Summer 2012

Cost effective, computer-aided analytical performance evaluation of chromosomal microarrays for clinical laboratories

Corey William Goodman
University of Iowa

COST EFFECTIVE, COMPUTER-AIDED ANALYTICAL PERFORMANCE EVALUATION OF CHROMOSOMAL MICROARRAYS FOR CLINICAL LABORATORIES

by
Corey William Goodman

A thesis submitted in partial fulfillment
of the requirements for the Master of
Science degree in Electrical and Computer
Engineering in the Graduate College of
The University of Iowa

July 2012

Thesis Supervisors: Professor Thomas L. Casavant
Assistant Professor Benjamin W. Darbro

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

MASTER'S THESIS

_____

This is to certify that the Master's thesis of

Corey William Goodman

has been approved by the Examining Committee for
the thesis requirement for the Master of Science degree in
Electrical and Computer Engineering at the July 2012 graduation.

Thesis Committee: _____
Thomas L. Casavant, Thesis Supervisor

_____
Benjamin W. Darbro, Thesis Supervisor

_____
Val C. Sheffield

_____
Terry A. Braun

_____
John P. Robinson

To my grandfather, William Conrad Goodman
who encouraged all his children and grandchildren
to attend an institution of higher learning, despite
lacking the opportunity to attend one himself.

ACKNOWLEDGMENTS

I would like to thank Professor Thomas Casavant for his guidance and support for this project, as well as all of his guidance through much of my academic career. With regard to this project, I would like to especially thank Benjamin Darbro whose assistance has been invaluable. I would also like to thank Val Sheffield, Terry Braun, and John Robinson for serving on my thesis committee. Lastly, I would also like to thank my coworkers, family, and friends for their support and encouragement. All have helped make my time at the University of Iowa very meaningful and enjoyable.

ABSTRACT

Many disorders found in humans are caused by abnormalities in DNA. Genetic testing of DNA provides a way for clinicians to identify disease-causing mutations in patients. Once patients with potentially disease-causing mutations are identified, they can be enrolled in treatment or preventative programs to improve the patients' long term quality of life. Array-based comparative genomic hybridization (aCGH) provides a high-resolution, genome-wide method for detecting chromosomal abnormalities. Using computer software, chromosome abnormalities, or copy number variations (CNVs) can be identified from aCGH data. The development of a software tool to analyze the performance of CGH microarrays is of great benefit to clinical laboratories. Calibration of parameters used in aCGH software tools can maximize the performance of these arrays in a clinical setting. According to the American College of Medical Genetics, the validation of a clinical chromosomal microarray platform should be performed by testing a large number (200-300) of well-characterized cases, each with unique CNVs located throughout the genome. Because of the Clinical Laboratory Improvement Amendment of 1988 and the lack of an FDA approved whole genome chromosomal microarray platform the ultimate responsibility for validating the performance characteristics of this technology falls to the clinical laboratory performing the testing. To facilitate this task, we have established a computational analytical validation procedure for CGH microarrays that is comprehensive, efficient, and low cost. This validation uses a higher resolution microarray to validate a lower resolution microarray with a receiver operating characteristic (ROC)-based analysis. From the results we are able to estimate an optimal $\log_2$ threshold range for determining the presence or absence (calling) of CNVs.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

CHAPTER 1

INTRODUCTION

Chromosomal microarray (CMA) is a broad term often used to describe clinical testing for DNA copy number variation (CNV) utilizing either single nucleotide polymorphism (SNP) or comparative genomic hybridization microarrays (aCGH). CMA has proven to have a much higher diagnostic yield than conventional chromosome analysis or sub-telomeric fluorescence in situ hybridization (FISH) for a range of developmental phenotypes (Hochstenback, et al. 2009; Shen, et al. 2010; Vissers, et al. 2010). This increase in diagnostic yield is facilitated in part by the much higher resolution of microarray strategies in comparison to conventional cytogenetic techniques. It has been recommended that chromosomal microarrays be a first-tier diagnostic test for individuals with non-syndromic intellectual disability/developmental delay, autism spectrum disorders, or multiple congenital anomalies (Miller, et al. 2010; Manning and Hudgins 2010).

In this thesis, CGH-based microarray platforms were used for all CMA tests performed. With the improvements in resolution that come from array-based CGH (aCGH) methods, discoveries of many polymorphic CNVs have been found in healthy individuals as well as novel pathogenic CNVs. Polymorphic (common) CNVs are regions of DNA in which deletions and/or duplications normally occur in the population at a frequency higher than would be expected to occur at random. Typically, polymorphic CNVs are benign or have no known clinical significance. The presence of common, polymorphic CNVs found in the "normal" population makes detection of novel pathogenic CNVs difficult.

A disadvantage of aCGH can be the test-to-test variability. This can arise from a number of different factors including, variation in equipment used in the testing process,

the individuals performing the tests, different laboratory environments, or laboratory environmental factors over time and between tests (especially humidity, heat and ozone levels). Other artifact sources include differential labeling efficiency of dyes for the test and reference sample, uneven spotting of DNA to the microarray, variations in washing efficiency, and variations in scanning efficiency of the microarray. The resulting variability can be partially addressed using normalization techniques, but variability in data from laboratory to laboratory can still exist. In comparisons between different array platforms, different laboratories performing tests, and different algorithms for CNV calling, it has been found that variation remains a vexing problem. When taking into account both large and small CNVs, a comparison among algorithms using the same raw data typically yielded less than 50%. It has also been demonstrated that the reproducibility of tests on many platforms was less than 70% (Pinto, et al. 2011).

Many algorithms used for calling CNVs have parameters, such as an intensity threshold or minimum length, that can be calibrated to each laboratory setting in order to optimize the performance of the tools. A software tool that can provide a general analytical calibration of parameters used in CNV calling algorithms can be of great use to clinical laboratories. No such tool is currently known to exist and would greatly enhance the diagnostic accuracy of this important clinical test.

Recently, the American College of Medical Genetics (ACMG) published new guidelines for the interpretation and reporting of constitutional CNVs as well as recommendations for the design and performance expectations of the microarrays used in clinical CMA testing (Kearney, South, et al. 2011; Kearney, Thorland, et al. 2011). Recommendations for microarray design and validation include the ability to detect genome-wide CNVs of at least 400Kb in size at 99% analytical sensitivity (with a lower limit of the 95% confidence interval >98%) and a false-positive rate of <1% (Kearney, South, et al. 2011). The ACMG places the task of this analytical validation on the array

manufacturers; however, in the absence of a manufacturer providing this data, it is ultimately the clinical laboratories' responsibility to validate the performance of these arrays, as there is still no FDA-approved, in-vitro diagnostic whole-genome chromosomal microarray platform or test kit.

According to the ACMG, the validation of a clinical chromosomal microarray should be performed by testing a sufficiently large number (200-300) of well-characterized cases that contain a collection of appropriately sized (majority being <1Mb), unique CNVs that are located throughout the genome. This is a difficult standard to meet for many clinical laboratories. Even with a diagnostic yield of ~15-20%, a clinical laboratory would have to test >1000 patients to naturally accumulate this many cases with diagnostic CNVs, and this does not account for factors such as recurrent CNVs or an uneven distribution across the genome.

In the research for this thesis a computer-aided receiver operator characteristic (ROC) based method has been developed that clinical laboratories can use to determine their own CMA platform analytical sensitivity and false positive rate. This method can utilize both novel and common/polymorphic CNVs with a per-probe approach (as opposed to per-CNV) to quantify outcome metrics such as true and false positive and negative results as well as sensitivity and specificity in general. The method developed can also be used to perform calibration of parameters used in algorithms that detect CNVs in aCGH data. Using two microarray designs that differ in their level of resolution it is shown that an analytical evaluation can be performed with as few as 20 cases in total.

CMA testing influences management of patients in many ways including generation of medical referrals, and providing guidance for diagnostic imaging and specific laboratory testing (Coulter, et al. 2011). Given the complexity of CMA testing and the implications it holds for future patient management, establishment of quality control and performance metrics is of utmost importance. Even if manufacturers of

chromosomal microarrays have performed extensive quality controls, it is still necessary for individual laboratories to verify these performance characteristics in their own setting and with their distinct patient populations. Thus, the establishment of a validation procedure for CMAs that is both comprehensive and low cost is of great significance to the clinical laboratory community.

CHAPTER 2

BACKGROUND

2.1 Array-Based Comparative Genomic Hybridization (aCGH)

Although the methods presented here can be applied to data from any chromosomal microarray with oligonucleotide probes, in this project the type of chromosomal microarray used was a comparative genomic hybridization array (array CGH or aCGH). Array-based comparative genomic hybridization is a method used to detect chromosomal anomalies on a genome-wide, high-resolution scale. An overview of the aCGH process is outlined in Figure 2.1 (Theisen 2008).

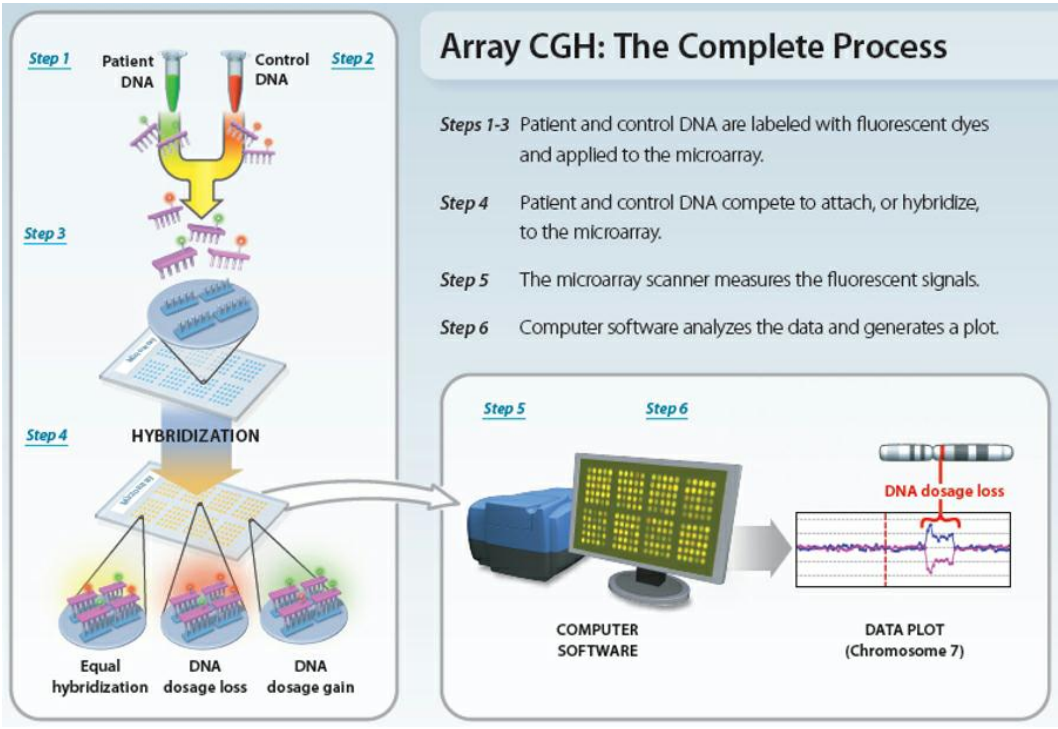

Figure 2.1: A diagram outlining the process of array-based comparative genomic hybridization. Used with permission

2.1.1 CGH Microarrays

Array CGH achieves a high testing resolution by using a substrate (e.g., a glass slide) with many small strands of single-stranded target DNA (oligonucleotides) attached to the substrate. These small amounts of DNA (commonly referred to as probes) are arranged into spots and immobilized onto a glass slide. The probes vary in length from small oligonucleotides of less than 50 base pairs (also referred to as mer) to large regions hundreds of thousands of base pairs long (commonly referred to as BACs). Probes are arranged in a way to prevent poor performance of a region of the microarray to affect the whole test. This is done by attempting to prevent probes found close together on the genome from being placed close together on the array. The length of the probes and the genomic distance between the probes determines the resolution of each array (Theisen 2008). It is not necessary that genomic locations of probes in a lower resolution microarray be a subset of probes in a higher resolution microarray. The primary array platforms used in this project were from the manufacturer Roche NimbleGen and are described in Table 2.1 (Roche NimbleGen n.d.).

Table 2.1: Details of Roche NimbleGen microarrays that were used for this project.

Array Description	Probe Length	Median Probe Spacing
Human CGH 385K Whole-Genome Tiling v2.0 Array	60mer	7073bp
Human CGH 720K Whole-Genome Tiling v3.0 Array	60mer	2509bp
Human CGH 2.1M Whole-Genome Tiling v2.0D Array	60mer	1169bp

2.1.2 Probe Hybridization

When a CMA test is performed, DNA is extracted from a sample of interest, or test sample, and labeled with a fluorescent dye with specific chromatic spectral characteristics (in most applications this is typically displayed as green as it is a reflection of the Cy3 dye used). A sample of DNA that is considered normal is used as a control/reference sample and labeled with a fluorescent dye differing in color from the sample of interest, typically red (Cy5 dye). The two samples are denatured into single-stranded DNA, mixed together, and then applied to the microarray. The single-stranded DNA from the test and reference samples will then be allowed to hybridize to probes matching their complementary sequence. The result will be regions with different levels of intensities between the two fluorescent dyes (Theisen 2008).

2.1.3 Probe Intensities

These levels of intensity between the two dyes provide a means for estimating the copy number of the test genome compared to the reference genome. For example, if a test sample was labeled with a green fluorescent dye and a reference sample was labeled with a red fluorescent dye, samples of equal copy number would appear yellow in color because spots on the microarray would have roughly equal amounts of each sample hybridizing to the probe for that region. Spots appearing green would indicate a higher copy number for that region in the test genome because more DNA from the test DNA hybridized to the probe for the region than reference DNA. Spots appearing red would indicate a lower copy number for that region in the test genome because less DNA from the test DNA hybridized to the probe for the region than reference DNA. Finally probes that appear black would indicate that no hybridization occurred for either sample in that region. Figure 2.2 illustrates a typical two-color microarray (Chen n.d.).

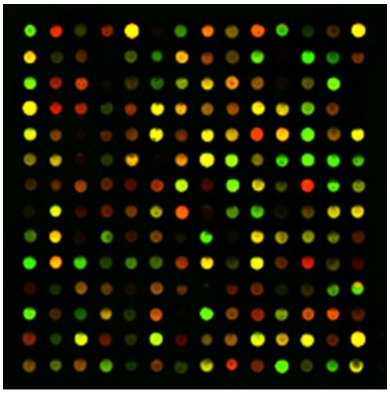

Figure 2.2: An example image of a microarray. The spots on this image show a range of fluorescent intensities between the test and reference sample.

2.1.4 Log_2 Ratios

A highly sensitive ultraviolet (UV) spectral scanner reads the intensities of each color on each spot on the microarray. Regions that show an increase in copy number by a factor of two will have a test/control ratio of 2. Genes that show a decrease in copy number by a factor of two will have a test/control ratio of 0.5. Because there is a difference in how the experiment treats gained versus lost regions on the microarray, intensities of green to red dyes are normalized with a base two logarithmic transformation (known as $\log_2$ ratio). This centers regions that have an equal test/control ratio of 1 to a $\log_2$ ratio of 0. Regions gained by a factor of two will have a $\log_2$ ratio of 1 and regions with a loss equivalent to a factor of two will have a $\log_2$ ratio of -1. In this thesis, when the value a $\log_2$ ratio threshold is mentioned, it is implied that it be negative for deletions and positive for duplications.

2.1.5 Normalization

Other normalization adjustments are made to the microarray data as a whole so that meaningful information can be found (Quackenbush 2002). In general normalization techniques consist of background correction, transformation (e.g. $\log_2$ ratio transformation described above), and rescaling. Some normalization techniques used on microarrays are: scale normalization, locally weighted scatterplot smoothing (LOWESS), quantile normalization, and variance stabilization and normalization (VSN). The LOWESS is a technique based on linear regression and is the most common type of normalization performed for two color microarrays (Beissbarth, et al. 2005). The underlying normalization technique should not have any unexpected influence on the algorithms developed for this thesis.

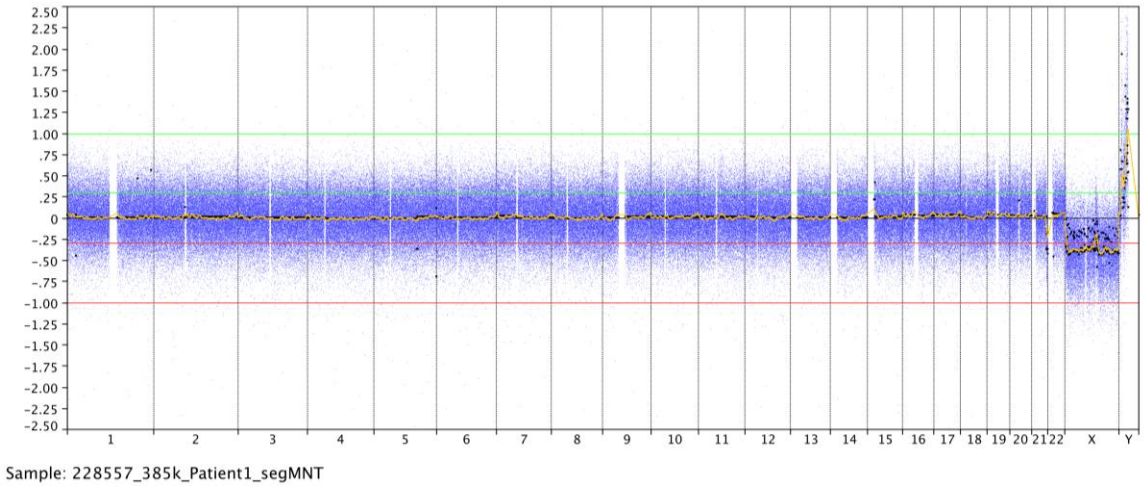

Figure 2.3: A genome wide view of $\log_2$ data for a patient. The y-axis shows the normalized $\log_2$ ratio data and the x-axis shows genomic position grouped by chromosome in increasing order. From this image it can be determined that the test sample used was a male and the reference sample used was a female. This can be seen in the X and Y chromosomes where the entire X chromosome appears to exhibit a decrease in copy number and the entire Y chromosome appears to exhibit a increase in copy number. The green and red lines indicate low and high copy number variations. This image was generated with Nexus Copy Number™ software licensed by Shivanand R. Patil Cytogenetics and Molecular Laboratory at the University of Iowa.

2.1.6 Copy Number Detection

There are many CNV calling algorithms that can be used to detect copy number variations in aCGH data (Venkatraman and Olshen 2007; Marioni, Thorne and Tavaré 2006; Zhang and Gerstein 2010). Many of these algorithms are specific to a certain type of microarray. The specific algorithm used for testing in this research was the FASST algorithm, which is part of the BioDiscovery Nexus Copy Number™ software. The FASST algorithm uses a Hidden Markov Model (HMM) approach that estimates possible segment levels that may occur between expected states (BioDiscovery 2011). The segments whose mean $\log_2$ ratios exceed a specified threshold are called as copy number variations.

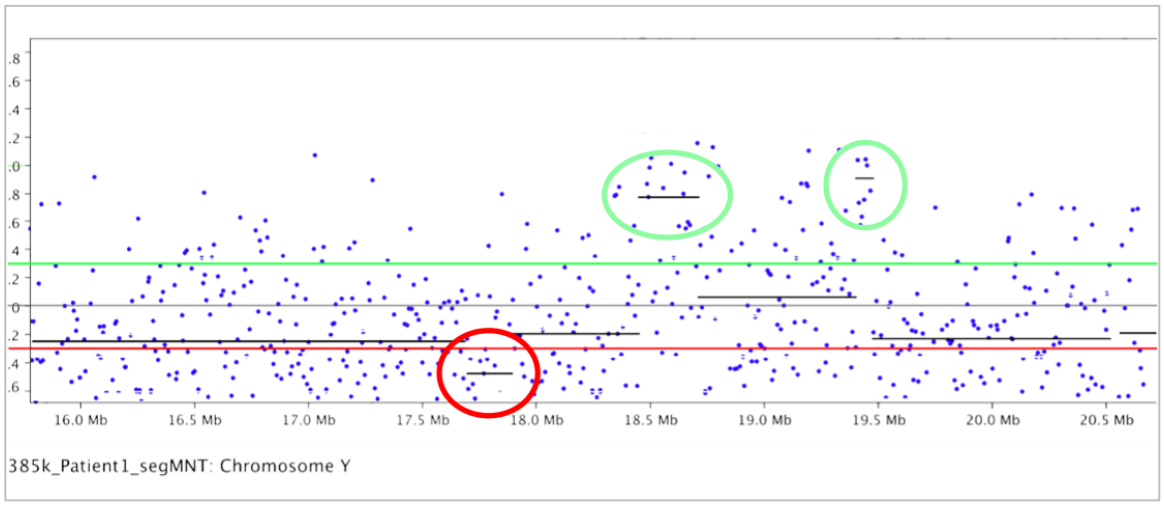

Figure 2.4: A diagram showing how the FASST algorithm could analyze aCGH data. The data are segmented and then segments classified as CNVs if they exceed a defined threshold. In this diagram the red and green lines signify deletion and duplication threshold respectively. Circled segments indicate segments that would be classified as deletions or duplications based on the shown threshold. Segments shown here have been fabricated to provide a visual demonstration of the aCGH CNV calling process.

2.2 Previous Work

This work builds upon a previously developed CNV-calling algorithm -- CGHTool. Finding small novel CNVs in aCGH data is challenging. The goal of CGHTool was to provide an algorithm that could detect small CNVs with high confidence. The CGHTool program creates a background file consisting of aCGH data from many patients (several hundred). From this it calculates an average and standard deviation of the data for each oligonucleotide probe on the CGH microarray. In general, a region of the genome can be classified (i.e., called) a CNVs if the number in the region exceed a multiple of the standard deviation in the same direction (gain or loss). While working on this algorithm, it was found that there was a need to calibrate several of the algorithms parameter. Among these was one essential metric: the base two logarithmically transformed signaling intensity ratio; hereafter referred to simplay as the $\log_2$ ratio. This ratio is taken between the measured intensity in a patient (unknown) sample and a control sample.

There have been several publications that have sought to prove the effectiveness of chromosomal microarrays and provide a strategy for clinical validation. Each publication has emphasized the need for validation of aCGH methods (Shaffer, et al. 2007; Shen and Wu, 2009; Yu, et al. 2009). It has been hypothesized that performance of specificity will increase when more consecutive probes are used when analyzing CNVs. There are a large number of benign and unknown CNVs that may contribute to poorer performance of microarrays. It is hypothesized that the majority of unknown CNVs are benign and as more known benign CNVs are discovered, arrays may be redesigned to improve performance (Shen and Wu, 2009). Currently, there has been no method provided that calculates true or false negatives using aCGH data. Since the calculations

of sensitivity and specificity rely on true and false negatives, these metrics are required to adhere to the guidelines set out by the ACMG.

2.2.1 Fluorescence in Situ Hybridization (FISH) Validation

Fluorescence in situ hybridization (FISH) is a technique used to detect DNA sequences. This technique first involves creating a fluorescent DNA probe compliment to a sequence of interest. The probe and target DNA are denatured and then combined so the probe may hybridize to its complement on the target DNA. This allows for visualization of the target sequence on the chromosome (O'Connor 2008). FISH can be used as an alternative technology for finding CNVs. Verification of aCGH can be done with FISH also, but it is a labor intensive and costly process as FISH can only interrogate one chromosomal region at a time (Balif, et al. 2007). A computational-based method that could provide validation of aCGH data could be very useful.

2.3 Receiver Operating Characteristic (ROC)

Receiver operating characteristic (ROC) curves were developed in World War II to improve signal to noise detection in radar signals (Krzanowski and Hand 2009). Since World War II, ROC analysis has become commonly used for evaluation of clinical diagnostic tests (Zou, O'Mally and Mauri 2007). A ROC analysis measures the costs and benefits of the changing prediction parameters using sensitivity and specificity. The approach taken in this research was to perform an analytical validation of aCGH data using a ROC-based method. A receiver operating characteristic or ROC is the plot of true positive rate (sensitivity) versus false positive rate (1- specificity) for a changing

parameter. The false positive rate (FPR) is plotted along the x-axis while sensitivity is plotted along the y-axis. Each axis is on a scale of 0 to 1.

2.3.1 Sensitivity and Specificity

Sensitivity and specificity are statistical measures of a binary classifier's performance. To calculate the sensitivity and specificity the classifier must be compared to a gold standard set. In a binary classifier, there are four outcomes possible. True positives (TP) occur when a result classified as positive matches the gold standard set. False positives (FP) occur when a result classified as positive does not match the gold standard set. True negatives (TN) occur when a result classified as negative matches the gold standard set. False negatives (FN) occur when a result classified as negative does not match the gold standard set. Even if a classifier is not binary (e.g. duplication, deletion, or no variation), the classifier can be made to emulate a binary classifier. This can be done by combining results (e.g. deletions or duplications) and classifying them as being positive and combining all other results (e.g. no variation) and classifying them as being negative.

The sensitivity is a measure from 0 to 1 of how well a classifier can identify positive results. The optimal measure for this statistic would be 1, meaning that no results that are negative are classified as positive by the gold standard set. The calculation of sensitivity using the outcome of classifier results compared to a gold standard set is shown below in Equation 2.1.

Equation 2.1: Measure of sensitivity for a classifier

$$Sensitivity = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

The specificity is also a measure from 0 to 1, but measures how well a classifier can identify negative results. The optimal measure for this statistic would also be 1, meaning that no results that are positive are classified as negative by the gold standard set. The false positive rate is a measure of 1- specificity giving it an optimal value of 0. The calculation of specificity using the outcome of classifier results compared to a gold standard set is shown below in Equation 2.2.

Equation 2.2: Measure of specificity for a classifier

$$Specificity = \frac{True\ Negatives}{True\ Negatives + \ False\ Positives}$$

2.3.2 Confidence Intervals

A confidence interval is used to determine the reliability of an estimate. Confidence intervals can be used with performance metrics such as sensitivity and specificity. For this research the confidence intervals used were calculated according to the efficient-score method (Newcombe 1998). When calculating a confidence interval, the efficient-score method accounts for the bounds of sensitivity (0 to 1) and FPR (0 to 1).

2.3.3 Receiver Operator Characteristic Plot

A ROC plot is created by running several tests with a classifier while varying a parameter used in a classifier (e.g. $\log_2$ ratio threshold). Each of the tests results in a point on the plot of FPR versus sensitivity. The points form a curve that allows visualization of the trade-offs between FPR and sensitivity for the changing classifier

parameters. For a ROC curve the optimal point would be (0, 1) with all results classified correctly. This outcome is rarely the case so an optimal point can be chosen based on desired results of classifier sensitivity and specificity. A general ROC plot is shown in Figure 2.5 (de Vet, et al. 2007).

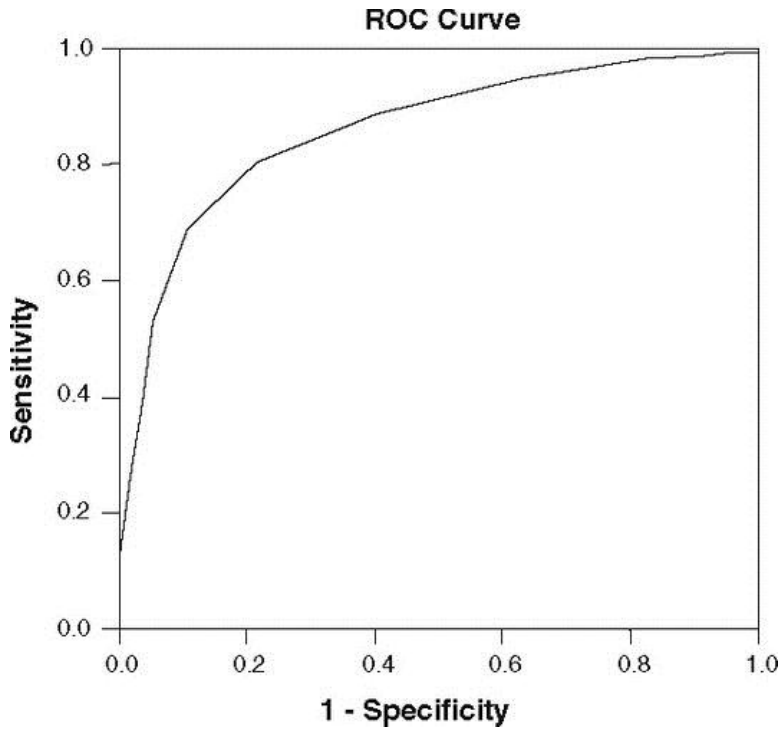

Figure 2.5: A plot of a receiver operating characteristic curve.

CHAPTER 3

APPROACH

The goal of the research for this thesis was to use a computer-aided approach to appropriately validate the performance characteristics of a whole-genome chromosomal microarray. To do this, two microarray platform designs were used that differed primarily in their level of resolution (i.e., number of unique probes), where the higher resolution CNVs were used as a "gold standard" set. This is similar to using a higher resolution photograph to validate a lower resolution photograph. With greater detail comes the ability to recognize patterns with more confidence. The two main platforms used in the validation were the NimbleGen 720,000 (720K array) and the NimbleGen 385,000 (385K array) microarrays. An additional validation was performed comparing the 385K array to a NimbleGen 2,100,000 (2.1M array) microarray in which similar results were found. Results of the analyses can be found in Appendix A. All analyses were performed using CNVs greater than 100Kb and CNVs greater than 400Kb. The minimum size of 100Kb was estimated to be the smallest CNV size that could be confidentially detected using a 385K array. The 400Kb minimum CNV size cutoff was used to adhere to the guidelines of the ACMG.

3.1

3.2 Validation Samples

Twenty samples were used for validation comparisons. All cases were previously analyzed and interpreted by standard clinical criteria. The set of samples included both

normal (no known pathogenic CNVs) and abnormal cases in which a known pathogenic lesion was identified (i.e. 17q12 or 22q11.2 deletion). An appropriate mix of male and female patients, as well as small (100-400Kb) and large (>400Kb) duplications and deletions were present. The CNVs present in these cases covered a significant portion of the genome (Table 3.1). Additionally, each case contained between 10 and 40 common, polymorphic CNVs.

Table 3.1: Genome wide aCGH probe coverage of CNV calls represented as probes, base pairs, percentage of probes, and percentage of genome covered.

Type of CNV	Total Number of Probes Contained in CNVs		% of Total Probes on the Array Contained in CNVs		Total Number of Base Pairs Covered by Probes Contained in CNVs	% of Genome Covered by Probes Contained in CNVs	
	Per Patient	Total	Per Patient	Total		Per Patient	Total
385K (Calls > 100Kb)	472	9,442	0.12236%	2.44729%	81,475,048	0.12730%	2.54610%
385K (Calls > 400Kb)	210	4,209	0.05455%	1.09094%	37,969,898	0.05933%	1.18656%
720K (All CNV Calls)	799	15,970	0.11095%	2.21901%	95,958,945	0.14994%	2.99872%

Note: Numbers shown are for CNV calls for a combination of all 20 samples. CNV calls used in the analysis were made using the FASST algorithm in Nexus Copy Number with a $\log_2$ ratio of 0.30 for both the 385K and 720K arrays.

3.3 Manual CNV-Based Validation

A manual CNV-based validation was performed using the FASST algorithm in Nexus Copy Number™ to call CNVs on two different array platforms, a higher resolution NimbleGen 720K array and a lower resolution NimbleGen 385K array. A $\log_2$ ratio threshold of 0.3 was used for calling CNVs for both platforms. The higher resolution arrays were used as a gold standard set for the lower resolution arrays. CNVs on the lower resolution arrays were compared to the higher resolution arrays to assess

false positives, true positives, false negatives, and true negatives. When comparing arrays, true negatives were defined as those intervening regions between CNVs of greater than 100Kb on the higher resolution array that had no corresponding CNV regions on the lower resolution array. Analyses were performed using a CNV size threshold of 100Kb and 400Kb as well as for both benign and clinically significant CNVs together and for clinically significant CNVs alone.

Aside from the laborious nature of this type of manual analysis it is clear that with a per-CNV based analysis and this few cases there are not enough CNVs to truly establish precise and meaningful sensitivity and specificity measures. When examining only clinically significant CNVs greater than 400Kb only18 CNVs were available for analysis. Because of the extensive time and effort involved in performing this manual comparison only one $\log_2$ ratio threshold could be examined for each array type precluding any analysis that might identify a different $\log_2$ ratio threshold that would further increase sensitivity without a significant loss in specificity (or vice versa) via ROC analysis. It was clear that a computer-aided approach was needed to appropriately validate the performance characteristics of a whole-genome chromosomal microarray for varying $\log_2$ ratios.

3.4 Analysis of Array Metrics

As part of our computer-aided validation, we sought to determine which array metric(s) would be most appropriate on which to base future ROC analysis. We did this by creating a (Perl) software tool to compare the CNV calls from the lower resolution arrays to the CNV calls from the higher resolution arrays. All CNV calls were produced by using the FASST algorithm from Nexus Copy Number™. The higher resolution arrays were again treated as a "gold standard" to which the lower resolution array data

were then compared. A previously FISH-validated threshold $\log_2$ ratio value of 0.3 was used for the FASST algorithm when calling the CNVs for both array platforms. The analysis, like the manual validation, was performed twice for all CNV calls of 100Kb or greater and all CNV calls of 400Kb or greater. For each experiment, every CNV called from the lower resolution arrays was checked for any overlapping CNV calls from the higher resolution arrays. From this, two sets of lower resolution CNVs could be made for each size-cutoff experiment, one set that had one or more overlapping higher resolution CNVs (true positives) and one set that had no overlapping higher resolution CNVs (false positives).

The CNVs within each set were examined by looking at the $\log_2$ ratios, CNV sizes in base pairs, and number of probes in each CNV. The median and mean values of each of these CNV metrics were calculated. By comparing these metrics in the two sets of CNVs for each experiment, we were able to perform an unpaired t-test for groups of unequal size and variance to determine if the metrics for the two sets differed significantly. We could then choose metrics that were statistically significant for use in our validation techniques.

3.5 Computer-Aided Probe-Based Validation

Once a significant parameter was found, the next goal was to find the optimal $\log_2$ ratio that could be used to call CNVs and maximize both sensitivity and specificity. This optimal $\log_2$ ratio could be calculated from a ROC analysis, which plots the sensitivity vs. false positive rate as the threshold of an experimental metric is varied. The $\log_2$ ratio threshold used to call CNVs in the aCGH data was found to be the most significant experimental metric. The results section gives justification for this decision.

Whereas the analysis could be performed on a per-CNV basis, as with the manual analysis, the number of CNVs would be the same and still lacking in ability to calculate meaningful performance metrics. This led to the use of a novel per-probe approach that compared probes in CNV calls from the lower resolution array to calls from the higher resolution array and assigned the probes of the lower resolution array "truth-values" based on corresponding locations in the higher resolution array. The CNV calling algorithm was run with varying $\log_2$ ratio thresholds to create many different sets of CNV calls for both the lower resolution arrays and the higher resolution arrays.

3.5.1 CNV Validation Sizes

To make the validation applicable to the guidelines of the ACMG, all CNV calls used in the lower resolution arrays had to be greater than 400Kb. Similarly, only CNVs in the higher resolution array greater than 400Kb could have probes counted as false positives if no corresponding CNVs in the lower resolution array existed. As not to penalize the arrays for performing better than ACMG expectations, smaller CNVs called on both lower and higher resolution arrays could be used to validate corresponding larger (>400Kb) CNVs on the opposite array. This approach allowed probes in larger (>400Kb) CNVs to be validated as true positives if smaller CNVs were called in the same region, but would not penalize the arrays with false positives and false negatives.

3.5.2 Probe Validation Range

Because it is not necessarily the case that the probes in the lower resolution array are a subset of the probes in the higher resolution array, each probe in the lower resolution array was validated by probes in the higher resolution array within a certain

genomic distance. In tests performed on correlation between genomic locations of probes in the 385K array and 720K array, it was found that only 17% of the probes in the 385K array had a corresponding probe found in the exact same genomic location on the 720K array. For all tests performed using the NimbleGen 385K array the genomic distance in which higher resolution array probes could validate a 385K array probe was 15Kb. This distance was chosen because the median probe spacing for the 385K array was 7Kb to 8Kb. Because lower resolution array probes could be validated by a range of higher resolution array probes, all lower resolution array probes that did not have a higher resolution array probe with the genomic distance validation range were excluded from the analysis.

3.5.3 ROC Analysis

Because of the variability between CNV calls for different $\log_2$ ratio thresholds, in the ROC analysis a single higher resolution $\log_2$ ratio set of CNVs was used as a gold standard while each lower resolution $\log_2$ ratio set of CNVs was compared to the gold standard. This analysis was run for every set of higher resolution $\log_2$ ratio set of CNVs. A ROC plot was generated for every higher resolution $\log_2$ ratio gold standard used. Each ROC plot contained a comparison of each of the lower resolution $\log_2$ ratios to a single higher resolution $\log_2$ ratio gold standard set.

When performing each comparison in the ROC analysis all probes in the lower resolution arrays and higher resolution arrays were labeled as either a CNV (deletion or duplication) or normal (not a CNV) according to the $\log_2$ ratio thresholds being compared. For each $\log_2$ ratio threshold tested against the gold standard, all lower resolution array probes for each sample were assigned a truth-value. The truth-value for each lower resolution array probe was assigned as a true-positive or true-negative if one

of the higher resolution array probes in its coverage range (15Kb) had the same outcome (gain, loss, normal/no CNV), otherwise the lower resolution array probe was labeled as a false-positive or false-negative. A flow diagram of the analysis pipeline is shown in Figure 3.1.

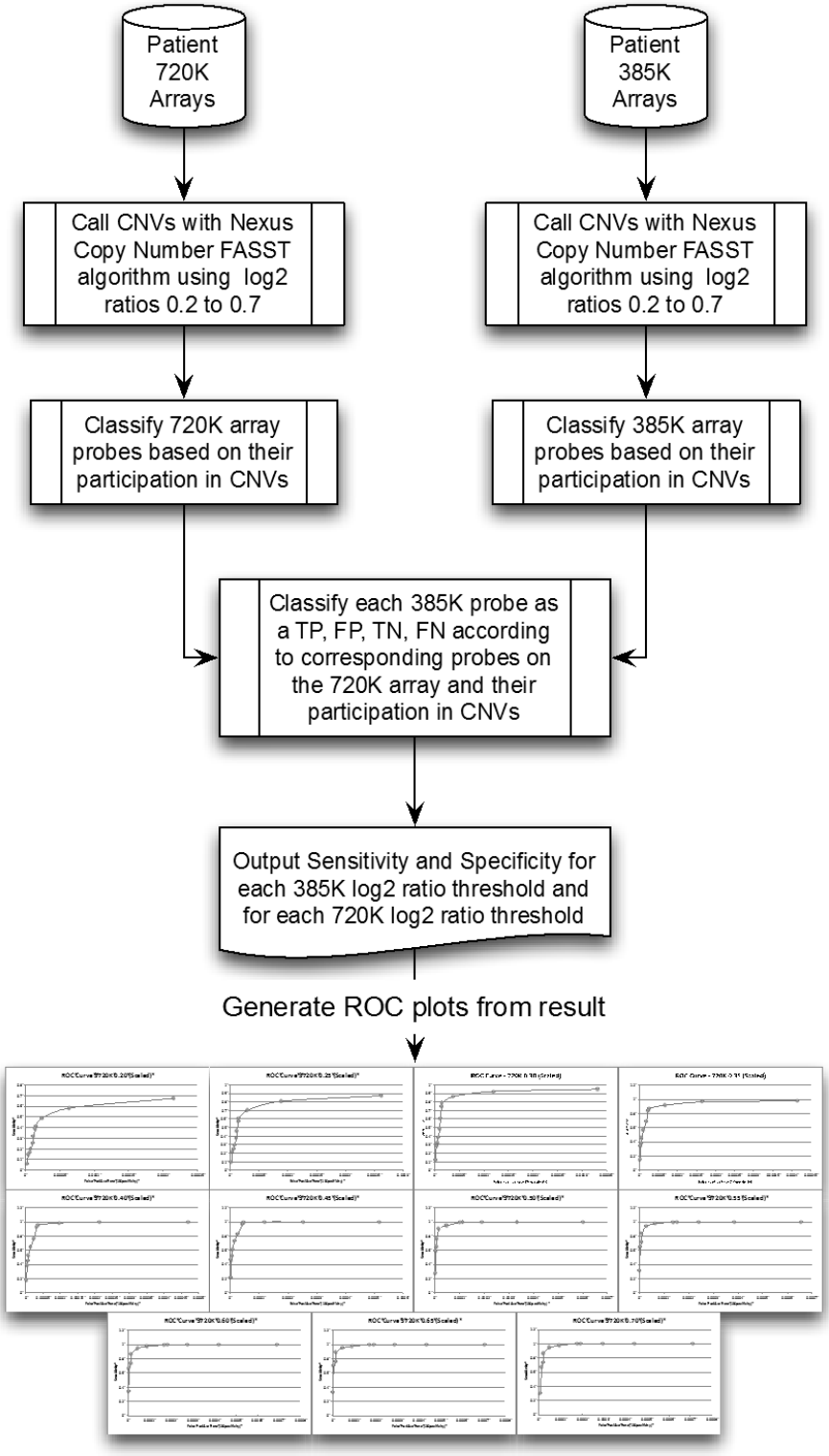

Figure 3.1: A Demonstration of a probe based ROC analysis performed using arrays of different resolution.

3.5.4 Variations of Analysis

In addition to varying the gold standard $\log_2$ ratio threshold used for the ROC analyses, the types of CNVs used in the analyses was also varied. There were three types of analyses performed: just for copy number gains, just for copy number losses, and for a combination of copy number gains and losses.

Another variation of the analysis allowed for exclusion of probes found in common, polymorphic regions. There were 158 of these common regions found where each region occurred in at least 5% of the patient population used in the dataset. Common CNVs were defined by their presence in HapMap samples as ascertained by Conrad and colleagues (Conrad, et al. 2010). These common regions contain clinically benign CNVs, so these regions were excluded in the hope of improving the performance of the analysis.

3.5.5 Optimal Experimental Metrics

Once ROC curves were created, they were utilized to determine the optimal $\log_2$ ratio threshold used for the experiment. The optimal points on each curve should provide the best trade off between sensitivity and false positive rate. Two methods for determining this metric were formed. The first method calculated the optimal point by using linear distance. A perfect classifier would perform with 100% specificity and a 0% false positive rate, or be a point at (0, 1) on the ROC plot. Using a linear method, the optimal point would be the closest point to (0, 1) on the plot. Because sensitivity and false positive rate did not always reach 1, the ROC plots were normalized to the maximum sensitivity and false positive rate before calculating the optimal $\log_2$ ratio (closest linear distance to (0, 1)). The second approach to calculate the optimal point was

to take the derivative of the ROC curve to find the maximum change in slope. This approach did not perform as consistently as our linear-distance approach because the rate of change for the curve was not consistent for all analyses. A demonstration of how the optimal point is calculated using a linear-distance approach can be seen in Figure 3.2 and Table 3.2.

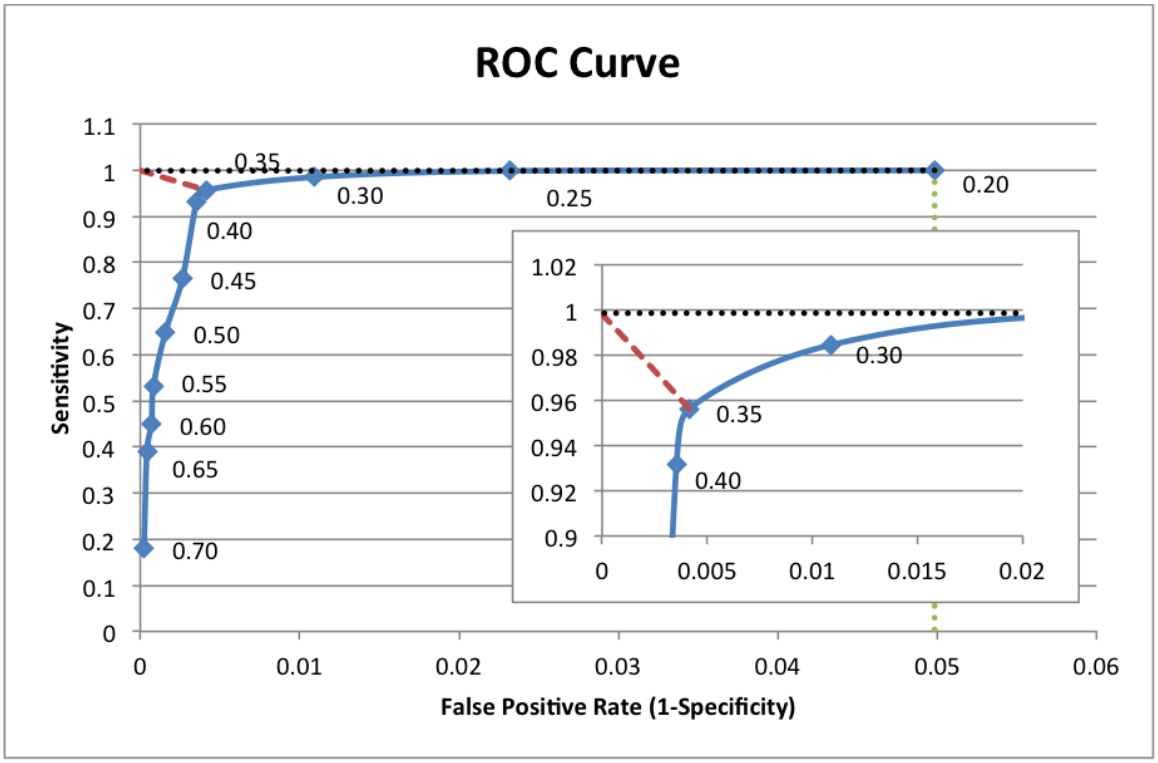

Figure 3.2: A demonstration of how the optimal point is determined based on a normalized ROC plot.

Table 3.2: An example of a computer-aided single log_2 ratio threshold comparison of two different array resolution designs.

720K Log_2 Ratio	385K Log_2 Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.04982	0.99864	1.00000
0.40	0.25	0.02321	0.99804	0.46592
0.40	0.30	0.01090	0.98462	0.21927
0.40	**0.35**	**0.00414**	**0.95636**	**0.09321**
0.40	0.40	0.00354	0.93152	0.09775
0.40	0.45	0.00265	0.76560	0.23933
0.40	0.50	0.00157	0.64768	0.35285
0.40	0.55	0.00082	0.53029	0.46928
0.40	0.60	0.00071	0.44948	0.55009
0.40	0.65	0.00042	0.39108	0.60845
0.40	0.70	0.00023	0.17975	0.82001

Note: Numbers correspond to the data points shown in Figure 3.2. The bolded row indicates the optimal log_2 ratio found based on the closest linear distance (greyed box).

Whereas this analysis was capable of elucidating the optimal log_2 ratio value to use to maximize sensitivity and specificity, the values obtained for specificity were clearly biased toward the very large number of true negative probes (non-CNV probes) present in the array comparisons. This resulted in specificities that routinely exceeded 99.999% across most 385K log_2 ratios thresholds tested. To better calculate the specificity, we took the number of TN probes and divided it by the average number of probes present in CNVs >400Kb. This was deemed appropriate given that a requirement for any true or false positive probe was inclusion within a CNV that was >400Kb.

3.5.6 Computational Analysis Design

A Perl script was developed to accurately compute the results of a ROC analysis given a paired set of CNV calls from raw microarray data. Using a duplicate set of 385K

array and 720K array data for 20 patients and 11 different $\log_2$ ratio thresholds, this could be performed on a typical desktop workstation in approximately 2 hours with an approximate memory footprint of 2GB. The Perl script developed (shown in Appendix B.1) allows a general input of aCGH probes genomic locations as well as a call type (copy gain/copy loss) paired with a general genomic start and stop position for CNVs. Because of this general input format performance of many different array platforms may be assessed. The Perl script also allows customizations described previously to be set using flags. For example the buffer range and minimum CNV size may both be customized. Other parameters such as inclusion or exclusion of common CNV regions may be set, as well as types of CNVs included in the analysis (duplications, deletions, or both).

A second Perl script was developed to combine the results of many higher resolution gold standard $\log_2$ ratios into one Microsoft Excel workbook with many worksheets. This script was created separately from the ROC analysis script because of it's dependencies on a 3rd-party Perl module Excel::Writer::XLSX (McNamara 2011). The script will read the standard output directory from the previous ROC analysis script and compile all the results into one Microsoft Excel workbook. Each analysis is shown in its own worksheet with optimal metrics calculated and a ROC plot generated. The script has options to adjust the false positive rate in way described previously.

3.6 Computer-Aided Region-Based Validation

It was thought that the FPR may be better represented if the ROC analysis were done for regions instead of probes, however, conducting region-based analyses proved challenging due to the incomplete classification of non-CNV regions with a truth value. The arrays could be quantified into CNV regions and non-CNV regions, but

precise classification of the regions proved difficult. For example, it is possible to classify a lower resolution CNV region based on the corresponding higher resolution regions. However, this was difficult due to the fact that not all regions were of a uniform size and regions from the different array types did not often correlate well with each other. The most intuitive way to classify these regions was to determine if a defined percentage of a given lower resolution region overlapped a similar higher resolution region.

One problem with this approach was that very large non-CNV (>> 400Kb) regions in a lower resolution array may have an overlapping higher resolution CNV region that is in total a small percentage of the lower resolution, non-CNV region. If the majority of a lower resolution, non-CNV region is overlapped by higher resolution non-CNV regions it would be classified as a true negative, despite the fact that there existed a clear false negative region. This is just one of the cases contributing to the difficulty of classifying CNV calls, especially true negatives, with truth-values as regions of the genome. Using a probe-based analysis provided a roughly uniform spacing and coverage between differing array types, which made it easier to assign truth-values to the probes.

CHAPTER 4

RESULTS

4.1 Manual Validation Results

Based on the results of the manual validation of twenty cases a sensitivity of 79.8% (with a lower limit of the 95% confidence interval of 74.4%) was achievable when using data from all CNVs (benign and clinically significant) greater than 100Kb and a single $\log_2$ ratio threshold of 0.3. For all CNVs greater than 400Kb a sensitivity of 80.9% (with a lower limit of the 95% confidence interval of 66.3%) and specificity of 99.2% (false positive rate of 0.8% with a higher limit of the 95% confidence interval of 2.2%) was achievable. When examining only those CNVs >400Kb and clinically significant (known pathogenic lesions and novel CNVs of unclear clinical significance) a sensitivity of 100% and a specificity of 100% (false positive rate of 0%) were achieved.

Even though the sensitivity and specificity were each 100%, the 95% confidence intervals were quite large (sensitivity with a lower limit of the 95% confidence interval of 78.1% and specificity with a lower limit of the 95% confidence interval of 99.0%). An example of the analysis performed for each case is represented in Table 4.1.

Table 4.1: Example case assessed by manual validation using two different array resolution designs.

CNV Location	Size (Kb) 385K	Size (Kb) 720K	Log_2 Ratio 385K	Log_2 Ratio 720K	Truth Value (>100Kb)	Truth Value (>400Kb)	CNV Interpretation
1p36.33	615	[264]	-0.37	[-0.28]	False Positive	False Positive	Benign
1p36.11	168	140	-0.552	-0.569	True Positive	Size Exclusion	Benign
1p21.1	177	112	-0.437	-0.388	True Positive	Size Exclusion	Benign
1p13.3	NA	73	NA	0.344	Size Exclusion	Size Exclusion	Benign
1q23.3	176	[162]	0.303	[0.22]	False Positive	Size Exclusion	Benign
1q44	NA	43	NA	0.708	Size Exclusion	Size Exclusion	Benign
2p11.2	517	214	0.337	0.346	True Positive	True Positive	Benign
2p11.2	139	78	-0.527	-0.339	True Positive	Size Exclusion	Benign
2q13	[640]	358	[0.28]	0.304	False Negative	Size Exclusion	Benign
6p25.3	[145]	97	[0.29]	0.331	Size Exclusion	Size Exclusion	Benign
6p22.1	NA	30	NA	0.442	Size Exclusion	Size Exclusion	VUCS
7p14.1	NA	40	NA	0.334	Size Exclusion	Size Exclusion	VUCS
7q35	153	NA	-0.302	NA	False Positive	Size Exclusion	Benign
8p23.2p23.1	517	520	0.466	0.497	True Positive	True Positive	VUCS
8p23.1	76	53	0.609	0.906	Size Exclusion	Size Exclusion	Benign
11p15.4	NA	95	NA	0.403	Size Exclusion	Size Exclusion	Benign
14q11.2	324	166	0.363	0.35	True Positive	Size Exclusion	Benign
14q11.2	120	NA	0.315	NA	False Positive	Size Exclusion	Benign
15q11.2	NA	439	NA	-0.421	False Negative	False Negative	Benign
16p13.3p13.2	545	546	-0.724	-0.84	True Positive	True Positive	VUCS
17q21.31	148	110	0.493	0.456	True Positive	Size Exclusion	Benign
17q21.31q21.32	124	139	-0.856	-1.015	True Positive	Size Exclusion	Benign
17q21.32	129	131	-0.672	-1.018	True Positive	Size Exclusion	Benign
19p13.3	216	NA	-0.322	NA	False Positive	Size Exclusion	Benign
22q11.21	237	187	-0.397	-0.462	True Positive	Size Exclusion	Benign
22q11.21	190	239	-0.419	-0.387	True Positive	Size Exclusion	Benign
Xq21.1q28	70741	70690	-0.721	-0.864	True Positive	True Positive	Abnormal

Note: Total True Negatives (>400Kb analysis) = 29. Values in brackets indicate an absence of a Nexus Copy Number CNV call but a CNV could be inferred through visual analysis of the region in question. The abnormality that was present in this array was the Xq21.1q28 deletion of approximately 70Mb in size. NA = Not Applicable due to Size Exclusion or inability to make a CNV call either by Nexus Copy Number or visual inspection. VUCS = Variant of Unclear Clinical Significance.

4.2 Significant CNV Metrics

In the 385K CNV set, for 20 samples there were a total of 45 CNVs that were greater than 400Kb and a total of 283 CNVs greater than 100Kb (Shown in Table 4.2). In the 400Kb CNV size cutoff analysis, CNVs with overlapping 720K CNVs also had a higher $\log_2$ ratio than those that had no overlapping 720K CNVs. In the 100Kb CNV size cutoff analysis, CNVs with overlapping 720K CNVs had a higher mean $\log_2$ ratio value (0.542 vs. 0.506) and a higher median $\log_2$ value (0.477 vs. 0.452) than the 385K CNVs with no overlapping 720K CNVs. An unpaired t-test was performed on the differing groups of $\log_2$ ratio means and found that the difference in $\log_2$ ratio means between the sets was statistically significantly different for the 400Kb analysis ($p = 6.43 \times 10^{-11}$), but not significant for the 100Kb analysis ($p = 0.275$). Subdividing these CNVs into deletions and duplications yielded statistically similar results for the 100Kb. Statistical tests could not be performed for duplications and deletions separately in the 400Kb CNV size cutoff group because of small numbers of CNVs. From these data we concluded the most appropriate threshold metric for future ROC analysis was the $\log_2$ ratio value and that our overall approach of using a higher resolution array to validate a lower resolution array was appropriate. The conclusions were based on the observation that CNVs with overlap (true positives) were represented by a significant amount of probes ($p = 0.0455$) that was greater than those CNVs without overlap (false positives). Thus, a CNV called with more probes had a higher probability of being confirmed with another array design

Table 4.2: Computer-aided single $\log_2$ ratio threshold comparison of two different array resolution designs.

Type of CNV	Total Number of All CNVs	Number of Common CNVs	Number of Unique CNVs
> 100Kb 385K CNVs with 720K Overlap	211	107	104
> 100Kb 385K CNVs without 720K Overlap	72	32	40
> 400Kb 385K CNVs with 720K Overlap	43	24	19
> 400Kb 385K CNVs without 720K Overlap	2	0	2

4.3 ROC Analysis Results

Before performing the ROC analysis, the Nexus Copy Number™ FASST algorithm was run with varying $\log_2$ ratio thresholds varying between 0.20 and 0.70 in increments of 0.05 to create 11 sets of CNV calls for both the 385K and 720K arrays. As described in the Approach section, a single 720K $\log_2$ threshold was used as a gold standard for each ROC analysis, while 385K arrays for each $\log_2$ ratio threshold value were analyzed against it. This analysis was re-run for every 720K $\log_2$ ratio threshold value. There were 82 probes in the 385K array that had poor coverage (no 720K array probes within 15Kb) that were discarded from the experiment.

From the ROC curves an optimal value for the $\log_2$ ratio threshold was determined using normalized curves with a low FPR. The optimal $\log_2$ ratio thresholds for the 385K array fell between 0.30 and 0.50 across all 720K array $\log_2$ ratio threshold conditions. Optimal $\log_2$ ratio thresholds for deletions and duplications were similar but not exactly the same (Table 4.3).

Table 4.3: Optimal log_2 ratio threshold determined for the 385K array at every 720K array log_2 ratio threshold value including probes in common regions.

720K Gold Standard Log_2 Ratio	All CNV Types	Deletions	Duplications
0.20	0.30	0.30	0.25
0.25	0.30	0.30	0.25
0.30	0.30	0.35	0.30
0.35	0.35	0.40	0.30
0.40	0.35	0.40	0.35
0.45	0.40	0.45	0.35
0.50	0.45	0.50	0.40
0.55	0.50	0.50	0.45
0.60	0.50	0.50	0.50
0.65	0.50	0.50	0.65
0.70	0.50	0.50	0.65

Note: The optimal log_2 ratio was determined for deletions, duplications, and a combination of both deletions and duplications. These optimal log_2 ratios were calculated from analyses in which probes in common CNV regions were included. Overall, larger log_2 ratio thresholds are more appropriate for duplications than for deletions. The larger range for duplications is due to smaller numbers of CNV duplications.

Once the performance metrics were accurately calculated, the most appropriate log_2 ratio threshold for the 385K array platform that would meet the ACMG's guidelines of >99% sensitivity and <1% FPR was determined. While performing this analysis two cases were looked at, one with all CNVs taken together as well as one where CNVs that were be presumed to have clinical significance in constitutional/congenital genetic disorders. If all CNVs are included in the analysis the ACMG's guidelines with the 385K array can not be met. However, when only those probes from CNVs that are considered clinically significant are included, it was determined that the best log_2 ratio threshold to call duplications is 0.35, which produces a sensitivity of 99.7% with a lower limit of the 95% confidence interval of 99.2% and a specificity of 99.9% (false positive rate of 0.06% with a higher limit at the 95% confidence interval of 0.08%). Furthermore, it was

determined that the best log_2 ratio threshold to call deletions is 0.40, which produces a sensitivity of 99.8% with a lower limit at the 95% confidence interval of 99.3% and a specificity of nearly 100% (99.97% specificity with a false positive rate of 0.03% with a higher limit at the 95% confidence interval of 0.05%). These performance metrics exceed those recommended by the ACMG. Table 4.4 shows optimal log_2 ratios determined for analyses performed that excluded probes from common CNV regions.

Table 4.4: Optimal log_2 ratio threshold determined for the 385K array at every 720K array log_2 ratio threshold value excluding probes in common regions.

720K Gold Standard Log_2 Ratio	All CNV Types	Deletions	Duplications
0.20	0.30	0.30	0.25
0.25	0.30	0.40	0.25
0.30	0.30	0.40	0.30
0.35	0.35	0.40	0.35
0.40	0.35	0.40	0.35
0.45	0.40	0.45	0.35
0.50	0.50	0.45	0.40
0.55	0.50	0.45	0.50
0.60	0.50	0.45	0.50
0.65	0.50	0.45	0.50
0.70	0.50	0.50	0.50

Note: These optimal log_2 ratios were calculated from analyses in which probes in common CNV regions were excluded.

In addition to the comparison of the 385K arrays to the 720K arrays using 100Kb and 400Kb CNV size cutoffs, an analysis was performed comparing the 385K arrays to the 2.1M arrays using a 400Kb CNV size cutoff. None of the 385K array data used in the experiment corresponded to the data used in the 385K array versus 720K array experiment. In this analysis, there were 91 probes in the 385K array that had poor coverage (no 2.1M array probes within 15Kb) that were discarded from the experiment.

The results of this analysis did not perform as well as the comparison of the 385K arrays to the 720K arrays in analysis both including and excluding common CNV probes. This was due to the conditions in which the replicate arrays were created.

A sample of figures and tables for these analyses is shown in Appendix A. Due to the large number of figures for the analyses (66 in all) only a subset of the figures are shown for a single $\log_2$ ratio gold standard of 0.40. Analysis for deletions, duplications, and a combination of both are shown on separate figures, while analyses including and excluding probes in common CNV regions are shown on the same figure.

CHAPTER 5

CONCLUSIONS

In this thesis it has been shown that clinical laboratories with access to two different resolution chromosomal microarray designs can perform a cost-effective analytical validation and calibration to establish the performance characteristics of their particular platform in their laboratory setting. Whereas we chose to perform this validation on NimbleGen whole-genome tiled CGH arrays, the methodology used is robust and should be applicable to other oligonucleotide-based designs.

The analytical procedure has been written so that it can be utilized on a typical desktop PC. When comparing 720K arrays and 385K arrays, the procedure uses approximately 2 GB of memory and takes approximately 2 hours to complete using a 2.5 GHz Core2 Duo processor. The run time and memory usage of the procedure is highly dependent on the resolution of the microarray data and the algorithm parameters used to call CNVs and may increase greatly with the use of higher resolution chromosomal microarrays and lower algorithm thresholds resulting in more CNV calls.

Several technical problems are present when trying to perform an analytical validation of either the CMA technology or a specific platform. To validate the technology, a different test methodology should be used to confirm or refute the finding of the microarray, ideally at a resolution at which CMA possesses its greatest utility. Readily available technologies include FISH, quantitative PCR (qPCR), and multiplex ligation-dependent probe amplification (MLPA). The shortcoming of these approaches is that none of them is a genome wide screen. Comparison of microarray CNV findings with these technologies provides for identification of true and false positives but is generally lacking in true and false negatives. Without the later values, sensitivity and

specificity cannot strictly be calculated. These values could be calculated, for a limited, specific set of genetic loci but this is not an economical way to assess the genome-wide coverage of CMA. Conventional chromosome analysis (karyotyping) interrogates the entire genome, and as a comparison technology could be used to validate a whole genome chromosomal microarray. However, the resolution of chromosome analysis is such that only lesions of >5Mb can be consistently and reliably validated. The method described here allows for calculation of true and false positives (as is done in most FISH, qPCR, or MLPA validations), as well as true and false negatives across the entire genome.

Previous publications have sought to prove the efficacy of chromosomal microarrays and provide a road map for clinical validation (Shaffer, et al. 2007; Shen and Wu, 2009; Yu, et al. 2009). Whereas each of these publications illustrates the importance of validation none provide a demonstrable method for the calculation of true and false negative values from CMA data. As both sensitivity and false positive rates rely on the ability to calculate true and false positives, as well as true and false negatives, these two additional metrics are necessary to adhere to the ACMG's guidelines.

This method is not without limitations. The gold standard used for the validation is another CMA design. The analysis of probe numbers in true and false positive CNVs suggests that this comparison is valid, yet does not address what is the most appropriate $\log_2$ ratio value to use with the higher resolution array. To compensate for this, the 385K array was tested against higher resolution arrays at various $\log_2$ ratio thresholds and calculated the most appropriate $\log_2$ ratio threshold to use for the 385K array at every higher resolution array $\log_2$ ratio threshold. This provided us with a range for the optimum $\log_2$ ratio threshold for the 385K array design. This range, which was different for duplications and deletions (duplication range: 0.25 – 0.65, deletion range: 0.30 – 0.50), allowed us to determine what the optimal $\log_2$ ratio threshold should be when using

the 385K array design to achieve the ACMG's recommendations for both sensitivity and specificity for simultaneous testing for both deletions and duplications.

We initially thought that the false positive rate might be better represented if the ROC analysis were performed for regions instead of probes, however conducting region-based analyses proved challenging due to the incomplete classification of non-CNV regions with a truth-value. The arrays could be quantified into CNV regions and non-CNV regions, but precise classification of the regions proved difficult. We found that a probe-based analysis provided roughly uniform spacing and coverage between differing array types making it easier to assign truth-values to the probes.

Our validation design also has a very important difference from that suggested by the ACMG: the ability to include common, polymorphic CNVs. By including both unique and common CNVs we greatly increase the total number of CNV-related probes available for evaluation. An additional advantage of this approach is that following this type of validation the microarray platform under consideration has been appropriately calibrated to detect a wide range of polymorphic CNVs in addition to those that are currently sought after in constitutional genetic conditions. Whereas polymorphic CNVs are generally not considered significant in terms of developmental disorders, they have been shown in several publications to convey quantifiable risk of common diseases (Craddock, et al. 2010; Gamazon, Nicolae and Cox 2011). We observed that the sensitivity of our array platform is lower when considering all CNVs than when considering just those that are unique and likely clinically significant. We believe the explanation for this is that most common CNVs are part of segmental duplications and are very polymorphic in the general population. This makes the choice of reference DNA very important. We chose to use reference DNA that is a combination of genomic DNA from several individuals of the same sex. It is likely that within this mix of individuals there are different genotypes at several segmental duplication loci. This results in

"blunted" $\log_2$ ratio values of patient to reference signal at probes representing these loci. These lower ratios are often right at the $\log_2$ ratio threshold and are occasionally not detected on one or the other platform. Given that we are dealing with higher and lower resolution platforms, it is more common for these instances to be missed on the lower resolution platform and be classified as false negatives, thus lowering the sensitivity. All the false negatives we identified were within highly polymorphic regions of the genome and considered to be clinically benign.

As of the date of writing this thesis, only one major chromosomal microarray manufacturer (Affymetrix) has provided a statement that they have performed an analytical validation that meets or exceeds the ACMG recommendations. For clinical laboratories using microarrays from manufacturers that have yet to comply with the ACMG recommendations, as well as for laboratories to perform their own internal validation, our approach provides an alternative method to produce the same quality control and performance metrics required of any CLIA-certified laboratory.

Future Work

Since the algorithm analyzes the data on a probe-by-probe basis it can be further optimized by examining the data on a CNV region based approach. A CNV region type approach could compare the CNV regions as they pertain to probes on the microarray. The amount probe overlap of lower resolution array CNVs by higher resolution array CNVs could be quantified to true positives, while the rest of the probes for the lower resolution array CNVs would be deemed false positives. The higher resolution array CNVs could be checked for overlapping lower resolution array CNVs. The false negatives could be quantified as the number of probes in the higher resolution array CNVs minus the amount of overlap that the higher resolution array CNVs had with the

lower resolution array CNVs. Finally the true negatives would be quantified as the total number of probes for all arrays minus the first three metrics calculated TP, FP, and FN. This is shown in equation 5.1.

Equation 5.1: Equation showing the sum of all binary classifier outcomes.

$$Total = True\ Positives + False\ Positives + True\ Negatives + False\ Positives$$

This type of approach would greatly reduce the system memory usage as well as improve the total run time. Since computations performed could be exponentially reduced, it is possible that with this approach the run time could be reduced to minutes rather than hours.

REFERENCES

Balif, B C, et al. "The clinical utility of enhanced subtelomeric coverage in array CGH." *Am J Med Genet* 143A, no. 16 (August 2007): 1850-7.

Beissbarth, Tim, Markus Ruschhaupt, David Jackson, Chris Lawerenz, and Ulrich Mansmann. "Recommendations for normalization of microarray data." 2005, 4.

BioDiscovery. "Nexus Copy Number User Manual Version 6." 2011.

Chen, Jeff. http://nim.vbi.vt.edu/~jeff/ImmArray/ (accessed June 2012).

Conrad, D F, D Pinto, R Redon, and et al. "Origins and functional impact of copy number variation in the human genome." *Nature* 464, no. 7289 (Apr 2010): 704-712.

Coulter, M E, D T Miller, D J Harris, and et al. "Chromosomal microarray testing influences medical management." *Genet Med* 13, no. 9 (Sep 2011): 770-776.

Craddock, N, M E Hurles, N Cardin, and et al. "Genome-wide association study of CNVs in 16,000 cases of eight common diseases and 3,000 shared controls." *Nature* 464, no. 7289 (Apr 2010): 713-720.

de Vet, Henrica C. W., et al. "Minimally important change determined by a visual method integrating an anchor-based and a distribution-based approach." *Quality of Life Research* 16 (2007): 131-142.

Gamazon, E R, D L Nicolae, and N J Cox. "A study of CNVs as trait-associated polymorphisms and as expression quantitative trait loci." *PLoS Genet* 7, no. 2 (2011): e1001292.

Hochstenback, R, E van Binsbergen, J Engelen, and et al. "Array analysis and karyotyping: workflow consequences based on a retrospective study of 36,325 patients with idiopathic developmental delay in the Netherlands." *Eur J Med Genet* 52, no. 4 (Jul-Aug 2009): 161-169.

Kearney, H M, E C Thorland, K K Brown, F Quintero-Rivera, and S T South. "American College of Medical Genetics recommendations for the design and performance expectations for clinical genomic copy number microarrays intended for use in the postnatal setting for detection of constitutional abnormalities." *Genet Med* 13, no. 7 (July 2011): 680-685.

Kearney, H M, S T South, D J Wolff, A Lamb, A Hamosh, and K W Rao. "American College of Medical Genetics recommendations for the design and performance expectations for clinical genomic copy number microarrays intended for use in the postnatal setting for detection of constitutional abnormalities." *Genet Med* 13, no. 7 (July 2011): 676-679.

Krzanowski, Wojtek J, and David J Hand. *ROC Curves for Continuous Datapublis.* Boca Raton, Florida: CRC Press, 2009.

Manning, M, and L Hudgins. "Array-based technology and recommendations for utilization in medical genetics practice for detection of chromosomal abnormalities." *Genet Med* 12, no. 11 (Nov 2010): 742-745.

Marioni, J C, N P Thorne, and S Tavaré. "BioHMM: a heterogeneous hidden Markov model for segmenting array CGH data." *Bioinformatics* 22, no. 9 (May 2006): 1144-6.

McNamara, John. *CPAN.* Jan 3, 2011. http://search.cpan.org/~jmcnamara/Excel-Writer-XLSX-0.05/.

Miller, D T, M P Adam, S Aradhya, and et al. "Consensus statement: chromosomal microarray is a first-tier clinical diagnostic test for individuals with developmental disabilities or congenital anomalies." *Am J Hum Genet* 86, no. 5 (May 2010): 749-764.

Newcombe, R G. "Two-sided confidence intervals for the single proportion: comparison of seven methods." *Stat Med* 17, no. 8 (Apr 1998): 857-872.

O'Connor, Claire C. "Fluorescence In Situ Hybridization (FISH)." *Nature Education* 1, no. 1 (2008).

Pinto, Dalila, et al. "Comprehensive assessment of array-based platforms and calling algorithms for detection of copy number variants." *Nature Biotechnology* 29, no. 6 (June 2011): 512-520.

Quackenbush, J. "Microarray data normalization and transformation." *Nature Genetics* 32, no. 5 (Dec 2002): 496-501.

Roche NimbleGen. *Human CGH Whole-Genome Tiling Arrays.* Roche NimbleGen. http://www.nimblegen.com/products/cgh/wgt/human/index.html (accessed June 2012).

Shaffer, L G, A L Beaudet, A R Brothman, and et al. "Microarray analysis for constitutional cytogenetic abnormalities." *Genet Med* 9, no. 9 (Sep 2007): 654-662.

Shen, Y, and B L Wu. "Microarray-based genomic DNA profiling technologies in clinical molecular diagnostics." *Clin Chem* 55, no. 4 (Apr 2009): 659-669.

Shen, Y, K A Dies, I A Holm, and et al. "Clinical genetic testing for patients with autism spectrum disorders." *Pediatrics* 125, no. 4 (Apr 2010): e727-735.

Theisen, Aaron. "Microarray-based Comparative Genomic Hybridization (aCGH)." *Nature Education* 1, no. 1 (2008).

Venkatraman, E S, and Adam B Olshen. "faster circular binary segmentation algorithm for the analysis of array CGH data." *Bioinformatics* 23, no. 6 (March 2007): 657-663.

Vissers, L E, B B de Vries, and J A Veltman. "Genomic microarrays in mental retardation: from copy number variation to gene, from research to diagnosis." *J Med Genet* 47, no. 5 (May 2010): 289-297.

Yu, S, D C Bittel, N Kibiryeva, D L Zwick, and L D Cooley. "Validation of the Agilent 244K oligonucleotide array-based comparative genomic hybridization platform for clinical cytogenetic diagnosis." *Am J Clin Pathol* 132, no. 3 (Sep 2009): 349-360.

Zhang, Zhengdong D, and Mark B Gerstein. "Detection of copy number variation from array intensity and sequencing read depth using a stepwise Bayesian model." *BMC Bioinformatics* 11 (October 2010): 539.

Zou, Kelly H, James O'Mally, and Laura Mauri. "Statistical Primer for Cardiovascular Research." *Circulation* 115 (2007): 654-657.

APPENDIX A

COMPUTATIONAL ANALYSIS FIGURES AND TABLES

A.1 Analysis of 385K Versus 720K (400Kb size cutoff)

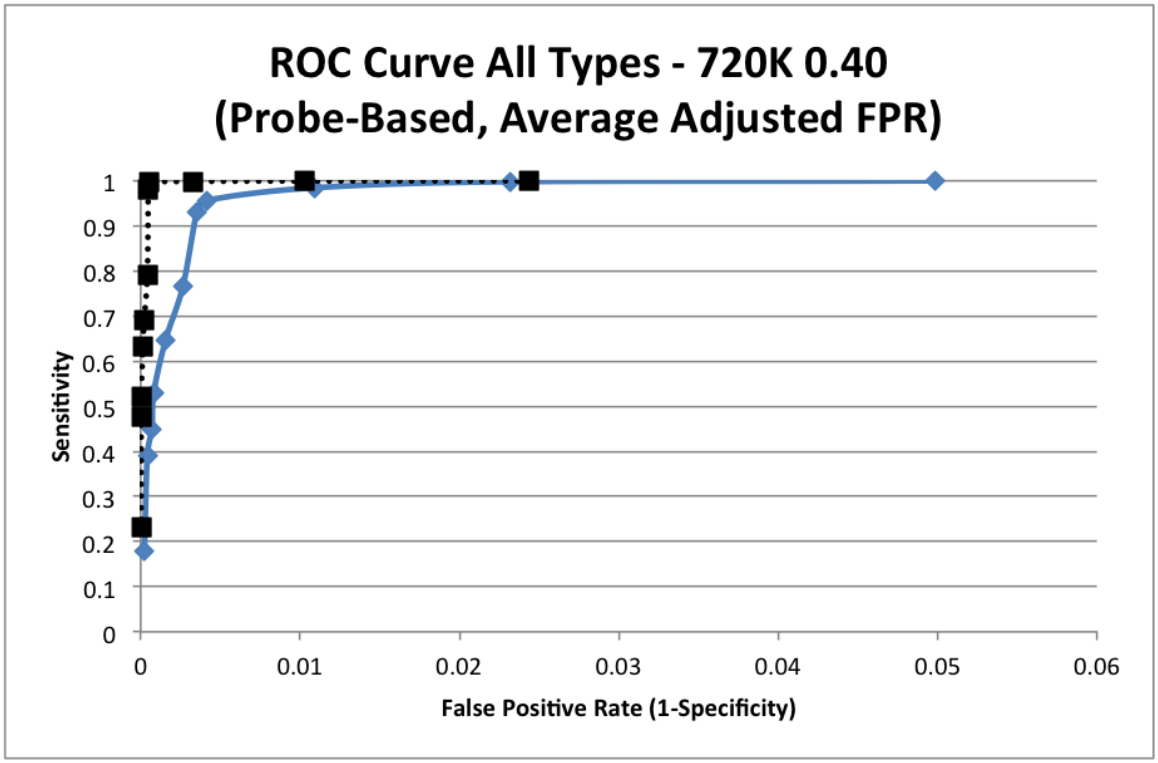

Figure A.1: ROC plot of analysis comparing all 385K CNVs (deletions and duplications) greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. The solid line shows the analysis performed using all probes and the dotted line shows the analysis performed excluding probes from common CNV regions.

Table A.1: A table containing analysis results comparing all 385K CNVs (deletions and duplications) greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were included.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.035016867	0.998640196	1
0.40	0.25	0.016180212	0.998039216	0.462069491
0.40	0.30	0.007570203	0.984619935	0.216642703
0.40	**0.35**	**0.002866955**	**0.956357928**	**0.092173428**
0.40	0.40	0.002449858	0.931515536	0.097019123
0.40	0.45	0.001833528	0.765604681	0.239155274
0.40	0.50	0.001086957	0.647682119	0.352804159
0.40	0.55	0.000568692	0.530287648	0.469271392
0.40	0.60	0.00048891	0.449477352	0.550087834
0.40	0.65	0.000289412	0.39107703	0.608446594
0.40	0.70	0.000159692	0.179754373	0.820013545

Note: Numbers correspond to the data points shown in Figure A.1. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

Table A.2: A table containing analysis results comparing all 385K CNVs (deletions and duplications) greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were excluded.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.016996157	1	1
0.40	0.25	0.007131854	0.999540863	0.419615948
0.40	0.30	0.002261443	0.998565966	0.133063859
0.40	**0.35**	**0.00038264**	**0.996510469**	**0.022782168**
0.40	0.40	0.000332306	0.979889392	0.028048372
0.40	0.45	0.000292038	0.792424242	0.208285711
0.40	0.50	0.000151076	0.691521961	0.308606078
0.40	0.55	0.000100721	0.633914422	0.366133541
0.40	0.60	2.01457E-05	0.521857923	0.478143546
0.40	0.65	2.01457E-05	0.477880939	0.522120406
0.40	0.70	2.01452E-05	0.230585424	0.769415489

Note: Numbers correspond to the data points shown in Figure A.1. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

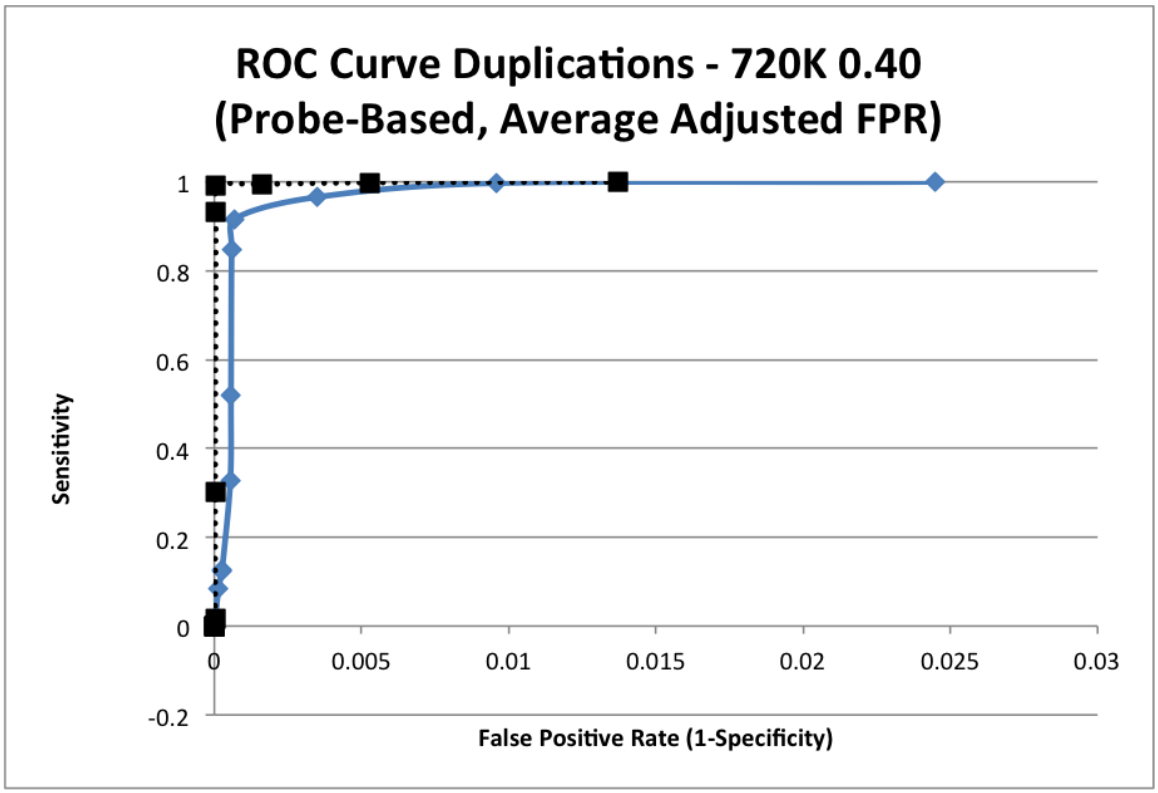

Figure A.2: ROC plot of analysis comparing only 385K CNV duplications greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. The solid line shows the analysis performed using all probes and the dotted line shows the analysis performed excluding probes from common CNV regions.

Table A.3: A table containing analysis results comparing all 385K CNVs duplications greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were included.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.021158649	0.9986053	1
0.40	0.25	0.008268968	0.997037037	0.390811076
0.40	0.30	0.002998191	0.965546218	0.145516266
0.40	**0.35**	**0.000580341**	**0.915913201**	**0.087231853**
0.40	0.40	0.000507835	0.847272727	0.153432805
0.40	0.45	0.000487112	0.520183486	0.479642822
0.40	0.50	0.000466394	0.325966851	0.673938463
0.40	0.55	0.000228069	0.127071823	0.872817263
0.40	0.60	0.000228069	0.124309392	0.875583342
0.40	0.65	0.000124414	0.084714549	0.915186025
0.40	0.70	0	0	1

Note: Numbers correspond to the data points shown in Figure A.2. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

Table A.4: A table containing analysis results comparing all 385K duplications greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were excluded.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.011841112	1	1
0.40	0.25	0.004562548	0.998491704	0.385317064
0.40	0.30	0.001379339	0.994809689	0.116602876
0.40	**0.35**	**4.18533E-05**	**0.992606285**	**0.008195134**
0.40	0.40	4.18533E-05	0.931098696	0.068991904
0.40	0.45	2.09269E-05	0.303030303	0.696971938
0.40	0.50	2.09269E-05	0.015267176	0.98473441
0.40	0.55	0	0	1
0.40	0.60	0	0	1
0.40	0.65	0	0	1
0.40	0.70	0	0	1

Note: Numbers correspond to the data points shown in Figure A.2. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

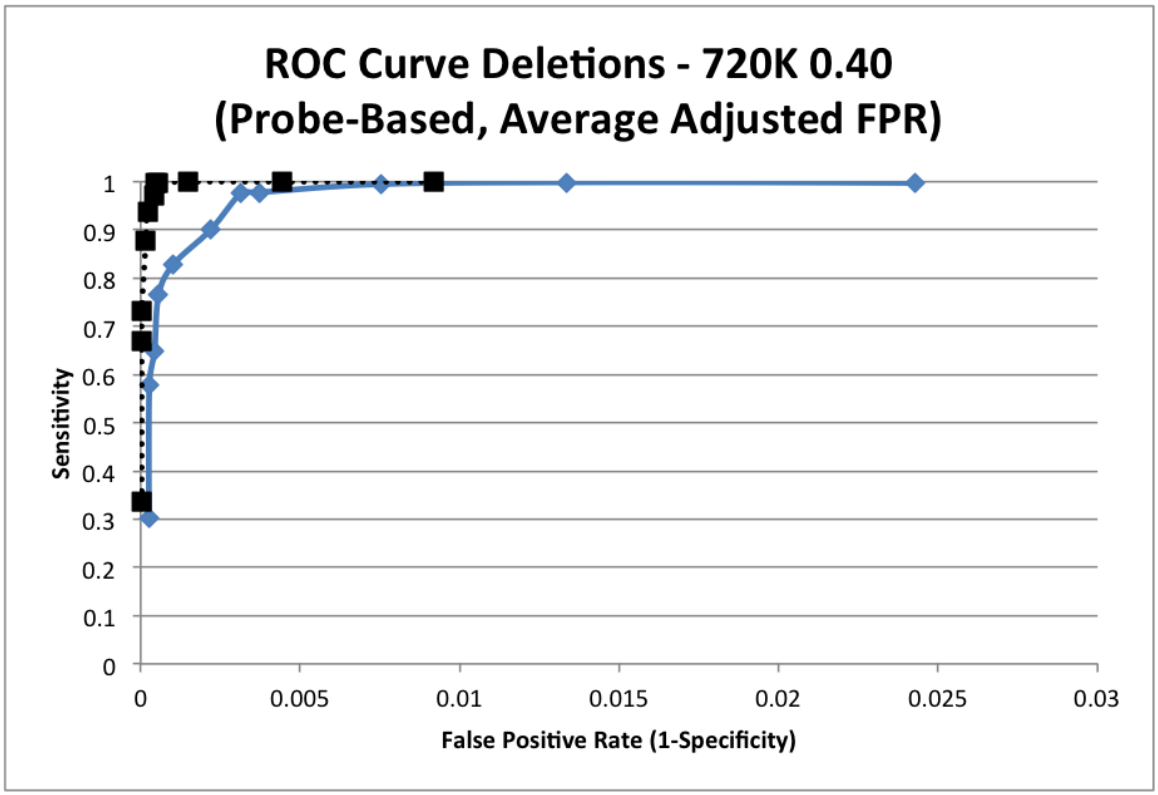

Figure A.3: ROC plot of analysis comparing all 385K CNV deletions greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. The solid line shows the analysis performed using all probes and the dotted line shows the analysis performed excluding probes from common CNV regions.

Table A.5: A table containing analysis results comparing all 385K CNVs deletions greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were included.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.00610576	0.906613757	1
0.40	0.25	0.002425657	0.824661247	0.407427755
0.40	0.30	0.000940622	0.753401833	0.228673868
0.40	**0.35**	**0.000539015**	**0.679071069**	**0.266053919**
0.40	0.40	0.000566407	0.640112202	0.308242931
0.40	0.45	0.000383765	0.575357845	0.370743741
0.40	0.50	0.000347229	0.495769882	0.456717478
0.40	0.55	0.000319822	0.432797243	0.525240616
0.40	0.60	0.000246735	0.358446145	0.605980809
0.40	0.65	0.000155365	0.32008245	0.6474475
0.40	0.70	0.000146224	0.150280025	0.834583975

Note: Numbers correspond to the data points shown in Figure A.3. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

Table A.6: A table containing analysis results comparing all 385K deletions greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were excluded.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.005497415	1	1
0.40	0.25	0.002655081	1	0.482968973
0.40	0.30	0.000899455	1	0.163614086
0.40	0.35	0.000329017	0.997952218	0.059884498
0.40	**0.40**	**0.000280643**	**0.997933884**	**0.051091851**
0.40	0.45	0.000261294	0.970385675	0.056001216
0.40	0.50	0.000125825	0.938633194	0.065496167
0.40	0.55	9.67904E-05	0.876552228	0.124697003
0.40	0.60	1.93594E-05	0.731240429	0.268782642
0.40	0.65	1.93594E-05	0.669472073	0.330546686
0.40	0.70	1.9359E-05	0.335652174	0.664357159

Note: Numbers correspond to the data points shown in Figure A.3. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

A.2 Analysis of 385K Versus 720K (100Kb size cutoff)

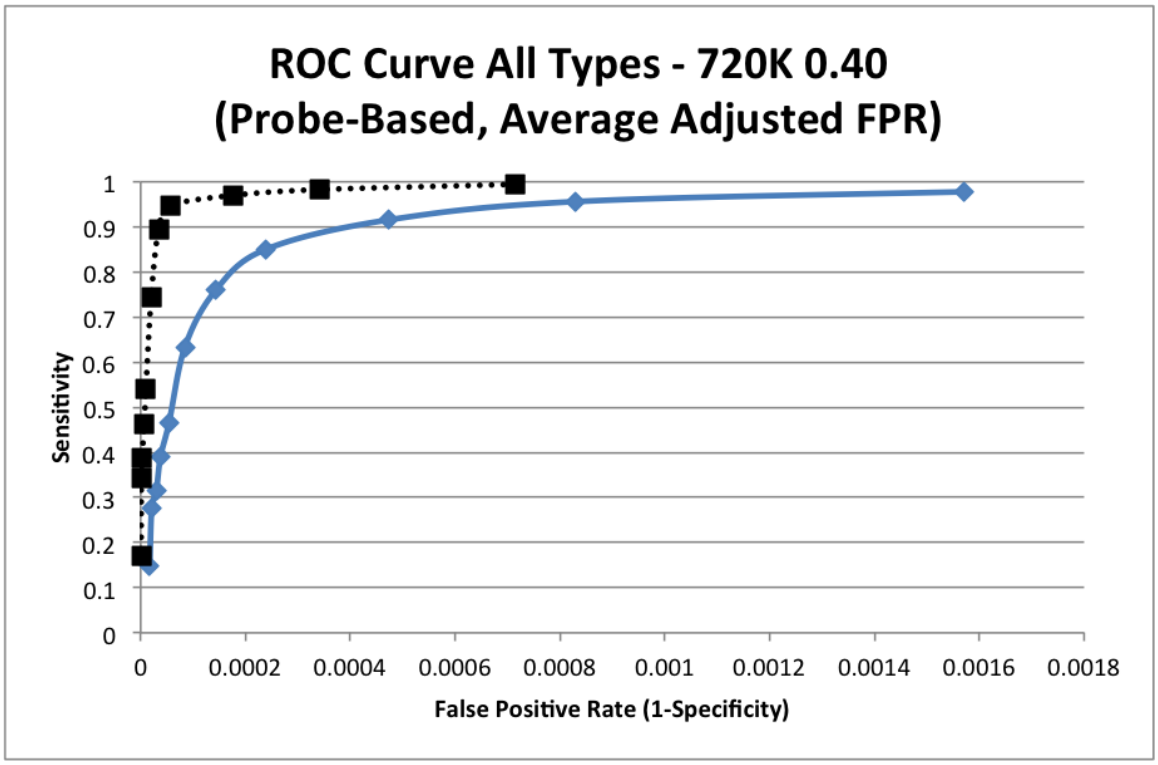

Figure A.4: ROC plot of analysis comparing all 385K CNVs (deletions and duplications) greater than 100Kb to a 720K gold standard $\log_2$ threshold of 0.40. The solid line shows the analysis performed using all probes and the dotted line shows the analysis performed excluding probes from common CNV regions.

Table A.7: A table containing analysis results comparing all 385K CNVs (deletions and duplications) greater than 100Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were included.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.001570593	0.978944247	1
0.40	0.25	0.000830048	0.956085132	0.529008939
0.40	0.30	0.000472095	0.915377475	0.307517581
0.40	**0.35**	**0.000238831**	**0.850191449**	**0.201051013**
0.40	0.40	0.000143086	0.759376037	0.242087164
0.40	0.45	8.44499E-05	0.633854955	0.356588895
0.40	0.50	5.42238E-05	0.465916837	0.525197905
0.40	0.55	3.78786E-05	0.390423889	0.60166219
0.40	0.60	2.99655E-05	0.316125691	0.677343636
0.40	0.65	2.11444E-05	0.275038921	0.719171388
0.40	0.70	1.73824E-05	0.149188623	0.847674786

Note: Numbers correspond to the data points shown in Figure A.4. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

Table A.8: A table containing analysis results comparing all 385K CNVs (deletions and duplications) greater than 100Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were excluded.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.000714305	0.993914807	0.455056827
0.40	0.25	0.000340923	0.982753581	0.21710135
0.40	0.30	0.000176022	0.969832736	0.112459378
0.40	**0.35**	**5.74521E-05**	**0.946553295**	**0.049324226**
0.40	0.40	3.42876E-05	0.893316599	0.090152559
0.40	0.45	2.01536E-05	0.745210728	0.239105373
0.40	0.50	1.00768E-05	0.542071197	0.446315709
0.40	0.55	6.93593E-06	0.462540717	0.527529149
0.40	0.60	2.48646E-06	0.387647832	0.604016468
0.40	0.65	1.17779E-06	0.34376029	0.648846329
0.40	0.70	9.16056E-07	0.171859296	0.824444451

Note: Numbers correspond to the data points shown in Figure A.4. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

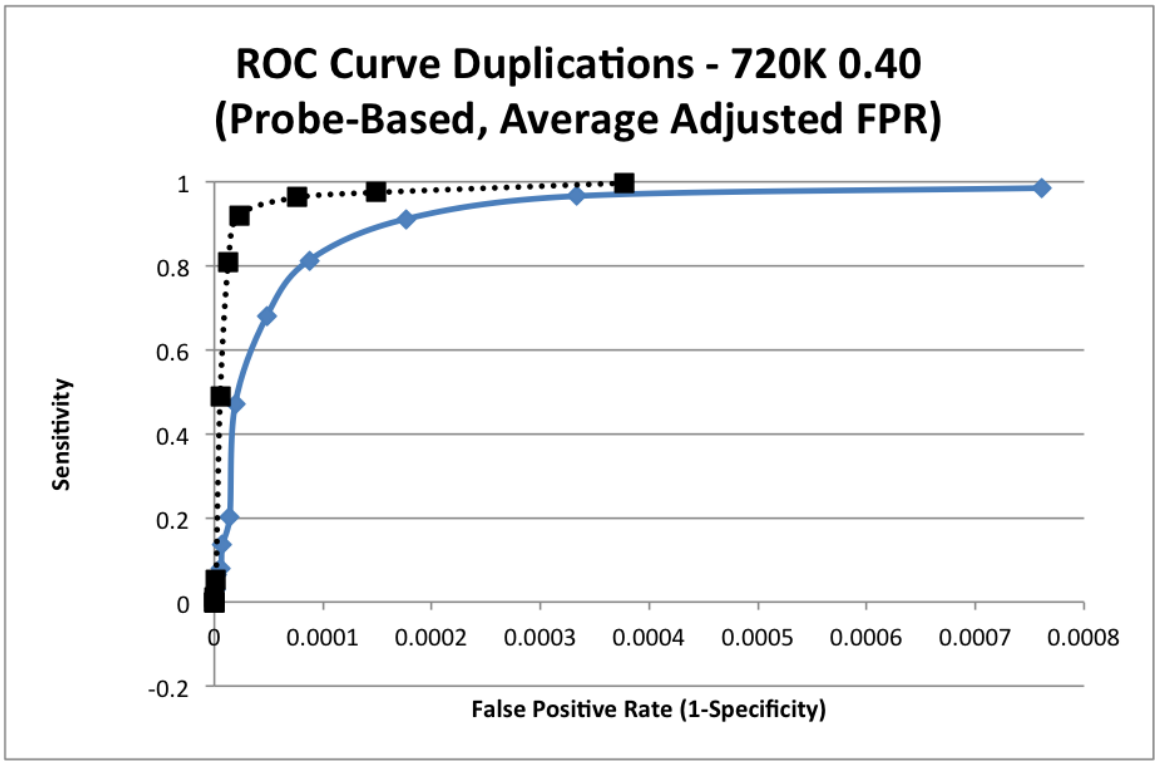

Figure A.5: ROC plot of analysis comparing only 385K CNV duplications greater than 100Kb to a 720K gold standard $\log_2$ threshold of 0.40. The solid line shows the analysis performed using all probes and the dotted line shows the analysis performed excluding probes from common CNV regions.

Table A.9: A table containing analysis results comparing all 385K CNVs duplications greater than 100Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were included.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.000760659	0.985650837	1
0.40	0.25	0.000333386	0.966031196	0.438737832
0.40	0.30	0.00017674	0.911764706	0.244144646
0.40	**0.35**	**8.71373E-05**	**0.813125695**	**0.209190661**
0.40	0.40	4.82358E-05	0.680457053	0.316063565
0.40	0.45	1.91904E-05	0.471463023	0.522283079
0.40	0.50	1.41334E-05	0.20261973	0.794647775
0.40	0.55	7.39086E-06	0.135858378	0.862218537
0.40	0.60	6.09421E-06	0.081270627	0.917581208
0.40	0.65	2.59328E-06	0.066006601	0.9330387
0.40	0.70	7.77984E-07	0.027640264	0.971957886

Note: Numbers correspond to the data points shown in Figure A.5. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

Table A.10: A table containing analysis results comparing all 385K duplications greater than 100Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were excluded.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.000376818	0.996642042	0.495508766
0.40	0.25	0.000149287	0.974692202	0.196575197
0.40	0.30	7.61479E-05	0.962728551	0.102773624
0.40	**0.35**	**2.25041E-05**	**0.919623461**	**0.073230758**
0.40	0.40	1.28219E-05	0.807573416	0.181454522
0.40	0.45	5.36425E-06	0.487980769	0.504964424
0.40	0.50	1.57002E-06	0.051894563	0.947352201
0.40	0.55	1.30835E-07	0.010752688	0.989090789
0.40	0.60	0	0	1
0.40	0.65	0	0	1
0.40	0.70	0	0	1

Note: Numbers correspond to the data points shown in Figure A.5. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

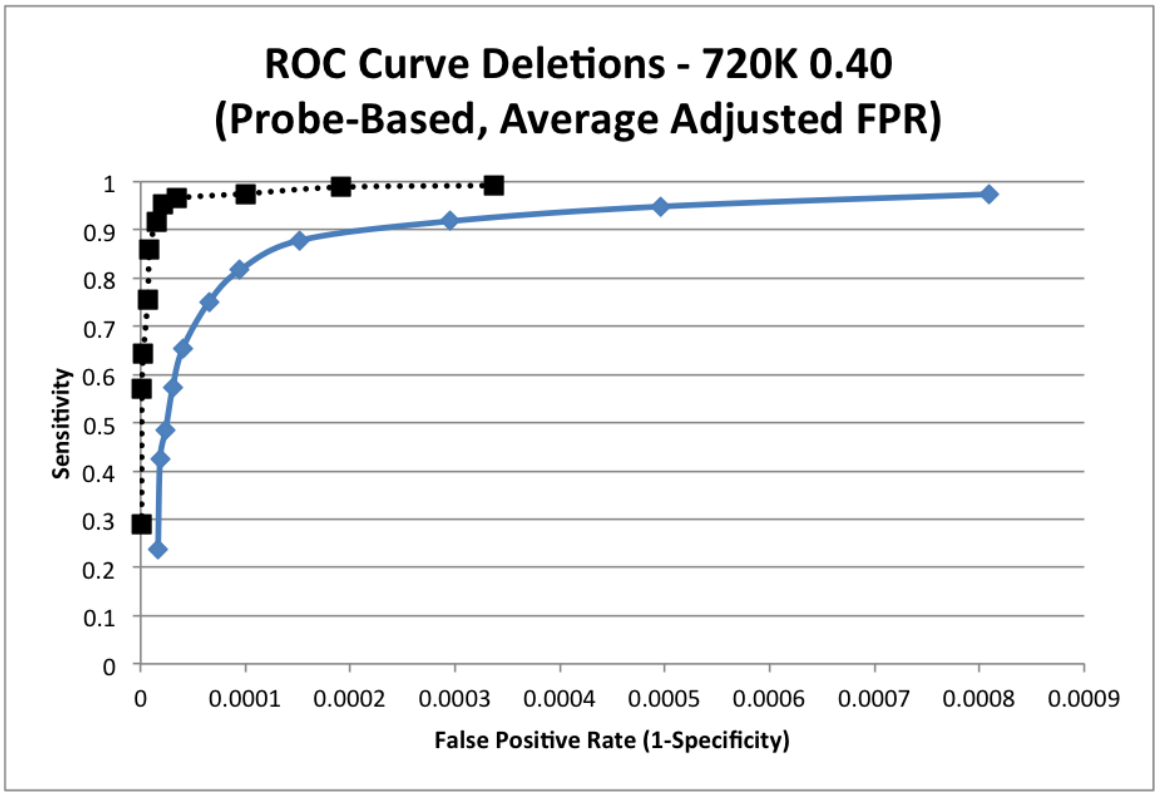

Figure A.6: ROC plot of analysis comparing all 385K CNV deletions greater than 100Kb to a 720K gold standard $\log_2$ threshold of 0.40. The solid line shows the analysis performed using all probes and the dotted line shows the analysis performed excluding probes from common CNV regions.

Table A.11: A table containing analysis results comparing all 385K CNVs deletions greater than 100Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were included.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.00080925	0.973801415	1
0.40	0.25	0.000496312	0.948508054	0.613848659
0.40	0.30	0.000295163	0.918117519	0.369192091
0.40	0.35	0.0001516	0.878185382	0.211506284
0.40	**0.40**	**9.47975E-05**	**0.816800459**	**0.199288215**
0.40	0.45	6.52298E-05	0.750795948	0.24277669
0.40	0.50	4.00714E-05	0.653722628	0.332398888
0.40	0.55	3.04749E-05	0.572395527	0.413921692
0.40	0.60	2.38611E-05	0.485154394	0.502658834
0.40	0.65	1.85442E-05	0.425975573	0.563030747
0.40	0.70	1.65989E-05	0.238282431	0.755585415

Note: Numbers correspond to the data points shown in Figure A.6. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

Table A.12: A table containing analysis results comparing all 385K deletions greater than 100Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were excluded.

720K $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.000337325	0.991845056	0.417248357
0.40	0.25	0.000191561	0.988769781	0.237212556
0.40	0.30	9.98363E-05	0.975077882	0.123375828
0.40	0.35	3.49361E-05	0.966004184	0.043907198
0.40	**0.40**	**2.14588E-05**	**0.951928156**	**0.034751631**
0.40	0.45	1.47856E-05	0.915605096	0.062492543
0.40	0.50	8.505E-06	0.859275053	0.118076167
0.40	0.55	6.80399E-06	0.756045137	0.223772673
0.40	0.60	2.48606E-06	0.64235166	0.340380749
0.40	0.65	1.17761E-06	0.570491803	0.414162574
0.40	0.70	9.15911E-07	0.288526434	0.703712129

Note: Numbers correspond to the data points shown in Figure A.6. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

A.3 Analysis of 385K Versus 2.1M (400Kb size cutoff)

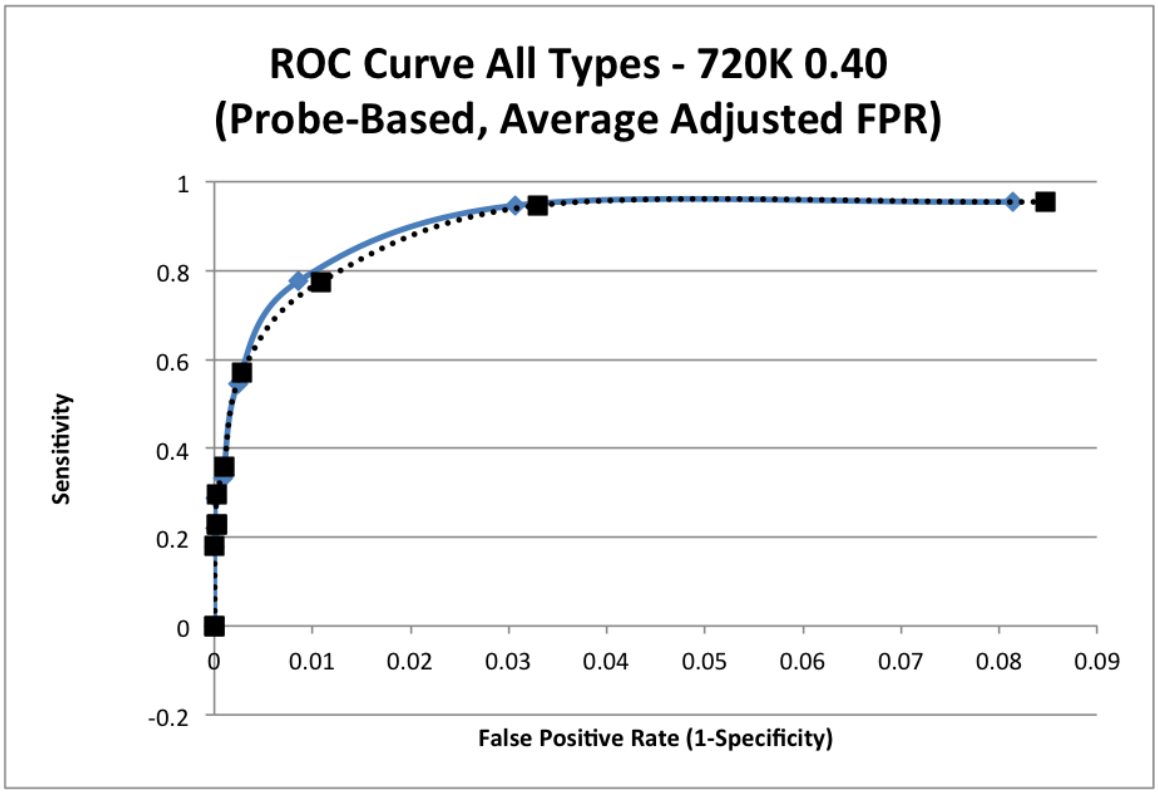

Figure A.7: ROC plot of analysis comparing all 385K CNVs (deletions and duplications) greater than 400Kb to a 2.1M gold standard $\log_2$ threshold of 0.40. The solid line shows the analysis performed using all probes and the dotted line shows the analysis performed excluding probes from common CNV regions.

Table A.13: A table containing analysis results comparing all 385K CNVs (deletions and duplications), greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were included.

2.1M $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.066232877	0.980582524	1
0.40	0.25	0.030079197	0.97810219	0.454150038
0.40	**0.30**	**0.012695772**	**0.89073051**	**0.212459361**
0.40	0.35	0.005159974	0.767844268	0.230514899
0.40	0.40	0.001608963	0.609715243	0.37899054
0.40	0.45	0.000512486	0.522790698	0.466921128
0.40	0.50	0.000153799	0.458891013	0.532027103
0.40	0.55	0.000146476	0.329174664	0.664310707
0.40	0.60	5.85956E-05	0.256689792	0.738227772
0.40	0.65	3.66228E-05	0.159585492	0.837254582
0.40	0.70	0	0.048245614	0.950799027

Note: Numbers correspond to the data points shown in Figure A.7. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

Table A.14: A table containing analysis results comparing all 385K CNVs (deletions and duplications), greater than 400Kb to a 720K gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were excluded.

2.1M $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.04483012	0.95359116	1
0.40	0.25	0.016972437	0.946496815	0.37866759
0.40	**0.30**	**0.005513895**	**0.774151436**	**0.224803858**
0.40	0.35	0.001431988	0.571428571	0.402032432
0.40	0.40	0.000495006	0.357963875	0.624712529
0.40	0.45	0.000110864	0.296610169	0.688959011
0.40	0.50	0.000110864	0.227118644	0.761832087
0.40	0.55	0.000103474	0.227118644	0.76183157
0.40	0.60	2.21746E-05	0.181328546	0.809846809
0.40	0.65	2.21746E-05	0	1.000000122
0.40	0.70	0	0	1

Note: Numbers correspond to the data points shown in Figure A.7. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

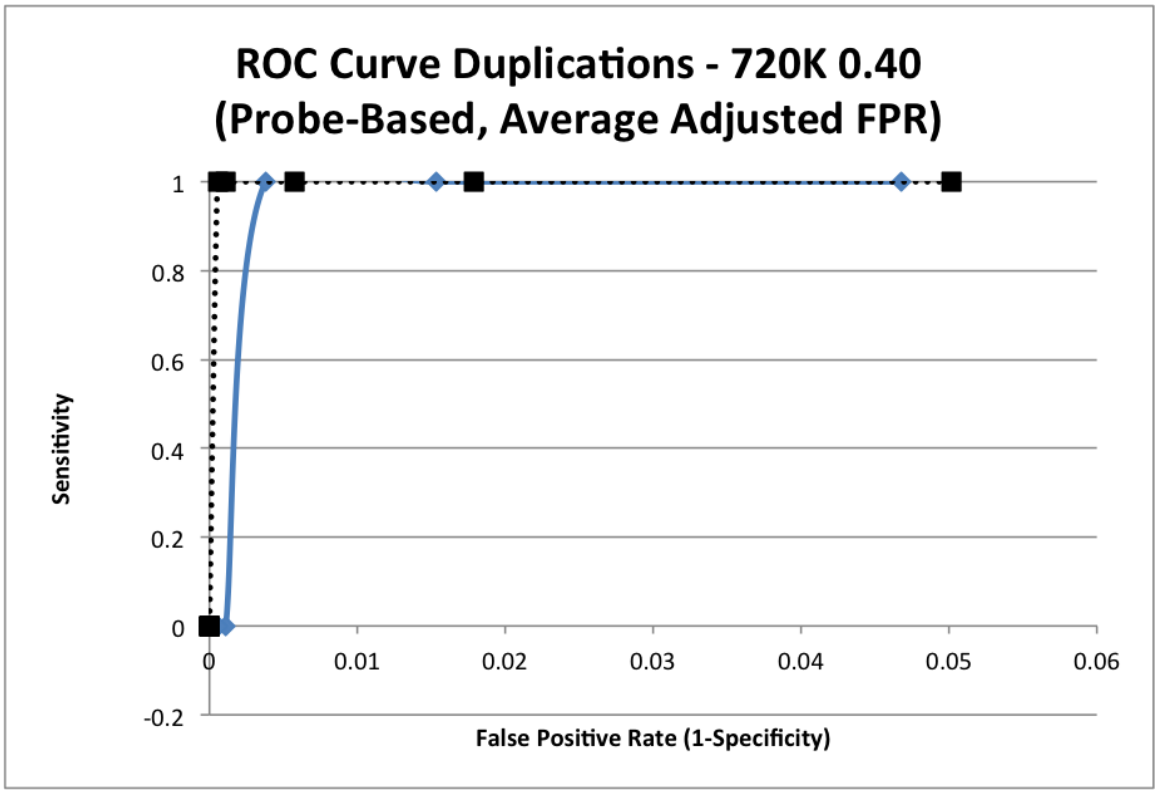

Figure A.8: ROC plot of analysis comparing all 385K CNV duplications greater than 400Kb to a 2.1M gold standard $\log_2$ threshold of 0.40. The solid line shows the analysis performed using all probes and the dotted line shows the analysis performed excluding probes from common CNV regions.

Table A.15: A table containing analysis results comparing all 385K CNVs duplications greater than 400Kb to a 2.1M gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were included.

2.1M $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.042554762	1	1
0.40	0.25	0.018065492	1	0.424523401
0.40	0.30	0.007054415	1	0.165772641
0.40	**0.35**	**0.002656034**	**0.889204545**	**0.127166041**
0.40	0.40	0.000839894	0.809756098	0.191264956
0.40	0.45	5.7746E-05	0.645454545	0.354548051
0.40	0.50	0	0.535714286	0.464285714
0.40	0.55	0	0.535714286	0.464285714
0.40	0.60	0	0.535714286	0.464285714
0.40	0.65	0	0.535714286	0.464285714
0.40	0.70	0	0	1

Note: Numbers correspond to the data points shown in Figure A.8. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

Table A.16: A table containing analysis results comparing all 385K duplications greater than 400Kb to a 2.1M gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were excluded.

2.1M $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.027314036	1	1
0.40	0.25	0.009570851	1	0.35040048
0.40	0.30	0.003066613	1	0.112272435
0.40	0.35	0.00058246	1	0.021324574
0.40	**0.40**	**0.000291313**	**1**	**0.010665324**
0.40	0.45	0	0	1
0.40	0.50	0	0	1
0.40	0.55	0	0	1
0.40	0.60	0	0	1
0.40	0.65	0	0	1
0.40	0.70	0	0	1

Note: Numbers correspond to the data points shown in Figure A.8. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

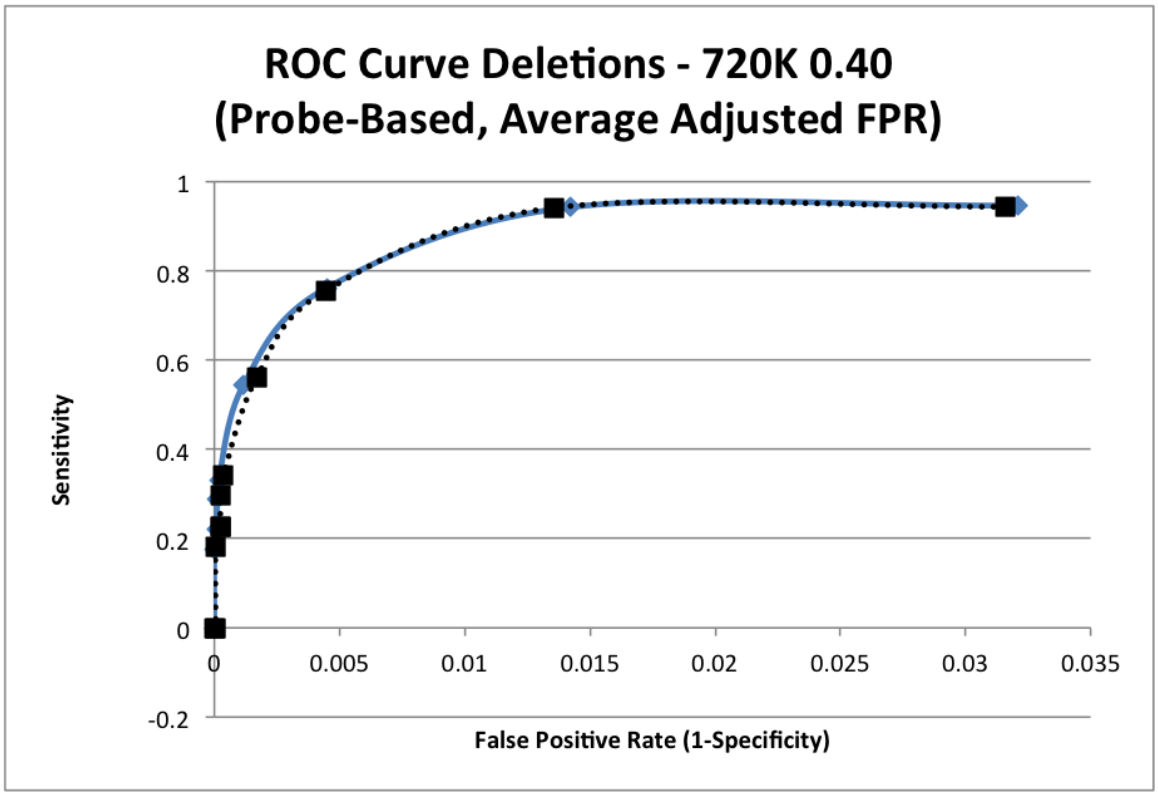

Figure A.9: ROC plot of analysis comparing all 385K CNV deletions greater than 400Kb to a 2.1M gold standard $\log_2$ threshold of 0.40. The solid line shows the analysis performed using all probes and the dotted line shows the analysis performed excluding probes from common CNV regions.

Table A.17: A table containing analysis results comparing all 385K CNVs deletions greater than 400Kb to a 2.1M gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were included.

2.1M $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.021191012	0.971409122	1
0.40	0.25	0.010629848	0.968773234	0.501627918
0.40	**0.30**	**0.005073996**	**0.846419327**	**0.271822656**
0.40	0.35	0.002300511	0.726570048	0.274430763
0.40	0.40	0.000699889	0.568250758	0.416336368
0.40	0.45	0.000479801	0.50880829	0.47675422
0.40	0.50	0.000165228	0.452182952	0.534565096
0.40	0.55	0.000157361	0.311064718	0.679820467
0.40	0.60	6.295E-05	0.231351351	0.76184522
0.40	0.65	3.93447E-05	0.123723042	0.87263747
0.40	0.70	0	0.050400916	0.948115665

Note: Numbers correspond to the data points shown in Figure A.9. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

Table A.18: A table containing analysis results comparing all 385K deletions greater than 400Kb to a 2.1M gold standard $\log_2$ threshold of 0.40. In this analysis, probes found in common CNV regions were excluded.

2.1M $\log_2$ Ratio	385K $\log_2$ Ratio	FPR	Sensitivity	Normalized Optimal Distance
0.40	0.20	0.015715403	0.943925234	1
0.40	0.25	0.006650783	0.941258741	0.42321097
0.40	**0.30**	**0.002179184**	**0.755304102**	**0.243225605**
0.40	0.35	0.000825213	0.559865093	0.410249965
0.40	0.40	0.000174677	0.340640809	0.639219746
0.40	0.45	0.000119104	0.296610169	0.685811302
0.40	0.50	0.000119104	0.227118644	0.759426977
0.40	0.55	0.000111165	0.227118644	0.759422104
0.40	0.60	2.38229E-05	0.181328546	0.807900884
0.40	0.65	2.38228E-05	0	1.000001149
0.40	0.70	0	0	1

Note: Numbers correspond to the data points shown in Figure A.9. The bolded row indicates the optimal $\log_2$ ratio found based on the closest linear distance (greyed box).

APPENDIX B

COMPUTATIONAL ANALYSIS CODE

B.1 Microarray Evaluation Script

```
#!/usr/bin/perl
# Written by Corey Goodman - Electrical and Computer Engineering, University of
Iowa
# Version 1.0
use strict;
use warnings;

use Getopt::Long qw(:config gnu_getopt);
use Scalar::Util qw(looks_like_number);

sub usage {
    print <<"USAGE";
Usage: perl $0 [OPTIONS] <input_directory> <output_directory>

<input_directory>   Input directory containing CNV call files, microarray
                    files, and a common location file if needed
<output_directory>  Name of the output directory where the results of the
                    analysis are placed. If specified directory does not
                    exist the directory is created.
-s, --small         Small resolution array unique identifier key used in
                    CNV call files in first column (required)
-l, --large         Large resolution array unique identifier key used in
                    CNV call files in first column (required)
-q, --quiet         Flag used to suppress output while analysis is running
                    If not specified script progress is printed while running
                    (optional)
-b, --buffer        Buffer used when calling CNVs (optional, default=15000)
-m, --minsize       Minimum CNV size used for TP/FN (optional, default=400000)
-c, --common        Exclude common CNV probes from analysis the average
                    CNV size (optional)
-g, --gains         Include CNV duplications in analysis.
                    If neither -g or -d flag is set both CNV types are included
-d, --deletions     Include CNV deletions in analysis.
                    If neither -g or -d flag is set both CNV types are included
-h, --help          Print this message

ARRAY IDs USED TO LABEL INDIVIDUAL CNV CALLS IN THE CNV CALL FILES AND USED TO
LABEL THE ARRAY FILE NAMES BE THE SAME. MAKE SURE THE -s AND -l ARGUMENTS MATCH
THESE IDENTIFIERS.

IF AN EXISTING FILE IS SPECIFIED AS OUTPUT IT WILL BE OVERWRITTEN

format of input_directory files
    <cnv_call_files>
        Required to begin with a log2 ratio threshold (number) used to call
CNVs
        followed by an underscore
        Example: 0.30_CNVCalls.txt
```

```
    <common_file> (optional)
        Required to be named: CommonLocations.txt
        Note: Can not perform analysis excluding common probes if this file
        does not exist.

    <low_resolution_array>
        Required to begin with the word "Array_" then a unique array type
        identifier different from the higher resolution array then and
        underscore and any desired text
        Example: Array_385k_segMNT.txt
        Note: The array is used for probe positions only so only information
        in the array file related to probe positions is needed.

    <high_resolution_array>
        Required to begin with the word "Array_" then a unique array type
        identifier different from the lower resolution array then and
        underscore and any desired text
        Example: Array_720k_segMNT.txt
        Note: The array is used for probe positions only so only information
        in the array file related to probe positions is needed.

Note: All files must end with a ".txt" extension and have UNIX line endings.
The first line of all files is expected to be a header line and is not parsed
out. The files are all tab delimitated with no quotation marks surrounding
fields.  On most unix platforms line endings can be converted with "dos2unix".
Array type identifier for microarrays must be the same identifier specified in
probe_roc.pl script

**When defining columns in files (below)
rc = required column      column must be present and meet formatting standards
                          set
uc = unimportant column   column must be present but does not need to meet any
                          format standards
nc = non-required column  column does not need to be present


format of CNV call file names:
[log2 ratio]_[filename].txt e.g. 0.30_CNVCalls.txt


format of <CNV_call_file>:
(Column    Field Description    [Field Format Descriptiion])
rc1:  Sample name
  Format: [patient unique id]_[array type identifier]
rc2:  Genomic position
  Format: [chromosome]:[start]-[stop]
  Note: chromosome is in the format chr[XX] where XX is chromosome identifier
rc3:  CNV event
  Possibilities: Gain, Loss
uc4:  CNV length
uc5:  Cytoband
uc6:  Probe log2 median
rc7:  Number of probes

Note: array type identifier in c1 must be the same identifier specified in
probe_roc.pl script

The analysis types possible are:
  Both - Analysis includes deletions and duplications
  Gain - Analysis includes duplications only
  Loss - Analysis includes deletions only
```

```
format of <microarray files>:
uc01: INDEX
uc02: X
uc03: Y
uc04: CONTAINER
uc05: SEQUENCE_ID
uc06: PROBE_ID
rc07: POSITION
  Note: this column is used by probe_roc.pl script
uc08: GC
rc09: CHROMOSOME
  Format: chr[XX] where XX is chromosome identifier
  Note: this column is used by probe_roc.pl script
nc10: CHR_POSITION
nc11: EXP_405948_2011-08-15_Slot2_532
nc12: REF_405948_2011-08-15_Slot2_635
nc13: RATIO
nc14: EXP_SPATIAL
nc15: REF_SPATIAL
nc16: RATIO_SPATIAL
nc17: EXP_NORM
nc18: REF_NORM
nc20: RATIO_CORRECTED


format of ROC analysis output file names:
[log2 ratio]_[analysis type]_Results.txt e.g. 0.30_Both_Results.txt


format of output files:
(Only results files should be in the directories)
rc1: gold standard (higher resolution) log2 ratio
rc2: lower resolution log2 ratio
rc3: True Positives (TP)
rc4: False Positives (FP)
rc5: True Negatives (TN)
rc6: False Negatives (FN)
rc7: Discarded Probes
rc8: Total Probes

Notice: analysis including CNVs on the X/Y chromosome should be
        interpreted with caution.
USAGE
    exit;
}


my $key_small = '';
my $key_large = '';
my $common_discard = '';
my $quiet = '';
my $distance_buffer = 15000;
my $size_cutoff = 400000;
my $gain_included = '';
my $loss_included = '';
my $opt_ok = GetOptions(
    's|small=s'     => \$key_small,
    'l|large=s'    => \$key_large,
    'q|quiet'     => \$quiet,
    'c|common'     => \$common_discard,
    'b|buffer=i'    => \$distance_buffer,
    'm|minsize=i'    => \$size_cutoff,
```

```
    'g|gains'     => \$gain_included,
    'd|deletions'     => \$loss_included,
    'h|help'            => sub { usage(); },
);

usage() unless ((@ARGV == 2) && $key_small && $key_large);

my $input_dir = $ARGV[0];
$input_dir .= '/' unless substr($input_dir,-1,1) eq '/';
my $output_dir = $ARGV[1];
$output_dir .= '/' unless substr($output_dir,-1,1) eq '/';

# check input and output directories
if (!(-d $input_dir)) {
    warn 'The input directory does not exist!'."\n\n";
    usage();
}

if (!(-d $output_dir)) {
    warn 'The output directory does not exist, creating directory...'."\n";
    if (!(mkdir $output_dir)) {
        warn 'Can not create directory: '.$output_dir."\n\n";
        usage();
    }
}

my $postfix = $common_discard ? '_NoCommon' : '_Results';


# default variables that can't be set in options
my $common_file = "CommonLocations.txt";

# Hash Keys
my $gain_key = "Gain";
my $loss_key = "Loss";
my $both_key = "Both";
my $none_key = "None";
my $chr_key = "chr";
my $start_key = "start";
my $stop_key = "stop";
my $call_type_key = "call type";
my $large_key = "Large";

# determined by -g and -d flags

# set CNV types used for analysis
my $type_key1 = $gain_key;
my $type_key2 = $loss_key;

if ($gain_included && !$loss_included) {
    $type_key2 = $gain_key;
}
elsif (!$gain_included && $loss_included) {
     $type_key1 = $loss_key;
}

my $output_type = ($type_key1 eq $type_key2) ? $type_key1 : $both_key;

my %common_cnvs = ();
my @log_ratios = ();
my %patient_ids = ();
my $array_file_large = '';
my $array_file_small = '';
```

```
my %cnv_calls_large = ();
my %cnv_calls_small = ();

print 'Reading input files...'."\n" unless $quiet;
opendir(IN_DIR, $input_dir) or die 'Could not open input directory: '
        .$input_dir." : $!\n";
my @input_files = readdir(IN_DIR);
foreach my $input_file (@input_files) {
    # ignore hidden files
    if(!($input_file =~ m/^(\.)/i)) {
        # Put Common CNV Calls into hash
        if ($input_file eq $common_file) {
            if($common_discard) {
                open(IN, '< '.$input_dir.$common_file) or die
                        'Can\'t open common file '.$common_file." : $!\n";
                my $line = <IN>; #Remove the header
                my $count = 0;

                # read common CNVs into hash
                while($line = <IN> ) {
                    chomp $line;

                    my @line_data = split("\t", $line);
                    #turn line into an array of elements

                    # fromat cnv location info
                    my $loc = $line_data[0];
                    $loc =~ tr/,//d;
                    $loc =~ tr/:/-/;

                    my @location_info = split("-", $loc);
                    my $chr = $location_info[0];
                    my $start = $location_info[1];
                    my $stop = $location_info[2];

                    $common_cnvs{$chr}{$start} = $stop;

                } # end looping through file
                close(IN);
            }
        }
        elsif ($input_file =~ m/^(Array)/i) {
            my @file_parts = split('_', $input_file);
            if ($file_parts[1] eq $key_small) {
                $array_file_small = $input_dir.$input_file;
            }
            elsif ($file_parts[1] eq $key_large) {
                $array_file_large = $input_dir.$input_file;
            }
        }
        # files should be cnv call files
        else {
            # ensure file name is formatted correctly
            my @file_parts = split('_', $input_file);
            if (looks_like_number($file_parts[0])) {
                my $log_ratio = $file_parts[0];
                # add log-ratio to list
                push(@log_ratios, $log_ratio);

                # Put cnv Calls into hash
                open(IN, '< '.$input_dir.$input_file) or die
                        'Can\'t open CNV file '.$input_file." : $!\n";
                my $line = <IN>; #Remove the header
```

```
                # loop over each CNV call
                while($line = <IN> ) {
                    chomp $line;
                    my @line_data = split("\t", $line);
                    #turn line into an array of elements

                    # format patient info
                    my @patient_info = split("_", $line_data[0]);
                    my $array_type = $patient_info[1];
                    my $patient = $patient_info[0];

                    # push patient id onto hash
                    $patient_ids{$patient} = 1;

                    my @temp = split(" ", $line_data[2]);
                    my $call_type =  $temp[@temp-1];

                    # fromat cnv location info
                    my $loc = $line_data[1];
                    $loc =~ tr/,//d;
                    $loc =~ tr/:/-/;

                    my @location_info = split("-", $loc);
                    my $chr = $location_info[0];
                    my $start = $location_info[1];
                    my $stop = $location_info[2];

                    # filter by designated call type
                    if ($call_type eq $type_key1 || $call_type eq $type_key2) {
                        #my $length = $stop - $start;

                        # put CNV calls in appropriate hash
                        if ($array_type eq $key_small) {
                            $cnv_calls_small{$log_ratio}{$patient}{$chr}
                                    {$start}{$chr_key} = $chr;
                            $cnv_calls_small{$log_ratio}{$patient}{$chr}
                                    {$start}{$start_key} = $start;
                            $cnv_calls_small{$log_ratio}{$patient}{$chr}
                                    {$start}{$stop_key} = $stop;
                            $cnv_calls_small{$log_ratio}{$patient}{$chr}
                                    {$start}{$call_type_key} = $call_type;
                        }# end type check
                        elsif ($array_type eq $key_large) {
                            $cnv_calls_large{$log_ratio}{$patient}{$chr}
                                    {$start}{$chr_key} = $chr;
                            $cnv_calls_large{$log_ratio}{$patient}{$chr}
                                    {$start}{$start_key} = $start;
                            $cnv_calls_large{$log_ratio}{$patient}{$chr}
                                    {$start}{$stop_key} = $stop;
                            $cnv_calls_large{$log_ratio}{$patient}{$chr}
                                    {$start}{$call_type_key} = $call_type;
                        }# end type check
                    }
                } # end looping through file
                close(IN);
            }
            else {
                warn 'Improperly formatted files exist in input directory: '
                        .$input_file."\n";
            }
        }
    }
```

```
}
closedir(IN_DIR);

if (!$array_file_small || !$array_file_large) {
    warn 'The microarray keys entered [',$key_small.', '.$key_large.
            '] could not be matched to corresponding microarray files'."\n";
    usage();
}

# sort ratios in ascending order
@log_ratios = sort { $a <=> $b } @log_ratios;

print "Analysis being performed for type: ".$output_type."\n" unless $quiet;

my %probes_small = ();
my %probes_large = ();
my $discarded = 0;

print 'Reading probes from large array file...'."\n" unless $quiet;
# Open large microarray file
open(IN, '< '.$array_file_large) or die "Can't open ".$array_file_large.
        " : $!\n";
my $line = <IN>; #Remove the header
while($line = <IN> ) {
    chomp $line;
    my @line_data = split("\t", $line); #turn line into an array of elements

    my $chr = $line_data[0];
    my $pos = $line_data[1];

    my $common_overlap = '';
    if ($common_discard) {
        foreach my $common_start (keys %{$common_cnvs{$chr}}) {
            my $common_stop = $common_cnvs{$chr}{$common_start};
            if ($pos >= $common_start && $pos <= $common_stop) {
                $common_overlap = 1;
                last;
            }
        }
    }

    if (!$common_overlap) {
        push(@{$probes_large{$chr}}, $pos);
    }
}
close(IN);

print 'Reading probes from small array file...'."\n" unless $quiet;
# open small microarray file
open(IN, '< '.$array_file_small) or die "Can't open ".$array_file_small.
        " : $!\n";
$line = <IN>;
while($line = <IN> ) {
    chomp $line;
    my @line_data = split("\t", $line); #turn line into an array of elements
    my $chr = $line_data[0];
    my $pos = $line_data[1];

    my $common_overlap = '';
    if ($common_discard) {
        foreach my $common_start (keys %{$common_cnvs{$chr}}) {
            my $common_stop = $common_cnvs{$chr}{$common_start};
            if ($pos >= $common_start && $pos <= $common_stop) {
```

```
                $common_overlap = 1;
                last;
            }
        }
    }

    if (!$common_overlap) {
        push(@{$probes_small{$chr}}, $pos);
    }
    else {
        $discarded++;
    }
}
close(IN);



print 'Sorting probe arrays...'."\n" unless $quiet;
#################
# Sort arrays
foreach my $chr (keys %probes_small) {
    @{$probes_small{$chr}} = sort { $a <=> $b } @{$probes_small{$chr}};
}
foreach my $chr (keys %probes_large) {
    @{$probes_large{$chr}} = sort { $a <=> $b } @{$probes_large{$chr}};
}


print 'Discarding poor coverage probes...'."\n" unless $quiet;
################
# Find probes to throw out
my %throw_out;

# loop over each chromosome
foreach my $chr (keys %probes_small) {
    my $lastIndex = 0;
    my $found;
    my $probe_pos;
    my $count = 0;

    my @indices = ();
    for(my $i = 0; $i < @{$probes_small{$chr}}; $i++) {
        my $probe_pos = ${$probes_small{$chr}}[$i];
        my $index = $lastIndex;
        $found = 0;

        # loop until:
        #   end of large array probes is reached
        #   large array probe location has surpassed small array probe
        #   location + distance_buffer a corresponding probe is found
        while ($index < @{$probes_large{$chr}} &&
                ${$probes_large{$chr}}[$index] < $probe_pos + $distance_buffer
                && !$found) {
            # check if large-array probe is within distance_buffer of
            # small-array probe
            if (abs($probe_pos - ${$probes_large{$chr}}[$index]) <
                    $distance_buffer) {
                push(@indices, $i);
                $discarded++;
                $found = 1;
            }
```

```
            if (${$probes_large{$chr}}[$index] < $probe_pos) {
                $lastIndex = ${$probes_large{$chr}}[$index];
            }
            $index++;
        }
    }

    foreach my $i (reverse(sort { $a <=> $b } @indices)) {
        splice(@{$probes_small{$chr}}, $i, 1);
    }
}

$discarded *= keys(%patient_ids);

################
# Map large CNV calls to hash
foreach my $ratio_large (@log_ratios) {

    #Building large to small maps
    my %overlap = ();

    foreach my $patient (keys %patient_ids) {
        my %cnv_map_large = ();

        print 'Building map for '.$ratio_large.', '.$patient.'...'."\n"
                unless $quiet;
        foreach my $chr (keys %probes_large) {
            for (my $i = 0; $i < @{$probes_large{$chr}}; $i++) {
                $cnv_map_large{$patient}{$chr}{${$probes_large{$chr}}[$i]} = 0;
            }

            my $last_index = 0;
            foreach my $position_large (sort {$a<=>$b} keys
                    %{$cnv_calls_large{$ratio_large}{$patient}{$chr}}) {
                my $start = $cnv_calls_large{$ratio_large}{$patient}
                        {$chr}{$position_large}{$start_key};
                my $stop = $cnv_calls_large{$ratio_large}{$patient}
                        {$chr}{$position_large}{$stop_key};
                my $type = ($cnv_calls_large{$ratio_large}{$patient}
                        {$chr}{$position_large}{$call_type_key} eq $gain_key) ?
                        1 : -1;
                my $index = $last_index;
                while ($index < @{$probes_large{$chr}} &&
                        ${$probes_large{$chr}}[$index] < $start) {
                    $index++;
                }

                while ($index < @{$probes_large{$chr}} &&
                        ${$probes_large{$chr}}[$index] < $stop) {
                    if (${$probes_large{$chr}}[$index] >= $start &&
                            ${$probes_large{$chr}}[$index] <= $stop) {
                        if ($stop - $start >= $size_cutoff) {
                            $cnv_map_large{$patient}{$chr}
                                    {${$probes_large{$chr}}[$index]} =
                                    $type * 2;
                        }
                        else {
                            $cnv_map_large{$patient}{$chr}
                                    {${$probes_large{$chr}}[$index]} = $type;
                        }
                    }
                    $index++;
                }
```

```
            $last_index = $index;
        }

        $last_index = 0;
        for (my $i = 0; $i < @{$probes_small{$chr}}; $i++) {
            my $position_small = ${$probes_small{$chr}}[$i];
            my $index = $last_index;
            $overlap{$patient}{$chr}{$position_small}{$gain_key} = 0;
            $overlap{$patient}{$chr}{$position_small}{$loss_key} = 0;
            $overlap{$patient}{$chr}{$position_small}{$none_key} = 0;
            while ($index < @{$probes_large{$chr}} &&
                    ${$probes_large{$chr}}[$index] <
                    ${$probes_small{$chr}}[$i]-$distance_buffer) {
                $index++;
            }
            $last_index = $index;

            while ($index < @{$probes_large{$chr}} &&
                    ${$probes_large{$chr}}[$index] <
                    ${$probes_small{$chr}}[$i]+$distance_buffer) {
                if ($cnv_map_large{$patient}{$chr}
                        {${$probes_large{$chr}}[$index]} >= 1) {
                    $overlap{$patient}{$chr}{$position_small}{$gain_key}=1;
                    if ($cnv_map_large{$patient}{$chr}
                            {${$probes_large{$chr}}[$index]} == 2) {
                        $overlap{$patient}{$chr}
                                {$position_small}{$large_key} = 1;
                    }
                    else {
                        $overlap{$patient}{$chr}{$position_small}
                                {$large_key} = 0;
                    }
                }
                elsif ($cnv_map_large{$patient}{$chr}
                        {${$probes_large{$chr}}[$index]} <= -1) {
                    $overlap{$patient}{$chr}{$position_small}{$loss_key}=1;
                    if ($cnv_map_large{$patient}{$chr}
                            {${$probes_large{$chr}}[$index]} == -2) {
                        $overlap{$patient}{$chr}
                                {$position_small}{$large_key} = 1;
                    }
                    else {
                        $overlap{$patient}{$chr}
                                {$position_small}{$large_key} = 0;
                    }
                }
                else {
                    $overlap{$patient}{$chr}
                            {$position_small}{$none_key} = 1;
                }

                $index++;
            }
        }
        %cnv_map_large = ();
        undef %cnv_map_large;
    }
}

# Done building large to small maps

# Region outfile
open(OUTFILE, ">".$output_dir.$ratio_large.'_'.
```

```
        $output_type.$postfix.'.txt');
print OUTFILE $key_large.' Ratio'."\t".$key_small.' Ratio'."\t".'TP'.
        "\t".'FP'."\t".'TN'."\t".'FN'."\t".'Discarded'."\t".'Total'."\n";
close(OUTFILE);

foreach my $ratio_small (@log_ratios) {
    print 'Assigning truth values '.$ratio_large.' <= '.
            $ratio_small.'...'."\n" unless $quiet;

    my $TP = 0;
    my $FP = 0;
    my $TN = 0;
    my $FN = 0;

    foreach my $patient (keys %patient_ids) {

        my %cnv_map_small = ();

        foreach my $chr (keys %probes_small) {
            for (my $i = 0; $i < @{$probes_small{$chr}}; $i++) {
                $cnv_map_small{$patient}{$chr}
                        {${$probes_small{$chr}}[$i]} = 0;
            }

            my $last_index = 0;
            foreach my $position_small (sort {$a<=>$b} keys
                    %{$cnv_calls_small{$ratio_small}{$patient}{$chr}}) {
                my $start = $cnv_calls_small{$ratio_small}{$patient}
                        {$chr}{$position_small}{$start_key};
                my $stop = $cnv_calls_small{$ratio_small}{$patient}
                        {$chr}{$position_small}{$stop_key};
                my $type = ($cnv_calls_small{$ratio_small}{$patient}
                        {$chr}{$position_small}{$call_type_key} eq
                        $gain_key) ? 1 : -1;
                my $index = $last_index;
                while ($index < @{$probes_small{$chr}} &&
                        ${$probes_small{$chr}}[$index] < $start) {
                    $index++;
                }

                while ($index < @{$probes_small{$chr}} &&
                        ${$probes_small{$chr}}[$index] < $stop) {
                    if (${$probes_small{$chr}}[$index] >= $start &&
                            ${$probes_small{$chr}}[$index] <= $stop) {
                        if ($stop - $start >= $size_cutoff) {
                            $cnv_map_small{$patient}{$chr}{${$probes_small
                                    {$chr}}[$index]} = $type * 2;
                        }
                        else {
                            $cnv_map_small{$patient}{$chr}{${$probes_small
                                    {$chr}}[$index]} = $type;
                        }
                    }
                    $index++;
                }
                $last_index = $index;
            }
        }

        # Done Mapping small CNV calls

        # determine final truth values
        foreach my $chr (keys %probes_small) {
```

```
for (my $i = 0; $i < @{$probes_small{$chr}}; $i++) {
    my $position_small = ${$probes_small{$chr}}[$i];

    if ($cnv_map_small{$patient}{$chr}{$position_small} > 1) {
        if ($overlap{$patient}{$chr}{$position_small}
                {$gain_key} == 1) {
            $TP++;
        }
        else {
            $FP++;
        }
    }
    elsif ($cnv_map_small{$patient}{$chr}{$position_small} <
            -1) {
        if ($overlap{$patient}{$chr}{$position_small}
                {$loss_key} == 1) {
            $TP++;
        }
        else {
            $FP++;
        }
    }
    else {
        if (defined($overlap{$patient}{$chr}{$position_small}
                {$none_key}) && $overlap{$patient}{$chr}
                {$position_small}{$none_key} == 1) {
            $TN++;
        }
        else {
            if (defined($overlap{$patient}{$chr}
                    {$position_small}{$large_key}) &&
                    $overlap{$patient}{$chr}{$position_small}
                    {$large_key} == 1) {
                # gain
                if (defined($overlap{$patient}{$chr}
                        {$position_small}{gain_key}) &&
                        $overlap{$patient}{$chr}
                        {$position_small}{gain_key} == 1) {
                    if ($cnv_map_small{$patient}{$chr}
                            {$position_small} >= 1) {
                        $TP++;
                    }
                    else {
                        $FN++;
                    }
                }
                # loss
                else {
                    if ($cnv_map_small{$patient}{$chr}
                            {$position_small} <= -1) {
                        $TP++;
                    }
                    else {
                        $FN++;
                    }
                }
            }
            else {
                $TN++;
            }
        }
    }
```

```
                }
            }
        }


        # Probe output
        open(OUTFILE, ">>".$output_dir.$ratio_large.'_'.$output_type.$postfix.
                '.txt');
        print OUTFILE $ratio_large."\t".$ratio_small."\t".$TP."\t".$FP."\t".
        $TN."\t".$FN."\t".$discarded."\t".($TP+$FP+$TN+$FN+$discarded)."\n";
        close(OUTFILE);
    }
}

print 'Analysis Complete!'."\n" unless $quiet;
```

B.2 Microsoft Excel Formatting Script

```
#!/usr/bin/perl
# Written by Corey Goodman - Electrical and Computer Engineering,
# University of Iowa

# THIS SCRIPT DEPENDS ON the Excel::Writer module
#


use strict;
use warnings;

eval { require Excel::Writer::XLSX; 1 } or die required_module();

use Getopt::Long qw(:config gnu_getopt);
use Scalar::Util qw(looks_like_number);

sub required_module {
    print 'Cannot use script because a dependant module is not installed'."\n";
    print 'Excel::Writer module can be found at the following location'."\n";
    print 'http://search.cpan.org/~jmcnamara/Excel-Writer-XLSX-0.05/'."\n\n";
    usage();
}

sub usage {
    print <<"USAGE";
Usage: perl $0 [OPTIONS] <list_of_directories>

-o, --output             Name of the output file placed in the same
                         directory as the result files. If not specified
                         default name is CombinedROCResults.xlsx
-a, --averages           Normalize the true negatives (TN) using
                         the average CNV size if not specified the
                         TN rate is not normalized (optional)
-d, --directory          Directory containing CNV call files
                         (required when using -a option)
-r, --resolution         Low resolution array type key used in
                         CNV call files in first column
                         (required when using -a option)
-q, --quiet              Flag used to suppress output while analysis is
                         running. If not specified script progress is printed
                         while running (optional)
-h, --help               Print this message

IF AN EXISTING FILE IS SPECIFIED AS OUTPUT IT WILL BE OVERWRITTEN

format of CNV call file names:
[log2 ratio]_[filename].txt e.g. 0.30_CNVCalls.txt

Note: All files must end with a ".txt" extension and have UNIX line endings.
The first line of all files is expected to be a header line and is not parsed
out.  The files are all tab delimitated with no quotation marks surrounding
fields.  On most UNIX platforms line endings can be converted with "dos2unix".
Array type identifier for microarrays must be the same identifier specified in
probe_roc.pl script
```

```
*When defining columns in files
rc = required column       column must be present and meet formatting
                           standards set
uc = unimportant column    column must be present but does not need to meet
                           any format standards
nc = non-required column   column does not need to be present


format of CNV call files in each directory:
(Column    Field Description    [Field Format Descriptiion])
rc1:  Sample name
  Format: [patient unique id]_[array type identifier]
rc2:  Genomic position
  Format: [chromosome]:[start]-[stop]
  Note: chromosome is in the format chr[XX] where XX is chromosome identifier
rc3:  CNV event
  Possibilities: Gain, Loss
uc4:  CNV length
uc5:  Cytoband
uc6:  Probe log2 median
rc7:  Number of probes


format of ROC analysis output file names:
[log2 ratio]_[analysis type]_Results.txt e.g. 0.30_Both_Results.txt


The analysis types possible are:
  Both - Analysis includes deletions and duplications
  Gain - Analysis includes duplications only
  Loss - Analysis includes deletions only


format of files in list of directories:
(Only results files should be in the directories)
rc1: gold standard (higher resolution) log2 ratio
rc2: lower resolution log2 ratio
rc3: True Positives (TP)
rc4: False Positives (FP)
rc5: True Negatives (TN)
rc6: False Negatives (FN)
rc7: Discarded Probes
rc8: Total Probes

Notice: analysis including CNVs on the X/Y chromosome should be
        interpreted with caution.
USAGE
    exit;
}


# default set variables
my $size_cutoff = 400000;
my $outfile = 'CombinedROCResults';
my $gain_key = 'Gain';
my $loss_key = 'Loss';
my $both_key = 'Both';

# potential future feature
my $derivitive = '';

my $averages = '';
my $directory = '';
```

```
my $res_key = '';
my $quiet = '';
my $opt_ok = GetOptions(
    'a|averages'        => \$averages,
    'q|quiet'         => \$averages,
    'd|directory=s'     => \$directory,
    'r|resolution=s'    => \$res_key,
    'o|output=s'        => \$outfile,
    'h|help'            => sub { usage(); },
);

usage() unless ((@ARGV > 0) && (!$averages || $directory));

# TN Column
my $TN = 4;
my $average_adjusted = 'Average_Adjusted';
my $output_ext = '.xlsx';


my @analysis_files = ();

my %average_probes = ();
my %cnv_count = ();
my %probe_sum = ();


# remove any path info if needed
my @out_split = split('/', $outfile);
if (@out_split > 1) {
    $outfile = $out_split[@out_split-1];
}

if ($averages && !($outfile =~ m/^($average_adjusted)/i)) {
     $outfile = $average_adjusted.'_'.$outfile
}
# append extension if needed
if (!($outfile =~ m/\.xlsx$/)) {
    $outfile .= '.xlsx';
}


# compute CNV averages if flag is set
if ($averages) {
     print 'Using average FPR adjustment'."\n" unless $quiet;

    opendir(MAIN_DIR, $directory) or die 'Could not open directory: '.
            $directory." : $!\n";
    @analysis_files = readdir(MAIN_DIR);

    # read each file in CNV call file directory
    foreach my $analysis_file (@analysis_files) {
        # ignore hidden files
        if(!($analysis_file =~ m/^(\.)/i)) {
            my $file_ok = 1; # true

            # check if filename meets analysis file naming standard
            my @file_parts = split('_', $analysis_file);
            my $log_ratio = $file_parts[0];

            if (looks_like_number($log_ratio)) {
                $cnv_count{$log_ratio}{$gain_key} = 0;
                $cnv_count{$log_ratio}{$loss_key} = 0;
                $cnv_count{$log_ratio}{$both_key} = 0;
```

```
$probe_sum{$log_ratio}{$gain_key} = 0;
$probe_sum{$log_ratio}{$loss_key} = 0;
$probe_sum{$log_ratio}{$both_key} = 0;

open(IN, '< '.$directory.'/'.$analysis_file);
my $line = <IN>; chomp $line; # remove header line

# iterate over all CNV calls for ratio
while($line = <IN> ) {
    chomp $line; # remove newline characters

    # double check file meets formatting standard
    my @data_array = split("\t", $line);
    #turn line into an array of elements

    # stop if file irregularity
    if (!(($data_array[1] =~ m/^(chr)/i) &&
            looks_like_number($data_array[6]))) {
        $file_ok = 0; last;
    }
    else {
        my @sample_info = split('_', $data_array[0]);
        my $call_type = $data_array[2];

        if ($sample_info[1] eq $res_key && $data_array[3] >
                $size_cutoff) {
            $cnv_count{$log_ratio}{$both_key}++;
            $probe_sum{$log_ratio}{$both_key} +=
                    $data_array[6];

            if ($call_type eq $gain_key) {
                $cnv_count{$log_ratio}{$gain_key}++;
                $probe_sum{$log_ratio}{$gain_key} +=
                        $data_array[6];
            }
            elsif ($call_type eq $loss_key) {
                $cnv_count{$log_ratio}{$loss_key}++;
                $probe_sum{$log_ratio}{$loss_key} +=
                        $data_array[6];
            }
        }
    }
}

# compute averages if file_ok
if ($file_ok) {
    $average_probes{$log_ratio}{$gain_key} =
            $cnv_count{$log_ratio}{$gain_key} > 0 ?
            $probe_sum{$log_ratio}{$gain_key} /
            $cnv_count{$log_ratio}{$gain_key} : 1;
    $average_probes{$log_ratio}{$loss_key} =
            $cnv_count{$log_ratio}{$loss_key} > 0 ?
            $probe_sum{$log_ratio}{$loss_key} /
            $cnv_count{$log_ratio}{$loss_key} : 1;
    $average_probes{$log_ratio}{$both_key} =
            $cnv_count{$log_ratio}{$both_key} > 0 ?
            $probe_sum{$log_ratio}{$both_key} /
            $cnv_count{$log_ratio}{$both_key} : 1;
}
else {
          warn 'A CNV call file may not be formatted correctly:
'.
                  $analysis_file."\n";
```

```
                }
                close(IN);
            }
        } # analysis_dir
    }

    if (keys %average_probes < 1) {
        warn 'The files used to calculate the probe averages may not be '.
                'formatted correctly'."\n";
    }

    closedir(MAIN_DIR);
}

# Loop through each directory
foreach my $dir (@ARGV) {
    if (substr($dir,-1,1) ne '/') {
        $dir .= '/';
    }
    print 'Compiling results in '.$dir."\n" unless $quiet;
    &read_data($dir, $outfile, \%average_probes);
}



# Subroutine that reads all result files in a directory
# and merges them into one excel workbook with multiple
# sheets
sub read_data {
    my $result_dir = shift;
    my $output_file = shift;
    my $reference = shift;
    use vars qw($TN);
    use vars qw($averages);
    use vars qw($derivitive);
    use vars qw($quiet);
    use vars qw(%average_probes);

    my @additional_headers = ('FPR', 'Sensitivity', 'Distance to (0,1)',
            'Optimal Linear Ratio');
    my @derivitive_headers = ('Normalized Slope', '% Change in Slope', 'Der1',
            'Der2', 'Optimal Der Ratio');
    my @alphabet = ("A".."Z");

    my $workbook  = Excel::Writer::XLSX->new( $result_dir.$outfile );

    # read all result files in directory
    opendir(RESULT_DIR, $result_dir) or die 'Could not open directory: '.
            $result_dir." : $!\n";
    my @result_files = readdir(RESULT_DIR);
    foreach my $result_file (@result_files) {
        # ignore hidden files
        if(!($result_file =~ m/^(\.)/i)) {

            # check if filename meets analysis file naming standard
            my @file_parts = split('_', $result_file);
            my $log_ratio = $file_parts[0];

            if (looks_like_number($log_ratio)) {
                 my $analysis_type = $file_parts[1];
                 my $no_common = ($file_parts[2] =~ m/common/i) ?
                         ' No Commmon' : '';
                open(IN, '< '.$result_dir.'/'.$result_file);
```

```
my $worksheet_name = $log_ratio.' '.$analysis_type.$no_common.
        ' Analysis';
my $worksheet = $workbook->add_worksheet($worksheet_name);
my $bold = $workbook->add_format( bold => 1 );

# read file into array
my @data;
my $count = 0;

my $line = <IN>; chomp $line; # remove header line
my @headers = split("\t", $line);
#turn line into an array of elements
while($line = <IN> ) {
    chomp $line;
    my @data_array = split("\t", $line);
    #turn line into an array of elements
    for (my $n = 0; $n < @data_array; $n++) {
        $data[$n][$count] = $data_array[$n];
    }
    $count++;
}
close(IN);

# adjust averages if it has been specified
if ($averages) {
    for (my $i = 0; $i < @{$data[0]}; $i++) {
        if ($average_probes{$log_ratio}{$analysis_type}) {
           my $old_count = $data[$TN][$i];
            $data[$TN][$i] = int(($data[$TN][$i] /
                    $average_probes{$log_ratio}
                    {$analysis_type}) + 0.5);

            # add to discarded probes
            $data[$TN+2][$i] += ($old_count - $data[$TN][$i]);

            # adjust new total probes
           $data[$TN+3][$i] -= ($old_count - $data[$TN][$i]);
        }
    }
}

$worksheet->write( 'A1', \@headers, $bold );
$worksheet->write( 'A2', \@data );

$worksheet->write( $alphabet[@headers].'1',
        \@additional_headers, $bold );
if ($derivitive) {
    $worksheet->write( $alphabet[@headers+@additional_headers].
            '1', \@derivitive_headers, $bold );
}

my $max_row = @{$data[0]} + 1;
my $col = @data;

# Write equations to Excel sheet
for (my $row = 2; $row <= $max_row; $row++) {

    # (I) FPR
    $worksheet->write( $alphabet[$col].$row, '='.
            $alphabet[$col-5].$row.'/('.$alphabet[$col-4].
            $row.'+'.$alphabet[$col-5].$row.')' );
```

```
# (J) Sensitivity
$worksheet->write( $alphabet[$col+1].$row, '='.
        $alphabet[$col-6].$row.'/('.$alphabet[$col-3].
        $row.'+'.$alphabet[$col-6].$row.')' );

# (K) Linear Distance (normalized)
$worksheet->write( $alphabet[$col+2].$row, '=SQRT(((('.
        $alphabet[$col].$row.'/MAX('.$alphabet[$col].
        '$2:'.$alphabet[$col].'$'.$max_row.'))^2)+(((MAX('.
        $alphabet[$col+1].'$2:'.$alphabet[$col+1].'$'.
        $max_row.')-'.$alphabet[$col+1].$row.')/MAX('.
        $alphabet[$col+1].'$2:'.$alphabet[$col+1].'$'.
        $max_row.'))^2))' );

# (L) Optimal using Line
$worksheet->write( $alphabet[$col+3].$row, '=IF('.
        $alphabet[$col+2].$row.'=MIN('.$alphabet[$col+2].
        '$2:'.$alphabet[$col+2].'$'.$max_row.'),'.
        $alphabet[$col-7].$row.',0)' );

if ($derivitive) {
    # (M) Normalized Slope
    if ($row < $max_row) {
        $worksheet->write( $alphabet[$col+4].$row,
                '=(((MAX('.$alphabet[$col+1].$row.','.
                $alphabet[$col+1].($row+1).')-MIN('.
                $alphabet[$col+1].$row.','.
                $alphabet[$col+1].($row+1).'))*MAX('.
                $alphabet[$col+1].'$2:'.$alphabet[$col+1].
                '$'.$max_row.'))/(MAX('.$alphabet[$col].
                $row.','.$alphabet[$col].($row+1).
                ')-MIN('.$alphabet[$col].$row.','.
                $alphabet[$col].($row+1).'))*MAX('.
                $alphabet[$col].'$2:'.$alphabet[$col].
                '$'.$max_row.'))' );
    }

    # (N) Change in slope
    if ($row > 2 && $row < $max_row) {
        $worksheet->write( $alphabet[$col+5].$row,
                '=(MAX('.$alphabet[$col+4].($row-1).','.
                $alphabet[$col+4].$row.')-MIN('.
                $alphabet[$col+4].($row-1).','.
                $alphabet[$col+4].$row.'))/'.
                $alphabet[$col+4].($row-1) );
    }

    # (O) Der1
    if ($row < $max_row) {
        $worksheet->write( $alphabet[$col+6].$row,
                '=IF(ISERROR(('.$alphabet[$col+1].$row.'-'.
                $alphabet[$col+1].($row+1).')/('.
                $alphabet[$col].$row.'-'.$alphabet[$col].
                ($row+1).')),10^100,('.$alphabet[$col+1].
                $row.'-'.$alphabet[$col+1].($row+1).')/('.
                $alphabet[$col].$row.'-'.$alphabet[$col].
                ($row+1).'))' );
    }

    # (P) Der2
    if ($row < $max_row-1) {
        $worksheet->write( $alphabet[$col+7].$row,
                '=IF(ISERROR(('.$alphabet[$col+6].$row.
```

```
                    '-'.$alphabet[$col+6].($row+1).')/('.
                    $alphabet[$col].$row.'-'.$alphabet[$col].
                    ($row+1).')),-1*10^100,('.
                    $alphabet[$col+6].$row.'-'.
                    $alphabet[$col+6].($row+1).')/('.
                    $alphabet[$col].$row.'-'.$alphabet[$col].
                    ($row+1).'))' );
        }

        # (Q) Optimal using Der
        $worksheet->write( $alphabet[$col+8].$row, '=IF('.
                $alphabet[$col+7].$row.'=MAX('.
                $alphabet[$col+7].'$2:'.$alphabet[$col+7].'$'.
                $max_row.'),'.$alphabet[$col-7].$row.',0)' );
    }
}

# =SUM(L2:L12)/COUNTIF(L2:L12,">0")
$worksheet->write($alphabet[$col+3].($max_row+1), '=SUM('.
        $alphabet[$col+3].'2:'.$alphabet[$col+3].$max_row.
        ')/COUNTIF('.$alphabet[$col+3].'2:'.$alphabet[$col+3].
        $max_row.',">0")' );
# =SUM(Q2:Q12)/COUNTIF(Q2:Q12,">0")
if ($derivitive) {
    $worksheet->write($alphabet[$col+8].($max_row+1), '=SUM('.
            $alphabet[$col+8].'2:'.$alphabet[$col+8].$max_row.
            ')/COUNTIF('.$alphabet[$col+8].'2:'.
            $alphabet[$col+8].$max_row.',">0")' );
}

# Make chart
my $chart = $workbook->add_chart(
    type => 'scatter',
    subtype => 'smooth_with_markers',
    embedded => 1
);

# Configure the chart series data
$chart->add_series(
    name       => 'ROC '.$log_ratio,
    categories => '=\''.$worksheet_name.'\'!$'.$alphabet[$col].
            '$2:$'.$alphabet[$col].'$'.$max_row,
    values     => '=\''.$worksheet_name.'\'!$'.
            $alphabet[$col+1].'$2:$'.$alphabet[$col+1].
            '$'.$max_row,
);
$chart->set_legend( position => 'none' );

    my $chart_type = ($analysis_type =~ m/$both_key/i) ?
            'All Types' : ($analysis_type =~ m/$gain_key/i) ?
            'Duplications' : 'Deletions';
# Add a chart title and some axis labels.
$chart->set_title ( name => 'ROC Curve '.$chart_type.' '.
        $log_ratio.$no_common."\n".'(Average Adjusted FPR)' );
$chart->set_x_axis(
    name => 'False Positive Rate (1-Specificity)',
    min  => 0,
);
$chart->set_y_axis(
    name => 'Sensitivity',
    max  => 1,
);
```

```
                # Insert the chart into the worksheet (with an offset).
                $worksheet->insert_chart( 'A'.($max_row+1),
                        $chart, 10, 10, 1.11, 1.43 );

            }
        }
    } # end reading files
    closedir(RESULT_DIR);
}
```