

Ingenuity Systems, Inc.

---



## Robust Unattended Microarray Analysis

Ingenuity Technical Report No. 2011-1

Richard L. Halpert

Sandeep Sanga

rhalpert@ingenuity.com

ssanga@ingenuity.com

March 25th, 2012

Document Revision History	
Date	Notes
March 15, 2011	Original Version: Affymetrix pipeline description, performance testing.
March 25, 2012	QC metric details, Illumina and Agilent pipeline descriptions, degenerate cases.

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Design</b>	<b>2</b>
2.1	Normalization . . . . .	3
2.1.1	Affymetrix Normalization . . . . .	3
2.1.2	Illumina Normalization . . . . .	3
2.1.3	Agilent Normalization . . . . .	3
2.2	Quality Control . . . . .	4
2.3	Exploratory Analysis . . . . .	4
2.4	Batch Effect Correction . . . . .	4
2.5	Differential Expression Analysis . . . . .	4
2.6	Determination of Statistical Significance . . . . .	5
2.7	Degenerate Cases . . . . .	5
2.8	Outputs . . . . .	6
<b>3</b>	<b>Quality Control</b>	<b>6</b>
3.1	Signal Density and Box Plots . . . . .	6
3.2	Array Similarity Heatmap . . . . .	8
3.3	Variance vs. Mean Plot . . . . .	8
3.4	MA Plot (Ratio vs. Intensity) . . . . .	10
3.5	Affymetrix-Specific Plots . . . . .	10
3.5.1	RNA Degradation Plot . . . . .	10
3.5.2	Relative Log Expression Plot . . . . .	10
3.5.3	Normalized Unscaled Standard Error Plot . . . . .	10
3.5.4	PM/MM Plot . . . . .	13
3.6	PCA and PCA-3D Plots . . . . .	13
3.7	PCA Plot Separation Analysis . . . . .	15
3.8	Experimental Effect Size Estimation (PC1 vs. PC2 Quality Plot) . . . . .	15
3.9	Batch Detection and Batch Effect Size Estimation (PC1 vs. PC2 Batch Effect Analysis Plot) . . . . .	15
3.10	Quality Recommendations . . . . .	18
<b>4</b>	<b>Experimental Validation</b>	<b>18</b>
4.1	Datasets for Tuning . . . . .	19

4.2	Curation of Gene Lists . . . . .	20
4.3	Evaluation Techniques . . . . .	20
4.4	Initial Results . . . . .	21
4.5	Determination of Cutoffs . . . . .	21
4.6	Final Results . . . . .	23
<b>5</b>	<b>Conclusion</b>	<b>24</b>

## List of Figures

1	Pipeline Overview . . . . .	2
2	Examples of Density Plots and Boxplots, with and without outliers. . . . .	7
3	Examples of Between-Sample Heatmaps, with and without outliers. . . . .	8
4	Examples of Variance vs. Mean Plots, with and without flaws. . . . .	9
5	Examples of MA Plots, with and without flaws. . . . .	11
6	Examples of RNA Degradation Plots, with and without flaws. . . . .	12
7	Examples of Relative Log Expression Plots, with and without outliers. . . . .	12
8	Examples of Normalized Unscaled Standard Error Plots, with and without outliers. . . . .	13
9	Example of a PM/MM Plot, without flaws. . . . .	14
10	Examples of PCA Plots, with and without flaws. . . . .	16
11	Examples of PCA-3D Plots, with and without flaws. . . . .	17
12	Examples of PC1 vs. PC2 Quality Plots, with and without warnings. . . . .	18
13	Examples of PC1 vs. PC2 Batch Effect Analysis Plots, with and without recommendations. . . . .	19
14	Benefit of switching to lenient criteria. . . . .	22
15	Benefit of relaxing fold change cutoff. . . . .	22

## List of Tables

1	Datasets for analysis and tuning of pipeline. . . . .	20
2	Sensitivity of our pipeline as measured against published lists of differentially expressed genes. . . . .	21
3	Pipeline results on original 22 datasets using automated preferences for statistical significance. . . . .	23
4	Pipeline validation on an independent set of 10 datasets. . . . .	24

## Abstract

In order to facilitate faster biological insight in gene expression experiments, we assemble a robust fully automated statistical analysis pipeline for microarray data. We base our pipeline on industry-standard open source components, primarily the widely used Bioconductor software package for R. We employ quantile normalization, platform-appropriate summarization and background correction, empirical Bayes methods for batch effect correction (ComBat), and empirical Bayes linear models for statistical analysis (Limma) in order to maximize the pipeline’s power to detect differentially expressed genes for a wide variety of experimental designs. We control type-I error using Benjamini and Hochberg’s False Discovery Rate when sufficient experimental replicates are available. We use automated quality controls to identify outlier arrays and eliminate those that are likely flawed. We also use a novel PCA plot separation analysis to determine when batch effects should be corrected, and alert users when their experiments contain too much noise. Finally, we collect all necessary sample meta-data up-front, enabling completely unattended analysis for most datasets. Our pipeline supports human, mouse, and rat Affymetrix® 3’ IVT and Whole-Transcript (GeneChip®, GeneTitan®, GeneAtlas™, and PrimeView™) arrays, Illumina® Whole-Genome Gene Expression BeadChip and DASL® arrays, and Agilent® one-color and two-color Gene Expression Microarrays. Additionally, our pipeline supports input of Cuffdiff RNA-Seq differential gene expression results, and can process arbitrary fold change results from other sources. This work primarily describes the supported microarray platforms.

We validate our pipeline experimentally by applying it to 28 peer reviewed publications with 32 associated publically available datasets for the Affymetrix platform. For each dataset, we curate the published list of genes. We then evaluate the pipeline’s performance by calculating the percentage of that list that our pipeline finds to be significantly differentially expressed. We are currently in the process of creating similar ”gold standard” dataset portfolios for Illumina and Agilent microarray platforms as well. We find that our pipeline performs very well against published gene lists from stringent analyses. We also find that relaxing our default significance cutoffs allows the pipeline to perform well against the remainder of published gene lists. These findings motivate the development of rules for the adjustment of significance and fold change cutoffs to best capture the biologically interesting part of the data in each dataset. These rules allow our pipeline to discover differential expression in a wide range of study types, including low-budget, pilot, and noisy studies.

## 1 Introduction

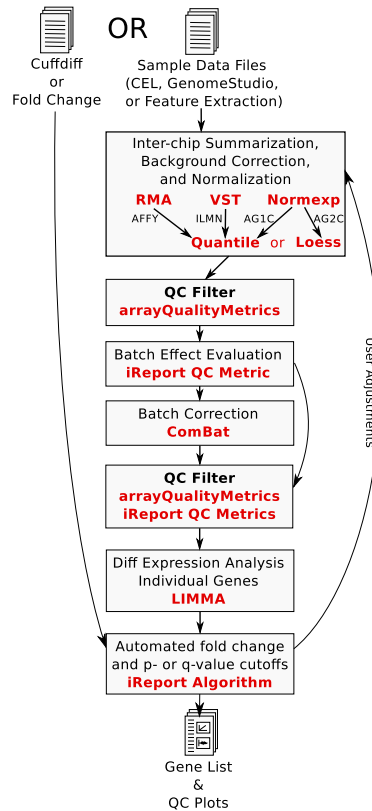
Microarray data analysis presents some unique challenges. Initially, to address these challenges, a large number of techniques were attempted. However, over time, a few robust, reliable techniques have become widely accepted [1]. With the availability of mature tools and well understood methods, there is now an opportunity to lower the barrier of entry to the use of microarrays and to ease the burden of microarray analysis on the user.

We aim to develop an ”80% solution”, a tool that can perform effective microarray data analysis for most studies without the oversight or intervention of an expert. To do this, we combine industry standard tools with new automated techniques to form a completely automated pipeline, described in Section 2. We enable the analysis to be unattended by providing automated quality controls (Section 3) and significance cutoffs (Section 4.6). We validate the effectiveness of the pipeline by comparing against published datasets (Section 4).

We fully expect to encounter datasets for which unattended analysis is untenable, due to quality control problems, unusual experimental design, or other exceptional situations. In Section 4.6, we discuss the percentage of our ”Gold Standard” test datasets that might fall into the category of *non-automatable* due to experimental design or statistical power problems.

## 2 Design

Our differential expression analysis pipeline is written in the R statistical programming language [23]. We make extensive use of the Bioconductor project [7], which is a widely used and frequently cited collection of tools for the analysis of high-throughput genomic data. The pipeline is summarized in Figure 1. Thanks to the extensive efforts of the Bioconductor project, we find this task to be primarily one of software development rather than statistical methods.



**Figure 1:** Pipeline Overview

The pipeline takes as input one or more data files containing raw probe-level expression measurements. These inputs may be Affymetrix CEL files, Illumina GenomeStudio report files, or Agilent Feature Extraction files. In all cases, the data are expected to be non-normalized, non-background-corrected data (though some of the formats contain processed data as well). The data files are accompanied by basic sample metadata specifying the experimental condition, batch, and block (for paired designs or technical replicates) for each file, and the comparison(s) to be analyzed.

The pipeline then performs normalization (Section 2.1), quality control (Sections 2.2 and 3), batch effect correction (Section 2.4), and differential expression analysis (Section 2.5). Finally, the pipeline makes a determination of statistical significance for each gene (Section 2.6), and produces textual and graphical outputs (Section 2.8).

## 2.1 Normalization

For each platform, an appropriate combination of robust, widely applicable background correction, summarization, and normalization techniques are used (described in the following subsections). After normalization, the data for all platforms are on a log<sub>2</sub> scale. For all one-color platforms, quantile normalization is used between arrays because of its applicability to datasets with virtually any within-array distribution, a characteristic which is nearly always stable from one array to another within a single experiment.

### 2.1.1 Affymetrix Normalization

We utilize the Robust Multi-Chip Average (RMA) for normalization. RMA consists of three interrelated steps: a background adjustment, normalization, and summarization.

Irizarry *et al.* propose a background correction  $B(\cdot)$  [14] based on the expected signal given the perfect-match (PM) probe intensity. They assume exponential signal and normal noise, and they then propose that the background-adjusted, normalized, and log-transformed PM intensities follow a linear additive model. They choose quantile normalization [3], and they employ median polish [10] to estimate model parameters, yielding a log-scale expression measure that is robust to outliers for each probeset. Irizarry *et al.* demonstrate [13] that RMA performs better than MAS 5.0 [11] and Li-Wong [19] for detection of differentially expressed genes.

RMA has wide acceptance for differential expression analysis in the bioinformatics community, and is even largely preferred over the chip manufacturer's own normalization method, MAS 5.0 [5, 9]. Although Lim *et al.* find RMA sub-optimal for reverse engineering gene networks (versus MAS 5.0) [20], Giorgi *et al.* determine that RMA's shortcomings do not negatively impact differential expression analysis [8].

### 2.1.2 Illumina Normalization

Our pipeline employs Lin *et al.*'s Variance Stabilizing Transformation [21] (VST) as the first step in processing Illumina data. Lin *et al.* demonstrated that applying VST before quantile normalization results in increased power to detect differential expression. This is because the VST algorithm utilizes the large number of within-array replicates that are characteristic of the Illumina platform to produce more robust per-array expression estimates than either a log<sub>2</sub> transformation (with quantile normalization) or a Variance Stabilizing Normalization (Huber *et al.* [12], originally proposed by Rocke and Durbin [25]). Specifically, VST (with quantile normalization) does a better job of reducing the heteroskedasticity (non-uniformity of variance) of the dataset, which makes the data more precisely fit the assumptions of the statistical techniques that are later applied to detect differential expression. This results in more power to detect differential expression.

When the pipeline receives data that lacks within-array replicate and error information, it gracefully degrades by applying a log<sub>2</sub> transformation instead of VST.

### 2.1.3 Agilent Normalization

Agilent data is background corrected using a normal model for the noise and exponential model for the signal, with estimation done using the saddle-point approximation to maximum likelihood from Ritchie *et al.* [24] and improved by Silver, Ritchie, and Smyth [26]. This method, called the "normexp" method,

produces a smooth monotonic transformation that ensures all resulting intensities are positive, and therefore avoids discarding data before normalization.

After background correction, *two-color* Agilent data is normalized within arrays using Yang, Dudoit, Luu, and Speed's loess method, which computes a loess regression line through the MA plot (log ratio versus average intensity plot) of the array and uses it as an adjustment to the M values so that the regression line becomes the zero-line of the plot.

*One-color* Agilent data is quantile normalized and converted to a log<sub>2</sub> scale.

## 2.2 Quality Control

In a typical microarray analysis workflow, quality control is the most essential function that is normally performed manually. Our approach is to evaluate quality automatically wherever possible on a per-sample and experiment-wide basis, to alert the user and provide actionable explanations whenever there are signs of a possible quality problem, and to make all quality control information available for review by both the user who runs the statistical analysis and the end-user who receives the results. Our quality control approach is described in detail in Section 3.

## 2.3 Exploratory Analysis

Exploratory analysis is another function normally carried out manually for microarray experiments. Typically such analysis is used to determine the ideal type of normalization, summarization, scale, batch correction, and differential expression analysis to use for a given dataset. We automate part of this process: experimental effect size estimation determines whether the experimental variable's effect is sufficiently large compared to the experimental noise to detect differential expression (described in Section 3.8), and batch effect size estimation determines whether batches in the data contribute significantly to the experimental noise (described in Section 3.9). For other decisions normally optimized through exploratory analysis, we instead use a fixed set of broadly applicable techniques that prove to perform well for a wide variety of datasets (as shown in Section 4).

## 2.4 Batch Effect Correction

Recently, the importance of compensating for non-biological batch variation has been extensively documented [17, 22]. In 2007, Johnson *et al.* proposed using empirical Bayes methods to remove batch effects [16]. Their approach was preferred by Luo *et al.* [22] in the MAQC-II project in which they studied six datasets and eight types of batch effect. This technique works even for small numbers of replicates and does not require the use of a reference phenotype, which makes it appropriate for a wide range of experiments.

We use ComBat, a tool provided by Johnson *et al.* implementing their approach. ComBat requires only that each experimental group be represented by at least one array in each batch. If this requirement is not met, then the approach is abandoned, batch effects are left un-corrected, and the user is informed of the problem.

## 2.5 Differential Expression Analysis

We employ Limma for differential expression analysis because it is highly robust to varying experimental designs and data distributions. Limma shows improved power over other approaches for low sample sizes

due to the use of a moderated t-statistic which allows for "borrowing information from the ensemble of genes which can assist in inference about each gene individually" [27]. Limma is also tolerant of data from populations having both normal and some non-normal distributions [27], and supports both blocked and non-blocked designs.

In a study of a large number of statistical methods, Jeffery *et al.* find that the method implemented by Limma performs well for both large and small numbers of samples [15], and "proved to be much more reliable than other methods examined in this study," including Significance Analysis for Microarrays (SAM) [29], ANOVA, and seven other popular techniques. Additionally, Dondrup *et al.* find that Limma has the highest level of agreement with the other methods they tested, which include SAM, the t-test, VarMixt, RankProd, and others [6]. In light of these comparisons, we choose Limma for its robustness.

## 2.6 Determination of Statistical Significance

By default, genes are filtered for statistical significance using Benjamini and Hochberg's False Discovery Rate (FDR) [2], with an industry-standard FDR-adjusted p-value cutoff of 0.05 (hereafter written  $FDR \leq 0.05$ ). However, this and other cutoffs are adjusted depending on the distribution of differentially expressed genes. If the pipeline fails to find at least 0.1% of genes significantly differentially expressed, it falls back on the use of raw p-values to rank genes by likelihood of differential expression (with a cutoff of 0.05). This fall-back is designed to allow the pipeline to return a list of genes ranked by likelihood of differential expression when there isn't enough statistical power to control the false discovery rate. Then, if more than 2000 genes are found to be significantly differentially expressed, the raw or FDR-adjusted p-value cutoff is tightened to 0.01 or 0.001 in order to reduce the size of the dataset being sent on to downstream analyses. In all cases, we apply a fold change cutoff (by default 1.5) in order to leave between 0.1% of genes and 2000 genes. These bounds are motivated by our findings in Section 4.5, and the upper bound is optional and adjustable.

Note that Limma's moderated t-statistic is useful for ranking genes by likelihood of differential expression even though it does not produce absolute p-values [28]. While it is not considered best-practice to publish based on raw p-values, it is often practical to use the findings therein to direct further investigation. We suggest the use of additional microarrays, RT-PCR, northern blot, or other techniques to confirm the behavior observed in any experiment failing to control the False Discovery Rate. In fact, independent confirmation of microarray results is generally advisable for key findings in any experiment [4].

## 2.7 Degenerate Cases

Our pipeline supports several degenerate statistical cases, including the case with no biological or technical replicates. In this case, fold change is computed as the ratio of expression values between condition and control, and p-values are all set to 1. Other degenerate cases include when there are four or fewer total samples in the dataset, and one or two pairs of those samples are technical replicates. In these cases, it is not possible to include both the experimental variable and the technical replicates in a single linear model because there are too few degrees of freedom in the model to estimate every parameter. For this degenerate case, the consensus correlation coefficient between technical replicates is assumed to be very high (0.95), which is virtually always a conservative over-estimate of the true correlation.

## 2.8 Outputs

The output of our pipeline consists of a list of differentially expressed genes. Also included are the cutoffs selected by the pipeline, a table of the number of differentially expressed genes that would be found for different combinations of cutoffs, a record of the statistical methods that were applied during the analysis, a variety of quality control graphs, and a list of quality recommendations and warnings (if any). In the gene list, each differentially expressed gene is listed by probeset ID (for most platforms) or by Entrez Gene ID (for Affymetrix exon arrays, older Illumina arrays, and some other platforms), and is associated with a log<sub>2</sub> fold change, a raw p-value, and (when applicable) an FDR-adjusted p-value. We annotate our gene list with gene symbols and Entrez Gene IDs.

The behavior, input, and output of the pipeline are designed to allow completely unattended analysis. However, it is a common use case for users to make manual adjustments to the cutoffs to match other experiments, to ensure particular genes are included in the results, or simply to see how it changes the biological analysis. To support this usage, the pipeline includes in its output a complete list of genes/probesets. It is also occasionally necessary for users to act on quality control recommendations made by the pipeline. As such, there is the option to rerun the pipeline with specific cutoffs, with batch correction forced on or off, with files added or removed, or with various other options.

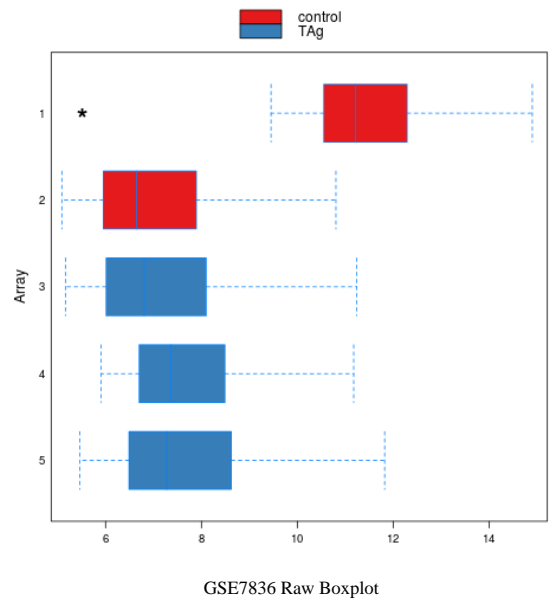
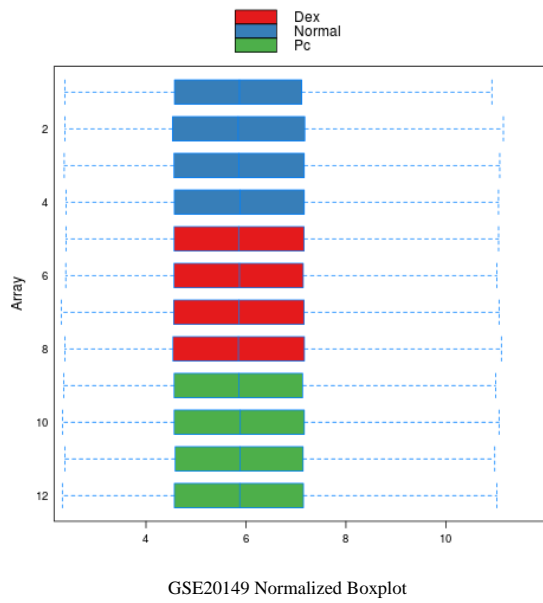
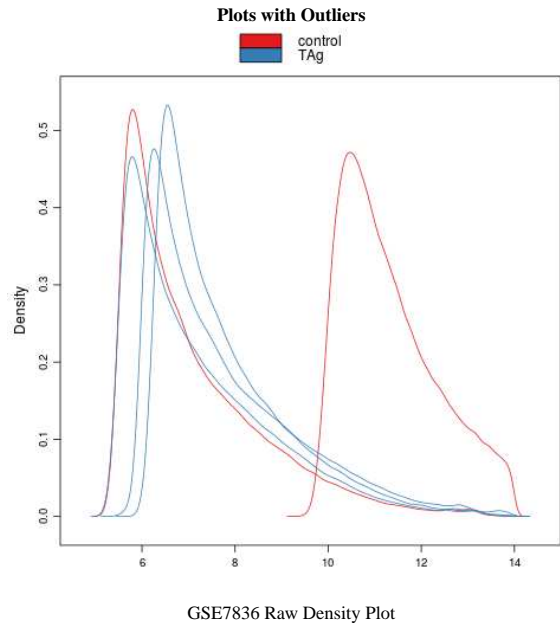
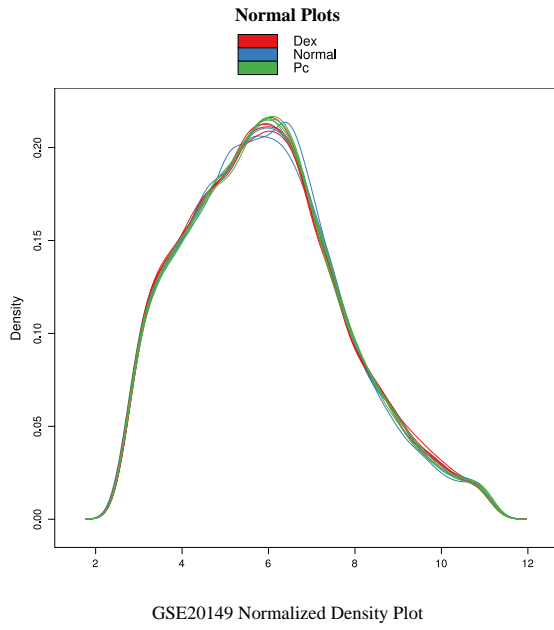
## 3 Quality Control

Quality Control (QC) is considered one of the most essential functions of the pipeline. Since the pipeline runs unattended, it is crucial that it be able to identify outliers, batch effects, overly noisy experiments, and experimental design problems automatically, and to advise the user of any recommendations or warnings.

We implement automated quality controls in our pipeline using the `arrayQualityMetrics` Bioconductor package [18]. This package computes a variety of QC plots, described in Sections 3.1 through 3.5. Additionally, the pipeline produces several PCA plots and uses them to estimate the experimental effect size and any batch effect size (if batch information is available). Finally, several basic experimental design requirements are checked.

### 3.1 Signal Density and Box Plots

Signal density plots show a smoothed histogram of the signal intensities in a dataset. In a typical dataset, the distributions of signal intensities on each array should be similar. Outlier detection is performed using a Kolmogorov-Smirnov test between each array's distribution and the distribution of the pooled data. Since this is a data-driven test, it varies in sensitivity depending on the level of variation in the dataset. In most cases, arrays flagged as outliers differ enough from the distributions of the other arrays for the differences to be visible on a box plot, so these outlier flags are generally associated with the signal box plot rather than the signal density plot. Both types of plot are generated for both raw and normalized data, and also for batch corrected data if batch correction is performed. Outliers may be flagged on the raw and either the normalized or batch corrected versions.



**Figure 2:** Examples of Density Plots and Boxplots, with and without outliers.

### 3.2 Array Similarity Heatmap

A heatmap is produced showing the relative distances between arrays. Manhattan distance is used. This plot is used for clustering analysis: if the arrays cluster by experimental condition, this is strong evidence that the experimental effect is much larger than any noise in the experiment. Outlier detection is performed by looking for arrays for which the sum of the distances to all other arrays is beyond the 95% confidence interval of the distribution of the sums for all arrays. This method works very well in practice: it is usually corroborated both visually and by the cluster plot. This plot is generated for both raw and normalized data, and also for batch corrected data when batch correction is performed.

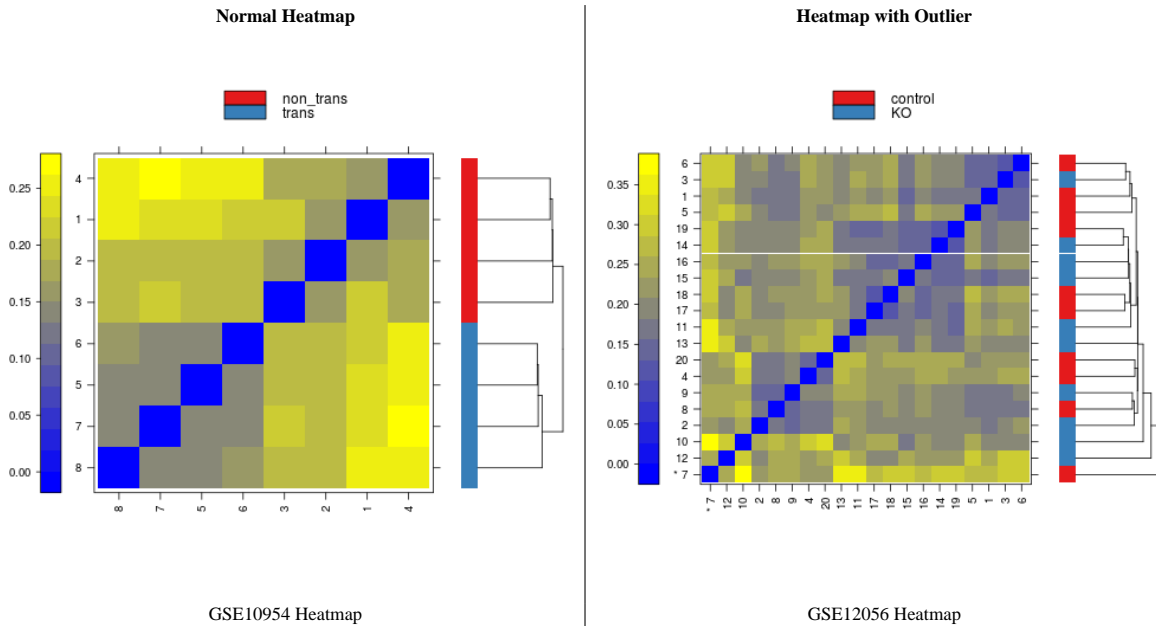
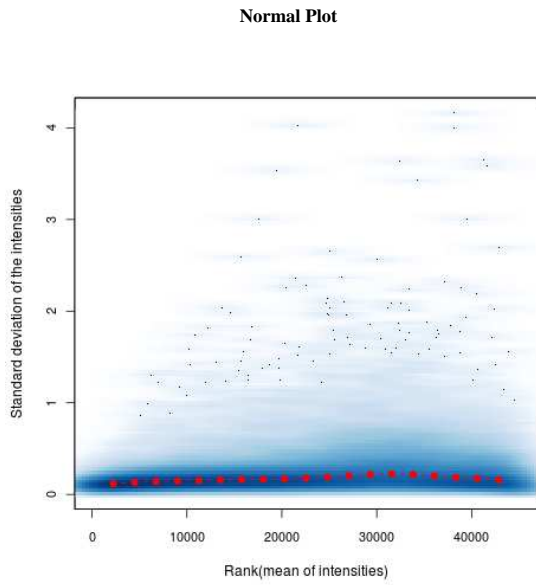


Figure 3: Examples of Between-Sample Heatmaps, with and without outliers.

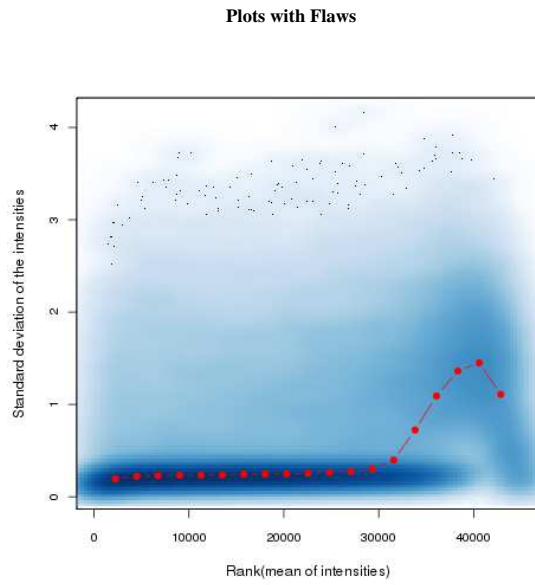
### 3.3 Variance vs. Mean Plot

To visualize the heteroskedasticity of the data, we plot standard deviation of each data point versus the rank of its mean. In an experiment in which background correction, normalization, and summarization have been done correctly, the data will in fact be homoskedastic, meaning that the variance is uniformly distributed over the data. The trend line (the red dotted line) in this case will be flat. A falling trend in the low range of the dataset usually indicates problems with the background correction method. A rising trend in the high range of the dataset often indicates a saturated microarray scan caused by incorrect scanner settings. Both of these problems are signs that additional sources of noise exist in the dataset. These sources of noise will oftentimes affect clustering on the heatmap and PCA plots.

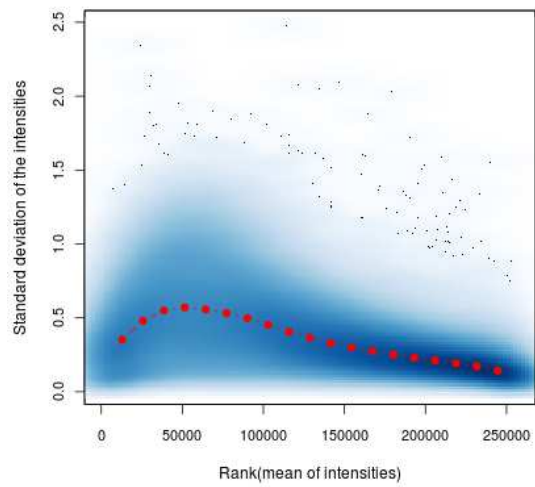
There are no outliers or warnings generated for the Variance vs. Mean Plot. This plot is generated only for the final (normalized and possibly batch corrected) data.



GSE18148 Variance vs. Mean Plot



GSE18617 Variance vs. Mean Plot  
This plot has a peak on the high end,  
which is probably due to saturation



GSE16622 Variance vs. Mean Plot  
This plot has a peak on the low end,  
which is due to a spatial defect

**Figure 4:** Examples of Variance vs. Mean Plots, with and without flaws.

### 3.4 MA Plot (Ratio vs. Intensity)

MA Plots are a classic per-array visualization for microarray data. They compare differential expression against average intensity for each probe or probeset in the dataset. MA plots for two-color Agilent data are typically computed for the direct comparison on the array. For other platforms, the data for all arrays are averaged to produce a baseline, and each individual array is compared against the baseline to determine M (ratio) and A (average intensity) values for the MA plot. MA plots for different arrays within the same experiment are expected to have similar distributions of differential expression versus intensity. Outlier detection for MA plots is performed by computing Hoeffding's statistic on the joint distribution of A and M for each array. Hoeffding's statistic is able to recognize a wide variety of distribution differences, but it is relatively lenient when applied to MA plots. These plots are generated for both raw and final (normalized and possibly batch corrected) data. The four highest and four lowest scoring arrays (according to Hoeffding's statistic) are shown in an 8-pane compilation of smoothed scatter plots. The other arrays are not shown because of the high computational cost and low benefit of generating them.

### 3.5 Affymetrix-Specific Plots

The arrayQualityMetrics package produces several additional plots for the analysis of Affymetrix raw data.

#### 3.5.1 RNA Degradation Plot

The RNA Degradation Plot shows the intensity trend over the probes of each probeset with the probes ordered from the 5' to 3' ends of the gene. The probesets are averaged to produce a single 5' to 3' trendline for each array. Since Affymetrix arrays are 3'-biased, it is expected that the RNA will show degradation (and therefore less hybridization) towards the 5' end of the probeset, so a positive slope is normal. If an array has a significantly different slope (or line shape) in the RNA Degradation Plot, it could mean that the RNA for that sample was mishandled. There are no outliers or warnings generated for the RNA Degradation Plot.

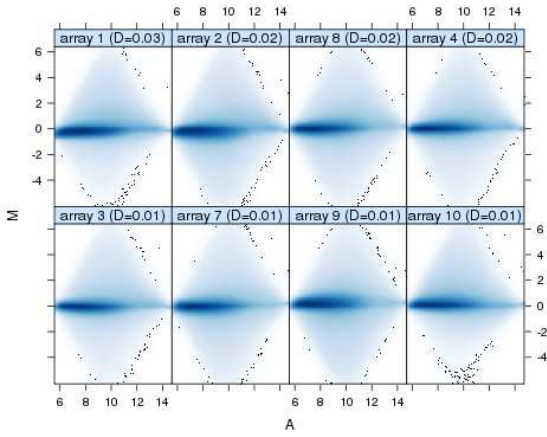
#### 3.5.2 Relative Log Expression Plot

The logged expression value for each probeset on each array is compared to the median logged expression value for the probeset across all arrays. The result is a relative log expression (RLE) value for each probeset. This box plot shows the quantiles and extents of the distribution of relative log expression values on each array. RLE distributions are one of the most reliable ways to detect flawed hybridization or scanning of Affymetrix arrays. In particular, the median of each RLE distribution should be very close to 0. Outliers are detected using a Kolmogorov-Smirnov test between each array's distribution and the distribution of the pooled data.

#### 3.5.3 Normalized Unscaled Standard Error Plot

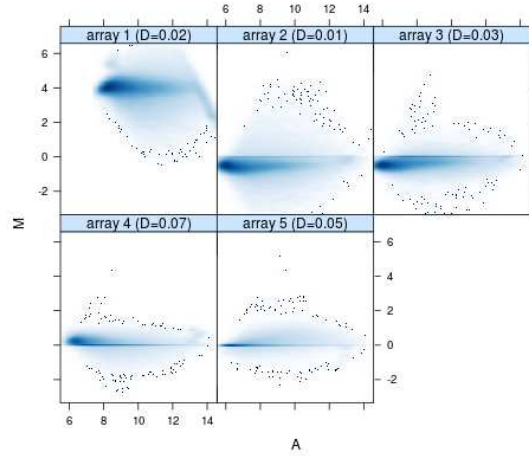
The standard error estimates for each gene are normalized across arrays so that the median for each gene is 1. The distribution of normalized standard errors are then shown as a boxplot. Each array's distribution should be centered at 1 (as a result of the normalization). An array with elevated normalized standard errors

**Normal Plots**

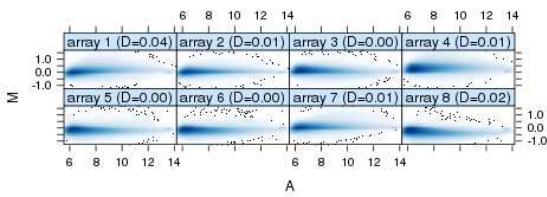


GSE18617 Raw MA Plot  
An unusually low-noise raw MA plot

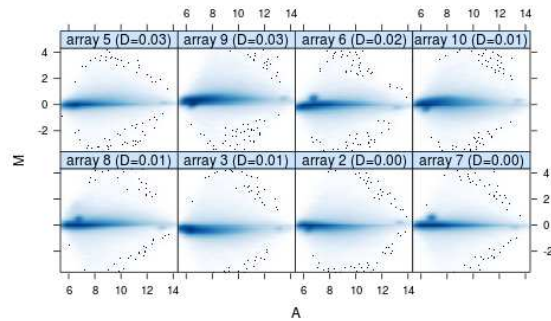
**Plots with Flaws**



GSE7836 Raw MA Plot  
Array 1 is severely shifted

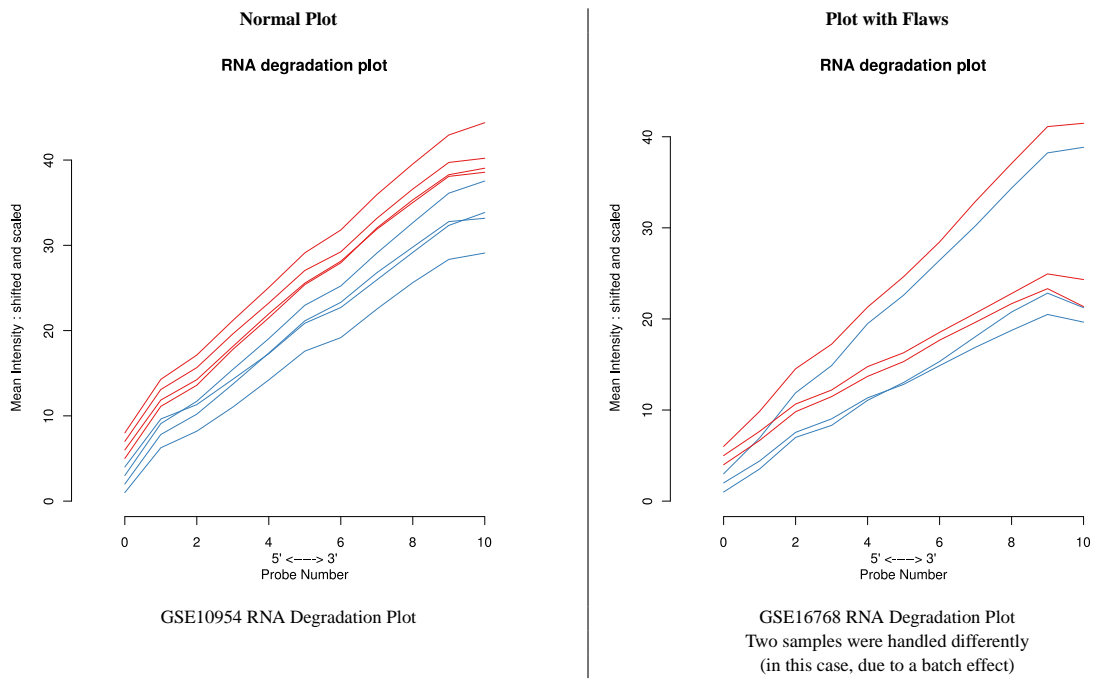


GSE10954 Raw MA Plot  
A typical raw MA plot

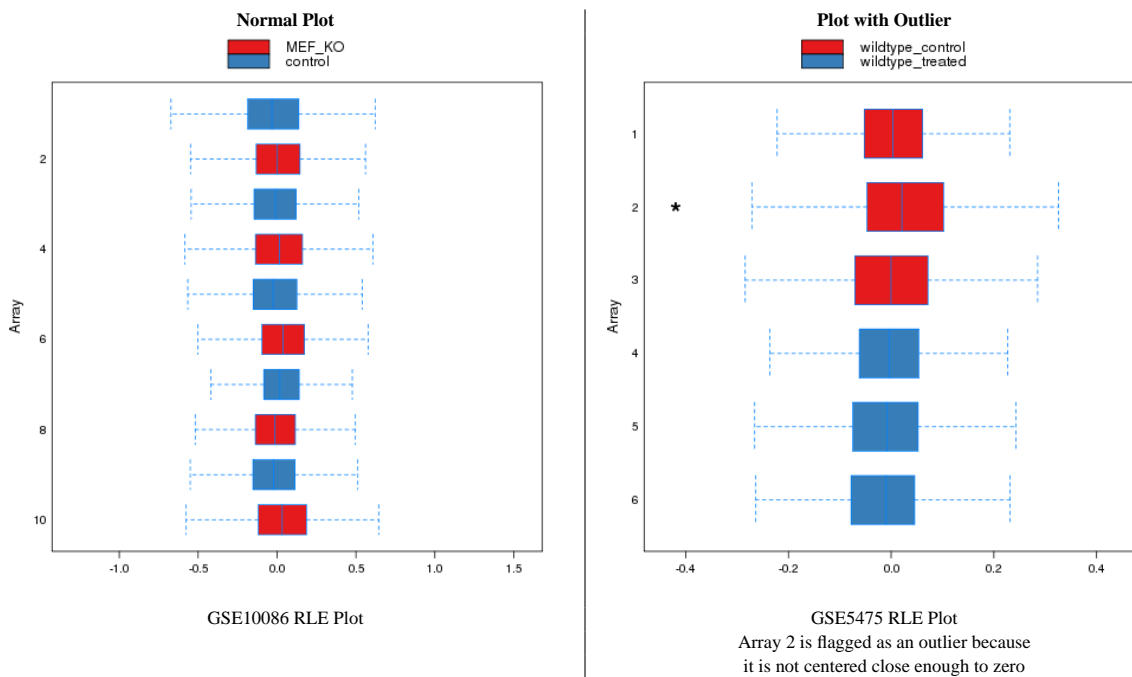


GSE16622 Raw MA Plot  
A spatial artifact (physical flaw on the array)  
appears as an off-center spot on this MA plot

**Figure 5:** Examples of MA Plots, with and without flaws.

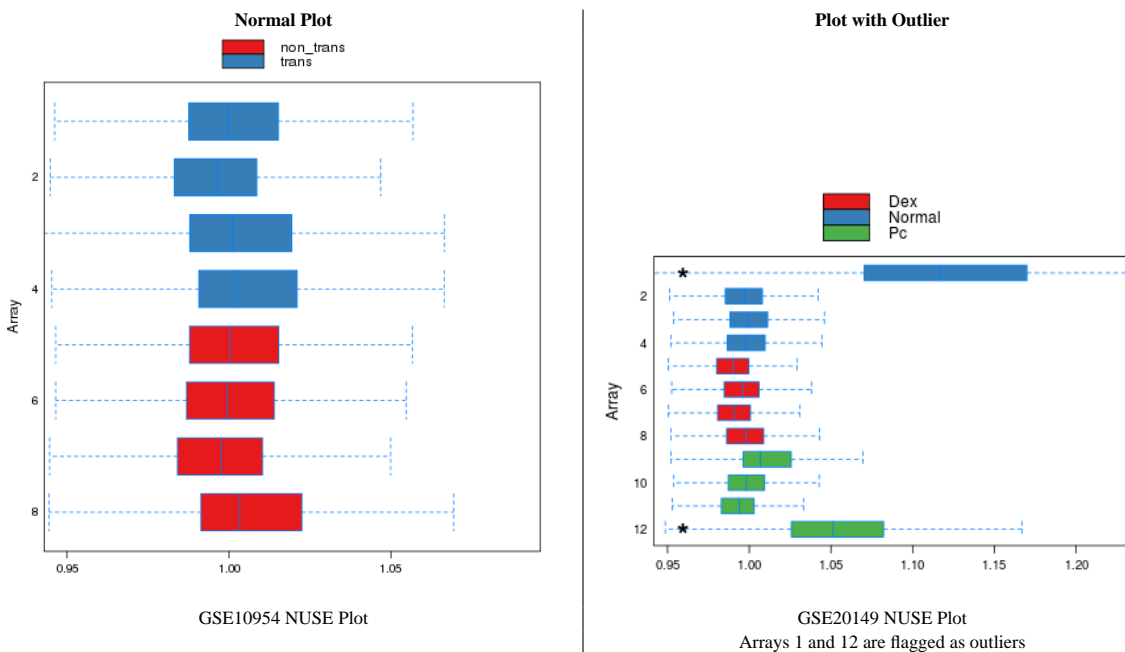


**Figure 6:** Examples of RNA Degradation Plots, with and without flaws.



**Figure 7:** Examples of Relative Log Expression Plots, with and without outliers.

is typically of lower quality. Outlier detection is performed by looking for arrays for which the upper quartile is beyond the 95% confidence interval of the distribution of the upper quartiles of all arrays.



**Figure 8:** Examples of Normalized Unscaled Standard Error Plots, with and without outliers.

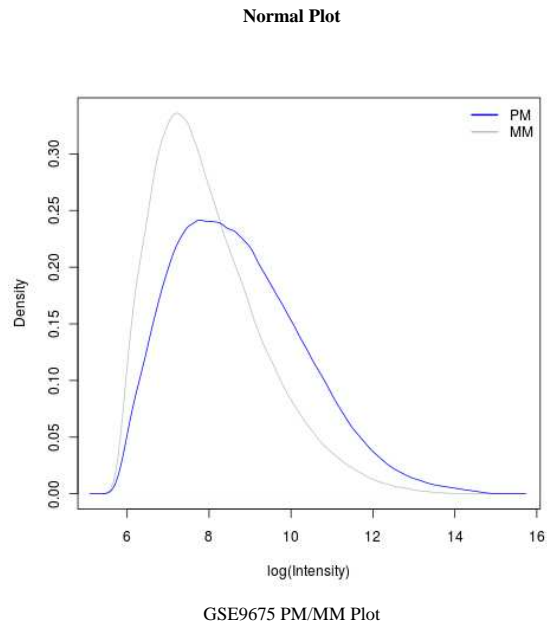
### 3.5.4 PM/MM Plot

Many Affymetrix microarrays contain both probes designed to match known transcripts, called *Perfect Match*, or *PM* probes, and probes designed to detect the amount of non-specific hybridization that occurs for those transcripts, called *Mis-Match* or *MM* probes. MM probes are identical to their PM counterparts except for the substitution of the complementary base at the 13th base position. This substitution prevents genuine copies of the probe’s target transcript from hybridizing to the probe, and in this way allows MM probes to serve as a type of background estimation.

The PM/MM Plot shows the smoothed histogram of the signals of PM and MM probes in the entire experiment. In any experiment where it is expected that the arrays will detect the presence of known transcripts, the PM signal should exceed the MM signal for some proportion of probesets. On the PM/MM plot, this situation appears as the PM distribution being slightly shifted to the right of the MM distribution over some segment of the plot. There are no outliers or warnings generated for the PM/MM plot.

### 3.6 PCA and PCA-3D Plots

*Principal Components Analysis* (PCA) plots are an extremely valuable tool for evaluating the quality and correctness of a microarray experiment. Normalized microarray data is a high-dimensional space: there is one dimension for each probeset. Each array can be thought of as a point in this high dimensional space. However, because of the high dimensionality, it is not practical to visualize the arrays this way. Principal Components Analysis is a technique for reducing the dimensionality of high-dimensional data in order to visualize and analyze the data. It works by detecting the direction of highest variability in the data, and



**Figure 9:** Example of a PM/MM Plot, without flaws.

calling this the first principal component. This process is then repeated, but each subsequent principal component must be orthogonal to all previously discovered principal components. The end result is a set of orthogonal vectors in microarray-space that are then used as the new axes (more precisely, the projection of each array onto the principal component vectors is taken). Each array can then be plotted as a single point in this new, lower-dimensional space. The PCA plot shows two-dimensional plots of each combination of the first three principal components. The PCA-3D plot shows a single three dimensional plot of the first three principal components. These two plots are best used in conjunction to construct a clear view of the organization of the data.

In a low-noise experiment with a large experimental effect, the first principal component (called *PC1*) will be the direction of change caused by manipulating the experimental variable, and the second principal component will be some of the biological variation between replicates. Subsequent principal components may be other types of biological variation, measurement noise, or other minor effects. The PCA plot in this case will show a wide separation between experimental groups along the PC1 axis, and will show smaller separation between the samples within each group along the PC2 axis.

In experiments with strong sources of noise, it's possible for a source of noise to cause the most variability in the data, in which case PC1 may be the direction of change caused by the noise. However, there may still be strong separation between the experimental groups along the PC2 axis, in which case it is still possible to detect differential expression reliably in the experiment.

In experiments with multiple strong sources of noise, or with systematic sources of noise that are confounded with the experimental variable, there may be little to no separation between experimental groups in the first two (or even three) principal components. In these experiments, it may not be possible to reliably detect differential expression. However, the statistical approaches used by the pipeline to detect differential expression take into account all types and sources of noise in the data when computing p-values, so even for this type of dataset, the results produced by the pipeline are not incorrect. In the worst case scenario, it may not be possible to detect differential expression in the experiment, and this circumstance will be reflected in the p-values produced by the pipeline.

In experiments where biological variation is larger than the variation caused by manipulating the experimental variable, the ability of the pipeline to detect differential expression can be improved drastically by performing a paired experiment. It can also be improved incrementally by increasing the number of biological replicates in the experiment.

### 3.7 PCA Plot Separation Analysis

Our pipeline employs a simple but novel technique for qualitatively evaluating the separation between groups of arrays on a PCA plot. Given an arbitrary grouping of the arrays on a (two-dimensional) PCA plot, we compute a minimum bounding ellipse. Then, for each pair of groups, we compute the minimum distance between the bounding ellipses of the groups, and divide it by the larger of the radii of the two ellipses in the direction of each other. The radius is considered a rough estimate of the maximum variance of the group in the direction of the other group, so a score of 1 on this scale represents a group whose center is at least twice as far from the other group as the variance of the data in that direction. If the ellipses overlap, then the size of the overlap is divided by the area of the smaller ellipse, and that value is negated to produce a negative score. Any score above 1 is interpreted as *strong separation*. Any score between 0 and 1 is considered *weak separation*. Any score below 0 is considered *no separation*. Note that a plot with no separation between groups could still have a statistically significant difference between the distributions of points for each group, and also that this approach is sensitive to outliers. This makes the separation score prone to underestimation. Therefore, this measure is best used in contexts where an underestimated value does not affect the conclusion, and for user-reviewable warnings. In the presence of outliers, drawing attention to the PCA plot by reporting that there is no separation may be an appropriate action, depending on how the separation score is being applied.

### 3.8 Experimental Effect Size Estimation (PC1 vs. PC2 Quality Plot)

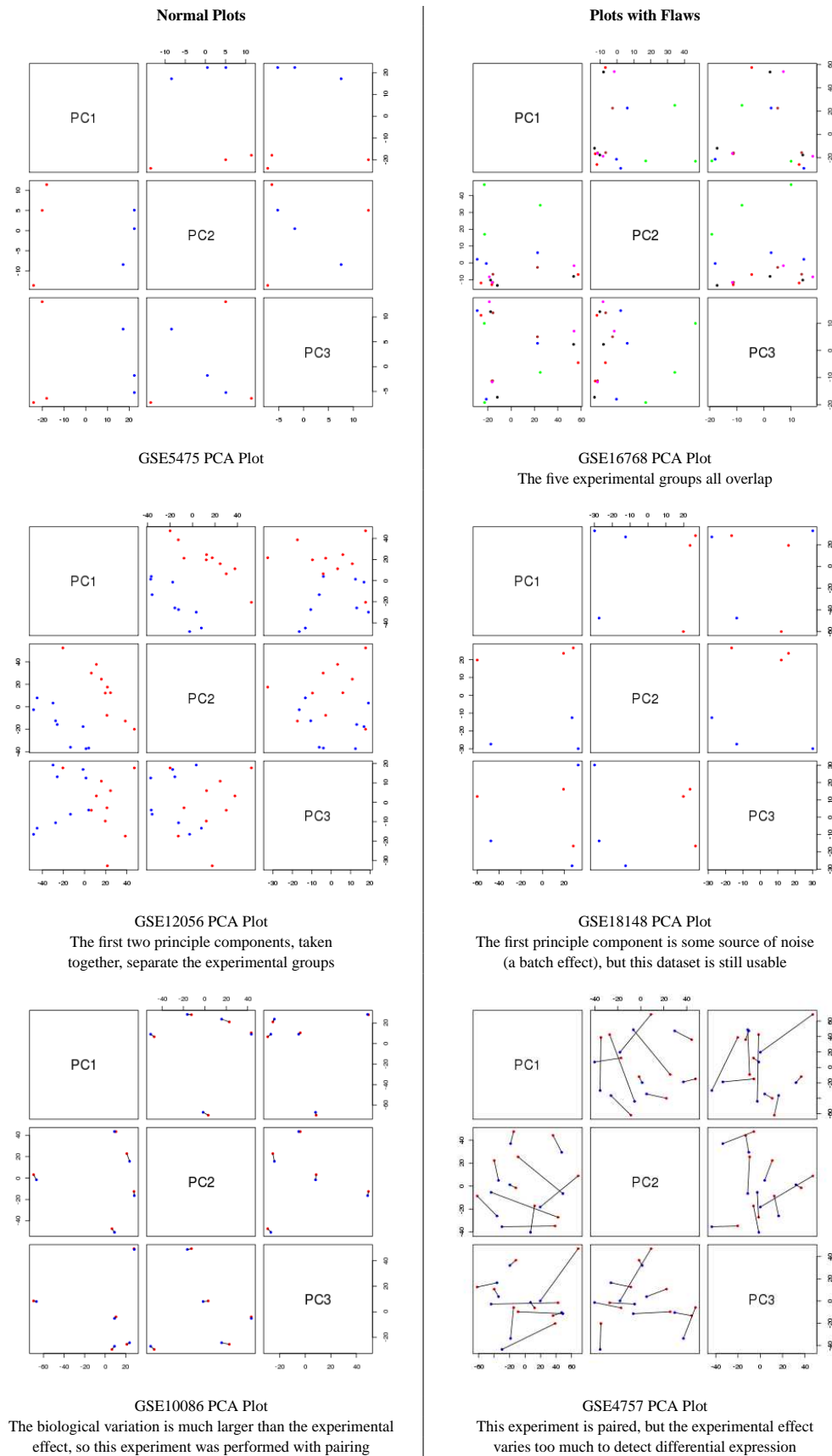
Separation Analysis is used to estimate the level of separation between experimental groups. It is applied to the plot of PC1 versus PC2 over the experimental groups. The experimental groups with the lowest separation score (indicating the least separation) are used to generate quality warnings for the experiment. If the separation score is negative, a "No Separation on PCA Plot" warning is issued. If the separation score is between 0 and 1, a "Weak Separation on PCA Plot" warning is issued. Either warning can be safely ignored if the user knows the dataset to have a sufficiently large number of replicates to overcome the noise. Otherwise, there is a chance of false negatives: that differentially expressed genes will not be recognizable statistically because of the level of noise in the experiment. The plot and ellipses are shown in the PC1 vs. PC2 Quality plot. This plot is generated both before and after batch correction (if batch correction is performed).

### 3.9 Batch Detection and Batch Effect Size Estimation (PC1 vs. PC2 Batch Effect Analysis Plot)

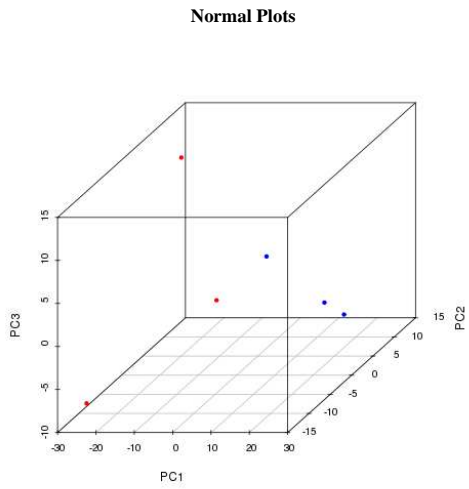
The pipeline detects the presence of batches in Affymetrix and Agilent datasets by checking the scan dates of the arrays. Unfortunately, scan date is not available for Illumina GenomeStudio output files, so no batch detection is done for these arrays. If batches are found, and each batch contains at least one array from each condition<sup>1</sup>, then Separation Analysis is used to determine if there is a significant batch effect associated

---

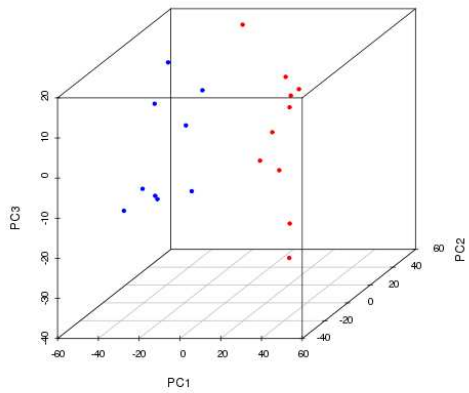
<sup>1</sup>A requirement in order to perform batch correction using ComBat



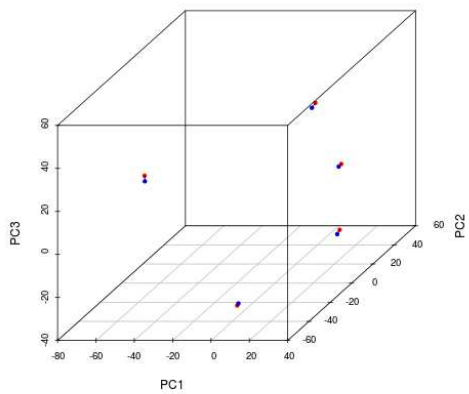
**Figure 10:** Examples of PCA Plots, with and without flaws.



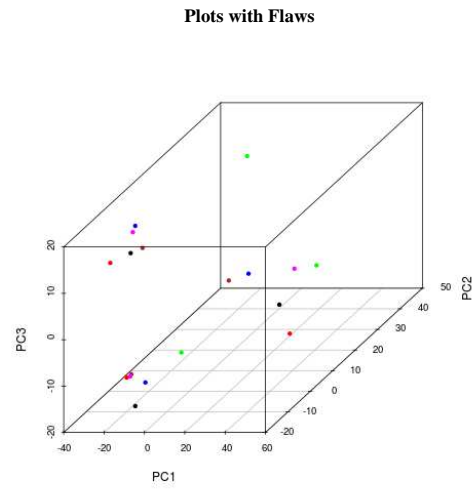
GSE5475 PCA-3D Plot



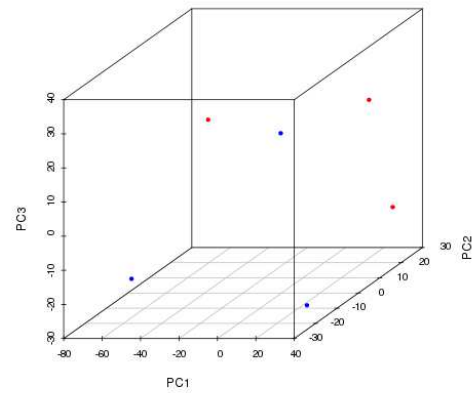
GSE12056 PCA-3D Plot  
The first two principle components, taken together, separate the experimental groups



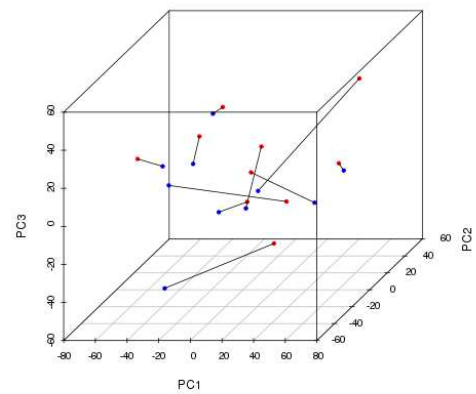
GSE10086 PCA-3D Plot  
The biological variation is much larger than the experimental effect, so this experiment was performed with pairing



GSE16768 PCA-3D Plot  
The five experimental groups all overlap

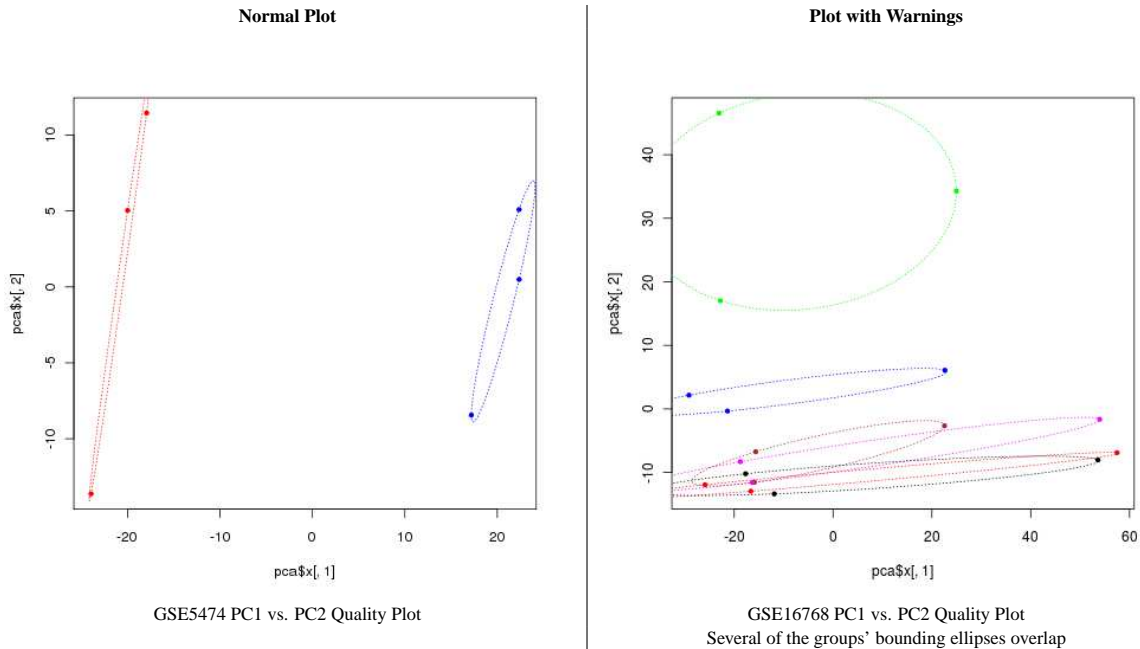


GSE18148 PCA-3D Plot  
The first principle component is some source of noise (a batch effect), but this dataset is still usable



GSE4757 PCA-3D Plot  
This experiment is paired, but the experimental effect varies too much to detect differential expression

**Figure 11:** Examples of PCA-3D Plots, with and without flaws.



**Figure 12:** Examples of PC1 vs. PC2 Quality Plots, with and without warnings.

with the batches. If any pair of batches in the experiment have a separation score greater than 1, then batch correction is recommended. Note that an underestimated separation score would lead to skipping batch correction, which may be a less optimal analysis, but is generally a more conservative one.

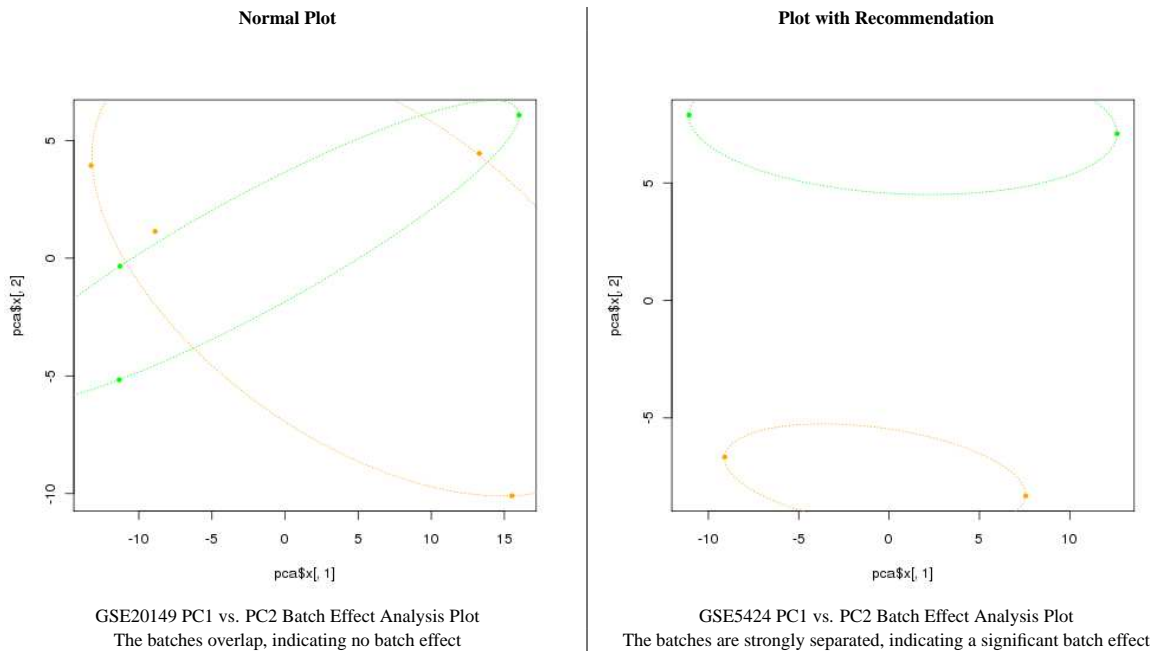
### 3.10 Quality Recommendations

In addition to batch correction recommendations, the pipeline also generates file removal recommendations. Each outlier flag is reported to the user as an array that did not pass all quality tests. If a single array receives three or more outlier flags, the pipeline recommends removing the file from the experiment. Although this is an extremely simple metric, it is quite effective in practice. An array that is deficient in just one metric is typically correctable by normalization, or was flagged in the first place due to an oversensitive metric. An array that has a minor problem typically receives two outlier flags, because most kinds of minor deficiencies are detectable by two of the metrics. Arrays that receive three or more outlier flags are either seriously deficient in one way, or marginally deficient in two or more. Arrays with these types of problems are not correctable by normalization, and detract from the ability of the pipeline to detect differentially expressed genes. Therefore, they are recommended for removal.

Users who wish to implement stricter quality control policies are free to review the outlier flags, warnings, and recommendations and make changes at will. However, the automated controls are sufficient to produce reliable results.

## 4 Experimental Validation

We evaluate the performance of our pipeline by analyzing 19 publically available datasets representing six broad areas of life sciences research: immunology, neuroscience, cell/cancer biology, developmental



**Figure 13:** Examples of PC1 vs. PC2 Batch Effect Analysis Plots, with and without recommendations.

biology, molecular biology, and metabolism/nutrition. Initially, we measure the pipeline’s output using several different sets of significance criteria. We use these initial results to tune the automatic selection of significance criteria, and we then validate our algorithm by evaluating the pipeline against 10 additional datasets.

#### 4.1 Datasets for Tuning

Datasets were selected from the NCBI’s Gene Expression Omnibus (GEO) (<http://www.ncbi.nlm.nih.gov/>) using the following criteria:

- From one of the areas of immunology, neuroscience, cell/cancer biology, developmental biology, molecular biology, and metabolism/nutrition.
- Uses a human, mouse, or rat Affymetrix 3’ IVT array.
- Includes a direct comparison of biological interest - one experimental condition versus control.
- Includes a publication in a peer reviewed journal, along with a list of differentially expressed genes.

Table 1 summarizes these datasets. Several of these datasets include more than one published comparison, which have been included in our testing for a total of 22 datasets. Our hope is that the 22 datasets will be approximately representative of the population of datasets that might be submitted to an automated pipeline such as this. That population is likely to include both flawed datasets and unusual experimental designs, and as such, the quality of the dataset and the quality of the publication are not a part of our criteria.

**Table 1:** Datasets for analysis and tuning of pipeline.

GEO Accession	Journal	Year	Subject	Description	Array Type	Replicates/ Condition*
GSE4757	Neurobiol Aging	2006	Neuroscience	Alzheimer's Disease	hgu133plus2	10
GSE5424	BMC Genomics	2008	Developmental Biology	Foxa2 mutant embryos	mgu74av2	2
GSE5475	Physiol Genomics	2007	Metabolism, Nutrition	PPARa activation	moe430a	3
GSE7836	Virology	2009	Cell Biology	SV40 transformation	rgu34a	3 (2)
GSE9675	BMC Genomics	2009	Developmental Biology	Maternal Diabetes	moe430a	5 (2)
GSE9914	PNAS	2007	Neuroscience	SCA1, SCA7 knock-in	moe430a	6
GSE10309	Am J Repir Crit Care Med	2008	Cancer Biology	Claudin-1 Overexpression	hgu133plus2	2
GSE10954	BMC Systems Biology	2008	Cancer Biology, Transcription	c-Myc transgenic mice	mouse4302	4
GSE12056	BMC Cancer	2008	Cancer Biology, Transcription	CREB knock-down	hgu133plus2	10
GSE13460	PloS one	2008	Developmental Biology	miR-122 overexpression in hESC	hgu133a2	2 (3)
GSE16768	BMC Genomics	2009	Neuroscience	UCB treatment of SH-SY5Y cells	hgu133a2	3
GSE17204	Journal of Biological Chemistry	2010	Neuroscience	DJ-1 silenced SH-SY5Y	hgu133a2	4
GSE18148	Cell Immunity	2009	Immunology	Cbfb-deficient Treg cells	mouse4302	3
GSE18617	PloS one	2010	Neuroscience	Bergman glial cell transcriptome	mouse4302	5
GSE18740	Journal of Neuroinflammation	2010	Immunology, Neuroscience	Luteolin-treated microglia	mouse4302	3
GSE20051	PNAS	2010	Cancer Biology	Raf inhibition in melanoma cells	hgu133a2	5
GSE20097	Nature Cell Bio	2010	Cancer Biology	miR-19 expression in FL5-12 cells	mouse430a2	3
GSE20149	BMC Microbiology	2010	Immunology	Pneumocystis carinii infection	rgu34a	4

\*Control replicates shown in parentheses for unbalanced experiments.

## 4.2 Curation of Gene Lists

For each dataset, we manually curate a list of Differentially Expressed Genes (DEGs). The gene list is taken from the publication text from a paragraph, table, or figure listing DEGs of interest. Twelve of these lists are genes whose differential expression was validated by followup experimentation. Five of the lists are genes selected by the authors for followup biological analysis, and found to be relevant. Two more of the lists are genes that are discussed in the publication for their biological interest, but no systematic followup study is mentioned. Of these, one list was not actually published, and had to be generated based on the criteria described in the publication. Two of the lists are the top genes by significance using an arbitrary cutoff (top 80 for one, top 100 for the other). Finally, one publication did not provide a published list comparable to the others. That publication provided only a list of all of the 748 genes that they found to be statistically significant. From that, we generated a list to stand-in for the published list by applying a fold change cutoff of 2, leaving 45 genes.

By taking biologically or experimentally validated lists of genes, we hope to capture the key discoveries (*key genes*) of each microarray experiment.

## 4.3 Evaluation Techniques

First, we measure the number of genes *selected* (found to be significantly differentially expressed) by the pipeline.

We then evaluate the pipeline by measuring its ability to find key genes. We express this measurement, which we call *coverage*, as the percentage of key genes that are selected by the pipeline. This definition intentionally excludes consideration of genes discarded by the authors. We consider the pipeline's ability to find key genes to be the ideal metric, as these key genes represent the end result of a microarray experiment. Filtering the list of selected genes down to a list of key genes is a valuable process of biological discovery that should be performed by the author, provided that the initial list of genes is of a manageable size.

**Table 2:** Sensitivity of our pipeline as measured against published lists of differentially expressed genes.

dataset	$FDR \leq 0.05$				$p \leq 0.05$			
	$ FoldChange  \geq 1.2$		$ FoldChange  \geq 1.5$		$ FoldChange  \geq 1.2$		$ FoldChange  \geq 1.5$	
	# genes	coverage (%)	# genes	coverage (%)	# genes	coverage (%)	# genes	coverage (%)
GSE4757	3	1.3	3	4.3	2857	72.0	296	34.8
GSE5424a	352	56.2	126	53.8	1149	87.5	135	53.8
GSE5424b	0	0.0	0	0.0	4988	86.2	637	65.4
GSE5475	2266	100.0	624	75.0	2571	100.0	643	75.0
GSE7836	99	58.5	99	62.9	974	100.0	752	100.0
GSE9675	666	87.0	234	78.3	2662	100.0	350	82.6
GSE9914a	1202	71.4	7	14.3	1391	71.4	11	14.3
GSE9914b	16	0.0	8	0.0	270	28.6	10	0.0
GSE10086	2043	100.0	693	100.0	3231	100.0	779	100.0
GSE10309	325	22.2	325	22.2	7719	88.9	3328	88.9
GSE10954	5464	100.0	1646	100.0	7456	100.0	1701	100.0
GSE12056	5289	77.3	1168	63.3	6036	80.3	1211	65.3
GSE13460	0	0.0	0	0.0	1029	100.0	176	96.0
GSE16768	158	88.9	151	88.9	1823	100.0	607	100.0
GSE17204	480	96.2	282	95.8	2887	96.2	806	95.8
GSE18148	41	44.4	41	44.4	1928	100.0	658	100.0
GSE18617	367	92.5	361	92.5	4861	98.1	2817	98.1
GSE18740a	51	12.5	50	12.5	4006	100.0	1639	100.0
GSE18740b	0	0.0	0	0.0	2463	100.0	556	100.0
GSE20051	3096	100.0	959	100.0	4179	100.0	1040	100.0
GSE20097	1	0.0	1	0.0	719	31.1	82	13.3
GSE20149	2191	97.1	860	97.1	2427	97.1	871	97.1

\*This dataset is actually GSE10086, but we use it in conjunction with the publication for GSE20051.

## 4.4 Initial Results

Results of the pipeline are shown for four sets of significance criteria: two fold change cutoffs, each with and without controlling the False Discovery Rate (Table 2).

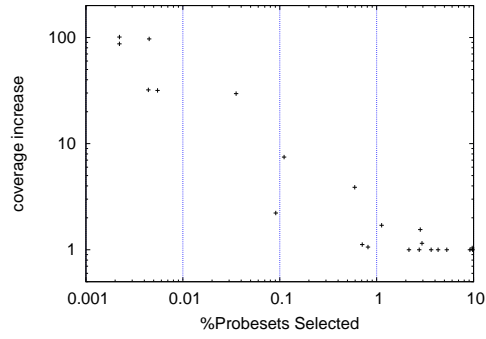
In our initial characterization, we look for two levels of coverage, *high coverage* (over 90%) and *adequate coverage* (over 50%). Using stringent standards for statistical significance ( $FDR \leq 0.05$ ,  $|FC| \geq 1.5$ ), our pipeline has high coverage with the published gene lists in almost one third of datasets (7/22), and adequate coverage in another 6 datasets. Falling back on lenient standards ( $p \leq 0.05$ ,  $|FC| \geq 1.5$ ), 4 additional datasets reach high coverage and 2 others reach adequate coverage. All of these 19 datasets reach the aforementioned levels of coverage of the published gene lists without selecting more than 5% of probesets.

Only 3 datasets fail to reach adequate coverage without selecting more than 5% of probesets. For 2 of these, the published genes are selected by the pipeline if a low enough fold-change cutoff is used, but the total number of genes selected in this case is prohibitively large. Finally, for GSE20097, most of the key genes are not selected by the pipeline using even the most lenient significance criteria. While a thorough treatment of this case is beyond the scope of this work, we note that the authors use an uncommon normalization technique, which may account for the low level of agreement between their results and ours. While we have no reason to believe that their results are not valid, we do believe that our pipeline's findings are valid for followup biological analysis as well. We take this dataset as an example of where exploratory analysis might be needed.

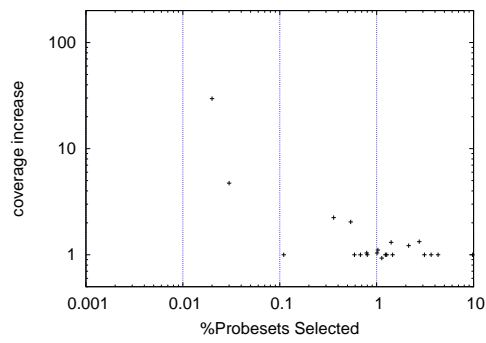
## 4.5 Determination of Cutoffs

We use a variety of observations to determine the ideal cutoffs for fold change and for the control of FDR:

- The authors in our studies typically start with a list of a few hundred significant genes (max of 4028, mean of 717, median of 255, excluding one extreme outlier who lists every probe), and narrow down



**Figure 14:** Benefit of switching to lenient criteria.  
Zero values have been offset.



**Figure 15:** Benefit of relaxing fold change cutoff.  
Zero values have been offset.

to a few dozen genes before publication (max of 104, mean of 40, median of 28).

- Figure 14 shows the percentage of probesets selected for each dataset using the stringent criteria versus the increase in coverage attained by switching to the lenient criteria. There is an inflection point around 0.1% of probesets, below which the increase in coverage is significant.
- Figure 15 shows the percentage of probesets selected for each dataset using a fold change cutoff of 1.5 versus the increase in coverage attained by switching to a fold change cutoff of 1.2. Again, there is some of an inflection point around 0.1% of probesets.

For these reasons, we design our pipeline to select at least 0.1% of probesets. Based on industry standards, we control FDR unless fewer than 0.1% of probesets are selected using this approach. After determining whether or not to control FDR, the pipeline chooses a (recommended) fold change cutoff. Based on the excellent coverage found using a fold change cutoff of 1.5, we take that as the preferred value. The fold change cutoff recommendation is raised to 2, 2.5, 3, 3.5, 4, 4.5, or 5 only if more than 2000 of probesets are selected. The fold change cutoff is lowered to 1.2, 1.1, or 1.0 only if fewer than 0.1% of probesets are selected. The results of applying these limits automatically to the first 22 datasets are shown in Table 3. Regardless of fold change cutoff recommendation, the pipeline retains information about all probesets.

## 4.6 Final Results

**Table 3:** Pipeline results on original 22 datasets using automated preferences for statistical significance.

Study	Error Control	Fold Change Cutoff	# Genes Found	Coverage (%)
GSE9914b	none	1.2	270	28.6
GSE20097	none	1.5	82	13.3
GSE4757	none	1.5	296	34.8
GSE5424b	none	1.5	637	65.4
GSE13460	none	1.5	176	96.0
GSE18148	none	1.5	658	100.0
GSE18740b	none	1.5	556	100.0
GSE9914a	FDR	1.2	1202	71.4
GSE18740a	FDR	1.5	50	12.5
GSE10309	FDR	1.5	325	22.2
GSE5424a	FDR	1.5	126	53.8
GSE7836	FDR	1.5	99	62.9
GSE12056	FDR	1.5	1168	63.3
GSE5475	FDR	1.5	624	75.0
GSE9675	FDR	1.5	234	78.3
GSE16768	FDR	1.5	151	88.9
GSE18617	FDR	1.5	361	92.5
GSE17204	FDR	1.5	282	95.8
GSE20149	FDR	1.5	860	97.1
GSE20051a	FDR	1.5	959	100.0
GSE20051b	FDR	1.5	693	100.0
GSE10954	FDR	1.5	1646	100.0
mean			521	70.5
median			343	76.7

On average, the pipeline selects 497 genes which include 70.5% of the published gene list for each experiment. Nearly two-thirds of the datasets exceed 70% coverage using these defaults. Fewer than a quarter of the datasets we studied have low coverage. In two of those five cases, lowering the fold change cutoff increases the coverage to more than 70%, and in two others, choosing not to control the false discovery rate does the same. For one of the 22 datasets (4.5%), exploratory analysis might be needed. Also of note is that the authors of four of these five studies used lenient definitions of statistical significance. Our pipeline,

however, errs on the side of caution by choosing a smaller set of differentially expressed genes to consider significant, and controlling FDR when possible.

Finally, we test the pipeline’s performance against an additional 10 datasets in order to validate the automated cutoffs. These datasets are shown in Table 4. The pipeline finds an average of 668 differentially expressed genes per dataset, which includes an average of 82% of published genes. None of the 10 validation datasets had less than adequate coverage, and none require exploratory analysis.

**Table 4:** Pipeline validation on an independent set of 10 datasets.

Study	Error Control	Fold Change Cutoff	# Genes Found	Coverage (%)
GSE7836	none	1.5	327	100
GSE10935	FDR	1.5	608	62.5
GSE18887	FDR	1.5	729	70
GSE3866	FDR	1.5	856	90
GSE24451	FDR	1.5	333	95.5
GSE16622	FDR	1.5	172	100
GSE4600	FDR	1.5	1542	100
GSE14043	FDR	2	509	67.9
GSE11414b	FDR	3	756	73.2
GSE11414a	FDR	4	846	62.5
mean			668	82
median			669	82

We believe this high level of agreement with authors’ own methods makes the results of the pipeline a solid basis for followup biological analysis.

## 5 Conclusion

We built and tested a fully automated microarray analysis pipeline. The pipeline incorporates industry standard normalization, statistical analysis, and error control. It also includes modern batch correction and quality controls. Furthermore, we add automatic decisions for quality control, and automatic determination of significance cutoffs. We validate our pipeline by applying it to 22 publically available datasets in five broad areas of life sciences research, and measuring its performance against the authors’ own findings. We find that our pipeline selects over 70% of the genes in authors’ published gene lists, while limiting the total genes found to a managable number. Overall, we believe these results constitute strong evidence that unattended microarray analysis is viable.

## Acknowledgements

We would like to acknowledge W. Evan Johnson for his assistance on the correct use of ComBat for two-color Agilent microarray data. We would also like to thank Andy Lyons and Nima Moshtagh for publishing liberally licensed R code to compute a minimum bounding ellipse for a set of points.

## References

- [1] D. B. Allison, X. Cui, G. P. Page, and M. Sabripour. Microarray data analysis: from disarray to consolidation and consensus. *Nat Rev Genet*, 7(1):55–65, Jan 2006.

- [2] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society Series B Methodological*, 57(1):289–300, 1995.
- [3] 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 (Oxford, England)*, 19(2):185–93, Jan 2003.
- [4] R. F. Chuaqui, R. F. Bonner, C. J. M. Best, J. W. Gillespie, M. J. Flaig, S. M. Hewitt, J. L. Phillips, D. B. Krizman, M. A. Tangrea, M. Ahram, W. M. Linehan, V. Knezevic, and M. R. Emmert-Buck. Post-analysis follow-up and validation of microarray experiments. *Nat Genet*.
- [5] L. M. Cope, R. A. Irizarry, H. A. Jaffee, Z. Wu, and T. P. Speed. A benchmark for affymetrix genechip expression measures. *Bioinformatics*, 20(3):323–331, 2004.
- [6] M. Dondrup, A. T. Hser, D. Mertens, and A. Goesmann. An evaluation framework for statistical tests on microarray data. *Journal of Biotechnology*, 140(1-2):18 – 26, 2009. Functional Genome Research on Bacteria Relevant for Agriculture, Environment and Biotechnology - Functional Genome Research.
- [7] R. C. Gentleman, V. J. Carey, D. M. Bates, et al. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.
- [8] F. Giorgi, A. Bolger, M. Lohse, and B. Usadel. Algorithm-driven artifacts in median polish summarization of microarray data. *BMC Bioinformatics*, 11(1):553, 2010.
- [9] B. Harr and C. Schlötterer. Comparison of algorithms for the analysis of affymetrix microarray data as evaluated by co-expression of genes in known operons. *Nucleic Acids Research*, 34(2):e8.
- [10] D. Holder, R. F. Raubertas, V. B. Pikounis, V. Svetnik, and K. Soper. Statistical analysis of high density oligonucleotide arrays: a safer approach. 2001.
- [11] E. Hubbell, W.-M. Liu, and R. Mei. Robust estimators for expression analysis. *Bioinformatics*, 18(12):1585–1592, 2002.
- [12] W. Huber, A. von Heydebreck, H. Sltmann, A. Poustka, and M. Vingron. Variance stabilization applied to microarray data calibration and to the quantification of differential expression. *Bioinformatics*, 18(suppl 1):S96–S104, 2002.
- [13] 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, 2003.
- [14] 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):249–264, 2003.
- [15] I. Jeffery, D. Higgins, and A. Culhane. Comparison and evaluation of methods for generating differentially expressed gene lists from microarray data. *BMC Bioinformatics*, 7(1):359, 2006.
- [16] 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.
- [17] L. JT, S. RB, B. HC, S. D, L. B, J. WE, G. D, B. K, and I. RA. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet*, 11(10):733–733, Oct 2010.

- [18] A. Kauffmann and W. Huber. *arrayQualityMetrics: Quality metrics on microarray data sets*. R package version 2.2.3.
- [19] C. Li and W. H. Wong. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proceedings of the National Academy of Sciences of the United States of America*, 98(1):31–36, 2001.
- [20] W. K. Lim, K. Wang, C. Lefebvre, and A. Califano. Comparative analysis of microarray normalization procedures: effects on reverse engineering gene networks. *Bioinformatics*, 23(13):i282–i288, 2007.
- [21] S. M. Lin, P. Du, W. Huber, and W. A. Kibbe. Model-based variance-stabilizing transformation for illumina microarray data. *Nucleic Acids Research*, 36(2):e11, 2008.
- [22] J. Luo, M. Schumacher, A. Scherer, D. Sanoudou, D. Megherbi, T. Davison, T. Shi, W. Tong, L. Shi, H. Hong, C. Zhao, F. Elloumi, W. Shi, R. Thomas, S. Lin, G. Tillinghast, G. Liu, Y. Zhou, D. Herman, Y. Li, Y. Deng, H. Fang, P. Bushel, M. Woods, and J. Zhang. A comparison of batch effect removal methods for enhancement of prediction performance using maqc-ii microarray gene expression data. *Pharmacogenomics J*, 10(4):278–291, Aug 2010.
- [23] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.
- [24] M. E. Ritchie, J. Silver, A. Oshlack, M. Holmes, D. Diyagama, A. Holloway, and G. K. Smyth. A comparison of background correction methods for two-colour microarrays. *Bioinformatics*, 23(20):2700–2707, 2007.
- [25] D. M. Rocke and B. Durbin. A model for measurement error for gene expression arrays. *Journal of Computational Biology*, 8:557–569, 2001.
- [26] J. D. Silver, M. E. Ritchie, and G. K. Smyth. Microarray background correction: maximum likelihood estimation for the normal-exponential convolution. *Biostatistics*, 10(2):352–363, 2009.
- [27] 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.
- [28] G. K. Smyth, Y. H. Yang, and T. Speed. Statistical issues in cdna microarray data analysis. In J. M. Walker, M. J. Brownstein, and A. B. Khodursky, editors, *Functional Genomics*, volume 224 of *Methods in Molecular Biology*, pages 111–136. Humana Press, 2003. 10.1385/1-59259-364-X:111.
- [29] V. G. Tusher, R. Tibshirani, and G. Chu. Significance analysis of microarrays applied to the ionizing radiation response. *Proceedings of the National Academy of Sciences of the United States of America*, 98(9):5116–5121, 2001.