

Critical Issues in the Processing of cDNA Microarray Images

Vincent Y. Jouenne

Thesis submitted to the faculty of the
Virginia Polytechnic Institute and State University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
Computer Science and Applications

Lenwood S. Heath, Chairman

Craig A. Struble

Ruth G. Alscher

June 28, 2001
Blacksburg, Virginia

Keywords: Image Processing, Automatic Gridding, Segmentation, Gene Expression,
Mann-Whitney Test, Seeded Region Growing Algorithm, Data Extraction.

Copyright 2001, Vincent Y. Jouenne

Critical Issues in the Processing of cDNA Microarray Images

by

Vincent Y. Jouenne

Committee Chairman: Lenwood S. Heath

Computer Science and Applications

(ABSTRACT)

Microarray technology enables simultaneous gene expression level monitoring for thousands of genes. While this technology has now been recognized as a powerful and cost-effective tool for large-scale analysis, the many systematic sources of experimental variations introduce inherent errors in the extracted data. Data is gathered by processing scanned images of microarray slides. Therefore robust image processing is particularly important and has a large impact on downstream analysis. The processing of the scanned images can be subdivided in three phases: gridding, segmentation and data extraction. To measure the gene expression levels, the processing of cDNA microarray images must overcome a large set of issues in these three phases that motivates this study.

This study presents automatic gridding methods and compares their performances. Two segmentation techniques already used, the Seeded Region Growing Algorithm and the Mann-Whitney Test, are examined. We present limitations of these techniques. Finally, we studied the data extraction method used in MicroArray Suite (MS), a microarray analysis software, via synthetic images and explain its intricacies.

Keywords: Image Processing, Automatic Gridding, Segmentation, Gene Expression, Mann-Whitney Test, Seeded Region Growing Algorithm, Data Extraction.

One of the symptoms of an approaching nervous breakdown is the belief that one's work is terribly important.

Bertrand Russell (1872 - 1970)

This work is dedicated to my parents, Rémi and Françoise Jouenne as well as my brother, Aurélien Jouenne.

ACKNOWLEDGEMENTS

The majority of my thanks go to Dr. Craig A. Struble who helped me explore the solutions to this problem. I also appreciate the time and the helpful advice he gave me throughout my learning on conducting research. I extend my thanks and owe my gratitude to Dr. Lenwood S. Heath and Dr. Ruth Alscher who also guided me through the chaos and confusion and accepted to introduce the little “French computer man” to the research world. This study would not have been possible without their expertise and thoughtful guidance on microarray issues.

I would like to thank the following for their support and help in reminding me there is a world beyond the computer terminal: Danielle for being so patient with her computer-addicted boyfriend; Sattadip, Wes, and Claudia for their friendship; everyone that helped me during my presidency of the French Students Association; and all the musicians from the Wind Symphony, Wind Ensemble and Clarion orchestra for all the great time I experienced playing with them.

Blacksburg, Virginia

Vincent Y. Jouenne

June 28, 2001

TABLE OF CONTENTS

1	Introduction	1
1.1	Motivation	1
1.2	Identification of the Problems	2
1.3	Summary of Results	5
1.4	Organization	6
2	cDNA Microarray Background	7
2.1	Biological Background	7
2.1.1	Proteins	7
2.1.2	Nucleic Acids: DNA and RNA	8
2.1.3	Transcription, Reverse Transcription and Translation	8
2.1.4	Polymerase Chain Reaction (PCR)	10
2.1.5	Regulation	10
2.2	Principle of a cDNA Microarray Experiment	10
2.2.1	Motivation	11
2.2.2	Process of a cDNA Microarray Experiment	11
3	Gridding	14
3.1	The Gridding Problem	14
3.1.1	Formalization	15
3.1.2	The Specifics of our Experiments	18
3.2	A Method Based on the DFT	19
3.2.1	The Discrete Fourier Transform (DFT)	19

3.2.2	Principle of our Method	19
3.2.3	Results	22
3.3	A Method Based on the Circular Hough Transform	25
3.3.1	The Hough Transform (HT)	25
3.3.2	Principle of our Method	25
3.3.3	Results	27
3.4	A Method Based on the Mann-Whitney Test	29
3.4.1	Principle of our Method	29
3.4.2	Results	29
3.5	Preprocessing of the Images	32
3.5.1	Image Data Analysis	32
3.5.2	Thresholding	33
3.5.3	Results	33
3.6	A Hybrid Method	37
3.6.1	Principle of our Method	37
3.6.2	Results	39
3.7	Discussion	41
4	Segmentation	44
4.1	The Segmentation Problem	44
4.2	A Classification of Some Segmentation Techniques	46
4.3	Fixed Circle Segmentation	47
4.4	Seeded Region Growing Algorithms (SRG)	48
4.4.1	Principle of the SRG Algorithm	49
4.4.2	Adaptation of the SRG Algorithm to the Segmentation Problem	50
4.4.3	A Critical Seed Choice	51
4.4.4	Improve the Seed Choice	59
4.5	Mann-Whitney Test (MWT)	61
4.5.1	Principle of the MWT	61
4.5.2	Adaptation of the MWT to our Segmentation Problem	62
4.5.3	Limitations of the MWT	63
4.6	Comparison of Performances	65

4.7 Discussion	67
5 Data Extraction and Analysis	72
5.1 Principle of the MicroArray Suite (MS) Procedure	72
5.2 Handling Outliers	75
5.3 Analysis of MS Performances	78
6 Conclusions and Future work	82
A Addressing Figures	88
B SRG Figures	103
C MWT Figures	124
D Data Extraction and Analysis	136

LIST OF FIGURES

2.1	Cell mechanisms	9
2.2	Process of a Microarray experiment	12
3.1	Gridding patches.	16
3.2	A 4-pin printing layout	16
3.3	A 3-pin printing layout	17
3.4	Algorithm of the Discrete Fourier Transform based method.	21
3.5	Circular hough transform accumulation principle for a circular target.	26
3.6	Algorithm of the Circular Hough Transform based method.	28
3.7	Algorithm of the Mann-Whitney Test based method.	30
3.8	Mediocre segmentation on a misaligned target	32
3.9	Algorithm of the hybrid method.	38
4.1	A target patch.	45
4.2	The Seeded Region Growing Algorithm	50
4.3	Labels at Iteration 1 of the SRG on the beignet.	53
4.4	Labels at Iteration 300 of the SRG on the beignet.	54
4.5	Labels at Iteration 545 of the SRG on the beignet.	55
4.6	Labels at Iteration 571 of the SRG on the beignet.	57
4.7	Principle of the Mann-Whitney Test used for microarray images segmentation.	62
5.1	MicroArray Suite Iterative Procedure to calibrate ratios	75
A.1	Frequency band of X_v for pin-array A1 in NS3	89

A.2	Sum of the row intensities on A1 of NS3	90
A.3	Gridding on A1 of NS3 with the FFT method	91
A.4	Gridding on A1 of NS5 with the FFT method.	91
A.5	Column sums on A1 of NS5.	92
A.6	Hough Transform result on A1 of NS5.	93
A.7	Gridding on A1 of NS5 with the CHT method.	94
A.8	Gridding on A2 of NS5 with the CHT method.	95
A.9	Histogram of NS3, channel 1(Cy3 dye).	96
A.10	Gridding on A1 of NS5 after preprocessing with the FFT method.	97
A.11	Gridding on A2 of NS5 image after preprocessing with the FFT method.	98
A.12	Gridding on A3 of NS5 after preprocessing with the FFT method.	99
A.13	Gridding on A1 of NS5 with the hybrid method.	100
A.14	Gridding on A1 of S4X3 thresholded with the hybrid method.	100
A.15	Gridding on A3 of S4X3 thresholded with the hybrid method.	101
A.16	Artifacts leading to wrong gridding of A2 and A4 – (S4X3,S4X5) thresholded.	102
B.1	Target (1, 7, 4, Cy3) or “beignet” zoomed in.	104
B.2	SRG result on the target (1, 7, 4, Cy3) with a seedsize of 2.	105
B.3	SRG result on the target (1, 7, 4, Cy3) with a seedsize of 3.	105
B.4	SRG result on the target (1, 7, 23, Cy3) with a “Max” seed	106
B.5	SRG result on the target (1, 7, 23, Cy3) with a “Center” Seed	106
B.6	SRG result on the target (1, 5, 3, Cy3) with a “Max” seed	107
B.7	SRG result on the target (1, 5, 3, Cy3) with a “Maximum Region” seed	107
B.8	SRG result on the target (1, 7, 10, Cy3) with a “Maximum Region” seed	108
B.9	SRG result on the target (1, 7, 10, Cy3) with a “Max” seed	108
B.10	SRG result on the target (1, 3, 14, Cy3) with a “Max” seed.	109
B.11	SRG result on the target (1, 3, 17, Cy3) with a “Max” seed.	109
B.12	SRG result on the target (1, 5, 3, Cy3) with a “Max” seed.	110
B.13	SRG result on the target (1, 7, 10, Cy3) with a “Max” seed.	110
B.14	SRG result on the target (1, 7, 4, Cy3) with a “Max” seed.	111
B.15	SRG result on the target (1, 7, 9, Cy3) with a “Max” seed.	111
B.16	SRG result on the target (2, 15, 7, Cy3) with a “Max” seed.	112

B.17	SRG result on the target (2, 6, 15, Cy3) with a “Max” seed.	112
B.18	SRG result on the target (1, 7, 23, Cy3) with a “Max” seed.	113
B.19	SRG result on the target (1, 7, 24, Cy3) with the “Max” seed.	113
B.20	SRG result on the target (1, 3, 14, Cy3) with a “Random” seed	114
B.21	SRG result on the target (1, 3, 17, Cy3) with a “Random” seed	114
B.22	SRG result on the target (1, 5, 3, Cy3) with a “Random” seed	115
B.23	SRG result on the target (1, 7, 10, Cy3) with a “Random” seed	115
B.24	SRG result on the target (1, 7, 4, Cy3) with a “Random” seed	116
B.25	SRG result on the target (1, 7, 9, Cy3) with a “Random” seed	116
B.26	SRG result on the target (2, 15, 7, Cy3) with a “Random” seed	117
B.27	SRG result on the target (2, 6, 15, Cy3) with a “Random” seed	117
B.28	SRG result on the target (1, 7, 23, Cy3) with a “Random” seed	118
B.29	SRG result on the target (1, 7, 24, Cy3) with a “Random” seed	118
B.30	SRG result on the target (1, 3, 14, Cy3) with a “Union” seed.	119
B.31	SRG result on the target (1, 3, 17, Cy3) with a “Union” seed.	119
B.32	SRG result on the target (1, 5, 3, Cy3) with a “Union” seed.	120
B.33	SRG result on the target (1, 7, 10, Cy3) with a “Union” seed.	120
B.34	SRG result on the target (1, 7, 4, Cy3) with a “Union” seed.	121
B.35	SRG result on the target (1, 7, 9, Cy3) with a “Union” seed.	121
B.36	SRG result on the target (2, 15, 7, Cy3) with a “Union” seed.	122
B.37	SRG result on the target (2, 6, 15, Cy3) with a “Union” seed.	122
B.38	SRG result on the target (1, 7, 23, Cy3) with a “Union” seed.	123
B.39	SRG result on the target (1, 7, 24, Cy3) with a “Union” seed.	123
C.1	Target areas obtained by MS after the application of the MWT on the image Cy3_S3	125
C.2	Result of the MWT on the target (1, 7, 4, Cy3) with $R = 5$ and $U_0 = 0$	126
C.3	Result of the MWT on the target (1, 7, 4, Cy3) with $R = 6$ and $U_0 = 0$	126
C.4	Result of the MWT on the target (1, 7, 4, Cy3) with $R = 7$ and $U_0 = 0$	127
C.5	Result of the MWT on the target (1, 7, 4, Cy3) with $R = 8$ and $U_0 = 0$	127
C.6	Result of the MWT on the target (1, 7, 4, Cy3) with $R = 9$ and $U_0 = 0$	128
C.7	Result of the MWT on the target (1, 7, 4, Cy3) with $R = 10$ and $U_0 = 0$	128
C.8	Result of the MWT on the target (1, 7, 23, Cy3) with $R = 5$ and $U_0 = 0$	129

C.9	Result of the MWT on the target (1, 7, 23, Cy3) with $R = 6$ and $U_0 = 0$	129
C.10	Result of the MWT on the target (1, 7, 23, Cy3) with $R = 7$ and $U_0 = 0$	130
C.11	Result of the MWT on the target (1, 7, 23, Cy3) with $R = 8$ and $U_0 = 0$	130
C.12	Result of the MWT on the target (1, 7, 23, Cy3) with $R = 9$ and $U_0 = 0$	131
C.13	Result of the MWT on the target (1, 7, 23, Cy3) with $R = 10$ and $U_0 = 0$	131
C.14	Noise segmentation above the target (1, 1, 8, Cy3) with $R = 7$ and $U_0 = 0$	132
C.15	Noise segmentation above the target (1, 1, 8, Cy5) with $R = 7$ and $U_0 = 0$	132
C.16	Noise segmentation above the target (1, 1, 14, Cy3) with $R = 7$ and $U_0 = 0$	133
C.17	Noise segmentation above the target (1, 1, 14, Cy5) with $R = 7$ and $U_0 = 0$	133
C.18	Noise segmentation left to the target (1, 7, 1, Cy3) with $R = 7$ and $U_0 = 0$	134
C.19	Noise segmentation left to the target (1, 7, 1, Cy3) with $R = 8$ and $U_0 = 0$	134
C.20	Noise segmentation left to the target (1, 7, 1, Cy3) with $R = 9$ and $U_0 = 0$	134
C.21	Noise segmentation left to the target (2, 7, 1, Cy3) with $R = 8$ and $U_0 = 0$	135
C.22	Noise segmentation left to the target (2, 7, 1, Cy3) with $R = 8$ and $U_0 = 8$	135
D.1	Image of perfect targets used as Channel 1	137
D.2	Image of perfect targets used as Channel 2	137
D.3	Targets used as Channel 1 to check the 4 highest-lowest pixels tossing hypothesis. . .	138
D.4	Targets used as Channel 2 to check the 4 highest-lowest pixels tossing hypothesis. . .	138
D.5	Targets in Channel 1 to test the radius influence on MS results.	139
D.6	Targets in Channel 2 to test the radius influence on MS Results.	139
D.7	Targets in Channel 1 to test the square shape influence on MS Results.	140
D.8	Targets in Channel 2 to test the square shape influence on MS Results.	140
D.9	Targets in Channel 1 to test the square shape influence on MS Results.	141
D.10	Targets in Channel 2 to test the square shape influence on MS Results.	141

LIST OF TABLES

2.1	Cy3-Cy5 wavelength	11
3.1	Manual Gridding of (NS3,NS5) with ScanAlyze.	22
3.2	Results on NS3 with the DFT based method.	23
3.3	Results on NS5 with the DFT based method.	23
3.4	Manual Gridding of (S4X3,S4X5) with ScanAlyze.	24
3.5	Results on S4X3 with the DFT method.	24
3.6	Results on S4X5 with the DFT method.	24
3.7	Results on NS3 with the CHT based method.	27
3.8	Results on NS5 with the CHT based method.	27
3.9	Results on S4X3 with the CHT based method.	28
3.10	Results on S4X5 with the CHT based method.	29
3.11	Results on NS3 with the MWT based method.	31
3.12	Results on NS5 with the MWT based method.	31
3.13	Results on S4X3 with the MWT based method.	31
3.14	Results on S4X5 with the MWT based method.	32
3.15	Results on NS3 equalized and thresholded with the DFT based method.	34
3.16	Results on NS5 equalized and thresholded with the DFT based method	35
3.17	Results on NS3 equalized and thresholded with the CHT based method	35
3.18	Results on NS5 equalized and thresholded with the CHT based method	35
3.19	Results on NS3 equalized and thresholded with the MWT based method	35
3.20	Results on NS5 equalized and thresholded with the MWT based method	35
3.21	Results on S4X3 equalized and thresholded with the DFT based method.	36

3.22	Results on S4X5 equalized and thresholded with the DFT based method.	36
3.23	Results on S4X3 equalized and thresholded with the CHT based method	36
3.24	Results on S4X5 equalized and thresholded with the CHT based method	36
3.25	Results on S4X3 equalized and thresholded with the MWT based method	36
3.26	Results on S4X5 equalized and thresholded with the MWT based method	37
3.27	Results on NS3 with the hybrid method.	39
3.28	Results on NS5 with the hybrid method.	40
3.29	Results on NS3 equalized and thresholded with the hybrid method.	40
3.30	Results on NS5 equalized and thresholded with the hybrid method.	40
3.31	Results on S4X3 with the hybrid method.	40
3.32	Results on S4X5 with the hybrid method.	40
3.33	Results on S4X3 equalized and thresholded with the hybrid method.	41
3.34	Results on S4X5 equalized and thresholded with the hybrid method.	41
3.35	Summary Table on the distances obtained for the NS3 image.	42
3.36	Summary Table on the distances obtained for the NS5 image.	42
3.37	Summary Table on the distances obtained for the S4X3 image.	43
3.38	Summary Table on the distances obtained for the S4X5 image.	43
4.1	Intensities of the <i>beignet</i> (seed size = 2)	52
4.2	Table of the SRG Iterations 545 to 571 on the beignet.	56
4.3	Sizes of the target site obtained with our SRG (Union method) on S3 images.	66
4.4	Sizes of the target site obtained with our MWT on images of the S3 experiment.	68
4.5	Background corrected Ratios of our SRG vs MS ones.	69
4.6	Background corrected Ratios of our MWT vs MS ones.	70
5.1	Target considered as Outliers by MS and tossed out to calibrate ratios	75
5.2	Results given by MS for the 4 highest-lowest pixels rejection	76
5.3	Ratios, Cal. ratios, calibration factor M Obtained by MS	76
5.4	Iterative procedure results	77
5.5	Results given by MS for the channel 1 of Figure D.2	77
5.6	Results given by MS for the channel 2 of Figure D.2	77
5.7	Ratios given by MS and ScanAnalyze on the same targets of S3 experiment	79

5.8	Micr. Suite Results on synthetic images of perfect targets with different radius. . . .	80
5.9	Micr. Suite Results on square 16 <i>times</i> 16 targets except one.	80
5.10	MicroArray Suite Results on doughnut-shape targets.	81

Chapter 1

Introduction

1.1 Motivation

A *gene* is defined as a contiguous stretch of DNA that contains the information necessary to build a protein or an RNA molecule [1]. Any cell of an organism is simultaneously producing numerous proteins and RNA molecules from the information in genes. The concept of monitoring the *expression level* of thousands of genes simultaneously in a single experiment was a simple fiction for geneticists a decade ago. The advent of new technologies such as *microarrays* allows today's geneticists to compare the relative quantities of mRNA molecules in a cell across a single factor of interest [2, 3, 4, 5]. As patterns in which a gene is expressed can be temporal, developmental and physiological, the factors studied could be different types of tissues, drug treatments or timepoints of a biological process.

Microarrays are now widely used to identify differentially expressed genes. However, the experimental process is complex and the results are subject to many sources of variations. The process consists of four steps: preparation of the biological material, printing, hybridization, image processing, and data analysis. Early literature focused on the process of microarray experiments [4, 5] as Schena et al. [6] and Hedge et al. [7]. The steps going from the array fabrication to the printing via *PCR amplification*, probe preparation and protocols are widely covered. To our knowledge, no software system supports all the steps of a microarray experiment. The motivation of this study is the development of *Expresso – A Microarray Experiment Management System* [8, 9]. Expresso's goal is to “close the loop” in the microarray experiment process. After their analysis, results of an

experiment often leads geneticists to find new hypotheses and design new experiments. We propose that Espresso encompass all the steps from the design of an experiment to the data analysis and mining. Espresso will also provide multiple methods at each step.

Major work has also been presented in the domain of the analysis of microarray data. Eisen et al. [2, 10] as well as Lazzeroni and Owen [11] present *cluster analysis* techniques. Kerr et al. [12, 13, 14] recently used *ANOVAs*. Newton et al. [15] as well as Sapir and Churchill [16] present models to improve the statistical analysis. *Data normalization* [17, 18, 19] has also been a major source of work. Dudoit et al. [20, 19] presents design and statistical techniques to normalize the data.

Davis et al. [17] surveys the sources of variations. To enhance the quality of microarray analysis, many research groups are currently working on the identification and minimization of these systematic variations. These variations could occur during the experiment material preparation (e.g. mRNA preparation, the reverse transcription, the labelling, PCR amplification) during the printing (e.g. systematic variations in target geometry, random fluctuations in target volume) and also in the detection process (e.g. scanner properties, labels efficiency, slide inhomogeneities) [18].

While more and more scientists are using this technology, no universal consensus exists on how to design and analyze an experiment [2]. Few references exist on the image processing of cDNA microarray images. However, we are convinced that the image processing is a critical step and the accuracy of the data extracted from it can have a large impact on the downstream analysis (clustering, statistical models). This study constitutes our first investigations about cDNA microarray image processing.

1.2 Identification of the Problems

The image processing of cDNA microarray experiments aims to locate and reduce hybrids fluorescence of varying shape and intensities in the image into a table of intensity measures and ratios for each hybrid. The many *sources of variations* (e.g within a slide or from slide to slide, from target or label incorporation characteristics) validates the need for quality measurements, robust confidence level and reliable normalization strategies. For reference, we present a list of identified sources of variations in [18]:

- mRNA preparation

- Transcription
- Labelling (variations in incorporation, photo-bleaching effect)
- PCR amplification
- Variations in pin geometry
- Random fluctuations in target volume
- Target fixation
- Hybridization parameters (time, temperature, buffering conditions, volume of probe)
- Slide inhomogeneities
- Non-specific hybridization
- Non-specific background and overshining
- Non-linear transmission characteristics
- Saturation effects
- Target shape variations

We add to this list the following:

- Human judgment in the gridding process
- Imperfection of the segmentation algorithm
- Lack of universal consensus on the data extraction, especially ratio computation.

Improving the image processing consist of eliminating these sources of variations. However, no common manner of processing the images and extracting this information exists.

The work of Dudoit et al. [21] is one of the few references on the image processing of cDNA microarrays. They presented a formal three-step process:

1. *Gridding* which aims at overlaying a grid on the arrayed hybrids fluorescence in the image. Automation of this step is critical to enabling the analysis of numerous experiments.

2. *Segmentation* which consists of differentiating the hybrid from the background of the image. This step fits a mask of pixels to each hybrid fluorescence in the image.
3. *Intensity extraction* which consists of computing the relative target intensity vs. the background intensity. This step involves estimating the background intensity, measuring a confidence level in the target quality and calibrating the results with a normalization method.

Many programs exist for processing and analyzing microarray images. However, scanning and image processing are currently resource-intensive tasks as human intervention is still required to locate hybrids, flag artifacts or reject faulty hybrids. Automation of this step is an important goal to achieve. Although ideally a grid overlay should match the hybrids, the reality is much different. Some hybrids can be misaligned, have various size and shape or the array can even be tilted. Automatic gridding is therefore not a simple problem. We prefer the term *gridding* rather than the term *addressing* used by Dudoit et al. [21] because it is expected that a grid will match the hybrids fluorescences. Some existing programs claim semi-automated or automated gridding but still requires human intervention. The gridding techniques used by these programs are generally unpublished and undocumented. We undertake in this study the implementation of automatic gridding methods that truly require no human intervention.

Identification of target boundaries and accurate extraction of target intensity is a second goal to achieve. *Segmentation* is the term used by image processing experts to name the process of subdividing an image into its constituents parts [22]. Once hybrids have been gridded, different segmentation techniques exist to delineate the hybrid from the background of the image. This task is complicated by the fact that each hybrid has its own shape, radius size and distribution in pixel intensities. Several segmentation methods are in use (e.g. Mann-Whitney Test (MWT) [23], Seeded Region Growing (SRG) [24, 21], *fixed circle* [10] or adaptive circle (see GenePix)) as well as background intensity estimation methods (e.g. neighboring square or circle, valleys, morphological opening). The literature always shows results obtained on clean hybrids despite the fact that noisy images with artifacts and hybrids of imperfect shapes are quite common.

Our primary concern is to identify a reliable measure(s) and be able to analyze with confidence the extracted data. Commercial packages use different solutions and we wish to compare the performance of some segmentation techniques. Though other techniques exist, only two segmentation techniques are detailed in the literature. They appear to be the most sophisticated techniques to our knowledge.

Chen et al. [23] propose the *Mann-Whitney Test (MWT)* [25, 26], a standard statistical method to differentiate two populations, based on the Wilcoxon rank-sum. Dudoit et al. [21] present the *Seeded Region Growing algorithm (SRG)* of Adams and Bischof [24], later improved by Mehnert and Jackway [27]. The algorithm uses starting pixels called *seeds* to grow regions of pixels based on a criteria, measuring the distance between pixel intensities. We compared and discovered limitations of the SRG and the MWT. We determined cases where these methods are not satisfactory and recommend cautious use of these algorithms.

Data extraction analysis implies estimation of the ratio of signal to background intensity. It also involves the development of normalization strategies using standard or specialized statistics. Many techniques have been presented (e.g. ANOVAs, regression). We examined the method used in MicroArray Suite, originally developed by Chen et al. [23] and provide additional details.

1.3 Summary of Results

This study examined different techniques for automatic gridding. Four methods have been presented respectively using the Discrete Fourier Transform (DFT), Circular Hough Transform (CHT), the Mann-Whitney test and a combination of the DFT and CHT. We made progress toward this goal as our final hybrid method succeeded in gridding two images of average quality NS3 and NS5 correctly. However, for images of poorer quality, such as our couple (S4X3,S4X5), none of our methods obtain satisfactory results.

The study examined two segmentation methods: the Seeded Region Growing algorithm and the Mann-Whitney test. Our implementation of the SRG does not have a satisfactory rate of good segmentation. The seed choice is critical in size and location. We experimented with various seed choice techniques. The seed choice needs to be adapted to the different hybrid fluorescence shape and size. Each technique performs better or worse than another depending on the particular hybrid. The random and unconnected seed choice techniques obtain a higher rate of satisfactory segmentation. The union of the different seed choice outcome gives the most satisfactory results though it includes extra pixels.

Our MWT obtains more consistent results than our SRG ones. Though it systematically includes a small number of noise pixels, we did not notice any catastrophic segmentation. However, the MWT result is dependent on the original mask used. Hybrids of different sizes obtain their best

segmentation with initial masks of different sizes. Therefore for each hybrid, estimating the radius (or largest distance between high intensity pixels) is necessary to determine the optimum initial mask to be used by the MWT.

In the Data Extraction and Analysis chapter, we present the theory proposed by Chen et al. [23] on ratio computation and the calibration procedure. Its implementation in MicroArray Suite uses some supplementary assumptions we deciphered. Evaluation of a segmentation technique performance is too complex on experimental data. We recommend the use of synthetic images. Our results show that the MicroArray Suite Mann-Whitney Test performs correctly on perfect targets with different radius, square targets or doughnut-shape targets. More complicated data remains to be tested.

1.4 Organization

The rest of this dissertation is structured as follows. After covering some biological basics, chapter 2 presents the principle of cDNA microarray experiments. In chapter 3, automatic gridding techniques are presented and compared. Chapter 4 deals with the segmentation problem. We implemented our own SRG and MWT and present some critical issues and improvements. Data extraction and analysis is discussed in chapter 5. We present the theory and undocumented details of MicroArray Suite data extraction and assess its validity on experimental and artificial results. We conclude and discuss possible future directions in chapter 6. For readability, most figures have been included in Appendices A for chapter 3, B and C for chapter 4 and D for chapter 5.

Chapter 2

cDNA Microarray Background

This chapter introduces a few biological concepts necessary to understand this study in section 2.1 and presents the principles of microarray technology in section 2.2.

2.1 Biological Background

All living organisms consist of cells, which contain nucleic acids and proteins. After reviewing the basic information on proteins and nucleic acids, this section presents the fundamental mechanisms of cellular function. We also present a biological reaction involved in microarray experiments, the Polymerase Chain Reaction (PCR) before introducing the notion of regulation in a living cell. Most of the material presented here can be found in greater details in [1].

2.1.1 Proteins

A cell relies on its proteins for a wide variety of functions. *Proteins* are chains of smaller molecules, called amino acids joined by peptide bonds. They generally consist of 100 to 5,000 *residues*, the combination of an amino acid and a peptide bond. Proteins can fold into three dimensions in a complex and non-symmetric way. Depending on its protein shape, a protein can bind to different kinds of molecule such as other copies of itself, other protein molecules or be the subunit of an enzyme.

The production of energy, the biosynthesis of all component macromolecules, the maintenance of

molecular architecture, and the ability to respond to intra and extracellular stimuli are all protein dependent. *Proteins* are the workhorse molecules of the cell, involved in cellular structuring, storage of energy and production or reproduction of other important biomolecules.

2.1.2 Nucleic Acids: DNA and RNA

Nucleic acids are responsible for encoding the information necessary to produce proteins. Two kinds of nucleic acids exist: *deoxyribonucleic acid*(DNA) and *ribonucleic acid* (RNA).

DNA is a double stranded chain of simpler molecules called *nucleotides*. A nucleotide is the combination of a phosphate, a sugar and one of four bases: adenine, guanine, cytosine, thymine. DNA is typically contained in *chromosomes*, each containing a few hundred *genes*. Genes are, roughly speaking, stretches of DNA containing genetic information. Each *gene* specifies the composition and structure of a protein but its entire strand is not coding for the protein.

DNA can also replicate itself. The strand used for replication is called the *template* while the other strand is called the *complementary strand*. This complementary strand is of interest for the understanding of microarray experiments. Indeed, when two separated but complementary strands are present, the two nucleic acids will eventually bond to form an hybrid. This property of nucleic acids is called *hybridization* and is a key reaction on which the microarray technology is based.

RNAs are much like DNA molecules but they are single-stranded. Different kinds of RNAs exist that have different functions. We are only interested in *messenger RNAs* (mRNAs) in our context because they are the form of RNA that carries the information from the DNA for the synthesis of proteins.

2.1.3 Transcription, Reverse Transcription and Translation

The mechanisms of a living cells consists of four transformations monitoring the flow of information. The transformations from:

1. DNA to RNA is called *Transcription*,
2. RNA to DNA is called *Reverse Transcription*,
3. RNA to Protein is called *Translation*,
4. DNA to DNA is called *Replication*.

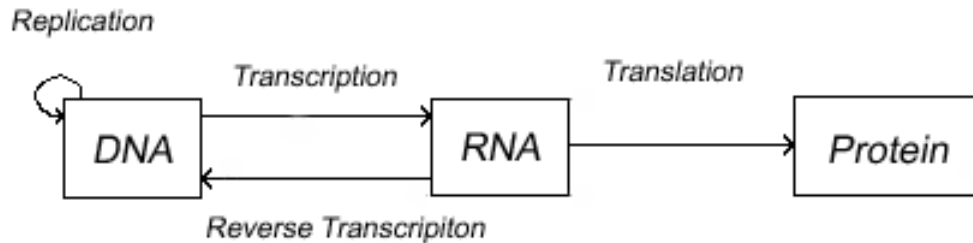


Figure 2.1: Cell mechanisms

The mechanism of transcription is started by the recognition of a *promoter*, a small part of DNA that indicates that a gene is ahead. A copy of the gene is then made into the messenger RNA or mRNA. In eukaryotes, organisms that have a nucleus containing the DNA, genes are not continuous. They have alternating parts termed introns and exons. After the copying of the gene into mRNA, the introns, not used in protein synthesis, are spliced out from the mRNA. The shortened mRNA is the molecule that brings the gene information to the *ribosome*, a structure outside the nuclear membrane of the cell where proteins are synthesized.

The shortened mRNA can be caught by biologists on its way to the ribosomes and by reversing the transcription process, a *complementary DNA* (cDNA), spliced complement of a gene is obtained. This transformation from RNA to DNA is called *reverse transcription*. *Complementary DNA* (cDNA) is the term used to define the spliced gene sequence of a complementary strand of DNA. An original gene sequence is called *genomic DNA*. This notion is important as biologists can reproduce a cDNA strand from the mRNA without using the genomic DNA. A cDNA is an artifact used instead of genomic DNA in microarray experiments.

Without the biologists' intervention, our shortened mRNA reaches the ribosome where the protein is synthesized based on the shortened mRNA information and a few other molecules as enzymes and tRNAs. This reaction is called *translation*. The amino acids are bound one by one to form the final protein and at ends the mRNA is released and recycled.

2.1.4 Polymerase Chain Reaction (PCR)

A microarray experiment requires a great amount of cDNA material. PCR amplification is used to prepare the large quantity of cDNA needed.

DNA is able to replicate itself via a process called replication. Biologists can create many copies of the same DNA molecule by the *polymerase chain reaction* (PCR). A *primer*, a small piece of single-stranded DNA used to initiate the reaction, and nucleotides are hydrogen bonded to a template strand via the action of an enzyme called DNA Polymerase that catalyzes the reaction.

The PCR process is made of two phases. First, double-stranded DNA is divided into two strands by heat. Second, primers and DNA Polymerase, respectively initiate and catalyze the reaction that leads to convert each single strand to a double-stranded molecule by addition of complementary bases. By repeating the process, many copies of the same piece of DNA can be obtained.

2.1.5 Regulation

Every cell in an organism contains a complete set of chromosomes for the organism. Consequently, each cell contains the information to reproduce the entire repertoire of proteins. Cells have differing properties, resulting from differences in abundance, distribution, and state of its proteins.

Many proteins serve specialized functions required in particular cell types. Differences in protein abundance are related to the types and levels of mRNA in the cell. The type and abundance of proteins and mRNAs a cell contains *regulate* cellular activity. A protein is one factor of regulation (among many others) in the production of other mRNA(s) or protein(s), which will themselves be at the origin of the production of other proteins and mRNAs and so on. *Regulation* is the biological term associated with this interdependence and interaction mechanism.

2.2 Principle of a cDNA Microarray Experiment

The biological concepts explained in the previous subsection are key elements involved in microarray technology. After explaining the motivation for the development of the stages of a microarray experiment, we describe next the process of a microarray experiment. The reader may refer to several references [3, 4, 5, 28] for more details on the technology.

Table 2.1: Cy3-Cy5 wavelength

	Absorption	Emission	Extinction
Cy3	552 nm	568 nm	130,000
Cy5	650 nm	667 nm	250,000

2.2.1 Motivation

Until recently, biologists were limited in their investigations study the presence and abundance of mRNAs in cells. With the advent of high-throughput technology such as microarrays, they have the means of observing the behavior of thousands of genes simultaneously and fasten the identification of gene functions and interactions. The knowledge of when and in what types of cell the protein product of a gene is expressed is an important goal to achieve for geneticists. It provides them with clues about the functions of particular genes, allows them to identify clusters of related genes, and motivates new hypotheses and experiments.

Recently, thanks to the reverse transcriptase and the PCR reaction, the cloning of cDNAs in a great quantity has been made possible. Libraries of publicly available *expressed sequence tags* (ESTs) have been created. ESTs, about a few hundred nucleotides, are small sequences of cDNA long enough to be unique and characteristic of one gene.

2.2.2 Process of a cDNA Microarray Experiment

Microarray experiment technology provides hybridization-based experiments that allow simultaneous *quantification* of the relative amount of each mRNA species in the cellular population.

The process of a microarray experiment starts with the biologist's hypotheses and selection of a set of genes of interest. This process is shown in Figure 2.2. After selection, DNA clones of interest (ESTs) are then amplified by PCR to generate a sufficient amount to allow 'printing' onto a glass microslide.

The printing is made by an arrayer. An arrayer is a robot with a certain number of pins programmed to deposit EST aliquots in an array configuration. An *aliquot* is a small quantity of premade nucleic acid. Different robots may have different numbers of pins as well as different pin configurations. Therefore microarray experiments do not have a universal layout, but rather the layout depends on the arrayer used.

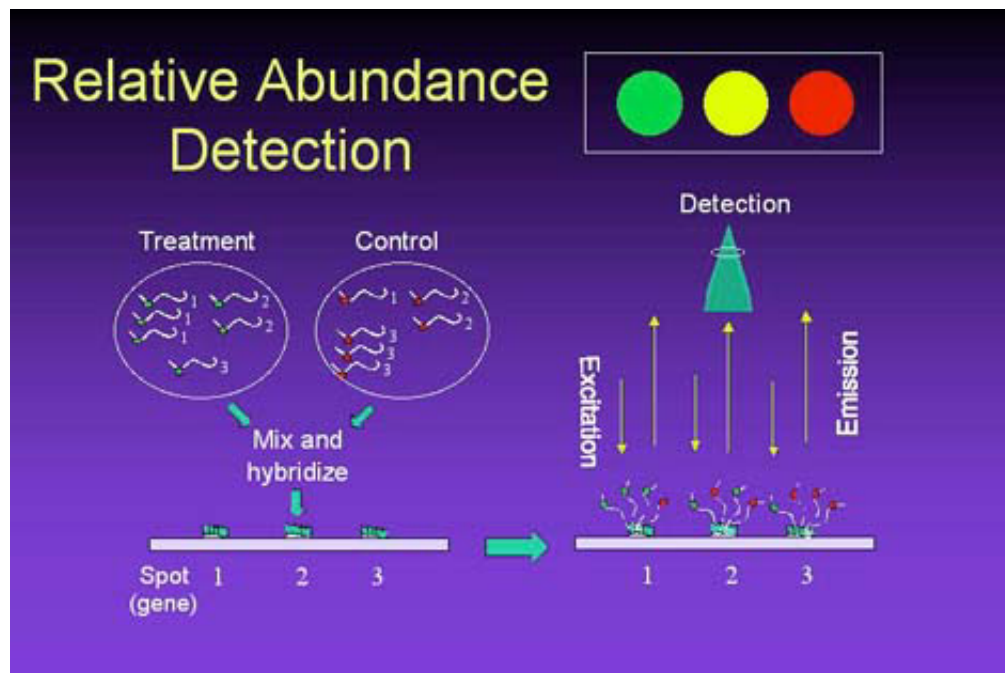


Figure 2.2: Process of a Microarray experiment

The fundamental concept in a microarray experiment is the quantification of cDNA-DNA hybrids. This binding is used for quantifying the presence of mRNAs in a particular cell. Geneticists isolate two samples of mRNAs, a control (reference) and a test (experimental) samples. The control sample contains mRNAs of a cell developed under normal conditions. The treatment sample contains mRNAs of a cell that has received a specific treatment (drought, chemicals, ...). In this study we used experiments on loblolly pine trees that have been subjected to different drought conditions. Some trees had been under severe or mild drought conditions whereas the control trees were grown in normal moisture conditions. The mRNA samples are reverse transcribed into cDNA samples that are then tagged with two different fluorescent dyes. The two samples are typically tagged with Cy3 and Cy5, fluorescent dyes with different wavelengths (see Table 2.1) visualized with green and red pseudo-color.

The two samples are mixed into water and the solution is dropped on the glass microslide. After 4 to 6 hours, the slide is washed to remove unbound cDNAs. The DNA-cDNA hybrids are then the only aggregates expected on the microslide. The slide is then placed in a laser scanner. Two scans are performed in the wavelength of each dye. The detector receives photons emitted by

the fluorescent dyes as the scanner excites the hybrids. Two images reporting the fluorescence of hybrids on the slide are generated. The image processing, motivation of this study, is performed to quantify the fluorescence at the site of each immobilized hybrid. The fluorescence intensities signal the relative presence of an hybrid compound in one sample and not the other or vice versa. This information allows the evaluation of the change and level of change in mRNA expression between the two samples.

Biologists speak of *gene expression level* as for the mRNA abundance in a cell corresponding to the level at which a gene has been expressed. Technically, the gene expression level is measured by the intensity mean of an immobilized hybrid compound in the red and the green channel. Channel is the term commonly used to refer to the image of hybrid fluorescent compounds for a particular dye. The ratio (or the log of a ratio) of the channel 2 over channel 1 is exported in a table and used to evaluate each gene expression level. Data analysis can then be performed. For reference, log ratios typically range from -4 to 4. Positive values indicate higher expression in the test(red) versus the control cell, and vice versa for negative values. The motivation for taking the log (red/green) ratio of the intensities in the two channels is due to the likely experimental variations as well as undesired contribution from the background intensities that can vary within the slide.

Chapter 3

Gridding

As the need to analyze a greater number of experiments in less time is emerging, automatic gridding is a desirable property for a microarray analysis package. No human intervention is a property that would hasten the analysis and avoid introducing a new source of variation. Semi-automatic gridding or “almost” automatic gridding methods already exist (see Spot [29] and DigitalGENOME [30] (<http://www.molecularware.com/digigenome.htm>)). However, examples shown always appear to result from images of relatively good quality. The experimental reality is different. Noisy images and artifacts appear to be very common elements the biologists are confronted with.

In this chapter, we introduce gridding techniques that require no user intervention. We first formalize the gridding problem and present the specifics of our experiments in section 3.1. In section 3.2, we present a technique based on a frequency analysis of the image realized with the *Discrete Fourier Transform (DFT)*. We then present a method based on the *Circular Hough Transform (CHT)* in section 3.3 and an other one based on the *Mann-Whitney Test (MWT)* in section 3.4. We show improvements brought by a *preprocessing* of the images in section 3.5 and a final hybrid method using both the DFT and the CHT in section 3.6. A discussion and future work is developed in section 3.7.

3.1 The Gridding Problem

The next paragraphs detail the gridding problem and introduce some vocabulary.

3.1.1 Formalization

We have seen in section 2.2 that an arrayer prints EST aliquots on a microslide via a few pins. Each pin is printing aliquots so that they are evenly spaced and in a grid with a definite number of rows and columns. These aliquots hybridize later with the two-sample cDNA mix dropped on the slide. Because of this organized printing, locating the hybrid compounds fluorescence on the image result from a process of *gridding*, which places a grid over the hybrid compounds fluorescence in the image so that each hybrid fluorescence is contained within a patch. This patch will actually be a square if we assume the pin-array is not tilted.

We next give a few definitions for a better understanding. An image has a finite number of pixel rows and columns, and we will refer to the dimensions of an image as its height H and width W . A *pixel* will be typically denoted by a pair (x, y) where x and y are its pixel coordinates or if needed by a triple (x, y, i) where i is the intensity of the pixel. The origin $(0, 0)$ is typically the top left pixel of the image and all image coordinates are non negative.

A *target* is the set of pixels corresponding to an hybrid compound fluorescence in the image. A *target patch* is a small image area expected to contain a target and its surrounding background (see Figure 3.1). Targets are expected to be evenly spaced along both image dimensions. We define by the term *spacing* S the deviation from one target center to another. Targets are expected to be placed in a regular pattern with an horizontal spacing S_h and a vertical spacing S_v . S_h and S_v are often the same.

Gridding consists therefore of fitting an array of target patches over each target in an optimal way. However, existing arrayers have different printing designs. An arrayer uses a certain number of pins P , and each pin is printing its own array of aliquots. We refer to the *array* (or microarray) as the whole mapping of targets. However, a *pin-array* is the individual array of targets resulting from what one pin prints.

Figure 3.2 and Figure 3.3 show two possible experiment layouts. Figure 3.2 shows the layout obtained with an four-pin arrayer while Figure 3.3 shows the layout as it would be with an arrayer that would have three pins in a one-column configuration. Gridding consists actually of fitting an array of target patches on each pin-array in an optimal way.

Each pin-array also has a definite number of rows and columns. We will refer to them later on as the *pin-array row-count* PR and *pin-array column-count* PC .

Each pin-array has a relative position in the image due to the printing layout. Each pin array

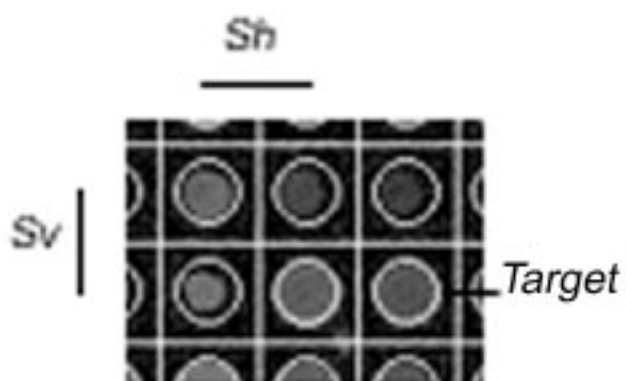


Figure 3.1: Gridding patches.

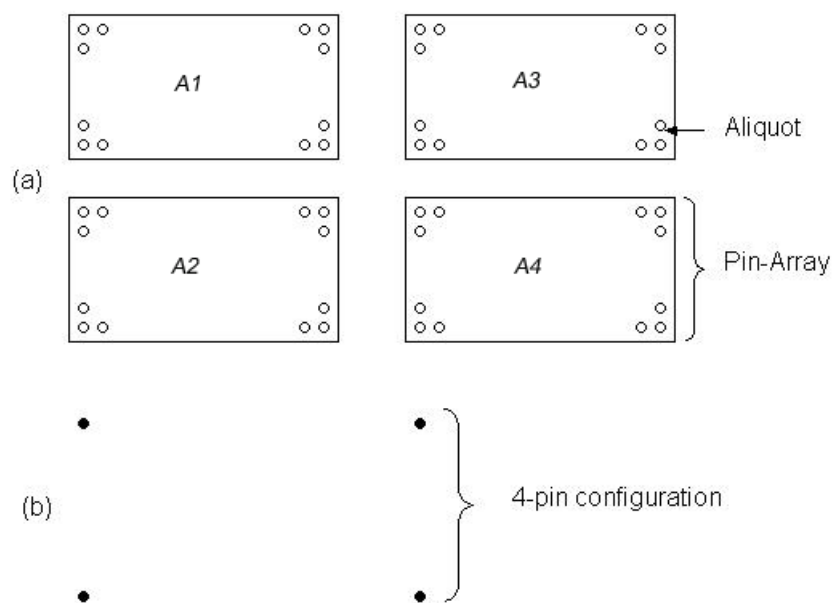


Figure 3.2: A 4-pin printing layout

(a) The four pin-arrays annotated A1, A2, A3 and A4. (b) the 4-pin configuration.

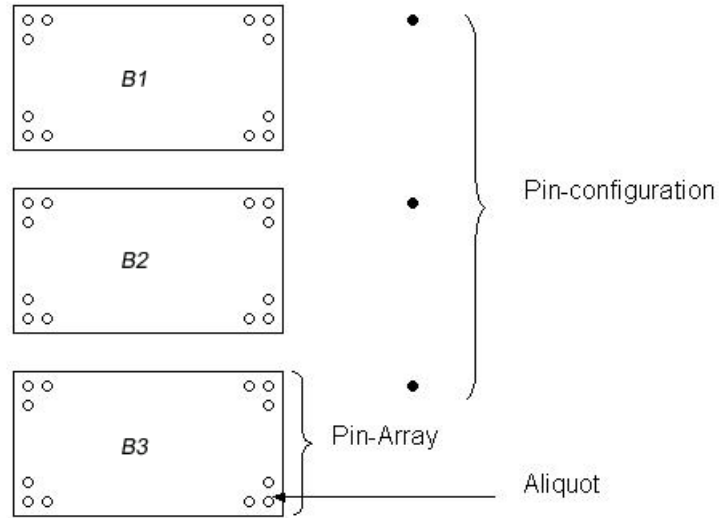


Figure 3.3: A 3-pin printing layout

B1, B2 and B3 are the three pin-arrays. The 3-pin configuration is shown on the right.

is self-contained in a rectangular part of the image of width RW and height RH . If no tilting is assumed, a grid can be simply defined by the coordinates of the top left corner target. Patches are the same for each target in a pin-array and pin-arrays have a finite row-count PR and column-count PC . Therefore the only top left target center coordinates (x_0, y_0) of a pin-array suffices to define the pin-array position in the image.

Each pin-array also has a relative position compared to other pin-arrays. The distance between the top left target of a pin-array to another pin-array top left target D is definite and depends on the distance between two robot pins.

Once a grid is overlayed over the targets, we have an estimation of the target and its surrounding background location. The methods presented in this Chapter assumed an *a priori* knowledge of:

1. The number of pin-arrays P ,
2. The position of a pin-array in a rectangular part of the image with a width RW and a height RH ; and
3. The row-count PR and column-count PC of pin-arrays.

These parameters are the only input used for our gridding methods. They are a reasonable

amount of information that a microarray designer package could easily transmit.

3.1.2 The Specifics of our Experiments

We would like first to mention a few problems we encountered with the gridding. First, a pin-array can be tilted. This problem needs to be taken into account by the gridding method. Target patches do not need to be square and can be quadrilaterals. The grid can also be tilted appropriately. A second problem occurs with misaligned targets because the arrayer may fail to print correctly. This problem introduces the need of a target adjustment method to optimize the target patch position.

The arrayer used for our experiments was built on the plans of Pat Brown's robot at Stanford Medical School (<http://cmgm.stanford.edu/pbrown/index.html>). Figure 3.2 shows the *printing layout* of our experiments. Since this arrayer has 4 pins, our images have four pin-arrays that we will refer to as A1, A2, A3 and A4.

Unless remarked otherwise, our pin-arrays typically have 16 rows and 24 columns which means $PR = 16$ and $PC = 24$. Experimental observations show that the *spacing* S of targets generally averages about 25 pixels on both vertical and horizontal dimension. Therefore $S_v = S_h = 25$.

Our gridding problem is then defined as follows. Assuming that tilting is not occurring, targets are perfectly aligned, spaced by $S = 25$ pixels, our gridding problem consists of overlaying a 16×24 grid of S times S square target patches S pixels large over the four pin-arrays A1, A2, A3 and A4 laid out in a square configuration. With knowledge of S , PR , PC as well as the assumption of no tilting effect, the top left target patch center (x_0, y_0) suffices to define a grid location.

Target pixels usually have relatively high intensities ranging from 10000 to 65535 whereas background pixels should have lower values between 1 and 5000. Note that 0 is not an expected value for background pixels as the glass is imperfect and emits a few photons. Therefore, background pixels are expected to have small but non-zero intensities. Any isolated background pixel that has an abnormally high intensity is categorized as *background noise*. We also use the term *artifacts* to designate aggregates of high-intensity pixels that do not correspond to any hybrid fluorescence.

We present our results on the image (NS3, NS5), a pair of images that are not too noisy and present few artifacts. Most of the targets are quite circular and well expressed. We also present our results on images (S4X3, S4X5) of a poorer quality. These images mostly have low expressed targets of non-circular shape. They also are extremely noisy and present huge artifacts.

3.2 A Method Based on the DFT

The following method takes advantage of the DFT to address the target location. After a brief introduction to the theory of the DFT we present the principle of our method and show the results obtained.

3.2.1 The Discrete Fourier Transform (DFT)

The *Discrete Fourier transform (DFT)* is one of the most advanced method of discrete signal processing. The Fourier transform X of a complex or real function x is

$$X(w) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t} dt. \quad (3.1)$$

X allows the spectral analysis of the function x .

In a discrete domain, a DFT D is defined as the Fourier transforms of a finite sequence of complex or real numbers d . If x_i , $i = 0, 1, 2, \dots, N - 1$ is a sequence d of N finite complex numbers, then D is given by a sequence of values X_k , where

$$X_k = \frac{1}{N} \sum_{i=0}^{N-1} x_i e^{-j2\pi ik/N} \quad (3.2)$$

$D=X_0, \dots, X_{N-1}$ is a sequence of complex numbers that reflects the frequencies of the original sequence d . The FFT algorithm is a fast algorithm for the computation of DFTs. The term Fast Fourier Transform (FFT) was first used in [31] and its basic properties were first described in [32]. The FFT reduces the number of multiplications and additions required to implement equation 3.2 from $O(N^2)$ to $O(N \ln N)$. More details about the theory and computations can be found in [22, 33].

3.2.2 Principle of our Method

The motivation for this method lies in the fact that targets are expected to be evenly spaced by S_v or S_h pixels along rows and columns. However, S_v and S_h can vary from experiment to experiment. The DFT allows us to compute S_v or S_h from an image.

If we sum the pixel intensities along rows and respectively columns, we obtain two vectors x_v and x_h of real numbers. A visualization of these vectors (see Figure A.2) show respectively PR and PC peaks spaced by respectively S_v and S_h pixels. In the frequency domain, this regular spacing

should result in a local maximum M at the frequency f_M but the DFTs X_v and X_h may have many local maxima.

However, M should be in a band-limited frequency domain $[f_{min}, f_{max}]$. The band limits are corresponding to two extremes cases of distribution of the targets. In the first case, S_v or S_h is maximum. The targets are spread over RH or RW in a regular but maximum spacing. This case corresponds to a minimum frequency limit f_{min} . In a second case, targets are juxtaposed. A maximum frequency limit f_{max} is deduced. Figure A.1 shows X_v , the DFT obtained in our image NS3.

Once f_M has been determined along a dimension of size R ($= RH$ or RW), the *period* p_M is computed by:

$$p_M = \frac{R}{f_M} \quad (3.3)$$

At this point, we have $p_{M_v} = S_v$ or $p_{M_h} = S_h$. We now need to determine (x_0, y_0) , the *offsets* of the pin-array. By offset, we mean the pixel coordinate of the first target center to the edge of the image.

As mentioned above, x_v and x_h , the vectors of the sum intensities along rows or respectively columns, present an expected number of *peaks* PR and PC . The peaks also have a regular spacing $p_{M_v} = S_v$ and $p_{M_h} = S_h$. However, Figure A.5 shows other peaks might appear corresponding to artifacts or noise in the image. Figure A.5 has a big peak on the left corresponding to a noisy left edge on the image NS3.

To determine the offset, we therefore used the following method. If N is the number of peaks expected along a dimension x , we compute the optimal sum Opt of N values evenly spaced by S_x . Opt is expected to occur with the N local maxima of the peaks $peak_{max}$.

The grid position has been determined. Indeed we now have the *spacing* $p_{M_v} = S_v$, $p_{M_h} = S_h$ and (x_0, y_0) the offset of the pin-array from the left and top border. Assuming the pin-array is not tilted, we can compute the relative position of each targets.

For obvious reasons as the *tilting effect* and the misalignment of targets, the current grid placement is not perfect. Therefore we developed an *adjustment method*

To adjusted the grid position, we move the *grid template* pixels by pixels in order to find the position that has a maximum intensity sum. We believe that rotating a template over a range of ± 4 degrees should take care of the tilting problem but we did not implement it so far. Figure 3.4

```

1  $P \leftarrow 4$  (or other number of pin-array in the image)
2  $PC \leftarrow 24$ 
3  $PR \leftarrow 16$ 
4  $Raster[W][H] \leftarrow readTIFF(Image)$ 
5 For  $i = 0$  to  $P$ 
6    $RW[i] \leftarrow width/2$ 
7    $RH[i] \leftarrow height/2$ 
8    $Rtop[i] \leftarrow 0$  or  $height/2$ 
9    $Rleft[i] \leftarrow 0$  or  $width/2$ 
10   $Rbottom[i] \leftarrow height/2$  or  $height$ 
11   $Rright[i] \leftarrow height/2$  or  $height$ 
12 For  $i = 0$  to  $P$ 
13   For  $j = Rtop[i]$  to  $Rbottom[i]$ 
14     For  $k = Rleft[i]$  to  $Rright[i]$ 
15        $R_Raster[i][j - Rtop[i]][k - Rleft[i]] \leftarrow Raster[j][k]$ 
16  $x_v[P][RH] \leftarrow NULL$ 
17  $x_h[P][RW] \leftarrow NULL$ 
18 For  $i = 0$  to  $P$ 
19   For  $j = 0$  to  $RH$ 
20      $x_v[i][j] \leftarrow \sum_{k=Rleft[i]}^{Rright[i]} R_Raster[i][j][k - Rleft[i]]$ 
21    $X_v[i] \leftarrow FFT(x_v[i])$ 
22    $p_M[i] \leftarrow RH / \max\{X_v[i][k], k = [f_{min}, f_{max}]\}$ 
23   For  $j = 0$  to  $RH$ 
24      $x_0[i] \leftarrow v$ 
25   with  $v = \{s \mid Opt(s) = \max\{\sum_{k=0}^{PR} x_v(s + k \times S_v), s \in [S_v/3, RH - p_M[i] \times PR - S_v/3]\}$ 
26   Do the same on columns.
27    $(x_0[i], y_0[i]) \leftarrow$  Adjust (find maximum intensity sum).
28   For  $j = 1$  to  $PR$ 
29     For  $k = 1$  to  $PC$ 
30        $target[i][j][k] \leftarrow (x_0[i] + j \times p_M[i], y_0[i] + k \times p_M[i])$ 

```

Figure 3.4: Algorithm of the Discrete Fourier Transform based method.

Table 3.1: Manual Gridding of (NS3,NS5) with ScanAlyze.

	Coord
A1	(167,123)
A2	(63,117)
A3	(180,224)
A4	(79,214)

The Table above and the following show the coordinates (x_0, y_0) for the top left targets of each pin-array.

present the algorithm described previously.

3.2.3 Results

In order to compare the results obtained by our algorithms, we performed manual gridding on the same images with ScanAlyze [10]. Table 3.1 shows (x_0, y_0) for each pin-array. These values will serve as reference. It takes about 45 minutes to grid the combined image manually while the DFT based method grids the 4 pin-array in approximately one second. Table 3.2 and Table 3.3 present the results obtained on the images NS3 and NS5. To assess the performance of each method presented, we also compute the distance between the results of our method and the ones of ScanAlyze. Given (x_1, y_1) the couple obtained with ScanALyze and (x_2, y_2) the one obtained by our automatic gridding method, we define the distance δ as:

$$\delta = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (3.4)$$

While all pin-arrays of the image NS3 have been correctly gridded within a few pixels, all pin-arrays of the NS5 image have been incorrectly located. The NS5 image has more low-expressed targets and more noise. A poor dye incorporation, a different wavelength and a different scanner laser used are potential factors involved in this problem. In this case, the algorithm failed to locate the left pin-arrays A1 and A2 correctly because of a noisy edge on the left border of the image. As stated in the previous subsection, this is resulting in a big peak our method fails to ignore. We identified this edge as the border of the microslide and we noticed the presence of this same noisy left edge on other images.

Figure A.3 shows the results obtained on the top left pin-array A1 of the NS3 image. Similar results are obtained for each pin-array. Figure A.4, top left pin-array A1 of the NS5 image is

Table 3.2: Results on NS3 with the DFT based method.

	Period	Fourier	Adjustment	ScanAlyze	δ
A1	(25,25)	(165,119)	(165,119)	(167,123)	4.47
A2	(25,25)	(62,116)	(62,116)	(63,117)	1.41
A3	(25,25)	(182,223)	(182,223)	(180,224)	2.24
A4	(25,25)	(79,209)	(79,209)	(79,214)	5.00

The “Period” column shows (p_{M_v}, p_{M_h}) , periods found by the DFT method. The “Fourier” column shows (x_0, y_0) , offsets found by the DFT method. The “Adjustment” column shows the new offsets after the adjustment method has been applied. The “ScanAlyze” column reminds the results obtained by manual gridding with ScanAlyze. The distance δ represents the results obtained with equation 3.4 between the “Adjustment” and “ScanAlyze” results

Table 3.3: Results on NS5 with the DFT based method.

	Period	Fourier	Adjustment	ScanAlyze	δ
A1	(25,25.64)	(165,12)	(165,12)	(167,123)	111.02
A2	(25,25.64)	(61,10)	(61,10)	(63,117)	107.02
A3	(25,25)	(131,223)	(131,223)	(180,224)	49.01
A4	(25,25)	(29,184)	(54,184)	(79,214)	39.05

showing the bad behavior of this algorithm in the case of the left noisy edge. The noisy edge is difficult to see on the figure.

Figure A.5 shows the column sums of the top left pin-array A1 of the NS5 image. The noisy edge on the left is corresponding to the big peak on the left. As this high-intensity edge is expanding along the whole height of the image, the peak is 2 to 3 times bigger than expected peaks. Our method to evaluate the offset is failing on this edge and does not identify correctly the location of expected peaks. This method does not take into account the geometry of the image. We implemented a scoring algorithm that would advantage a succession of peaks with a deviation of the appropriate period. The method was unsuccessful and results are not shown here. Cropping the left border of the image is not satisfactory with our goal since it requires a human judgment. We turned our investigations to other methods but there may be place for improvements in future works.

Table 3.4: Manual Gridding of (S4X3,S4X5) with ScanAlyze.

	Coord
A1	(256,215)
A2	(153,220)
A3	(263,315)
A4	(163,314)

Table 3.5: Results on S4X3 with the DFT method.

	Period	Fourier	Adjustment	ScanAlyze	δ
A1	(25,25)	(252,261)	(252,286)	(256,215)	71.11
A2	(25,25)	(151,265)	(576,365)	(153,220)	447.16
A3	(25,24.59)	(258,72)	(258,22)	(263,315)	293.04
A4	(25,25)	(567,31)	(567,31)	(163,314)	493.26

As expected the result on the experiment S4X are worse. The manual gridding results are presented in Table 3.4 and the results on the S4X3 image (Cy3 dye) in Table 3.5. Only 2 offsets are correct within a few pixels out of 8. The rest of the results are incorrect. We attribute these results to noisy areas. In this case, the adjustment method is failing and actually does not improve the results given by the Fourier procedure. If the algorithm could correctly identify the pin-arrays location in the NS3 image, it appeared not robust enough for our goals to fully automate the addressing of poor quality images as the one of the S4X experiments. However, in the case of poor quality image, combining the images may play a determining factor that we did not test yet.

Table 3.6: Results on S4X5 with the DFT method.

	Period	Fourier	Adjustment	ScanAlyze	δ
A1	(25,24.39)	(278,420)	(278,420)	(256,215)	206.18
A2	–	Fail	–	–	–
A3	–	–	–	–	–
A4	–	–	–	–	–

3.3 A Method Based on the Circular Hough Transform

In the following section we present a method based on the *circular Hough Transform* (CHT) to address the targets location of each pin-array. After an introduction to the theory of the Hough Transform, we explain the principle of our method and show the results obtained.

3.3.1 The Hough Transform (HT)

The Hough Transform is an image processing technique originally used to detect lines and circles. However, the method has been generalized so that it can detect objects of arbitrary shapes of a reasonable size [34]. In our context, we took interest in the method to find circles, the *Circular Hough Transform* (CHT).

Duda and Hart [35] present the Hough Transform. In a first step, the HT is computing an intensity *gradient image* at all pixel locations. A *gradient image* is an image of the first derivative of each pixel with its neighboring pixels. It is obtained by *convolution* of a small *operator* with the image and aims at detecting edges in an image. A large number of operators exist including the Sobel, Roberts, Prewitt ones [22]. The gradient image is then thresholded to keep the significant edge points. In a second step, a parameter space is computed. In the case of the *Linear Hough Transform*, for each edge pixel (x, y) all the line going through this point in the (m, c) space with $y = mx + c$ are plotted. The polar (r, q) space with $r = x \times \cos q + y \times \sin q$ is more often used. r is the length of a normal from the origin to this line and θ is the orientation of r with respect to the X-axis. The highest accumulator points in (r, q) space correspond to the strongest line edges in the image. In the *circular Hough Transform*, for each edge point, all the possible center locations at a distance R are accumulated in a parameter space R where R is an anticipated radius for our circles. A pixel that has been accumulated a large number of times is most probably a target center. Figure 3.5 shows the higher accumulation at the center of a circular target.

3.3.2 Principle of our Method

The Hough Transform is generally applied on the gradient image. The gradient image is thresholded and only the high values of the gradient image are used to compute the Hough space. Our implementation is more computationally intensive as we compute a Hough Space by going over all the pixels in the image without applying any gradient operator in the first place. We estimated a

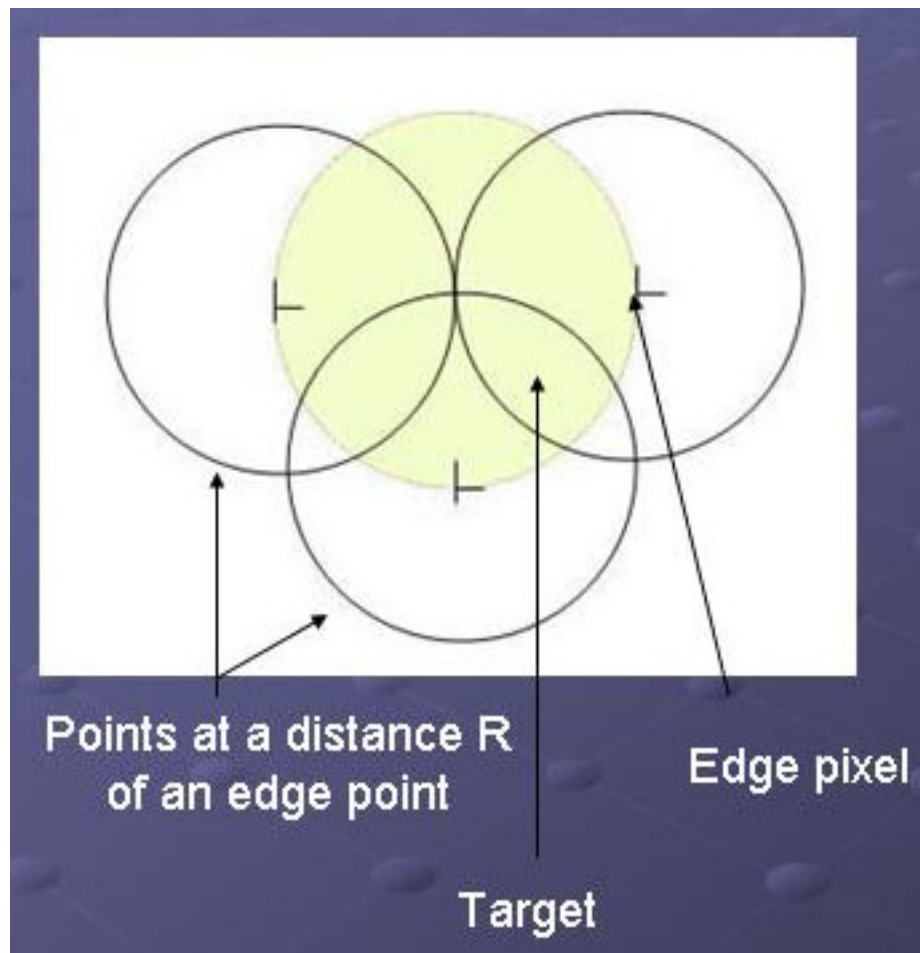


Figure 3.5: Circular hough transform accumulation principle for a circular target.

The circular target is in yellow. We show 3 edge points of the target and their accumulated points describing circles of radius R centered on one of the 3 edge points. The center of the target is the only point to be accumulated three times.

Table 3.7: Results on NS3 with the CHT based method.

	Hough	ScanAlyze	δ
A1	(167,119)	(167,123)	4.00
A2	(62,113)	(63,117)	4.12
A3	(155,221)	(180,224)	25.18
A4	(80,209)	(79,214)	5.10

Table 3.8: Results on NS5 with the CHT based method.

	Hough	ScanAlyze	δ
A1	(167,119)	(167,123)	4.00
A2	(62,14)	(63,117)	103.00
A3	(155,221)	(180,224)	25.18
A4	(56,209)	(79,214)	23.53

radius $R = 7$ that appeared to be the one occurring the most in our image. Once the parameter space had been computed, we searched the optimal sum of 16×24 values in the parameter space with a constant deviation equal to the period. The Figure 3.6 shows the algorithm implemented.

3.3.3 Results

Table 3.7 and Table 3.8 present the result obtained on the (NS3,NS5) images. Three out of 4 pin-arrays in the NS3 image have been correctly placed. The pin-array A2 is off by 25 pixels, a row, due to a quite non-expressed row at the bottom of the pin-array. We noticed our images often have low-expressed rows or columns in the borders of the pin-array that leads to such results.

The NS5 image shows also an interesting result. Unlike the Fourier Transform based method, the pin-array A0 has been correctly located despite the noisy left edge. The pin-array A1 was not correctly located and again because of the noisy left border of the microslide. The pin-array A3 and A3 are both off by a row. Note however that no grid template adjustment is used in this method.

We show the image obtained after our Hough Transform in appendix 3.3. The algorithm performed quite correctly on the NS3 image but we still encountered the same problem on the bottom left pin-array A2 of the image NS5 with the noisy left edge as shown in Figure A.8.

Table 3.9 and Table 3.10 show the results obtained with the Hough Transform based method on the images (S4X3,S4X5). The results are better than the ones obtained with the Fourier method

```

1  $T \leftarrow \text{treshold}$ 
2  $R \leftarrow \text{radius}$ 
3  $P \leftarrow 4$  (or other number of pin-array in the image.)
4  $\text{Period} \leftarrow 25$  (by using DFT eventually)
5  $PC \leftarrow 24$ 
6  $PR \leftarrow 16$ 
7  $\text{Raster}[W][H] \leftarrow \text{readTIFF}(\text{Image})$ 
8 For  $i = 0$  to  $P$ 
9   For  $j = \text{Rtop}[i]$  to  $\text{Rbottom}[i]$ 
10    For  $k = \text{Rleft}[i]$  to  $\text{Rright}[i]$ 
11       $R_{\text{Raster}}[i][j - \text{Rtop}[i]][k - \text{Rleft}[i]] \leftarrow \text{Raster}[j][k]$ 
12    For  $i = 0$  to  $RH$ 
13      For  $j = 0$  to  $RW$ 
14        If  $R_{\text{Raster}}[i][j] \geq T$ 
15          For each  $R_{\text{Raster}}[u][v] = R_{\text{Raster}}[i][j] + R$ 
16             $\text{houghSpace}[u][v] = \text{houghSpace}[u][v] + R_{\text{Raster}}[i][j]$ 
17    For  $u = R$  to  $RH - \text{Period} \times PR - R$ 
18      For  $v = R$  to  $RW - \text{Period} \times PC - R$ 
19        For  $r = 0$  to  $PR$ 
20          For  $s = 0$  to  $PC$ 
21             $a \leftarrow u + r \times \text{Period}$ 
22             $b \leftarrow v + s \times \text{Period}$ 
23             $\text{score} \leftarrow \text{score} + \text{houghSpace}[a][b]$ 
24          If  $\text{score} > \text{maxscore}$ 
25             $x_0(i) \leftarrow u$ 
26             $y_0(i) \leftarrow v$ 
27    For  $j = 1$  to  $PR$ 
28      For  $k = 1$  to  $PC$ 
29         $\text{target}[i][j][k] \leftarrow (x_0[i] + j \times \text{Period}, y_0[i] + k \times \text{Period})$ 

```

Figure 3.6: Algorithm of the Circular Hough Transform based method.

Table 3.9: Results on S4X3 with the CHT based method.

	Hough	ScanAlyze	δ
A1	(254,257)	(256,215)	42.05
A2	(608,416)	(153,220)	495.42
A3	(254,8)	(263,315)	307.13
A4	(608,41)	(163,314)	522.07

Table 3.10: Results on S4X5 with the CHT based method.

	Hough	ScanAlyze	δ
A1	(251,215)	(256,215)	5.00
A2	(605,416)	(153,220)	492.66
A3	(254,407)	(263,315)	92.44
A4	(605,41)	(163,314)	519.51

but still unsatisfactory. The pin-array A1 has been correctly placed in S4X5 and A3 is off by 4 columns. A2 and A4 pin-arrays are always incorrectly located because of a noisy bottom edge on the experiment S4X. The method is performing slightly better than the Fourier Transform based method. However, the performance are still unsatisfactory for an automatic addressing of a noisy image.

3.4 A Method Based on the Mann-Whitney Test

In this section, we present a method using the Mann-Whitney test to address the targets. The Mann-Whitney test is not presented in this Chapter but is presented in details in section 4.5.1. The following paragraphs present first the principle of our method and secondly the results obtained.

3.4.1 Principle of our Method

The *Mann-Whitney Test* is used to segment the targets by MicroArray Suite. Our motivation was to use it as an optimization function to find the optimum addressing position. We built a template of 16×24 circles of radius 8 evenly spaced by 25 pixels. Each circle representing a potential target mask. We then run the Mann-Whitney Test (MWT) on each circular mask and sum the number of pixels kept by the MWT. The algorithm then iterates over every position the template could possibly fit in the rectangular area where the pin-array is. It keeps the iteration that has the maximum pixels over all targets and induce the offset of the pin-array from the top and left border of the image.

3.4.2 Results

The results shown in Table 3.11 and Table 3.12 show the MWT is not robust enough. None of the pin-array has been located correctly. However, this method seems less sensitive to the noisy left

```

1  $R \leftarrow \text{radius}$  (user specified)
2  $P \leftarrow 4$  (or other number of pin-arrays in the image)
3  $\text{Period} \leftarrow 25$  (or use FFT)
4  $PC \leftarrow 24$ 
5  $PR \leftarrow 16$ 
6  $\text{Raster}[H][W] \leftarrow \text{readTIFF}(\text{Image})$ 
7 For  $i = 0$  to  $P$ 
8   For  $j = \text{Rtop}[i]$  to  $\text{Rbottom}[i]$ 
9     For  $k = \text{Rleft}[i]$  to  $\text{Rright}[i]$ 
10        $R_{\text{Raster}}[i][j - \text{Rtop}[i]][k - \text{Rleft}[i]] \leftarrow \text{Raster}[j][k]$ 
11   For  $u = R$  to  $RH - \text{Period} \times PR - R$ 
12     For  $v = R$  to  $RW - \text{Period} \times PC - R$ 
13        $\text{score} \leftarrow 0$ 
14       For  $r = 0$  to  $PR$ 
15         For  $s = 0$  to  $PC$ 
16            $a \leftarrow u + r \times \text{Period}$ 
17            $b \leftarrow v + s \times \text{Period}$ 
18            $\text{score} \leftarrow \text{score} + \text{MWT}(a, b)$ 
19       If  $\text{score} > \text{maxscore}$ 
20          $\text{maxscore} \leftarrow \text{score}$ 
21          $x_0[i] \leftarrow u$ 
22          $y_0[i] \leftarrow v$ 
23        $(x_0[i], y_0[i]) \leftarrow \text{Adjust}()$ 
24   For  $j = 1$  to  $PR$ 
25     For  $k = 1$  to  $PC$ 
26        $\text{target}[i][j][k] \leftarrow (x_0[i] + j \times \text{Period}, y_0[i] + k \times \text{Period})$ 

```

Figure 3.7: Algorithm of the Mann-Whitney Test based method.

Table 3.11: Results on NS3 with the MWT based method.

	Score	MWT	ScanAlyze	δ
A1	45279	(143,143)	(167,123)	31.24
A2	43151	(38,113)	(63,117)	25.32
A3	40378	(128,173)	(180,224)	72.84
A4	43874	(53,158)	(79,214)	61.74

The “Score” column presents the optimum number of pixels kept by the MWT.

Table 3.12: Results on NS5 with the MWT based method.

	Score	MWT	ScanAlyze	δ
A1	38989	(143,143)	(167,123)	31.24
A2	39142	(38,113)	(63,117)	25.32
A3	34812	(203,173)	(180,224)	55.95
A4	35400	(83,233)	(79,214)	19.42

edge. This noisy left edge has been efficiently avoided and no pin-array is off by more than one row or column. This behavior is however recurrent and the biggest problem comes out of the efficiency of this algorithm. It takes about 40 minutes to grid the 4 pin-array whereas the Fourier Transform based method is executing in a second and the Hough-transform based in about 30 seconds.

The major issue on time performance kept us to spend more time in that direction. Therefore the results on the S4X images are not reported. However, we noticed the method tend to make a good adjustment of the targets and we think once a gridding has been performed, the MWT method could be used to adjust the gridding either at a pin-array or a target level. Figure 3.8 shows the result obtained by MicroArray Suite on a target that is misaligned. We think errors due to target misalignment could be avoided by evaluating the MWT at multiple positions or multiple radii.

Table 3.13: Results on S4X3 with the MWT based method.

	Score	MWT	ScanAlyze	δ
A1	27970	(278,188)	(256,215)	34.83
A2	30220	(128,263)	(153,220)	49.74
A3	27337	(233,308)	(263,315)	30.81
A4	28645	(158,308)	(158,308)	7.81

The “Score” column presents the optimum number of pixels kept by the MWT.

Table 3.14: Results on S4X5 with the MWT based method.

	Score	MWT	ScanAlyze	δ
A1	20176	(278,338)	(256,215)	124.95
A2	20410	(203,263)	(153,220)	65.94
A3	21108	(233,308)	(263,315)	30.81
A4	20154	(158,383)	(163,314)	69.18

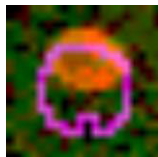


Figure 3.8: Mediocre segmentation on a misaligned target

3.5 Preprocessing of the Images

In this section, we first point out a few problems we noticed in our images and lead us to preprocess the images. We then present the result obtained after histogram equalization and thresholding.

3.5.1 Image Data Analysis

As the previous methods were unable to avoid the left border noisy edge, we investigated methods to reduce the noise by preprocessing the image. We looked more closely at the pixel intensities distribution in order to find a way to turn around this problem. We report the following issues as it is likely to be a common issue for many cDNA microarray images.

The Figure A.9 shows the histogram of our 16-bit image S3 (Cy3 dye). We observe a decreasing exponential in other images too. We believe this is due to *exponential noise*. This type of noise appears to be a typical noise in scanned images and it is difficult to eliminate.

This issue also have shown us our image had a large number of zero values across the entire image as well as a certain number of saturated values at 65535 for highly-expressed cDNA targets. The zero values can be attributed to the exponential noise. The 65535 values are the maximum values that can be obtained in a 16-bit image. They are the expression of a *saturation* effect. The saturation effect is undesirable in particular for the next image processing step, the segmentation. Saturated values are the expression of the limitations of CCD cameras. When using too much power,

the number of photons emitted by the hybridized cDNA increases. Though the targets become more visible on the image, the cells of the CCD camera are limited in the number of photons they can correctly translate. CCD cells are capturing all photons during an integration time period. Above a certain threshold of photons captured during this time period, the CCD cells will transmit the same maximum electric voltage, which results in the 65535 value observed. The cell reached a saturation level. Obviously, the saturation effect is resulting in a truncation of the target intensity information and erroneous data. Calibration of the scanner and careful use of the power voltage is recommended.

3.5.2 Thresholding

By applying an histogram equalization, we were able to assess visually the contrast on our images. The concept of histogram equalization is described in [22]. The principle is to redistribute the pixel intensities to obtain an even distribution of the number of pixels for each intensity from 0 to 65535. This preprocessing technique enhanced the contrast of our images and allowed us to visualize hidden noise. We then were able to notice that target pixel intensities are strongly concentrated in the highest 20% of intensity values, an information that was not contained by the histogram in Figure A.9. We then tried a *tresholding* preprocessing, which consists of keeping only pixels with intensities above or under a predetermined threshold.

The motivation for thresholding is to remove a good part of the exponential noise but keep most of the target intensities. Notice that it also remove the background pixels and therefore the image obtained after tresholding can only be used for addressing the target locations but not for the segmentation or extraction of the data.

3.5.3 Results

We kept only the 20% highest intensities on our image and then resubmitted the images to our different algorithms. Table 3.15 and Table 3.16 presents the result obtained after equalization. On the NS3 image, the results are quite similar. The Fourier method was already performing well and the results after tresholding remain correct. However, the results on the NS5 image are quite improved for the image NS5. The Fourier method in itself still did not avoid the noisy left edge. However, the adjustment method did identify the pin-array A0 quite correctly and estimated the pin-array A1 location with an error of 32 pixels or a little bit more than a column. The tresholding

Table 3.15: Results on NS3 equalized and thresholded with the DFT based method.

	Period	Fourier	Adjustment	ScanAlyze	δ	Prev. Fourier	Prev. δ
A1	(25,25)	(165,119)	(165,119)	(167,123)	4.47	(165,119)	4.47
A2	(25,25)	(64,116)	(64,116)	(63,117)	1.41	(62,116)	1.41
A3	(25,25)	(179,223)	(179,223)	(180,224)	1.41	(182,223)	2.24
A4	(25,25)	(80,209)	(80,209)	(79,214)	5.10	(79,209)	5.00

The “Prev. Fourier” column reminds the result obtained previously without preprocessing.

did also helped in locating the pin-array A4 correctly and got the pin-array location of A2 wrong at 26 pixels instead of 59 pixels with the non-tresholded image.

Figure A.10 shows that preprocessing was beneficial for the pin-array A1 on NS5. We obtain a satisfactory addressing compared to the one obtained with no preprocessing shown in Figure A.4. Figure A.11 and Figure A.12 show however the unsatisfactory results on pin-array A2 and A3. As the noisy edge had not been removed by the preprocessing, we actually noticed that the offset was still not correctly determined before the adjustment function.

The positive result of the preprocessing is actually due to the adjustment function. The function is summing all the pixel intensities in each square over a 25×25 squared 16×24 grid template. The template is moved in every possible positions the grid could fit around the initial position and the algorithm keep the optimal position. It seems that the pixels between the edge and the first column of targets were bringing an important contribution to the sum computed by our adjustment function. This adjustment function is however not robust enough and can fail as shown in Figure A.11 and Figure A.12.

Table 3.17 and Table 3.18 shows the result obtained with the Hough transform based method. The four pin-arrays of image NS3 are yet almost correctly gridded. The pin-array A1 is yet correctly gridded compared to the non-equalized method and only the pin-array A2 is still a row off. However, the gridding of image NS5 has not been improved by the thresholding. The pin-array A1 is still wrongly located because of the noisy left edge and the pin-array A2 and A3 are still a row off.

Table 3.5, Table 3.9, and Table 3.10 present results obtained from the preprocessed images (S4X3,S4X5). Once again the results are slightly improved. The pin-array A1 and A3 are correctly located along the rows dimension but still off along the columns. The results on pin-array A2 and A4 are still very far from a satisfactory results.

Table 3.16: Results on NS5 equalized and thresholded with the DFT based method

	Period	Fourier	Adjustment	ScanAlyze	δ	Prev. Fourier	Prev. δ
A1	(25,25.64)	(164,11)	(164,113)	(167,123)	10.44	(165,12)	111.02
A2	(25,25)	(62,9)	(62,85)	(63,117)	32.02	(61,10)	107.02
A3	(25,25.64)	(154,223)	(154,223)	(180,224)	26.02	(131,223)	49.01
A4	(25,25)	(79,209)	(79,209)	(79,214)	25.50	(54,184)	39.05

Table 3.17: Results on NS3 equalized and thresholded with the CHT based method

	Hough	ScanAlyze	δ	Prev. Hough	Prev. δ
A1	(167,119)	(167,123)	4.00	(167,119)	4.00
A2	(65,116)	(63,117)	2.24	(62,113)	4.12
A3	(155,221)	(180,224)	25.18	(155,221)	25.18
A4	(80,209)	(79,214)	5.10	(80,209)	5.10

Table 3.18: Results on NS5 equalized and thresholded with the CHT based method

	Hough	ScanAlyze	δ	Prev. Hough	Prev. δ
A1	(167,119)	(167,123)	4.00	(167,119)	4.00
A2	(62,14)	(63,117)	103.00	(62,14)	103.00
A3	(155,221)	(180,224)	25.18	(155,221)	25.18
A4	(56,209)	(79,214)	23.53	(56,209)	23.53

Table 3.19: Results on NS3 equalized and thresholded with the MWT based method

	MWT	ScanAlyze	δ	Prev. MWT	Prev. δ
A1	(143,143)	(167,123)	31.24	(143,143)	31.24
A2	(38,113)	(63,117)	25.32	(38,113)	25.32
A3	(128,173)	(180,224)	72.84	(128,173)	72.84
A4	(53,158)	(79,214)	61.74	(53,158)	61.74

Table 3.20: Results on NS5 equalized and thresholded with the MWT based method

	MWT	ScanAlyze	δ	Prev. MWT	Prev. δ
A1	(113,68)	(167,123)	77.08	(143,143)	31.24
A2	(38,113)	(63,117)	25.32	(38,113)	25.32
A3	(128,173)	(180,224)	72.84	(203,173)	55.95
A4	(53,158)	(79,214)	61.74	(83,233)	19.42

Table 3.21: Results on S4X3 equalized and thresholded with the DFT based method.

	Period	Fourier	Adjustment	δ	ScanAlyze	Prev. Fourier	Prev. δ
A1	(25,25)	(252,261)	(252,286)	71.11	(256,215)	(252,286)	71.11
A2	(25.64,25)	(580,390)	(580,390)	459.59	(153,220)	(576,365)	447.16
A3	(25,25)	(258,309)	(258,384)	69.58	(263,315)	(258,22)	293.04
A4	(24.33,25)	(584,334)	(584,384)	426.77	(163,314)	(567,31)	493.26

The “Prev. Fourier” column reminds the result obtained previously without preprocessing.

Table 3.22: Results on S4X5 equalized and thresholded with the DFT based method.

	Period	Fourier	Adjustment	δ	ScanAlyze	Prev. Fourier	Prev. δ
A1	(25,25)	(278,212)	(278,212)	51.89	(256,215)	(278,420)	206.18
A2	–	Fail	–	–	(153,220)	Fail	–
A3	–	–	–	–	–	–	–
A4	–	–	–	–	–	–	–

The “Prev. Fourier” column reminds the result obtained previously without preprocessing.

Table 3.23: Results on S4X3 equalized and thresholded with the CHT based method

	Hough	δ	ScanAlyze	Prev. Hough	Prev. δ
A1	(248,284)	69.46	(256,215)	(254,257)	42.05
A2	(608,416)	495.42	(153,220)	(608,416)	495.42
A3	(263,332)	17.00	(263,315)	(254,8)	307.13
A4	(608,17)	535.00	(163,314)	(608,41)	522.07

Table 3.24: Results on S4X5 equalized and thresholded with the CHT based method

	Hough	δ	ScanAlyze	Prev. Hough	Prev. δ
A1	(248,260)	45.7	(256,215)	(251,215)	5.00
A2	(605,416)	492.66	(153,220)	(605,416)	492.66
A3	(230,407)	97.74	(263,315)	(254,407)	92.44
A4	(611,44)	524.63	(163,314)	(605,41)	519.51

Table 3.25: Results on S4X3 equalized and thresholded with the MWT based method

	MWT	δ	ScanAlyze	Prev. MWT	Prev. δ
A1	(278,263)	52.80	(256,215)	(278,188)	34.83
A2	(128,263)	49.74	(153,220)	(128,263)	49.74
A3	(233,308)	30.81	(263,315)	(233,308)	30.81
A4	(158,308)	7.81	(163,314)	(158,308)	7.81

Table 3.26: Results on S4X5 equalized and thresholded with the MWT based method

	MWT	δ	ScanAlyze	Prev. MWT	Prev. δ
A1	(278,263)	52.80	(256,215)	(278,338)	124.95
A2	(128,263)	49.74	(153,220)	(203,263)	65.94
A3	(263,308)	7.00	(263,315)	(233,308)	30.81
A4	(188,308)	25.70	(163,314)	(158,383)	69.18

3.6 A Hybrid Method

The methods tried so far are not robust enough to find the exact place of the grids and fail because of noise or low expression but in particular because of a noisy left edge of the microslide. In this section, we present first the principle of a method that uses both the FFT and the CHT. We then show the results obtained in a second paragraph.

3.6.1 Principle of our Method

The methods attempted are almost able to automatically grid images (NS3,NS5). The main problem we are confronted with is the noisy left edge of the image. A method that could find a correct region of interest in a first step and would really ignore the noisy edge before refining its decision in a second step seems more likely to work.

We noticed our DFT method never failed to give us a period close to 25. We then used the following approach. In a first step, we use the DFT to find the offsets of the grid. The principle consists of considering small slices of the image large like a target. If we find a slice which period is equal to the period of the pin-array predetermined by the DFT and if a number of the following slices also have the same period as the pin-array, then we can consider these slices are part of the array. By applying this process on rows and columns, the first slices found respectively on the top and on the left should therefore give approximate offsets of the array. We would have therefore determined a region of interest where our grid should be. In a second step, we would use our method based on the circular Hough Transform to adjust the grid position.

```

1  $P \leftarrow 4$  (or other number of pin-arrays in the image)
2 .
3 .
4 .
5 For  $i = 0$  to  $P$ 
6    $p_{M_v}[i] \leftarrow RH / \max\{X_v[k] \mid k = [f_{min}, f_{max}]\}$ 
7    $last \leftarrow RH - p_{M_v}[i] \times PR - R$ 
8    $bool \leftarrow 1$ 
9   For  $off = R$  to  $last$ 
10    For  $row = off$  to  $off + 2 \times R$ 
11      For  $col = 0$  to  $RH$ 
12         $rslice[row - off][col] \leftarrow RectRaster[row][col]$ 
13       $rslicePeriod \leftarrow \max\{FFT(rsliceSum[k]), k = [min, max]\} / Qheight$ 
14      If  $rslicePeriod == p_M[i] \&\& bool == 0$ 
15         $cnt \leftarrow 1$ 
16        For  $k = 0$  to  $PR$ 
17           $off \leftarrow off + k \times p_M[i]$ 
18           $rslicePeriod(off) \leftarrow \max(FFT(rsliceSum(off)))$ 
19          If  $rslicePeriod[u] == p_M[i]$ 
20             $cnt \leftarrow cnt + 1$ 
21          If  $cnt \geq PR - X$ 
22             $x_0[i] \leftarrow off$ 
23             $bool \leftarrow 1$ 
24      Do the same on columns and find  $p_{M_h}[i]$ .
25       $RectRaster \leftarrow rectangle(x_0[i] - p_{M_v}[i], y_0[i] - p_{M_h}[i])$ 
26       $(x_0[i], y_0[i]) \leftarrow HoughTransform(RectRaster)$ 
27      For  $j = 1$  to  $MArrayheight$ 
28        For  $k = 1$  to  $MArraywidth$ 
29           $target[i][j][k] \leftarrow (x_0[i] + j \times p_{M_v}[i], y_0[i] + k \times p_{M_h}[i])$ 

```

Figure 3.9: Algorithm of the hybrid method.

Table 3.27: Results on NS3 with the hybrid method.

	Period	Fourier	Hough	ScanAlyze	δ
A1	(25,25)	(154,106)	(166,119)	(167,123)	4.12
A2	(25,25)	(55,94)	(63,114)	(63,117)	3.00
A3	(25,25)	(193,200)	(181,222)	(180,224)	2.24
A4	(25,25)	(65,187)	(81,209)	(79,214)	5.39

3.6.2 Results

This algorithm succeeded to do an effective gridding of the (NS3,NS5) images after preprocessing. The method missed however to place correctly the pin-array A2 of the not preprocessed NS5 image from 54 pixels (about 2 columns).

This method is based on the assumption that the target spacing is unlikely to be improperly estimated. Unfortunately this assumption appears to be wrong. Table 3.32 shows the period can be wrongly estimated and have values as 66.67. Big artifacts on the (S4X3, S4X5) images may have a local maximum frequency in the appropriate band bigger than the frequency corresponding to the regular spacing of targets. It is also possible that the slices of the image taken fail to identify the correct period because that one may not be strong enough in any slice.

The method suffers from being dependent on the period estimation. In the case of bad images where targets have low expression levels and irregular shapes, we sometime obtained a period that would be twice the expected period of 25. In a number of cases, our implementation may place 2 or 3 pin-arrays accurately as the data in Table 3.33 shows it.

In Figure A.13, we show the results for the bottom left pin-array A2 of the preprocessed NS5 image. All previous algorithm failed on that pin-array because of the noisy left edge. Results on the set of images (S4X3,S4X5) are also better after thresholding. We show the pin-array A1 off by one column placed as in Figure A.14 and pin-array A3 correctly placed in Figure A.15. Figure A.16 shows the artifacts responsible for the systematic misplacement of pin-array A2 and A4 with the previous algorithm. The hybrid method is doing better by correctly identifying the position of the pin-array along the rows. The results are only off by a few columns. Overall, this method obtains much better results, particularly after preprocessing, than previous methods but is dependent on a good period estimation.

Table 3.28: Results on NS5 with the hybrid method.

	Period	Fourier	Hough	ScanAlyze	δ
A1	(25,25.64)	(169,110)	(165,119)	(167,123)	4.47
A2	(25,25)	(66,96)	(60,63)	(63,117)	54.08
A3	(25,25)	(208,204)	(181,222)	(180,224)	2.24
A4	(25,25)	(66,196)	(81,209)	(79,214)	5.38

Table 3.29: Results on NS3 equalized and thresholded with the hybrid method.

	Period	Fourier	Hough	δ	SA	P. Fourier	P. Hough	P. δ
A1	(25,25)	(153,99)	(167,119)	4.00	(167,123)	(154,106)	(166,119)	4.12
A2	(25,25)	(58,93)	(64,115)	2.23	(63,117)	(55,94)	(63,114)	3.00
A3	(25,25)	(189,202)	(181,222)	2.23	(180,224)	(193,200)	(181,222)	2.24
A4	(25,25)	(64,187)	(81,209)	5.38	(79,214)	(65,187)	(81,209)	5.39

SA: ScanAlyze results

Table 3.30: Results on NS5 equalized and thresholded with the hybrid method.

	Period	Fourier	Hough	δ	ScanAl.	P. Fourier	P. Hough	P. δ
A1	(25,25.64)	(168,102)	(166,118)	5.10	(167,123)	(169,110)	(165,119)	4.47
A2	(25,25.64)	(58,94)	(63,114)	3.00	(63,117)	(66,96)	(60,63)	54.08
A3	(25,25)	(195,203)	(181,222)	2.23	(180,224)	(208,204)	(181,222)	2.24
A4	(25,25)	(65,251)	(80,209)	5.10	(79,214)	(66,196)	(81,209)	5.38

P. states for the previous results with the original image.

Table 3.31: Results on S4X3 with the hybrid method.

	Period	Fourier	Hough	ScanAlyze	δ
A1	(25,25)	(238,196)	(250,233)	(256,215)	18.97
A2	(25,25)	(137,39)	(155,63)	(153,220)	157.01
A3	(25,25)	(248,19)	(254,57)	(263,315)	258.16
A4	(25,25)	(145,320)	(164,307)	(163,314)	7.07

Table 3.32: Results on S4X5 with the hybrid method.

	Period	Fourier	Hough	ScanAlyze	δ
A1	(25,24.39)	(242,0)	(252,8)	(256,215)	207.03
A2	(66.67,32.26)	(168,0)	Fail	(153,220)	–
A3	–	–	–	–	–
A4	–	–	–	–	–

Table 3.33: Results on S4X3 equalized and thresholded with the hybrid method.

	Period	Fourier	Hough	δ	SA	P. Fourier	P. Hough	P. δ
A1	(25,25)	(238,197)	(250,232)	18.00	(256,215)	(238,196)	(250,233)	18.97
A2	(25,25)	(137,39)	(146,55)	165.10	(153,220)	(137,39)	(155,63)	157.01
A3	(25,25)	(248,297)	(263,311)	4.00	(263,315)	(248,19)	(254,57)	258.16
A4	(25,25)	(146,325)	(165,361)	47.00	(163,314)	(145,320)	(164,307)	7.07

Table 3.34: Results on S4X5 equalized and thresholded with the hybrid method.

	Period	Fourier	Hough	δ	SA.	P. Fourier	P. Hough	P. δ
A1	(25,25)	(267,0)	(251,8)	207.06	(256,215)	(242,0)	(252,8)	207.03
A2	(66.67,25)	(171,0)	Fail	–	(153,220)	(168,0)	Fail	–
A3	–	–	–	–	–	–	–	–
A4	–	–	–	–	–	–	–	–

3.7 Discussion

We have made progress toward automation but none of the algorithms presented is robust enough for bad images. The DFT based and CHT based method perform quite equally. It is important to notice our CHT based method did not exploit all the potential of the HT (no use of the gradient image) and we did not try the Linear HT. The preprocessing brought improvements in the addressing and appear to be a necessary step. The hybrid method was able to correctly address the pair of preprocessed image (NS3,NS5) and go over the problem of the noisy edge but it failed to address pin-arrays on images containing big noisy areas as shown in Figure A.14.

Improvements on these methods can be made. A 2-dimensional DFT could be more efficient at locating the grid since this DFT would take into account periodicity in both dimensions. The Circular Hough Transform implemented did not use the gradient image and though we are unsure the results would be better, it is an easy improvement to implement for this method. If the CHT did not give us the best results, it always obtained results and never failed as the DFT based methods which are dependent on a good period estimation.

We continue to investigate other methods. In the context of our study, Paul Ignatius Echevarria and Jerome Punzalan, two undergraduate students at Ateneo de Manila University, Phillipines implemented an other method using morphological operators and considering the tilting effect. This method at least located the pair of images (NS3,NS5) correctly.

Table 3.35: Summary Table on the distances obtained for the NS3 image.

Pin-Array	δ							
	DFT		CHT		MWT		Hybrid	
	NT	T	NT	T	NT	T	NT	T
A1	4.47	4.47	4	4	31.24	31.24	4.12	4
A2	1.41	1.41	4.12	2.24	25.32	25.32	3	2.23
A3	2.24	1.41	25.18	25.18	72.84	72.84	2.24	2.23
A4	5	5.10	5.10	5.10	61.74	61.74	5.39	5.38
Mean	3.28	3.10	9.6	9.13	47.78	47.78	3.69	3.46

NT : Image non-thresholded ; T: Image thresholded

Table 3.36: Summary Table on the distances obtained for the NS5 image.

Pin-Array	δ							
	DFT		CHT		MWT		Hybrid	
	NT	T	NT	T	NT	T	NT	T
A1	111.02	10.44	4	4	31.24	77.08	4.47	5.1
A2	107.02	32.02	103	103	25.32	25.32	54.08	3
A3	49.01	26.02	25.18	25.18	55.95	72.84	2.24	2.23
A4	39.05	25.50	23.53	23.53	19.42	61.74	5.38	5.1
Mean	76.52	23.50	38.92	38.92	32.98	59.25	16.54	3.86

NT : Image non-thresholded ; T: Image thresholded

We believe the following algorithms will be implemented in the future. The connected component algorithm of Rosenfeld [36] first surveys aggregation of targets that are quite uniform in the image and then eliminates all aggregation that do not have a satisfactory number of pixels (too small or too large). The use of a shrinking algorithm could be an other preprocessing operation implemented to eliminate noise and small artifacts.

Table 3.37: Summary Table on the distances obtained for the S4X3 image.

Pin-Array	δ							
	DFT		CHT		MWT		Hybrid	
	NT	T	NT	T	NT	T	NT	T
A1	71.11	71.11	42.05	69.46	34.83	52.80	18.97	18.0
A2	447.16	459.59	495.42	495.42	49.74	49.74	157.01	165.1
A3	293.04	69.58	307.13	17	30.81	30.81	258.16	4
A4	493.26	426.77	522.07	535	7.81	7.81	7.07	47.0
Mean	326.14	256.76	341.67	279.22	30.79	35.29	110.30	58.5

NT : Image non-tresholded ; T: Image thresholded

Table 3.38: Summary Table on the distances obtained for the S4X5 image.

Pin-Array	δ							
	DFT		CHT		MWT		Hybrid	
	NT	T	NT	T	NT	T	NT	T
A1	206.18	51.89	5	45.7	124.95	52.80	207.03	207.06
A2	Fail	Fail	492.66	492.66	65.94	49.74	Fail	Fail
A3	-	-	92.44	97.74	30.81	7	-	-
A4	-	-	519.51	524.63	69.18	25.70	-	-
Mean	-	-	277.40	290.18	72.72	33.81	-	-

NT : Image non-tresholded ; T: Image thresholded

Chapter 4

Segmentation

Segmentation is the process of partitioning an image into its constituent parts. The segmentation step of microarray image analysis plays a major role in the downstream data analysis as it is the step where the data is generated. While the gridding of targets can be done manually or semi-automatically, segmentation can not and is always done with automated segmentation techniques. Our primary concern is to evaluate the performance of the segmentation algorithms already in use. Ultimately we wish to identify a reliable means to segment in order to have more confidence in the extracted data.

After defining the segmentation problem in section 4.1, we provide a classification of segmentation techniques already used in section 4.2. We present the fixed circle segmentation and its use in the program ScanAlyze in section 4.3. We have implemented the SRG and MWT, two algorithms more sophisticated than the simple approach of the fixed circle segmentation and have examined their performance. We present our results with the SRG in section 4.4 and the MWT in section 4.5. We discuss the performances in section 4.6 and future work in section 4.7.

4.1 The Segmentation Problem

In the context of microarrays, the segmentation consists of identifying the target from the background. In section 3.1, we introduced the notion of a target patch. A target patch is a small area expected to delineate a target and its surrounding pixels after gridding of a pin-array. A target patch however does not really isolate the target from the background.

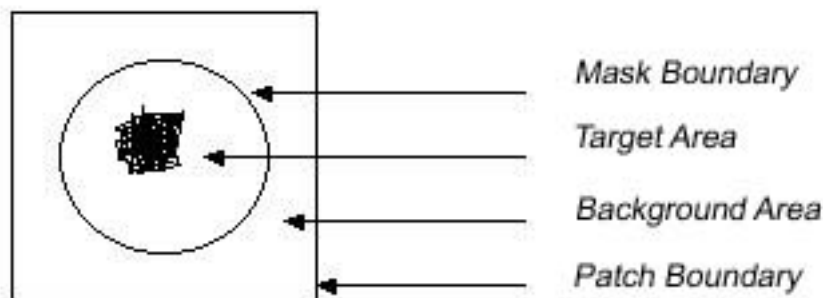


Figure 4.1: A target patch.

In the case of a perfect experiment, targets should be circular. One approach is therefore to take a circle of radius R in the center of the patch as the *target boundary*. The circle and its inside area define a region called the *target mask* TM (see Figure 4.1). However, as targets have various shapes and sizes, this approach is inadequate. Indeed the target mask may contain pixels that really belong to the background or on the contrary part of the target may be outside the target mask (circle) as for a big target or a misaligned target for instance.

To improve the accuracy of extracted data, the segmentation needs to go further than the naive approach. It has to remove background pixels from the naive target mask (circle) and add target pixels initially in the background mask. The goal is to obtain a target mask that perfectly matches the target. We call the *target site*, the final target mask. *Target pixels* are any pixels inside the target mask and its boundary. A *background pixel* is any pixel outside the target mask but inside the target patch. The set of all the background pixels constitutes the *background mask* BM . We illustrate these definitions in Figure 4.1.

Whether the *target site* needs to be connected or not is controversial. Some targets are made of disconnected areas. In this case, not allowing the target site to be disconnected is critical as a part of the target is left by the segmentation as part of the background. On the contrary, allowing the target site to be disconnected has the disadvantage of introducing an opportunity for considering small artifacts and surrounding noise as target pixels. In the context of this controversy, it is interesting to examine and classify the different segmentation techniques already in use.

In the future, a target is defined by a quadruple ($pin - array, row, column, dye$). The quadruple (1, 7, 4, Cy3) corresponds to the target of the pin-array A1 at the 7th row and the 4th column of the image resulting from the scan in the Cy3 dye wavelength.

4.2 A Classification of Some Segmentation Techniques

Dudoit et al. [21] categorized some segmentation methods in four groups according to the geometry of the targets they produce. The classification groups are the followings:

1. *fixed circle*
2. *adaptive circle*
3. *adaptive shape*
4. *histogram-based*

The fixed circle segmentation technique is used in *ScanAlyze*, a program written by Mark Eisen [10]. It corresponds to naive segmentation using a square target patch with a circular target mask. We illustrate this technique by presenting *ScanAlyze* in section 4.3.

The adaptive circle technique consists of adjusting automatically the radius of each circular target mask. This technique is then able to adapt to various target sizes in case they are quite circular. The technique is used by *Genepix*. Unfortunately we do not have access to *Genepix* or any program performing this kind of segmentation.

To our knowledge two references only detail the segmentation technique they use. Chen et al. [23] proposed a segmentation technique based on the *Mann-Whitney test*(MWT) [25, 26]. Based on the Wilcoxon rank-sum, the MWT is a standard distribution-free statistical method used to differentiate two populations. The MWT is implemented in the program *MicroArray Suite* (MS). Dudoit et al. [21] categorized MS in the histogram segmentation group. The histograms are actually used as a data calibration tool. The actual segmentation is made by the MWT. This statistical test is potentially able to adapt to the size and shape of the targets. Therefore we prefer to categorize the MWT as an adaptive shape segmentation technique.

The second segmentation technique we identified in the literature has been proposed more recently by Dudoit et al. [21]. They presented a segmentation technique based on the *seeded region*

growing algorithm of Adams and Bischof [24], later improved by Mehnert and Jackway [27]. The technique is used in the program *Spot* [29] and falls in the category of the adaptive shape segmentation technique. This study looked at the performance of the two adaptive shape segmentation techniques: the SRG and MWT.

Dudoit et al. [21] classifies *QuantArray* as an histogram-based technique. *Quantarray* defaults the background mean to be the mean of the 5th and 20th percentiles of the histogram and the target mean to be the mean of the 80th to 95th percentiles.

4.3 Fixed Circle Segmentation

ScanAlyze [10] was developed by Mark Eisen in 1998-1999 at Stanford University. It supports 8 and 16-bit TIFF images and provides a semi-automated gridding. ScanAlyze can generate different grid sizes but the user must interactively place the grid. Target positions and diameters can be *adjusted* manually but Dudoit et al. [21] do not categorize ScanAlyze as an adaptive circle segmentation technique because of the manual intervention required. However, after the user has placed a grid, ScanAlyze can optimize the gridding or particular target positions through a refine option that uses a Sobel operator. ScanAlyze also provides the option to rotate the grid and manage the tilting effect.

In terms of segmentation, ScanAlyze uses a *fixed circle segmentation*. All pixels inside a circle constitutes what is called the *target mask*. All these pixels are then part of the target regardless of their actual intensity. The background contains every external pixels that is not in the target mask and is in a square area for which the radius can be user-defined. The radius defaults to 20 and these settings generate background areas containing about 1300 pixels. Notice that background areas are significantly larger than the ones determined by MicroArray Suite (approximately 400 pixels).

ScanAlyze provides several ratio estimates in addition to a quality control parameter. The first ratio RAT2 is called an *uncorrected mean ratio*. Let μ be the mean of a population and θ its median. For a particular target mask s_i and the corresponding background regions bk_i , where i represents the channel, RAT2 is:

$$RAT2 = \frac{Ch_2(red)}{Ch_1(green)} = \frac{\mu_{s_2} - \theta_{bk_2}}{\mu_{s_1} - \theta_{bk_1}} \quad (4.1)$$

It is argued in the ScanAlyze manual that the median is a good estimator for the background region if we assume a uniform distribution for the background pixels. However, the median is a bad estimator for targets sites, as the amount of DNA across a target cannot be assumed to be uniformly distributed. Therefore the manual advises the use of the mean for targets and median for background. However, the target mean is susceptible to inaccuracies due to noise or artifacts.

A second estimation of the ratio is given by the median of the background-corrected pixel ratios. ScanAlyze computes, for every pixel x in the target mask with the intensities i_1 and i_2 in the respective channel 1 and 2, the following formulae called the background corrected pixel ratio:

$$\frac{Ch'_2}{Ch'_1}(x) = \frac{i_2 - \theta_{bk2}}{i_1 - \theta_{bk1}} \quad (4.2)$$

$$(4.3)$$

The final ratio exported as MRAT in the ScanAlyze output is the median of all the background corrected pixel ratios that is:

$$MRAT = \theta(A) \text{ where } A = \left\{ \frac{Ch'_2}{Ch'_1}(x) \mid \forall x \in target \right\} \quad (4.4)$$

$$(4.5)$$

ScanAlyze provides 2 other estimators based on linear regression and least-square minimum as well. They are based on the assumption that a plot of Channel 2 pixel intensity against the corresponding pixel intensity in Channel 1 will fall approximately on a line of slope equal to the ratio. The slope estimated by *linear regression* or *least-square minimum* provides the two estimators exported as REGR and LRAT in the ScanAlyze output. ScanAlyze also provides estimates of the quality of a target. *Correlation* between target and background, fraction of pixels in the target greater than the background or values of the *Kolmogorov-Smirnov* statistic to identify weak targets are the estimates provided.

The method is simple to reproduce and yields results that are identical when the gridding and target adjustment are correctly done. This segmentation method is naive assuming a perfect target and therefore includes extra pixels.

4.4 Seeded Region Growing Algorithms (SRG)

Adams and Bischof [24] present the seeded region growing algorithm (SRG) as a robust and fast segmentation technique that is parameter free. However, the algorithm is dependent on the order

in which pixels are processed. Mehnert and Jackway [27] present an improved algorithm which is order-independent. The following section presents issues encountered with our implementation of the SRG of Adams and Bischof. The section is organized as follows. After a presentation of the algorithm, we introduce the features of our implementation. We demonstrate that the seed choice is critical in size and location and present improvements resulting in a more appropriate seed choice.

4.4.1 Principle of the SRG Algorithm

The Adams and Bischof algorithm relies on the assumption that the pixels within a region (e.g target or background) are quite similar. Therefore, for regions that are small and of very similar intensity, the dependence on order is undesirable. Mehnert and Jackway [27] propose an alternate seeded region growing algorithm we are not discussing, which has the same advantages as the original one but also is pixel *order independent*. We are conscious our implementation is order-dependent but are unsure whether the problem described below can be addressed with the Mehnert and Jackway approach.

The SRG algorithm uses a small set of pixels, called *seeds*, as the initial points of a region. Each region is assigned a unique *label*. The seeds for a single region can be of various sizes and do not need to be connected. At each iteration the algorithm will consider simultaneously the neighbors of every region grown from a seed. These neighbors are stored in a sorted linked list(SSL) for efficiency reasons. They are sorted in increasing order by a criterion δ , the distance of a pixel intensity to the mean intensities of the neighboring region under the assumption that the noise is of equal variance. Therefore for a pixel x of intensity $I(x)$, neighbor of a region U of N pixels y , we have :

$$\delta(x) = | I(x) - 1/N \sum_{y \in U} I(y) | . \quad (4.6)$$

Other criteria may be used if the assumption of equal variance of the noise is not justified [24]. The algorithm iterates until all pixels have been assigned to a region or labelled as a *boundary pixel*. At each iteration, the first element in the list L , or the neighboring pixel z the closest to a neighboring region U is considered such that $\delta(z) = \min\{\delta(x) \mid x \in L\}$. The algorithm then labels z as a region or as a boundary pixel between two regions. The SRG algorithm will decide to add z to the region if the only neighboring pixels of z that are already labelled are the pixels of a single region. Therefore if z has only one neighbor that belongs to an other region, z is marked as

```

1   Label seed points according to their initial grouping
2   Put neighbors of seed points (the initial T) in the SSL
3   While the SSL is not empty
4       Remove first point  $y$  from the SSL
5       Test the neighbors of this point:
6       If all neighbors of  $y$  which are already labelled
          (other than boundary label) have the same label -
7           Set  $y$  to this label
8           Update running mean of corresponding region
9           Add neighbors of  $y$  which are either already set nor already in the SSL
          to the SSL according to their value of  $\delta$ 
10      Otherwise
11          Label  $y$  with the boundary label.

```

Figure 4.2: The Seeded Region Growing Algorithm .

boundary pixel. We reproduced in Figure 4.2 the pseudo-code of the algorithm as it is in Adams and Bischof [24].

4.4.2 Adaptation of the SRG Algorithm to the Segmentation Problem

Recently, Dudoit et al. [21] used the SRG algorithm for their package *Spot*. They choose the *target seeds* as $n \times n$ square regions centered on the maximum intensity pixel in the square regions obtained by the gridding. *Background seeds* are *crosses* at the intersection of the fitted grids. Dudoit et al. [21] grow every background and foreground regions simultaneously. They argue their seed choice allows a local estimation of the background and prevents each target from bleeding into another one [29, 21].

To gauge the performance of this algorithm, we implemented our own seeded region growing algorithm. Our initial seeded region growing implementation is equivalent to the one developed by Dudoit et al. in the following ways. The foreground seeds are small $n \times n$ square region centered on the pixel of maximum intensity in the target patch (n specified by the user). Background seeds are n -wide crosses at the fitted grid intersection points. By growing all targets simultaneously, this initial implementation exhibited a catastrophic behavior, as we typically experienced the *bleeding* of one background seed over the whole image. This background region will be grown first and target regions were almost not grown. Though our implementation may be different from the *Spot* one, we are skeptical about the appropriateness of the Dudoit et al. [21] seed choice. We are not convinced

it really avoids the bleeding effect in the case of a mediocre image where targets are not highly expressed and of non-uniform intensity.

In a second implementation, we considered a 25×25 target patch and executed the SRG algorithm on one target at a time. This eliminates the bleeding effect and still provides enough background pixels to do a local estimation (as much as in MicroArray Suite in fact). Our *background seed* is comprised of four starting half-crosses in the corners of our squared image. No information is given by Dudoit et al. on the criteria δ they used or whether they implemented the improved version of Mehnert and Jackway or the original version of Adams and Bischof. We implemented the order-dependent version of Adams and Bischof with a δ using the previous formula: $\delta(z) = \min\{\delta(x) \mid x \in L\}$. Other possible differences between the two implementations are the following. On the first hand, the Spot implementation seems to be applied to the combined image. The seed choice is therefore susceptible to be made on the combined image whereas our implementation chooses a seed for each channel. The Spot implementation may also grow the regions from the combined image whereas we grow a region for both channels and take the union of the two regions to compute our results.

4.4.3 A Critical Seed Choice

Dudoit et al. [21] mention that poor performance is expected if the region is not homogeneous in intensity. This case actually occurs quite often in mediocre images, and this issue raises a major doubt on the utility of this algorithm. Indeed, we know targets can have a mountain, doughnut, crescent or other shape. In this section, we show the influence of the seed choice in the results obtained for particular targets. We present results obtained with different seed choice(s).

4.4.3.1 Seed size sensitivity

We use the term “*beignet*”¹ for the type of target resembling to the target (1, 7, 4, Cy3) of our experiment S3 and shown in Figure B.1. This target has a circular shape but does not have a good uniformity of intensities. Though the target is comprised of high intensity values that make it easily distinguishable from the background, there are big discrepancies between the target pixel values. Table 4.1 presents the distribution of intensities in this target after the SRG was run with a seed size

¹“Beignet” is the french term for a pastry that is circular, of a uniform color and does not have a hole like the doughnut. It is often filled with marmelade.

Table 4.1: Intensities of the *beignet* (seed size = 2)

R	C	10				15				20			
6	B:428	B:2s	B:2s	B:83	B:0	B:0	B:197	B:0	B:974	B:1s	B:3s	B:439	B:50
	B:267	B:2s	B:0	B:0	B:22	B:572	B:505	B:2s	B:0	B:45	B:1s	B:330	B:0
	B:909	B:1s	X:79	X:1s	X:1s	X:292	B:0	B:428	B:1s	B:3s	B:5s	B:1s	B:3s
10	B:152	B:472	X:2s	S:6s	S:45s	X:50s	X:54s	X:44s	X:61s	X:52s	B:17s	B:4s	B:0
	B:2s	B:110	X:3s	S:45s	S:65s	S:52s	S:53s	S:43s	S:53s	X:65s	B:43s	B:6s	B:630
	B:88	B:1s	X:17s	S:16s	S:21s	S:22s	S:23s	S:24s	X:16s	X:13s	B:26s	B:31s	B:38s
	B:190	B:9s	X:56s	X:22s	X:20s	X:11s	X:19s	X:25s	X:24s	B:18s	B:35s	B:15s	B:37s
	B:1s	B:14s	B:48s	B:18s	B:20s	B:18s	B:14s	B:10s	B:15s	B:13s	B:14s	B:17s	B:25s
15	B:2s	B:10s	B:36s	B:30s	B:32s	B:20s	B:19s	B:8s	B:9s	B:26s	B:8s	B:13s	B:10s
	B:1s	B:5s	B:37s	B:25s	B:22s	B:20s	B:18s	B:24s	B:18s	B:25s	B:19s	B:30s	B:30s
	B:46	B:4s	B:49s	B:24s	B:18s	B:24s	B:19s	B:12s	B:12s	B:20s	B:24s	B:21s	B:39s
	B:40	B:156	B:2s	B:30s	B:62s	B:54s	B:41s	B:35s	B:28s	B:37s	B:25s	B:21s	B:51s
	B:0	B:3s	B:1s	B:20s	B:55s	B:65s	B:56s	B:45s	B:57s	B:35s	B:37s	B:46s	B:14s
20	B:0	B:0	B:674	B:139	B:41	B:1s	B:5s	B:6s	B:9s	B:8s	B:9s	B:1s	B:306
	B:940	B:2s	B:1s	B:351	B:974	B:1s	B:4s	B:5s	B:21s	B:930	B:1s	B:1s	B:88
	B:1s	B:886	B:832	B:1s	B:1s	B:1s	B:3s	B:872	B:1s	B:475	B:195	B:2s	B:2
	B:0	B:0	B:0	B:715	B:352	B:0	B:0	B:1s	B:3s	B:235	B:379	B:786	B:18

Each pixel is defined by a couple “Label:Intensity”. The labels B, X and S correspond to the label assigned by our SRG algorithm as for background, boundary, or target. A pixel intensity between 61 000 and 61999 is denoted 61s.

of 2. The target is comprised of a pixel area of high-intensity ranging from the 45s to the 60s. Note that this area is also close to the boundary of the target. We then observe a jump in the intensities and most of the target pixels intensity have values comprised in the 20s except for another small high-intensity regions.

We obtained various results by changing the seed size on a certain number of targets with the same characteristics as the previous *beignet*. Figure B.2 and Figure B.3 show the results obtained for a seed size of 2 and 3 on our *beignet*. Though the initial seed is in the same location, the results are radically different.

One undesirable behavior occurs when a high-intensity pixel may be isolated near the edge of the target but right next to background pixels. In that case the seed choice can include background pixels. We believe this unfortunate effect may actually be the reason why the seed size 3, by including more background pixels, was able to grow a region larger than the high-intensity region. We explain in the next paragraphs how the algorithm obtained the result of the target (1, 7, 4) with an initial seed size equal to 2 shown in Figure B.2.

We show first the initial *labelling* reflecting the seed choice for the target and background. The target seed is centered on the pixel (10,12,65535), a saturated pixel. This target has only 3 pixels that have the saturation value 65535. We took care to choose a target that is relatively circular and sufficiently expressed so that it is easily distinguishable from the background. The initial seed region consist of the set of pixels (9, 11, 6985), (9, 12, 45807), (10, 11, 45931), (10, 12, 65535).


```

BBOB00BB0BB00BBBBB00000BB
BBBB00B0B00B00BB0B0B0BB00B
0000B0BB000BB0BBB0000B0BB0
00B0BBBBB0B0B0B00BB000B00B
00B0000B000BB00BB00B0B0B0
00BB00B00000B0BB0B00B00B0
00BB000B0000BB0000BB0000B0
00BB0BBB00000BB0000BB0000
BBBBBB0BB0SS0000000BBBBBBB
BB000BB000SS0000000B0BBBB0
0BB00000B0000000000000B00BB
00BB0BB000000000000000BBBBB
0B0BBB0B000000000000000BBBBB
0B00B000000000000000000B0BB
B000B0BB000000000000000B0BBB
0B0B00BB000000000000000BBBBB
0BB00B0BB000000000000000BBBBB
0BB00B000000000000000000BBB0
0B00B0B00BBBB00000BBBBBBBBB
BBBBB0BB0BBB000BBBBBBBBBBB
00BB0BBBBBB00BBBB0BB0B0BB
00BB00000BBBB00BBBBBBBBBBB
00BB0B0BBB0BBBBBBBBBBBBB00
B00000BBB00B0B0BB0BBB0000B
B00BBBBBB0B0B0BBBBBB0BB0B
B0BBB0BBB000BBBBBBBB000BB

```

Figure 4.4: Labels at Iteration 300 of the SRG on the beignet.

The SRG starts growing the background first as for any target. This behavior is a consequence of the small values and differences among most background pixels. After a few hundred iterations, the target region though not yet labelled is distinguishable. We show the layout of our labelling at iteration 300 (Figure 4.4) and 545 (Figure 4.5). In the last one, we can recognize the target that has not yet been labelled. It is easy to check that the pixel values not labelled are perfectly matching the target.

Let us see now how the algorithm fails from that iteration. Table 4.2 shows that pixels of intensities in the range of 16,000 to 20,000 are labelled as background. A look at the labelling layout in Table 4.1 allows one to understand how it happens. The seed region has a very high mean (in the 40,000) while many of its neighboring pixels are in the 15,000 and 20,000. This unfortunate barrier of middle range targets delays the growing of the target region to the profit of the background. Indeed,

```
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBB0000BBBBBBBBBBBBBBB
BBBBBBBBB0SS000000BBBBBBBBBB
BBBBBBBBB0SS000000BBBBBBBBBB
BBBBBBBBB000000000000BBBBBBB
BBBBBBBBB00000000000000BBBBB
BBBBBBBBB00000000000000BBBBB
BBBBBBBBB00000000000000BBBBB
BBBBBBBBB00000000000000BBBBB
BBBBBBBBB00000000000000BBBBB
BBBBBBBBB00000000000000BBBBB
BBBBBBBBB00000000000000BBBBB
BBBBBBBBB00000000000000BBBBB
BBBBBBBBB00000000000000BBBBB
BBBBBBBBBBBBBBBBB0000BBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBBB
```

Figure 4.5: Labels at Iteration 545 of the SRG on the beignet.

Table 4.2: Table of the SRG Iterations 545 to 571 on the beignet.

Iteration	Target	δ	Pixel Int.	Label	Mean of the Region
545	(1,18)	6268		B	
546	(11,20)	6313	6959	B	646
547	(18,16)	7345	8022	B	677
548	(11,8)	8407	9297	B	890
549	(18,17)	8858	9535	B	677
550	(18,15)	9013	9690	B	677
551	(8,12)		50698	X	42991.2
552	(15,20)	9790	10442	B	652
553	(13,8)	10104	14555	B	4451
554	(9,12)	11661	52726	S	41065
555	(9,13)	8634	53248	S	44614
556	(9,14)	2300	43547	S	41247
557	(9,15)	8101	53661	S	45560
558	(12,8)	13665	14555	B	890
559	(17,19)	14113	14795	B	682
560	(8,17)	16840	17320	B	480
561	(10,12)	18536	22528	S	3992
562	(9,16)		46018	X	
563	(17,10)	19444	20230	B	786
564	(14,20)	19509	20161	B	652
565	(13,19)	9659	10794	B	1135
566	(13,18)	12014	13166	B	1152
567	(13,17)	7318	8491	B	1173
568	(12,16)	12096	13282	B	1186
569	(13,15)	7972	9179	B	1207
570	(13,14)	7193	8414	B	1221
571	(12,14)	8998	10219	B	1221

a neighboring pixel in the 15,000 is closer to the background mean in the range of 500 than it is from the 40,000. When the background is starting to add pixel intensities in the range of the 6,000 and 10,000, the mean of the background region is increasing. The mean is then closer to target pixels in the 15s, the next kind of target to be added and so on. Iteration 571, shown in Figure 4.6, shows that the background region is growing a path into the target by the lowest values of the target.

This result is a consequence of a small high-intensity region of the target. As this region is grown, it maintains a very high mean, and the nearby middle-range intensity pixels are added to the list with a big value of delta. These middle-range intensity targets will only be examined very

BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBB0000BBBBBBBBBBBBBB
BBBBBBBBB0SSX0000BBBBBBBBBB
BBBBBBBBB0SSSSSX0BBBBBBBBBB
BBBBBBBBB000S0000000BBBBBBB
BBBBBBBBB00000000000BBBBBBB
BBBBBBBBB00000B0B0000BBBBBB
BBBBBBBBB00000BB0BB0BBBBBBB
BBBBBBBBB000000000000BBBBBBB
BBBBBBBBB000000000000BBBBBBB
BBBBBBBBB000000000000BBBBBBB
BBBBBBBBB000000000000BBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB
BBBBBBBBBBBBBBBBBBBBBBBBBBBB

Figure 4.6: Labels at Iteration 571 of the SRG on the beignet.

late. These targets create a “barrier” to the expansion of the target region allowing the background to insert and add neighboring pixels with high intensity but a lower δ . A target with a big δ can still be added to a region it clearly does not happen to be a member of. The pitfall of the SRG is that its decision to add a pixel to a region is based on the label of surrounding pixels. If all of them are background pixels, then regardless of the value of the considered pixel it will insert it in the background region. This undesirable effect is occurring quite often when we have a local high-intensity region picked as the initial seed. It is a known fact that targets do not have uniform pixel values. Therefore this behavior is occurring often for highly-expressed targets that have a wider distribution of intensities and especially in saturated images. We argue the problem lies in considering the maximum intensity as the initial seed. If we take a lower value but in the middle of the range of the target intensity values, this problem should not occur. We avoid considering a particular region of very high-intensity but this region will still be grown by a seed region starting with a lower average since the background should still create bigger δ 's.

4.4.3.2 Seed Location

The location of the seed also has a crucial effect on the result of the SRG algorithm. We experimented with different methods to choose the target seed.

If n is the seed size, we call a “Maximum” seed, an $n \times n$ area centered on the maximum intensity pixel in the entire target patch. This seed choice is the one used by Dudoit et al. [21]. We call a “Maximum Mean” seed, an $n \times n$ square area of *maximum mean* in the target patch. This seed choice can be justified by our willingness to avoid taking a high-intensity noise pixel as the initial seed. Indeed, Figure 8 in [21] shows some initial seeds are in extreme positions of the target patch and away from the center of the target patch. We noticed this behavior in our implementation too and remarked that the “Max” seed in these cases is often a high-intensity noise pixel disconnected from the target.

We call a “Center” seed an $n \times n$ area centered on the *maximum intensity pixel in a $s \times s$ square area S* where s is small and S is in the center of the target patch. The principle is again to restrict the target seed to be close to the center of the target patch. We show in Figure B.4 to Figure B.9 the different results that have been obtained for different targets with different seed choices.

The Figure B.4 to Figure B.9 show that neither method is better than the other. Depending on the shape and intensities of a target, either of the methods may work better. For some targets,

the result will always be unsatisfactory. Doughnut-shaped targets are the hardest to segment. We assume this shape is occurring when the pin is coming too close to the glassslide to drop the material. Therefore the material may be spread out all around. Justifying this assumption can be accomplished by direct experimentation. As Figure B.4 shows, doughnut-shaped targets are typically badly segmented by the SRG algorithm when taking the maximum intensity as an initial seed. Taking a centered seed, as in Figure B.5 yields better results for this type of targets. In the case of a doughnut-shaped target, a central seed will start growing a region whose mean is below the high-intensity values. The δ of these high-intensity values will therefore be closer to this mean than the background ones. Unfortunately if the crown has a hole and we get a target in the shape of a “Croissant”, the background seed will step into the target area as in section 4.4.3.1.

4.4.4 Improve the Seed Choice

The SRG algorithm behavior is unreliable as it is seed-size and seed-location dependent. These examples shows the SRG performance are unsatisfactory in many cases. While the seed choice is critical in size and location, the SRG can be very efficient at segmenting target in a few cases.

A good segmentation technique is expected to perform well on all targets except a few. We argue the seed choice must be adapted to the different target shape. However, most of the time no *a priori* knowledge of a target shape is initially available. Improving the SRG requires finding an appropriate seed choice that would adapt to the various targets. We present in the following section *seed choices* we tried to improve the SRG performance.

4.4.4.1 A random and unconnected seed

Dudoit et al. [21] choose a square region centered on the maximum intensity pixel in the target patch. This seed choice is not appropriate for multiple reasons. First, it can choose a small artifact or noise as the initial target seed. Second, it can also pick a target seed in a region of very high intensities that does not represent the range of all intensities in the target. It would then only grow a region of very high intensity and fail to grow the rest of the target as shown previously. Figure B.10 to Figure B.19 show some results for this seed choice.

We tried to choose as a target seed *random* pixels that satisfy the two following criteria:

1. Any target seed pixel must be located inside a circle of radius 9 centered on the target patch;

and

2. Any seed pixel should have an intensity value in the 20 percent highest values of the target within the circle.

We justify the threshold of 20 percent by experimental observation. Experimental observations showed that with a larger percentage, noise pixels or artifacts are susceptible to be mistaken as target pixels whereas a smaller percentage is too restrictive and does not allow the algorithm to consider a sufficient intensity range for an average target. It can also be argued that the histogram-based segmentation technique used by *QuanArray* is using the 80th to 95th percentiles for determining the target mean. *QuantArray* also uses the 20 percent highest intensity values. Obviously, the right percentage is target-dependent. We present the results of this seed choice on a set of interesting targets from Figure B.20 to Figure B.29.

These examples show the choice of a few random pixels in the highest intensities generally lead to a better result. It can be argued that this method takes additional background pixels as target pixels. The Figure B.20 to Figure B.29 show the *doughnut-shaped* segmentation is still imperfect as the segmentation of low-expressed targets. However, the results are still more satisfactory.

4.4.4.2 Union Method

When segmentation fails with either seed choice, the result is most often a subset of the expected target site. We decided to take the *union* of the results obtained by these different seed choices. As each seed choice is successful on some set of targets, the union consists generally of the best solution and a few extra pixels. We show the results on the same target as previously.

The results appear to be quite satisfactory. The doughnut-shaped and low-level expression target especially have yet interesting segmented region. Some target like the target (2, 6, 15, Cy3) or (1, 3, 17, Cy3) have extra pixels. No target can be detected on the location (2, 15, 7, Cy3) and this method grew the whole background. Perhaps this union of seed choice can also be used to find unexpressed target and decide whether a target exists or not based on the size of the target area grown.

4.5 Mann-Whitney Test (MWT)

The Mann-Whitney test is also called the *Wilcoxon two-sample test* [25]. Historically Wilcoxon did present a similar test first but Mann and Whitney coined a name U for their statistic and accompanied their work with tables [26]. After a presentation of the theory of the Mann-Whitney Test and its application in the context of MicroArray Suite, we present some results showing the limits of the MWT.

4.5.1 Principle of the MWT

The Mann-Whitney test is a distribution free statistical test aimed at finding a significant difference between the population distributions of two sets A and B . In statistical terms, it is a test of the *null hypotheses* H_0 that two independent samples of observations X_1, \dots, X_m and Y_1, \dots, Y_n have come from populations having the same distribution [25]. Let μ_X and μ_Y be the mean of X and Y , two independent samples of two sets A and B . We can then also define the null hypotheses as $H_0: \mu_X = \mu_Y$.

The test is based on *Wilcoxon's rank sum* W defined as the sum of the rank of the elements of a sample Y that has been ordered with the X . To illustrate the previous definition, let us take a random sample R of n observations ordered as follows $R_1 < R_2 < \dots < R_n$ and S be the subset $\{S_1, S_2, S_3\} = \{R_1, R_2, R_n\}$. The rank sum of S against $R - S$ is $W_S = rk(S_1) + rk(S_2) + rk(S_3) = 1 + 2 + n = n + 3$.

The principle of the MWT is as follows. To compare the set A and B , the MWT pick a sample of observations X of size n in A and a sample of observations Y of size m in B . The MWT then order the $(n + m)$ observations. The *statistic* U_X is the sum of the counts of observations of X that precedes each observation of Y . Therefore the MWT can also be viewed as a one-tail test to detect the shift of the X (Y) distribution to the right of the Y (X) distribution. Initially observations of X (Y) are all placed left from the observations of Y (X) and the way samples are marked does not influence the ranks. If we consider two independent samples X and Y of n and m observations respectively, it can be shown that:

$$U_X = n \times m + n \times (n + 1)/2 - W_X \quad (4.7)$$

and

$$U_Y = n \times m + m \times (m + 1)/2 - W_Y \quad (4.8)$$

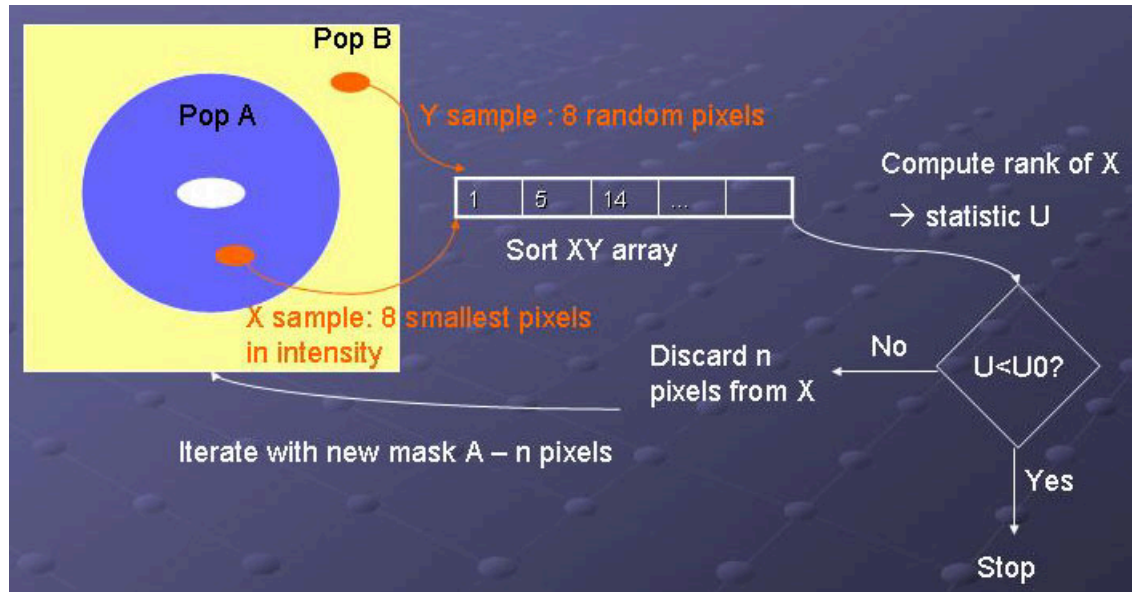


Figure 4.7: Principle of the Mann-Whitney Test used for microarray images segmentation.

Notice that $U_X(U_Y)$ is small when $W_X(W_Y)$ is large. Therefore $U_X(U_Y)$ is small when the population distribution of the $X(Y)$ measurements is shifted to the right of the population distribution of $Y(X)$.

The MWT rejects H_0 based on a significance level α . The desired significance level α serves as the dividing point between rejection and non-rejection. With α , n and m , we can find U_0 in probability tables such that $P(U < U_0) \leq \alpha$. This probability yields the confidence in *rejecting* the null hypotheses H_0 when $U < U_0$.

U is sometime called a *non-parametric* statistic because the dependent variables are the ranks. Though the Mann-Whitney test is a distribution free test (there is no assumption on the distribution of the original sets A and B), notice that the statistic U is close to being normally distributed for samples as small as 8.

4.5.2 Adaptation of the MWT to our Segmentation Problem

Chen et al. [23] present the Mann-Whitney test for cDNA microarray target segmentation as follows.

Manual or automatic gridding yields square *target patches*. The method first determines a pre-defined *target mask* T for each pin-array. T will be used as the original mask for the MWT of every

target in the corresponding pin-array. T is obtained as follows. Given σ the standard deviation of pixel intensities in a pin-array, w a user specified weight defaulted to 3 and μ_b the mean of the background, any target patch pixel (x, y, i) such that $i > w\sigma\mu_b$ is included in the mask T . We can also view T as the initial population A for the MWT. The set of pixels outside the *target mask* make up the *target background* or also the population B for the MWT.

The MWT then takes two independent samples from the two population A and B . Chen et al. [23] take the n smallest pixel values from the target mask as sample X and m random pixels from the background as sample Y . They advise to take 8 pixels for each sample as the MWT statistic U is approximately normal for this size of sample. We followed that advice and therefore $X = (x_1, y_1, i_1), \dots, (x_8, y_8, i_8)$ and $Y = (x'_1, y'_1, i'_1), \dots, (x'_8, y'_8, i'_8)$

The MWT [25] [26] is applied to each target mask as follows. The two samples are ordered and the shift of the pixel intensities i_1 to i_8 in sample X to the right of the pixel intensities i'_1 to i'_8 is assessed. As long as $U \geq U_0$, the null hypotheses is not rejected. The MWT proceeds to remove the pixels of smallest intensity from the target mask. New samples X and Y are chosen and iteratively the pixel values are removed from the mask until H_0 is rejected. When $U < U_0$, H_0 is rejected with the confidence α . We typically used a confidence α of 95 percent or 99.9999 percent. After the MWT rejected H_0 , the segmented regions are called *target site*.

The final state of the algorithm yields an unchanged background area (consisting of all pixels that were not contained in the original target mask) and a target site consisting of the original target mask minus the pixels rejected by the MWT. The algorithm proceeds then to compute the ratio. In [23], the *gene expression level* is measured as the median of the target site minus the median of the background area. MicroArray Suite actually evaluates the expression level by the mean of the target site minus the mean of the background.

4.5.3 Limitations of the MWT

The subsection is divided in two parts. In a first part, we present critical results obtained with MicroArray Suite. In a second part, we show a few observations obtained from our own version of the MWT.

4.5.3.1 MicroArray Suite Performance

As we confronted our implementation to the results of MicroArray Suite, we became aware of the limits of MicroArray Suite MWT. It is possible to visualize the target site resulting from the Mann-Whitney Test. We already showed in Figure 3.8 that a misaligned target could be badly segmented. This result is the consequence of a bad addressing more than a bad segmentation. However, the Figure 3.8 also shows the segmentation was not well done as a large part of the target site seems to belong to the background. Figure 3.8 is a case where the original target mask is not covering the whole target. Only half of the target is in the final target mask and half the mask is containing invalid pixels. By assessing the other segmentations, the original target looked however to be quite circular.

Figure C.1 shows segmentation results on a few targets by MS, all extracted from the top left pin-array of the image S3. Though the first three targets obviously appear to have no hybridization, the final target site are quite different and of a consequent dimension. The same original target mask is normally used. The fourth target shows what we think is close to the original target mask chosen by MS. This site was actually the biggest target site found on the whole grid. This mask is abnormally big due to a large noisy area on our image. This example demonstrates that the original target mask choice used by MicroArray Suite can be unsatisfactory. The last target picture shows a typical result for a target of good quality. As Dudoit et al. [21] pointed out, we confirm here the MicroArray Suite MWT usually selects extra pixels as being part of the target.

4.5.3.2 Critical Observations on the MWT

We have observed that targets typically have a circular shape of a radius varying between 6 and 8 pixels for an average microarray hybridization. Even in the case of bad microarray images, the target mask considered should be a circle. Our implementation differs from the Chen et al. [23] one as our target patches are circle centered on the square region. We believe the MWT should be able to handle the different target shapes regardless of the original target mask. As long as the target patch is big enough to include all the target pixels, the MWT should be strong enough to eliminate background pixels. In our attempts to obtain closer results to the MicroArray Suite' ones, we modified parameters such as the radius of the target patch and the confidence level.

Our first observation is that the trend to take extra pixels is less important in our implementation than it seems to be in the MicroArray Suite one. Our result are all consistent and we do not obtain

large target areas. We are conscious our implementation is using the most restrictive significance level (99.9999%) but the MicroArray Suite results shown were at a 95%.

Unless noted otherwise all these pictures result from a MWT with 8 pixel samples and a confidence level that is maximum ($U_0 = 0$). The Figures on the segmentation of the target (1, 7, 4) and (1, 7, 23) from the target mask with different radius show that the MWT is radius-dependent and more precisely mask-dependent. Secondly the best segmentation is obtained with a radius of 7 for the target (1, 7, 4) and a radius of 9 gives the best segmentation with the target (1, 7, 23). We could show other targets whose best segmentation is obtained with a radius of 5, 6, or some other value. This result shows the target mask must be adapted to the target shape. This result shows an adaptive circle addressing can give better results than the current method used by MicroArray Suite of computing an average target mask. This issue also raises doubts about the validity of the data extracted.

Intrigued by the target site obtained by MicroArray Suite on apparently non-hybridized areas, we looked at the effect of the MWT on a background area outside the array. On a statistical point of view, the MWT should not detect any difference in the distributions of the target mask and target background even with the presence of noise supposedly random.

The Figures C.14 to C.17 show that the MWT is extracting noise pixels in a no-target area. A few pixels, generally about 30, are still selected by our implementation. This behavior is occurring in both channels and under the maximum significance level. This result suggests the MWT results needs to be used cautiously as this behavior will influence the estimation of weak targets. Figures C.18 to C.20 shows the radius of the target mask still has an influence on the result. Figures C.21 and C.22 show the expected dependency of the result on the confidence level.

4.6 Comparison of Performances

We present next data obtained from the segmentation of our SRG (Union method with a seed size of 3) and MWT algorithms. The size of the union of the target sites obtained in each channel was chosen as the criterion to evaluate the performance of our segmentation technique. We compare our results to the one obtained by MicroArray Suite (MS). We also compare the background corrected ratios (RAT2 of ScanAlyze) of our implementations to the one of MS.

Table 4.3 shows our SRG implementation typically undersegments a target. In many cases, the

Table 4.3: Sizes of the target site obtained with our SRG (Union method) on S3 images.

Pin-Array	Row	Col	Our SRG site	MS site
1	2	22	68	118
1	3	8	64	116
1	3	14	122	117
1	3	17	70	114
1	3	24	165	113
1	4	5	53	114
1	5	3	62	118
1	5	8	67	113
1	7	4	140	117
1	7	5	63	118
1	7	9	298	109
1	7	10	82	117
1	7	23	66	118
1	7	24	59	116
1	11	22	222	109
1	11	23	81	101
1	15	3	127	109
1	16	24	65	113
1	17	5	87	110
1	17	7	120	116
1	18	22	91	110
1	19	4	132	118
2	3	11	118	131
2	6	15	88	130
2	15	7	117	117
3	3	11	68	120
3	6	15	38	123
3	15	7	68	127
4	10	14	69	197
4	11	20	646	217
4	14	9	89	320
4	15	6	301	322

Table 4.3 and Table 4.4 show the results obtained on the same spots by our SRG and MWT implementations.

union target site is much smaller than the one obtained by MS. MS typically oversegment targets and our SRG still tend to undersegment targets in many cases. We also think MS site sizes are suspiciously close from each other in a common pin-array and have no idea why this occur.

The SRG can also have extreme behaviors and grow a site of 646 pixels (vs 217) or 298 (vs 109). When the target is low-expressed, the SRG can fail to locate the target and misinterpret and grow a large background area as the target. The first case can be explained as the pin-array A4 was overlayed by a large artifact. However, in the second case (1, 7, 9) is a reasonable target to segment. The target is small and slightly low-expressed and the patch is not very noisy.

Table 4.4 show our MWT systematically has larger site. We think this result is due to the fact that we applied our segmentation to both channel and then took the union of the segmented regions whereas MS computes its site on the combined image. The intersection of MS regions is close to the union. However, our implementation has a bigger number of pixels that are not in the intersection and are in the union. Paradoxically, our results visually appear to match more closely the targets than the one shown by MS.

Table 4.5 and Table 4.6 present the background corrected ratios obtained with our SRG and MWT against the background corrected (uncalibrated) ratio of MWT. The background corrected ratio is commonly used (Refer to equation 4.1). It is hard to hold a judgment on these results as even MS ratios can not be considered as a safe reference. However, the SRG shows some extreme behavior with negative ratios or values as big as 30. A negative value can be obtained when the background mean is bigger than the target site mean. This behavior occurs with weak targets as for (1, 11, 22, Cy3) and (1, 15, 3, Cy3) or noisy patches as for (4, 11, 20). The MWT has a more consistent behavior and all ratios obtained are in a trustful range. Our ratios are typically smaller than the MS ratios and we think it is due to the fact our background means are bigger than those obtained by MS. We are actually unsure how MS obtains such small background means.

4.7 Discussion

The results obtained with the MWT are more reliable than the ones obtained with the SRG. The SRG can in some cases make a quite perfect segmentation. However, in too many cases the seed choice is critical and the result is catastrophic. We have shown the performance of the SRG can be improved by a more appropriate seed choice. However, the union method that appears to perform

Table 4.4: Sizes of the target site obtained with our MWT on images of the S3 experiment.

Pin-Array	Row	Col	Our MWT site	MS Union Area
1	2	22	167	118
1	3	8	136	116
1	3	14	174	117
1	3	17	124	114
1	3	24	120	113
1	4	5	116	114
1	5	3	130	118
1	5	8	133	113
1	7	4	170	117
1	7	5	138	118
1	7	9	110	109
1	7	10	124	117
1	7	23	195	118
1	7	24	179	116
1	11	22	102	109
1	11	23	74	101
1	15	3	100	109
1	16	24	124	113
1	17	5	158	110
1	17	7	135	116
1	18	22	137	110
1	19	4	163	118
2	3	11	100	131
2	6	15	112	130
2	15	7	104	117
3	3	11	80	120
3	6	15	79	123
3	15	7	80	127
4	10	14	72	197
4	11	20	41	217
4	14	9	105	320
4	15	6	104	322

Table 4.5: Background corrected Ratios of our SRG vs MS ones.

Pin-Array	Row	Col	Our SRG ratios	MS ratios
1	2	22	0.223	0.324
1	3	8	0.132	0.207
1	3	14	0.135	0.187
1	3	17	0.056	0.355
1	3	24	0.138	0.389
1	4	5	0.146	0.267
1	5	3	0.232	0.295
1	5	8	0.073	0.178
1	7	4	0.074	0.148
1	7	5	0.106	0.168
1	7	9	-0.0220	0.435
1	7	10	0.131	0.264
1	7	23	0.046	0.149
1	7	24	0.139	0.369
1	11	22	-8.776	4.345
1	11	23	0.494	7.252
1	15	3	30.655	1.759
1	16	24	0.128	0.184
1	17	5	0.050	0.190
1	17	7	0.262	0.467
1	18	22	0.161	0.285
1	19	4	0.100	0.150
2	3	11	0.124	3.080
2	6	15	0.140	0.726
2	15	7	1.636	7.423
3	3	11	1.534	1.794
3	6	15	1.181	1.394
3	15	7	3.124	1.641
4	10	14	0.223	0.551
4	11	20	-0.016	0.502
4	14	9	0.055	1.451
4	15	6	0.036	1.952

Table 4.6: Background corrected Ratios of our MWT vs MS ones.

Pin-Array	Row	Col	Our MWT ratios	MS ratios
1	2	22	0.133	0.324
1	3	8	0.172	0.207
1	3	14	0.167	0.187
1	3	17	0.174	0.355
1	3	24	0.195	0.389
1	4	5	0.198	0.267
1	5	3	0.242	0.295
1	5	8	0.126	0.178
1	7	4	0.089	0.148
1	7	5	0.121	0.168
1	7	9	0.158	0.435
1	7	10	0.152	0.264
1	7	23	0.065	0.149
1	7	24	0.158	0.369
1	11	22	0.935	4.345
1	11	23	1.826	7.252
1	15	3	0.759	1.759
1	16	24	0.163	0.184
1	17	5	0.083	0.190
1	17	7	0.236	0.467
1	18	22	0.215	0.285
1	19	4	0.067	0.150
2	3	11	1.024	3.080
2	6	15	0.231	0.726
2	15	7	1.610	7.423
3	3	11	1.104	1.794
3	6	15	0.596	1.394
3	15	7	1.000	1.641
4	10	14	0.401	0.551
4	11	20	1.305	0.502
4	14	9	0.418	1.451
4	15	6	0.629	1.952

the best generally takes extra noise or even background pixels in the target site. The results are still unsatisfactory compared to the MWT ones. If the MWT systematically includes noise or small artifact pixels in the target site, it does not fail to identify a major part of a target unless the original mask is too small to cover the entire target. The MWT is dependent on the original target mask (circle), but we believe it is possible to adjust the target mask in size and location to cover the target in an appropriate way. An improvement is also needed to compensate for the noise. Perhaps we can restrict the target area to be the biggest connected area after the application of the MWT.

The MWT variance of the result is less important than the SRG ones. Future work can include an implementation of the Mehnert and Jackway SRG version and further investigations to make the SRG more reliable on different target shapes. We are more optimistic about improving the MWT segmentation. Future work could consist of the implementation of the *Wilcoxon–Wilcox multiple comparison test* involving K samples. The *randomness* involved in the choice of the background set in the application of the MWT devised by Chen et al. [23] is influencing the result. It may be able to reduce the number of extra pixels in the final segmented target site by running multiple MWT. An alternative method may be designed to reinsert target pixels originally outside the target site because the original target mask was not covering the real target correctly.

Analysis of the data obtained is complex because of the numerous parameters involved in a microarray experiment. The difficulty of this research is the lack of reference. No software system can serve as a reference. Only the similarity of results between many system can bring some confidence in a result. A reliable way to evaluate a segmentation technique is to create synthetic images. Such images have results known in advance that can serve as a reference. We started to create and look forward to creating more complex synthetic image.

Chapter 5

Data Extraction and Analysis

Many methods exist to analyze the extracted data. The background corrected ratio (see Equation 4.1), also called the raw ratio is commonly used. It is accepted that background corrected ratios need to be calibrated because of the difference in dye incorporation at least. Many methods exist to normalize the data and obtain calibrated ratios. However, we have been interested so far by the method used by MicroArray Suite (MS). As going from the theory to the practice still requires an effort to understand how the data was obtained, we present our work to decipher the MS calibration method. We present the theory and iterative procedure used by MicroArray Suite to calibrate ratios in section 5.1. In section 5.2, we present experiments made with synthetic images made to test and confirm this calibration procedure. We present a few results revealing how MicroArray Suite selects target and pixel outliers. In section 5.3, we present the results obtained by MS on various artificial data.

5.1 Principle of the MicroArray Suite (MS) Procedure

MicroArray Suite (MS) [37] was initially developed by Dr. Chen based upon work carried out at the Cancer Research Lab at NHGRI. Further development has been done by ScanAlytics Inc. which developed a commercial version of MS that is included in *IPLab*. One key features govern the way MS operates: the use of designated control (housekeeping or reference) genes to normalize the data.

From a user point of view, the MS process is the following. Gridding is done manually. A manual registration (large X-Y shift) of the two images is sometimes necessary. We experienced

once a failure of MS to correctly register the images with the automatic registration (small X-Y shift) and the combined image were off by a column. MS provides a picture of the target site after the MWT. The program undertakes a sequence of calculations summarized below:

1. Compute the mean of the total target intensities ;
2. Compute the mean of the local background;
3. Determine the corrected mean target intensity (subtract local background mean);
4. Calculate the background corrected ratios for all the R/G targets;
5. Determine the calibration factor M with the iterative procedure computing the “calibrated” ratios of the control genes; and
6. Divide each background corrected ratio by M to obtain a calibrated ratio for each gene.

The next paragraph present some computational details and the calibration procedure. Given s_i target site and bk_i background site in the channel i , the background corrected mean ratio is:

$$Ratio = \frac{Ch2(red)}{Ch1(green)} = \frac{\mu_{s_2} - \mu_{bk_2}}{\mu_{s_1} - \mu_{bk_1}}. \quad (5.1)$$

The motivation to subtract the mean of the background pixels from the mean of target site pixels is to eliminate any background contribution such as the glass slide or noise. The background corrected mean $\mu_{s_i} - \mu_{bk_i}$ is corresponding to the *gene expression*. It represents the expression level of the gene in this channel of this particular experiment. We wish to outstand the fact that the MS data file is confusing at this point. The reported S#1 Mean and S#2 Mean are not target site means μ_{s_i} but background corrected mean (see Table 5.2) such that:

$$S\#iMean = \mu_{s_i} - S\#iBkMean. \quad (5.2)$$

Because of the incorporation of dyes and the characteristics of the scanner at least, gene expression may differ significantly for the red and the green samples. The background corrected ratio of Equation 5.1 is an unsatisfactory measure. Chen et al. [23] developed a statistical method to calibrate the ratio distribution used in MicroArray Suite. Chen et al. [23] point out that even if red and green measurements are identically distributed, the mean of the ratio distribution will not be 1. Indeed the ratio distribution is dependent on the genes printed. The expression of these genes

is unknown and subject to variations. There is therefore no reason for the mean of all ratios to be equal to 1. Because of the differences in the characteristics of dyes, it is unacceptable to consider a desired null hypotheses $H_0: \mu_{R_k} = \mu_{G_k}$ where R_k and G_k are genes of the red and green channel. Chen et al. [23] assume it momentarily to explain the basis of their theory but then argue a mRNA abundance is dependent on the abundance of the factors leading to its selection. Assuming the variations for any particular mRNA are normally distributed and independent of other transcripts, Chen et al. consider a constant coefficient of variation c for the entire gene set printed such that $\sigma_{R_k} = c\mu_{R_k}$ and $\sigma_{G_k} = c\mu_{G_k}$. Assuming R_k and G_k , independent and identically distributed normal random variables and c , a constant coefficient of variation, Chen et al. [23] derive an asymmetric density function f peaking close to 1. f is used to derive a maximum-likelihood estimator \hat{c} for c ,

$$\hat{c} = \sqrt{\frac{1}{n} \sum_{i=1}^n \frac{(t_i - 1)^2}{(1 + t_i^2)}} \quad (5.3)$$

and a polynomial approximation of the mean μ given by

$$\mu = 0.364c^3 + 1.279c^2 - 0.0427c + 1.001. \quad (5.4)$$

However, in practice the null hypotheses of equal means is not an appropriate assumption. Chen et al. put forward the assumption that red and green mean signals are related by a constant gain (or calibration) factor m such that $\mu_{R_k} = m\mu_{G_k}$. The ratio density function f of the uncalibrated case now depends on m and c which leads to a recurrence relation. Given $f(\cdot, c, 1)$ the density function satisfies $f(t, c, m) = \frac{1}{m} f(t/m, c, 1)$. Intuitively, the calibration consists of moving the ratio histogram mode to 1. The mode is actually the ratio occurring with the highest frequency, that is also the abscise of the maximum in the histogram. As the ratio density function is skewed, the simple approach of moving the histogram maximum to 1 is not correct. c is a parameter of the ratio density function and therefore influence the peak of the ratio density. Therefore an iterative procedure is proposed by Chen et al. [23] to estimate the calibration factor. Figure 5.1 illustrate the procedure as described in [23].

- 1 Initialize mean estimate $\hat{\mu}_0$ of the ratio density to 1.
- 2 **While** $i < T$ (convergence is usually satisfactory after 5 iterations)
- 3 $\hat{m}_i = \frac{1}{\hat{\mu}_{i-1}} (\frac{1}{n} \sum_{j=1}^n t_j)$
 ▷ Calibrate ratio samples by \hat{m}_i so that the red and green signals are approximately equal.
- 4 $(t'_1, \dots, t'_n) = (t_1/\hat{m}_i, \dots, t_n/\hat{m}_i)$
- 5 Use Equation 5.3 to calculate \hat{c}_i
- 6 Use Equation 5.4 to calculate $\hat{\mu}_i$
- 7 Compute the confidence limits interval $(\theta_1.\hat{m}, \theta_2.\hat{m})$

Figure 5.1: MicroArray Suite Iterative Procedure to calibrate ratios

Table 5.1: Target considered as Outliers by MS and tossed out to calibrate ratios

No.	CloneID	Ratio	Cal. Ratio	S#1 Mean	S#2 Mean	Array Pos.
1	STF00009	0.4	0.491	4000	10000	[1-4-1]
2	STF00006	0.3	0.368	3000	10000	[1-3-1]
3	STF00003	0.2	0.246	2000	10000	[1-2-1]
4	STF00000	0.1	0.123	1000	10000	[1-1-1]

5.2 Handling Outliers

The next paragraph presents implementation details of the MS calibration method. MS provide the user a way to indicate control genes of an experiment via a file (GIPO file). Control genes are not always part of an experiment. In that case, MS uses all the genes to calibrate ratios except for outliers. In MS, any target with a ratio outside the ratio limits $[0.5, 2.0]$ is considered an outlier. We created artificial data and Table 5.1 presents the outliers chosen by MicroArray Suite on two synthetic images shown in Appendix by Figure D.2. These images have uniform targets of radius 8 of different intensities. The expected ratios are 0.1, 0.2, 0.3, ..., 1.2. As we did not specify any control genes, the targets (1,1)(2,1)(3,1)(4,1) of respective ratios 0.1, 0.2, 0.3 and 0.4 are considered to be outliers as expected.

MS also discard “outlier” pixels, pixels of extreme intensities from the calculations. Dr. Chen invented a special algorithm to determine how many high and low pixels should be tossed out from each target based upon how strong it was. We do not know the details of this algorithm but we report results obtained with synthetic images designed in the purpose of determining how outlier pixels were chosen. The MS data exported in T. Int. columns is the sum of all “valid” pixels within

Table 5.2: Results given by MS for the 4 highest-lowest pixels rejection

Finger#	Row	Column	S#1 T.Int.	S#1 Mean	Union Area	S#1BkMean
1.00	2.00	1.00	2130000	9000	221	1000
1.00	3.00	1.00	639000	2000	221	1000
1.00	4.00	1.00	2343000	10000	221	1000

Notice that S#1 Mean is 9000 instead of 10000 as expected. S#1 Mean is the background corrected mean or gene expression. S#1BkMean has been subtracted.

Table 5.3: Ratios, Cal. ratios, calibration factor M Obtained by MS

Finger #	Row	Column	Union Area	Ratio	Cal. Ratio	M
1	1	2	221	0.5	0.6141602	0.8141198
1	1	3	221	0.9	1.1054880	0.8141198
1	2	2	221	0.6	0.7369923	0.8141198
1	2	3	221	1.0	1.2283200	0.8141198
1	3	2	221	0.7	0.8598243	0.8141198
1	3	3	221	1.1	1.3511530	0.8141198
1	4	2	221	0.8	0.9826564	0.8141198
1	4	3	221	1.2	1.4739850	0.8141198

the target site outline and “Union Area” (common pixels in both channels). However, experimenting with our synthetic images revealed that the Total Intensity is the sum of “valid” pixels minus 8 pixels. After investigation, we noticed MicroArray Suite eliminates the 4 pixels with the highest intensity and the 4 pixels with the lowest intensity in the Union Area. Table 5.2 shows the results obtained with MS on the images of Figure D.4. The channel 2 image contains perfect targets but the Channel 1 has three targets with exactly 4 pixels of higher and 4 pixels of lower intensity in the center of the target. The Union Area has 221 pixels as expected for a perfect target of radius 8. However, Table 5.2 shows that the T. Int. for the target (1, 2, 1) are 2 130 000 which means only 213 pixels of intensity 10000 were considered. Eight pixels have been tossed out and it is necessarily the 4 highest pixels (of intensity value 20000) and the 4 lowest pixels (of intensity value 2000).

By removing the target and pixel outliers, we were able to reproduce the MS results and find the calibration factor M from all the target minus the outliers (see Table 5.4 and Table 5.3). For reference, Table 5.5 and Table 5.6 present T. Int, Mean, Area size obtained.

Table 5.4: Iterative procedure results

Finger #	Row	Column	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5
1.00	1.00	2.00	0.5882353	0.6144570	0.6141634	0.6141602	0.6141602
1.00	1.00	3.00	1.0588235	1.1060227	1.1054942	1.1054884	1.1054884
1.00	2.00	2.00	0.7058824	0.7373484	0.7369961	0.7369923	0.7369923
1.00	2.00	3.00	1.1764706	1.2289141	1.2283268	1.2283205	1.2283204
1.00	3.00	2.00	0.8235294	0.8602399	0.8598288	0.8598243	0.8598243
1.00	3.00	3.00	1.2941176	1.3518055	1.3511595	1.3511525	1.3511525
1.00	4.00	2.00	0.9411765	0.9831313	0.9826615	0.9826564	0.9826563
1.00	4.00	3.00	1.4117647	1.4746969	1.4739922	1.4739846	1.4739845
		c_i	0.1961425	0.1951433	0.1951324	0.1951323	0.1951323
		μ_i	1.0445770	1.0440778	1.0440724	1.0440724	1.0440724
		m_i	0.8137265	0.8141156	0.8141198	0.8141198	0.8141198

Table 5.5: Results given by MS for the channel 1 of Figure D.2

Finger #	Row	Column	S#1 T. Int.	S#1 Mean	S#1 Area	S#1 BkMean
1	1	2	1278000	5000	221	1000
1	1	3	2130000	9000	221	1000
1	2	2	1491000	6000	221	1000
1	2	3	2343000	10000	221	1000
1	3	2	1704000	7000	221	1000
1	3	3	2556000	11000	221	1000
1	4	2	1917000	8000	221	1000
1	4	3	2769000	12000	221	1000

Table 5.6: Results given by MS for the channel 2 of Figure D.2

Finger #	Row	Column	S#2 T. Int.	S#2 Mean	S#2 Area	S#2 BkMean
1	1	2	2343000	10000	221	1000
1	1	3	2343000	10000	221	1000
1	2	2	2343000	10000	221	1000
1	2	3	2343000	10000	221	1000
1	3	2	2343000	10000	221	1000
1	3	3	2343000	10000	221	1000
1	4	2	2343000	10000	221	1000
1	4	3	2343000	10000	221	1000

5.3 Analysis of MS Performances

Table 5.7 presents the result obtained by MicroArray Suite and ScanAlyze on the images of our experiment S3. The two programs gives alternatively the same and different results. This behavior justify the work done in this study to improve gridding, and segmentation. The results given by a particular program have to be interpreted with caution.

The formula of RAT2 and Ratio differ from taking the median instead of the mean of the background to correct the target site means. Some results are close but need to be proof-checked over a few replicas. ScanAlyze inherently has biased results due to its simple segmentation approach. Except a general feeling of a possible consistency, it is hard to interpret these results. Any assumption has a counterexample.

We are convinced that assessing the performance of a segmentation technique and its data extraction method can only be made with synthetic images whose correct results are known in advance. Table 5.8, Table 5.9 and Table 5.10 are reporting the MS results obtained for the synthetic images in Figure D.6, Figure D.8 and Figure D.10, experiments designed to test the effect of targets with different radius, targets shape (square) and doughnut-shape targets.

For the different radius, the results are the one expected. MS does take a larger original target mask in the case of a bigger target and the union area are corresponding to the size of the mask of biggest radius.

In the case of the square targets, MS also detect the only half square and find relevant targets sizes.

The results are also mostly correct for the doughnut-shape targets. The doughnuts of the pin-array A1 are correctly segmented with a union size of 44 but the T. Int. are not appropriate. We are unsure if MS take these pixels for outliers or not.

Table 5.7: Ratios given by MS and ScanAlyze on the same targets of S3 experiment

Pin-array	Row	Column	RAT2	MRAT	Ratio	Cal. Ratio
1	2	22	0.18	0.33	0.32	0.30
1	3	8	0.17	0.31	0.21	0.19
1	3	14	0.18	0.22	0.19	0.18
1	3	17	0.23	1.47	0.36	0.33
1	3	24	0.21	0.67	0.39	0.36
1	4	5	0.20	0.45	0.27	0.25
1	5	3	0.29	0.56	0.30	0.28
1	5	8	0.15	0.38	0.18	0.17
1	7	4	0.11	0.19	0.15	0.14
1	7	5	0.14	0.23	0.17	0.16
1	7	9	0.22	0.47	0.44	0.41
1	7	10	0.20	0.31	0.26	0.25
1	7	23	0.08	0.13	0.15	0.14
1	7	24	0.13	0.27	0.37	0.35
1	11	22	1.21	4.73	4.35	4.06
1	11	23	1.18	2.85	7.25	6.77
1	15	3	0.49	1.11	1.76	1.64
1	16	24	0.19	0.84	0.18	0.17
1	17	5	0.15	0.26	0.19	0.18
1	17	7	0.31	0.54	0.47	0.44
1	18	22	0.23	0.38	0.29	0.27
1	19	4	0.11	0.16	0.15	0.14
2	3	11	0.46	1.62	3.08	2.88
2	6	15	0.62	1.81	0.73	0.68
2	15	7	0.11	1.43	7.42	6.93
3	3	11	1.30	2.36	1.79	1.67
3	6	15	0.34	1.49	1.39	1.30
3	15	7	1.91	2.28	1.64	1.53
4	10	14	0.43	0.59	0.55	0.52
4	11	20	0.00	-1.06	0.50	0.47
4	14	9	0.39	1.08	1.45	1.35
4	15	6	0.64	1.35	1.95	1.82

Table 5.8: Micr. Suite Results on synthetic images of perfect targets with different radius.

F	R	C	Signal #1				Signal #2				U	R	CR
			TInt	M	A	BkM	TInt	M	A	BkM			
1	1	1	10650000	49500	221	500	10650000	49500	221	500	221	1.00	0.77
...													
1	4	2	10650000	49500	221	500	10650000	49500	221	500	221	1.00	0.77
1	4	3	1	1	0	500	1	1	0	500	0	1.00	0.77
2	1	1	10912000	31500	221	500	17050000	49500	349	500	349	0.64	0.49
...													
2	4	2	10912000	31500	221	500	17050000	49500	349	500	349	0.64	0.49
2	4	3	1	1	0	500	1	1	0	500	0	1.00	0.77
3	1	1	10650000	49500	221	500	6294000	29049	129	500	221	1.70	1.31
...													
3	4	2	10650000	49500	221	500	6294000	29049	129	500	221	1.70	1.31
3	4	3	1	1	0	500	1	1	0	500	0	1.00	0.77
4	1	1	10650000	49500	221	500	2928000	13246	61	500	221	3.74	2.87
...													
4	4	2	10650000	49500	221	500	2928000	13246	61	500	221	3.74	2.87
4	4	3	1	1	0	500	1	1	0	500	0	1.00	0.77

F: Finger(Pin-Array), R: Row, C:Column, TInt: Target Total Intensity, M: Mean , A: Target Site Area (Size), BkM: Background Mean, U: size of the union of target sites, R: Ratio, CR: Calibrated Ratio

Table 5.9: Micr. Suite Results on square 16 *times*16 targets except one.

F	R	C	Signal #1				Signal #2				U	R	CR
			TInt	M	A	BkM	TInt	M	A	BkM			
1	1	1	496000	1000	256	1000	496000	1000	256	1000	256	1	1
1	1	2	496000	1000	256	1000	496000	1000	256	1000	256	1	1
1	1	3	2400000	19000	128	1000	2400000	19000	128	1000	128	1	1
1	2	1	496000	1000	256	1000	496000	1000	256	1000	256	1	1
...													

F: Finger(Pin-Array), R: Row, C:Column, TInt: Target Total Intensity, M: Mean , A: Target Site Area (Size), BkM: Background Mean, U: size of the union of target sites, R: Ratio, CR: Calibrated Ratio

Table 5.10: MicroArray Suite Results on doughnut-shape targets.

F	R	C	Signal #1				Signal #2				U	R	CR
			TInt	M	A	BkM	TInt	M	A	BkM			
1	1	1	1	1	44	1000	1	1	44	1000	0	1.00	1.000971
...													
1	4	2	1	1	44	1000	1	1	44	1000	0	1.00	1.000971
1	4	3	1	1	0	1000	1	1	0	1000	0	1.00	1.000971
2	1	1	117160	10	124	1000	117160	10	124	1000	124	1.00	1.000971
2	1	2	580000	4000	124	1000	580000	4000	124	1000	124	1.00	1.000971
2	1	3	1740000	14000	124	1000	1740000	14000	124	1000	124	1.00	1.000971
2	2	1	127600	100	124	1000	127600	100	124	1000	124	1.00	1.000971
2	2	2	1160000	9000	124	1000	1160000	9000	124	1000	124	1.00	1.000971
2	2	3	1740000	14000	124	1000	1740000	14000	124	1000	124	1.00	1.000971
2	3	1	174000	500	124	1000	174000	500	124	1000	124	1.00	1.000971
2	3	2	1740000	14000	124	1000	1740000	14000	124	1000	124	1.00	1.000971
2	3	3	1740000	14000	124	1000	1740000	14000	124	1000	124	1.00	1.000971
2	4	1	232000	1000	124	1000	232000	1000	124	1000	124	1.00	1.000971
2	4	2	1740000	14000	124	1000	1740000	14000	124	1000	124	1.00	1.000971
2	4	3	1	1	0	1000	1	1	0	1000	0	1.00	1.000971
3	1	1	84840	10	92	1000	84840	10	92	1000	92	1.00	1.000971
3	1	2	420000	4000	92	1000	420000	4000	92	1000	92	1.00	1.000971
3	1	3	1260000	14000	92	1000	1260000	14000	92	1000	92	1.00	1.000971
3	2	1	92400	100	92	1000	92400	100	92	1000	92	1.00	1.000971
3	2	2	840000	9000	92	1000	840000	9000	92	1000	92	1.00	1.000971
3	2	3	1260000	14000	92	1000	1260000	14000	92	1000	92	1.00	1.000971
3	3	1	126000	500	92	1000	126000	500	92	1000	92	1.00	1.000971
3	3	2	1260000	14000	92	1000	1260000	14000	92	1000	92	1.00	1.000971
3	3	3	1260000	14000	92	1000	1260000	14000	92	1000	92	1.00	1.000971
3	4	1	168000	1000	92	1000	168000	1000	92	1000	92	1.00	1.000971
3	4	2	1260000	14000	92	1000	1260000	14000	92	1000	92	1.00	1.000971
3	4	3	1	1	0	1000	1	1	0	1000	0	1.00	1.000971
4	1	1	153520	10	160	1000	153520	10	160	1000	160	1.00	1.000971
4	1	2	760000	4000	160	1000	760000	4000	160	1000	160	1.00	1.000971
4	1	3	2280000	14000	160	1000	2280000	14000	160	1000	160	1.00	1.000971
4	2	1	167200	100	160	1000	167200	100	160	1000	160	1.00	1.000971
4	2	2	1520000	9000	160	1000	1520000	9000	160	1000	160	1.00	1.000971
4	2	3	2280000	14000	160	1000	2280000	14000	160	1000	160	1.00	1.000971
4	3	1	228000	500	160	1000	228000	500	160	1000	160	1.00	1.000971
4	3	2	2280000	14000	160	1000	2280000	14000	160	1000	160	1.00	1.000971
4	3	3	2280000	14000	160	1000	2280000	14000	160	1000	160	1.00	1.000971
4	4	1	304000	1000	160	1000	304000	1000	160	1000	160	1.00	1.000971
4	4	2	2280000	14000	160	1000	2280000	14000	160	1000	160	1.00	1.000971
4	4	3	1	1	0	1000	1	1	0	1000	0	1.00	1.000971

Chapter 6

Conclusions and Future work

This study is motivated by the development of a Microarray Experiment Management System, Espresso [8, 9]. Espresso is aimed at supporting all the stages of a microarray experiment from design to analysis. At each stage, our goal is to provide the user with a choice of multiple methods. Espresso image processing tools will include various gridding, segmentation and data extraction/analysis methods.

This study first examined automatic gridding methods. We defined gridding as the process of overlaying a grid of patches over the hybrid compounds fluorescence called targets over the different pin-arrays in the image. We presented methods based on the Discrete Fourier Transform, Circular Hough Transform, Mann-Whitney Test and a final method that combined the use of the Discrete Fourier Transform and Circular Hough Transform, we called the hybrid method. Our images typically are subject to an exponential noise, artifacts and saturation. Thresholding was applied to lessen the noise effect.

The hybrid method obtained the best results. Technically, after thresholding, the hybrid method is able to grid the four pin-arrays when the image is of an average quality. The Discrete Fourier Transform and Circular Hough Transform based methods perform a good gridding on 7 out of 8 pin-arrays in the image (NS3, NS5). They are more time efficient but miss the address of a pin-array by a few columns or rows apart. For images of poorer quality, none of our methods obtained satisfactory results.

During this phase, we were confronted with four problems: the noisy edges of the image, the

tilting of pin-arrays, the misalignment of targets, and the different sizes and shape of targets. We look forward to improving these methods by adjusting methods for these problems. We look forward to implementing other methods. We are currently experimenting a method using morphological operators and taking into account the tilting effect and the targets sizes. In the future, we are planning to examine a 2-dimensional Discrete Fourier Transform, a shrinking algorithm [22] to reduce noise, artifacts and identify a region of interest for each pin-array. The connected components algorithm from Rosenfeld [36] identifies all components of an image (target, artifacts, noisy edges) and selection of components can be done by defining criterion later on. The algorithm is efficient at counting small stomata in a leaf cell.

In the segmentation phase, we examined two adaptive shape segmentation techniques, the Seeded Region Growing algorithm and the Mann-Whitney Test. We did not obtain satisfactory results with our Seeded Region Growing algorithm implementation but this implementation is order-dependent. We avoided the bleeding effect by limiting our segmentation to one target at a time. We showed that the target seed choice and location are critical and can lead to very different segmentation results. We tried to improve the target seed choice, and the random unconnected seed is performing the best. The best seed choice is target dependent. Therefore the union of the different results obtained for different seed choices provides the best result. For targets that are not uniform and connected, bad segmentation is expected. These targets are common in images of poor quality. A transformation could be applied to such images so that the pixels ranging from an arbitrary threshold T to 65535 intensity values are concentrated in a smaller band of intensities. The Seeded Region Growing algorithm should perform a better segmentation as the target will be more uniform. However, the problem of unconnected targets is not solved. We showed the Mann-Whitney Test has more consistent results. The problems involved with the Mann-Whitney Test are the choice of the original target mask and the extra noise or small artifacts pixels systematically in the final target mask. The Mann-Whitney Test segmentation is dependent of the original target mask. We tried circular target masks of different radii and showed different targets obtained their best segmentation with different radii. This result suggests the mask selection proposed by Chen et al. [23] is not optimal. We believe that circular target mask with different radii should be used and in the future an adjustment method in between the gridding and segmentation should estimate target radii. The Mann-Whitney Test is also including systematically a few extra noise pixels. We are convinced an unconnected target mask method is more appropriate than a connected one. However, a criteria

may be used with the Mann-Whitney Test to limit the unconnected area to consist of a minimum number of pixels.

We undertook to look at the performance of MicroArray Suite. After showing non-obvious details of its data extraction method, we convinced ourselves the use of synthetic images is necessary to evaluate the performance of a segmentation technique. We developed simple synthetic images and MS obtained satisfactory results. We look toward developing more complicated data.

To summarize, progress have been made toward automatic gridding. In the future, our methods may be good enough as the quality of images improve. Our goal is to automatically grid bad quality images. The Mann-Whitney Test is the most satisfactory segmentation technique to our knowledge. An intelligent choice of the original target mask is however required as well as a refined method to eliminate the extra noise pixels. No consensus exists on the data extraction method. We examined the MS method. The calibration procedure used is the most sophisticated method to our knowledge and we plan to generate more complex synthetic images to study the performance of MS and the Mann-Whitney Test.

We look forward to improve and provide multiple gridding, segmentation, and extraction methods in Espresso – A Microarray Experiment Management System [8, 9].

REFERENCES

- [1] J. Setubal and J. Meidanis, *Introduction to Computational Molecular Biology*. PWS Publishing Comp., 1st ed., 1997. 1.1, 2.1
- [2] D.E. Bassett Jr., M. B Eisen and M. S. Boguski, “Gene expression informatics - it’s all in your mine,” *Nature genetics supplement*, vol. 21, pp. 51–55, January 1999. 1.1
- [3] M. Schena et al., “Microarrays: biotechnology’s discovery platform for functional genomics,” *TIBTECH*, vol. 16, pp. 301–306, July 1998. 1.1, 2.2
- [4] M. Johnston, “Gene chips: Array of hope for understanding gene regulation,” *Current Biology*, vol. 8, pp. R171–R174, 1998. 1.1, 2.2
- [5] E. S. Lander, “Array of hope,” *Nature Genetics Supplement*, vol. 21, pp. 3–4, January 1999. 1.1, 2.2
- [6] M. Schena, D. Shalon, R.W. Davis and P. O. Brown, “Quantitative monitoring of gene expression patterns with a complementary DNA microarray,” *Science*, vol. 270, p. 467 (4), October 1995. 1.1
- [7] P. Hedge, R. Qi, K. Abernathy, C. Gay, S. Dharap, R. Gaspard, J.E. Hughues, E. Snestrud, N. Lee and J. Quackenbush, “A Concise Guide to cDNA Microarray Analysis,” *BioTechniques*, vol. 29, pp. 548–562, September 2000. 1.1
- [8] L. S. Heath, R. G. Alscher, N. Ramakrishnan, L. Watson and J. Weller, “A Microarray Experiment Mangement System.” NGS Proposal, 2000. 1.1, 6
- [9] R.G. Alscher, B. Chevone, L.S. Heath and N. Ramakrishnan, “A Problem Solving Environment for BioInformatics: Finding Answers with Microarray Technology,” *Proceedings of the High Performance Computing Symposium, 2001. Advanced Technologies Conference, Society for Computer Simulation International, to appear*. 1.1, 6
- [10] M. Eisen, “SCANALYZE User Manual.” Stanford Univ., Stanford, CA, Ver 2.32, 1999. 1.1, 1.2, 3.2.3, 4.2, 4.3
- [11] L. Lazzeroni and A. Owen, “Plaid models for gene expression data,” Stanford BioStatistics Series 211, Dept. of BioStat., Stanford Univ., Stanford, CA, March 2000. 1.1
- [12] M. K. Kerr and G. A. Churchill, “Experimental Design for Gene Expression Micro arrays,” *Biostatistics*, vol. 2, pp. 183–201, 2001. 1.1

- [13] M. K. Kerr and G. A. Churchill, "Statistical Design and the Analysis of Gene Expression," *Genetical Research*, vol. 77, pp. 123–128, 2001. 1.1
- [14] M. K. Kerr, M. Martin and G.A. Churchill, "Analysis of Variance for Gene Expression Microarray Data," *Journal of Computational Biology*, vol. 7, pp. 819–837, 2000. 1.1
- [15] M.A. Newton, C.M Kemdzinski, C.S. Richmond, F.R. Blattner and K.W. Tsui, "On differential variability of expression ratios: Improving statistical inference about gene expression changes from microarray data," *Journal of Computational Biology*, vol. 8, pp. 37–52, 2001. 1.1
- [16] M. Sapir and G. A. Churchill, "Estimating the Posterior Probability of Differential Gene Expression from Microarray Data," 2000. 1.1
- [17] H. A. Davis, D. T. Wong, I. Colbert, S. Soares, J. A. Sorge, R. L. Mullinax, "Normalize and Validate Array Systems Using Exogenous Nucleic Acid Controls," *Strategies, Stratagene* (<http://www.stratagene.com>), vol. 13, pp. 128–130, 2000. 1.1
- [18] J. Schuchhardt, D. Beule, A. Malik, E. Wolski, H. Eickhoff, H. Lehrach and H. Herzog, "Normalization strategies for cDNA microarrays," *Nucleic Acids Research*, vol. 28, no. 10, p. 5, 2000. 1.1, 1.2
- [19] Y. H. Yang, S. Dudoit, Percy Luu and T. P. Speed, "Normalization for cDNA Microarray Data," *SPIE BiOS 2001, San Jose, California*, January 2001. 1.1
- [20] S. Dudoit, Y. H. Yang, M. J. Callow and T. P. Speed, "Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments," Tech. Rep. 578, Dept. of Stat., Univ. of California, Berkeley, August 2000. 1.1
- [21] Y. H. Yang, M. J. Buckley, S. Dudoit and T. P. Speed, "Comparison of methods for image analysis on cDNA microarray data," Tech. Rep. 584, Dept. of Stat., Univ. of California, Berkeley, November 2000. 1.2, 1.2, 4.2, 4.2, 4.3, 4.4.2, 4.4.3, 4.4.3.2, 4.4.4.1, 4.5.3.1
- [22] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Addison-Wesley, 2nd ed., 1992. 1.2, 3.2.1, 3.3.1, 3.5.2, 6
- [23] Y. Chen, E. R. Dougherty and M. L. Bittner, "Ratio-based decision and the quantitative analysis of cDNA microarray images," *Journal of Biomedical Optics*, vol. 2, Oct 1997. 1.2, 1.3, 4.2, 4.5.2, 4.5.2, 4.5.3.2, 4.7, 5.1, 5.1, 6
- [24] R. Adams and L. Bischof, "Seeded region growing," *IEEE Trans. on Pat. Anal. and Mach. Intell.*, vol. 16, pp. 641–647, June 1994. 1.2, 4.2, 4.4, 4.4.1
- [25] G. E. Noether, *Introduction to Statistics: The Nonparametric Way*. Springer-Verlag, 1991. 1.2, 4.2, 4.5, 4.5.1, 4.5.2
- [26] C. Spatz, *Basic Statistics: Tales of Distributions*. Brooks/Cole, 1997. 1.2, 4.2, 4.5, 4.5.2
- [27] A. Mehnert and P. Jackway, "An improved seeded region growing algorithm," *Pattern Recognition Letters*, vol. 18, pp. 1065–1071, 1997. 1.2, 4.2, 4.4, 4.4.1
- [28] V. G. Cheung et al., "Making and reading microarrays," *Nature Genetics Supplement*, vol. 21, pp. 15–19, January 1999. 2.2

- [29] Y. H. Yang, "SPOT User Guide." (<http://www.cmis.csiro.au/IAP/spotmanual.htm>), 2000. 3, 4.2, 4.4.2
- [30] Molecularware., "DigitalGENOME-Analyzer DG." <http://www.molecularware.com/digigenome.htm>, 2000. 3
- [31] J. W. Cooley, P. A. W. Lewis, P. D. Welch, "Historical Notes on the Fast Fourier Transform," *Trans. IEEE*, vol. 15, pp. 76–84, 1967. 3.2.1
- [32] J. W. Cooley, P. A. W. Lewis, P. D. Welch, "The Fast Fourier Transform Algorithm: Programming Consideration in the Calculation of Sine, Cosine and Laplace Transforms," *Journ. of Sound. Vib.*, vol. 12, pp. 315–337, July 1970. 3.2.1
- [33] V. Cizek, *Discrete Fourier Transforms and their applications*. Adam Hilger Ltd., 1986. 3.2.1
- [34] D. H. Ballard, C. M. Brown, "Generalizing the Hough Transform to detect arbitrary shapes," *Pattern Recogn.*, vol. 13, pp. 111–122, 1981. 3.3.1
- [35] R. O. Duda, P. E. Hart, "Use of the Hough Transformation to detect lines and curves in pictures," *Comm. of the ACM*, vol. 15, pp. 11–15, 1972. 3.3.1
- [36] A. Rosenfeld, "Connectivity in Digital Pictures," *Journal of the ACM*, vol. 17, pp. 146–160, January 1970. 3.7, 6
- [37] ScanAlytics Inc., "Microarray suite supplement." User's Guide for use with IPLab for Macintosh, Ver 3.2 or later, 1989-1999. 5.1

Appendix A

Addressing Figures

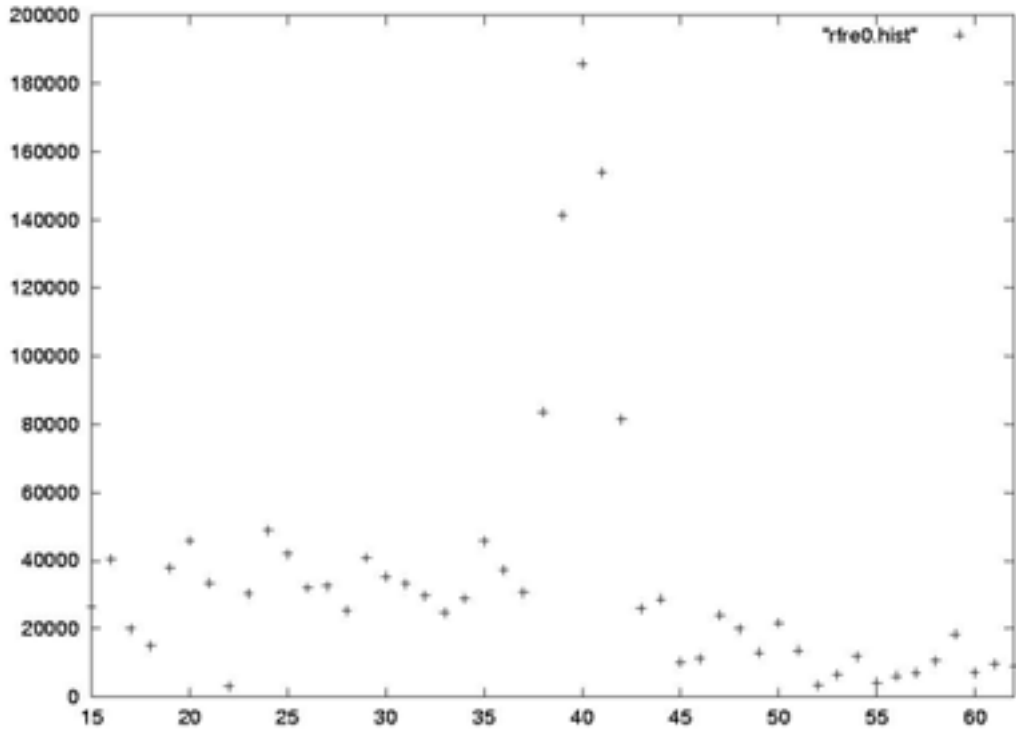


Figure A.1: Frequency band of X_v for pin-array A1 in NS3

In this example, the frequency band $[f_{min}, f_{max}]$ of X_v for the pin-array A1 in NS3. $[f_{min}, f_{max}] = [15, 62]$ and $RH = RW = 1000$. M is occurring at frequency $f_M = 40$. Therefore, $p_M = RH/f_M = 25$.

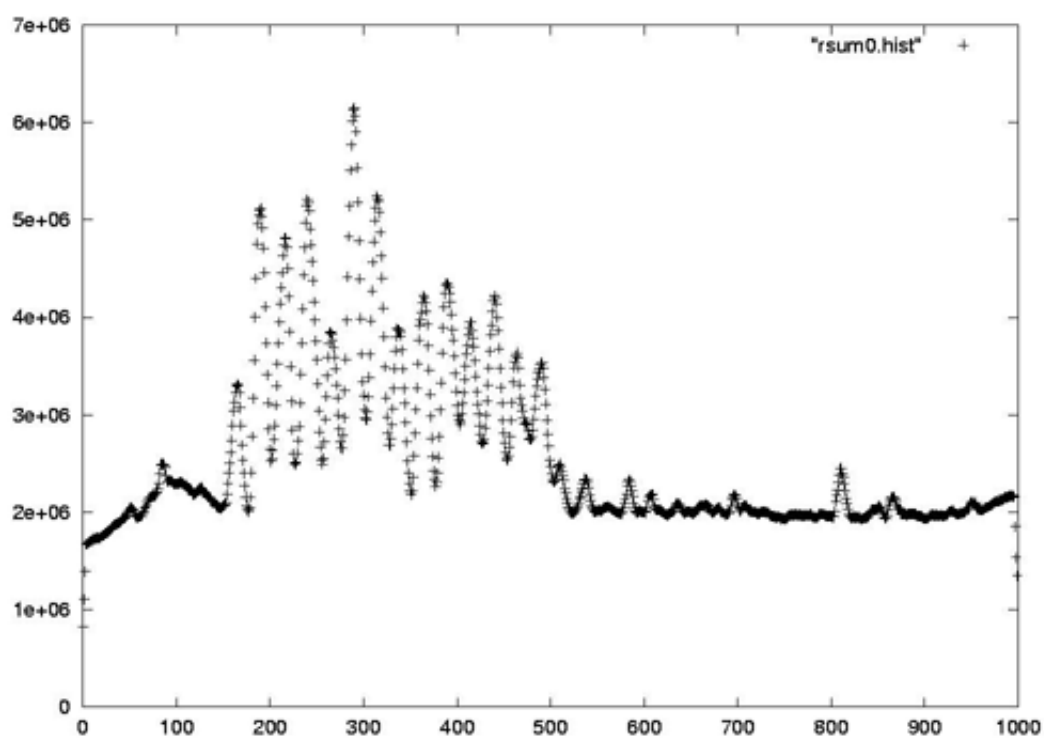


Figure A.2: Sum of the row intensities on A1 of NS3

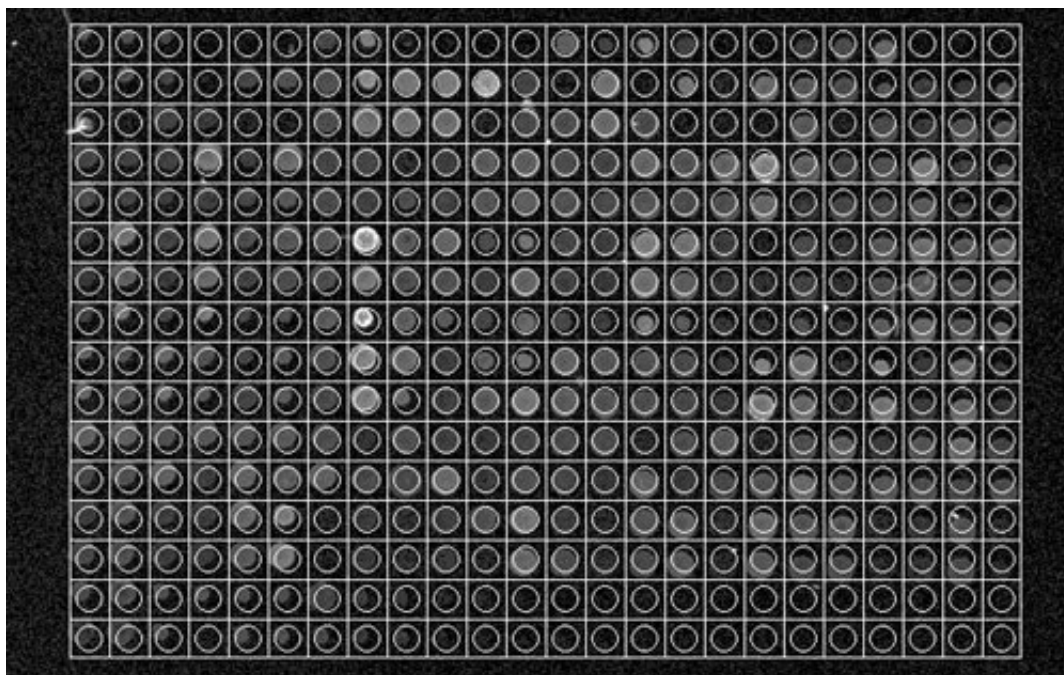


Figure A.3: Gridding on A1 of NS3 with the FFT method

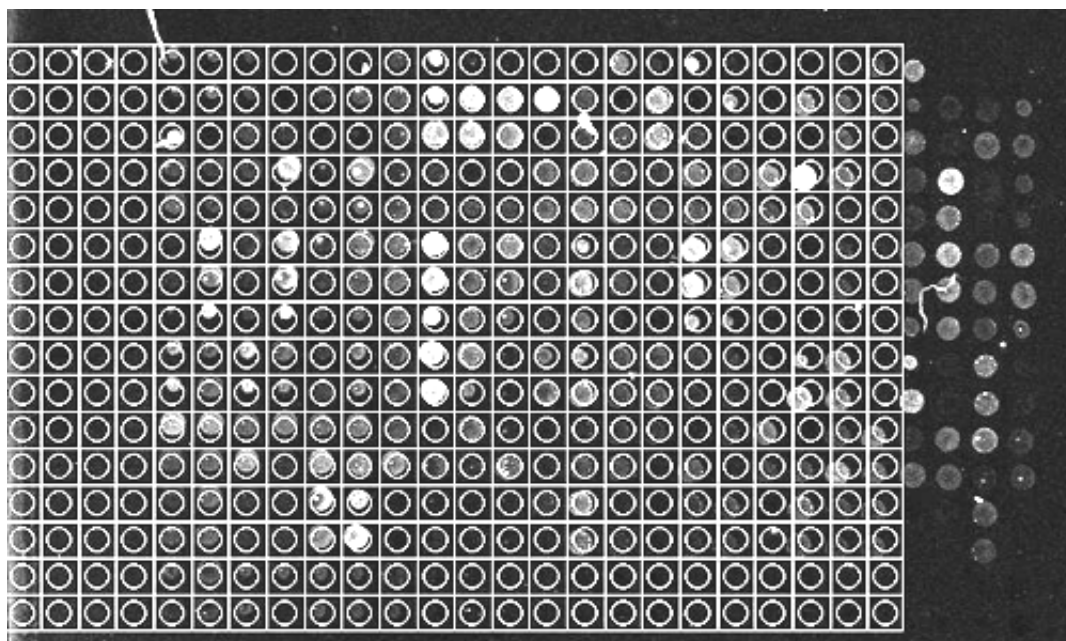


Figure A.4: Gridding on A1 of NS5 with the FFT method.
The noisy edge tend not to appear on this image.

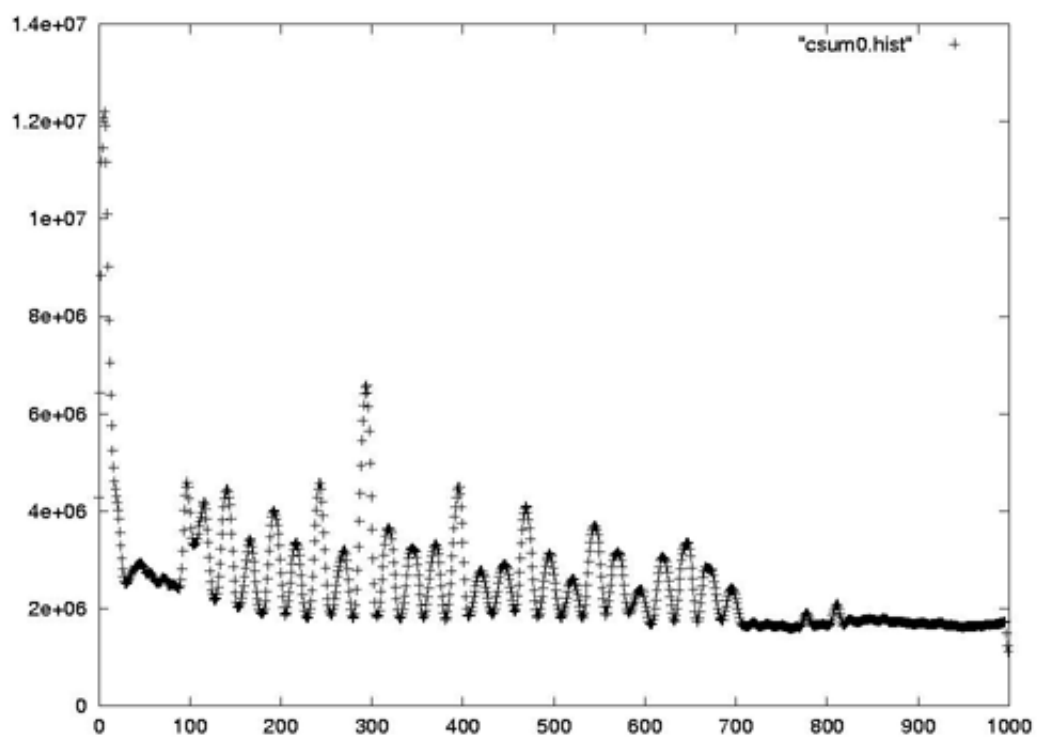


Figure A.5: Column sums on A1 of NS5.

We observe the noisy edge on the left. Our method by using this column sum fails to recognize the succession of 24 peaks. It does not take into account the geometry of the image.

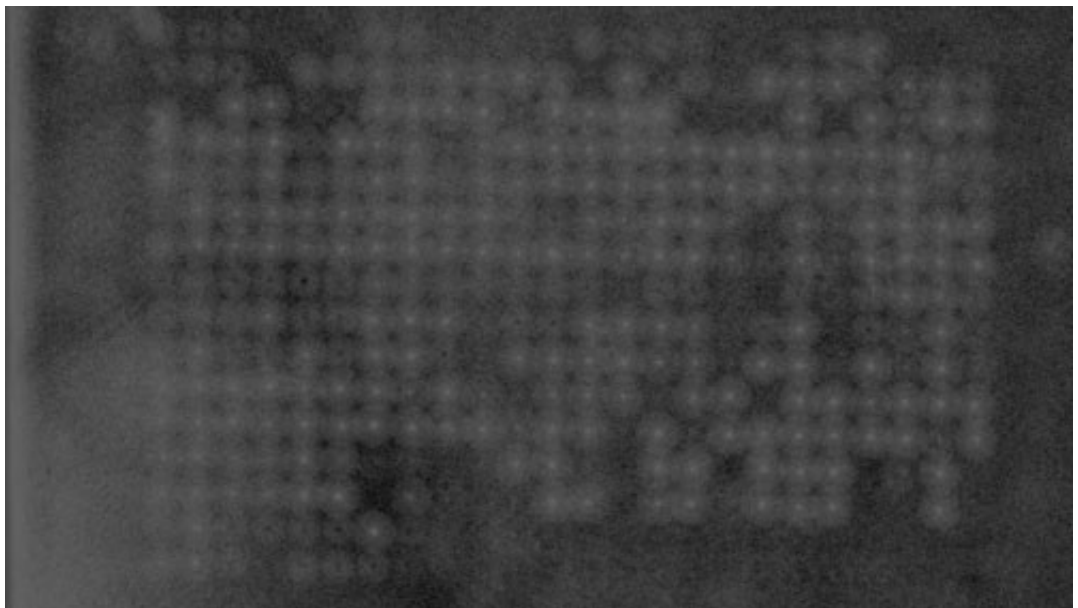


Figure A.6: Hough Transform result on A1 of NS5.

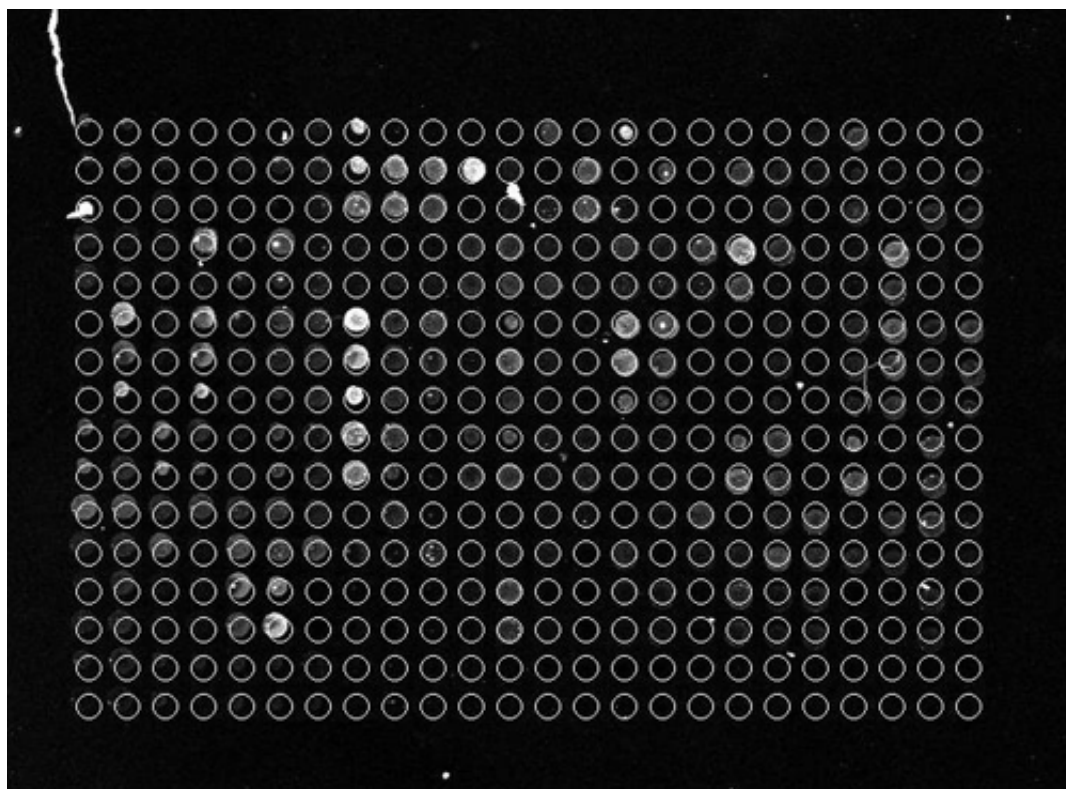


Figure A.7: Gridding on A1 of NS5 with the CHT method. The gridding is correct but the image is not too noisy and the target have nice shape. Our program is not taking care of the tilting of the pin-array. An adjustment is needed.

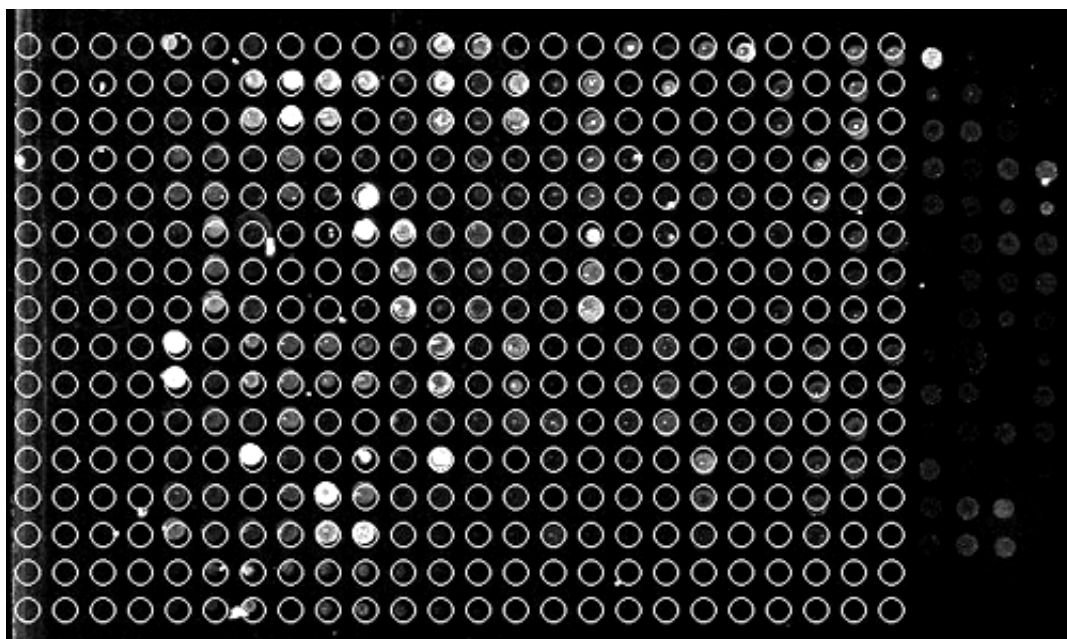


Figure A.8: Gridding on A2 of NS5 with the CHT method.
The gridding is completely off to the left border of the image because of a strong noise in this area.

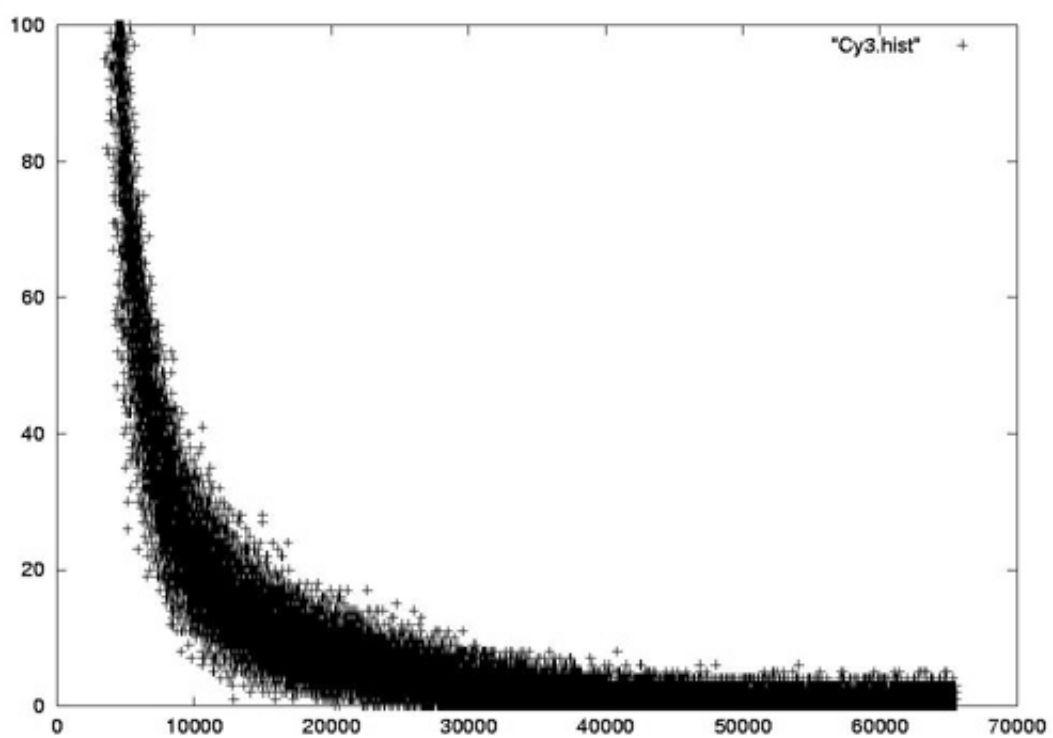


Figure A.9: Histogram of NS3, channel 1(Cy3 dye).

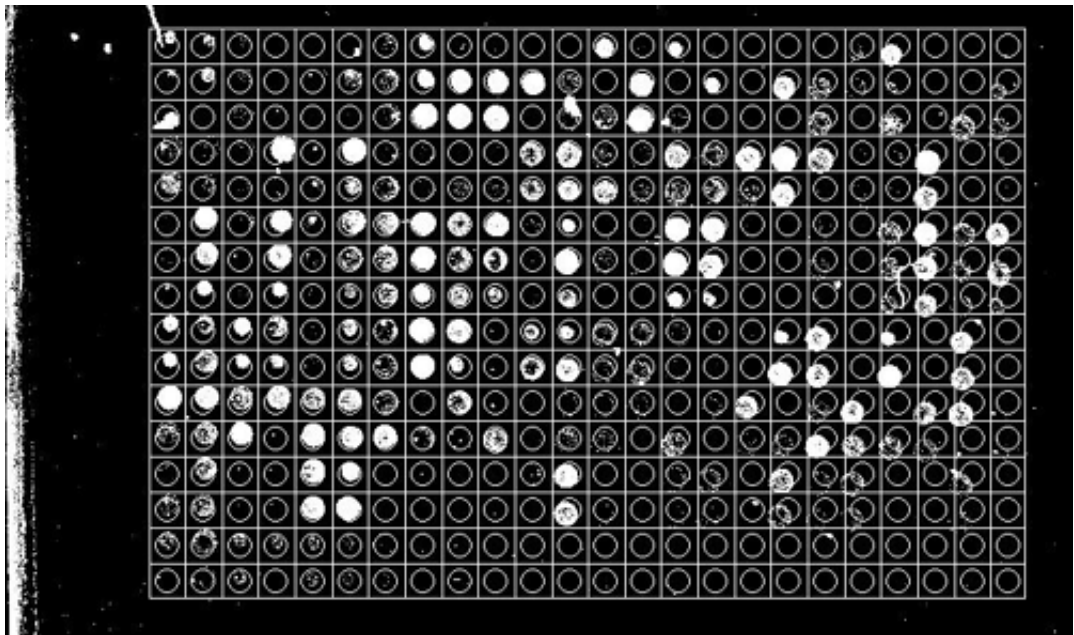


Figure A.10: Gridding on A1 of NS5 after preprocessing with the FFT method.

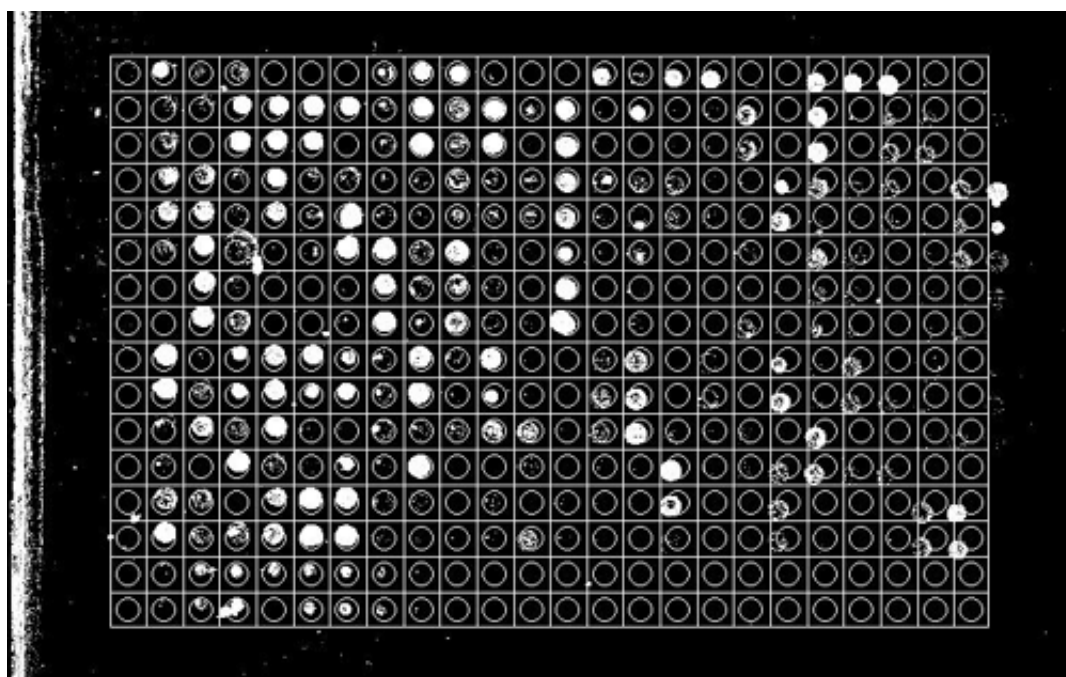


Figure A.11: Gridding on A2 of NS5 image after preprocessing with the FFT method. The result is off by a column. We attribute this result to the tilting problem. In the last right columns the circles are almost not overlapping the targets.

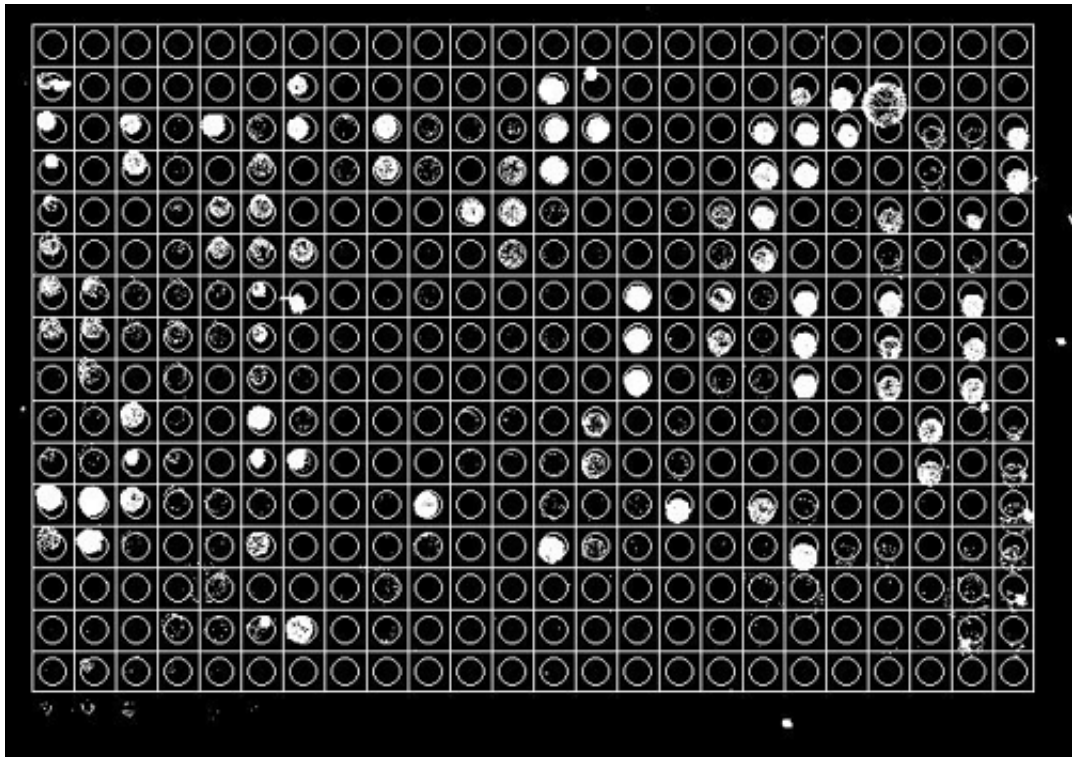


Figure A.12: Gridding on A3 of NS5 after preprocessing with the FFT method. The result is off by a row. The tilting effect is again the possible problem.

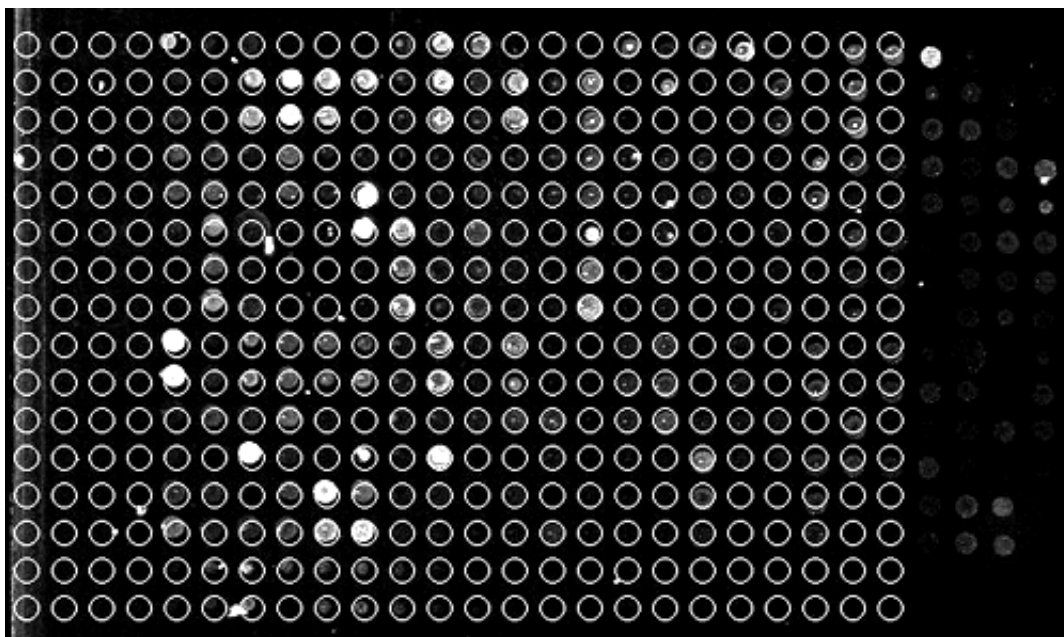


Figure A.13: Gridding on A1 of NS5 with the hybrid method.

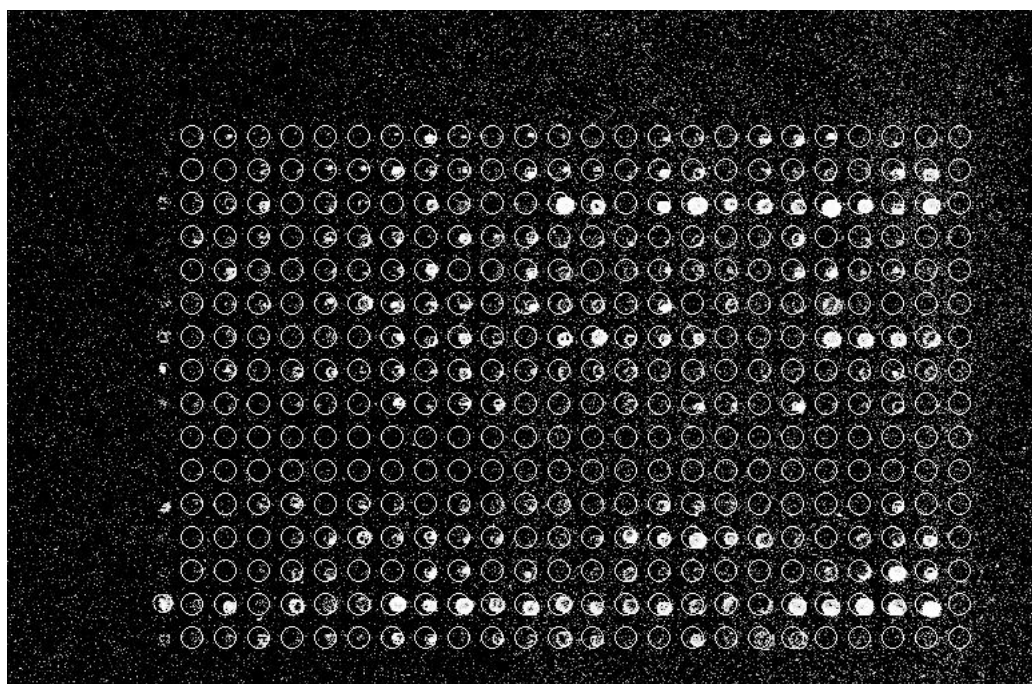


Figure A.14: Gridding on A1 of S4X3 thresholded with the hybrid method.

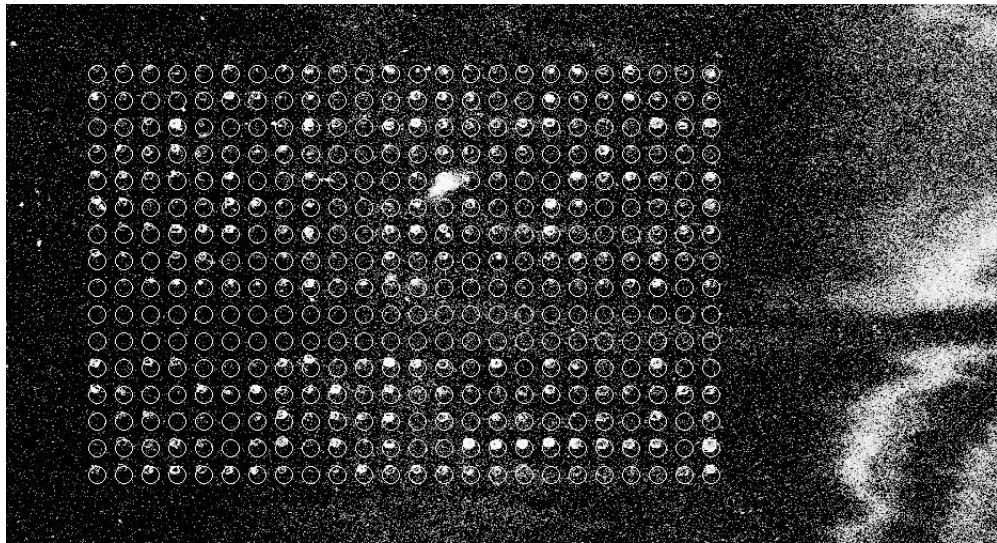


Figure A.15: Gridding on A3 of S4X3 thresholded with the hybrid method.



Figure A.16: Artifacts leading to wrong gridding of A2 and A4 – (S4X3,S4X5) thresholded.

Appendix B

SRG Figures

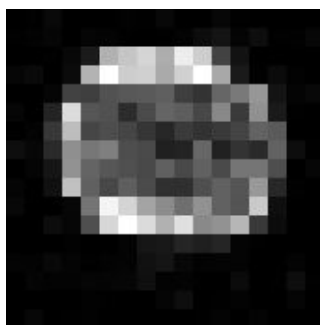


Figure B.1: Target (1, 7, 4, Cy3) or “beignet” zoomed in.

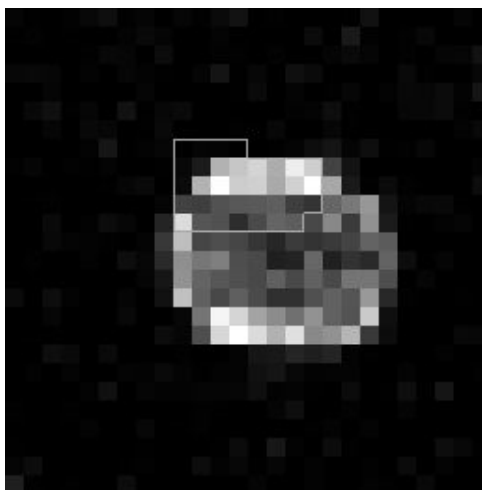


Figure B.2: SRG result on the target (1, 7, 4, Cy3) with a seedsize of 2.

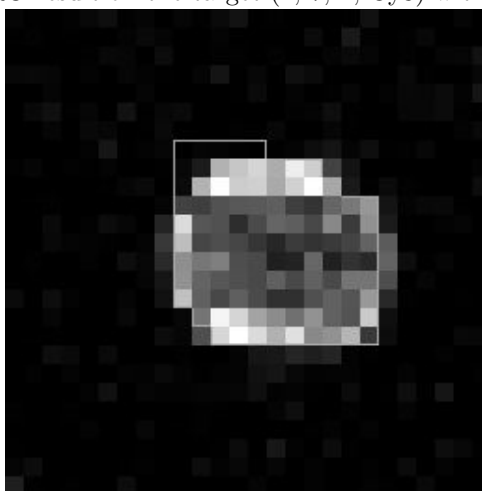


Figure B.3: SRG result on the target (1, 7, 4, Cy3) with a seedsize of 3.

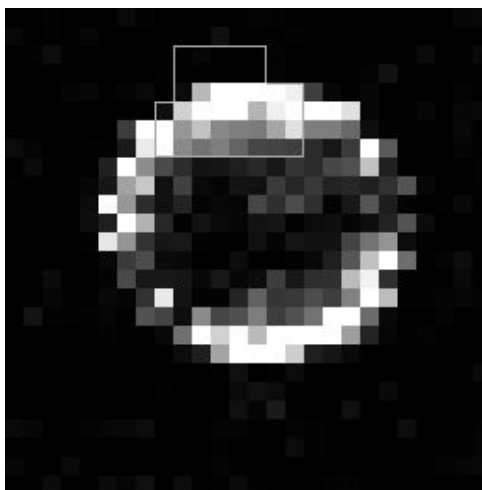


Figure B.4: SRG result on the target (1, 7, 23, Cy3) with a “Max” seed

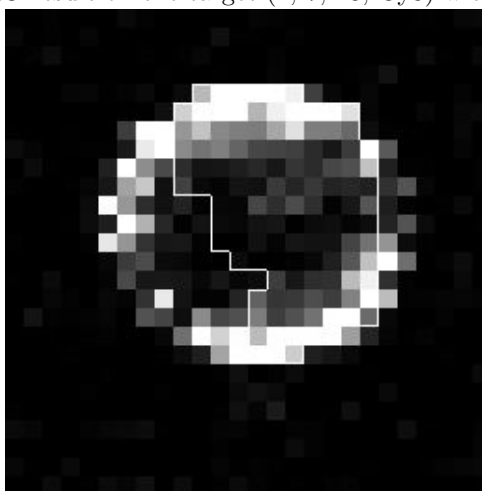


Figure B.5: SRG result on the target (1, 7, 23, Cy3) with a “Center” Seed

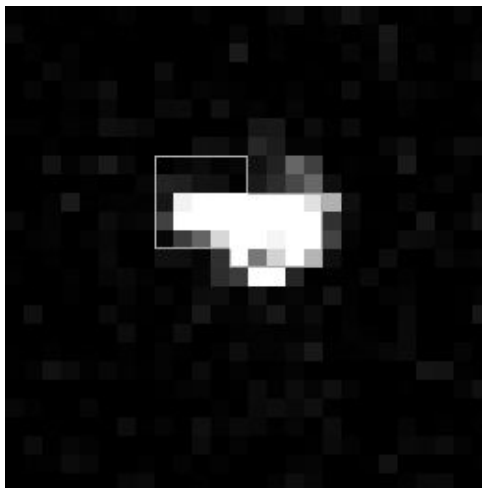


Figure B.6: SRG result on the target (1, 5, 3, Cy3) with a "Max" seed

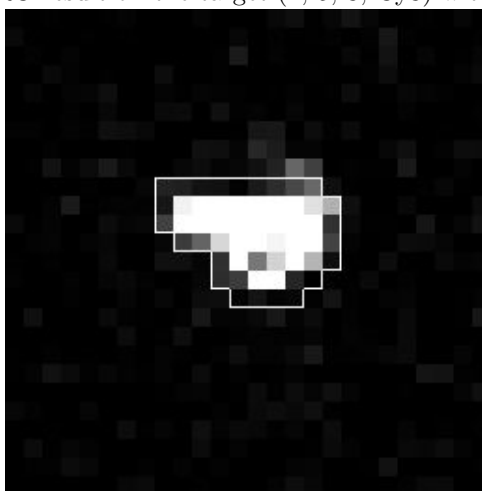


Figure B.7: SRG result on the target (1, 5, 3, Cy3) with a "Maximum Region" seed

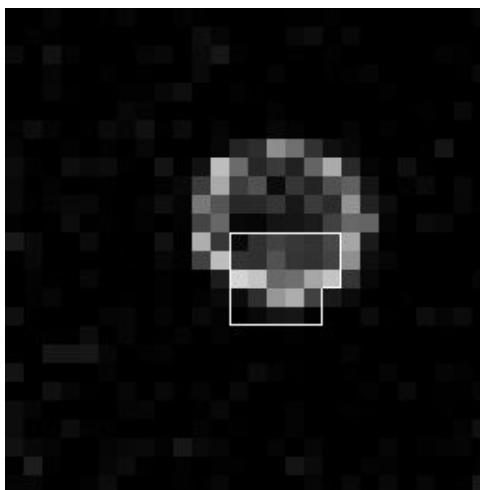


Figure B.8: SRG result on the target (1, 7, 10, Cy3) with a "Maximum Region" seed

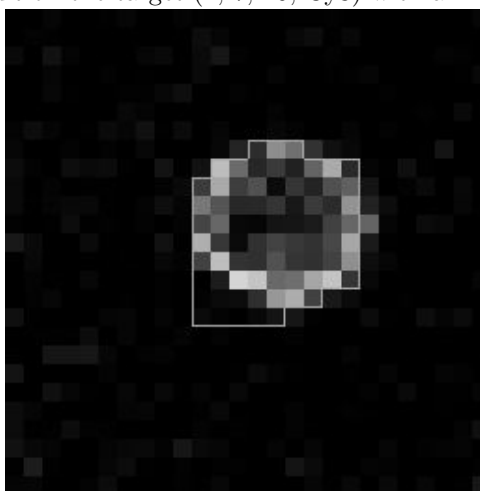


Figure B.9: SRG result on the target (1, 7, 10, Cy3) with a "Max" seed

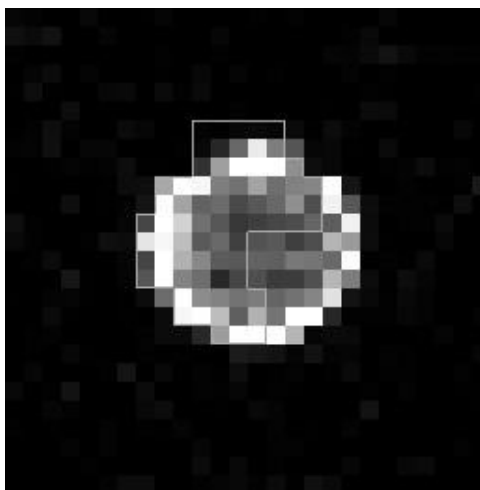


Figure B.10: SRG result on the target (1, 3, 14, Cy3) with a “Max” seed.

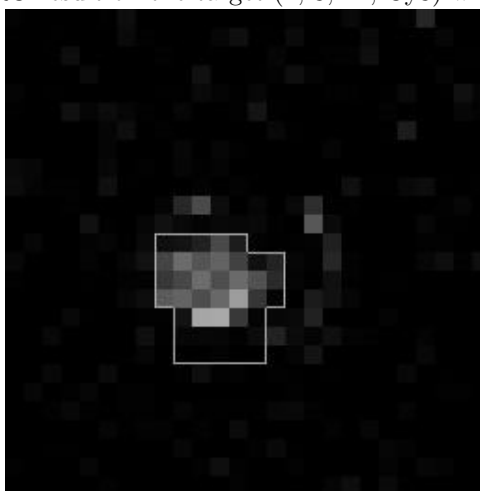


Figure B.11: SRG result on the target (1, 3, 17, Cy3) with a “Max” seed.

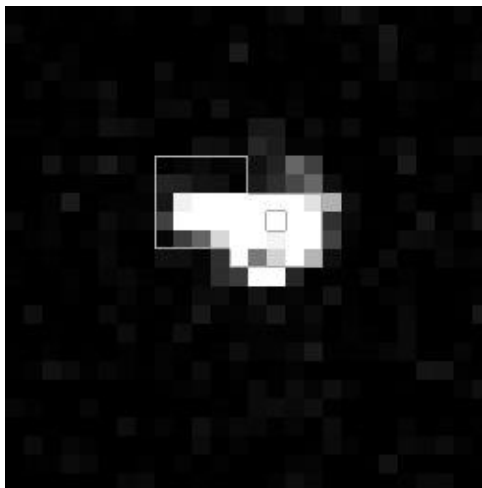


Figure B.12: SRG result on the target (1, 5, 3, Cy3) with a “Max” seed.

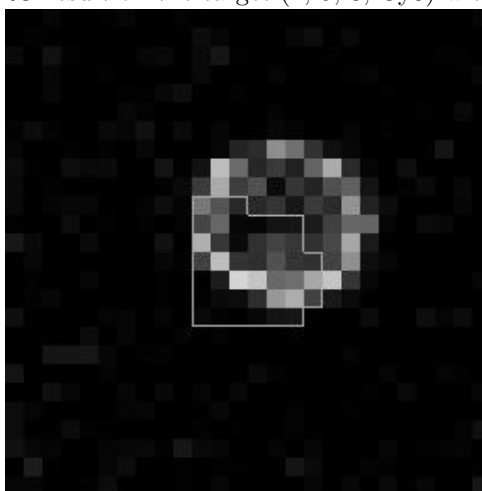


Figure B.13: SRG result on the target (1, 7, 10, Cy3) with a “Max” seed.

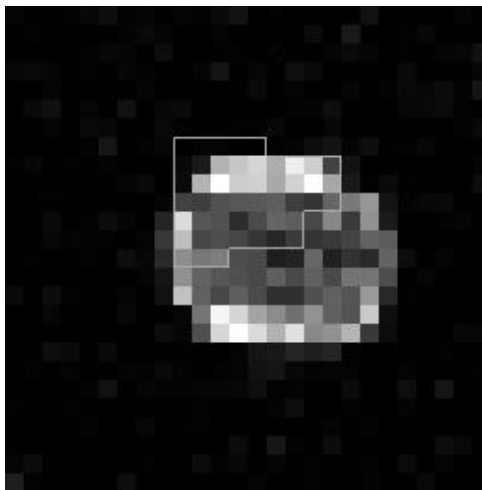


Figure B.14: SRG result on the target (1, 7, 4, Cy3) with a “Max” seed.

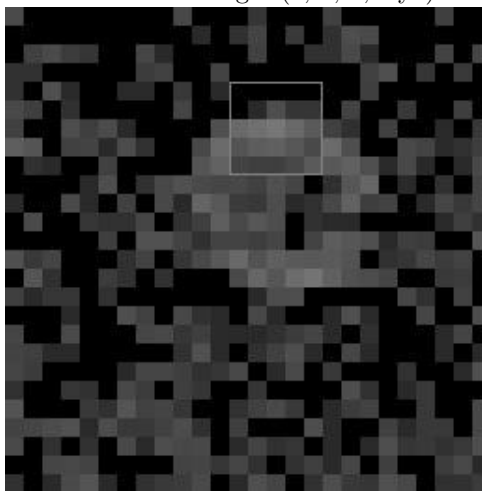


Figure B.15: SRG result on the target (1, 7, 9, Cy3) with a “Max” seed.

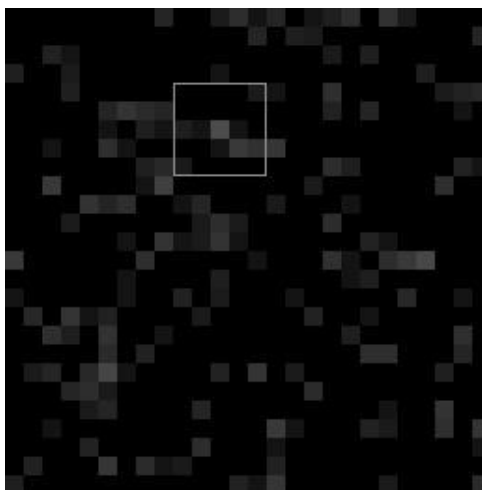


Figure B.16: SRG result on the target (2, 15, 7, Cy3) with a “Max” seed.

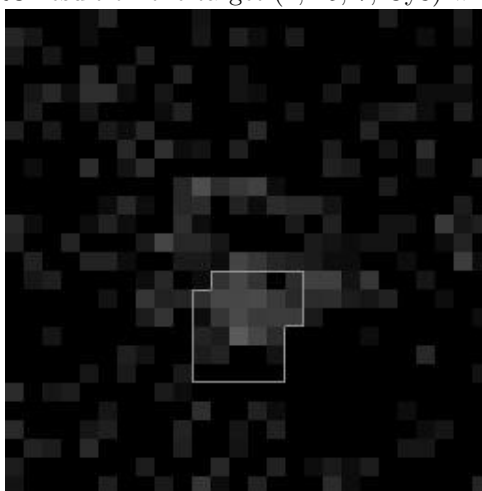


Figure B.17: SRG result on the target (2, 6, 15, Cy3) with a “Max” seed.

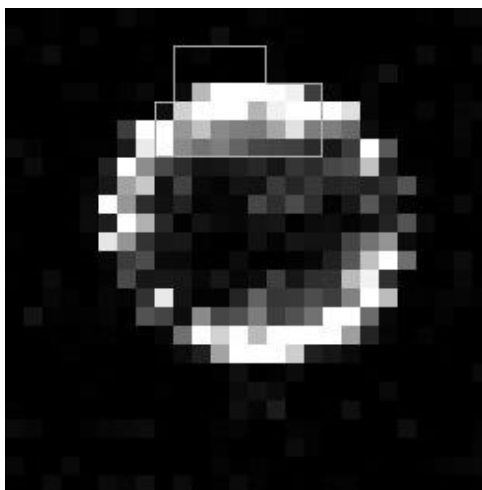


Figure B.18: SRG result on the target (1, 7, 23, Cy3) with a “Max” seed.

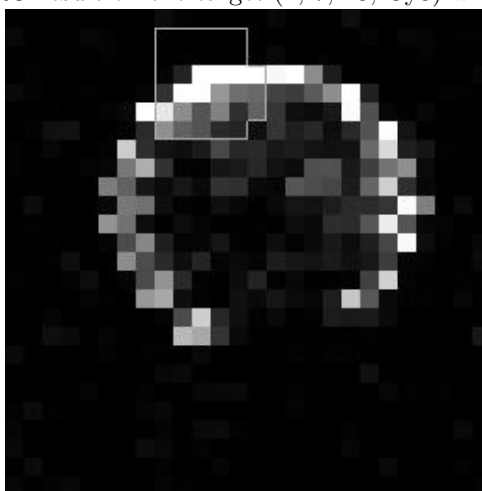


Figure B.19: SRG result on the target (1, 7, 24, Cy3) with the “Max” seed.

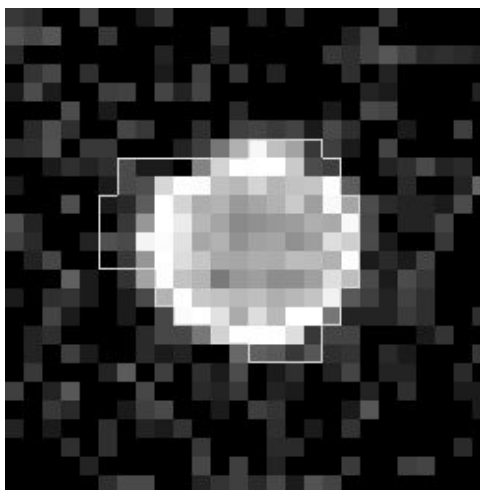


Figure B.20: SRG result on the target (1, 3, 14, Cy3) with a “Random” seed

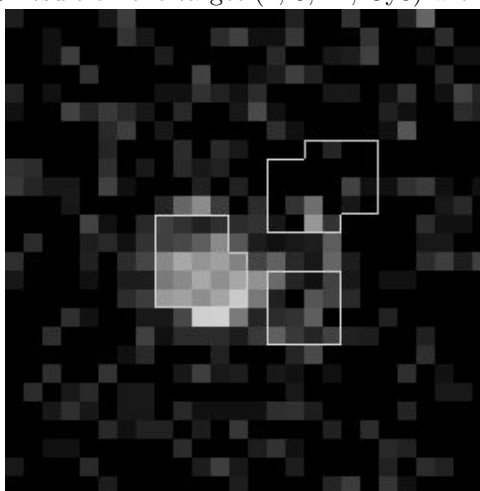


Figure B.21: SRG result on the target (1, 3, 17, Cy3) with a “Random” seed

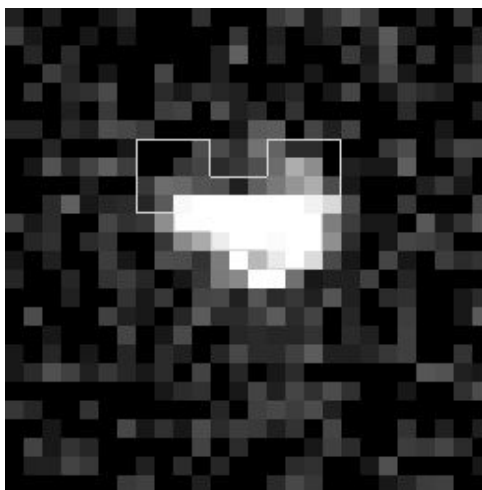


Figure B.22: SRG result on the target (1, 5, 3, Cy3) with a “Random” seed

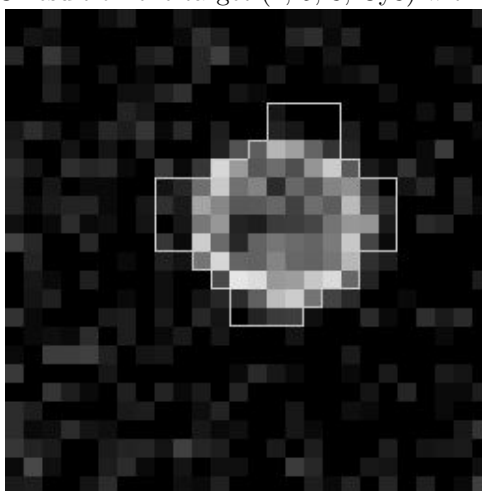


Figure B.23: SRG result on the target (1, 7, 10, Cy3) with a “Random” seed

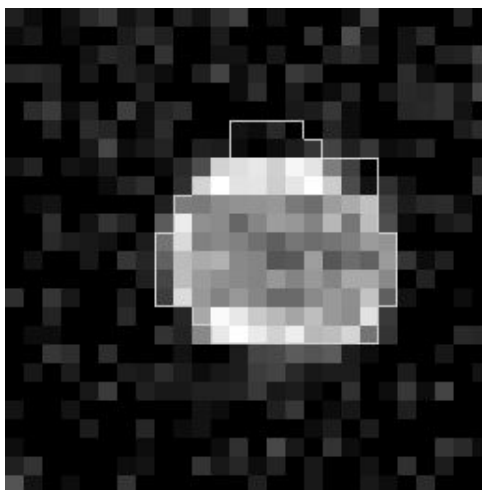


Figure B.24: SRG result on the target (1, 7, 4, Cy3) with a “Random” seed

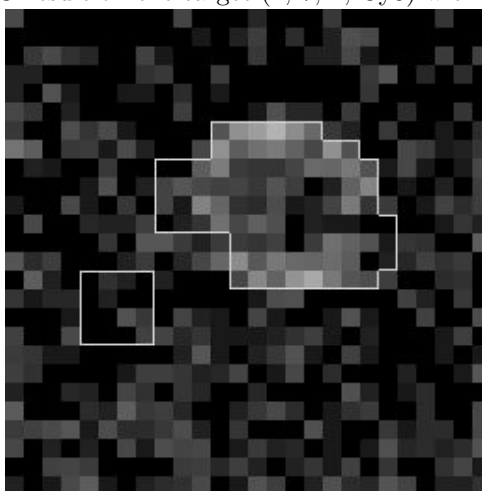


Figure B.25: SRG result on the target (1, 7, 9, Cy3) with a “Random” seed

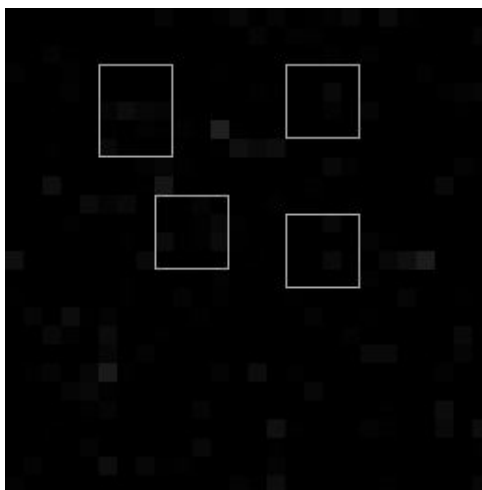


Figure B.26: SRG result on the target (2, 15, 7, Cy3) with a “Random” seed

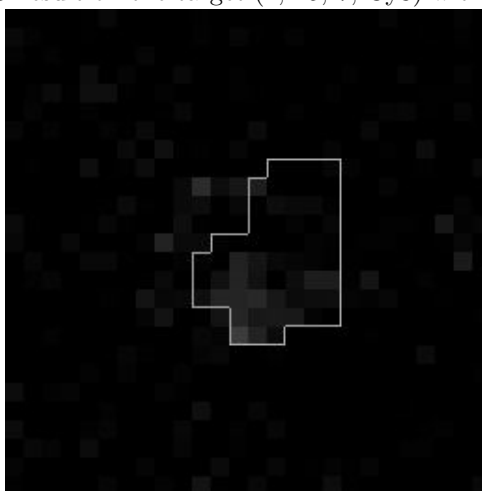


Figure B.27: SRG result on the target (2, 6, 15, Cy3) with a “Random” seed

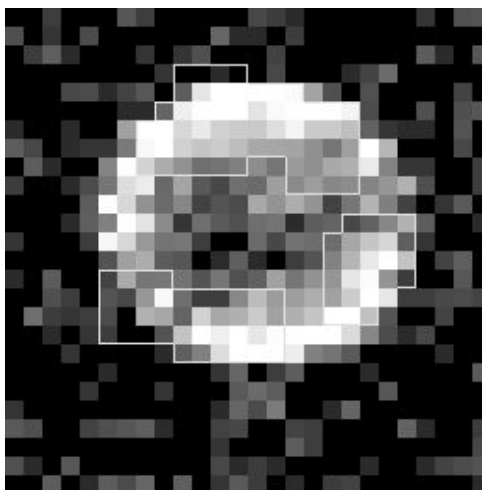


Figure B.28: SRG result on the target (1, 7, 23, Cy3) with a “Random” seed

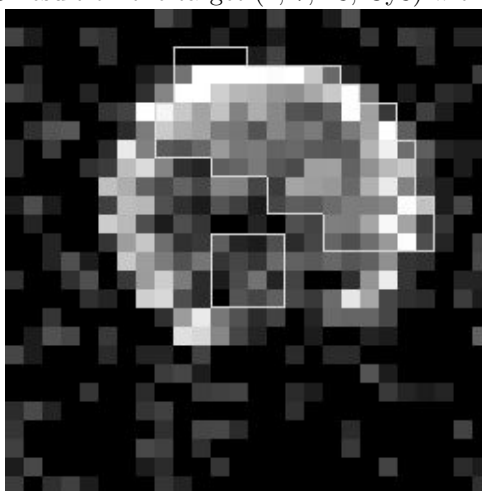


Figure B.29: SRG result on the target (1, 7, 24, Cy3) with a “Random” seed

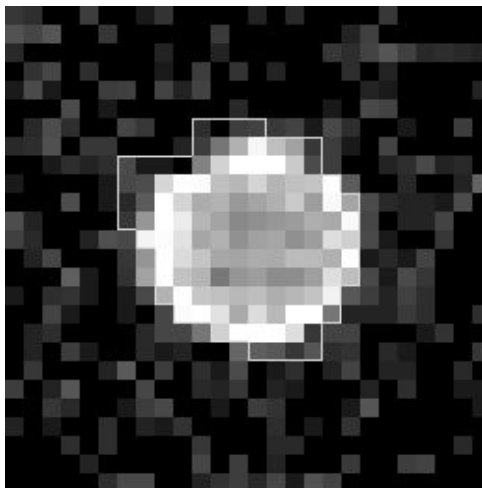


Figure B.30: SRG result on the target (1, 3, 14, Cy3) with a “Union” seed.

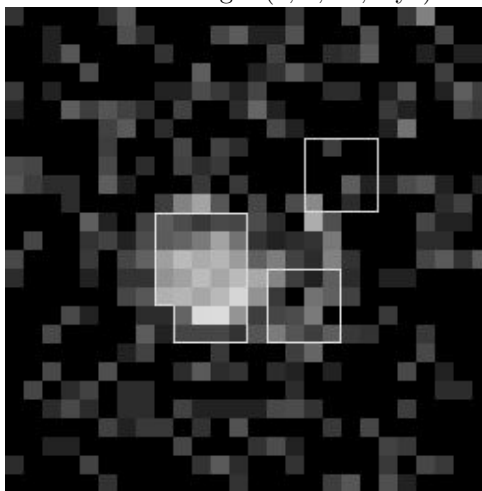


Figure B.31: SRG result on the target (1, 3, 17, Cy3) with a “Union” seed.

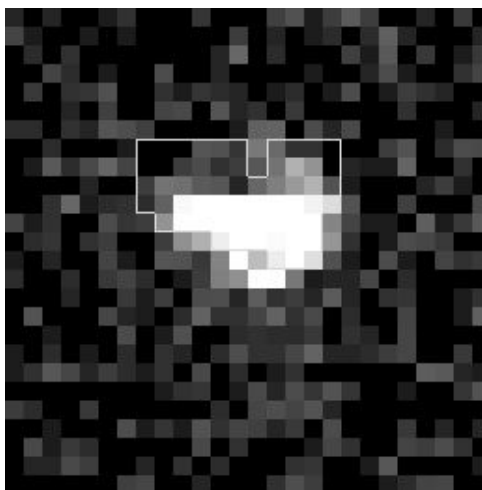


Figure B.32: SRG result on the target (1, 5, 3, Cy3) with a “Union” seed.

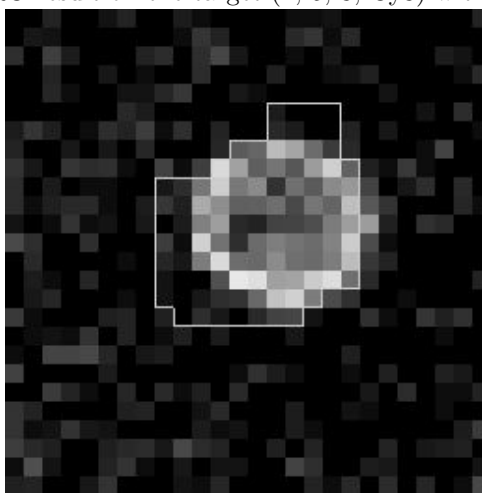


Figure B.33: SRG result on the target (1, 7, 10, Cy3) with a “Union” seed.

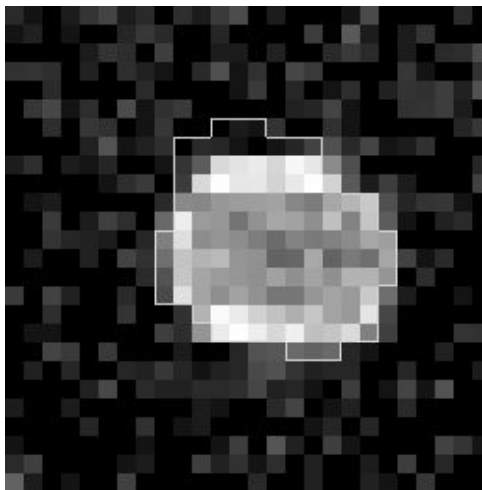


Figure B.34: SRG result on the target (1, 7, 4, Cy3) with a “Union” seed.

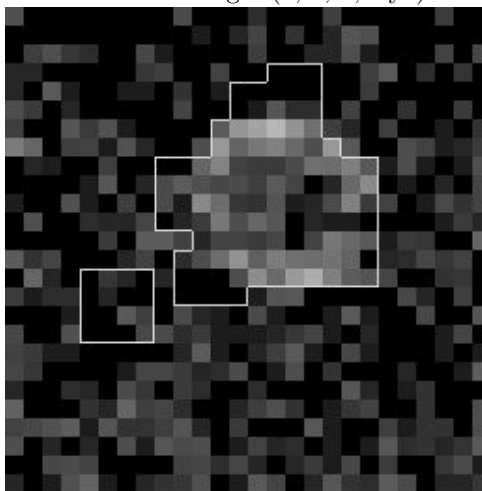


Figure B.35: SRG result on the target (1, 7, 9, Cy3) with a “Union” seed.

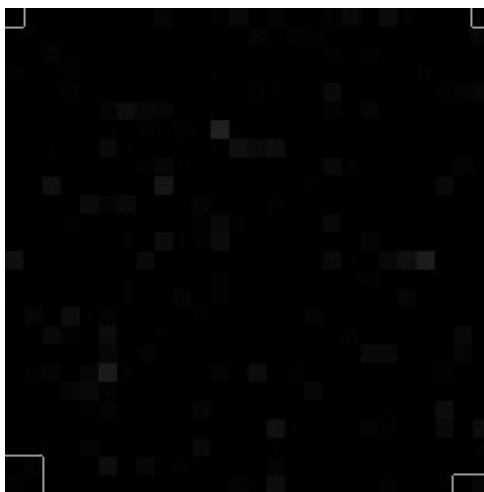


Figure B.36: SRG result on the target (2, 15, 7, Cy3) with a “Union” seed.

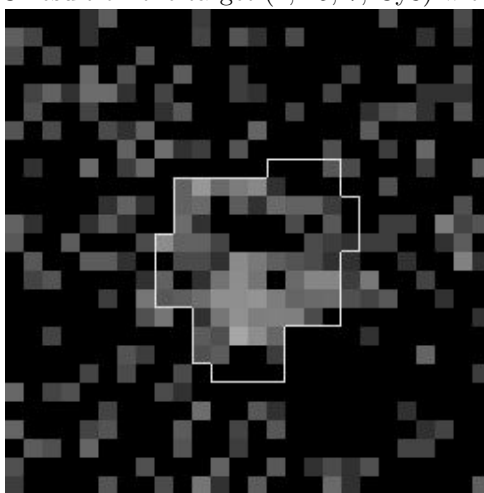


Figure B.37: SRG result on the target (2, 6, 15, Cy3) with a “Union” seed.

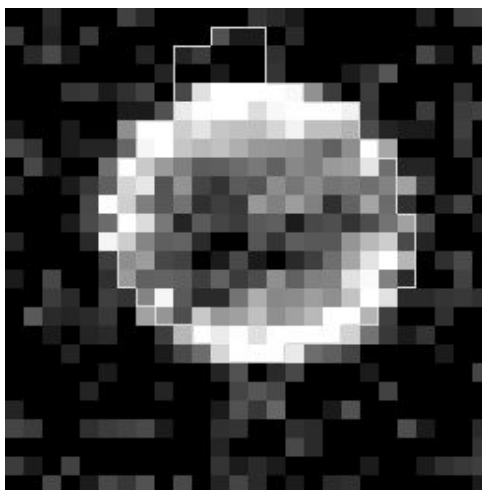


Figure B.38: SRG result on the target (1, 7, 23, Cy3) with a “Union” seed.

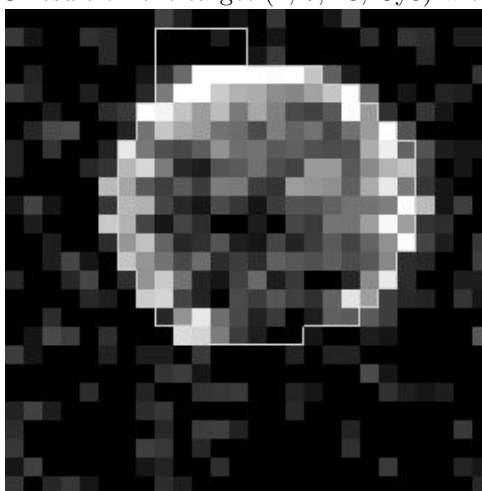


Figure B.39: SRG result on the target (1, 7, 24, Cy3) with a “Union” seed.

Appendix C

MWT Figures

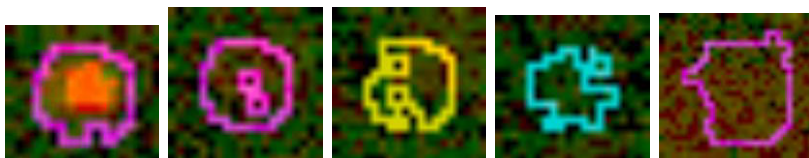


Figure C.1: Target areas obtained by MS after the application of the MWT on the image Cy3_S3

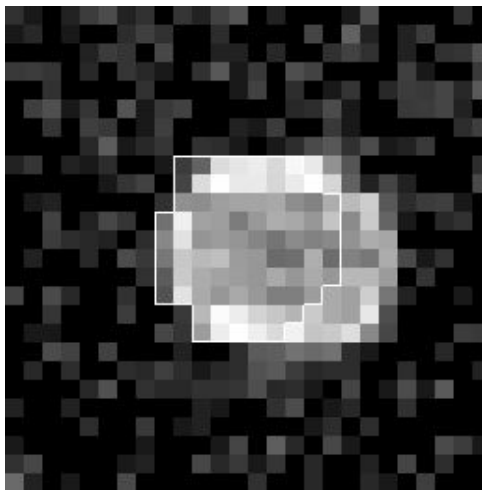


Figure C.2: Result of the MWT on the target (1, 7, 4, Cy3) with $R = 5$ and $U_0 = 0$

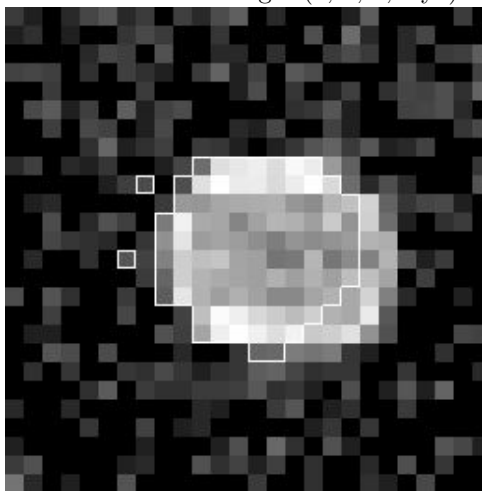


Figure C.3: Result of the MWT on the target (1, 7, 4, Cy3) with $R = 6$ and $U_0 = 0$

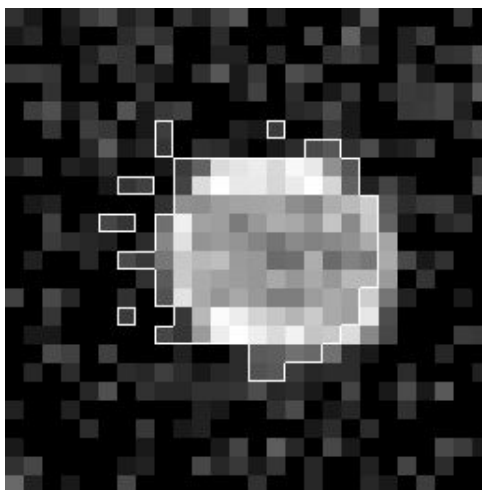


Figure C.4: Result of the MWT on the target (1, 7, 4, Cy3) with $R = 7$ and $U_0 = 0$

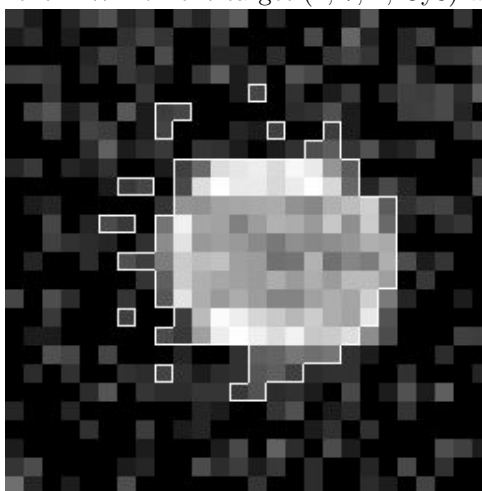


Figure C.5: Result of the MWT on the target (1, 7, 4, Cy3) with $R = 8$ and $U_0 = 0$

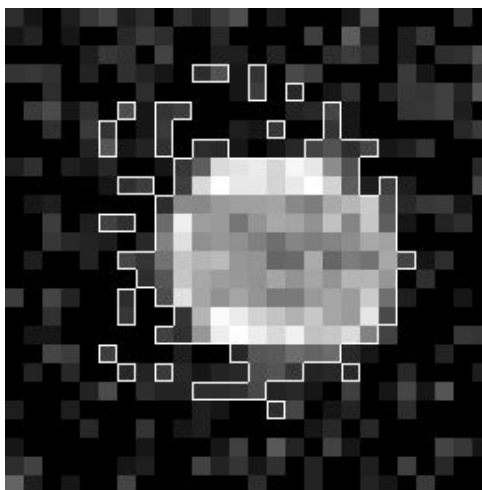


Figure C.6: Result of the MWT on the target (1, 7, 4, Cy3) with $R = 9$ and $U_0 = 0$

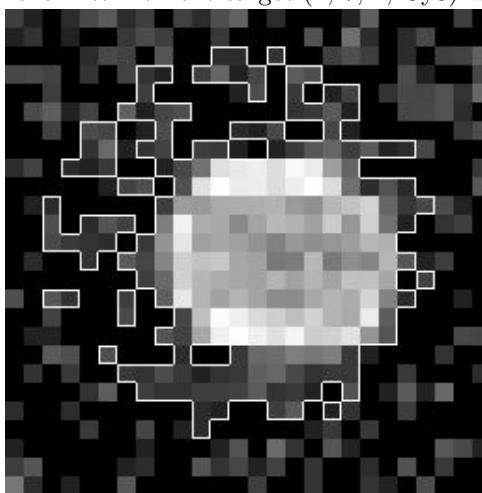


Figure C.7: Result of the MWT on the target (1, 7, 4, Cy3) with $R = 10$ and $U_0 = 0$

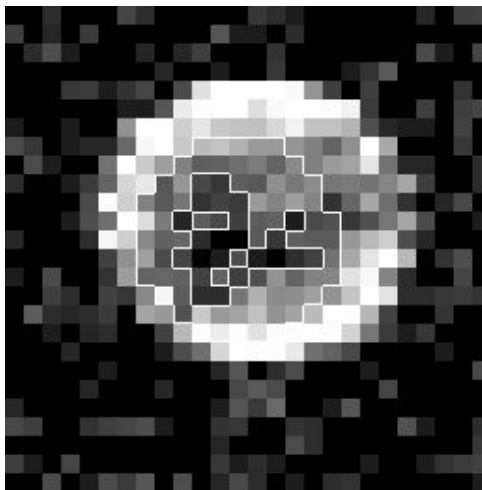


Figure C.8: Result of the MWT on the target (1, 7, 23, Cy3) with $R = 5$ and $U_0 = 0$

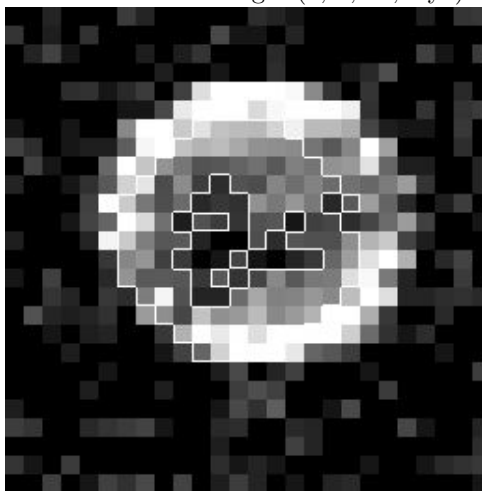


Figure C.9: Result of the MWT on the target (1, 7, 23, Cy3) with $R = 6$ and $U_0 = 0$

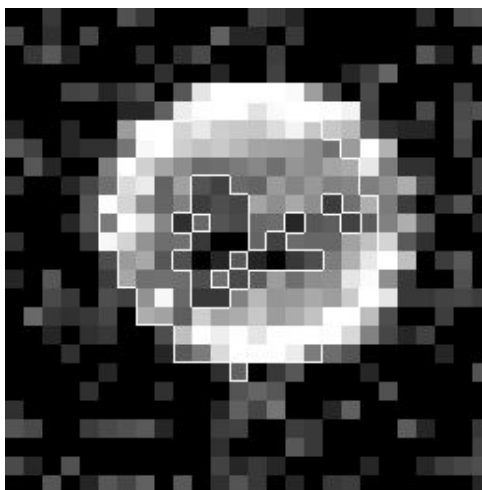


Figure C.10: Result of the MWT on the target (1, 7, 23, Cy3) with $R = 7$ and $U_0 = 0$

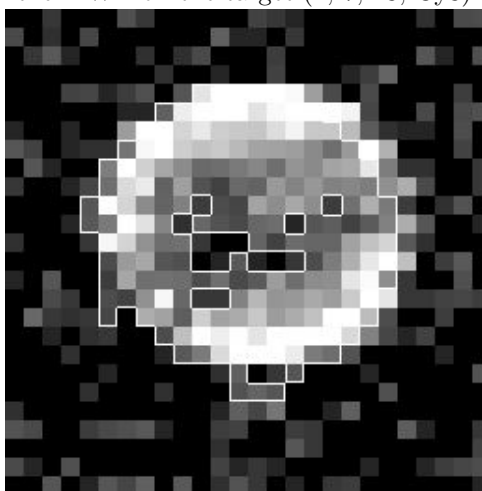


Figure C.11: Result of the MWT on the target (1, 7, 23, Cy3) with $R = 8$ and $U_0 = 0$

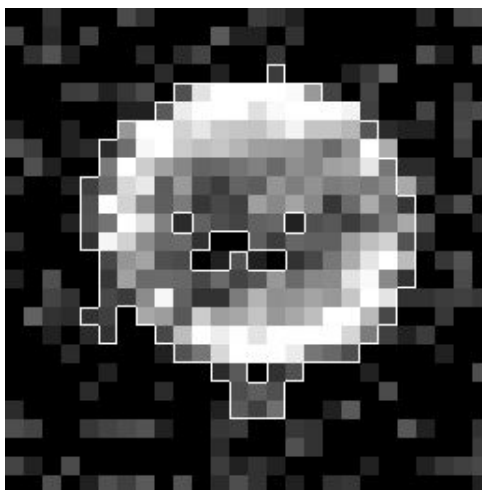


Figure C.12: Result of the MWT on the target (1, 7, 23, Cy3) with $R = 9$ and $U_0 = 0$

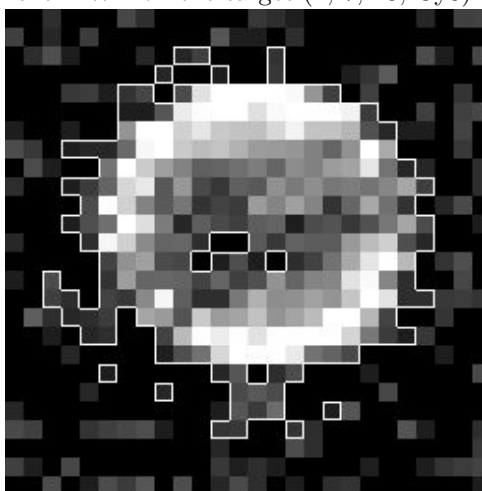


Figure C.13: Result of the MWT on the target (1, 7, 23, Cy3) with $R = 10$ and $U_0 = 0$

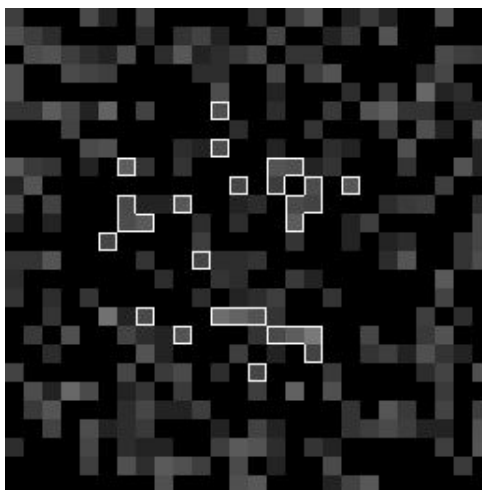


Figure C.14: Noise segmentation above the target (1, 1, 8, Cy3) with $R = 7$ and $U_0 = 0$.

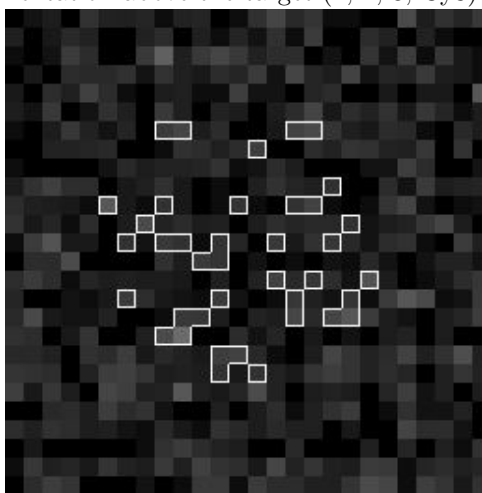


Figure C.15: Noise segmentation above the target (1, 1, 8, Cy5) with $R = 7$ and $U_0 = 0$.

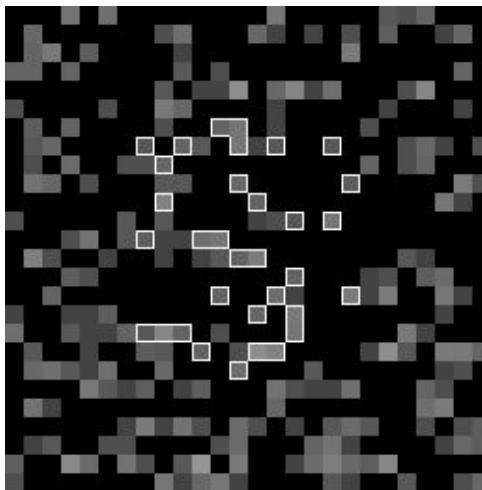


Figure C.16: Noise segmentation above the target (1, 1, 14, Cy3) with $R = 7$ and $U_0 = 0$.

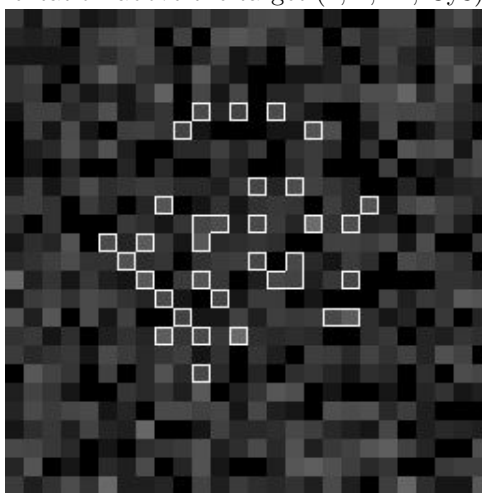


Figure C.17: Noise segmentation above the target (1, 1, 14, Cy5) with $R = 7$ and $U_0 = 0$.

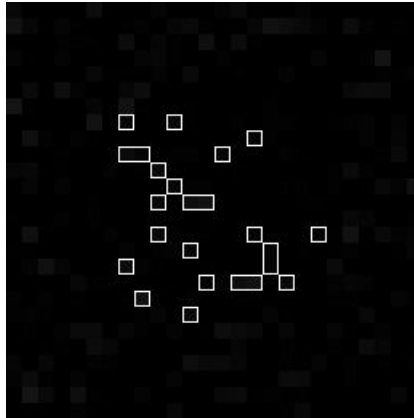


Figure C.18: Noise segmentation left to the target $(1, 7, 1, \text{Cy}3)$ with $R = 7$ and $U_0 = 0$.

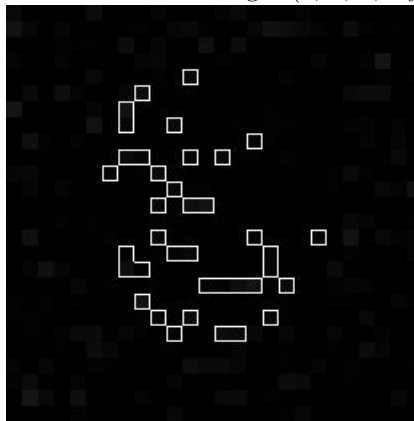


Figure C.19: Noise segmentation left to the target $(1, 7, 1, \text{Cy}3)$ with $R = 8$ and $U_0 = 0$.

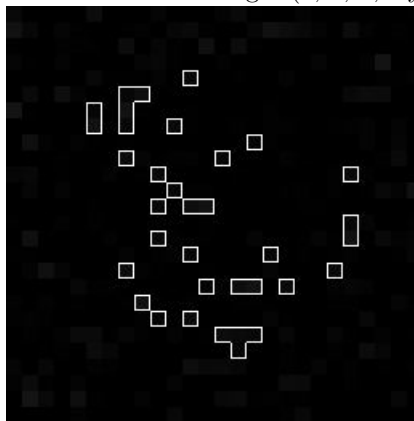


Figure C.20: Noise segmentation left to the target $(1, 7, 1, \text{Cy}3)$ with $R = 9$ and $U_0 = 0$.

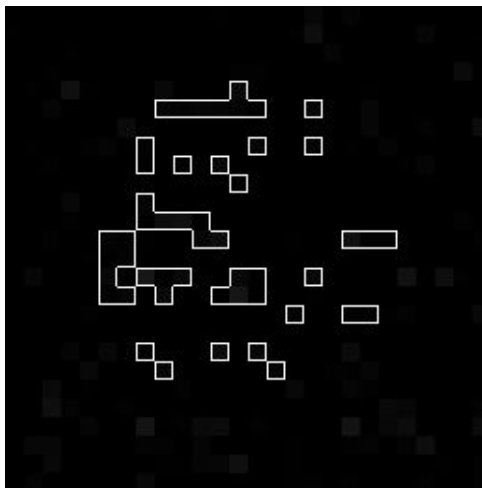


Figure C.21: Noise segmentation left to the target $(2, 7, 1, \text{Cy}3)$ with $R = 8$ and $U_0 = 0$.

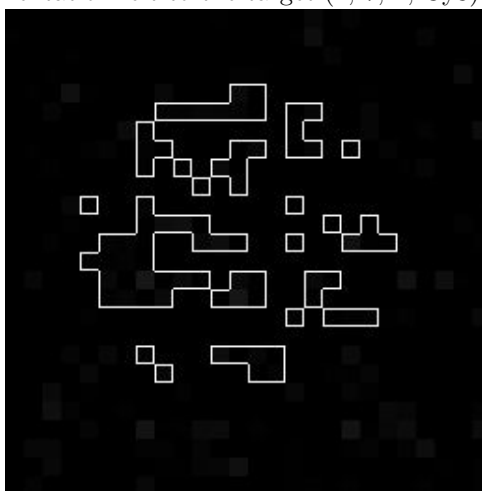


Figure C.22: Noise segmentation left to the target $(2, 7, 1, \text{Cy}3)$ with $R = 8$ and $U_0 = 8$.

Appendix D

Data Extraction and Analysis

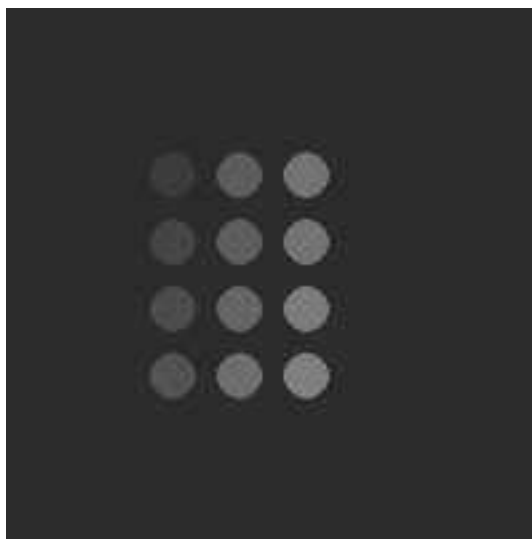


Figure D.1: Image of perfect targets used as Channel 1

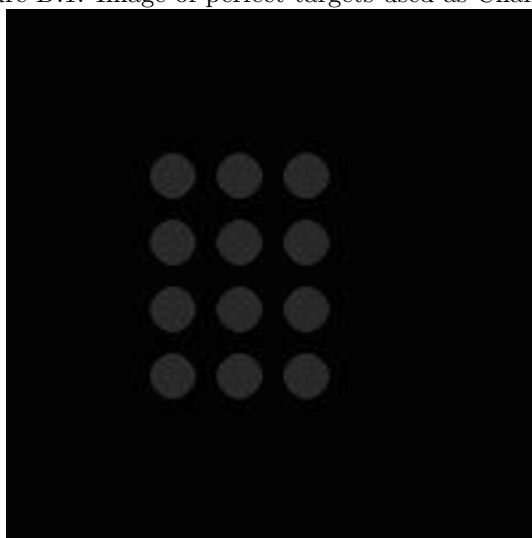


Figure D.2: Image of perfect targets used as Channel 2

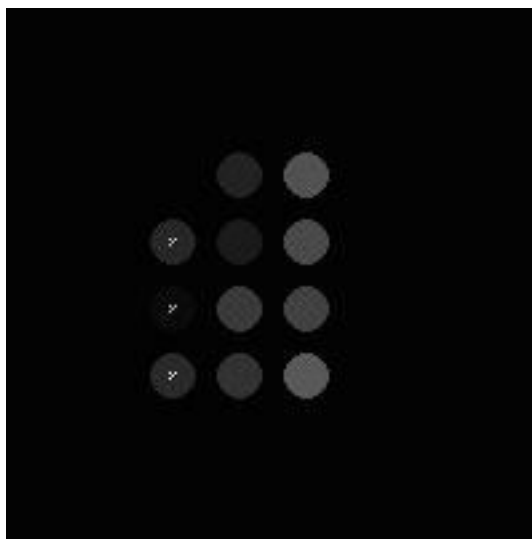


Figure D.3: Targets used as Channel 1 to check the 4 highest-lowest pixels tossing hypothesis.

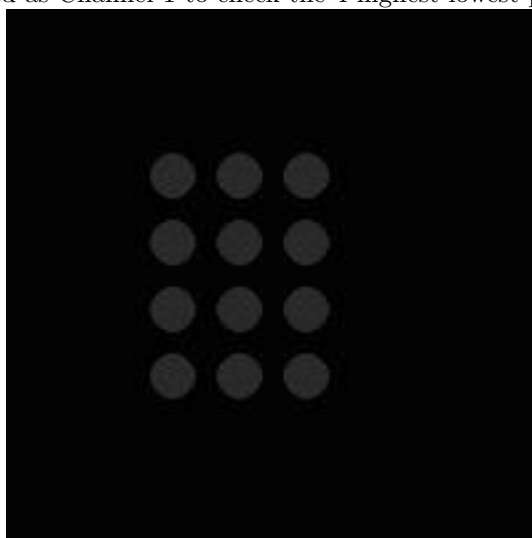


Figure D.4: Targets used as Channel 2 to check the 4 highest-lowest pixels tossing hypothesis.

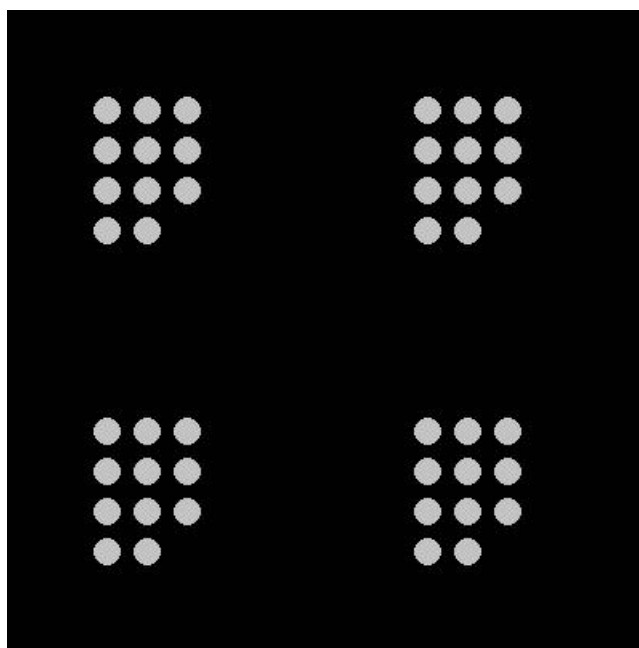


Figure D.5: Targets in Channel 1 to test the radius influence on MS results.

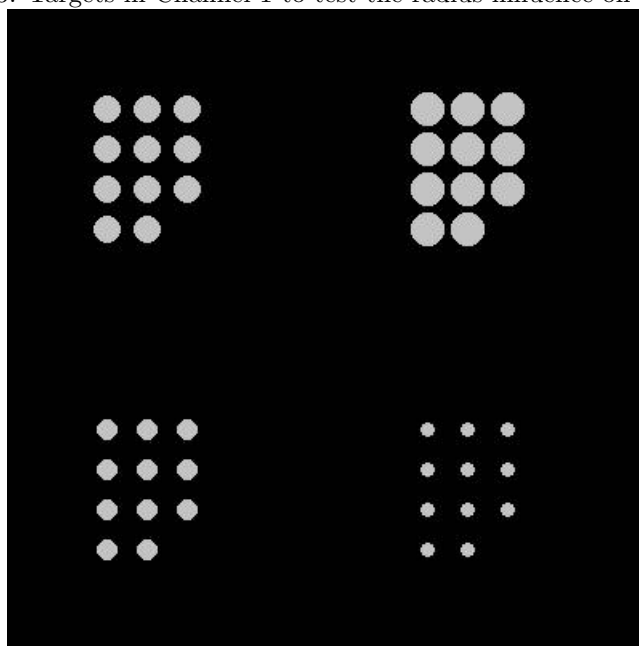


Figure D.6: Targets in Channel 2 to test the radius influence on MS Results.

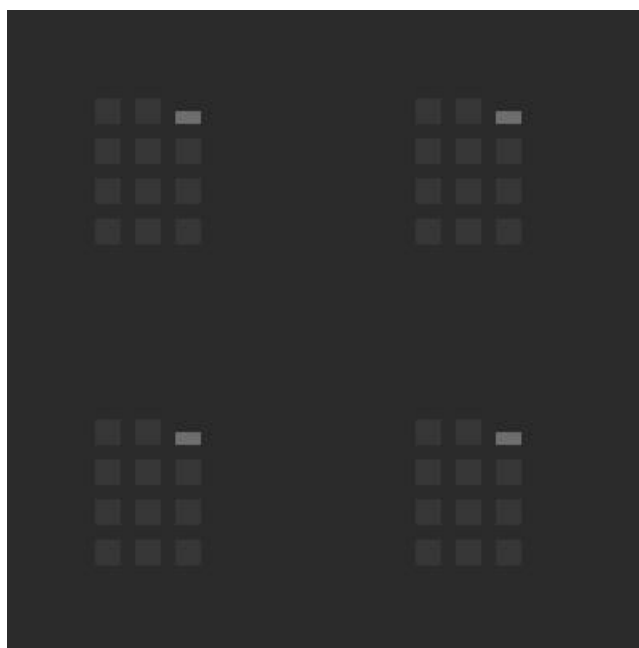


Figure D.7: Targets in Channel 1 to test the square shape influence on MS Results.

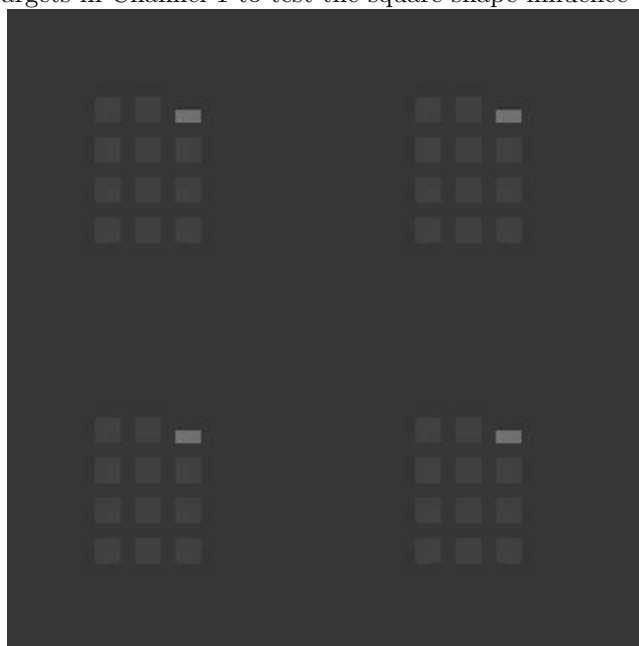


Figure D.8: Targets in Channel 2 to test the square shape influence on MS Results.

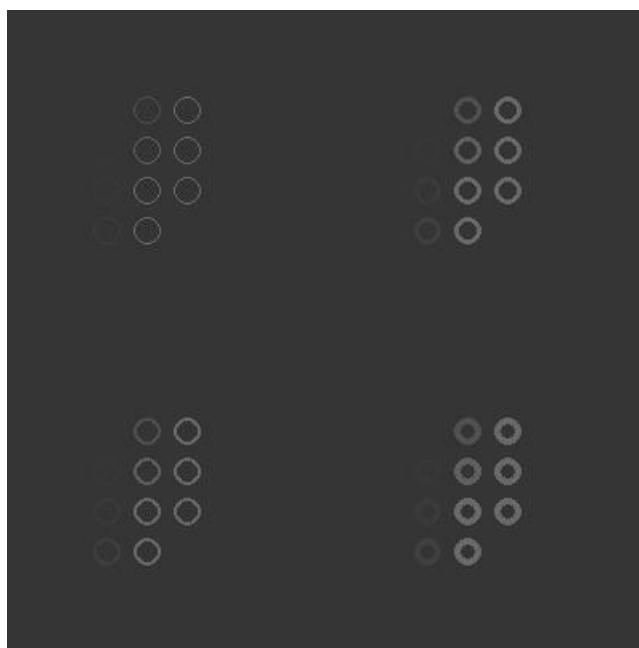


Figure D.9: Targets in Channel 1 to test the square shape influence on MS Results.

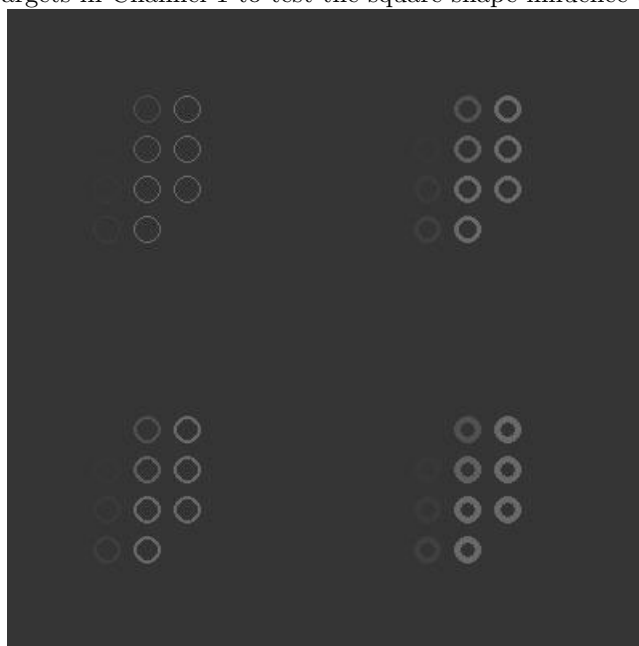


Figure D.10: Targets in Channel 2 to test the square shape influence on MS Results.

Index

- acid
 - deoxyribonucleic, 8
 - nucleic, 8
 - ribonucleic, 8
- addressing, 4
- adjustment, 20, 47
- aliquot, 11
- Analysis of Variance, 2
- array, 15
- artifacts, 18

- background
 - pixel, 45
 - mask, 45
- background
 - noise, 18
- bleeding effect, 50

- chromosomes, 8
- cluster analysis, 2
- convolution, 25
- correlation, 48

- DNA
 - complementary, 9
 - genomic, 9

- exponential noise, 32
- expressed sequence tags, 11
- expression level, 1

- Fourier Transform
 - Discrete, 14
 - discrete, 19

- gene, 1, 8
- gene expression, 73
- gene expression level, 13, 63
- gradient image, 25
- grid template, 20
- gridding, 3, 4, 15

- Hough Transform
 - circular, 14, 25
 - linear, 25
- hybridization, 8

- intensity extraction, 4

- Kolmogorov-Smirnov, 48

- label, 49
- labelling, 52
- least-square minimum, 48
- linear regression, 48

- Mann-Whitney
 - statistic, 61
 - test, 14, 29, 46
 - non-parametric, 62
 - randomness, 71
- Mann-Whitney
 - test, 5
 - Wilcoxon two sample test, 61
- messenger RNAs, 8
- microarray, 1
- microarray experiment, 11

- normalization, 2
- nucleotides, 8
- null hypotheses, 61
 - rejection, 62

- offset, 20
- operator, 25
- order independent, 49

- PCR Amplification, 1
- peaks, 20
- period, 20
- pin-array, 15
 - column-count , 15
 - row-count , 15
- pixel, 15
 - boundary, 49
- polymerase chain reaction, 10
- preprocessing, 14
- primer, 10
- printing layout, 18

- program
 - GenePix, 46
 - MicroArray Suite, 46
 - IPLab, 72
 - QuantArray, 47
 - ScanAlyze, 46
 - Spot, 47
- promoter, 9
- protein, 7, 8

- quantification, 11

- ratio
 - uncorrected mean, 47
- regulation, 10
- replication, 8
- residue, 7
- reverse transcription, 9
- ribosome, 9

- saturation, 32
- seed, 5
 - background, 50, 51
 - cross, 50
 - target, 50
- seed choice, 59
 - centered maximum, 58
 - maximum mean, 58
 - random, 59
 - union, 60
- Seeded Region Growing algorithm, 5
- segmentation, 4, 44
 - adaptive circle, 46

- adaptive shape, 46
- fixed circle, 4, 46, 47
- histogram, 46
- software
 - Espresso, 1
- source of variation, 2
- spacing, 15, 18
 - of targets, 20
- strand
 - complementary, 8
 - template, 8
- target, 15
 - background, 63
 - beignet, 51, 52
 - boundary, 45
 - doughnut, 60
 - mask, 45, 47, 62
 - patch, 15
 - pixel, 45
 - site, 45, 63
- thresholding, 33
- tilting effect, 20
- transcription, 8
 - reverse, 8
- translation, 8, 9
- Wilcoxon
 - rank sum, 61
 - Wilcox multiple comparison test, 71

VITA

Vincent Jouenne was born on August 10, 1976 in Poissy, France. He received a DEUG in Applied Mathematics and Informatics to Sciences from the University of Versailles-St Quentin, France in 1997. From September 1997 to June 1999, he was an undergraduate student at the University of Technology in Compiègne (UTC), France. He began graduate studies at Virginia Tech in August 1999. The work reported in this thesis was completed between September 2000 and June 2001. His research interests include bioinformatics, image processing and Human Computer Interaction (Peripheral Displays, PDAs). He was an application developer at Parexel GmbH, Frankfurt, Germany from September 1998 to February 1999 and a bioinformatician at Novartis NABRI in the Research Triangle Park, NC, USA from June 2000 to August 2000. He was a teaching assistant at Virginia Tech from August 1999 to May 2001.