

$\frac{1}{n_c}\left(Y_c Y_c'\right)$ is a biased estimator of Σ_j. In this section we will consider alternative estimators of Σ_j. Specifically, we will consider estimators of the form

$$\hat{\Sigma}_j = \frac{1}{n_c} Y_c Y_c' + \lambda I. \qquad \text{(B.22)}$$

For a given estimator $\hat{\Sigma}_j$ it can be shown that

$$\mathrm{Var}\left[\hat{\beta}_j \middle| \hat{\Sigma}_j\right] = \left(X'\hat{\Sigma}_j^{-1}X\right)^{-1} X'\hat{\Sigma}_j^{-1}\Sigma_j\hat{\Sigma}_j^{-1}X\left(X'\hat{\Sigma}_j^{-1}X\right)^{-1}. \qquad \text{(B.23)}$$

For our purposes, a good estimator of $\hat{\Sigma}_j$ is one such that

$$\mathbb{E}\left\{\mathrm{Var}\left[\hat{\beta}_j \middle| \hat{\Sigma}_j\right]\right\} \qquad \text{(B.24)}$$

is small.

Let

$$\begin{aligned}
\hat{\Sigma}_j(\lambda) &\equiv \frac{1}{n_c} Y_c Y_c' + \lambda I && \text{(B.25)}\\
\hat{\beta}_j(\hat{\Sigma}_j) &\equiv \left(X'\hat{\Sigma}_j^{-1}X\right)^{-1} X'\hat{\Sigma}_j^{-1}Y_{\star j} && \text{(B.26)}\\
\lambda^* &\equiv \underset{\lambda}{\operatorname{argmin}}\,\mathrm{Var}\left[\hat{\beta}_j\left(\hat{\Sigma}_j(\lambda)\right) \middle| \hat{\Sigma}_j(\lambda)\right]. && \text{(B.27)}
\end{aligned}$$

We will consider 5 “estimators.”

$$\begin{aligned}
\hat{\Sigma}_j^{(1)} &= \Sigma + \sigma_j^2 I && \text{(B.28)}\\
\hat{\Sigma}_j^{(2)} &= \frac{1}{n_c} Y_c Y_c' + \lambda^* I && \text{(B.29)}\\
\hat{\Sigma}_j^{(3)} &= \frac{1}{n_c} Y_c Y_c' + \left(\sigma_j^2 - \bar{\sigma}_c^2\right) I && \text{(B.30)}\\
\hat{\Sigma}_j^{(4)} &= \frac{1}{n_c} Y_c Y_c' + 0.2\left(\sigma_j^2 - \bar{\sigma}_c^2\right) I && \text{(B.31)}\\
\hat{\Sigma}_j^{(5)} &= \frac{1}{n_c} Y_c Y_c' && \text{(B.32)}\\
& && \text{(B.33)}
\end{aligned}$$

Note that only $\hat{\Sigma}_j^{(5)}$ is a real estimator that can be computed from data. $\hat{\Sigma}_j^{(1)} = \Sigma_j$ is the true parameter. $\hat{\Sigma}_j^{(2)}$ is the optimal estimator of the form $\frac{1}{n_c} Y_c Y_c' + \lambda I$. To compute it requires knowledge of Σ_j. $\hat{\Sigma}_j^{(3)}$ is an idealization of the estimator briefly mentioned in Section 3.3.7.4. Here we have substituted the parameter $\left(\sigma_j^2 - \bar{\sigma}_c^2\right)$ for the estimate $\left(\hat{\sigma}_j^2 - \dot{\sigma}_c^2\right)$. $\hat{\Sigma}_j^{(4)}$ is a modified version of $\hat{\Sigma}_j^{(3)}$. The choice of 0.2 is arbitrary.

We want to compare the performance of the $\hat{\Sigma}_j^{(i)}$. This is difficult to accomplish analytically. Instead we will use simulations. Define

$$\hat{\beta}_j^{(i)} \equiv \hat{\beta}_j \left(\hat{\Sigma}_j^{(i)}\right). \tag{B.34}$$

Let

$$v_i(\sigma_j^2) \equiv \mathbb{E}\left\{\operatorname{Var}\left[\hat{\beta}_j \left(\hat{\Sigma}_j^{(i)}\right) \middle| \hat{\Sigma}_j^{(i)}\right]\right\} \tag{B.35}$$

denote the expected variance when σ_j^2 is the true parameter. Note that v_i should be indexed by j and should also be a function of Σ, σ_c^2 and X but we suppress this in the notation. We consider three quantities of interest: $\sqrt{v_i(\sigma_j^2)}$, $\sqrt{v_i(\sigma_j^2)/v_1(\sigma_j^2)}$, and $\sqrt{v_i(\sigma_j^2)/v_2(\sigma_j^2)}$. The first is the RMSE of $\hat{\beta}_j \left(\hat{\Sigma}_j^{(i)}\right)$. The second is the RMSE of $\hat{\beta}_j \left(\hat{\Sigma}_j^{(i)}\right)$ as a fraction of the RMSE of the ideal estimator $\hat{\beta}_j(\Sigma)$. The third is the RMSE of $\hat{\beta}_j \left(\hat{\Sigma}_j^{(i)}\right)$ as a fraction of the RMSE of the "best possible" estimator $\hat{\Sigma}_j^{(2)}$. Of course, in practice, $\hat{\Sigma}_j^{(2)}$ is not actually a possible estimator. Simulation results are given in Figure B.1.

A quick glance at Figure B.1 reveals that the performance does vary from one estimator to the next, but not always by very much. The one exception is $\hat{\beta}_j^{(3)}$, which performs horribly when $\sigma_j^2/\bar{\sigma}_c^2 < 1$. Therefore, even if unbiased estimates $\hat{\sigma}_j^2$ of σ_j^2 and $\dot{\sigma}_c^2$ of $\bar{\sigma}_c^2$ are available, and $\frac{1}{n_c}\left(Y_c Y_c'\right) + \left(\hat{\sigma}_j^2 - \dot{\sigma}_c^2\right) I$ is therefore an unbiased estimator of Σ_j, this estimator should not be used! This result is perhaps to be anticipated. The smallest eigenvalue of $\frac{1}{n_c}\left(Y_c Y_c'\right)$ will often be considerably smaller than σ_j^2. As a result $\frac{1}{n_c}\left(Y_c Y_c'\right) + \left(\sigma_j^2 - \bar{\sigma}_c^2\right) I$ may have tiny — or even negative — eigenvalues.

$\hat{\beta}_j^{(4)}$ improves on $\hat{\beta}_j^{(3)}$ by "shrinking" the ridge-adjustment term $\left(\sigma_j^2 - \bar{\sigma}_c^2\right) I$. Indeed, $\hat{\beta}_j^{(4)}$ performs quite well. However, the shrinkage factor of 0.2 is arbitrary; we chose 0.2 simply because it gave good results. Implementing this "shrunken ridge adjustment" strategy in practice would require a method for choosing a good shrinkage factor. This would presumably depend on Σ, n_c, the distribution of σ^2, etc. In practice we would also need to account for the fact that σ_j^2 and $\bar{\sigma}_c^2$ are not known, but estimated.

Figure B.1 suggests an alternative strategy. Set

$$\hat{\beta}_j^{(6)} \equiv \begin{cases} \hat{\beta}_j^{(5)} & \text{if } \sigma_j^2 < \bar{\sigma}_c^2 \\ \hat{\beta}_j^{(3)} & \text{if } \sigma_j^2 \geq \bar{\sigma}_c^2 \end{cases}. \tag{B.36}$$

In the simulations of Figure B.1, $\hat{\beta}_j^{(6)}$ performs quite well. We have not investigated whether this strategy works well more generally. In practice, of course, we would need to replace σ_j^2 with $\hat{\sigma}_j^2$ and $\bar{\sigma}_c^2$ with $\dot{\sigma}_c^2$. If these estimates are very noisy, $\hat{\beta}_j^{(6)}$ may no longer perform well.

Finally, we note that in the big picture, the performance of $\hat{\beta}_j^{(5)}$ is adequate. In the lightly and moderately correlated examples, the RMSE of $\hat{\beta}_j^{(5)}$ is only larger than the "best possible" RMSE by about 5%. In the figures and tables of Section B.2 we see that a 5-10% increase in the RMSE of $\hat{\beta}$ is relatively minor compared to the choice of K, the choice of method, the choice of controls, etc. Perhaps more importantly, in Section 3.4.2 of the main text we argue that the primary determinant of the discriminative power is the performance of $\hat{\sigma}^2$, not the performance of $\hat{\beta}$. It is for these reasons that we feel that $\hat{\beta}_j^{(5)}$ is generally adequate.

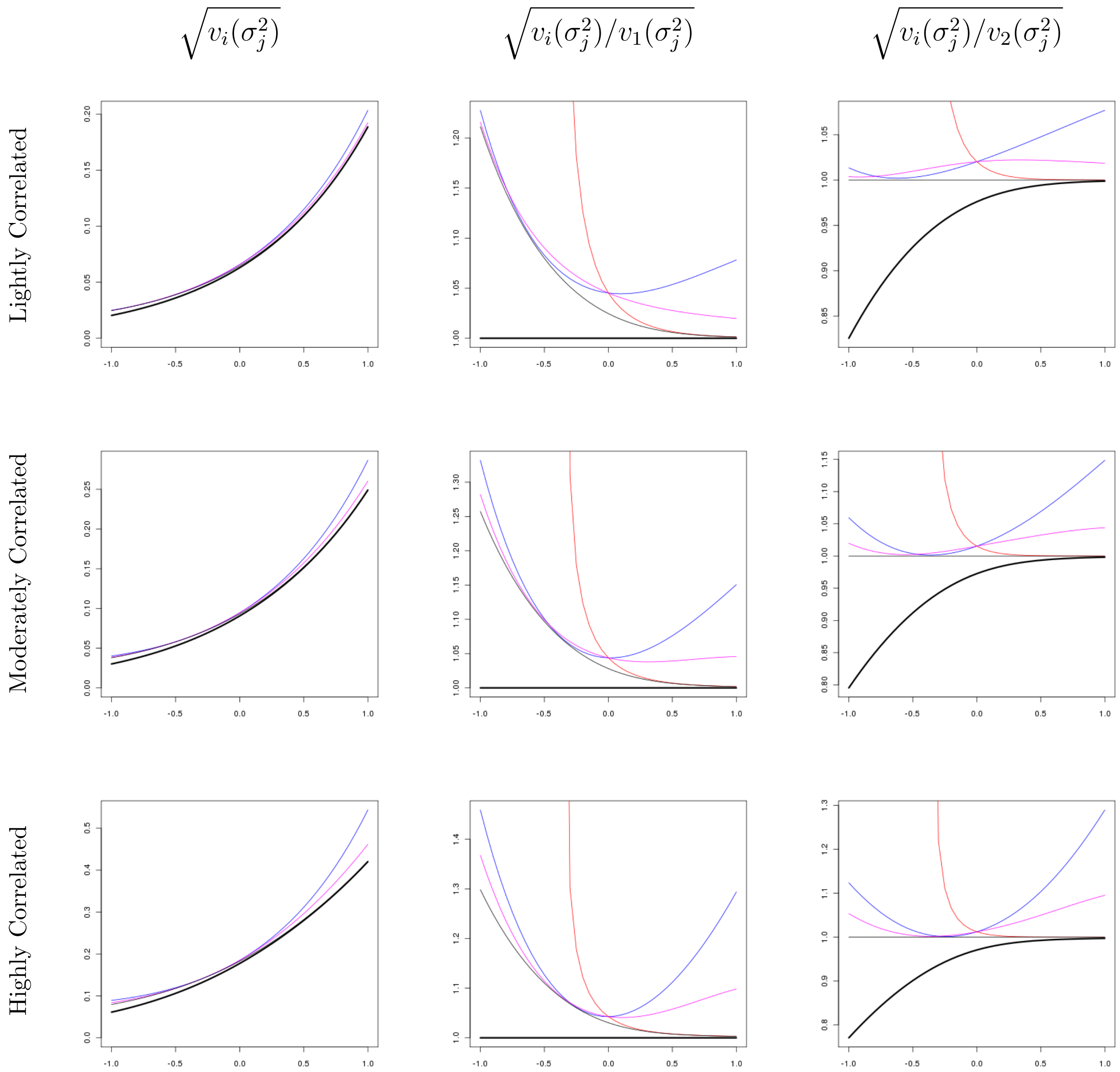


Figure B.1: Plots of $\sqrt{v_i(\sigma_j^2)}$, $\sqrt{v_i(\sigma_j^2)/v_1(\sigma_j^2)}$, and $\sqrt{v_i(\sigma_j^2)/v_2(\sigma_j^2)}$ as σ_j^2 is varied from $0.1\bar{\sigma}_c^2$ to $10\bar{\sigma}_c^2$. The vertical axis is the quantity of interest, e.g. $\sqrt{v_i(\sigma_j^2)}$, and the horizontal axis is $\log_{10}(\sigma_j^2/\bar{\sigma}_c^2)$. The coloring scheme is as follows: $i = 1$, thick black line; $i = 2$, thin black line; $i = 3$, red; $i = 4$, violet; $i = 5$, blue. The simulation parameters are the same as those of the simulations presented in Section 3.4. $n_c = 1000$. We show the results for three separate simulations ("Lightly Correlated," "Moderately Correlated," "Highly Correlated."). $i = 3$ is omitted from the left column because it behaves too erratically.

B.2 Additional Simulation Results

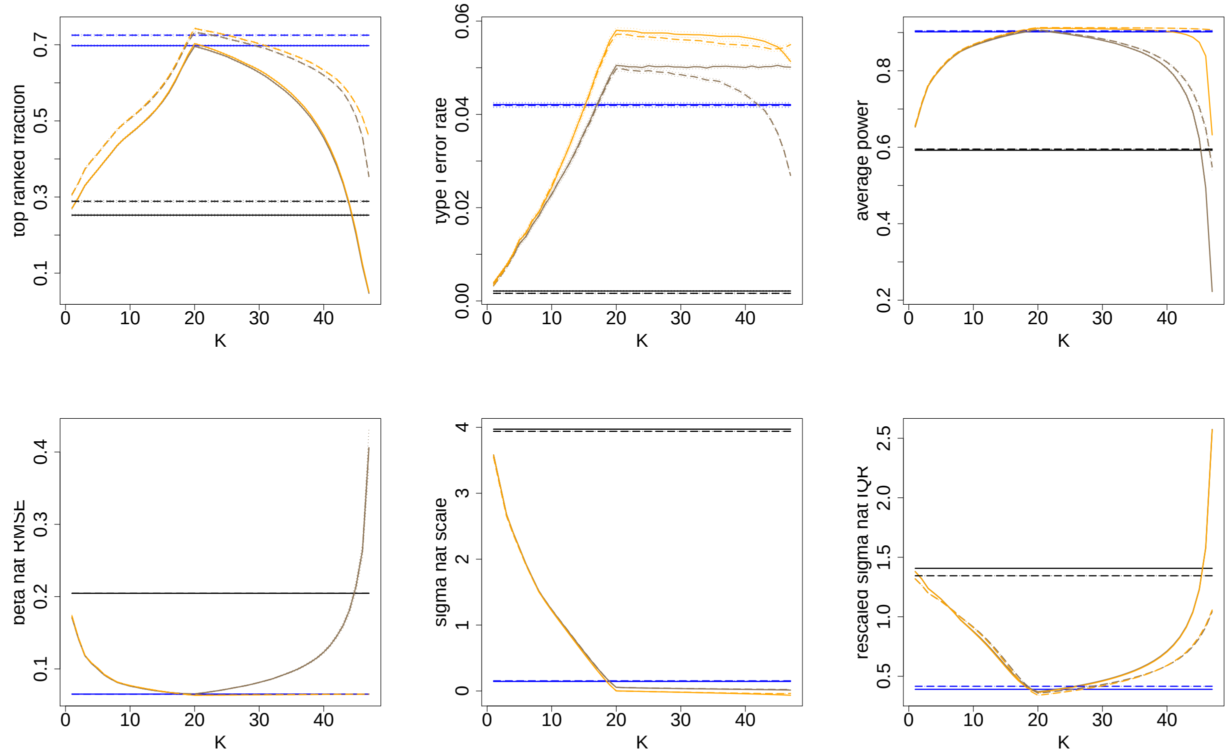


Figure B.2: $k = 20$, lightly correlated, good controls.

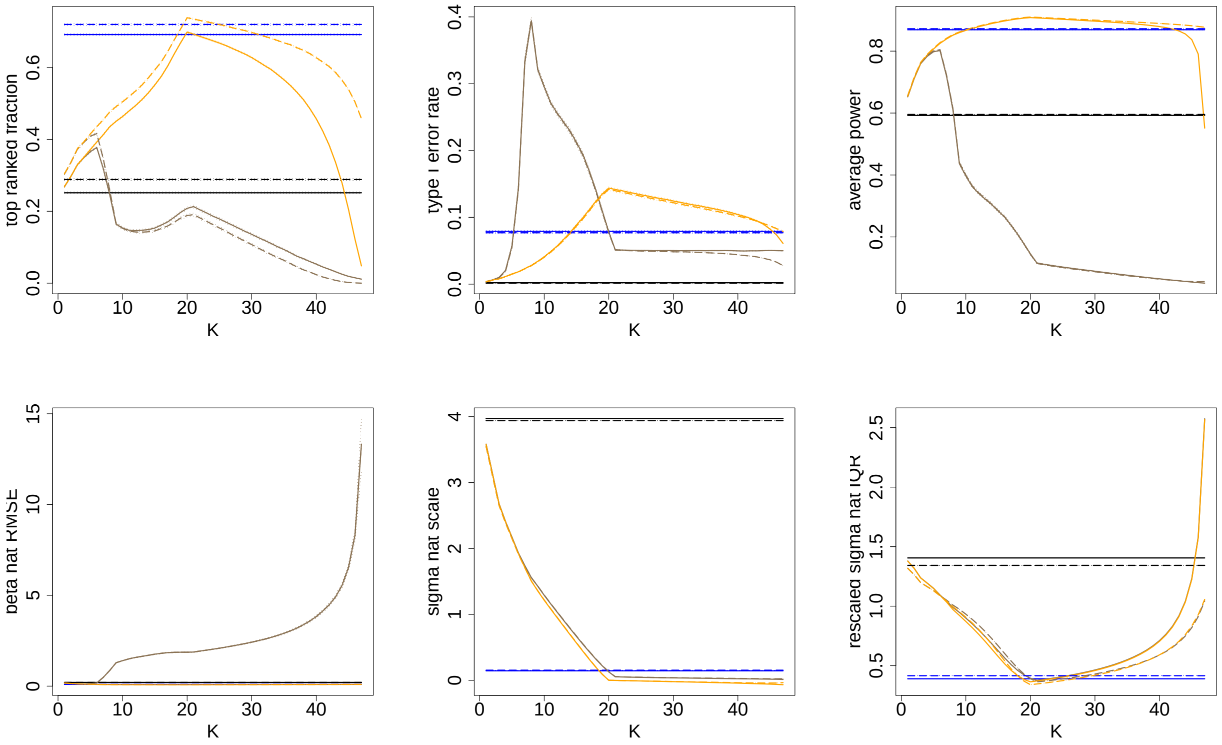


Figure B.3: $k = 20$, lightly correlated, bad controls.

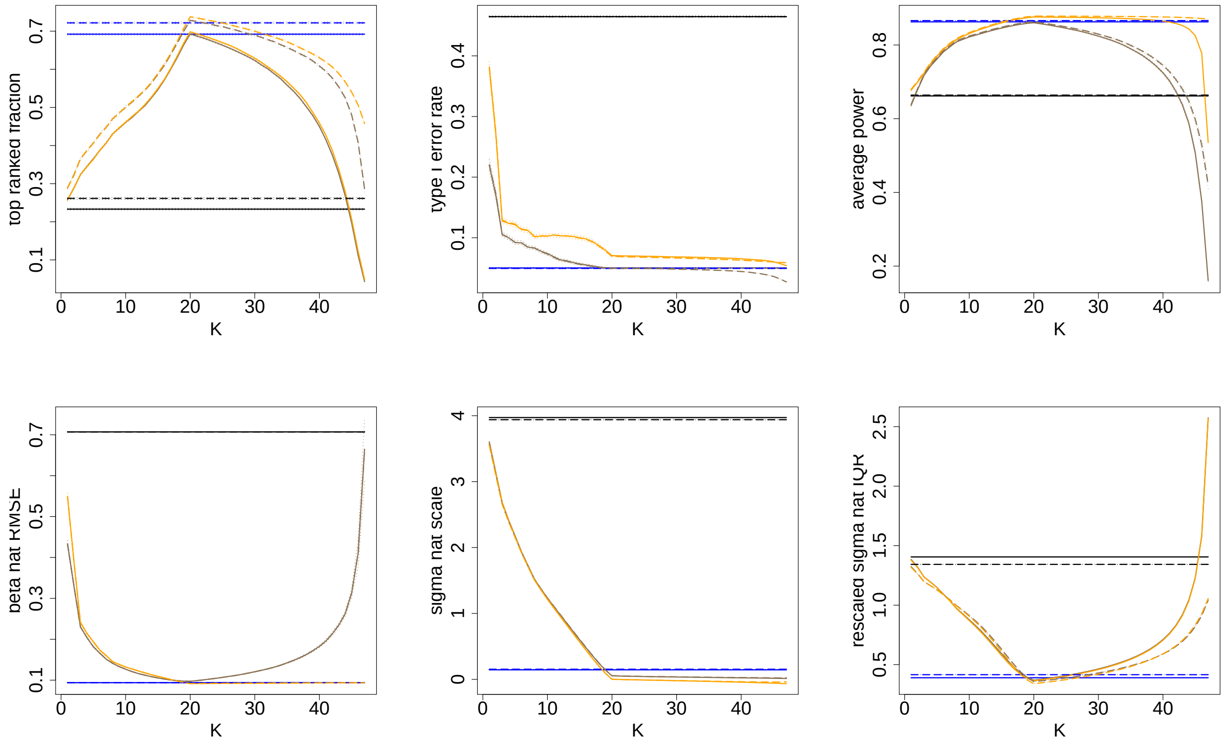


Figure B.4: $k = 20$, moderately correlated, good controls.

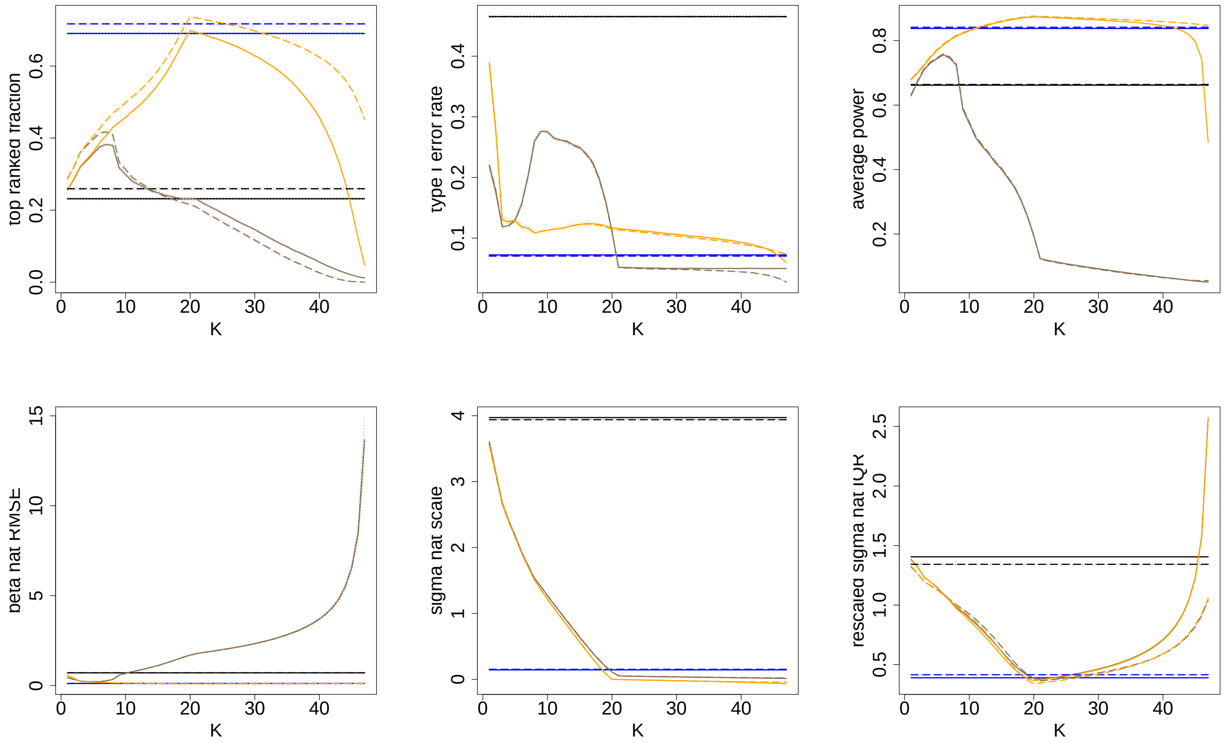


Figure B.5: $k = 20$, moderately correlated, bad controls.

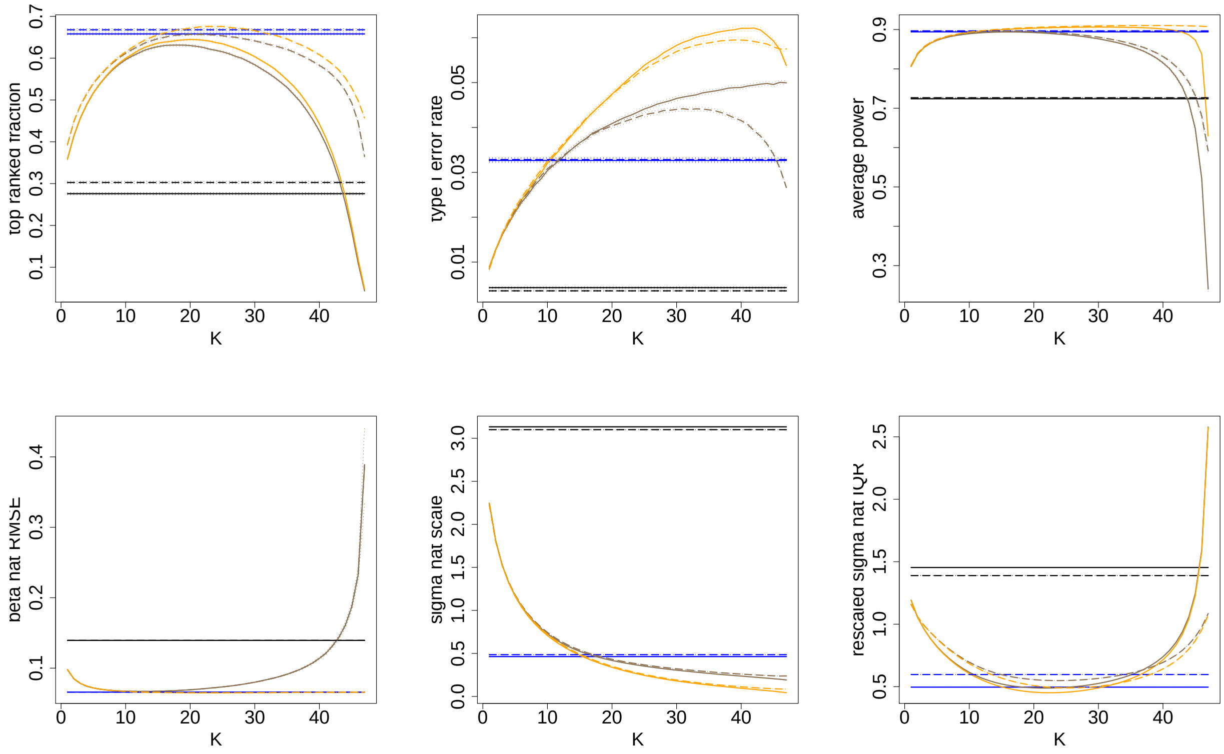


Figure B.6: Moderate decay, lightly correlated, good controls.

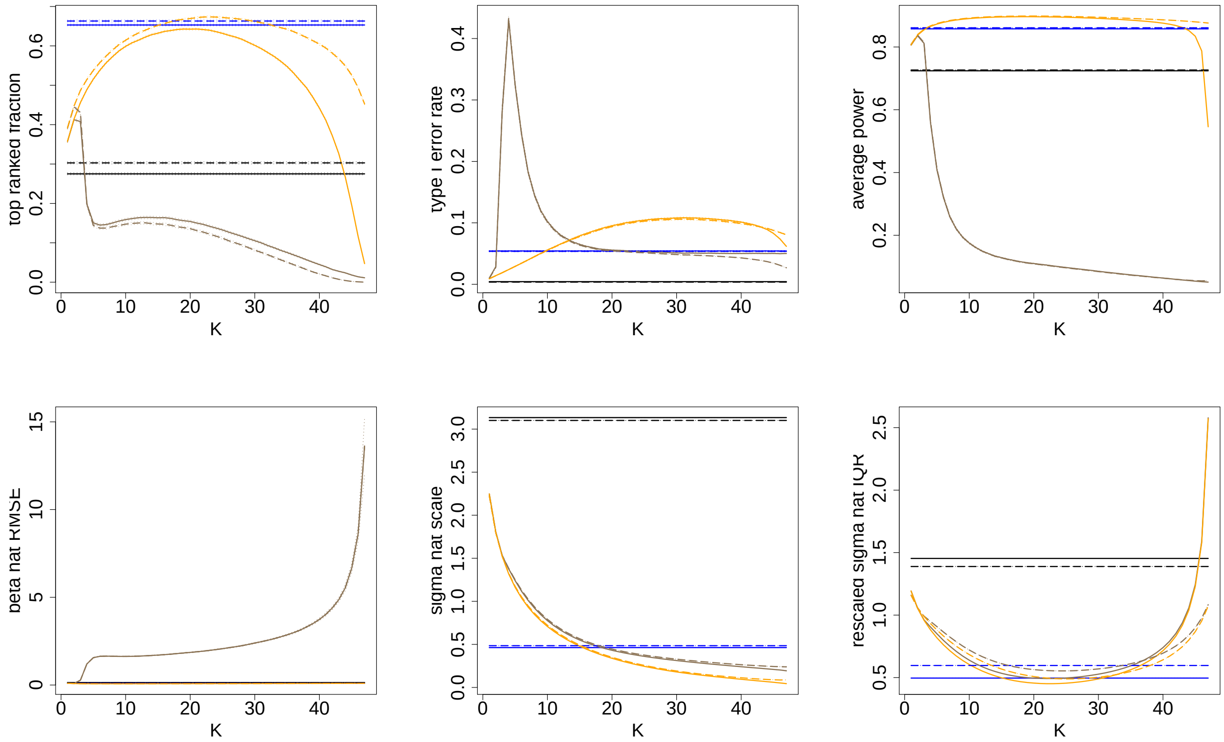


Figure B.7: Moderate decay, lightly correlated, bad controls.

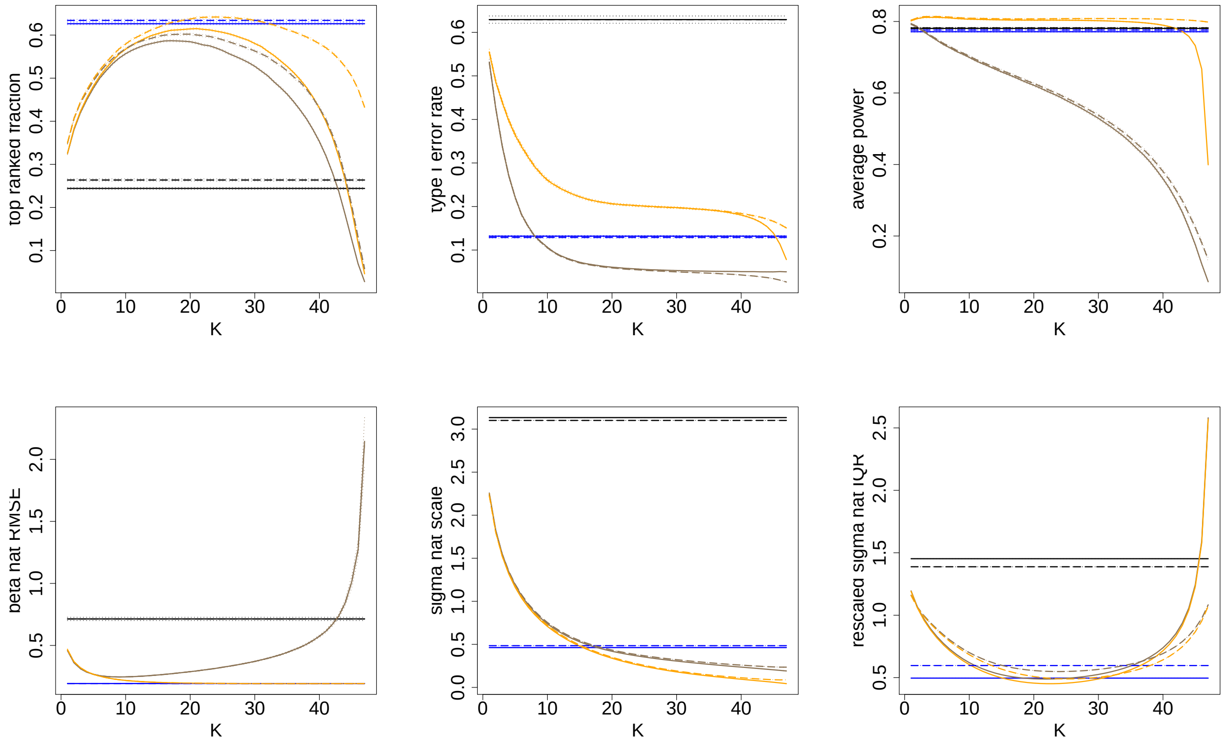


Figure B.8: Moderate decay, highly correlated, good controls.

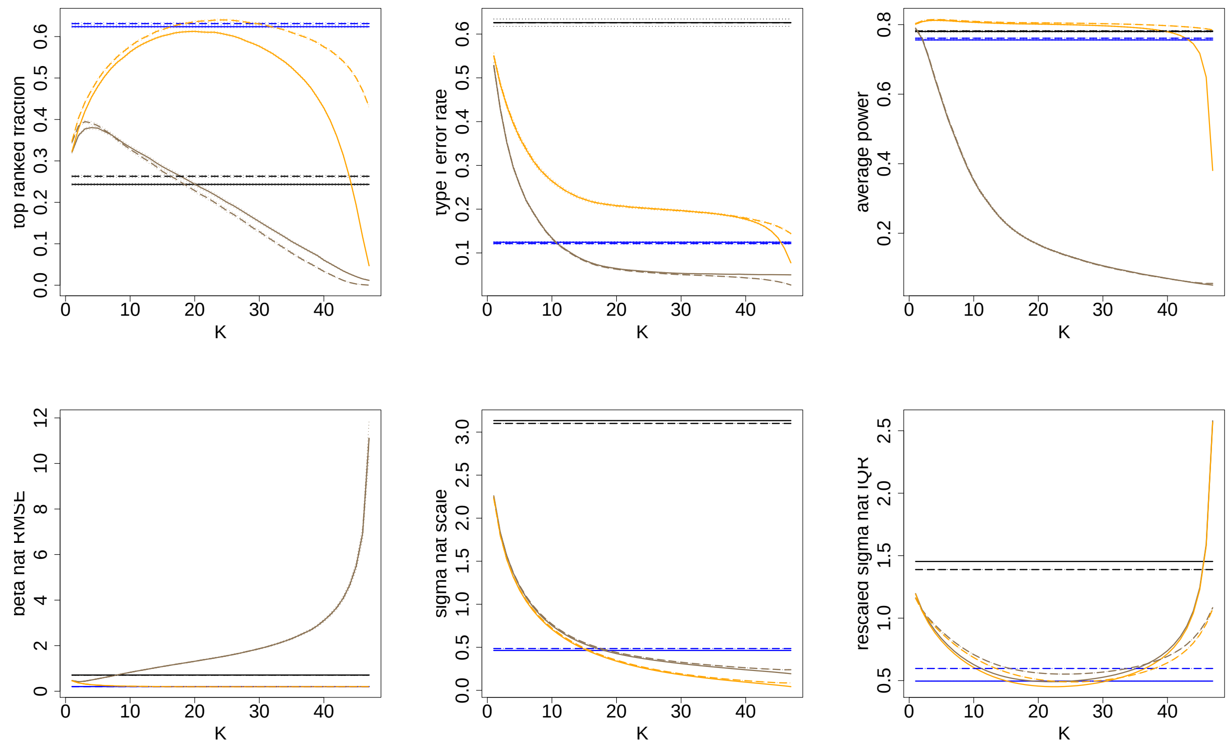


Figure B.9: Moderate decay, highly correlated, bad controls.

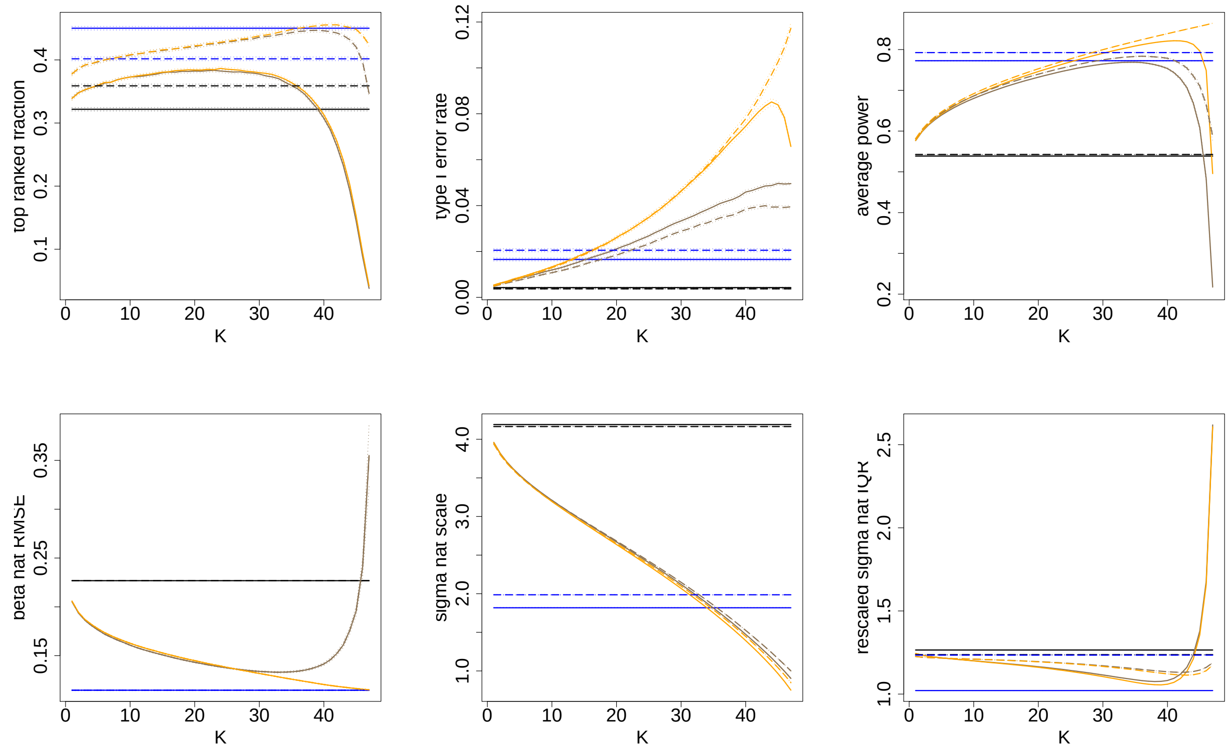


Figure B.10: Slow decay, lightly correlated, good controls.

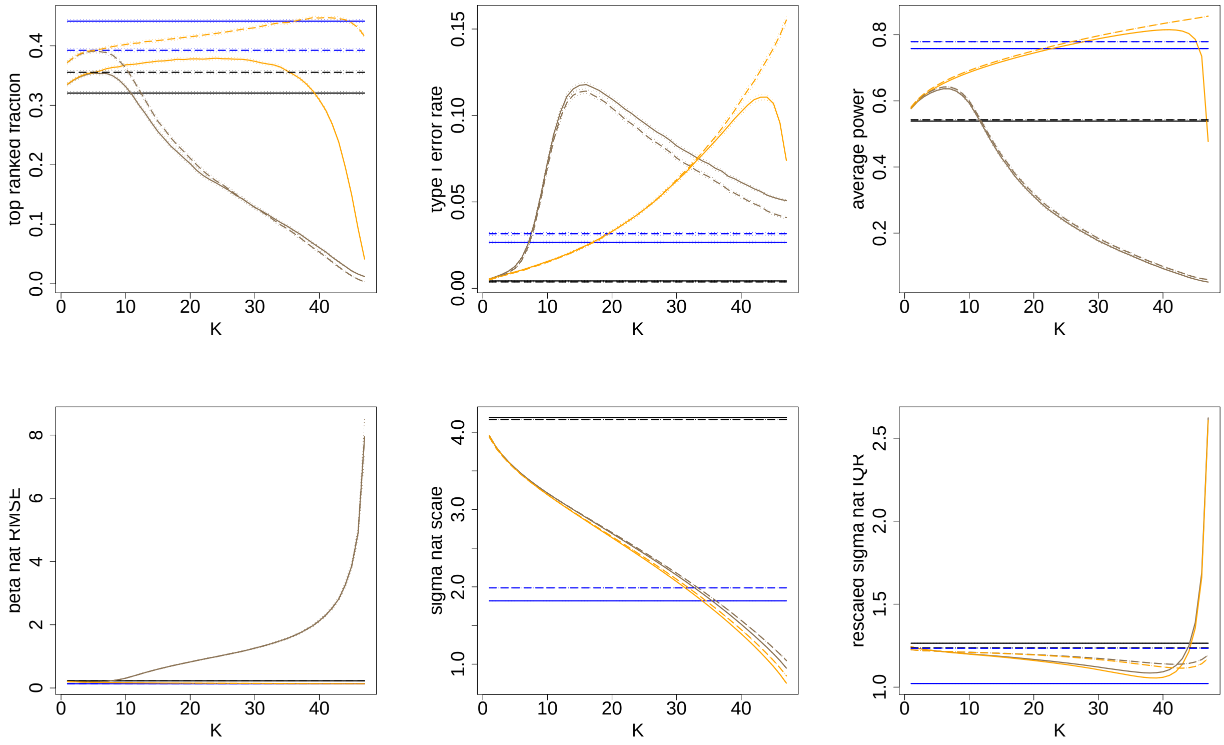


Figure B.11: Slow decay, lightly correlated, bad controls.

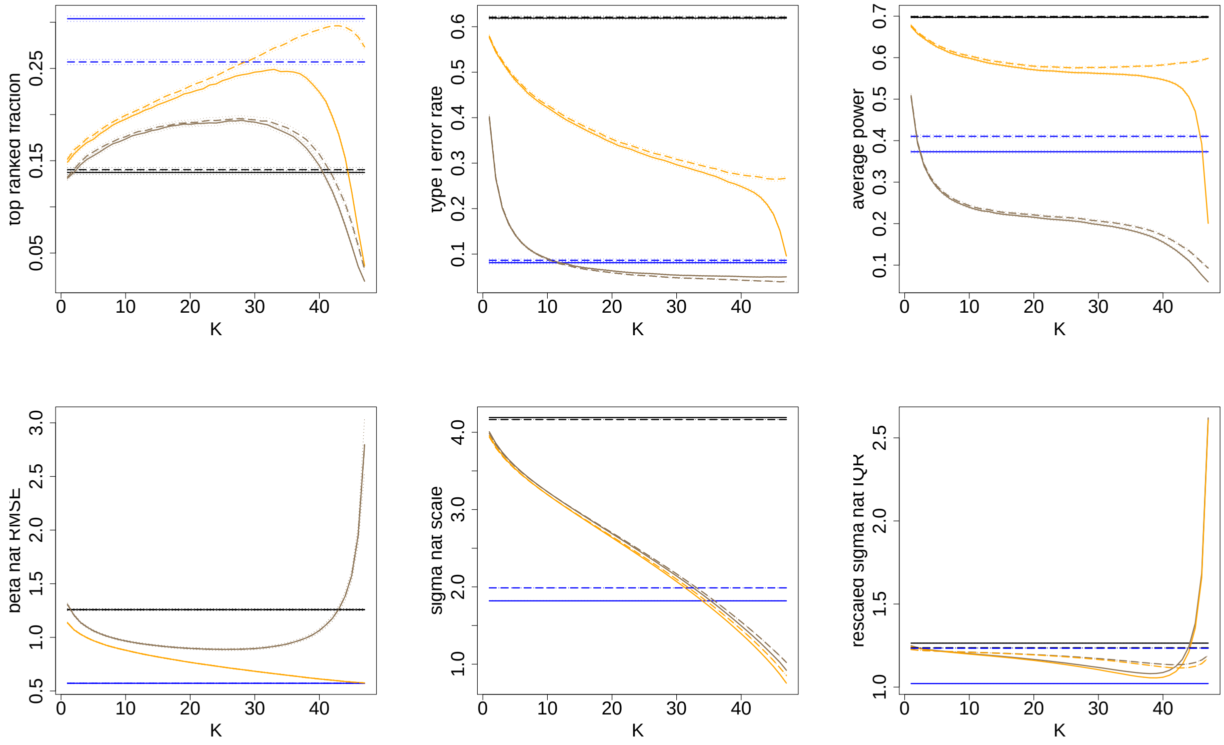


Figure B.12: Slow decay, highly correlated, good controls.

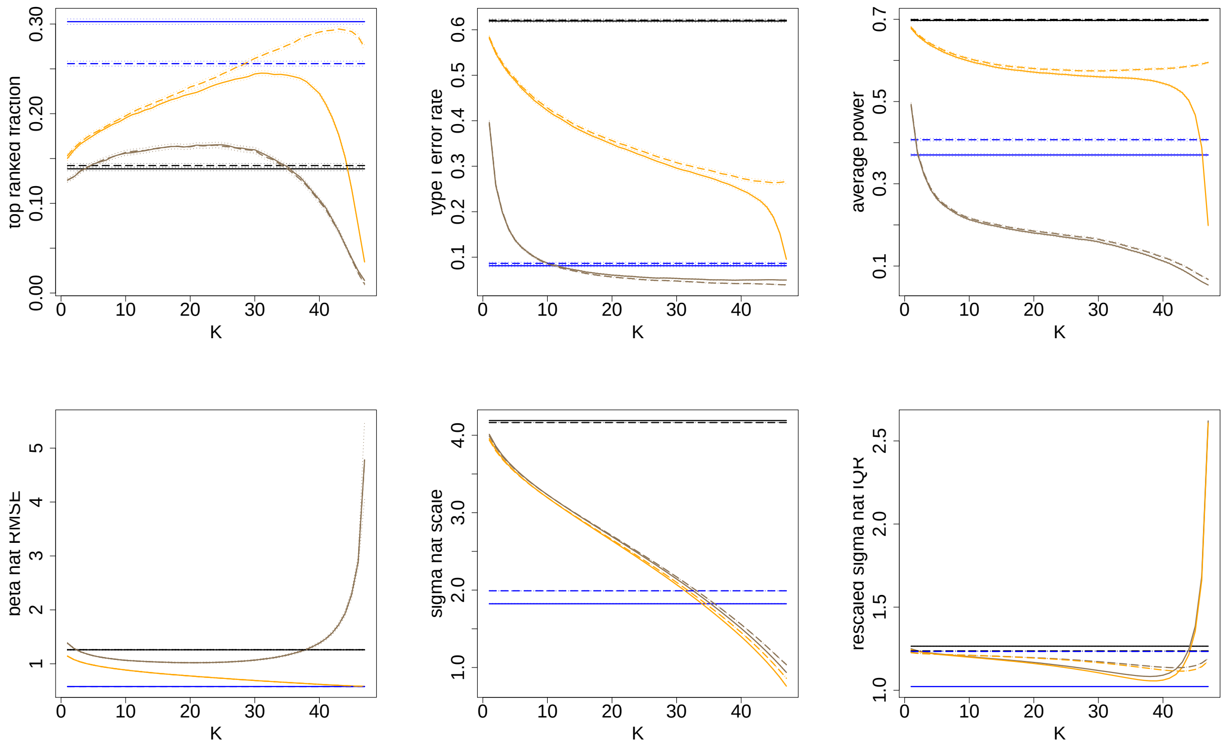


Figure B.13: Slow decay, highly correlated, bad controls.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.27	(5×10^{-3})	0.47	(5×10^{-4})	0.66	(5×10^{-3})	0.707	(5×10^{-4})	53.08	(6×10^{-2})	1.41	(2×10^{-3})
SVA (IRW)	0.66	(6×10^{-3})	0.09	(5×10^{-3})	0.79	(4×10^{-3})	0.164	(2×10^{-3})	6.15	(7×10^{-2})	1.06	(3×10^{-3})
SVA (2-step)	0.66	(6×10^{-3})	0.11	(5×10^{-3})	0.80	(4×10^{-3})	0.168	(3×10^{-3})	6.12	(7×10^{-2})	1.06	(3×10^{-3})
LEAPP	0.66	(6×10^{-3})	0.28	(7×10^{-3})	0.86	(4×10^{-3})	0.102	(2×10^{-3})	5.28	(6×10^{-2})	1.06	(3×10^{-3})
ICE	0.79	(4×10^{-3})	0.01	(2×10^{-4})	0.82	(4×10^{-3})	0.090	(7×10^{-4})				
RUV-4	0.78	(4×10^{-3})	0.07	(1×10^{-3})	0.88	(3×10^{-3})	0.092	(8×10^{-4})	1.00	(3×10^{-4})	0.37	(7×10^{-4})
RUV-inv	0.78	(5×10^{-3})	0.05	(8×10^{-4})	0.86	(4×10^{-3})	0.094	(8×10^{-4})	1.15	(1×10^{-3})	0.39	(5×10^{-4})
RUV-rinv	0.78	(5×10^{-3})	0.06	(2×10^{-3})	0.86	(4×10^{-3})	0.094	(8×10^{-4})	1.40	(2×10^{-3})	0.42	(8×10^{-4})
RUV-inv (Ectl)	0.79	(4×10^{-3})	0.05	(8×10^{-4})	0.87	(4×10^{-3})	0.090	(7×10^{-4})	1.13	(5×10^{-4})	0.35	(4×10^{-4})
RUV-rinv (Ectl)	0.78	(5×10^{-3})	0.06	(2×10^{-3})	0.87	(4×10^{-3})	0.092	(8×10^{-4})	1.37	(8×10^{-4})	0.40	(5×10^{-4})
RUV-inv-rsvar (Ectl)	0.79	(4×10^{-3})	0.04	(5×10^{-4})	0.87	(4×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.78	(5×10^{-3})	0.04	(5×10^{-4})	0.86	(4×10^{-3})						
RUV-inv-evar (Ectl)	0.79	(4×10^{-3})	0.05	(3×10^{-4})	0.87	(4×10^{-3})						
RUV-rinv-evar (Ectl)	0.79	(5×10^{-3})	0.05	(3×10^{-4})	0.86	(4×10^{-3})						

Table B.1: $k = 20$, moderately correlated, $n_c = 1000$, very sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.52	(5×10^{-3})	0.47	(6×10^{-4})	0.66	(2×10^{-3})	0.707	(5×10^{-4})	53.13	(5×10^{-2})	1.41	(2×10^{-3})
SVA (IRW)	0.71	(3×10^{-3})	0.11	(5×10^{-3})	0.79	(2×10^{-3})	0.172	(3×10^{-3})	5.99	(6×10^{-2})	1.05	(3×10^{-3})
SVA (2-step)	0.71	(3×10^{-3})	0.11	(6×10^{-3})	0.79	(2×10^{-3})	0.165	(3×10^{-3})	5.97	(6×10^{-2})	1.05	(3×10^{-3})
LEAPP	0.71	(3×10^{-3})	0.27	(8×10^{-3})	0.86	(2×10^{-3})	0.134	(2×10^{-3})	5.13	(6×10^{-2})	1.05	(3×10^{-3})
ICE	0.80	(3×10^{-3})	0.00	(4×10^{-5})	0.72	(2×10^{-3})	0.091	(6×10^{-4})				
RUV-4	0.89	(2×10^{-3})	0.08	(1×10^{-3})	0.88	(2×10^{-3})	0.092	(6×10^{-4})	1.00	(3×10^{-4})	0.37	(7×10^{-4})
RUV-inv	0.88	(2×10^{-3})	0.05	(8×10^{-4})	0.86	(2×10^{-3})	0.094	(6×10^{-4})	1.16	(1×10^{-3})	0.39	(6×10^{-4})
RUV-rinv	0.87	(2×10^{-3})	0.06	(1×10^{-3})	0.87	(2×10^{-3})	0.094	(7×10^{-4})	1.40	(2×10^{-3})	0.42	(9×10^{-4})
RUV-inv (Ectl)	0.89	(2×10^{-3})	0.05	(7×10^{-4})	0.87	(2×10^{-3})	0.091	(6×10^{-4})	1.14	(5×10^{-4})	0.35	(4×10^{-4})
RUV-rinv (Ectl)	0.88	(2×10^{-3})	0.06	(1×10^{-3})	0.87	(2×10^{-3})	0.093	(7×10^{-4})	1.37	(9×10^{-4})	0.40	(5×10^{-4})
RUV-inv-rsvar (Ectl)	0.89	(2×10^{-3})	0.04	(5×10^{-4})	0.86	(2×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.88	(2×10^{-3})	0.04	(4×10^{-4})	0.86	(2×10^{-3})						
RUV-inv-evar (Ectl)	0.83	(4×10^{-3})	0.01	(2×10^{-4})	0.82	(2×10^{-3})						
RUV-rinv-evar (Ectl)	0.81	(3×10^{-3})	0.01	(3×10^{-4})	0.81	(2×10^{-3})						

Table B.2: $k = 20$, moderately correlated, $n_c = 1000$, sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.27	(4×10^{-3})	0.47	(6×10^{-4})	0.66	(8×10^{-4})	0.708	(4×10^{-4})	53.07	(5×10^{-2})	1.41	(2×10^{-3})
SVA (IRW)	0.22	(1×10^{-2})	0.70	(1×10^{-2})	0.82	(5×10^{-3})	1.082	(6×10^{-2})	6.06	(6×10^{-2})	1.06	(3×10^{-3})
SVA (2-step)	0.42	(6×10^{-3})	0.32	(7×10^{-3})	0.78	(2×10^{-3})	0.359	(6×10^{-2})	5.99	(6×10^{-2})	1.05	(3×10^{-3})
LEAPP	0.46	(4×10^{-3})	0.27	(7×10^{-3})	0.86	(8×10^{-4})	0.169	(2×10^{-3})	5.12	(6×10^{-2})	1.05	(3×10^{-3})
ICE	0.59	(4×10^{-3})	0.00	(9×10^{-6})	0.32	(5×10^{-4})	0.130	(6×10^{-4})				
RUV-4	0.73	(4×10^{-3})	0.07	(1×10^{-3})	0.88	(8×10^{-4})	0.092	(5×10^{-4})	1.00	(3×10^{-4})	0.37	(1×10^{-3})
RUV-inv	0.73	(3×10^{-3})	0.05	(7×10^{-4})	0.86	(8×10^{-4})	0.094	(5×10^{-4})	1.16	(1×10^{-3})	0.39	(5×10^{-4})
RUV-rinv	0.73	(4×10^{-3})	0.06	(1×10^{-3})	0.86	(8×10^{-4})	0.093	(5×10^{-4})	1.40	(2×10^{-3})	0.42	(8×10^{-4})
RUV-inv (Ectl)	0.75	(3×10^{-3})	0.06	(1×10^{-3})	0.87	(8×10^{-4})	0.095	(5×10^{-4})	1.14	(4×10^{-4})	0.35	(4×10^{-4})
RUV-rinv (Ectl)	0.74	(3×10^{-3})	0.07	(2×10^{-3})	0.87	(7×10^{-4})	0.095	(6×10^{-4})	1.38	(8×10^{-4})	0.40	(5×10^{-4})
RUV-inv-rsvar (Ectl)	0.75	(3×10^{-3})	0.04	(4×10^{-4})	0.85	(1×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.74	(3×10^{-3})	0.04	(5×10^{-4})	0.85	(1×10^{-3})						
RUV-inv-evar (Ectl)	0.49	(4×10^{-3})	0.00	(0)	0.24	(3×10^{-4})						
RUV-rinv-evar (Ectl)	0.49	(4×10^{-3})	0.00	(0)	0.24	(3×10^{-4})						

Table B.3: $k = 20$, moderately correlated, $n_c = 1000$, not sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.28	(5×10^{-3})	0.47	(5×10^{-4})	0.67	(4×10^{-3})	0.707	(5×10^{-4})	53.10	(6×10^{-2})	1.41	(2×10^{-3})
SVA (IRW)	0.66	(6×10^{-3})	0.09	(5×10^{-3})	0.79	(4×10^{-3})	0.159	(2×10^{-3})	6.12	(6×10^{-2})	1.06	(3×10^{-3})
SVA (2-step)	0.66	(6×10^{-3})	0.11	(6×10^{-3})	0.80	(4×10^{-3})	0.164	(2×10^{-3})	6.16	(6×10^{-2})	1.06	(3×10^{-3})
LEAPP	0.66	(6×10^{-3})	0.26	(7×10^{-3})	0.86	(4×10^{-3})	0.099	(2×10^{-3})	5.26	(6×10^{-2})	1.06	(3×10^{-3})
ICE	0.78	(4×10^{-3})	0.01	(2×10^{-4})	0.82	(4×10^{-3})	0.090	(6×10^{-4})				
RUV-4	0.56	(1×10^{-2})	0.12	(3×10^{-3})	0.80	(8×10^{-3})	0.154	(3×10^{-3})	0.96	(2×10^{-3})	0.76	(3×10^{-2})
RUV-inv	0.57	(6×10^{-3})	0.02	(1×10^{-3})	0.64	(9×10^{-3})	0.200	(4×10^{-3})	1.57	(1×10^{-2})	1.00	(1×10^{-2})
RUV-rinv	0.74	(4×10^{-3})	0.06	(2×10^{-3})	0.84	(4×10^{-3})	0.111	(1×10^{-3})	2.00	(1×10^{-2})	0.62	(3×10^{-3})
RUV-inv (Ectl)	0.79	(4×10^{-3})	0.05	(7×10^{-4})	0.87	(4×10^{-3})	0.090	(7×10^{-4})	1.14	(5×10^{-4})	0.35	(4×10^{-4})
RUV-rinv (Ectl)	0.79	(4×10^{-3})	0.06	(1×10^{-3})	0.87	(4×10^{-3})	0.092	(7×10^{-4})	1.35	(9×10^{-4})	0.39	(5×10^{-4})
RUV-inv-rsvar (Ectl)	0.79	(4×10^{-3})	0.05	(2×10^{-3})	0.87	(4×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.79	(4×10^{-3})	0.05	(2×10^{-3})	0.87	(4×10^{-3})						
RUV-inv-evar (Ectl)	0.79	(4×10^{-3})	0.05	(4×10^{-4})	0.87	(4×10^{-3})						
RUV-rinv-evar (Ectl)	0.79	(4×10^{-3})	0.05	(3×10^{-4})	0.86	(4×10^{-3})						

Table B.4: $k = 20$, moderately correlated, $n_c = 60$, very sparse.

	Top Rank Frac.	Type 1	Average Power	RMSE($\hat{\beta}$)	AVG($\hat{\sigma}_j^2/\sigma_j^2$)	IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]
Unadjusted	0.52 (4×10^{-3})	0.47 (5×10^{-4})	0.66 (2×10^{-3})	0.707 (5×10^{-4})	53.02 (5×10^{-2})	1.41 (2×10^{-3})
SVA (IRW)	0.72 (4×10^{-3})	0.10 (5×10^{-3})	0.78 (2×10^{-3})	0.170 (3×10^{-3})	6.08 (7×10^{-2})	1.06 (3×10^{-3})
SVA (2-step)	0.72 (4×10^{-3})	0.11 (5×10^{-3})	0.80 (2×10^{-3})	0.163 (3×10^{-3})	6.03 (7×10^{-2})	1.05 (4×10^{-3})
LEAPP	0.72 (4×10^{-3})	0.26 (8×10^{-3})	0.86 (2×10^{-3})	0.133 (2×10^{-3})	5.22 (7×10^{-2})	1.06 (3×10^{-3})
ICE	0.80 (2×10^{-3})	0.00 (4×10^{-5})	0.72 (2×10^{-3})	0.091 (6×10^{-4})		
RUV-4	0.78 (1×10^{-2})	0.12 (3×10^{-3})	0.80 (7×10^{-3})	0.151 (3×10^{-3})	0.96 (1×10^{-3})	0.73 (3×10^{-2})
RUV-inv	0.72 (4×10^{-3})	0.02 (1×10^{-3})	0.62 (1×10^{-2})	0.209 (5×10^{-3})	1.57 (1×10^{-2})	1.01 (1×10^{-2})
RUV-rinv	0.82 (3×10^{-3})	0.06 (2×10^{-3})	0.84 (2×10^{-3})	0.110 (1×10^{-3})	1.98 (1×10^{-2})	0.62 (3×10^{-3})
RUV-inv (Ectl)	0.90 (2×10^{-3})	0.05 (8×10^{-4})	0.87 (2×10^{-3})	0.090 (6×10^{-4})	1.14 (5×10^{-4})	0.35 (4×10^{-4})
RUV-rinv (Ectl)	0.89 (2×10^{-3})	0.06 (1×10^{-3})	0.87 (2×10^{-3})	0.092 (7×10^{-4})	1.35 (1×10^{-3})	0.39 (5×10^{-4})
RUV-inv-rsvar (Ectl)	0.90 (2×10^{-3})	0.05 (2×10^{-3})	0.87 (2×10^{-3})			
RUV-rinv-rsvar (Ectl)	0.89 (2×10^{-3})	0.05 (2×10^{-3})	0.86 (3×10^{-3})			
RUV-inv-evar (Ectl)	0.83 (4×10^{-3})	0.01 (2×10^{-4})	0.82 (2×10^{-3})			
RUV-rinv-evar (Ectl)	0.83 (3×10^{-3})	0.01 (3×10^{-4})	0.82 (2×10^{-3})			

Table B.5: $k = 20$, moderately correlated, $n_c = 60$, sparse.

	Top Rank Frac.	Type 1	Average Power	RMSE($\hat{\beta}$)	AVG($\hat{\sigma}_j^2/\sigma_j^2$)	IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]
Unadjusted	0.28 (4×10^{-3})	0.47 (7×10^{-4})	0.66 (6×10^{-4})	0.708 (5×10^{-4})	53.13 (6×10^{-2})	1.40 (2×10^{-3})
SVA (IRW)	0.21 (1×10^{-2})	0.72 (1×10^{-2})	0.83 (5×10^{-3})	1.173 (6×10^{-2})	6.12 (7×10^{-2})	1.06 (3×10^{-3})
SVA (2-step)	0.42 (4×10^{-3})	0.32 (6×10^{-3})	0.78 (1×10^{-3})	0.295 (7×10^{-3})	6.02 (6×10^{-2})	1.06 (3×10^{-3})
LEAPP	0.45 (4×10^{-3})	0.28 (8×10^{-3})	0.86 (9×10^{-4})	0.173 (2×10^{-3})	5.17 (6×10^{-2})	1.05 (3×10^{-3})
ICE	0.59 (4×10^{-3})	0.00 (6×10^{-6})	0.32 (5×10^{-4})	0.129 (7×10^{-4})		
RUV-4	0.52 (2×10^{-2})	0.12 (3×10^{-3})	0.80 (6×10^{-3})	0.151 (2×10^{-3})	0.96 (1×10^{-3})	0.74 (3×10^{-2})
RUV-inv	0.43 (5×10^{-3})	0.02 (9×10^{-4})	0.64 (7×10^{-3})	0.197 (4×10^{-3})	1.60 (1×10^{-2})	1.00 (1×10^{-2})
RUV-rinv	0.61 (4×10^{-3})	0.06 (2×10^{-3})	0.84 (1×10^{-3})	0.111 (1×10^{-3})	2.00 (1×10^{-2})	0.63 (3×10^{-3})
RUV-inv (Ectl)	0.75 (4×10^{-3})	0.07 (1×10^{-3})	0.86 (9×10^{-4})	0.097 (7×10^{-4})	1.14 (5×10^{-4})	0.35 (4×10^{-4})
RUV-rinv (Ectl)	0.74 (4×10^{-3})	0.07 (2×10^{-3})	0.87 (8×10^{-4})	0.097 (8×10^{-4})	1.35 (1×10^{-3})	0.39 (5×10^{-4})
RUV-inv-rsvar (Ectl)	0.75 (4×10^{-3})	0.08 (3×10^{-3})	0.87 (2×10^{-3})			
RUV-rinv-rsvar (Ectl)	0.74 (4×10^{-3})	0.07 (3×10^{-3})	0.86 (2×10^{-3})			
RUV-inv-evar (Ectl)	0.49 (5×10^{-3})	0.00 (2×10^{-6})	0.24 (4×10^{-4})			
RUV-rinv-evar (Ectl)	0.48 (5×10^{-3})	0.00 (0)	0.24 (4×10^{-4})			

Table B.6: $k = 20$, moderately correlated, $n_c = 60$, not sparse.

	Top Rank Frac.	Type 1	Average Power	RMSE($\hat{\beta}$)	AVG($\hat{\sigma}_j^2/\sigma_j^2$)	IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]
Unadjusted	0.12 (4×10^{-3})	0.62 (9×10^{-3})	0.73 (6×10^{-3})	1.091 (2×10^{-2})	52.98 (5×10^{-2})	1.41 (2×10^{-3})
SVA (IRW)	0.29 (1×10^{-2})	0.60 (3×10^{-2})	0.79 (9×10^{-3})	0.892 (5×10^{-2})	8.81 (0.3)	1.18 (1×10^{-2})
SVA (2-step)	0.36 (8×10^{-3})	0.36 (1×10^{-2})	0.73 (6×10^{-3})	0.400 (8×10^{-3})	6.09 (7×10^{-2})	1.06 (3×10^{-3})
LEAPP	0.36 (7×10^{-3})	0.64 (6×10^{-3})	0.86 (4×10^{-3})	0.304 (7×10^{-3})	5.16 (6×10^{-2})	1.06 (3×10^{-3})
ICE	0.63 (6×10^{-3})	0.02 (4×10^{-4})	0.72 (5×10^{-3})	0.173 (2×10^{-3})		
RUV-4	0.57 (1×10^{-2})	0.17 (1×10^{-2})	0.78 (5×10^{-3})	0.211 (7×10^{-3})	1.36 (6×10^{-2})	0.45 (1×10^{-2})
RUV-inv	0.62 (6×10^{-3})	0.06 (1×10^{-3})	0.75 (5×10^{-3})	0.182 (2×10^{-3})	1.16 (1×10^{-3})	0.39 (5×10^{-4})
RUV-rinv	0.61 (6×10^{-3})	0.11 (4×10^{-3})	0.77 (5×10^{-3})	0.187 (3×10^{-3})	1.45 (8×10^{-3})	0.43 (2×10^{-3})
RUV-inv (Ectl)	0.63 (6×10^{-3})	0.07 (1×10^{-3})	0.76 (5×10^{-3})	0.175 (2×10^{-3})	1.14 (5×10^{-4})	0.35 (4×10^{-4})
RUV-rinv (Ectl)	0.61 (6×10^{-3})	0.11 (4×10^{-3})	0.77 (5×10^{-3})	0.184 (3×10^{-3})	1.42 (7×10^{-3})	0.41 (2×10^{-3})
RUV-inv-rsvar (Ectl)	0.63 (6×10^{-3})	0.04 (5×10^{-4})	0.74 (6×10^{-3})			
RUV-rinv-rsvar (Ectl)	0.61 (6×10^{-3})	0.04 (5×10^{-4})	0.71 (6×10^{-3})			
RUV-inv-evar (Ectl)	0.64 (6×10^{-3})	0.05 (2×10^{-4})	0.75 (5×10^{-3})			
RUV-rinv-evar (Ectl)	0.62 (6×10^{-3})	0.05 (2×10^{-4})	0.73 (6×10^{-3})			

Table B.7: $k = 20$, highly correlated, $n_c = 1000$, very sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.27	(7×10^{-3})	0.64	(7×10^{-3})	0.73	(4×10^{-3})	1.136	(2×10^{-2})	53.04	(6×10^{-2})	1.40	(2×10^{-3})
SVA (IRW)	0.53	(1×10^{-2})	0.67	(2×10^{-2})	0.81	(9×10^{-3})	1.070	(5×10^{-2})	9.32	(0.3)	1.20	(1×10^{-2})
SVA (2-step)	0.64	(7×10^{-3})	0.36	(1×10^{-2})	0.73	(5×10^{-3})	0.433	(2×10^{-2})	6.20	(0.1)	1.06	(5×10^{-3})
LEAPP	0.65	(5×10^{-3})	0.64	(7×10^{-3})	0.87	(2×10^{-3})	0.321	(7×10^{-3})	5.15	(6×10^{-2})	1.05	(3×10^{-3})
ICE	0.80	(3×10^{-3})	0.01	(3×10^{-4})	0.64	(3×10^{-3})	0.177	(2×10^{-3})				
RUV-4	0.83	(5×10^{-3})	0.17	(1×10^{-2})	0.78	(3×10^{-3})	0.210	(6×10^{-3})	1.32	(4×10^{-2})	0.45	(1×10^{-2})
RUV-inv	0.86	(2×10^{-3})	0.06	(1×10^{-3})	0.75	(3×10^{-3})	0.184	(2×10^{-3})	1.16	(1×10^{-3})	0.39	(5×10^{-4})
RUV-rinv	0.85	(3×10^{-3})	0.11	(3×10^{-3})	0.77	(3×10^{-3})	0.188	(3×10^{-3})	1.44	(5×10^{-3})	0.43	(2×10^{-3})
RUV-inv (Ectl)	0.87	(2×10^{-3})	0.07	(1×10^{-3})	0.76	(3×10^{-3})	0.177	(2×10^{-3})	1.14	(5×10^{-4})	0.35	(4×10^{-4})
RUV-rinv (Ectl)	0.85	(3×10^{-3})	0.11	(4×10^{-3})	0.77	(3×10^{-3})	0.186	(3×10^{-3})	1.41	(5×10^{-3})	0.41	(2×10^{-3})
RUV-inv-rsvar (Ectl)	0.87	(2×10^{-3})	0.04	(5×10^{-4})	0.74	(4×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.85	(3×10^{-3})	0.04	(5×10^{-4})	0.71	(5×10^{-3})						
RUV-inv-evar (Ectl)	0.84	(3×10^{-3})	0.02	(4×10^{-4})	0.71	(4×10^{-3})						
RUV-rinv-evar (Ectl)	0.80	(3×10^{-3})	0.03	(4×10^{-4})	0.70	(4×10^{-3})						

Table B.8: $k = 20$, highly correlated, $n_c = 1000$, sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.22	(4×10^{-3})	0.63	(1×10^{-2})	0.72	(4×10^{-3})	1.124	(3×10^{-2})	53.06	(5×10^{-2})	1.40	(2×10^{-3})
SVA (IRW)	0.18	(6×10^{-3})	0.86	(7×10^{-3})	0.88	(4×10^{-3})	1.708	(4×10^{-2})	6.89	(0.1)	1.10	(5×10^{-3})
SVA (2-step)	0.27	(1×10^{-2})	0.59	(2×10^{-2})	0.75	(1×10^{-2})	1.254	(0.1)	7.08	(0.2)	1.10	(9×10^{-3})
LEAPP	0.42	(5×10^{-3})	0.65	(7×10^{-3})	0.87	(9×10^{-4})	0.366	(7×10^{-3})	5.20	(6×10^{-2})	1.05	(3×10^{-3})
ICE	0.57	(4×10^{-3})	0.00	(2×10^{-5})	0.30	(9×10^{-4})	0.232	(3×10^{-3})				
RUV-4	0.67	(8×10^{-3})	0.17	(1×10^{-2})	0.78	(3×10^{-3})	0.207	(6×10^{-3})	1.34	(6×10^{-2})	0.45	(1×10^{-2})
RUV-inv	0.72	(4×10^{-3})	0.06	(1×10^{-3})	0.75	(3×10^{-3})	0.181	(2×10^{-3})	1.16	(1×10^{-3})	0.39	(6×10^{-4})
RUV-rinv	0.70	(4×10^{-3})	0.11	(4×10^{-3})	0.77	(2×10^{-3})	0.185	(3×10^{-3})	1.45	(8×10^{-3})	0.43	(2×10^{-3})
RUV-inv (Ectl)	0.74	(4×10^{-3})	0.08	(2×10^{-3})	0.77	(2×10^{-3})	0.178	(2×10^{-3})	1.14	(6×10^{-4})	0.35	(4×10^{-4})
RUV-rinv (Ectl)	0.71	(4×10^{-3})	0.13	(4×10^{-3})	0.78	(2×10^{-3})	0.188	(3×10^{-3})	1.42	(7×10^{-3})	0.41	(2×10^{-3})
RUV-inv-rsvar (Ectl)	0.74	(4×10^{-3})	0.04	(5×10^{-4})	0.73	(3×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.71	(4×10^{-3})	0.04	(5×10^{-4})	0.71	(4×10^{-3})						
RUV-inv-evar (Ectl)	0.51	(5×10^{-3})	0.00	(5×10^{-6})	0.23	(5×10^{-4})						
RUV-rinv-evar (Ectl)	0.50	(5×10^{-3})	0.00	(4×10^{-6})	0.22	(5×10^{-4})						

Table B.9: $k = 20$, highly correlated, $n_c = 1000$, not sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.11	(4×10^{-3})	0.62	(9×10^{-3})	0.72	(6×10^{-3})	1.084	(2×10^{-2})	53.01	(6×10^{-2})	1.41	(2×10^{-3})
SVA (IRW)	0.30	(1×10^{-2})	0.56	(3×10^{-2})	0.78	(9×10^{-3})	0.834	(5×10^{-2})	8.44	(0.3)	1.16	(1×10^{-2})
SVA (2-step)	0.35	(9×10^{-3})	0.39	(1×10^{-2})	0.74	(6×10^{-3})	0.438	(2×10^{-2})	6.26	(8×10^{-2})	1.07	(4×10^{-3})
LEAPP	0.36	(8×10^{-3})	0.65	(7×10^{-3})	0.86	(4×10^{-3})	0.314	(8×10^{-3})	5.22	(6×10^{-2})	1.06	(3×10^{-3})
ICE	0.62	(6×10^{-3})	0.02	(5×10^{-4})	0.71	(6×10^{-3})	0.172	(2×10^{-3})				
RUV-4	0.42	(1×10^{-2})	0.14	(4×10^{-3})	0.68	(9×10^{-3})	0.275	(5×10^{-3})	0.97	(3×10^{-3})	0.58	(2×10^{-2})
RUV-inv	0.34	(8×10^{-3})	0.02	(1×10^{-3})	0.40	(1×10^{-2})	0.397	(9×10^{-3})	1.58	(1×10^{-2})	1.02	(1×10^{-2})
RUV-rinv	0.55	(7×10^{-3})	0.09	(4×10^{-3})	0.72	(6×10^{-3})	0.219	(3×10^{-3})	2.02	(2×10^{-2})	0.63	(4×10^{-3})
RUV-inv (Ectl)	0.63	(7×10^{-3})	0.07	(1×10^{-3})	0.76	(6×10^{-3})	0.174	(2×10^{-3})	1.14	(5×10^{-4})	0.35	(5×10^{-4})
RUV-rinv (Ectl)	0.61	(7×10^{-3})	0.10	(2×10^{-3})	0.77	(6×10^{-3})	0.180	(3×10^{-3})	1.36	(1×10^{-3})	0.39	(6×10^{-4})
RUV-inv-rsvar (Ectl)	0.63	(7×10^{-3})	0.05	(2×10^{-3})	0.74	(6×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.61	(7×10^{-3})	0.05	(2×10^{-3})	0.73	(7×10^{-3})						
RUV-inv-evar (Ectl)	0.63	(7×10^{-3})	0.05	(2×10^{-4})	0.75	(6×10^{-3})						
RUV-rinv-evar (Ectl)	0.62	(7×10^{-3})	0.05	(2×10^{-4})	0.74	(7×10^{-3})						

Table B.10: $k = 20$, highly correlated, $n_c = 60$, very sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.28	(8×10^{-3})	0.64	(1×10^{-2})	0.73	(4×10^{-3})	1.135	(3×10^{-2})	53.09	(6×10^{-2})	1.40	(2×10^{-3})
SVA (IRW)	0.55	(1×10^{-2})	0.61	(3×10^{-2})	0.79	(9×10^{-3})	0.965	(6×10^{-2})	8.86	(0.3)	1.18	(1×10^{-2})
SVA (2-step)	0.63	(6×10^{-3})	0.39	(1×10^{-2})	0.74	(4×10^{-3})	0.430	(1×10^{-2})	6.13	(9×10^{-2})	1.06	(4×10^{-3})
LEAPP	0.64	(5×10^{-3})	0.66	(6×10^{-3})	0.87	(1×10^{-3})	0.334	(7×10^{-3})	5.20	(6×10^{-2})	1.05	(3×10^{-3})
ICE	0.80	(3×10^{-3})	0.01	(3×10^{-4})	0.64	(3×10^{-3})	0.176	(2×10^{-3})				
RUV-4	0.77	(7×10^{-3})	0.13	(4×10^{-3})	0.68	(6×10^{-3})	0.280	(5×10^{-3})	0.97	(1×10^{-3})	0.59	(2×10^{-2})
RUV-inv	0.66	(6×10^{-3})	0.02	(1×10^{-3})	0.42	(9×10^{-3})	0.390	(8×10^{-3})	1.59	(1×10^{-2})	0.99	(1×10^{-2})
RUV-rinv	0.79	(3×10^{-3})	0.10	(4×10^{-3})	0.72	(4×10^{-3})	0.223	(3×10^{-3})	2.02	(2×10^{-2})	0.63	(4×10^{-3})
RUV-inv (Ectl)	0.87	(2×10^{-3})	0.07	(1×10^{-3})	0.77	(4×10^{-3})	0.176	(2×10^{-3})	1.14	(5×10^{-4})	0.35	(4×10^{-4})
RUV-rinv (Ectl)	0.86	(2×10^{-3})	0.10	(3×10^{-3})	0.77	(3×10^{-3})	0.182	(2×10^{-3})	1.35	(1×10^{-3})	0.39	(6×10^{-4})
RUV-inv-rsvar (Ectl)	0.87	(2×10^{-3})	0.05	(2×10^{-3})	0.75	(5×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.86	(2×10^{-3})	0.05	(2×10^{-3})	0.73	(5×10^{-3})						
RUV-inv-evar (Ectl)	0.84	(3×10^{-3})	0.02	(4×10^{-4})	0.72	(4×10^{-3})						
RUV-rinv-evar (Ectl)	0.81	(3×10^{-3})	0.03	(4×10^{-4})	0.70	(4×10^{-3})						

Table B.11: $k = 20$, highly correlated, $n_c = 60$, sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.23	(4×10^{-3})	0.63	(1×10^{-2})	0.72	(4×10^{-3})	1.110	(2×10^{-2})	53.14	(6×10^{-2})	1.41	(2×10^{-3})
SVA (IRW)	0.19	(6×10^{-3})	0.86	(7×10^{-3})	0.88	(4×10^{-3})	1.638	(3×10^{-2})	7.06	(9×10^{-2})	1.11	(4×10^{-3})
SVA (2-step)	0.28	(1×10^{-2})	0.59	(2×10^{-2})	0.76	(1×10^{-2})	1.159	(8×10^{-2})	7.09	(0.2)	1.10	(7×10^{-3})
LEAPP	0.42	(5×10^{-3})	0.65	(6×10^{-3})	0.86	(8×10^{-4})	0.368	(7×10^{-3})	5.32	(6×10^{-2})	1.06	(3×10^{-3})
ICE	0.58	(4×10^{-3})	0.00	(3×10^{-5})	0.30	(9×10^{-4})	0.230	(3×10^{-3})				
RUV-4	0.56	(1×10^{-2})	0.14	(5×10^{-3})	0.68	(8×10^{-3})	0.282	(6×10^{-3})	0.97	(2×10^{-3})	0.60	(2×10^{-2})
RUV-inv	0.40	(5×10^{-3})	0.02	(1×10^{-3})	0.41	(9×10^{-3})	0.395	(9×10^{-3})	1.56	(1×10^{-2})	0.99	(1×10^{-2})
RUV-rinv	0.60	(5×10^{-3})	0.10	(4×10^{-3})	0.72	(3×10^{-3})	0.221	(4×10^{-3})	1.99	(1×10^{-2})	0.62	(3×10^{-3})
RUV-inv (Ectl)	0.74	(4×10^{-3})	0.09	(2×10^{-3})	0.77	(3×10^{-3})	0.181	(3×10^{-3})	1.14	(5×10^{-4})	0.35	(4×10^{-4})
RUV-rinv (Ectl)	0.72	(4×10^{-3})	0.12	(3×10^{-3})	0.78	(3×10^{-3})	0.187	(3×10^{-3})	1.36	(1×10^{-3})	0.39	(5×10^{-4})
RUV-inv-rsvar (Ectl)	0.74	(4×10^{-3})	0.07	(2×10^{-3})	0.75	(4×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.72	(4×10^{-3})	0.06	(2×10^{-3})	0.73	(5×10^{-3})						
RUV-inv-evar (Ectl)	0.50	(4×10^{-3})	0.00	(8×10^{-6})	0.23	(5×10^{-4})						
RUV-rinv-evar (Ectl)	0.50	(5×10^{-3})	0.00	(6×10^{-6})	0.22	(6×10^{-4})						

Table B.12: $k = 20$, highly correlated, $n_c = 60$, not sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.44	(5×10^{-3})	0.50	(6×10^{-4})	0.76	(4×10^{-3})	0.482	(3×10^{-4})	22.97	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.73	(4×10^{-3})	0.09	(3×10^{-3})	0.84	(4×10^{-3})	0.120	(1×10^{-3})	3.36	(2×10^{-2})	0.84	(3×10^{-3})
SVA (2-step)	0.74	(4×10^{-3})	0.09	(3×10^{-3})	0.85	(4×10^{-3})	0.118	(1×10^{-3})	3.36	(2×10^{-2})	0.84	(3×10^{-3})
LEAPP	0.74	(4×10^{-3})	0.22	(5×10^{-3})	0.89	(3×10^{-3})	0.079	(6×10^{-4})	3.01	(2×10^{-2})	0.83	(3×10^{-3})
ICE	0.78	(4×10^{-3})	0.02	(3×10^{-4})	0.84	(4×10^{-3})	0.092	(5×10^{-4})				
RUV-4	0.77	(4×10^{-3})	0.10	(2×10^{-3})	0.88	(3×10^{-3})	0.096	(6×10^{-4})	1.43	(9×10^{-3})	0.46	(2×10^{-3})
RUV-inv	0.78	(4×10^{-3})	0.06	(1×10^{-3})	0.87	(3×10^{-3})	0.096	(5×10^{-4})	1.59	(2×10^{-3})	0.50	(7×10^{-4})
RUV-rinv	0.78	(4×10^{-3})	0.07	(1×10^{-3})	0.87	(3×10^{-3})	0.095	(5×10^{-4})	1.89	(2×10^{-3})	0.56	(8×10^{-4})
RUV-inv (Ectl)	0.78	(4×10^{-3})	0.07	(1×10^{-3})	0.87	(3×10^{-3})	0.093	(5×10^{-4})	1.58	(1×10^{-3})	0.47	(6×10^{-4})
RUV-rinv (Ectl)	0.78	(4×10^{-3})	0.07	(1×10^{-3})	0.88	(3×10^{-3})	0.095	(5×10^{-4})	1.87	(2×10^{-3})	0.55	(7×10^{-4})
RUV-inv-rsvar (Ectl)	0.78	(4×10^{-3})	0.04	(5×10^{-4})	0.86	(4×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.78	(4×10^{-3})	0.04	(5×10^{-4})	0.86	(4×10^{-3})						
RUV-inv-evar (Ectl)	0.79	(4×10^{-3})	0.05	(3×10^{-4})	0.87	(4×10^{-3})						
RUV-rinv-evar (Ectl)	0.78	(4×10^{-3})	0.05	(3×10^{-4})	0.86	(4×10^{-3})						

Table B.13: $k = 70$, moderately correlated, $n_c = 1000$, very sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.60	(4×10^{-3})	0.50	(5×10^{-4})	0.76	(2×10^{-3})	0.482	(3×10^{-4})	22.91	(3×10^{-2})	1.46	(2×10^{-3})
SVA (IRW)	0.76	(3×10^{-3})	0.19	(7×10^{-3})	0.82	(2×10^{-3})	0.185	(5×10^{-3})	3.41	(3×10^{-2})	0.84	(3×10^{-3})
SVA (2-step)	0.78	(4×10^{-3})	0.10	(3×10^{-3})	0.84	(2×10^{-3})	0.123	(1×10^{-3})	3.39	(3×10^{-2})	0.84	(3×10^{-3})
LEAPP	0.78	(3×10^{-3})	0.22	(5×10^{-3})	0.88	(2×10^{-3})	0.107	(7×10^{-4})	3.07	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.79	(3×10^{-3})	0.00	(8×10^{-5})	0.76	(2×10^{-3})	0.093	(4×10^{-4})				
RUV-4	0.86	(2×10^{-3})	0.10	(2×10^{-3})	0.88	(2×10^{-3})	0.096	(5×10^{-4})	1.42	(9×10^{-3})	0.46	(2×10^{-3})
RUV-inv	0.86	(3×10^{-3})	0.06	(9×10^{-4})	0.86	(2×10^{-3})	0.096	(4×10^{-4})	1.59	(2×10^{-3})	0.50	(8×10^{-4})
RUV-rinv	0.85	(3×10^{-3})	0.07	(1×10^{-3})	0.87	(2×10^{-3})	0.096	(5×10^{-4})	1.89	(3×10^{-3})	0.56	(9×10^{-4})
RUV-inv (Ectl)	0.87	(2×10^{-3})	0.06	(9×10^{-4})	0.87	(2×10^{-3})	0.093	(4×10^{-4})	1.58	(1×10^{-3})	0.47	(7×10^{-4})
RUV-rinv (Ectl)	0.85	(3×10^{-3})	0.07	(1×10^{-3})	0.87	(2×10^{-3})	0.094	(4×10^{-4})	1.87	(2×10^{-3})	0.55	(8×10^{-4})
RUV-inv-rsvar (Ectl)	0.87	(2×10^{-3})	0.04	(5×10^{-4})	0.86	(2×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.85	(3×10^{-3})	0.04	(5×10^{-4})	0.85	(2×10^{-3})						
RUV-inv-evar (Ectl)	0.81	(3×10^{-3})	0.01	(2×10^{-4})	0.81	(2×10^{-3})						
RUV-rinv-evar (Ectl)	0.80	(3×10^{-3})	0.01	(3×10^{-4})	0.81	(2×10^{-3})						

Table B.14: $k = 70$, moderately correlated, $n_c = 1000$, sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.32	(4×10^{-3})	0.50	(8×10^{-4})	0.75	(7×10^{-4})	0.482	(3×10^{-4})	22.97	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.38	(5×10^{-3})	0.72	(3×10^{-3})	0.85	(7×10^{-4})	0.585	(5×10^{-3})	3.38	(2×10^{-2})	0.84	(3×10^{-3})
SVA (2-step)	0.45	(6×10^{-3})	0.51	(7×10^{-3})	0.80	(1×10^{-3})	0.422	(1×10^{-2})	3.38	(2×10^{-2})	0.84	(3×10^{-3})
LEAPP	0.55	(4×10^{-3})	0.22	(5×10^{-3})	0.88	(6×10^{-4})	0.129	(1×10^{-3})	3.04	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.50	(4×10^{-3})	0.00	(2×10^{-5})	0.40	(6×10^{-4})	0.127	(5×10^{-4})				
RUV-4	0.68	(4×10^{-3})	0.10	(1×10^{-3})	0.88	(6×10^{-4})	0.096	(5×10^{-4})	1.42	(8×10^{-3})	0.46	(2×10^{-3})
RUV-inv	0.70	(4×10^{-3})	0.06	(1×10^{-3})	0.86	(7×10^{-4})	0.096	(5×10^{-4})	1.59	(2×10^{-3})	0.50	(8×10^{-4})
RUV-rinv	0.69	(4×10^{-3})	0.07	(1×10^{-3})	0.87	(6×10^{-4})	0.096	(5×10^{-4})	1.89	(3×10^{-3})	0.56	(9×10^{-4})
RUV-inv (Ectl)	0.71	(4×10^{-3})	0.07	(1×10^{-3})	0.86	(6×10^{-4})	0.098	(4×10^{-4})	1.58	(1×10^{-3})	0.47	(6×10^{-4})
RUV-rinv (Ectl)	0.69	(4×10^{-3})	0.08	(1×10^{-3})	0.87	(6×10^{-4})	0.097	(5×10^{-4})	1.87	(2×10^{-3})	0.55	(8×10^{-4})
RUV-inv-rsvar (Ectl)	0.71	(4×10^{-3})	0.04	(4×10^{-4})	0.85	(1×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.69	(4×10^{-3})	0.04	(4×10^{-4})	0.85	(1×10^{-3})						
RUV-inv-evar (Ectl)	0.49	(5×10^{-3})	0.00	(0)	0.24	(4×10^{-4})						
RUV-rinv-evar (Ectl)	0.48	(5×10^{-3})	0.00	(0)	0.24	(4×10^{-4})						

Table B.15: $k = 70$, moderately correlated, $n_c = 1000$, not sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.44	(5×10^{-3})	0.50	(6×10^{-4})	0.75	(4×10^{-3})	0.481	(3×10^{-4})	22.94	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.72	(5×10^{-3})	0.09	(3×10^{-3})	0.84	(4×10^{-3})	0.122	(1×10^{-3})	3.42	(3×10^{-2})	0.84	(3×10^{-3})
SVA (2-step)	0.73	(4×10^{-3})	0.10	(3×10^{-3})	0.84	(4×10^{-3})	0.121	(1×10^{-3})	3.40	(3×10^{-2})	0.84	(3×10^{-3})
LEAPP	0.73	(4×10^{-3})	0.22	(5×10^{-3})	0.88	(3×10^{-3})	0.080	(8×10^{-4})	3.08	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.78	(4×10^{-3})	0.02	(3×10^{-4})	0.84	(4×10^{-3})	0.093	(5×10^{-4})				
RUV-4	0.54	(1×10^{-2})	0.13	(3×10^{-3})	0.79	(9×10^{-3})	0.159	(3×10^{-3})	1.13	(6×10^{-3})	0.77	(4×10^{-2})
RUV-inv	0.55	(7×10^{-3})	0.02	(1×10^{-3})	0.62	(9×10^{-3})	0.207	(4×10^{-3})	1.90	(1×10^{-2})	1.02	(8×10^{-3})
RUV-rinv	0.74	(4×10^{-3})	0.07	(2×10^{-3})	0.84	(4×10^{-3})	0.110	(8×10^{-4})	2.24	(9×10^{-3})	0.67	(2×10^{-3})
RUV-inv (Ectl)	0.78	(4×10^{-3})	0.07	(1×10^{-3})	0.87	(4×10^{-3})	0.093	(5×10^{-4})	1.57	(1×10^{-3})	0.47	(7×10^{-4})
RUV-rinv (Ectl)	0.77	(4×10^{-3})	0.07	(1×10^{-3})	0.87	(3×10^{-3})	0.095	(5×10^{-4})	1.83	(2×10^{-3})	0.54	(8×10^{-4})
RUV-inv-rsvar (Ectl)	0.78	(4×10^{-3})	0.05	(2×10^{-3})	0.86	(4×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.77	(4×10^{-3})	0.05	(2×10^{-3})	0.86	(4×10^{-3})						
RUV-inv-evar (Ectl)	0.78	(4×10^{-3})	0.05	(3×10^{-4})	0.86	(4×10^{-3})						
RUV-rinv-evar (Ectl)	0.78	(4×10^{-3})	0.05	(3×10^{-4})	0.86	(4×10^{-3})						

Table B.16: $k = 70$, moderately correlated, $n_c = 60$, very sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.59	(4×10^{-3})	0.50	(6×10^{-4})	0.75	(2×10^{-3})	0.481	(4×10^{-4})	22.95	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.76	(4×10^{-3})	0.20	(7×10^{-3})	0.81	(3×10^{-3})	0.190	(5×10^{-3})	3.38	(3×10^{-2})	0.84	(3×10^{-3})
SVA (2-step)	0.78	(3×10^{-3})	0.10	(3×10^{-3})	0.84	(2×10^{-3})	0.123	(1×10^{-3})	3.36	(3×10^{-2})	0.84	(3×10^{-3})
LEAPP	0.78	(3×10^{-3})	0.22	(4×10^{-3})	0.88	(1×10^{-3})	0.106	(7×10^{-4})	3.04	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.79	(3×10^{-3})	0.00	(7×10^{-5})	0.76	(2×10^{-3})	0.093	(4×10^{-4})				
RUV-4	0.75	(1×10^{-2})	0.13	(3×10^{-3})	0.78	(8×10^{-3})	0.159	(3×10^{-3})	1.13	(8×10^{-3})	0.77	(4×10^{-2})
RUV-inv	0.71	(4×10^{-3})	0.02	(1×10^{-3})	0.62	(7×10^{-3})	0.209	(4×10^{-3})	1.90	(1×10^{-2})	1.03	(9×10^{-3})
RUV-rinv	0.81	(3×10^{-3})	0.07	(1×10^{-3})	0.84	(2×10^{-3})	0.109	(7×10^{-4})	2.25	(1×10^{-2})	0.67	(2×10^{-3})
RUV-inv (Ectl)	0.87	(2×10^{-3})	0.06	(9×10^{-4})	0.87	(2×10^{-3})	0.092	(4×10^{-4})	1.58	(1×10^{-3})	0.47	(7×10^{-4})
RUV-rinv (Ectl)	0.85	(2×10^{-3})	0.07	(1×10^{-3})	0.87	(2×10^{-3})	0.094	(5×10^{-4})	1.83	(2×10^{-3})	0.54	(8×10^{-4})
RUV-inv-rsvar (Ectl)	0.87	(2×10^{-3})	0.05	(2×10^{-3})	0.86	(3×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.85	(2×10^{-3})	0.05	(2×10^{-3})	0.85	(3×10^{-3})						
RUV-inv-evar (Ectl)	0.81	(4×10^{-3})	0.01	(3×10^{-4})	0.81	(2×10^{-3})						
RUV-rinv-evar (Ectl)	0.80	(4×10^{-3})	0.01	(3×10^{-4})	0.81	(2×10^{-3})						

Table B.17: $k = 70$, moderately correlated, $n_c = 60$, sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.31	(4×10^{-3})	0.50	(7×10^{-4})	0.75	(6×10^{-4})	0.481	(4×10^{-4})	22.91	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.37	(5×10^{-3})	0.72	(3×10^{-3})	0.86	(6×10^{-4})	0.586	(6×10^{-3})	3.36	(2×10^{-2})	0.84	(3×10^{-3})
SVA (2-step)	0.44	(8×10^{-3})	0.51	(7×10^{-3})	0.80	(1×10^{-3})	0.421	(1×10^{-2})	3.36	(2×10^{-2})	0.84	(3×10^{-3})
LEAPP	0.55	(4×10^{-3})	0.24	(4×10^{-3})	0.88	(6×10^{-4})	0.131	(9×10^{-4})	3.02	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.50	(4×10^{-3})	0.00	(3×10^{-5})	0.40	(5×10^{-4})	0.127	(5×10^{-4})				
RUV-4	0.51	(2×10^{-2})	0.14	(3×10^{-3})	0.79	(9×10^{-3})	0.158	(3×10^{-3})	1.13	(6×10^{-3})	0.76	(4×10^{-2})
RUV-inv	0.40	(4×10^{-3})	0.02	(1×10^{-3})	0.61	(7×10^{-3})	0.216	(5×10^{-3})	1.93	(1×10^{-2})	1.04	(9×10^{-3})
RUV-rinv	0.60	(4×10^{-3})	0.07	(2×10^{-3})	0.84	(9×10^{-4})	0.111	(7×10^{-4})	2.27	(1×10^{-2})	0.68	(2×10^{-3})
RUV-inv (Ectl)	0.71	(3×10^{-3})	0.08	(1×10^{-3})	0.86	(7×10^{-4})	0.100	(5×10^{-4})	1.58	(1×10^{-3})	0.47	(6×10^{-4})
RUV-rinv (Ectl)	0.69	(3×10^{-3})	0.09	(2×10^{-3})	0.87	(6×10^{-4})	0.099	(5×10^{-4})	1.83	(2×10^{-3})	0.54	(8×10^{-4})
RUV-inv-rsvar (Ectl)	0.71	(3×10^{-3})	0.08	(3×10^{-3})	0.86	(2×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.69	(3×10^{-3})	0.07	(3×10^{-3})	0.86	(2×10^{-3})						
RUV-inv-evar (Ectl)	0.48	(5×10^{-3})	0.00	(2×10^{-6})	0.24	(3×10^{-4})						
RUV-rinv-evar (Ectl)	0.48	(5×10^{-3})	0.00	(0)	0.24	(4×10^{-4})						

Table B.18: $k = 70$, moderately correlated, $n_c = 60$, not sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.26	(6×10^{-3})	0.62	(1×10^{-2})	0.78	(4×10^{-3})	0.686	(2×10^{-2})	22.91	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.42	(8×10^{-3})	0.40	(2×10^{-2})	0.79	(6×10^{-3})	0.352	(1×10^{-2})	3.59	(4×10^{-2})	0.87	(4×10^{-3})
SVA (2-step)	0.46	(7×10^{-3})	0.36	(8×10^{-3})	0.80	(5×10^{-3})	0.276	(6×10^{-3})	3.41	(3×10^{-2})	0.84	(4×10^{-3})
LEAPP	0.47	(6×10^{-3})	0.59	(5×10^{-3})	0.89	(3×10^{-3})	0.204	(4×10^{-3})	3.05	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.59	(6×10^{-3})	0.05	(5×10^{-4})	0.72	(5×10^{-3})	0.183	(2×10^{-3})				
RUV-4	0.51	(6×10^{-3})	0.31	(1×10^{-2})	0.82	(5×10^{-3})	0.230	(4×10^{-3})	2.50	(6×10^{-2})	0.69	(1×10^{-2})
RUV-inv	0.58	(5×10^{-3})	0.14	(2×10^{-3})	0.77	(5×10^{-3})	0.189	(2×10^{-3})	1.59	(2×10^{-3})	0.50	(7×10^{-4})
RUV-rinv	0.56	(5×10^{-3})	0.22	(4×10^{-3})	0.81	(5×10^{-3})	0.195	(2×10^{-3})	2.00	(6×10^{-3})	0.58	(2×10^{-3})
RUV-inv (Ectl)	0.59	(5×10^{-3})	0.15	(3×10^{-3})	0.79	(5×10^{-3})	0.183	(2×10^{-3})	1.58	(1×10^{-3})	0.47	(7×10^{-4})
RUV-rinv (Ectl)	0.56	(5×10^{-3})	0.23	(4×10^{-3})	0.81	(5×10^{-3})	0.193	(2×10^{-3})	1.97	(5×10^{-3})	0.58	(1×10^{-3})
RUV-inv-rsvar (Ectl)	0.59	(5×10^{-3})	0.04	(5×10^{-4})	0.70	(5×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.56	(5×10^{-3})	0.05	(4×10^{-4})	0.69	(6×10^{-3})						
RUV-inv-evar (Ectl)	0.61	(5×10^{-3})	0.05	(2×10^{-4})	0.72	(5×10^{-3})						
RUV-rinv-evar (Ectl)	0.59	(5×10^{-3})	0.05	(2×10^{-4})	0.71	(6×10^{-3})						

Table B.19: $k = 70$, highly correlated, $n_c = 1000$, very sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.49	(5×10^{-3})	0.64	(1×10^{-2})	0.78	(3×10^{-3})	0.723	(2×10^{-2})	22.96	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.64	(7×10^{-3})	0.55	(2×10^{-2})	0.79	(6×10^{-3})	0.506	(2×10^{-2})	3.67	(4×10^{-2})	0.87	(5×10^{-3})
SVA (2-step)	0.73	(4×10^{-3})	0.36	(9×10^{-3})	0.80	(3×10^{-3})	0.282	(4×10^{-3})	3.33	(2×10^{-2})	0.83	(3×10^{-3})
LEAPP	0.74	(4×10^{-3})	0.60	(5×10^{-3})	0.89	(1×10^{-3})	0.223	(3×10^{-3})	2.98	(2×10^{-2})	0.83	(3×10^{-3})
ICE	0.77	(3×10^{-3})	0.02	(5×10^{-4})	0.67	(3×10^{-3})	0.188	(2×10^{-3})				
RUV-4	0.78	(5×10^{-3})	0.30	(1×10^{-2})	0.81	(3×10^{-3})	0.235	(4×10^{-3})	2.47	(6×10^{-2})	0.68	(1×10^{-2})
RUV-inv	0.83	(3×10^{-3})	0.13	(2×10^{-3})	0.77	(3×10^{-3})	0.192	(2×10^{-3})	1.59	(2×10^{-3})	0.50	(7×10^{-4})
RUV-rinv	0.81	(3×10^{-3})	0.22	(4×10^{-3})	0.80	(2×10^{-3})	0.197	(2×10^{-3})	1.99	(6×10^{-3})	0.58	(2×10^{-3})
RUV-inv (Ectl)	0.84	(3×10^{-3})	0.15	(3×10^{-3})	0.78	(3×10^{-3})	0.186	(2×10^{-3})	1.58	(1×10^{-3})	0.47	(6×10^{-4})
RUV-rinv (Ectl)	0.81	(3×10^{-3})	0.23	(4×10^{-3})	0.81	(3×10^{-3})	0.197	(2×10^{-3})	1.97	(6×10^{-3})	0.57	(2×10^{-3})
RUV-inv-rsvar (Ectl)	0.84	(3×10^{-3})	0.04	(5×10^{-4})	0.70	(3×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.81	(3×10^{-3})	0.05	(5×10^{-4})	0.68	(4×10^{-3})						
RUV-inv-evar (Ectl)	0.77	(3×10^{-3})	0.03	(3×10^{-4})	0.69	(3×10^{-3})						
RUV-rinv-evar (Ectl)	0.74	(4×10^{-3})	0.03	(3×10^{-4})	0.67	(3×10^{-3})						

Table B.20: $k = 70$, highly correlated, $n_c = 1000$, sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.29	(5×10^{-3})	0.63	(1×10^{-2})	0.78	(3×10^{-3})	0.716	(2×10^{-2})	22.93	(2×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.26	(1×10^{-2})	0.82	(8×10^{-3})	0.89	(3×10^{-3})	1.104	(5×10^{-2})	3.44	(3×10^{-2})	0.85	(4×10^{-3})
SVA (2-step)	0.26	(1×10^{-2})	0.74	(1×10^{-2})	0.84	(4×10^{-3})	1.308	(9×10^{-2})	3.49	(3×10^{-2})	0.85	(4×10^{-3})
LEAPP	0.51	(5×10^{-3})	0.59	(6×10^{-3})	0.89	(6×10^{-4})	0.256	(4×10^{-3})	3.05	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.48	(4×10^{-3})	0.00	(1×10^{-4})	0.37	(1×10^{-3})	0.236	(4×10^{-3})				
RUV-4	0.58	(7×10^{-3})	0.30	(1×10^{-2})	0.81	(3×10^{-3})	0.234	(4×10^{-3})	2.49	(7×10^{-2})	0.69	(1×10^{-2})
RUV-inv	0.67	(4×10^{-3})	0.13	(2×10^{-3})	0.77	(3×10^{-3})	0.192	(2×10^{-3})	1.59	(2×10^{-3})	0.50	(7×10^{-4})
RUV-rinv	0.65	(4×10^{-3})	0.22	(5×10^{-3})	0.81	(2×10^{-3})	0.198	(2×10^{-3})	1.99	(6×10^{-3})	0.58	(2×10^{-3})
RUV-inv (Ectl)	0.68	(4×10^{-3})	0.17	(3×10^{-3})	0.79	(2×10^{-3})	0.190	(2×10^{-3})	1.58	(1×10^{-3})	0.47	(6×10^{-4})
RUV-rinv (Ectl)	0.65	(4×10^{-3})	0.24	(5×10^{-3})	0.81	(2×10^{-3})	0.200	(2×10^{-3})	1.98	(6×10^{-3})	0.58	(2×10^{-3})
RUV-inv-rsvar (Ectl)	0.68	(4×10^{-3})	0.05	(4×10^{-4})	0.70	(3×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.65	(4×10^{-3})	0.05	(5×10^{-4})	0.68	(4×10^{-3})						
RUV-inv-evar (Ectl)	0.48	(4×10^{-3})	0.00	(3×10^{-6})	0.22	(5×10^{-4})						
RUV-rinv-evar (Ectl)	0.46	(5×10^{-3})	0.00	(0)	0.22	(5×10^{-4})						

Table B.21: $k = 70$, highly correlated, $n_c = 1000$, not sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.23	(6×10^{-3})	0.62	(1×10^{-2})	0.78	(5×10^{-3})	0.699	(2×10^{-2})	22.95	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.40	(8×10^{-3})	0.43	(2×10^{-2})	0.80	(5×10^{-3})	0.379	(2×10^{-2})	3.65	(5×10^{-2})	0.87	(6×10^{-3})
SVA (2-step)	0.46	(7×10^{-3})	0.36	(9×10^{-3})	0.81	(5×10^{-3})	0.274	(4×10^{-3})	3.37	(2×10^{-2})	0.84	(3×10^{-3})
LEAPP	0.46	(7×10^{-3})	0.59	(5×10^{-3})	0.89	(3×10^{-3})	0.205	(4×10^{-3})	3.04	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.58	(5×10^{-3})	0.05	(5×10^{-4})	0.72	(5×10^{-3})	0.185	(2×10^{-3})				
RUV-4	0.43	(6×10^{-3})	0.22	(6×10^{-3})	0.75	(6×10^{-3})	0.257	(3×10^{-3})	1.31	(2×10^{-2})	0.50	(7×10^{-3})
RUV-inv	0.31	(6×10^{-3})	0.03	(1×10^{-3})	0.39	(1×10^{-2})	0.430	(9×10^{-3})	1.93	(1×10^{-2})	1.03	(9×10^{-3})
RUV-rinv	0.51	(6×10^{-3})	0.18	(5×10^{-3})	0.77	(5×10^{-3})	0.222	(2×10^{-3})	2.27	(9×10^{-3})	0.67	(2×10^{-3})
RUV-inv (Ectl)	0.58	(6×10^{-3})	0.14	(3×10^{-3})	0.79	(5×10^{-3})	0.185	(2×10^{-3})	1.58	(1×10^{-3})	0.47	(7×10^{-4})
RUV-rinv (Ectl)	0.56	(6×10^{-3})	0.20	(4×10^{-3})	0.81	(5×10^{-3})	0.191	(2×10^{-3})	1.86	(2×10^{-3})	0.54	(9×10^{-4})
RUV-inv-rsvar (Ectl)	0.58	(6×10^{-3})	0.05	(2×10^{-3})	0.71	(6×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.56	(6×10^{-3})	0.05	(2×10^{-3})	0.70	(6×10^{-3})						
RUV-inv-evar (Ectl)	0.60	(6×10^{-3})	0.05	(2×10^{-4})	0.73	(5×10^{-3})						
RUV-rinv-evar (Ectl)	0.59	(5×10^{-3})	0.05	(2×10^{-4})	0.72	(5×10^{-3})						

Table B.22: $k = 70$, highly correlated, $n_c = 60$, very sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.50	(6×10^{-3})	0.65	(1×10^{-2})	0.79	(3×10^{-3})	0.749	(2×10^{-2})	22.97	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.65	(8×10^{-3})	0.50	(2×10^{-2})	0.78	(5×10^{-3})	0.478	(2×10^{-2})	3.66	(5×10^{-2})	0.87	(5×10^{-3})
SVA (2-step)	0.73	(4×10^{-3})	0.34	(9×10^{-3})	0.80	(3×10^{-3})	0.274	(6×10^{-3})	3.39	(3×10^{-2})	0.84	(3×10^{-3})
LEAPP	0.74	(4×10^{-3})	0.58	(5×10^{-3})	0.89	(1×10^{-3})	0.216	(3×10^{-3})	3.05	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.77	(3×10^{-3})	0.02	(5×10^{-4})	0.67	(3×10^{-3})	0.186	(2×10^{-3})				
RUV-4	0.79	(5×10^{-3})	0.22	(6×10^{-3})	0.74	(6×10^{-3})	0.262	(5×10^{-3})	1.30	(2×10^{-2})	0.50	(8×10^{-3})
RUV-inv	0.63	(7×10^{-3})	0.02	(1×10^{-3})	0.39	(1×10^{-2})	0.429	(1×10^{-2})	1.94	(1×10^{-2})	1.04	(1×10^{-2})
RUV-rinv	0.79	(3×10^{-3})	0.18	(4×10^{-3})	0.77	(3×10^{-3})	0.221	(2×10^{-3})	2.28	(1×10^{-2})	0.68	(2×10^{-3})
RUV-inv (Ectl)	0.84	(3×10^{-3})	0.15	(3×10^{-3})	0.79	(3×10^{-3})	0.184	(2×10^{-3})	1.58	(1×10^{-3})	0.47	(7×10^{-4})
RUV-rinv (Ectl)	0.82	(3×10^{-3})	0.20	(3×10^{-3})	0.81	(3×10^{-3})	0.190	(2×10^{-3})	1.85	(3×10^{-3})	0.55	(1×10^{-3})
RUV-inv-rsvar (Ectl)	0.84	(3×10^{-3})	0.06	(2×10^{-3})	0.72	(4×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.82	(3×10^{-3})	0.06	(2×10^{-3})	0.71	(5×10^{-3})						
RUV-inv-evar (Ectl)	0.77	(3×10^{-3})	0.03	(3×10^{-4})	0.70	(3×10^{-3})						
RUV-rinv-evar (Ectl)	0.75	(3×10^{-3})	0.03	(3×10^{-4})	0.69	(3×10^{-3})						

Table B.23: $k = 70$, highly correlated, $n_c = 60$, sparse.

	Top Rank Frac.		Type 1		Average Power		RMSE($\hat{\beta}$)		AVG($\hat{\sigma}_j^2/\sigma_j^2$)		IQR[log($\hat{\sigma}_j^2/\sigma_j^2$)]	
Unadjusted	0.29	(5×10^{-3})	0.64	(1×10^{-2})	0.78	(3×10^{-3})	0.731	(2×10^{-2})	22.90	(3×10^{-2})	1.45	(2×10^{-3})
SVA (IRW)	0.25	(1×10^{-2})	0.83	(8×10^{-3})	0.89	(3×10^{-3})	1.163	(5×10^{-2})	3.45	(3×10^{-2})	0.85	(3×10^{-3})
SVA (2-step)	0.23	(1×10^{-2})	0.76	(1×10^{-2})	0.84	(5×10^{-3})	1.423	(8×10^{-2})	3.54	(3×10^{-2})	0.86	(4×10^{-3})
LEAPP	0.51	(5×10^{-3})	0.59	(6×10^{-3})	0.89	(5×10^{-4})	0.257	(4×10^{-3})	3.05	(3×10^{-2})	0.84	(3×10^{-3})
ICE	0.49	(4×10^{-3})	0.00	(9×10^{-5})	0.37	(1×10^{-3})	0.240	(4×10^{-3})				
RUV-4	0.61	(5×10^{-3})	0.22	(5×10^{-3})	0.75	(6×10^{-3})	0.260	(4×10^{-3})	1.33	(2×10^{-2})	0.49	(6×10^{-3})
RUV-inv	0.38	(5×10^{-3})	0.02	(1×10^{-3})	0.38	(9×10^{-3})	0.434	(8×10^{-3})	1.92	(1×10^{-2})	1.03	(9×10^{-3})
RUV-rinv	0.58	(4×10^{-3})	0.17	(4×10^{-3})	0.76	(3×10^{-3})	0.224	(2×10^{-3})	2.29	(1×10^{-2})	0.68	(2×10^{-3})
RUV-inv (Ectl)	0.67	(4×10^{-3})	0.17	(3×10^{-3})	0.79	(3×10^{-3})	0.193	(2×10^{-3})	1.58	(1×10^{-3})	0.47	(6×10^{-4})
RUV-rinv (Ectl)	0.65	(4×10^{-3})	0.22	(3×10^{-3})	0.81	(2×10^{-3})	0.199	(2×10^{-3})	1.86	(3×10^{-3})	0.55	(9×10^{-4})
RUV-inv-rsvar (Ectl)	0.67	(4×10^{-3})	0.07	(3×10^{-3})	0.72	(4×10^{-3})						
RUV-rinv-rsvar (Ectl)	0.65	(4×10^{-3})	0.06	(3×10^{-3})	0.70	(4×10^{-3})						
RUV-inv-evar (Ectl)	0.49	(4×10^{-3})	0.00	(3×10^{-6})	0.22	(5×10^{-4})						
RUV-rinv-evar (Ectl)	0.48	(4×10^{-3})	0.00	(2×10^{-6})	0.22	(5×10^{-4})						

Table B.24: $k = 70$, highly correlated, $n_c = 60$, not sparse.

B.3 Data Results (Figures)

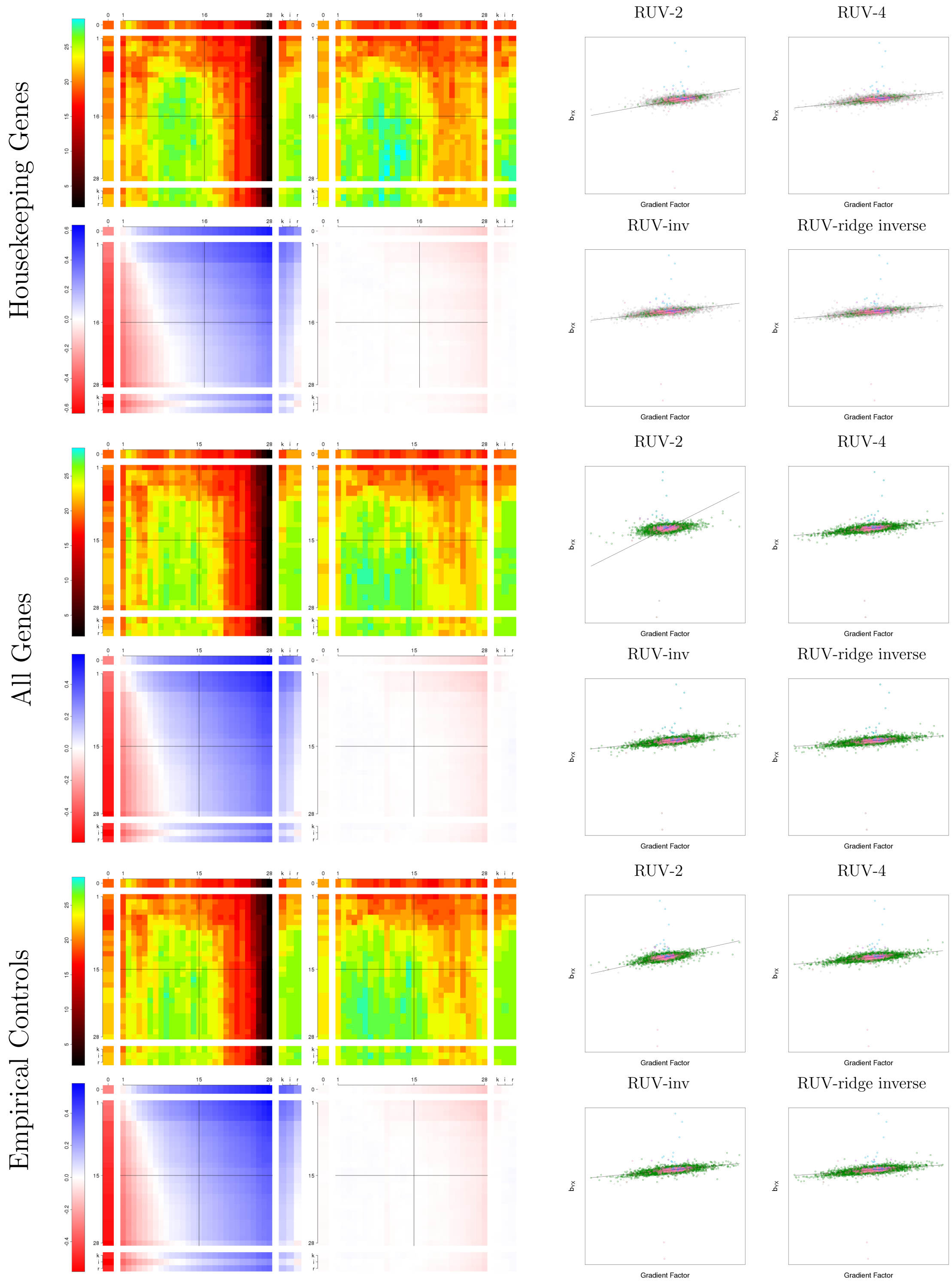


Figure B.14: Alzheimer's (Preprocessed). X/Y gene counts are out of the top 40 genes.

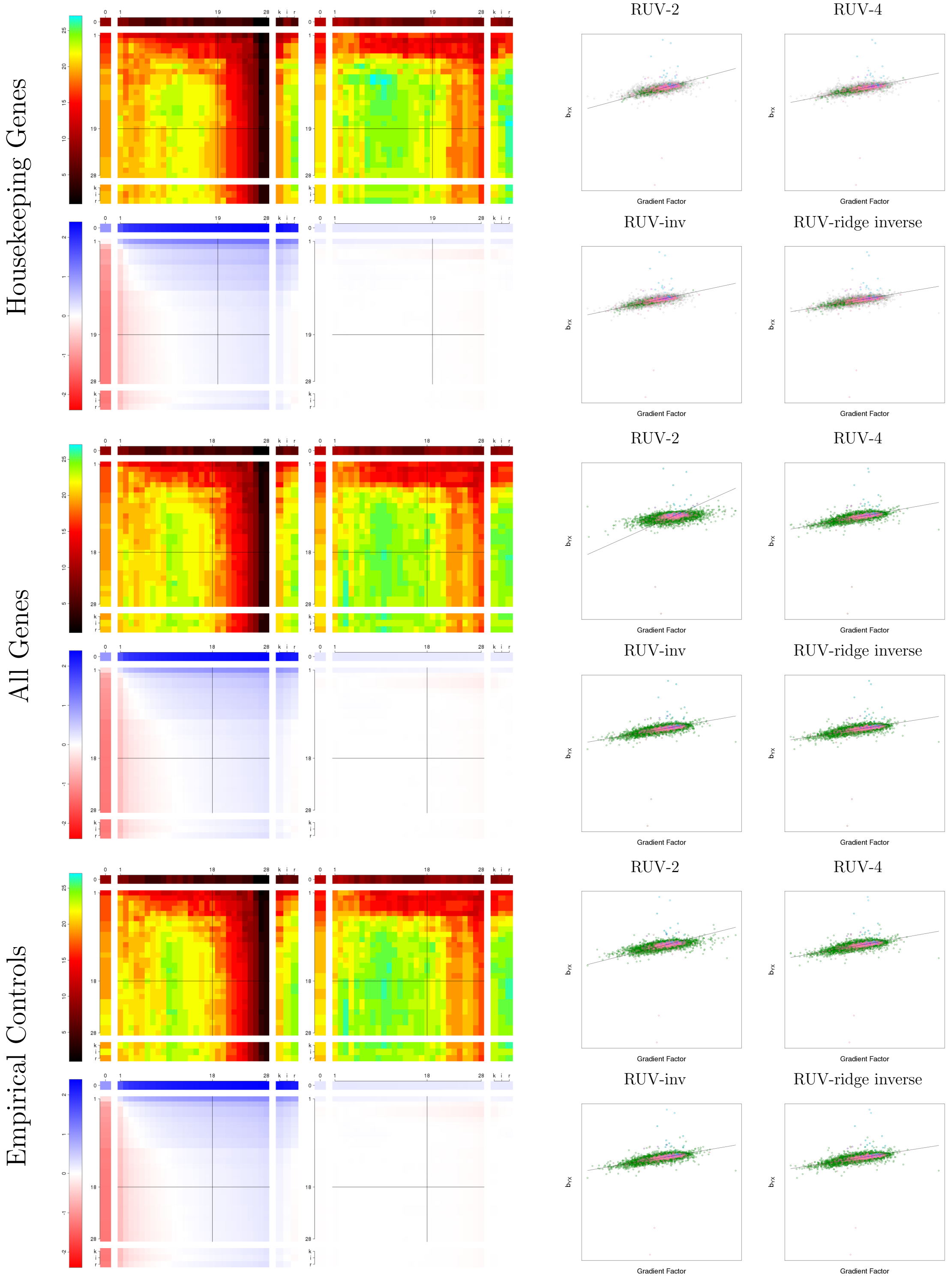


Figure B.15: Alzheimer's (Not Preprocessed). X/Y gene counts are out of the top 40 genes.

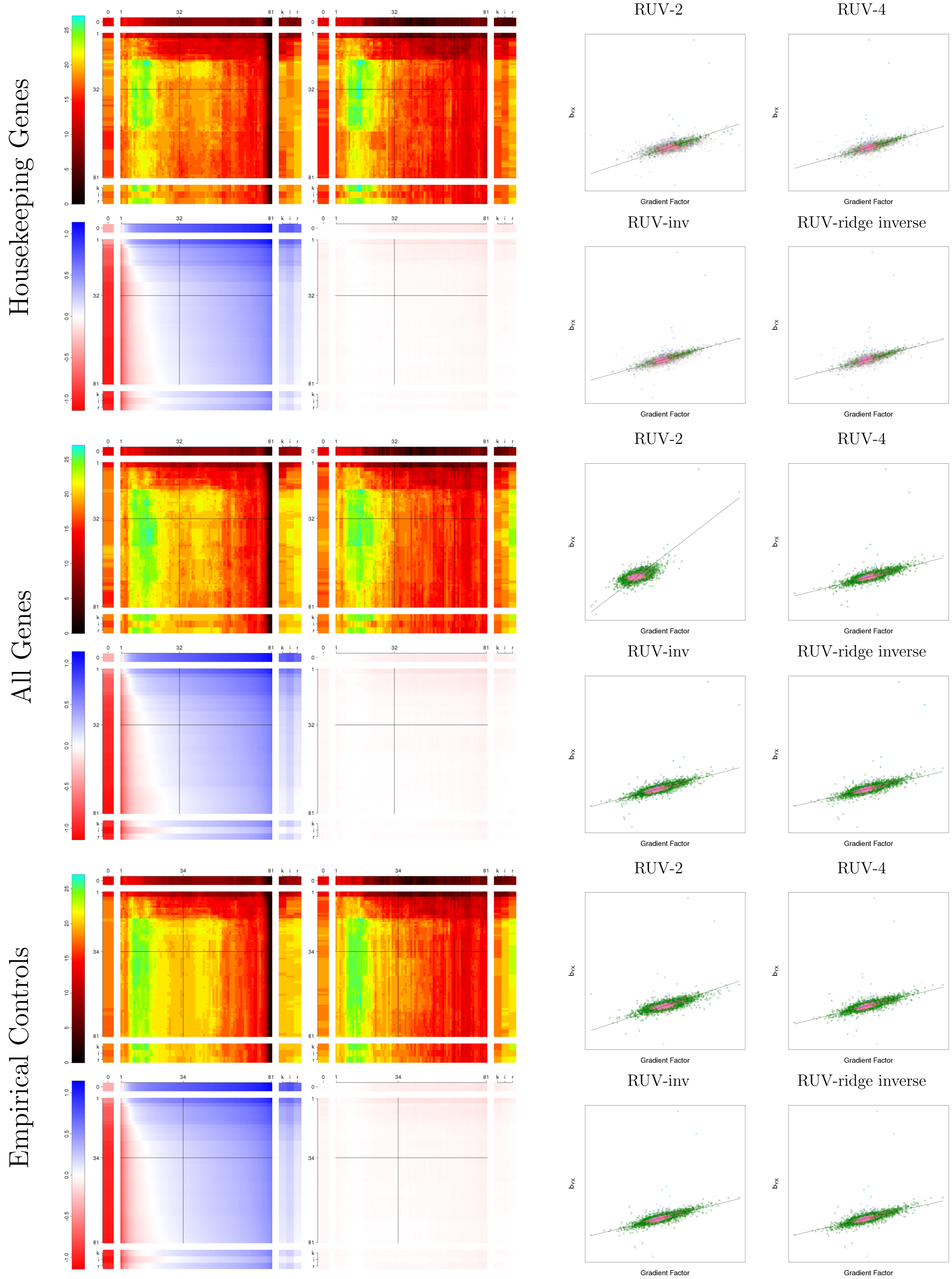


Figure B.16: Gender (Preprocessed). X/Y gene counts are out of the top 40 genes.

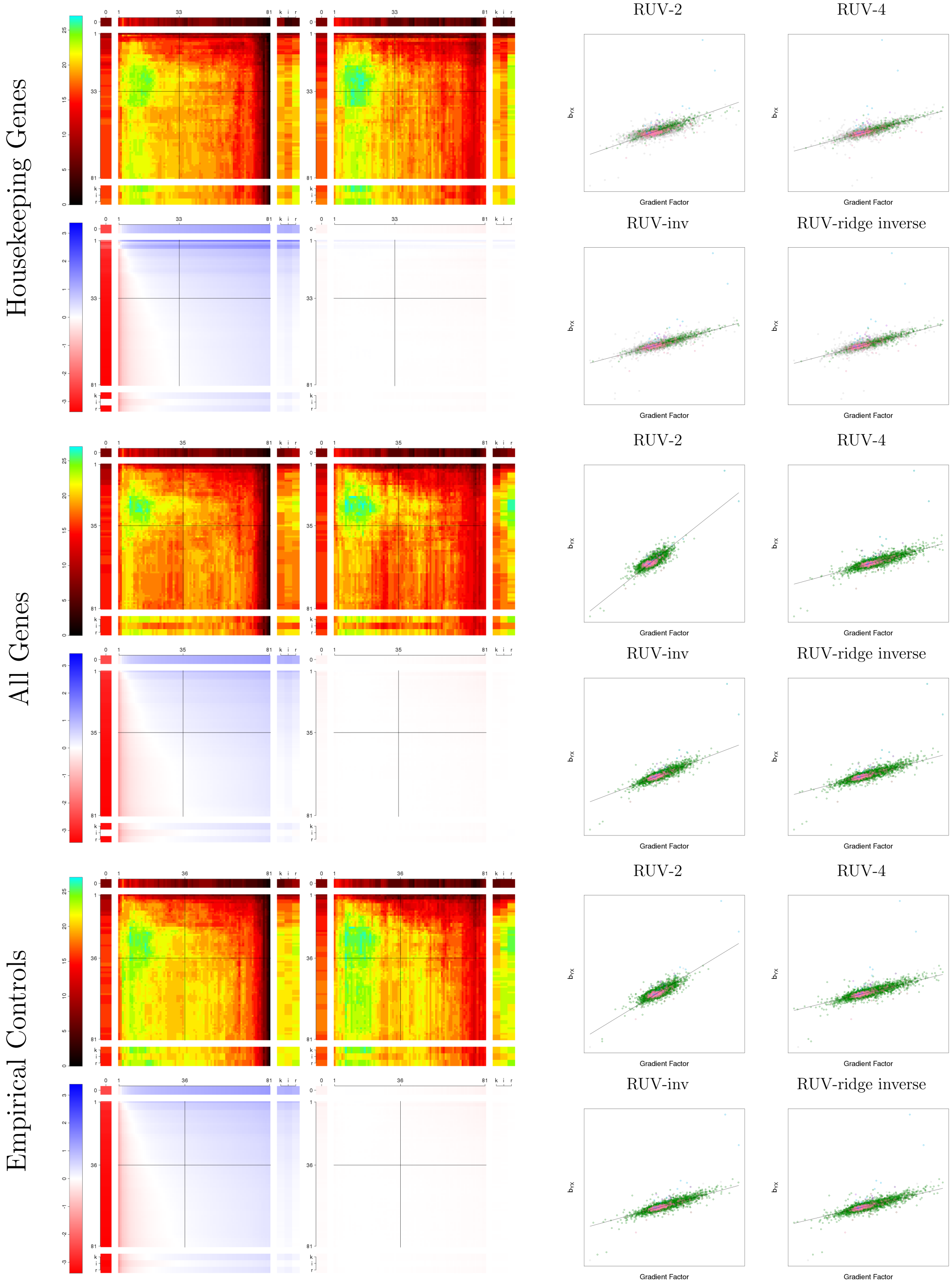


Figure B.17: Gender (Not Preprocessed). X/Y gene counts are out of the top 40 genes.

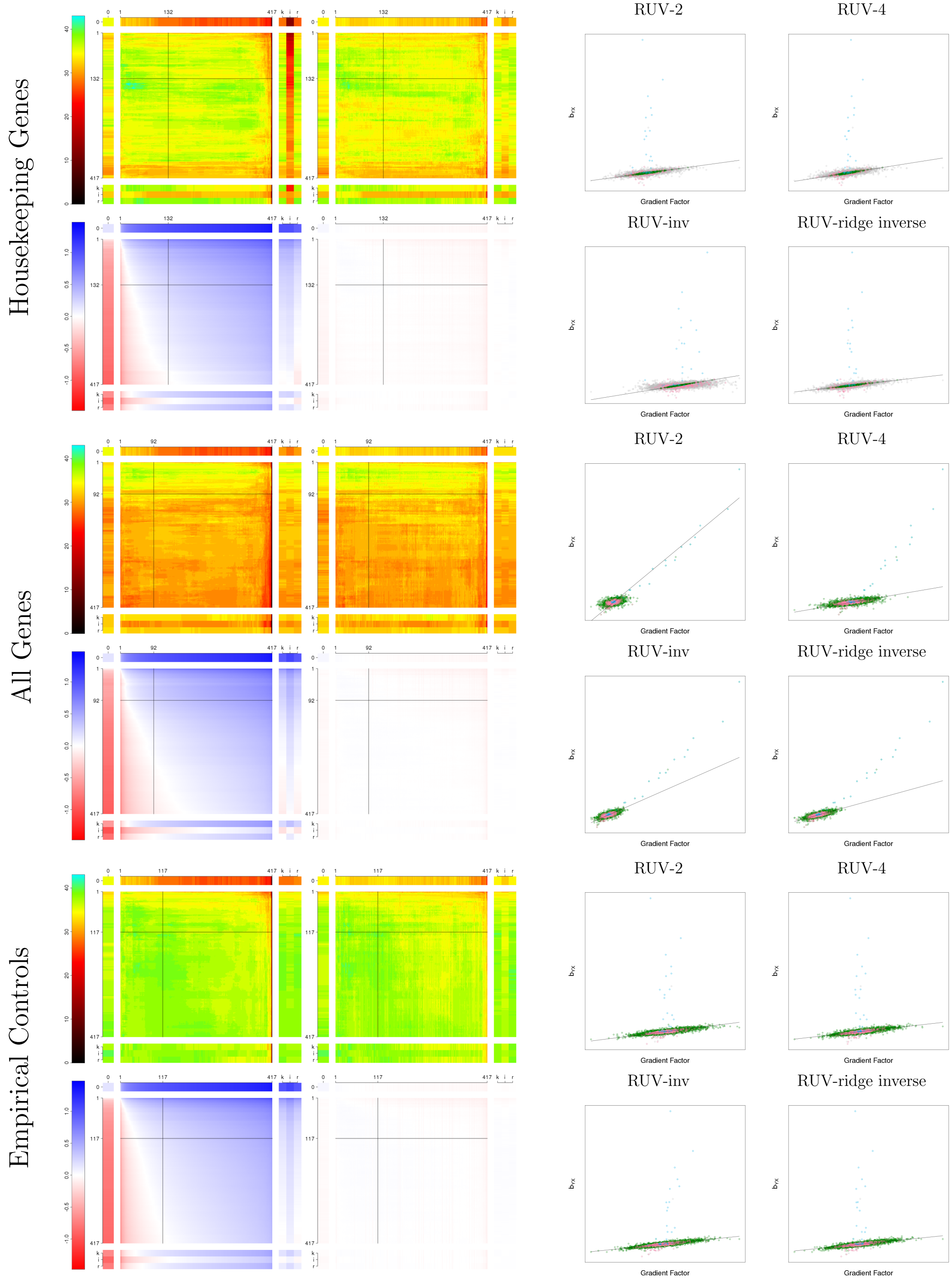


Figure B.18: TCGA Exon. X/Y gene counts are out of the top 80 genes.

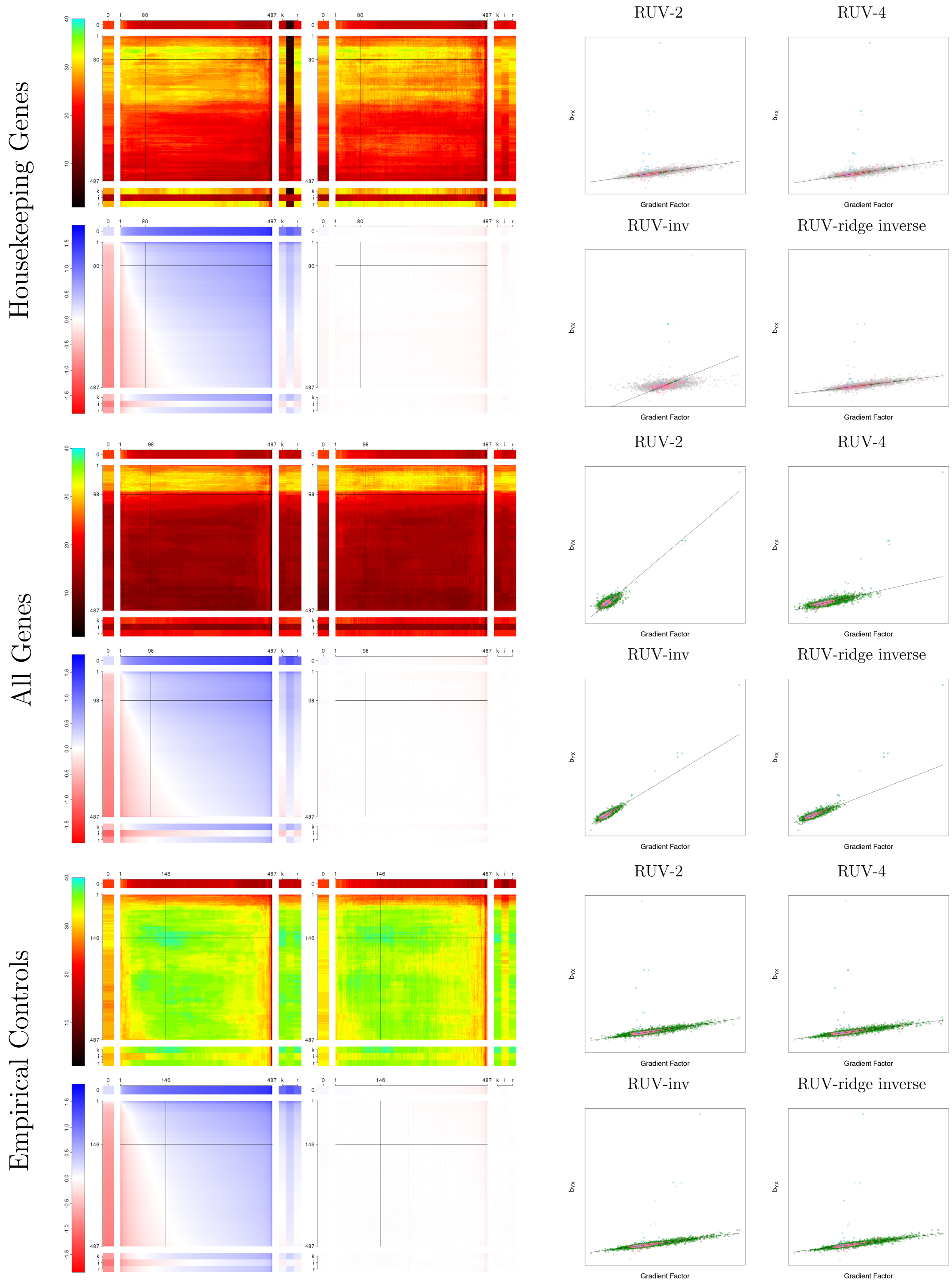


Figure B.19: TCGA HG-U133A. X/Y gene counts are out of the top 80 genes.

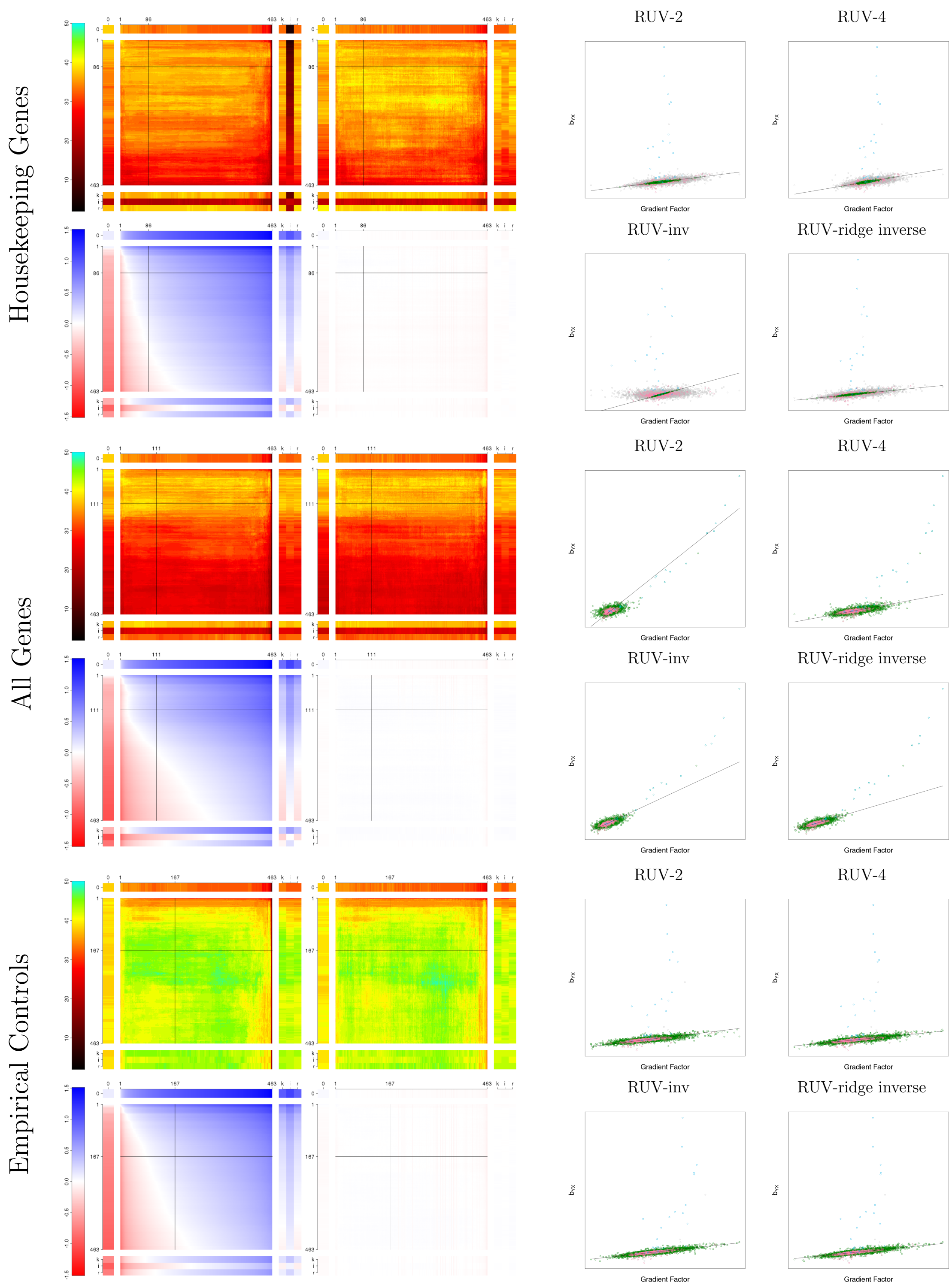


Figure B.20: TCGA Agilent. X/Y gene counts are out of the top 80 genes.

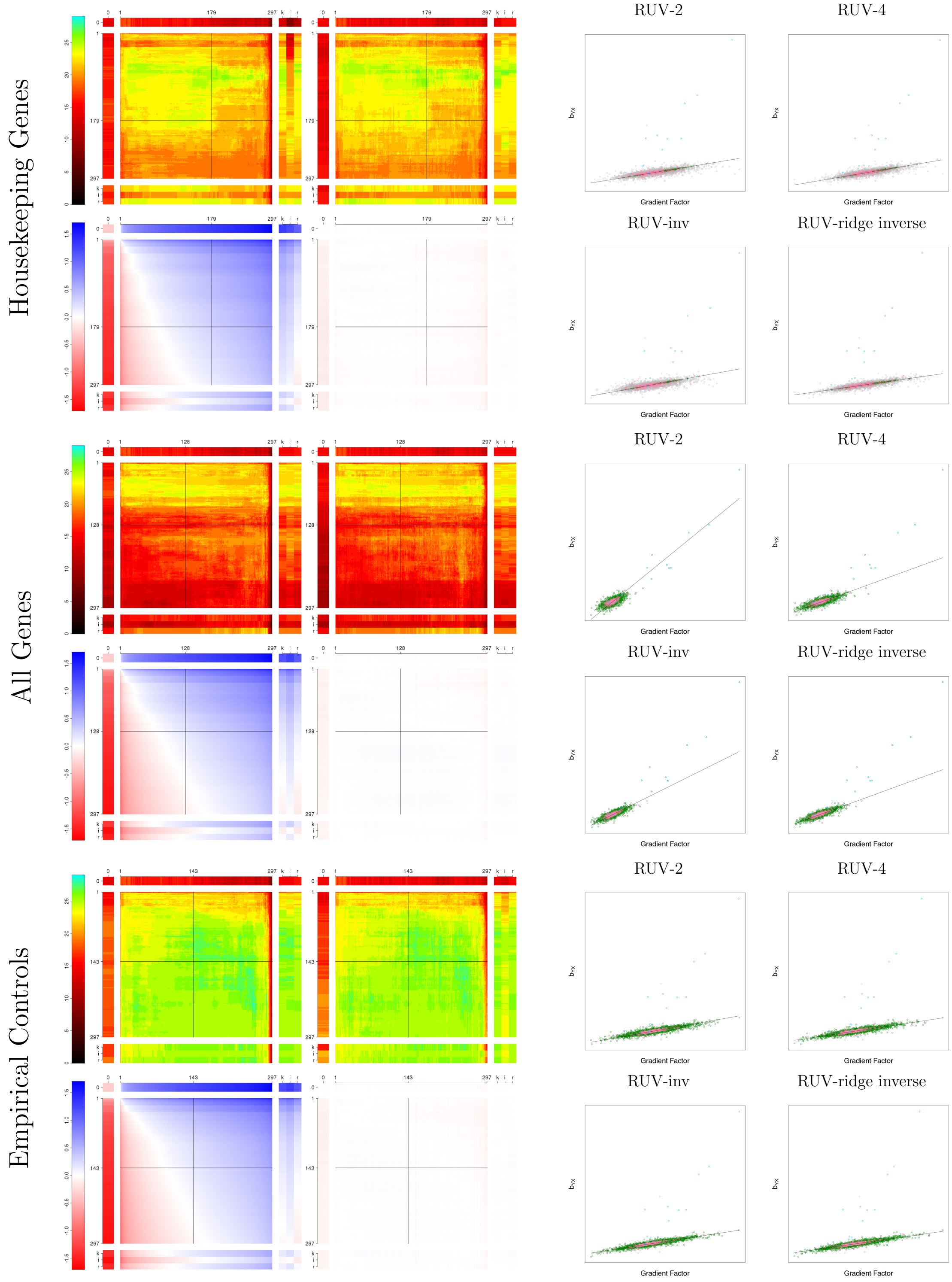


Figure B.21: TCGA Combined Subsets. X/Y gene counts are out of the top 40 genes.

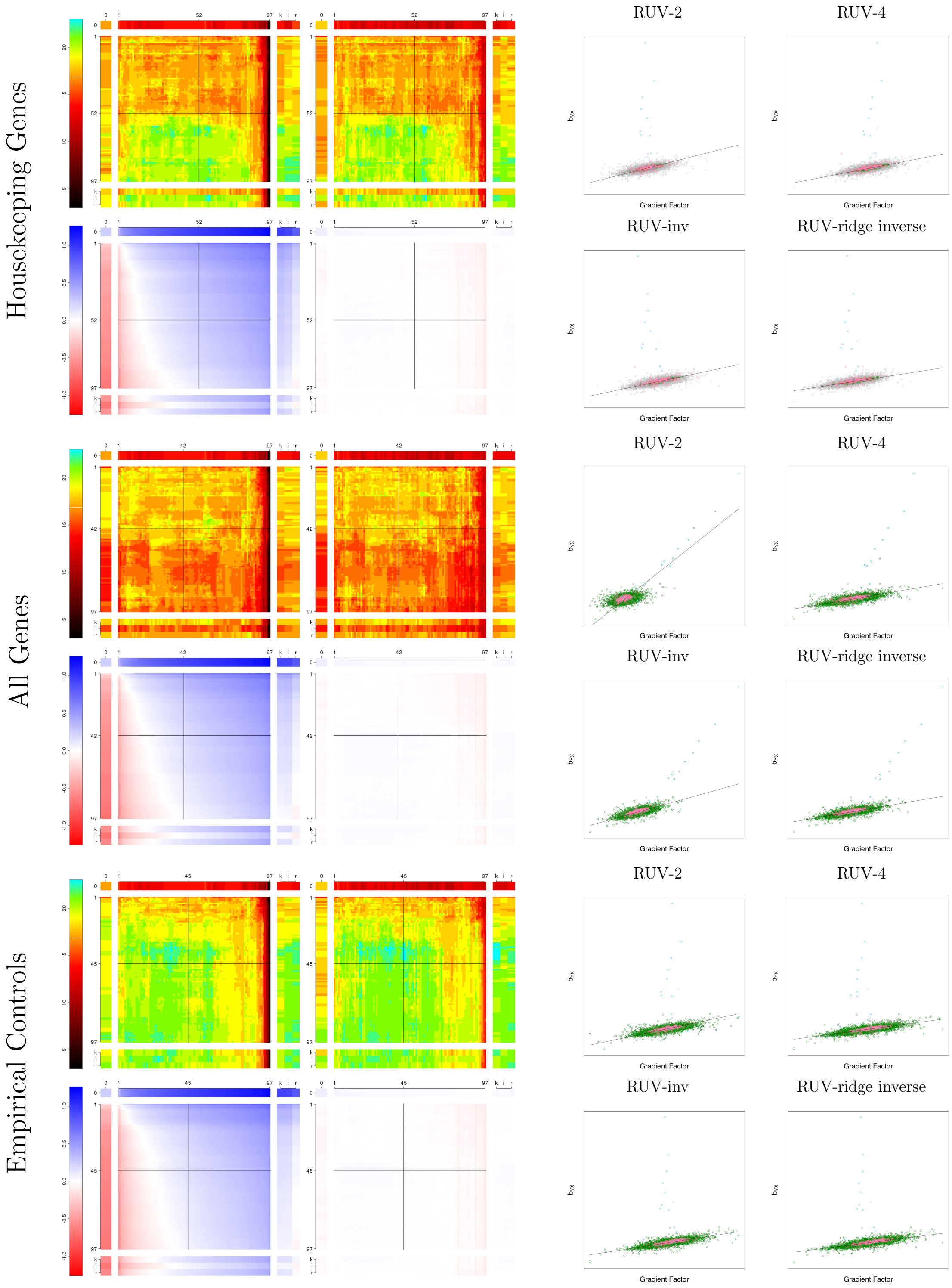


Figure B.22: TCGA Exon (subset). X/Y gene counts are out of the top 40 genes.

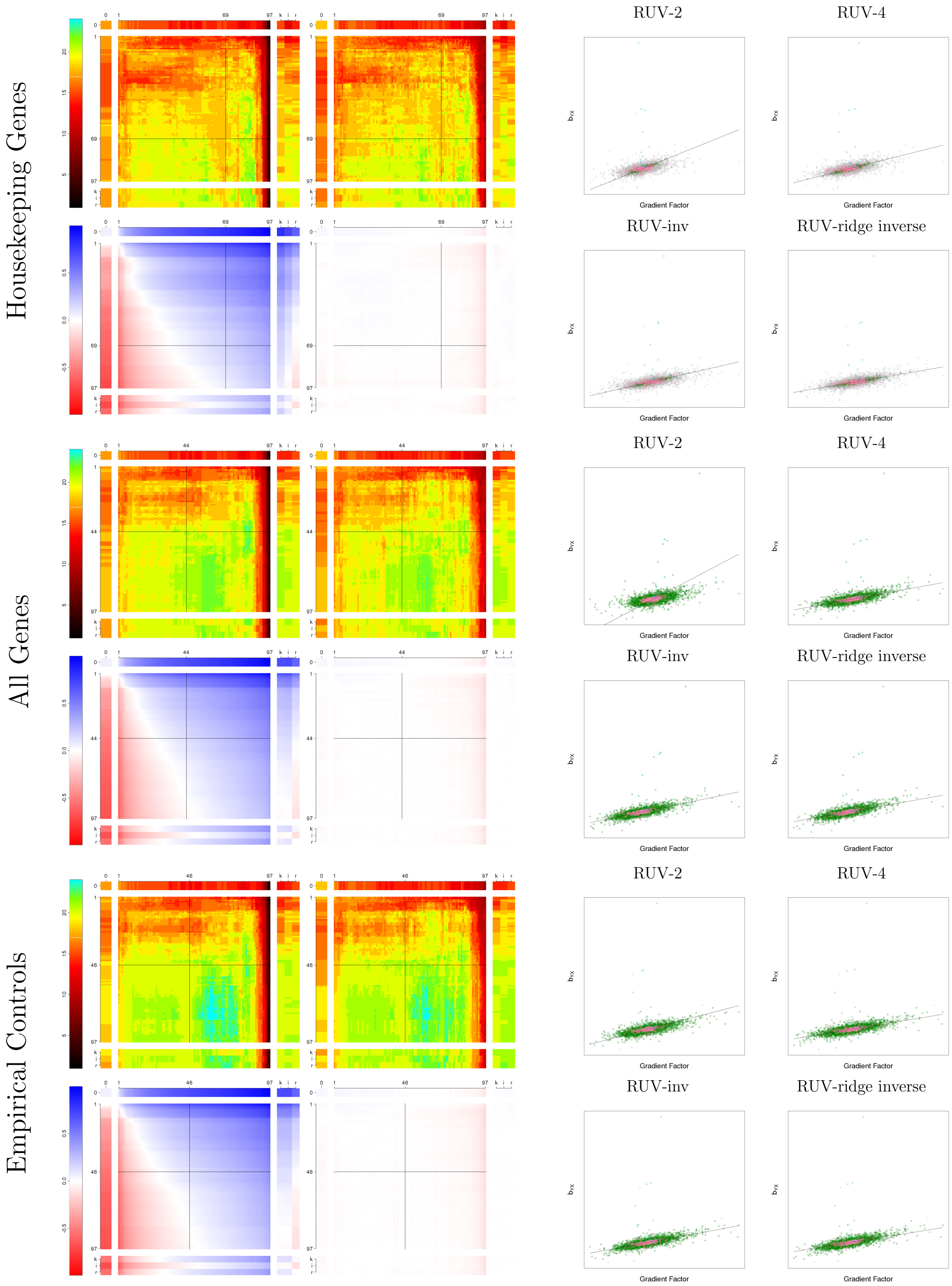


Figure B.23: TCGA HG-133a (subset). X/Y gene counts are out of the top 40 genes.

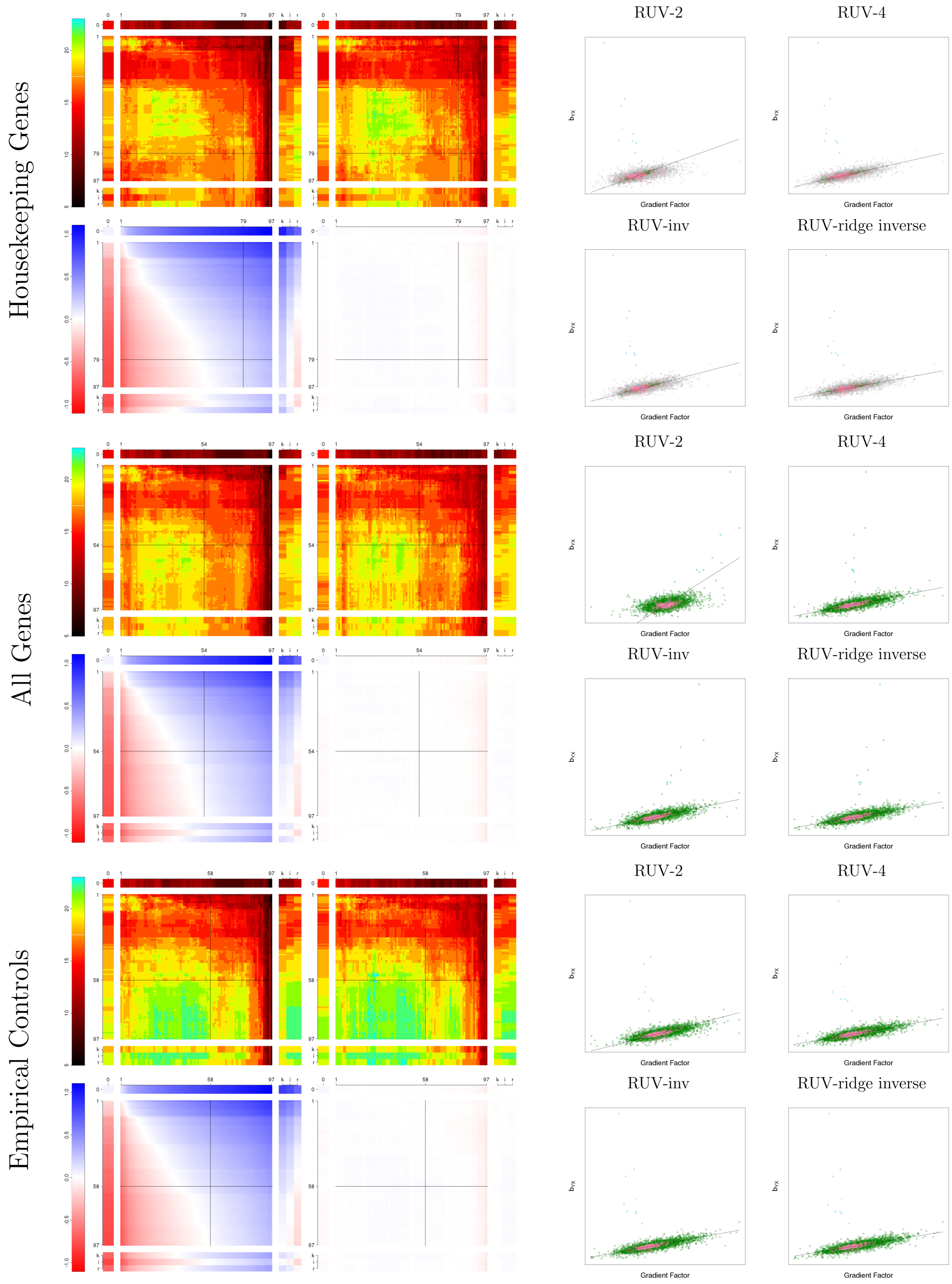


Figure B.24: TCGA Agilent (subset). X/Y gene counts are out of the top 40 genes.

B.4 Data Results (Tables)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	15	19	23	25	26	0.02
SVA-IRW	18	19	21	24	24	0.04
SVA-TS	16	18	19	19	21	0.09
LEAPP	16	24	24	26	27	0.13
ICE	20	27	29	31	31	0.04
RUV-4 (HK)	18	24	27	29	31	0.1
RUV-inv (HK)	19	26	29	31	33	0.05
RUV-rinv (HK)	20	26	30	32	33	0.05
RUV-4-evar (HK)	19	24	29	30	31	0.06
RUV-inv-evar (HK)	19	26	29	30	35	0.06
RUV-rinv-evar (HK)	19	27	30	30	33	0.06
RUV-4 (Full)	19	23	28	30	30	0.1
RUV-inv (Full)	20	25	29	31	34	0.05
RUV-rinv (Full)	20	26	29	32	35	0.05
RUV-4-evar (Full)	19	25	28	29	31	0.06
RUV-inv-evar (Full)	20	26	29	30	34	0.06
RUV-rinv-evar (Full)	20	26	29	31	32	0.05
RUV-4 (Empi)	19	24	29	30	32	0.1
RUV-inv (Empi)	20	25	29	31	33	0.05
RUV-rinv (Empi)	20	26	29	33	35	0.05
RUV-4-evar (Empi)	20	26	28	29	31	0.06
RUV-inv-evar (Empi)	20	26	29	31	33	0.06
RUV-rinv-evar (Empi)	20	26	29	32	34	0.06

Table B.25: Alzheimer's (Preprocessed)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	8	9	9	13	15	0.3
SVA-IRW	8	9	12	13	14	0.26
SVA-TS	0	0	0	0	0	NA
LEAPP	17	23	24	26	26	0.13
ICE	13	16	17	17	21	0.23
RUV-4 (HK)	14	19	23	24	26	0.1
RUV-inv (HK)	17	21	22	24	28	0.05
RUV-rinv (HK)	18	21	26	28	31	0.05
RUV-4-evar (HK)	17	22	25	27	29	0.06
RUV-inv-evar (HK)	20	23	26	28	30	0.06
RUV-rinv-evar (HK)	19	26	30	32	32	0.06
RUV-4 (Full)	16	21	23	26	26	0.1
RUV-inv (Full)	17	23	25	25	27	0.05
RUV-rinv (Full)	18	23	26	28	30	0.05
RUV-4-evar (Full)	19	23	26	28	29	0.06
RUV-inv-evar (Full)	19	23	25	29	29	0.06
RUV-rinv-evar (Full)	19	24	28	28	30	0.06
RUV-4 (Empi)	16	21	23	26	26	0.1
RUV-inv (Empi)	17	23	25	25	27	0.05
RUV-rinv (Empi)	18	23	27	28	30	0.05
RUV-4-evar (Empi)	19	23	25	28	29	0.06
RUV-inv-evar (Empi)	19	23	26	28	30	0.06
RUV-rinv-evar (Empi)	18	25	28	28	30	0.06

Table B.26: Alzheimer's (No Preprocessing)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	11	13	15	17	19	0.01
SVA-IRW	14	17	19	19	19	0.08
SVA-TS	16	21	23	25	27	0.09
LEAPP	18	20	22	25	26	0.12
ICE	16	23	26	27	28	0.04
RUV-4 (HK)	14	19	21	24	28	0.1
RUV-inv (HK)	13	18	20	21	22	0.08
RUV-rinv (HK)	16	20	22	25	28	0.08
RUV-4-evar (HK)	14	17	22	25	27	0.06
RUV-inv-evar (HK)	12	17	20	22	23	0.06
RUV-rinv-evar (HK)	15	21	23	27	28	0.06
RUV-4 (Full)	13	20	23	24	27	0.11
RUV-inv (Full)	14	20	21	23	27	0.07
RUV-rinv (Full)	15	21	25	27	28	0.08
RUV-4-evar (Full)	13	18	23	26	27	0.06
RUV-inv-evar (Full)	14	19	23	24	25	0.06
RUV-rinv-evar (Full)	16	22	26	27	27	0.06
RUV-4 (Empi)	13	20	23	24	26	0.11
RUV-inv (Empi)	15	19	24	25	27	0.08
RUV-rinv (Empi)	16	22	25	28	28	0.09
RUV-4-evar (Empi)	14	17	22	23	25	0.06
RUV-inv-evar (Empi)	14	20	22	25	27	0.06
RUV-rinv-evar (Empi)	15	22	26	27	28	0.06

Table B.27: Gender (Preprocessed)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	7	7	7	8	10	0
SVA-IRW	4	6	8	9	12	0
SVA-TS	0	0	0	0	0	NA
LEAPP	11	16	18	19	19	0.01
ICE	8	11	13	14	17	0
RUV-4 (HK)	13	20	22	26	29	0.12
RUV-inv (HK)	14	18	23	23	26	0.07
RUV-rinv (HK)	14	22	24	25	28	0.08
RUV-4-evar (HK)	12	20	25	26	27	0.06
RUV-inv-evar (HK)	12	19	22	24	26	0.06
RUV-rinv-evar (HK)	16	23	25	28	28	0.06
RUV-4 (Full)	12	20	23	24	29	0.12
RUV-inv (Full)	12	17	23	25	25	0.06
RUV-rinv (Full)	14	21	24	27	30	0.07
RUV-4-evar (Full)	14	18	24	28	28	0.06
RUV-inv-evar (Full)	13	18	19	22	24	0.06
RUV-rinv-evar (Full)	16	23	27	31	32	0.06
RUV-4 (Empi)	12	21	24	25	29	0.12
RUV-inv (Empi)	14	21	23	25	26	0.08
RUV-rinv (Empi)	14	23	24	27	30	0.09
RUV-4-evar (Empi)	14	20	24	28	30	0.06
RUV-inv-evar (Empi)	16	20	23	25	28	0.06
RUV-rinv-evar (Empi)	17	24	26	30	30	0.06

Table B.28: Gender (No Preprocessing)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	17	30	33	35	35	0.08
SVA-IRW	17	31	32	33	34	0.12
SVA-TS	17	33	34	37	40	0.08
LEAPP	17	33	34	35	36	0.12
ICE	17	33	35	35	36	0.01
RUV-4 (HK)	17	33	34	38	39	0.13
RUV-inv (HK)	17	28	29	30	30	0.05
RUV-rinv (HK)	17	34	37	39	40	0.07
RUV-4-evar (HK)	17	33	35	38	38	0.06
RUV-inv-evar (HK)	17	27	30	30	32	0.05
RUV-rinv-evar (HK)	17	33	37	39	40	0.05
RUV-4 (Full)	17	31	32	33	34	0.13
RUV-inv (Full)	17	26	29	29	30	0.04
RUV-rinv (Full)	17	30	31	31	32	0.06
RUV-4-evar (Full)	17	30	31	33	34	0.05
RUV-inv-evar (Full)	17	25	29	29	30	0.05
RUV-rinv-evar (Full)	17	30	31	31	32	0.05
RUV-4 (Empi)	17	33	35	38	40	0.1
RUV-inv (Empi)	17	33	38	38	38	0.07
RUV-rinv (Empi)	17	33	36	37	40	0.08
RUV-4-evar (Empi)	17	33	35	38	40	0.05
RUV-inv-evar (Empi)	17	33	36	37	38	0.05
RUV-rinv-evar (Empi)	17	33	36	39	41	0.05

Table B.29: TCGA (Exon)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	16	22	22	24	25	0.12
SVA-IRW	17	24	25	26	26	0.04
SVA-TS	17	27	31	31	32	0.08
LEAPP	17	29	32	32	34	0.11
ICE	17	28	31	32	32	0.02
RUV-4 (HK)	17	24	26	29	32	0.14
RUV-inv (HK)	10	13	16	16	18	0.03
RUV-rinv (HK)	17	29	32	32	32	0.08
RUV-4-evar (HK)	17	23	25	27	31	0.06
RUV-inv-evar (HK)	9	12	15	18	19	0.06
RUV-rinv-evar (HK)	16	29	31	32	33	0.05
RUV-4 (Full)	15	17	20	23	24	0.15
RUV-inv (Full)	9	10	12	14	15	0.04
RUV-rinv (Full)	13	16	19	22	23	0.08
RUV-4-evar (Full)	15	17	20	23	24	0.06
RUV-inv-evar (Full)	8	10	11	14	14	0.05
RUV-rinv-evar (Full)	13	16	20	23	25	0.06
RUV-4 (Empi)	17	29	35	38	38	0.1
RUV-inv (Empi)	17	27	31	35	36	0.06
RUV-rinv (Empi)	17	28	34	36	36	0.08
RUV-4-evar (Empi)	17	29	33	38	39	0.05
RUV-inv-evar (Empi)	17	24	29	32	32	0.05
RUV-rinv-evar (Empi)	17	29	33	36	36	0.06

Table B.30: TCGA (HG U133A)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	17	30	36	38	40	0.07
SVA-IRW	17	33	37	41	42	0.08
SVA-TS	17	33	37	38	42	0.08
LEAPP	17	33	37	38	43	0.12
ICE	17	33	34	37	41	0.01
RUV-4 (HK)	17	33	34	37	40	0.13
RUV-inv (HK)	16	19	22	22	23	0.04
RUV-rinv (HK)	17	33	38	39	41	0.08
RUV-4-evar (HK)	16	33	35	37	40	0.05
RUV-inv-evar (HK)	16	18	20	22	24	0.05
RUV-rinv-evar (HK)	17	33	37	40	42	0.05
RUV-4 (Full)	17	32	36	38	39	0.15
RUV-inv (Full)	17	22	23	25	26	0.05
RUV-rinv (Full)	17	29	31	33	34	0.05
RUV-4-evar (Full)	17	32	35	38	38	0.05
RUV-inv-evar (Full)	16	22	23	24	24	0.05
RUV-rinv-evar (Full)	17	29	29	32	34	0.05
RUV-4 (Empi)	17	33	40	44	46	0.11
RUV-inv (Empi)	17	33	38	43	43	0.07
RUV-rinv (Empi)	17	33	39	42	46	0.07
RUV-4-evar (Empi)	17	33	41	42	46	0.05
RUV-inv-evar (Empi)	18	32	37	43	45	0.05
RUV-rinv-evar (Empi)	17	33	39	43	45	0.05

Table B.31: TCGA (Agilent)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	12	14	16	17	17	0.02
SVA-IRW	17	21	25	25	26	0.07
SVA-TS	17	22	24	26	26	0.09
LEAPP	17	22	23	23	25	0.1
ICE	17	24	25	27	27	0.01
RUV-4 (HK)	17	22	24	25	25	0.16
RUV-inv (HK)	17	20	20	21	21	0.05
RUV-rinv (HK)	17	24	27	28	28	0.06
RUV-4-evar (HK)	17	23	25	25	26	0.06
RUV-inv-evar (HK)	17	20	20	20	21	0.05
RUV-rinv-evar (HK)	17	24	26	29	30	0.05
RUV-4 (Full)	13	16	18	22	23	0.11
RUV-inv (Full)	10	14	18	20	20	0.05
RUV-rinv (Full)	16	19	20	24	25	0.05
RUV-4-evar (Full)	13	16	18	20	21	0.05
RUV-inv-evar (Full)	10	13	16	20	20	0.05
RUV-rinv-evar (Full)	16	19	22	23	25	0.05
RUV-4 (Empi)	17	26	28	29	29	0.09
RUV-inv (Empi)	17	25	25	29	29	0.05
RUV-rinv (Empi)	17	25	27	28	31	0.06
RUV-4-evar (Empi)	17	26	27	29	29	0.05
RUV-inv-evar (Empi)	17	23	25	27	29	0.05
RUV-rinv-evar (Empi)	17	25	27	29	31	0.05

Table B.32: TCGA (Combined)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	13	17	18	18	18	0.08
SVA-IRW	15	18	19	24	25	0.05
SVA-TS	16	18	19	19	21	0.06
LEAPP	16	20	24	25	26	0.12
ICE	17	22	22	24	24	0.03
RUV-4 (HK)	15	18	21	24	25	0.13
RUV-inv (HK)	15	22	22	23	25	0.06
RUV-rinv (HK)	17	21	22	23	24	0.07
RUV-4-evar (HK)	15	18	20	23	24	0.05
RUV-inv-evar (HK)	15	21	22	24	25	0.05
RUV-rinv-evar (HK)	16	21	23	24	24	0.05
RUV-4 (Full)	15	18	20	22	23	0.09
RUV-inv (Full)	14	16	18	20	23	0.05
RUV-rinv (Full)	15	17	20	23	23	0.06
RUV-4-evar (Full)	14	18	20	21	22	0.05
RUV-inv-evar (Full)	13	16	18	22	23	0.05
RUV-rinv-evar (Full)	15	18	20	22	23	0.05
RUV-4 (Empi)	16	20	23	24	25	0.09
RUV-inv (Empi)	17	22	23	24	24	0.06
RUV-rinv (Empi)	16	22	22	23	25	0.06
RUV-4-evar (Empi)	16	22	23	24	25	0.05
RUV-inv-evar (Empi)	16	21	24	24	24	0.05
RUV-rinv-evar (Empi)	16	22	22	23	23	0.05

Table B.33: TCGA (Exon Subset)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	15	17	18	19	19	0.05
SVA-IRW	14	16	16	19	20	0.07
SVA-TS	14	19	20	21	22	0.06
LEAPP	15	18	19	22	22	0.12
ICE	17	21	21	22	22	0.03
RUV-4 (HK)	16	19	20	22	26	0.11
RUV-inv (HK)	16	19	19	21	22	0.04
RUV-rinv (HK)	16	19	21	23	23	0.05
RUV-4-evar (HK)	16	19	21	26	27	0.06
RUV-inv-evar (HK)	16	19	20	21	24	0.06
RUV-rinv-evar (HK)	16	19	21	22	23	0.06
RUV-4 (Full)	16	20	22	22	22	0.09
RUV-inv (Full)	15	20	22	22	23	0.05
RUV-rinv (Full)	15	20	20	21	23	0.06
RUV-4-evar (Full)	16	21	22	22	22	0.06
RUV-inv-evar (Full)	15	20	22	22	24	0.06
RUV-rinv-evar (Full)	16	20	21	22	23	0.06
RUV-4 (Empi)	16	20	21	22	23	0.09
RUV-inv (Empi)	16	20	22	22	25	0.05
RUV-rinv (Empi)	16	20	21	24	24	0.06
RUV-4-evar (Empi)	16	21	22	23	24	0.06
RUV-inv-evar (Empi)	16	21	23	24	27	0.06
RUV-rinv-evar (Empi)	16	21	23	23	23	0.06

Table B.34: TCGA (HG U133A Subset)

	Top 20	Top 40	Top 60	Top 80	Top 100	Type I
unadjusted	11	14	18	18	19	0.05
SVA-IRW	13	16	16	17	18	0.06
SVA-TS	14	17	20	21	22	0.05
LEAPP	13	16	18	19	21	0.1
ICE	17	22	22	23	23	0.03
RUV-4 (HK)	15	17	19	19	19	0.11
RUV-inv (HK)	16	18	20	20	20	0.04
RUV-rinv (HK)	16	19	22	22	23	0.04
RUV-4-evar (HK)	15	18	19	21	22	0.06
RUV-inv-evar (HK)	16	18	20	20	20	0.05
RUV-rinv-evar (HK)	16	19	21	21	22	0.05
RUV-4 (Full)	14	17	21	22	22	0.09
RUV-inv (Full)	15	19	19	20	20	0.04
RUV-rinv (Full)	15	20	20	20	22	0.05
RUV-4-evar (Full)	13	19	21	22	22	0.06
RUV-inv-evar (Full)	14	19	19	20	20	0.06
RUV-rinv-evar (Full)	15	19	20	20	22	0.06
RUV-4 (Empi)	16	19	21	21	24	0.09
RUV-inv (Empi)	16	22	22	23	23	0.04
RUV-rinv (Empi)	16	21	23	23	23	0.04
RUV-4-evar (Empi)	15	20	21	22	23	0.06
RUV-inv-evar (Empi)	16	22	22	23	23	0.05
RUV-rinv-evar (Empi)	16	21	23	23	23	0.05

Table B.35: TCGA (Agilent Subset)