WAVELET BASED ARRAY COMPARATIVE GENOMIC HYBRIDIZATION AND MASS SPECTROMETRY DATA ANALYSIS

by

NHA NGUYEN

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2010

Copyright © by Nha Nguyen 2010

All Rights Reserved

To my parents Nguyen Thi Lat & Nguyen Thanh Long

and my wife An Vo

who encourage and inspire me to pursue doctoral study.

ACKNOWLEDGEMENTS

I would like to thank my supervising professors Dr.Heng Huang and Dr. Soontorn Oraintara for constantly motivating and encouraging me, and also for their invaluable advice during the course of my doctoral studies. I wish to thank my committee members, Dr. K. R. Rao, Dr. Michael Manry and Dr. Qilian Liang for their interest in my research and for taking time to serve in my dissertation committee.

I am grateful to all the teachers who taught me during the years I spent in school, first in Vietnam and then in the Unites States. To all of the Genomics and MSP lab alumni and current members with whom I shared long discussions and many late night working hours, I would like to give my sincere thanks. Finally, I would like to dedicate this work to my grandparents, parents, aunts and wife for their sacrifice, encouragement and patience. I also thank several of my friends who have helped me throughout my career.

June 30, 2010

ABSTRACT

WAVELET BASED ARRAY COMPARATIVE GENOMIC HYBRIDIZATION AND MASS SPECTROMETRY DATA ANALYSIS

Nha Nguyen, Ph.D.

The University of Texas at Arlington, 2010

Supervising Professor: Soontorn Oraintara and Heng Huang

As a highly efficient technique, array-based comparative genomic hybridization (aCGH) methods allow the simultaneous measurement of genomic DNA copy number at hundreds or thousands of loci and the reliable detection of local one-copy-level variations. The identification of these DNA copy number changes provides insights to facilitate both the basic understanding of cancer and its diagnosis. In order to effectively analyze aCGH data, various techniques have been proposed to help researchers smooth the DNA copy number data and subsequently to quantify the alterations. In this dissertation, many wavelet based methods are proposed to smooth and segment the aCGH data that is the key step to detect DNA copy number alterations. The proposed smooth methods are combinations of shift-invariant wavelet transforms (such as dual tree complex wavelet transform and stationary wavelet packet transform) and bivariate shrinkage estimators. The proposed segmentation method includes two main steps such as heavy-tailed noise suppression and derivative wavelet scalogram based segmentation. The proposed method is performed on both synthetic and real

datasets. The experimental results show that proposed method outperforms the previous approaches.

Mass Spectrometry (MS) is increasingly being used to discover diseases related proteomic patterns. The smooth and peak detection steps are important steps in the typical analysis of MS data. Recently, many new algorithms have been proposed to increase true position rate with low false discovery rate in peak detection. In this dissertation, two peak detection methods are proposed. The first proposed method is GaborEnvelop method which is a combination of Gabor filters and envelope analysis. The second proposed method is GDWavelet method which is used to process mass spectrometry based on Gaussian derivative wavelet. Both the proposed methods can detect more true peaks with a lower false discovery rate than previous methods. The proposed methods have been performed on the real SELDI-TOF spectrum with known polypeptide positions and on two synthetic data with Gaussian and real noise. The experimental results demonstrate the proposed methods outperform other common used methods in the Receiver Operating Characteristic (ROC) curve.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS iv

ABSTRACT v

LIST OF FIGURES xi

LIST OF TABLES xiv

Chapter Page

1. INTRODUCTION 1

1.1 Array-based Comparative Genomic Hybridization Data Processing . . 1

1.2 Mass Spectrometry Data Processing 2

2. WAVELET BASED ARRAY COMPARATIVE GENOMIC HYBRIDIZATION DATA SMOOTHING 4

2.1 Array Comparative Genomic Hybridization 4

2.1.1 Introduction 4

2.1.2 Artificial Chromosome Generation 6

2.1.3 New Synthetic Data Model 7

2.2 Wavelet Transforms 9

2.3 Previous Works 13

2.4 DTCWTi and DTCWTi-bi Algorithms 15

2.4.1 Proposed Methods 15

2.4.2 Performance Evaluation by RMSE 17

2.5 Improved Bivariate Shrinkage Model for SWPT 18

2.6 SWPT-LaBi Algorithm 32

2.6.1 Signal Extension 32

2.6.2 Proposed Method 32

2.6.3 Performance Evaluation by RMSE . . . 33
2.6.4 Performance Evaluation by ROC curves . . . 35
2.7 SWPT-AdaBi Algorithm . . . 37
2.7.1 Proposed Method . . . 37
2.7.2 Comparisons of Extension Methods . . . 38
2.7.3 Experiments Design . . . 39
2.7.4 Performance Evaluation by RMSE . . . 40
2.7.5 Performance Evaluation by ROC Curve . . . 42
2.8 Conclusion . . . 43
3. HEAVY-TAILED NOISE SUPPRESSION AND DERIVATIVE WAVELET SCALOGRAM BASED SEGMENTATION OF ARRAY-CGH DATA . . . 44
3.1 Introduction . . . 44
3.2 Array CGH Noise Characteristic . . . 46
3.2.1 Data Description . . . 46
3.2.2 Distribution Noise Candidates in Array CGH . . . 48
3.2.3 Validation of New Array CGH Data Noise Model . . . 51
3.3 Proposed Methods . . . 52
3.3.1 Heavy-Tailed Noise Suppression . . . 54
3.3.2 One-Directional Derivative Wavelet Scalogram . . . 59
3.3.3 Derivative Wavelet Scalogram based Segmentation Method . . . 61
3.4 Results . . . 63
3.4.1 Improved Synthetic Data Model . . . 63
3.4.2 Performance Evaluation of DWSS Method . . . 66
3.5 Discussion . . . 69
3.6 Conclusion . . . 72
4. GABOR FILTERS AND ENVELOPE ANALYSIS BASED MASS SPEC-

TROMETRY PEAK DETECTION 77
4.1 Introduction 77
4.2 Complex Gabor Filters 79
4.3 Envelope Analysis 80
4.4 GaborLocal and GaborEnvelop Methods 85
4.4.1 Full Frequency MS Signal Seneration 86
4.4.2 Peak Detection and Peak Quantification in GaborLocal . . . 92
4.4.3 Peak Detection and Peak Quantification in GaborEnvelop . . 93
4.4.4 Intersection 97
4.5 Experiments and Discussions 97
4.5.1 Cromwell Method 97
4.5.2 CWT Method 98
4.5.3 Evaluation Using ROC Curve 99
4.6 Conclusion 102
5. GAUSSIAN DERIVATIVE WAVELET BASED MASS SPECTROMETRY DATA PROCESSING 103
5.1 Introduction 103
5.2 Smoothing by Bivariate Shrinkage Function 105
5.3 Peak Detection by Gaussian Derivative Wavelet 105
5.3.1 Theory of Zero-Crossing Lines in Multi-Scale 106
5.3.2 Application of Zero-Crossing to Peak Detection 108
5.4 Saving Small Energy Peaks by Envelope Analysis 114
5.5 Gaussian Derivative Wavelet based Method (GDWavelet) 114
5.6 Experiments and Discussions 117
5.6.1 Experimental Setup 117
5.6.2 Experimental Results 119

5.7 Conclusion 120
6. CONCLUSION 122
6.1 Array-CGH 122
6.2 Mass Spectrometry 123
Appendix
A. ABBREVIATION LIST 125
REFERENCES 128
BIOGRAPHICAL STATEMENT 136

LIST OF FIGURES

Figure Page

2.1 Normalized distribution of real noise 8

2.2 The 3 level DWT filter bank structure 9

2.3 The 3 level DWPT filter bank structure 10

2.4 The 3 level DTCWT filter bank structure. 11

2.5 The 3 level SWT filter bank structure 12

2.6 The 3 level SWPT filter bank structure 14

2.7 The positions of child, parent and cousin coefficients. 19

2.8 The histograms computed from true aCGH signal. 19

2.9 The joint distribution of w_1 and w_3 created from decomposition of true aCGH signal 20

2.10 The proposed pdf with two variables: w_1 and w_3 20

2.11 The joint distribution of w_1 and w_3 created from decomposition of true aCGH signal 21

2.12 The proposed pdf with two variables w_1 and w_3 22

2.13 The distribution of the adaptive parameter K 23

2.14 Joint distribution of w_1 and w_3 28

2.15 (a) The Laplacian pdf with two variables: w_1 and w_3, (b) The proposed pdf with two variables: w_1 and w_3 29

2.16 Three extension methods 31

2.17 The flowchart of SWPT-LaBi method 32

2.18 Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with each of 7 noise levels (Gaussian noise) 34

2.19 Comparison of average RMSEs obtained from the 1,000 artificial chro-

mosomes with real noise 35

2.20 Receiver operating characteristic (ROC) curves obtained from 270 artificial chromosomes 36

2.21 Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with each of 7 noise levels using SWPT-Adabi with three extension methods in the preprocessing step 39

2.22 Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with each of 7 noise levels using the proposed method SWPT-AdaBi and some methods in time domain such as Loess, Lowess, Moving Average, Quantreg, Smoothseg 40

2.23 Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with each of the 7 noise levels using the proposed method SWPT-AdaBi and some previous methods using wavelet transform such as SWTi, DTCWTi, DTCWTi-bi 41

2.24 Receiver operating characteristic (ROC) curves obtained from 280 artificial chromosomes with each of the different noise levels using SWPT-AdaBi and some aCGH algorithms 42

3.1 Examples of array CGH and their empirical histograms 47

3.2 Four probability density candidates with zero mean 50

3.3 The relative entropy between the histogram of the chromosome 15 of GSM232967 and four distribution candidates 53

3.4 The relative entropy between the histogram of the chromosome 13 of GSM215042 and four distribution candidates 54

3.5 The position of child and parent coefficients 55

3.6 The histograms computed from true array CGH signals 56

3.7 The joint distribution of w_1 and w_2 57

3.8 The proposed pdf with two variables: w_1 and w_2 57

3.9 One example of DWSS method 62

3.10 A procedure to create real noise 65

3.11 Summary result 66

3.12 One hundred real noise simulated samples 69

3.13 One hundred simulated samples 73
3.14 Real noise simulated data without bias 75
3.15 Real noise simulated data with bias 76
4.1 Frequency supports of complex Gabor filters 79
4.2 General flowchart of envelope analysis 83
4.3 Proposed envelope analysis for peak detection 85
4.4 Flowchart of GaborLocal method in the MS peak detection 86
4.5 Flowchart of GaborEnvelop method in the MS peak detection 87
4.6 The frequency response of the summary filter 89
4.7 One example of the full frequency MS generation step 90
4.8 One example of GaborLocal in peak detection 94
4.9 One example of envelope analysis 95
4.10 One example of GaborEnvelop at intersection step 96
4.11 Detailed ROC curves obtained from 60 MS signals 99
4.12 Average receiver operating characteristic (ROC) curves obtained from 60 MS signals 100
5.1 An illustration of zero-crossing line and ridge line comparison 109
5.2 GDWavelet Method's Flowchart 114
5.3 An Illustration of GDWavelet 116
5.4 ROC Curves - Simulated data with Gaussian noise 120
5.5 ROC Curves - Simulated data with real noise and real data 120

LIST OF TABLES

Table		Page
2.1	Comparison of average RMSEs obtained from the 1,000 artificial chromosomes	18
2.2	Comparison table of the improved bivariate shrinkage function and function in [1]	31
3.1	Five datasets	47
3.2	Average $\Delta H/H$ of five distributions	52
3.3	Estimated values for the parameters α, β of real array CGH noises	64
3.4	Comparison of average RMSEs obtained from six simulated data-sets with 1,000 arrays each will be shown in this table	74
3.5	Comparison of average RMSEs obtained from three real noise simulated data-sets with 1,000 arrays each	74
3.6	Comparison of average RMSEs obtained from Lee 2008 array including 40 samples, Smith2007 array including 69 samples and Nicolas2009 array including 23 samples	74
4.1	The values of vector $v(k)$ with different lengths.	81
4.2	Definition of peak rank in GaborLocal	92
4.3	Definition of peak rank (PR) in GaborEnvelop	95
5.1	Error of Peak Position Estimation	112
5.2	Error of Peak 's Standard Deviation Estimation	113
5.3	Error of Peak 's Height Estimation	113

CHAPTER 1

INTRODUCTION

1.1 Array-based Comparative Genomic Hybridization Data Processing

As a highly efficient technique, array-based comparative genomic hybridization (aCGH) methods allow the simultaneous measurement of genomic DNA copy number at hundreds or thousands of loci and the reliable detection of local one-copy-level variations. The identification of these DNA copy number changes provides insights to facilitate both the basic understanding of cancer and its diagnosis. In order to effectively analyze aCGH data, various techniques have been proposed to help researchers smooth the DNA copy number data and subsequently to quantify the alterations. In chapter 2, with Gaussian noise assumption, a framework using the stationary wavelet packet transform with new adaptive bivariate shrinkage functions is proposed to smooth the aCGH data that is the key step to detect DNA copy number alterations.

Current array comparative genomic hybridization (array CGH) data analysis methods and evaluation data model assumed that probability density function (PDF) of noise in the array CGH data is a Gaussian distribution. However, in practice this noise distribution is peaky and heavy-tailed. Therefore a Gaussian PDF is not adequate to behaviors of noise in the array CGH data and can introduce wrong detections of chromosomal aberrations and lead misunderstanding on disease pathogenesis. A more accurate and sufficient model of noise in the array CGH data is necessary and beneficial to detection of DNA copy number variations. In chapter 3, first, the real array CGH data in many platforms is analyzed. Distribution of noise in the array

CGH data is fitted very well by generalized Gaussian distribution (GGD). Next, a novel array CGH analysis method combining the advantages of both smoothing and segmentation approaches is proposed. The proposed method, DWSS, uses generalized Gaussian bivariate shrinkage function and one-direction derivative wavelet scalogram in generalized Gaussian noise. In smoothing step, with the new generalized Gaussian noise model, the heavy-tailed noise suppression algorithm is derived in stationary wavelet domain. In segmentation step, the 1-D Gaussian derivative wavelet scalogram is applied to suppress heavy-tailed noise and obtain the final segmentations.

Many simulated array CGH data with different noises (such as Gaussian noise, GGD noise and real noise) and many real array CGH data are used in experiments. New fast method performs better than other most commonly used methods, in terms of both Root Mean Squared Errors (RMSEs) and Receiver Operating Characteristic (ROC) curves.

1.2 Mass Spectrometry Data Processing

Mass Spectrometry (MS) is increasingly being used to discover diseases related proteomic patterns. Peaks are the key information in Mass Spectrometry (MS) which has been increasingly used to discover diseases related proteomic patterns and improve biological studies. Peak detection is an essential step for MS based proteomic data analysis. Recently, several peak detection algorithms have been proposed with good performance. However, in these algorithms, there are three major deficiencies: 1) noise is removed as much as possible, but the true signal could also be removed; 2) baseline removal step may get rid of true peaks and create new false peaks; 3) in peak quantification step, a threshold of signal-to-noise ratio (SNR) is usually used to remove false peaks. However, noise estimations in SNR calculation are often inaccu-

rate in time or wavelet domain. In this dissertation, serveral proposed MS processing methods will be introduced in chapter 4 and chapter 5.

In chapter 4, two novel methods (GaborLocal and GaborEnvelop) are proposed. Both of them can detect more true peaks with a lower false discovery rate than previous methods. The Gaussian local maxima is employed to detect peaks, because it is robust to noise in signals. A new approach, peak rank, is defined at the first time to identify peaks instead of using the signal-to-noise ratio. Meantime the Gabor filter is used to amplify important information and compress noise in the raw MS signal. Moreover, the envelope analysis is also proposed to improve the quantification of peaks and remove more false peaks.

In chapter 5, new algorithm is proposed to solve these problems and improve MS peak detection. First, a bivariate shrinkage estimator is used in stationary wavelet domain to avoid removing true peaks in denoising step. Second, without baseline removal, zero-crossing lines in multi-scale of derivative Gaussian wavelets are investigated with using mixture of Gaussian to estimate discriminative parameters of peaks. Third, in quantification step, a novel approach using frequency, standard deviation, height, and rank of peaks is used to detect both high and small energy peaks with robustness to noise. A novel Gaussian Derivative Wavelet (GDWavelet) method is proposed to more accurately detect true peaks with a lower false discovery rate than existing methods.

The proposed methods have been performed on the real Surface-Enhanced Laser Desorption/Ionization Time-Of-Flight (SELDI-TOF) spectrum with known polypeptide positions and on two synthetic data with Gaussian and real noise. All experimental results demonstrate the proposed method outperforms other commonly used methods. The standard receiver operating characteristic (ROC) curves are used to evaluate the experimental results.

CHAPTER 2
WAVELET BASED ARRAY COMPARATIVE GENOMIC HYBRIDIZATION DATA SMOOTHING

2.1 Array Comparative Genomic Hybridization

2.1.1 Introduction

The enormous recent progress has been achieved in understanding cancer at a molecular level, but the precise details are still elusive for many types of carcinomas. Recently the lack of large-scale genome-wide mutation or DNA copy number data has been addressed by SNP arrays and array Comparative Genomic Hybridization (aCGH) [2] that reveals important molecular features of human genetics and disease. The research achievement in the genome-wide identification and localization of genetic alterations by high resolution aCGH technologies have furthered author's understanding of gene mutations, oncogene amplifications, or deletions of tumor-suppressor genes that could be very helpful in cancer understanding and diagnosis. Bacterial Artificial Chromosomes (BAC) based CGH arrays were amongst the first genomic arrays to be introduced [3] and are routinely used to detect single copy changes in the genome, owing to their high resolution in the order of 1 Mb [3, 4]. More recently Oligonucleotide aCGH [5, 6] was developed to allow flexibility in probe design, greater coverage, and much higher resolution in the order of 35-100 Kb.

Because aCGH is very noisy, many diseases related chromosomal aberrations are buried by noise. For example, in cDNA array CGH data, the signal to noise ratio is often approximately 1 (0 dB) [7]. In order to develop effective methods to identify aberration regions from array CGH data, many research works focus on both smoothing/denosing-based and segmentation-based data processing. Segmentation-

based methods target to model data as a series of discrete segments with unknown boundaries and unknown heights. Since the boundary points are highly possible to be identified as aberration region, the false positives are introduced. Smoothing-based methods reduce noise by comparing each data point to its adjacent ones and reduce the number of identified false aberration regions.

In previous research, many approaches have been proposed to smooth/denoise aCGH data. More recently, wavelet transform is considered as one of the best tools to remove noise from the aCGH data. Lai (2005) [8] compared 11 different algorithms for analyzing array CGH data. Lai concluded that the Wavelet [9], Quantreg [10] and Lowess method [11] give better detection results (higher TPR and lower FPR) than the others. The MODWT is also stationary wavelet transform (SWT). Besides the MODWT, there are many different kinds of wavelet transforms which can capture more information from the aCGH data and can be more useful for aCGH smoothing. Therefore, the wavelet based smoothing is considered as the promising approach.

In this chapter, shift invariant SWPT with two novel adaptive bivariate shrinkage estimators (called SWPT-AdaBi) is proposed to use for aCGH data smoothing. In the SWPT, all sub-bands are shift invariant and each sub-band provides a shiftable description of signal in a specific scale as same as the SWT or the MODWT. Such shift invariant property is crucial to apply wavelet based method into aCGH data smoothing. Although the Discrete Wavelet Transform (DWT) with the redundant ratio of $1:1$ is efficient for computation, it is not suitable for aCGH smoothing application. Because DWT creates artifacts around the discontinuities of the input signal [12] and is shift-variant. Because the SWPT also decomposes signal to many uniform frequency sub-bands, information in both of low and high frequency sub-bands are captured. However the previous wavelet based methods lose the information in high frequency [9, 13, 14].

Moreover, three improved bivariate shrinkage functions are proposed to exploit the dependency between child and cousin coefficients in SWPT to improve the performance. The performance of proposed approach is validated through theoretical and experimental explorations of a set of aCGH data. The performance between proposed method and previous methods is compared by two standard performance evaluation criterions: root mean squared error (RMSE) and receiver operating characteristic (ROC) curves. The experimental results show that new methods outperform the previous approaches consistently on both synthetic and real data.

2.1.2 Artificial Chromosome Generation

A wide variety of methods have been proposed for pre-processing aCGH data. Not surprisingly, it can be difficult to determine which methods are better than the others. Simulated aCGH data will be used to overcome that problem. Willenbrock and Fridlyand [15] proposed a simulation model to create the synthetic array CGH data. In their model, a primary tumor dataset of 145 samples is segmented and the probes are equally spaced along the chromosome. Actually, real aCGH data has randomly space between two probes. More recently Y. Wang and S. Wang [13] extended this model by placing unequally spaced probes along chromosome. The primary tumor data set is segmented using DNAcopy number levels from the empirical distribution of segment mean values smv as

$$c = \begin{cases} 0 \quad (0 \text{ copies}) & : smv < -0.4, \\ 1 \quad (\text{one copy}) & : -0.4 < smv < -0.2, \\ 2 \quad (\text{two copies}) & : -0.2 < smv < 0.2, \\ 3 \quad (\text{three copies}) & : 0.2 < smv < 0.4, \\ 4 \quad (\text{four copies}) & : 0.4 < smv < 0.6, \\ 5 \quad (\text{five copies}) & : smv > 0.6. \end{cases}$$

The synthetic DNA copy number data on a chromosome was generated as follows

1. Determine copy number probability and the distribution of segment length. As suggested in [15] and [13], the chromosomal segments with DNA copy number $c = 0, 1, 2, 3, 4$ and 5 are generated with probability $0.01, 0.08, 0.81, 0.07, 0.02$ and 0.01. The lengths for segments are picked up randomly from the corresponding empirical length distribution given in [15].

2. Compute $log_2 ratio$. Each sample is a mixture of tumor cells and normal cells. A proportion of tumor cells is P_t, whose value is from a uniform distribution between 0.3 and 0.7. As in [15], the $log2ratio$ is calculated by

$$log_2 ratio = \log_2 \left(\frac{cP_t + 2(1 - P_t)}{2} \right), \tag{2.1}$$

where c is the assigned copy number. The expected $log_2 ratio$ value is then the latent true signal.

3. Add Gaussian noises. Gaussian noises with zero mean and variance σ_n^2 are added to the latent true signal. Till now, the equally spaced aCGH signal is obtained.

4. Create unequally spaced probes. Because the distances between probe i and probe $i + 1$ are randomly, the best way to get these distances is from the UCSF HumArray2 BAC array. Thus, a real aCGH signal is created from the equally spaced aCGH signal when the unequally spaced probes are placed on the chromosome. Now, many artificial chromosomes of length 200 Mbase are created by many noise levels $\sigma_n = 0.1, 0.125, 0.15, 0.175$, 0.2, 0.225 and 0.25.

2.1.3 New Synthetic Data Model

In new synthetic data model, the four above steps should be followed but in the third step, the real noise should be added instead of Gaussian noise. There are many aCGH data source such as [16], [17], [18], but only data from [18] can be used to get

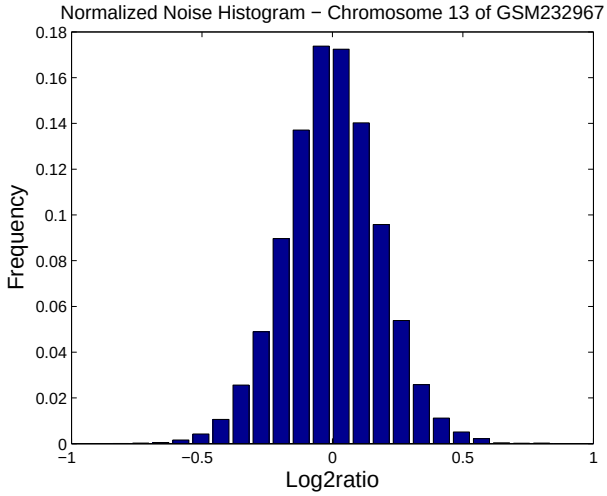


Figure 2.1. Normalized distribution of real noise from chromosome 13 of GSM232967.

real noise. The number of probes in [16] and [17] are not enough. Data from [16] has hundreds of probes and data from [17] has about several thousand probes. Both of them have not enough probes to estimate the correct distribution of noise. However, the length of data from [18] is long enough (more than ten thousands of probes). For example, from [18], chromosome 13 of GSM232967 has 18323 probes. If 64 bins are used, the distributions of noise from the above chromosomes are shown in Fig. 2.1. Now, it is easy to create arrays with random values under the above distributions. These arrays are added into true signal to create simulated data with real noise. During this step, chromosomes that only have the copy two (zero means) are selected randomly. There are many chromosomes which can be used to extract real noise model, *e.g.* chromosome $1, 3, 4, 6, 8, 9, 10, 12, 13, 14, 17, 18, 19, 20$ of GSM232967 and chromosome 18 of GSM232968.

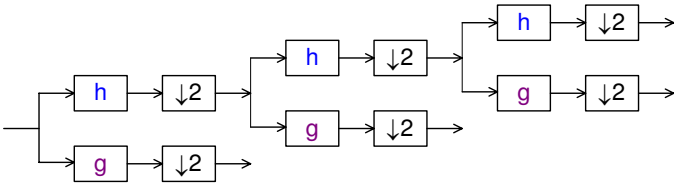


Figure 2.2. The 3 level DWT filter bank structure.

2.2 Wavelet Transforms

Four wavelet transforms including discrete wavelet transform, discrete wavelet packet transform, stationary wavelet transform and stationary wavelet packet transform will be introduced briefly.

1. *Discrete Wavelet Transform:* The discrete wavelet transform (DWT), showed in Fig. 2.2, based on the octave band tree structure, can be viewed as the multiresolution decomposition of a signal. It takes a length N sequence, and generates an output sequence of length N using a set of lowpass and highpass fiters followed by a decimator. It has $N/2$ values at the highest resolution, $N/4$ values at the next resolution, and $N/2^L$ at the level L. Because of decimation, the DWT is a critically sampled decomposition. However, the drawback of DWT is the shift variant property. In signal denoising, the DWT creates artifacts around the discontinuities of the input signal [12]. These artifacts degrade the performance of the threshold-based denoising algorithm. From the structure in Fig. 2.2, the DWT has non-uniform frequency supports.
2. *Discrete Wavelet Packet Transform:* Similarly to the DWT, the discrete wavelet packet transform (DWPT) is a critically sampled decomposition. However, it has uniform supports shown in Fig. 2.3. All of DWPT scales are performed at the same level j. The jth level DWPT decomposes the frequency interval $[0, 1/2]$ into 2^j equal and individual intervals, each of which has $N/2^j$ values if

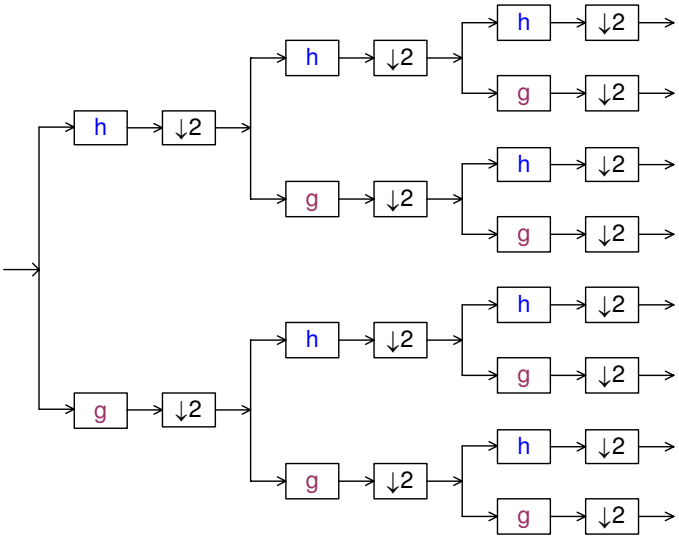


Figure 2.3. The 3 level DWPT filter bank structure.

taking a length N sequence. The drawback of the DWPT is the shift variance as the DWT.

3. Dual-Tree Complex Wavelet Transform A dual-tree structure that produces a dyadic complex DWT is proposed by Kingsbury [19, 20]. Since array CGH data are one dimensional signals, in this chapter, only the 1-D case of dual-tree CWT is mentioned. The DTCWT filter bank structure is shown in Fig. 2.4. The analysis FB for the DTCWT is an iterative multiscale FB. Each resolution level consists of a pair of two-channel FBs. The input signal is passed through the first level of a multiresolution FB. The low frequency component, after decimation by 2, is fed into the second level decomposition for the second resolution. The outputs of the two trees are the real and imaginary parts of complex-valued subbands. To reconstruct the signal, the real part and imaginary part are inverted to obtain two real signals, respectively. These two real signals are then averaged to obtained the final output. For more details of the construction of the dual-tree, the reader is referred to [21].

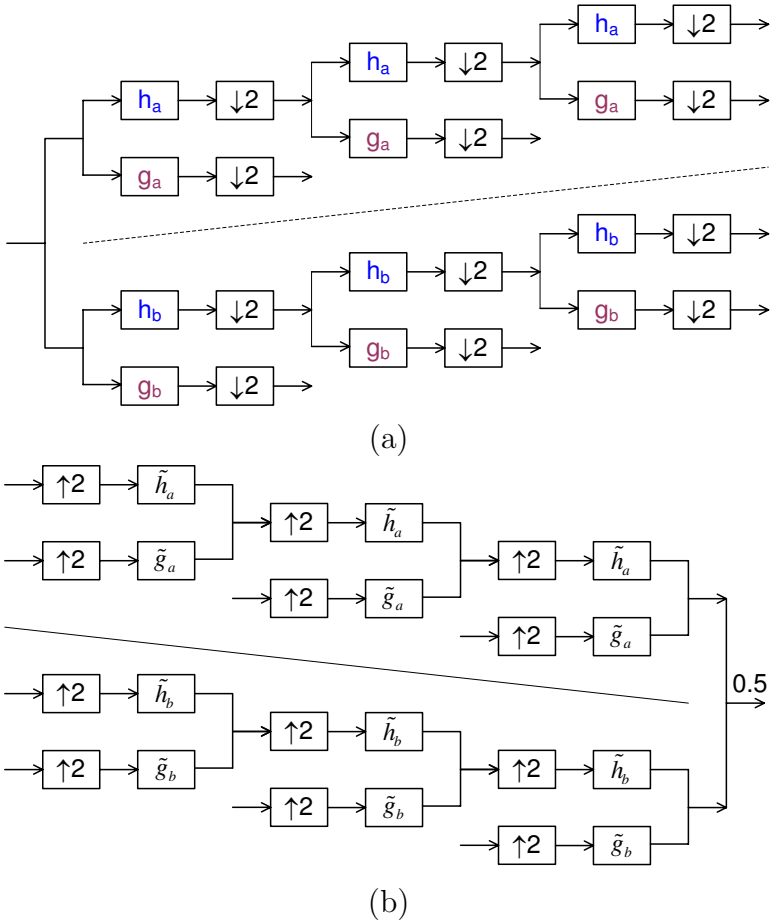


Figure 2.4. The 3 level DTCWT filter bank structure. (a) Analysis FB, (b) Synthesis FB.

The most important property of the DTCWT is that all complex subbands are shift invariant in the sense that there is no significant aliasing in the decimated complex subbands. Therefore, each complex subband provides a shiftable description of signal in a specific scale. By construction of the dual-tree CWT, each pair of corresponding filters has the Hilbert transform relation [21]. It is therefore an overcomplete representation with a redundant ratio of $2:1$. In the two trees, the filters are designed in such a way that the aliasing in one branch in the first tree is approximately canceled by the corresponding branch in the second tree. The relation between the wavelet filters of the two trees yields shift

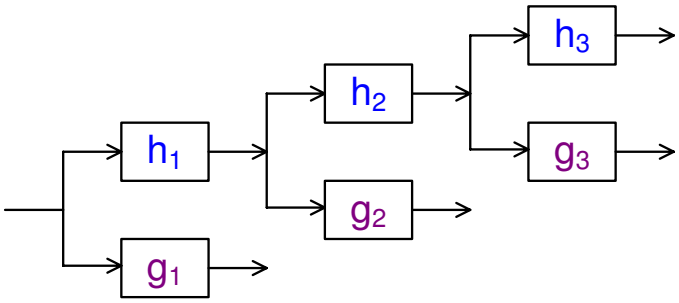


Figure 2.5. The 3 level SWT filter bank structure.

invariant property [19]. The equivalent complex filter for each subband has one-sided frequency support. The real part of the complex filter is symmetric while the imaginary part is anti-symmetric.

4. *Stationary Wavelet Transform:* The stationary wavelet transform (SWT) [12], showed in Fig. 2.5, is similar to the DWT except that it does not employ a decimator after filtering, and each level's filters are up-sampled versions of the previous ones. The SWT is known as the shift invariant DWT. The absence of a decimator leads to a full rate decomposition. Each sub-band contains the same number of samples as the input. So for a decomposition of L levels, there is a redundant ratio of $(L+1) : 1$. However, the shift invariant property of the SWT makes it preferable for the usage in various signal processing applications such as denoising and classification because it relies heavily on spatial information. It has been shown that many of the artifacts could be suppressed by a redundant representation of the signal [12]. The SWT has the same non-uniform sub-bands as the the DWT.

5. *Stationary Wavelet Packet Transform:*

   Stationary wavelet packet transform (SWPT), showed in Fig. 2.6, is a generalization of stationary wavelet decomposition (SWT). First, a signal is decomposed into a low frequency subband and a high frequency subband by using

two channel filter bank. Similar to the SWT, the SWPT does not employ a decimator after filtering. Then the low frequency subband as well as the high frequency subband can be decomposed into a second-level low and high frequency subband, and the process is repeated as in Fig. 2.6. Each level's filter are upsampled versions of the previous ones. The absence of a decimator leads to a full rate decomposition. Each subband contains the same number of samples as the input. So for a decomposition of L levels, there is a redundant ratio of $2^L : 1$. However, the absence of a decimator makes the SWPT shift invariant. In the SWT, the low frequency subband is itself decomposed into two second-level subbands. Therefore, the SWT has nonuniform frequency supports while the SWPT has uniform frequency supports. So, the SWPT offers a richer range of possibilities for signal analysis. With the uniform shift-invariant subbands, the SWPT may capture more information from the aCGH data. So, the SWPT is proposed for denoising of aCGH data.

2.3 Previous Works

- Loess: The locally weighted scatter plot smooth using least squares quadratic polynomial fitting has been used in previous work [11].
- Lowess: This is the locally weighted scatter plot smooth using least squares linear polynomial fitting. It uses a first-degree polynomial instead of second-degree polynomial in Loess. This method was compared to other methods in [8].
- Quantreg: This is a quantile regression method which has been proposed by Eilers in [10]. The total variation was used as the roughness penalty. In 2007, Li [22] modified this method by incorporating the physical distance between adjacent clones.

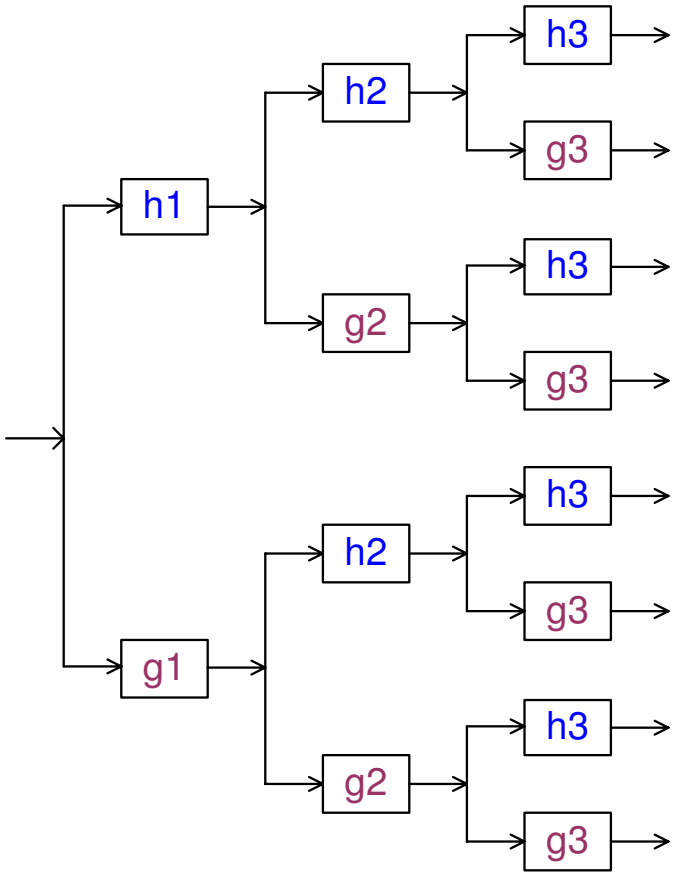


Figure 2.6. The 3 level SWPT filter bank structure.

- Smoothseg: A smooth segmentation method [23] for aCGH data analysis is based on a doubly heavy-tail-random-effect model. This heavy-tailed model on error term deals with outliers in observations. To deal with possible jumps in the copy-number pattern, the i.i.d Cauchy distribution is proposed for modeling the second-order differences of original data. The denoised data is estimated by the iterative weighted least-squares algorithm.
- Moving-Average: This is exactly a low pass filter which takes the average of neighboring data points. In this case, sliding window spanning 30 probes is used.

- SWTi: SWTi method comes from paper [13]. Compared with SWPT-bi, SWTi method has three different steps: 1) the aCGH data which has the unequal distances between two samples is interpolated to reduce the difference of those distances; 2) the array CGH signal is decomposed by the SWT; 3) the term by term thresholding is applied to estimate the SWT coefficients [13].

2.4 DTCWTi and DTCWTi-bi Algorithms

2.4.1 Proposed Methods

The complex coefficients W_i are obtained by decomposing the data Y with the DTCWT. All complex coefficients whose magnitudes are less than a particular threshold are set to zero as follows

$$W_i = \begin{cases} 0 & \text{if } |W_i| \leq \delta_U, \\ W_i & \text{if } |W_i| > \delta_U. \end{cases} \tag{2.2}$$

After that, the subband coefficients denoised are used to reconstruct the data $\hat{D}$. Next, how to choose the threshold values is discussed. The universal threshold is defined in [24, 25] by:

$$\delta_U \equiv \sigma_n \sqrt{2 \log_{10}(N)}, \tag{2.3}$$

where N denotes the number of samples in data Y and σ_n is the standard deviation of Gaussian noise which is removed. In real situations, the variance of noise to be removed is unknown. So Donnoho [24] proposed a special method to estimate this value by using the following equation:

$$\sigma_n \equiv \frac{\text{median}\left(|W_{1,0}^{(D)}|, |W_{1,1}^{(D)}|, ..., |W_{1,N_1-1}^{(D)}|\right)}{0.6745}, \tag{2.4}$$

where N_1 is the length of DWT subband at level 1.

Noise in the DNA data is assumed as IID. The DTCWT with interpolating (DTCWTi) method can be summarized as follows

Step 1 *: Interpolate and insert zeros into Y. Next, new data is decomposed by using the DTCWT.*

Step 2 *: Estimate the noise variance σ_n^2 by (2.4).*

Step 3 *: Calculate the threshold by (2.3)*

Step 4 *: Find the denoised coefficients from noisy coefficients by (2.2).*

Step 5 *: Reconstruct data $\hat{D}$ from the denoised coefficients by taking inverse DTCWT.*

For the SWT, only the scaling coefficients are denoised. However, for the DTCWT, all sub-band coefficients are denoised. In DTCWT method, complex sub-bands are obtained, thus the absolute values of the real SWT coefficients are replaced by the magnitudes of complex coefficients. This gives out a better result than the method using real and imaginary sub-bands separately.

A simple denoising algorithm via wavelet transform consists of three steps: decompose the noisy signal by wavelet transform, denoise the noisy wavelet coefficients according to some rules and take the inverse wavelet transform from the denoised coefficients. To estimate wavelet coefficients, some of the most well-known rules are universal thresholding, soft thresholding [25, 24, 26] and BayesShrink [27]. In these algorithms, the authors assumed that wavelet coefficients are independent. However, recently, algorithms utilizing the dependency between coefficients can give better results if compared with the ones using an independency assumption [1]. Sendur [1] has exploited this dependency between coefficients and proposed a non-Gaussian bivariate pdf for the child coefficient w_1 and its parent w_2 as follows

$$p_{\mathbf{w}}(\mathbf{w}) = \frac{3}{2\pi\sigma^2}\exp(-\frac{\sqrt{3}}{\sigma}\sqrt{|w_1|^2+|w_2|^2}). \tag{2.5}$$

The marginal variance σ^2 is dependent on the coefficients index k. Using this bivariate pdf and the Bayesian estimation theory, the MAP estimator of w_1 [1] is derived to be

$$\hat{w_1} = \frac{(\sqrt{|y_1|^2+|y_2|^2} - \frac{\sqrt{3}{\sigma_n}^2}{\sigma})_+}{\sqrt{|y_1|^2+|y_2|^2}}.y_1, \tag{2.6}$$

where $()_+$ is defined by

$$(u)_+ = \begin{cases} 0 & \text{if } u < 0, \\ u & \text{otherwise.} \end{cases} \tag{2.7}$$

This estimator can be called as a bivariate shrinkage function. In (2.6), σ can be estimated by

$$\hat{\sigma} = \sqrt{(\hat{\sigma_y}^2 - \hat{\sigma_n}^2)_+}, \tag{2.8}$$

where $\hat{\sigma_n}$ is the noise deviation which is estimated from the finest scale wavelet coefficients by using a robust median estimator [24] as follows

$$\hat{\sigma_n}^2 = \frac{\text{median}(|y_i|)}{0.6745}. \tag{2.9}$$

$\hat{\sigma_y}$ is the deviation of observation signal estimated by

$$\hat{\sigma_y}^2 = \frac{1}{M} \sum_{y_i \in N(k)} |y_i|^2, \tag{2.10}$$

where M is the size of the neighborhood $N(k)$.

The proposed method as called the DTCWTi-bi can be summarized as follows

Step 1 : *Interpolate the DNA copy number data Y and get the interpolated DNA copy number data $Y^{'}$.*

Step 2 : *Insert zeros into $Y^{'}$ and decompose new data $Y^{''}$ by the DTCWT.*

Step 3 : *Calculate the noise variance $\hat{\sigma_n}^2$ and the marginal variance $\hat{\sigma}^2$ for wavelet coefficient y_k by using (2.9), (2.10) and (2.8).*

Step 4 : *Estimate the coefficients $\hat{w_k}$ as in (2.6).*

Step 5 : *Reconstruct data $\hat{D}$ from the denoised coefficients $\hat{w_k}$ by taking inverse DTCWT.*

2.4.2 Performance Evaluation by RMSE

One thousand artificial chromosomes with six different noise levels σ_n = 0.125, 0.15, 0.2, 0.25, 0.275 and 0.3 are denoised.

Table 2.1. Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with each of 6 noise levels using the SWT, the SWTi, the DTCWTi and the DTCWTi-bi.

σ	**SWT**	**SWTi**	**DTCWTi**	**DTCWTi-bi**
0.125	0.0460	0.0422	0.0350	0.0347
0.15	0.0548	0.0497	0.0393	0.0387
0.2	0.0715	0.0631	0.0469	0.0463
0.25	0.0874	0.0751	0.0530	0.0525
0.275	0.0952	0.0810	0.0558	0.0555
0.3	0.1027	0.0867	0.0587	0.0585

The denoising results of all methods are shown in table 2.1. The proposed DTCWTi-bi method yields the better performance than the others. The DTCWTi-bi outperforms the SWT by $24.6\% - 43\%$, the SWTi [13] by $17.8\% - 32.5\%$ and the DTCWTi by $0.9\% - 1.5\%$ in terms of the RMSEs. Moreover, the DTCWTi-bi is more efficient and has less computation than the SWTi because the redundancy ratio of the DTCWT $2:1$ is much less than that of the SWT $4:1$ (if the number of level decomposition $L = 3$). For all noise levels, the DTCWTi-bi consistently achieves much better results than the SWT and SWTi.

From table 2.1, the evidence to prove that the bivariate shrinkage function should be applied to the CGH data denoising instead of the universal thresholding or the term by term thresholding. For example, at the noise level $\sigma_n = 0.15$, the RMSE of the DTCWTi-bi method is 0.0387, but that of the DTCWTi method is 0.0393. In this case, the DTCWTi-bi outperforms DTCWTi by 1.5%.

2.5 Improved Bivariate Shrinkage Model for SWPT

In this section, the bivariate shrinkage function which describes the relationship of child and parent (Fig. 2.7(a)(b)) coefficients will be reminded. Because SWPT,

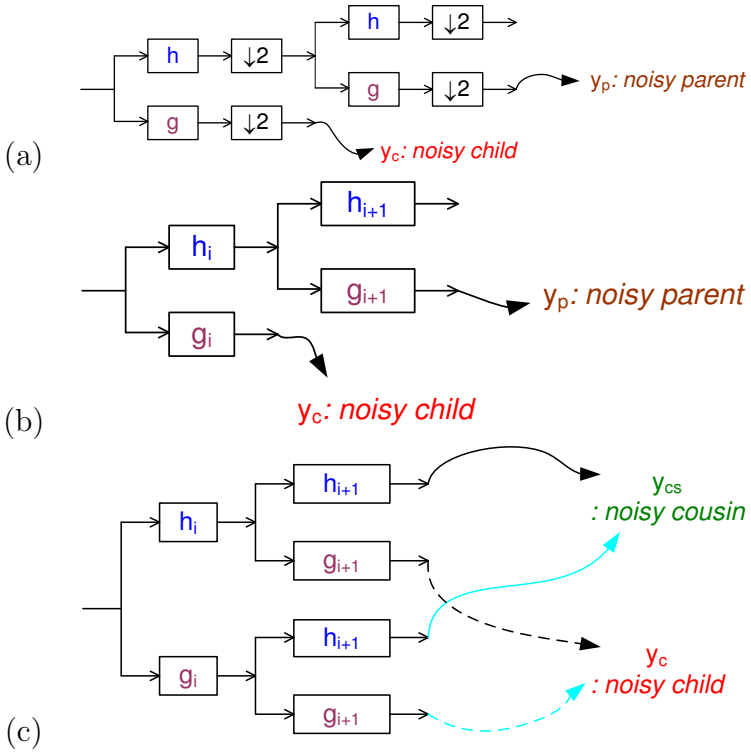


Figure 2.7. The positions of child, parent and cousin coefficients. (a) DWT, (b) SWT and (c) SWPT.

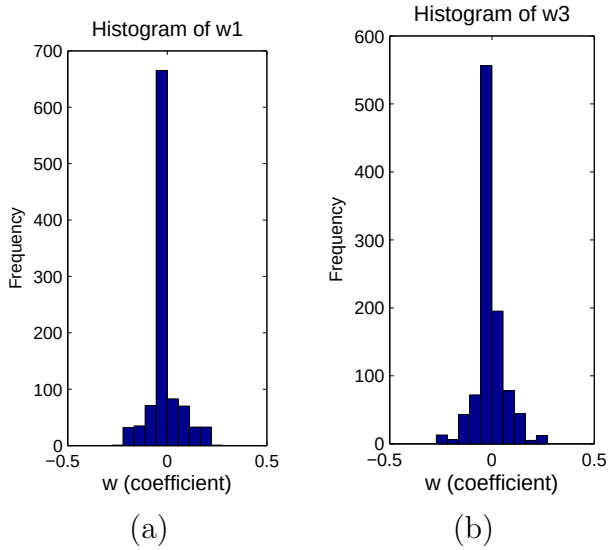


Figure 2.8. The histograms computed from true aCGH signal. (a) Histogram of w_1 , (b) Histogram of w_3.

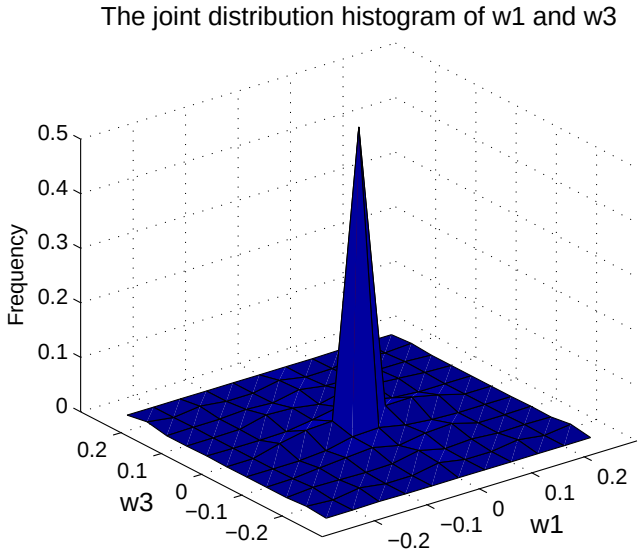


Figure 2.9. The joint distribution of w_1 and w_3 created from decomposition of true aCGH signal.

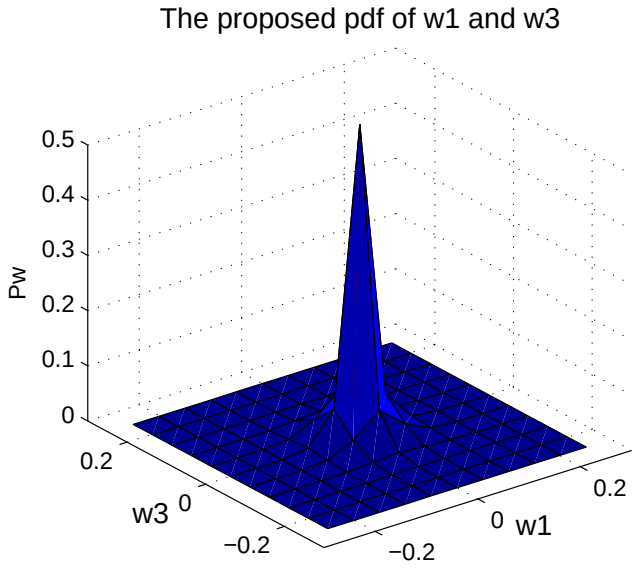


Figure 2.10. The proposed pdf with two variables: w_1 and w_3.

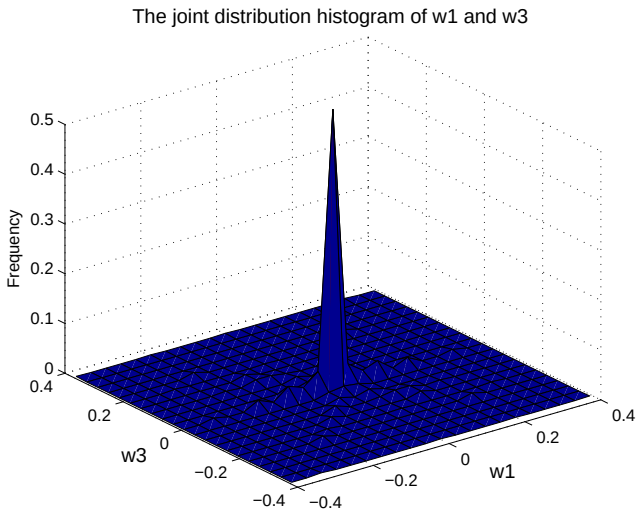


Figure 2.11. The joint distribution of w_1 and w_3 created from decomposition of true aCGH signal.

which decomposes a signal into many subbands at the same scale, just has child and cousin (Fig. 2.7(c)) at the same level, three new adaptive bivariate shrinkage functions will be developed to exploit the relationship of child and cousin coefficients.

Sendur and Selesnick [1] have recently exploited the dependency between coefficients and proposed a non-Gaussian bivariate pdf for the child coefficient w_c and its parent w_p in the complex wavelet transform domain. Nguyen et el [14] applied that function in the complex wavelet transform domain to recover aCGH data successfully and got some promising results.

From the previous section, the SWPT offers a richer range of shift-invariant subbands than the DTCWT and the SWT, and the SWPT is proposed for denoising of aCGH data. However the SWPT, which decomposes a signal into many uniform subbands at the same scale, just has child and cousin coefficients as in Fig. 2.7. Inspired by the idea of dependency between child coefficients and its parent in [1], in this section, two new adaptive bivariate shrinkage functions which model the relationship of child and cousin coefficients in the SWPT operation of aCGH data are developed.

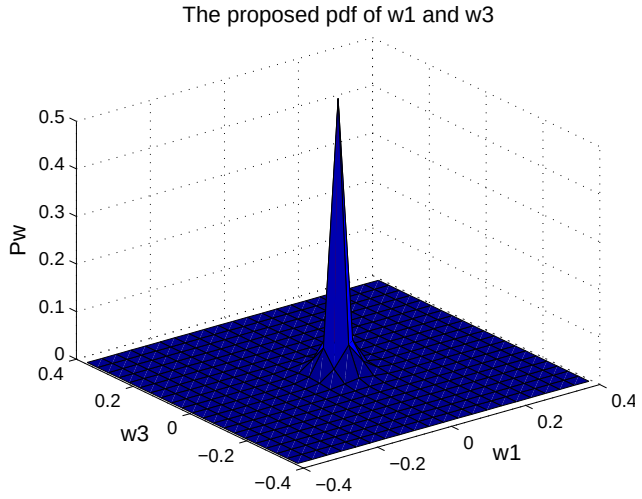


Figure 2.12. The proposed pdf with two variables w_1 and w_3.

The aCGH data Y which includes the deterministic signal D and the independent and identically distributed (IID) Gaussian noise $\mathcal{N}$ are obtained. This Gaussian noise has zero mean and variance σ_n^2.

$$Y = D + \mathcal{N}. \tag{2.11}$$

After decomposing the data Y by the SWPT, the coefficients $\mathbf{y}_k$ are obtained and those coefficients can be formulated as

$$\begin{aligned} y_{1} &= w_{1} + n_{1}, \\ y_{2} &= w_{2} + n_{2}, \end{aligned} \tag{2.12}$$

where y_{1} and y_{2} are noisy wavelet coefficients, w_{1} and w_{2} are true coefficients, w_{2} represents the cousin of w_{1} (child), n_{1} and n_{2} are independent Gaussian noise coefficients. If the cousin scale y_{2} is decomposed, detail and approximation coefficients should be obtained. Let us call y_3 as approximation coefficients of y_{2}. y_3 from y_{2} can be calculated by the following equations:

$$\begin{aligned} y_3 &= w_3 + n_3, \\ y_3[n] &= h[n] * y_{2}[n] = \textstyle\sum_{i=1}^{N} (h[n-i].y_{2}[i]), \end{aligned} \tag{2.13}$$

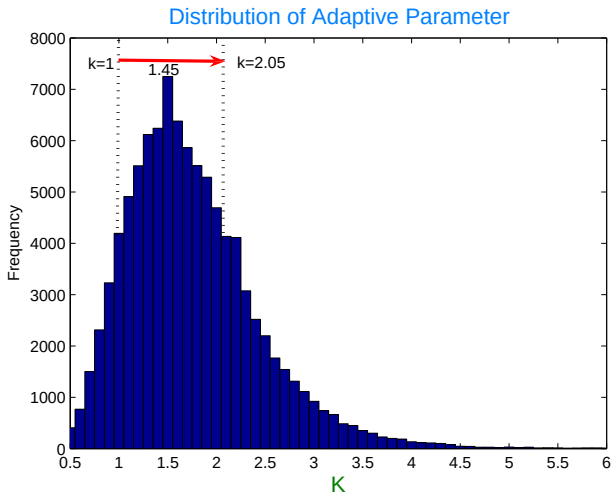


Figure 2.13. The distribution of the adaptive parameter K.

where $h[n]$ is the wavelet filter and N is the length of signal y_2. In general, $\mathcal{Y}$ can be written as follows

$$\mathcal{Y} = \mathcal{W} + \mathcal{N}, \tag{2.14}$$

where $\mathcal{Y} = (y_1, y_3)$, $\mathcal{W} = (w_1, w_3)$ and $\mathcal{N} = (n_1, n_3)$. The noise pdf should be followed as

$$p_{\mathbf{n}}(\mathbf{n}) = \frac{1}{2\pi {\sigma_n}^2} \exp(-\frac{n_1^2 + n_3^2}{2\sigma_n^2}). \tag{2.15}$$

The standard MAP estimator [1] of $\mathbf{w}$ from $\mathbf{y}$ is obtained as follows

$$\widehat{\mathbf{w}}(\mathbf{y}) = \arg\max_{\mathbf{w}}[\log_e(p_{\mathbf{n}}(\mathbf{y}\text{-}\mathbf{w})) + \log_e(p_{\mathbf{w}}(\mathbf{w}))]. \tag{2.16}$$

The Fig. 2.8 illustrates the histogram of the wavelet coefficient w_1 (child) and the approximation coefficient w_3 of w_2(cousin). The w_1 and w_2 are computed from aCGH data without noise by using the SWPT. Fig. 2.9 shows the joint distribution of w_1 and w_3. Three pdfs will be proposed to fit that joint distribution.

1. *Model 1*: the idea from [1] is imitated and a non-gaussian bivariate pdf for w_1 and w_3 is proposed as

$$p_{\mathbf{w}}(\mathbf{w}) = \frac{\mathbf{k}}{2\pi\sigma^2}\exp(-\frac{\sqrt{2\mathbf{k}}}{\sigma}\sqrt{|w_1|^2+|w_3|^2}). \tag{2.17}$$

This pdf (2.17) is sketched in Fig. 2.10. With this pdf, two variables w_1 and w_3 are really dependent. Let us define:

$$f(w) = \log_e(P_w(w)) = \log_e(\frac{\mathbf{k}}{2\pi\sigma^2}) - \frac{\sqrt{2\mathbf{k}}}{\sigma}\sqrt{|w_1|^2+|w_3|^2}. \tag{2.18}$$

By using (2.15), (2.16) becomes:

$$\widehat{\mathbf{w}}(\mathbf{y}) = \arg\max_{\mathbf{w}}[\log_e(\frac{1}{2\pi{\sigma_n}^2}) - \frac{(y_1-w_1)^2+(y_3-w_3)^2}{2\sigma_n^2} + f(w)]. \tag{2.19}$$

Solving (2.19) is the same as solving the two following equations:

$$\frac{(y_1-w_1)}{\sigma_n^2} + f_{w_1}(\widehat{w}) = 0, \tag{2.20}$$

$$\frac{(y_3-w_3)}{\sigma_n^2} + f_{w_3}(\widehat{w}) = 0, \tag{2.21}$$

where f_{w_1} and f_{w_3} represent the derivative of $f(w)$ with respect to w_1 and w_3, respectively. f_{w_1} and f_{w_3} can be obtained from (2.18)

$$f_{w_1}(\widehat{w}) = -\frac{\sqrt{2\mathbf{k}}w_1}{\sigma\sqrt{|w_1|^2+|w_3|^2}}. \tag{2.22}$$

$$f_{w_3}(\widehat{w}) = -\frac{\sqrt{2\mathbf{k}}w_3}{\sigma\sqrt{|w_1|^2+|w_3|^2}}. \tag{2.23}$$

substituting (2.22) and (2.23) into (2.20) and (2.21) gives:

$$\widehat{w_1}.(1+\frac{\sqrt{2\mathbf{k}}\sigma_n^2}{\sigma r}) = y_1, \quad \widehat{w_3}.(1+\frac{\sqrt{2\mathbf{k}}\sigma_n^2}{\sigma r}) = y_3, \tag{2.24}$$

where $r = \sqrt{|\widehat{w_1}|^2+|\widehat{w_3}|^2}$. Drawing r from (2.24):

$$r = (\sqrt{|y_1|^2+|y_3|^2} - \frac{\sqrt{2\mathbf{k}}{\sigma_n}^2}{\sigma})_+. \tag{2.25}$$

The MAP estimator can be obtained by replacing r from (2.25) into (2.24)

$$\hat{w_1} = \frac{(\sqrt{|y_1|^2+|y_3|^2} - \frac{\sqrt{2\mathbf{k}}{\sigma_n}^2}{\sigma})_+}{\sqrt{|y_1|^2+|y_3|^2}}.y_1, \tag{2.26}$$

where $()_+$ is defined by

$$(u)_+ = \begin{cases} 0 & \text{if } u < 0, \\ u & \text{otherwise.} \end{cases} \tag{2.27}$$

Replacing y_3 from (2.13) to (2.26), the MAP estimator can be rewritten as

$$\hat{w_1} = \frac{(\sqrt{|y_1|^2 + |\sum_{i=1}^{N}(h[n-i].y_2[i])|^2} - \frac{\sqrt{2\mathbf{k}}{\sigma_n}^2}{\sigma})_+}{\sqrt{|y_1|^2 + |\sum_{i=1}^{N}(g[n-i].y_2[i])|^2}}.y_1, \tag{2.28}$$

where $\mathbf{K}$ is called the adaptive parameter. In (2.28), σ can be estimated by

$$\hat{\sigma} = \sqrt{(\hat{\sigma}_y^2 - \hat{\sigma}_n^2)_+}, \tag{2.29}$$

where $\hat{\sigma}_n$ is the noise deviation which is estimated from the finest scale wavelet coefficients by using a robust median estimator.

2. *Model 2*: The second model in which the variance of the w_1 and the w_3 are different is proposed.

$$p_{\mathbf{w}}(\mathbf{w}) = \frac{\mathbf{k}}{2\pi\sigma_1\sigma_3}\exp(-\sqrt{2\mathbf{k}}\sqrt{(\frac{w_1}{\sigma_1})^2 + (\frac{w_3}{\sigma_3})^2}). \tag{2.30}$$

From (2.30), the distribution shape of the *model 2* in Fig. 2.12 matches the one in Fig. 2.11. This pdf is more general than the pdf of the *model 1* because if $\sigma_1 = \sigma_3$, the *model 2* becomes the *model 1*. In the *model2*, $f(w)$ can be written as follows

$$f(w) = \log_e(P_w(w)) = \log_e(\frac{\mathbf{k}}{2\pi\sigma_1\sigma_3}) - \frac{\sqrt{2\mathbf{k}}}{\sigma}\sqrt{|\frac{w_1}{\sigma_1}|^2 + |\frac{w_3}{\sigma_3}|^2}. \tag{2.31}$$

f_{w_1} and f_{w_3} are obtained from (2.31)

$$f_{w_1}(\widehat{w}) = -\frac{\sqrt{2\mathbf{k}}w_1}{\sigma_1^2\sqrt{|\frac{w_1}{\sigma_1}|^2 + |\frac{w_3}{\sigma_3}|^2}}, \tag{2.32}$$

$$f_{w_3}(\widehat{w}) = -\frac{\sqrt{2\mathbf{k}}w_3}{\sigma_3^2\sqrt{|\frac{w_1}{\sigma_1}|^2 + |\frac{w_3}{\sigma_3}|^2}}. \tag{2.33}$$

Substituting (2.32) and (2.33) into (2.20) and (2.21) gives:

$$\widehat{w_1}.(1 + \frac{\sqrt{2\mathbf{k}}\sigma_n^2}{\sigma_1^2 r}) = y_1, \quad \widehat{w_3}.(1 + \frac{\sqrt{2\mathbf{k}}\sigma_n^2}{\sigma_3^2 r}) = y_3, \tag{2.34}$$

where $r = \sqrt{|\frac{\widehat{w_1}}{\sigma_1}|^2 + |\frac{\widehat{w_3}}{\sigma_3}|^2}$. From (2.34), one can get:

$$\frac{(\sigma_1 y_1)^2}{(\sigma_1^2 r + \sqrt{2\mathbf{k}}\sigma_n^2)^2} + \frac{(\sigma_3 y_3)^2}{(\sigma_3^2 r + \sqrt{2\mathbf{k}}\sigma_n^2)^2} = 1. \tag{2.35}$$

With aCGH data, σ_1 and σ_3 are small and $\sigma_1^2 \approx \sigma_3^2$. w_1 can be approximately as follows

$$\hat{w_1} = \frac{(\sqrt{|\frac{y_1}{\sigma_1}|^2 + |\frac{y_3}{\sigma_3}|^2} - \sqrt{2\mathbf{k}}\frac{{\sigma_n}^2}{\sigma_1^2})_+}{\sqrt{|\frac{y_1}{\sigma_1}|^2 + |\frac{y_3}{\sigma_3}|^2}}.y_1. \tag{2.36}$$

y_3 is replaced by y_2 in (2.13). This following equation is really an improved bivariate shrinkage function:

$$\hat{w_1} = \frac{(\sqrt{|\frac{y_1}{\sigma_1}|^2 + |\frac{\sum_{i=1}^{N}(h[n-i]y_2[i])}{\sigma_3}|^2} - \sqrt{2\mathbf{k}}\frac{{\sigma_n}^2}{\sigma_1^2})_+}{\sqrt{|\frac{y_1}{\sigma_1}|^2 + |\frac{\sum_{i=1}^{N}(h[n-i].y_2[i])}{\sigma_3}|^2}}.y_1. \tag{2.37}$$

If the assumption $\sigma_1^2 \approx \sigma_3^2$ does not happen, a simple closed-form solution of (2.34) cannot be found. The successive substitution method can be used to find solution $\hat{W}_1$ as follows

(a) Set up the initial values of $\hat{W}_1^{[0]}$ and $\hat{W}_3^{[0]}$. For example, $\hat{W}_1^{[0]} = y_1$ and $\hat{W}_3^{[0]} = y_3$.

(b) Find r vector by replacing two above values to the following equation $r = \sqrt{|\frac{\hat{W}_1^{[i]}}{\sigma_1}|^2 + |\frac{\hat{W}_3^{[i]}}{\sigma_3}|^2}$.

(c) Estimate new values of $\hat{W}_1^{[i]}$ and $\hat{W}_3^{[i]}$ by using (2.38) as:

$$\hat{W}_1^{[i+1]} = \frac{y_1}{(1+\frac{\sqrt{2\mathbf{k}}\sigma_n^2}{\sigma_1^2 r})}, \quad \hat{W}_3^{[i+1]} = \frac{y_3}{(1+\frac{\sqrt{2\mathbf{k}}\sigma_n^2}{\sigma_3^2 r})}. \tag{2.38}$$

(d) Calculate the error of $\hat{W}^{[i+1]}$ and $\hat{W}^{[i]}$ as follows

$$\epsilon_1 = \hat{W}_1^{[i+1]} - \hat{W}_1^{[i]}, \quad \epsilon_3 = \hat{W}_3^{[i+1]} - \hat{W}_3^{[i]}. \tag{2.39}$$

(e) Terminate the iteration if ϵ_1 and ϵ_3 are small enough. Otherwise, jump to step (b) and continue the iteration.

3. *Model 3*:

First, the joint distribution can be assumed as an independent Laplacian as follows

$$p_w(w) = \frac{1}{2\sigma^2}\exp(-\frac{\sqrt{2}}{\sigma}|w_1| + |w_3|). \tag{2.40}$$

It is clear that the independent Laplacian distribution in Fig. 2.15 (a) does not fit well the empirical histogram. So, it is not possible to model the empirical histogram with the independent Laplacian distribution. In [1], a general joint pdf which is combined by the independent Laplacian pdf and the dependent component is proposed for image in CWT. However, the parameters of the model is tunable. So, in the case of the SWPT coefficients of the aCGH data, this bivariate model with two specific parameters is proposed to use as follows

$$p_w(w) = \frac{1}{2\pi\sigma^2}\exp(-\frac{\sqrt{3}}{\sigma}\sqrt{|w_1|^2 + |w_3|^2} - \frac{\sqrt{2}}{\sigma}(|w_1| + |w_3|)). \tag{2.41}$$

The proposed bivariate pdf in Fig. 2.15 (b) fits well the empirical histogram in Fig. 2.14. With this pdf, two variables w_1 and w_3 are really dependent and the Eq.(2.41) is named as dependent Laplacian bivariate model.

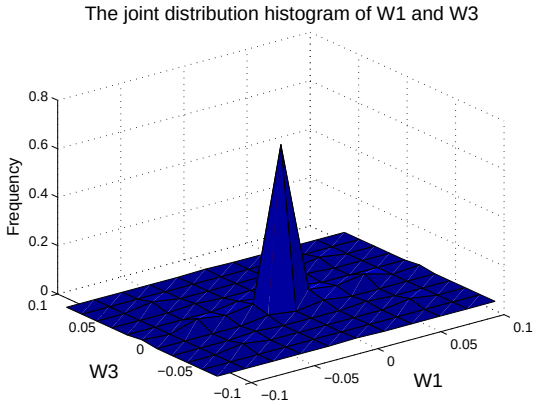


Figure 2.14. Joint distribution of w_1 and w_3 which are created from decomposition of true CGH signal.

Let us define

$$f(w) = \log_e(P_w(w)) = \log_e(\frac{1}{2\pi\sigma^2}) - \frac{\sqrt{3}}{\sigma}\sqrt{|w_1|^2 + |w_3|^2} - \frac{\sqrt{2}}{\sigma}(|w_1| + |w_3|). \tag{2.42}$$

By using Eq.(2.15), Eq.(2.16) becomes:

$$\widehat{w}(y) = \arg\max_{w}[\log_e(\frac{1}{2\pi{\sigma_n}^2}) - \frac{(y_1 - w_1)^2 + (y_3 - w_3)^2}{2\sigma_n^2} + f(w)]. \tag{2.43}$$

Solving (2.43) is the same as solving two following equations:

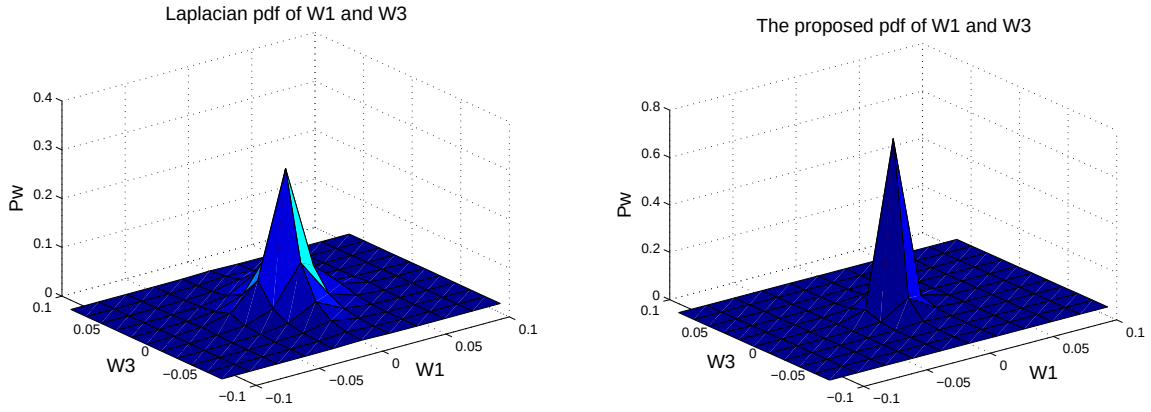
$$\frac{(y_1 - w_1)}{\sigma_n^2} + f_{w_1}(\widehat{w}) = 0, \tag{2.44}$$

$$\frac{(y_3 - w_3)}{\sigma_n^2} + f_{w_3}(\widehat{w}) = 0, \tag{2.45}$$

where f_{w_1} and f_{w_3} represent the derivative of $f(w)$ with respect to w_1 and w_3, respectively. f_{w_1} and f_{w_3} can be obtained from (2.42) as

$$f_{w_1}(\widehat{w}) = -(\frac{\sqrt{3}w_1}{\sigma\sqrt{|w_1|^2 + |w_3|^2}} + \frac{\sqrt{2}}{\sigma}sign(w_1)), \tag{2.46}$$

$$f_{w_3}(\widehat{w}) = -(\frac{\sqrt{3}w_3}{\sigma\sqrt{|w_1|^2 + |w_3|^2}} + \frac{\sqrt{2}}{\sigma}sign(w_3)), \tag{2.47}$$



(a) (b)

Figure 2.15. (a) The Laplacian pdf with two variables: w_1 and w_3, (b) The proposed pdf with two variables: w_1 and w_3.

where $sign(w)$ is defined as follow:

$$sign(w) = \begin{cases} 0 & \text{if } w = 0, \\ \dfrac{w}{|w|} & \text{otherwise.} \end{cases} \tag{2.48}$$

Substituting (2.46) and (2.47) into (2.44) and (2.45) gives

$$\begin{aligned} \widehat{w_1} \cdot (1 + \tfrac{\sqrt{3}\sigma_n^2}{\sigma r}) &= (|y_1| - \tfrac{\sqrt{2}\sigma_n^2}{\sigma})_+ \cdot sign(y_1) = soft(y_1, \tfrac{\sqrt{2}\sigma_n^2}{\sigma}), \\ \widehat{w_3} \cdot (1 + \tfrac{\sqrt{3}\sigma_n^2}{\sigma r}) &= (|y_2| - \tfrac{\sqrt{2}\sigma_n^2}{\sigma})_+ \cdot sign(y_2) = soft(y_2, \tfrac{\sqrt{2}\sigma_n^2}{\sigma}), \end{aligned} \tag{2.49}$$

where $r = \sqrt{|\widehat{w_1}|^2 + |\widehat{w_3}|^2}$ and $()_+$ is defined by

$$(u)_+ = \begin{cases} 0 & \text{if } u < 0, \\ u & \text{otherwise,} \end{cases} \tag{2.50}$$

and $soft(y, t)$ can be calculated by

$$soft(y, t) = (|y| - t)_+ \cdot sign(y). \tag{2.51}$$

Drawing r from (2.49)

$$r^2 = \frac{soft(y_1, \frac{\sqrt{2}\sigma_n^2}{\sigma})}{(1 + \frac{\sqrt{3}\sigma_n^2}{\sigma r})} + \frac{soft(y_2, \frac{\sqrt{2}\sigma_n^2}{\sigma})}{(1 + \frac{\sqrt{3}\sigma_n^2}{\sigma r})},$$

$$
\begin{aligned}
(r+\frac{\sqrt{3}\sigma_n^2}{\sigma})^2 &= soft(y_1,\frac{\sqrt{2}\sigma_n^2}{\sigma})+soft(y_2,\frac{\sqrt{2}\sigma_n^2}{\sigma}),\\
r &= (\sqrt{soft(y_1,\frac{\sqrt{2}\sigma_n^2}{\sigma})+soft(y_2,\frac{\sqrt{2}\sigma_n^2}{\sigma})}-\frac{\sqrt{3}\sigma_n^2}{\sigma})_+\\
&= (R-\frac{\sqrt{3}{\sigma_n}^2}{\sigma})_+ \ .
\end{aligned}
\tag{2.52}
$$

The MAP estimator can be obtained by replacing r from (2.52) into (2.49)

$$
\hat{w_1} = \frac{(R-\frac{\sqrt{3}{\sigma_n}^2}{\sigma})_+}{R}\cdot soft(y_1,\frac{\sqrt{2}\sigma_n^2}{\sigma}), \tag{2.53}
$$

where R is as follows

$$
R=\sqrt{soft(y_1,\frac{\sqrt{2}\sigma_n^2}{\sigma})^2+soft(y_3,\frac{\sqrt{2}\sigma_n^2}{\sigma})^2}. \tag{2.54}
$$

Eq.(2.53) is called as dependent Laplacian bivariate shrinkage function. Packet wavelet transform does not include any parent scale. In this case, hard thresholding estimator [25] can be used to recover cousin coefficients $\hat{w_{cs}}$:

$$
\hat{w_{cs}} = (y_{cs}-\sigma_n\sqrt{2\log_{10}N})_+. \tag{2.55}
$$

In (2.37), σ_1 and σ_3 can be estimated by

$$
\begin{aligned}
\hat{\sigma}_1 &= \sqrt{(\hat{\sigma}_{y_1}^2-\hat{\sigma}_n^2)_+},\\
\hat{\sigma}_3 &= \sqrt{(\hat{\sigma}_{y_3}^2-\hat{\sigma}_n^2)_+},
\end{aligned}
\tag{2.56}
$$

where $\hat{\sigma}_{y_1}^2$ and $\hat{\sigma}_{y_3}^2$ are the variances of y_1 and y_3. They can be estimated by

$$
\begin{aligned}
\hat{\sigma}_{y_1}^2 &= \tfrac{1}{M_1}\textstyle\sum_{y_{1i}\in N_1(i)}|y_{1i}|^2,\\
\hat{\sigma}_{y_3}^2 &= \tfrac{1}{M_3}\textstyle\sum_{y_{3i}\in N_3(i)}|y_{3i}|^2,
\end{aligned}
\tag{2.57}
$$

where M_1 and M_3 are the size of the neighborhood $N_1(i)$ and $N_3(i)$ respectively. $N_1(i)$ and $N_3(i)$ are two windows with y_{1i} and y_{3i} at the center. The y_{3i} can be created by applying (2.13). The relationship of child and cousin coefficients was really exploited

Table 2.2. Comparison table of the improved bivariate shrinkage function and function in [1].

Method	Improved function.	Old function [1].
Applying to	aCGH data.	image.
Relationship	child and cousin coefficient.	child and parent coefficient.
adaptive **k**	use **k** from 1 to 2.05.	no use.
Model2	a simple-closed-form (2.37).	no simple-closed-form.
Transform	SWPT and DWPT.	DWT and DTCWT.

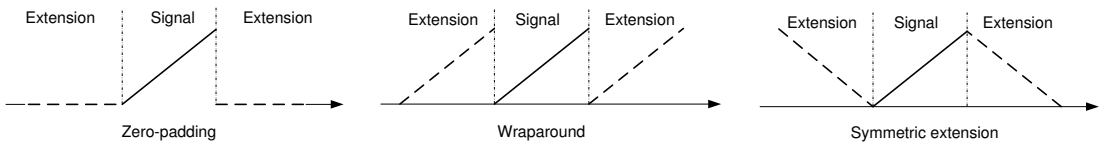


Figure 2.16. Three extension methods [28]: Zero-padding, Wraparound and Symmetric extension.

as (2.28) , (2.37) and (2.53). In those estimators, the way choosing value of **k** is very interesting. If the value of **k** depends on each scale, those estimators can be called as adaptive bivariate shrinkage function. The distribution of **k** is showed in Fig. 2.13. In the packet wavelet transform, the cousin scales have not any parent scale. In this case, hard thresholding estimator [25] can be used to recover cousin coefficients $\hat{w_{cs}}$:

$$\hat{w_{cs}} = (y_{cs} - \sigma_n \sqrt{2 \log_{10} N})_+. \tag{2.58}$$

Now, after improved bivariate shrinkage functions are obtained, these new two functions should be compared to the bivariate function of Sendur [1] as table. 2.2. From this table, these functions have five different parts with Sendur's. So, three functions (2.28), (2.37) and (2.53) can be considered as really new ones.

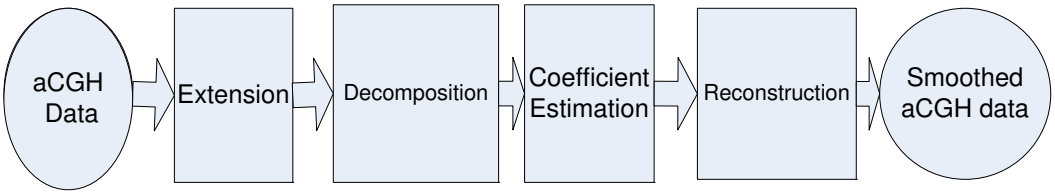


Figure 2.17. The flowchart of SWPT-LaBi method.

2.6 SWPT-LaBi Algorithm

2.6.1 Signal Extension

Array CGH data is a finite signal. If real data is denoised directly, error may be obtained at the border of denoised signal. So, extension step is a very important preprocessing step before denoising. There are three main extension methods shown in Fig. 2.16. According to the book [28] (chapter 8), symmetric extension is the best if applied to a filtered image because information can be saved at the border better. With aCGH data, saving the information is necessary at the border. In 2.7.2, three extension methods are applied and agreed with [28]. From the results of the section 2.7.2 and Fig. 2.21, the symmetric extension method should be use as a preprocessing step before denoising. Let us assume that the length of the aCGH signal is N. In order to get the best performance in the wavelet denoising algorithm, the length of the input signal is required to be a power of two [29]. If N is not a power of two, signal can be extended to make sure $N = 2^j$ by using symmetric extension method.

2.6.2 Proposed Method

Fig. 2.17 is the flowchart of SWPT-LaBi algorithm which can be summarized as follows

Step 1 : *Extend aCGH data Y using symmetric extension method and decompose new data $Y^{'}$ by the SWPT to L levels as Eq.(2.61). The number of decomposition levels [30] (at the remark* 11*) can be computed by*

$$L = \log_2(N) - J, \tag{2.59}$$

where $J = 3, 4, 5, 6$. This is a perfect number of levels [30] which yields the best denoising result. In this chapter, $J = 4$ can be used as the same as [9] and [13].

Step 2 : *Calculate the noise variance $\hat{\sigma}_n^2$ and the marginal variance $\hat{\sigma}^2$ for wavelet coefficient y_k by using Eq.(2.9), Eq.(2.10) and Eq.(2.8).*

Step 3 : *Estimate the child coefficients $\hat{w}_c = \hat{w}_1$ as in Eq.(2.53) and estimate the counsin coefficients $\hat{w}_{cs}$ as in Eq.(2.55).*

Step 4 : *Reconstruct data $\hat{D}$ from the denoised coefficients $\hat{w}_c$ and $\hat{w}_{cs}$ by taking the inverse SWPT.*

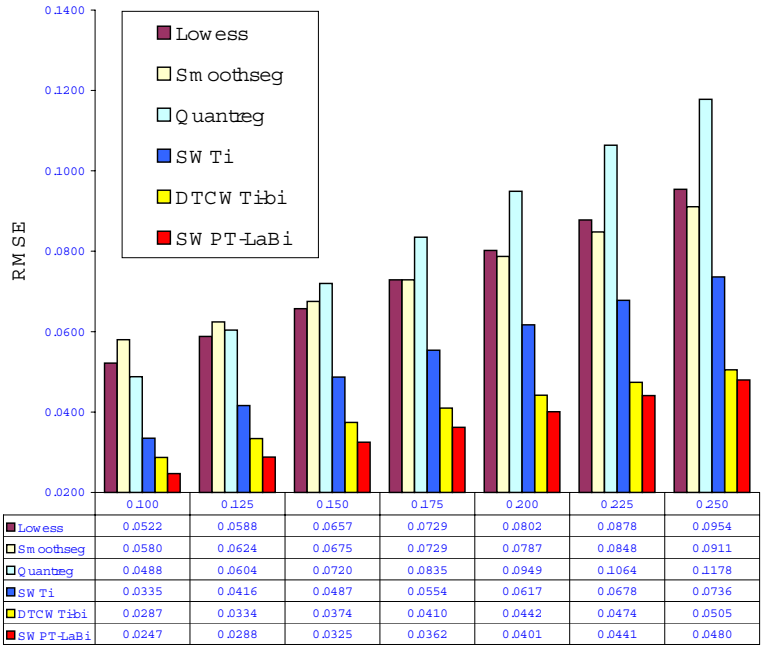
The error of smoothing result could be measured by the root mean squared error (RMSE) that is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i}^{N}(\hat{D}_i - D_i)^2}, \tag{2.60}$$

where N is the number of input samples, $D = \{D_i\}$ and $\hat{D} = \{\hat{D}_i\}$ are the values of data points before and after smoothing.

2.6.3 Performance Evaluation by RMSE

In this section, the experimental results of Lowess [8], Quantreg [10, 22], Smoothseg [23], SWTi [13], DTCWTi-bi [14], and SWPT-LaBi methods will be compared. One thousand artificial chromosomes with Gaussian noise in seven different levels $\sigma_n =$ 0.1, 0.125, 0.15, 0.175, 0.2, 0.25 and 0.275 are denoised. Meantime, simulated chromosomes with real noise are also used to test above six methods.



	0.100	0.125	0.150	0.175	0.200	0.225	0.250
Lowess	0.0522	0.0588	0.0657	0.0729	0.0802	0.0878	0.0954
Smoothseg	0.0580	0.0624	0.0675	0.0729	0.0787	0.0848	0.0911
Quantreg	0.0488	0.0604	0.0720	0.0835	0.0949	0.1064	0.1178
SW Ti	0.0335	0.0416	0.0487	0.0554	0.0617	0.0678	0.0736
DTCW Ti-bi	0.0287	0.0334	0.0374	0.0410	0.0442	0.0474	0.0505
SW PT-LaBi	0.0247	0.0288	0.0325	0.0362	0.0401	0.0441	0.0480

Figure 2.18. Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with each of 7 noise levels (Gaussian noise).

The denoising results of all methods are shown in the Fig. 2.18. The proposed SWPT-LaBi method has a better performance than the others. The SWPT-LaBi outperforms the Lowess by $49.7\% - 52.7\%$, the Quantreg by $47.3\% - 57.4\%$, the Smoothseg by $49.4\% - 59.3\%$, the SWTi by $26.3\% - 35\%$ and the DTCWTi-bi by $5\% - 13.9\%$ in terms of the root mean squared errors. For all noise levels, the SWPT-LaBi consistently achieves much better results than others.

In the experiments using real noise synthetic data, 15 chromosomes are used to create real noise for synthetic data. One thousand chromosomes are tested with six above methods. Fig. 2.19 shows that the RMSE of the SWPT-LaBi (0.0453) is the smallest one when compared to the Lowess (0.0771), the Smoothseg (0.0813), the Quantreg (0.0940), the SWTi (0.0700), and the DTCWTi-bi (0.0492). The proposed method outperforms all previous methods between 7.9% and 51.8% on the synthetic data with real noise.

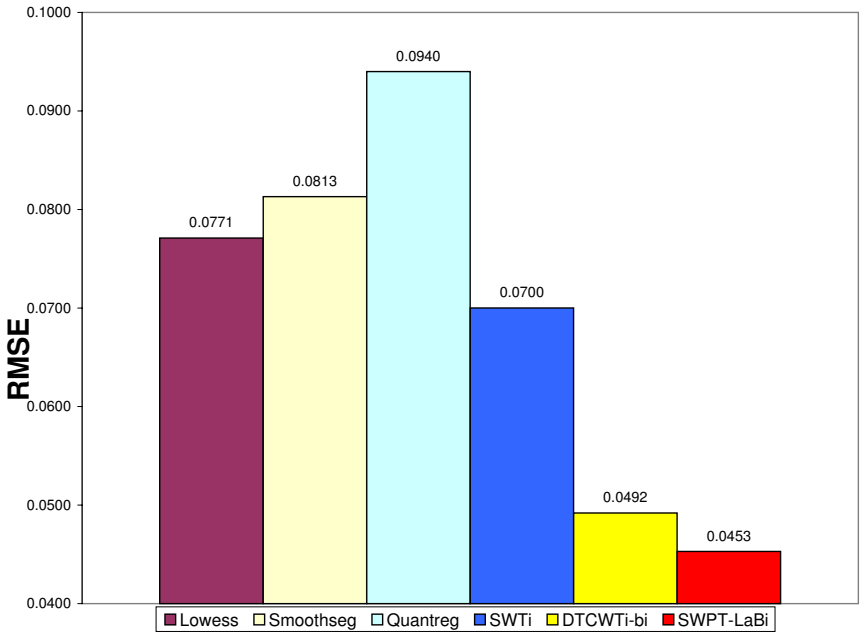


Figure 2.19. Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with real noise.

2.6.4 Performance Evaluation by ROC curves

Paper [8] introduced another method to evaluate aCGH smoothing algorithms by ROC curve. Several hundred artificial chromosomes, consisting of 100 probes and with the square-wave signal at the center of the chromosome, are created from four templates. In 2007, Huang *et al.* [23] modified this setting to make the problem harder. The modification decreased the width of the center square-wave and increased the noise level. However people usually want to test the performance of methods not only at the middle of signal but also at the border of signal. Therefore, four templates with the aberration widths of 5, 20, 30 and 40 are kept. Three more templates (one or two aberrations) with the aberration widths of 20, 10 and 5 should be added at the border.

From seven genomic templates with one or two aberrations, 270 samples are generated with unequal space probes. The ROC profiles of SWTi, DTCWTi-bi,

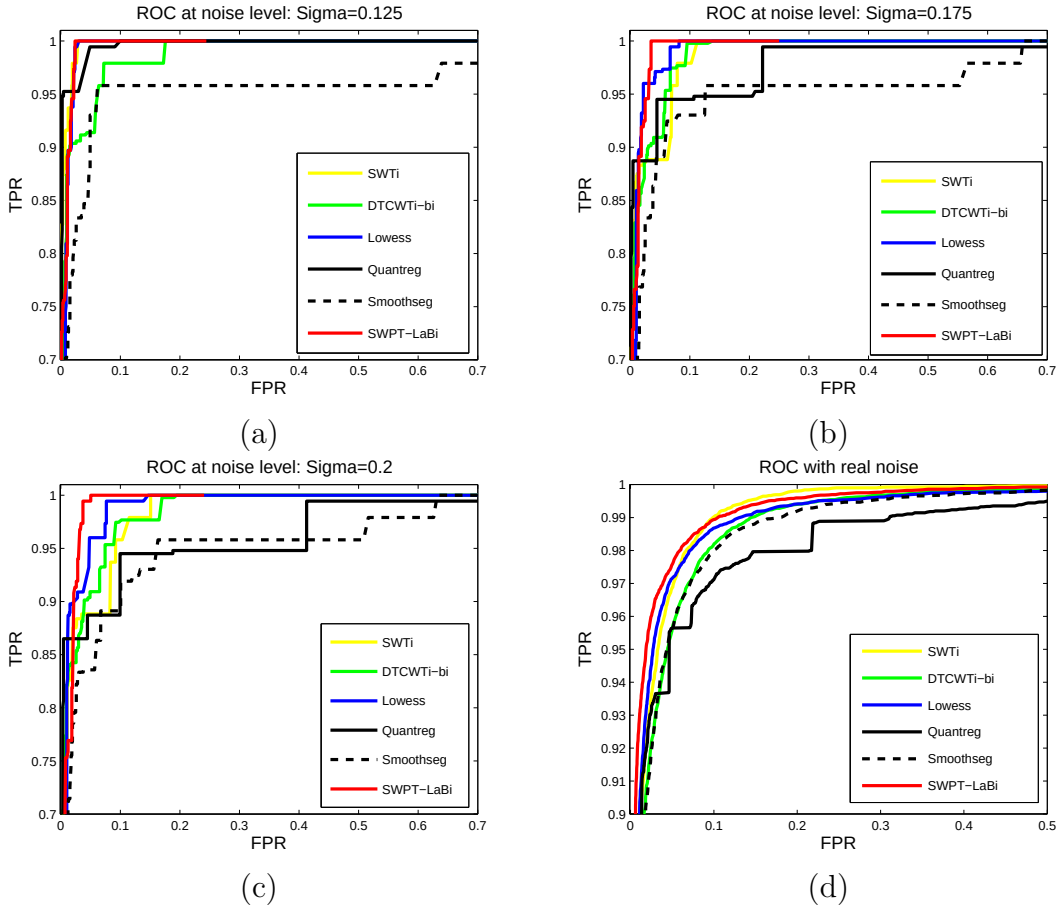


Figure 2.20. Receiver operating characteristic (ROC) curves obtained from 270 artificial chromosomes (generated from 7 genomic templates) with each of the different noise levels using the SWPT-LaBi and other most common used CGH algorithms such as SWTi, DTCWTi-bi, Lowess, Quantreg, and Smoothseg.

Lowess, Quantreg, Smoothseg, and SWPT-LaBi methods are calculated. Fig. 2.20 illustrates the ROC curves with different noise levels: $\sigma = 0.125, 0.175, 0.2$. The TPR is defined as the number of probes inside the aberration whose absolute values are above the threshold level divides by the number of probes in the aberration. The FPR is defined as the number of probes outside the aberration or the number of probes inside the copy two region whose absolute value are above the threshold level divides by the total number of probes outside the aberration. The threshold level is changed from 0 to 1.

In Fig 2.20 (a,b,c), SWPT-LaBi method clearly performs better than other methods. If some methods in time domain such as Lowess and Quantreg are just compared, the Lowess looks better. This result also agrees with the experimental results in [8]. In low noise level ($\sigma = 0.125$), in Fig 2.20 (a), most methods operate well except Smoothseg. If noise is increased, the Quantreg gets worse in Fig 2.20 (b,c). Of course, with Gaussian noise, the Smoothseg always gets worse than the others because it is designed to operate with the student't noise. From three above figures, the proposed method always gives out the best results.

Real noise from the chromosome 13 of GSM232967 is extracted and 270 simulated aCGHs are created with real noise from seven genomic templates with one or two aberrations. Fig 2.20 (d) shows the ROC curve results of 270 above simulated aCGHs with real noise. The performance of Smoothseg becomes better but it still gets worse than the proposed method. In this case, the SWPT-LaBi is the best in the low FDR area that is more meaningful in practical aCGH applications. In summary, the proposed method is the best one in ROC curve comparison.

2.7 SWPT-AdaBi Algorithm

2.7.1 Proposed Method

The proposed method is named as SWPT-AdaBi. Before configuration of SWPT-AdaBi, a simpler method is set up: SWPT-bi. The method as called the SWPT-bi can be summarized as follows

Step 1 : *Extend Y by using symmetric extension method and decompose new data Y' by SWPT to L levels as (2.61). The numbers of decomposition levels [30] (at the remark* 11*) can be computed by*

$$L = \log_2(N) - J, \tag{2.61}$$

where $J = 3, 4, 5, 6$. This is a perfect number of levels [30] which yields the best denoising result. In this chapter, $J = 4$ is used as the same as in [9] and [13].

Step 2 : *Calculate the noise variance $\hat{\sigma}_n^2$ and the marginal variance $\hat{\sigma}^2$ for wavelet coefficient y by using (2.9), (2.10) and (2.8).*

Step 3 : *Estimate the child coefficients $\hat{w}_c = \hat{w}_1$ as in (2.28) and estimate the counsin coefficients $\hat{w}_{cs}$ as in (2.58). In this case, $\boldsymbol{k} = 1.45$ should be chosen.*

Step 4 : *Reconstruct data $\hat{D}$ from the denoised coefficients $\hat{w}_c$ and $\hat{w}_{cs}$ by taking the inverse SWPT.*

Two models have been developed in *model 1* and *model 2*. That is the reason why two methods are proposed. They are SWPT-AdaBi (with *model 1*) and SWPT-AdaBi2 (with *model 2*). The "Ada" means using adaptive parameter **k** in functions (2.28) and (2.37). The SWPT-AdaBi uses the estimator (2.28) from the *model 1* and the adaptive parameter **k**. Both of two adaptive methods will give the better results than SWPT-bi. It is an evidence that the adaptive parameter **k** will be a right choice instead of a fixed $\mathbf{k} = 1.45$.

2.7.2 Comparisons of Extension Methods

The reason why the symmetric extension method is chosen in the preprocessing step will be demonstrated. Three extension methods such as zero-padding, wraparound, and symmetric extension are applied to extend at both sides of original aCGH signal, respectively. After that, extended signals will be denoised using SWPT-AdaBi. From the results in Fig. 2.21, the RMSE of denoised signal using symmetric extension is the best. The next is wraparound method. The worst one is the zero-padding method. Results are consistent with [28] when they are applied to denoise images. If SWPT is used directly, the transform will use circular convolution [28]. In MATLAB, they use the function "wextend" with extension mode as

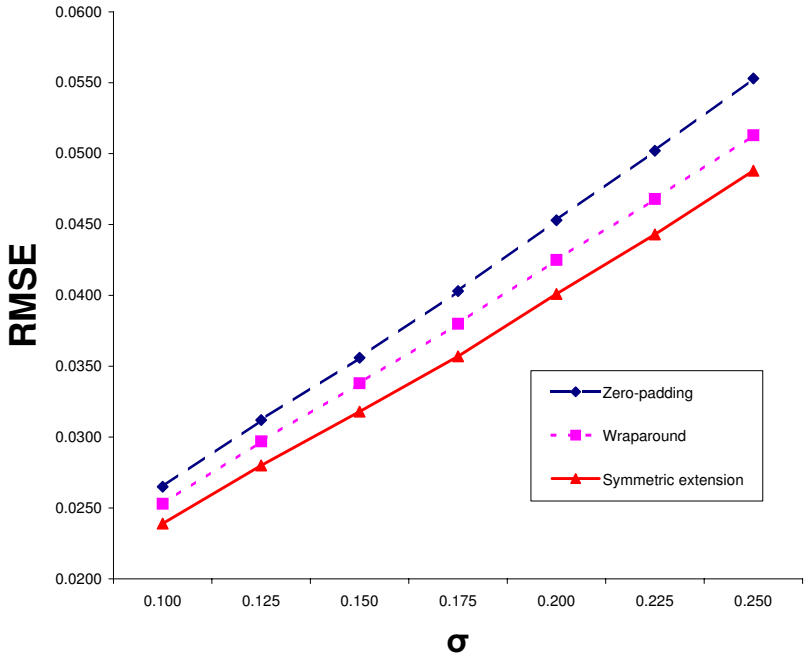
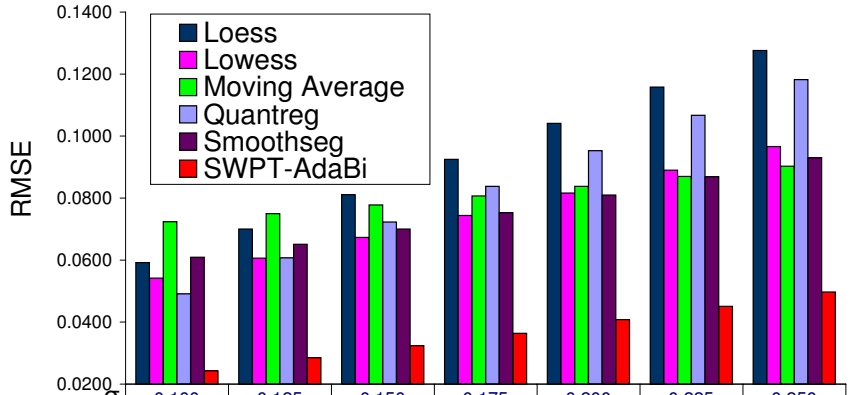


Figure 2.21. Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with each of 7 noise levels using SWPT-Adabi with three extension methods in the preprocessing step.

period or wraparound before taking convolution in wavelet transform. That means the result can be obtained as the same the middle line (wraparound) in the Fig. 2.16 if SWPT is applied to aCGH data directly. In conclusion, symmetric extension is the best choice to preprocess aCGH data before denoising.

2.7.3 Experiments Design

In order to compare the performance of SWPT-AdaBi to other most commonly used methods and also observe how new bivariate shrinkage functions improved performance, the previous methods are separated into two groups: five most commonly used methods in literature and three other methods using wavelet transform from other researchers.



σ	0.100	0.125	0.150	0.175	0.200	0.225	0.250
Loess	0.0592	0.0700	0.0811	0.0925	0.1041	0.1158	0.1276
Lowess	0.0542	0.0606	0.0673	0.0744	0.0816	0.0890	0.0966
Moving Average	0.0724	0.0750	0.0778	0.0807	0.0838	0.0870	0.0903
Quantreg	0.0491	0.0607	0.0723	0.0838	0.0953	0.1067	0.1182
Smoothseg	0.0609	0.0651	0.0700	0.0753	0.0810	0.0869	0.0930
SWPT-AdaBi	0.0243	0.0285	0.0324	0.0364	0.0408	0.0451	0.0497

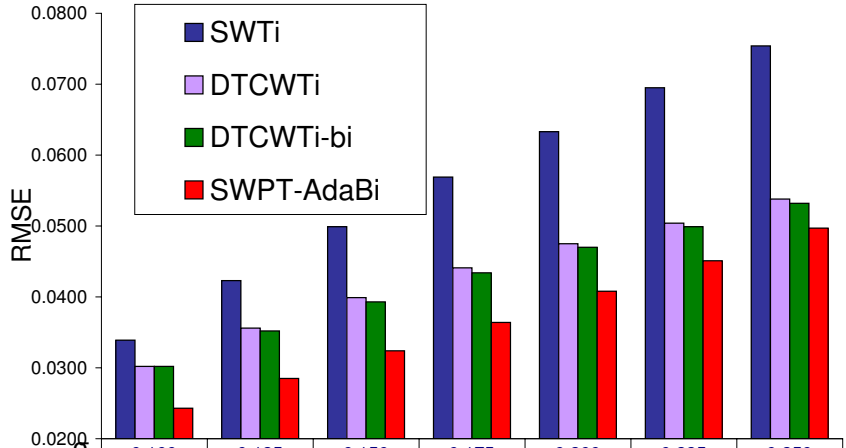
Figure 2.22. Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with each of the 7 noise levels using the proposed method SWPT-AdaBi and some methods in time domain such as Loess, Lowess, Moving Average, Quantreg, Smoothseg.

The first group includes other most commonly used methods that analyze aCGH in time domain [8], such as Loess, Lowess, Quantreg, Smoothseg, and Moving Average.

The second group includes previous works using various wavelet transforms such as SWTi [13], DTCWTi [31] and DTCWTi-bi [14]. The MODWT method [9] which was compared in [13] is worse than SWTi method. Thus, the proposed method will be compared to SWTi method.

2.7.4 Performance Evaluation by RMSE

One thousand artificial chromosomes with seven different levels $\sigma_n = 0.1$, 0.125, 0.15, 0.175, 0.2, 0.225 and 0.25 are smoothed. The SWPT-AdaBi method will be compared to the other methods without using wavelet transform in Fig. 2.22, with using wavelet transform in the previous work in Fig. 2.23.



σ	0.100	0.125	0.150	0.175	0.200	0.225	0.250
SWTi	0.0339	0.0423	0.0499	0.0569	0.0633	0.0695	0.0754
DTCWTi	0.0302	0.0356	0.0399	0.0441	0.0475	0.0504	0.0538
DTCWTi-bi	0.0302	0.0352	0.0393	0.0434	0.0470	0.0499	0.0532
SWPT-AdaBi	0.0243	0.0285	0.0324	0.0364	0.0408	0.0451	0.0497

Figure 2.23. Comparison of average RMSEs obtained from the 1,000 artificial chromosomes with each of the 7 noise levels using the proposed method SWPT-AdaBi and some previous methods using wavelet transform such as SWTi, DTCWTi, DTCWTi-bi.

From Fig. 2.22, the proposed SWPT-AdaBi method yields the performance much better than the others such as Loess, Lowess, Moving Average, Quantreg and Smoothseg. The SWPT-AdaBi outperforms Loess by 60%−61.1%, Lowess by 48.9%−55.2%, Moving Average by 45% − 66.4%, Quantreg by 50.5% − 58% and Smoothseg by 46.6% − 60% in terms of the RMSEs.

From Fig. 2.22 and the above discussion, the proposed method gives the best performance if compared to some denoised methods in time domain.

The RMSE of SWPT-AdaBi is still the lowest if compared to the other wavelet methods such as SWTi, DTCWTi and DTCWTi-bi showed in table and Fig. 2.23. The SWPT-AdaBi achieves the results better than SWTi by 28.3% − 36%, DTCWTi by 7.6% − 20% and DTCWTi-bi by 6.6% − 19.5% in terms of the RMSEs.

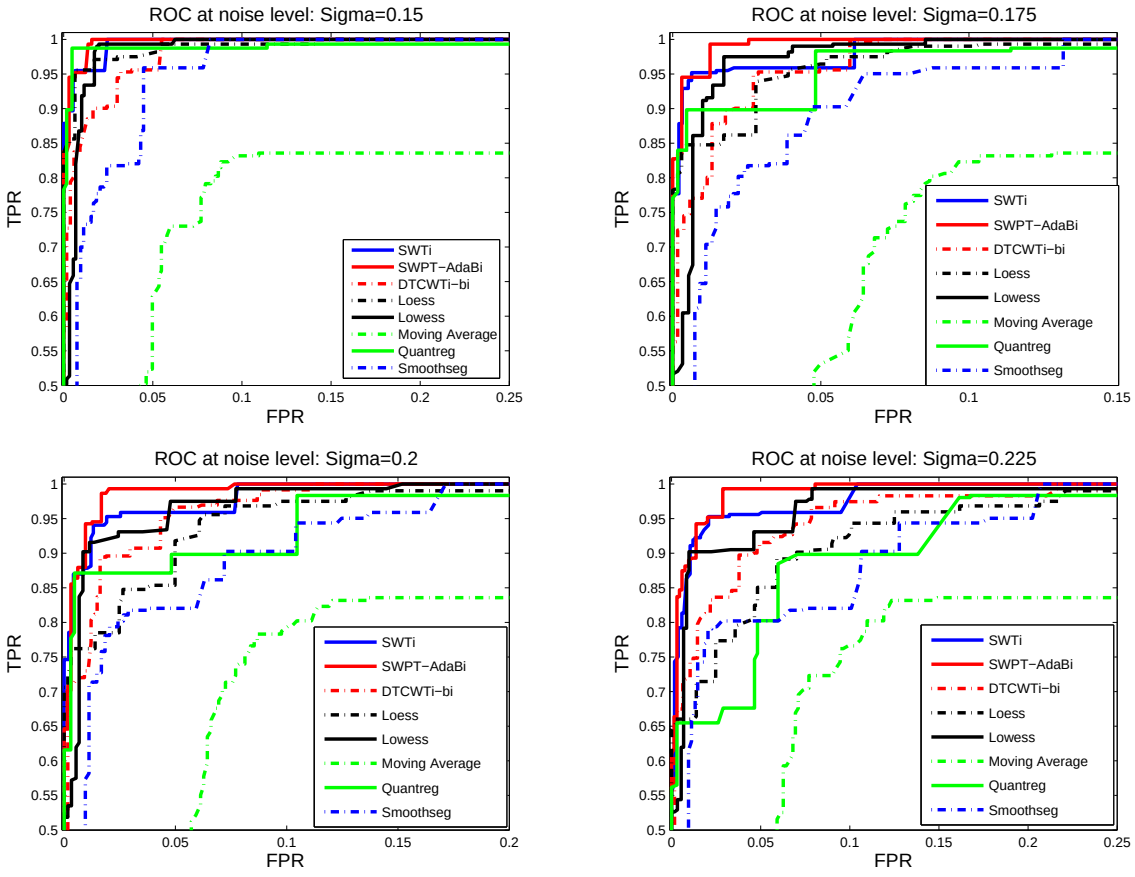


Figure 2.24. Receiver operating characteristic (ROC) curves obtained from the 280 artificial chromosomes (generated from 7 genomic templates) with each of the different noise levels using SWPT-AdaBi and some aCGH algorithms such as SWTi, DTCWTi-bi, Loess, Lowess, Moving Average, Quantreg, Smoothseg.

2.7.5 Performance Evaluation by ROC Curve

From seven genomic templates with one or two aberrations (the other copies except for copy two), 280 samples are generated with unequal space probes. The ROC profiles of many algorithms such as SWTi, DTCWTi-bi, Loess, Lowess, Moving Average, Quantreg, Smoothseg and SWPT-AdaBi method are caculated. Fig 2.24 illustrates the ROC curve at different noise levels: $\sigma = 0.15, 0.175, 0.2, 0.225$.

In Fig 2.24, SWPT-AdaBi method clearly performs better than the other methods. If some methods (Loess, Lowess, Moving Average, Quantreg, Smoothseg) which

are performed in time domain method are compared, Lowess looks better. This result also agrees with the result in [8]. MODWT [9] should be not compared because it is worse than SWTi [13]. In the lower noise (not showed here), most of methods work very well and their TPRs are very close 1 with FPRs less than 0.1 except for Moving Average. The Moving Average does not work well because it is very difficult to choose a right window for this method. In summary, proposed method still is still the best if compared by ROC curve.

2.8 Conclusion

In this chapter, three novel models which exploit the dependency of child and cousin coefficients of aCGH data in SWPT domain are proposed. Three improved bivariate shrinkage estimators are derived to adaptively estimate noise thresholding. The proposed methods were compared to most commonly used methods in literature by using standard synthetic aCGH data. The denoising results from SWPT-AdaBi are much better than previous time domain methods with $45\% - 66.4\%$ improvement and better than previous wavelet transform based methods with $6.6\% - 36\%$ improvement in terms of the root mean squared error measurements at different noise levels. Furthermore, the performance of the proposed methods are also demonstrated by using the real aCGH data.

CHAPTER 3
HEAVY-TAILED NOISE SUPPRESSION AND DERIVATIVE WAVELET SCALOGRAM BASED SEGMENTATION OF ARRAY-CGH DATA

3.1 Introduction

When designing and evaluating chromosomal aberration detection algorithms, most researchers assume that noise in array CGH follows Gaussian distribution [10, 32, 13, 15, 8, 31, 14]. However, this important assumption has been queried and discussed by Hu *et. al.* [33]. They showed that array CGH noise distribution is heavy-tailed, but they did not make conclusion on array CGH noise distribution. Huang *et. al.* [23] considered array CGH noise distribution as a student's t distribution. To address this important problem, in this chapter, noise in real array CGH data is explored and any deviations from zero values in normal or self-self test samples (true signal is expected zero over whole sample) are considered as noise. After the real array CGH data are analyzed, noise distribution of array CGH data should be assumed as generalized Gaussian distribution (GGD) which also covers Gaussian distribution. Five real array CGH data sets with different resolutions are used to support for new noise model assumption. Based on new noise model, two new synthetic array CGH data models with GGD noise and real noise are introduced. Hybridization bias problem [34] is also considered in synthetic array CGH data models. New synthetic array CGH data models are more accurate than traditional models to evaluate array CGH analysis algorithms.

In order to develop effective methods identifying aberration regions from array CGH data, the previous research works focus on two major issues: smoothing-based and segmentation-based methods. In 2005, Lai *et al.* [8] compared 11 different

array CGH analysis algorithms through empirical experiments and concluded that segmentation-based methods perform consistently well and when the noise is high, smoothing methods work better.

Smoothing-based methods based on frequency domain to remove noise can discover small amplitude aberration regions and reduce the number of identified false aberration regions. However, smoothing-based method could not detect exactly breakpoints of abberation regions because changing points in array CGH are corresponding to high frequency which could be cut in denoising process. Segmentation-based methods targeted to model data as a series of discrete segments under certain optimization criterion try to detect breakpoints and directly give out the final results with visible gain, deletion or normal cases. The segmentation-based methods could more accurately detect the boundary points. Since the small aberration regions are highly possible to be buried into its neighbors in high noise case, the false positives are introduced. It would be very desirable to develop new methods to analyze array CGH data with advantages from both smoothing and segmentation approaches [33].

In this chapter, a novel derivative wavelet scalogram based segmentation (DWSS) method that is integrated by both smoothing and segmentation steps is proposed. This method works well with heavy-tailed noise. DWSS method includes two main steps: heavy-tailed noise suppression and breakpoint detection. In [8], Wave method using stationary wavelet transform works very well with Gaussian noise signal. Instead of hard threshold [9], generalized Gaussian bivariate shrinkage function will be designed in stationary wavelet transform to suppress heavy-tailed noise in array CGH. In 2008, Ben *et al.* [9] proposed HaarSeg algorithm using simple wavelet based pattern-matching or wavelet footprint to detect breakpoints in array CGH. HaarSeg algorithm running very fast gave a promising result in array CGH's segmentation. More pattern-matching by Gaussian derivative wavelet scalogram will be studied to

segment processed array CGH. From 2008 to 2009, Pique-Regi *et al.* [35] [34] proposed two GADA algorithms in which GADA1 is designed with an assumption that measurements are unbiased, GADA2 works well with probe hybridization bias. It is necessary to have a segment method working with both bias and unbias cases. Our method is designed to be robust with probe hybridization biases [34] in array CGH. Both Root Mean Square Errors (RMSE) and Receiver Operating Characteristic (ROC) curves are calculated to demonstrate performance. In all experiments, the new DWSS method whose speed is faster than two previous methods in state of the art outperforms the previous most commonly used array CGH analysis algorithms

3.2 Array CGH Noise Characteristic

In this section, array CGH noise distribution analysis which used to be studied by Hu *et. al.* [33] will continue being analyzed. More datasets which include some self-test samples will be used to fit noise models in array CGH data. Generalized Gaussian distribution is proposed as a fittest noise model for this data. Relative entropy will be used to validate new array CGH noise model.

3.2.1 Data Description

Five real array CGH datasets such as Lee 2008 array [36], Snijders 2001 array [37], Bredel 2005 array [38], Smith 2007 array [39] and Nicolas 2009 array [40] are analyzed. They are the public array CGH data that could be used to study array CGH noise. Since the true signal of a normal chromosome should only include copy two, deviations from zero ($\log_2(2/2) = 0$) values in real signal of normal chromosome are considered as noise in array CGH.

In the Lee 2008 array [36] whose platform is Nimblegen Macaque Whole genome CGH 385K array, there are two self-self test samples (GSM232967, GSM232968) of

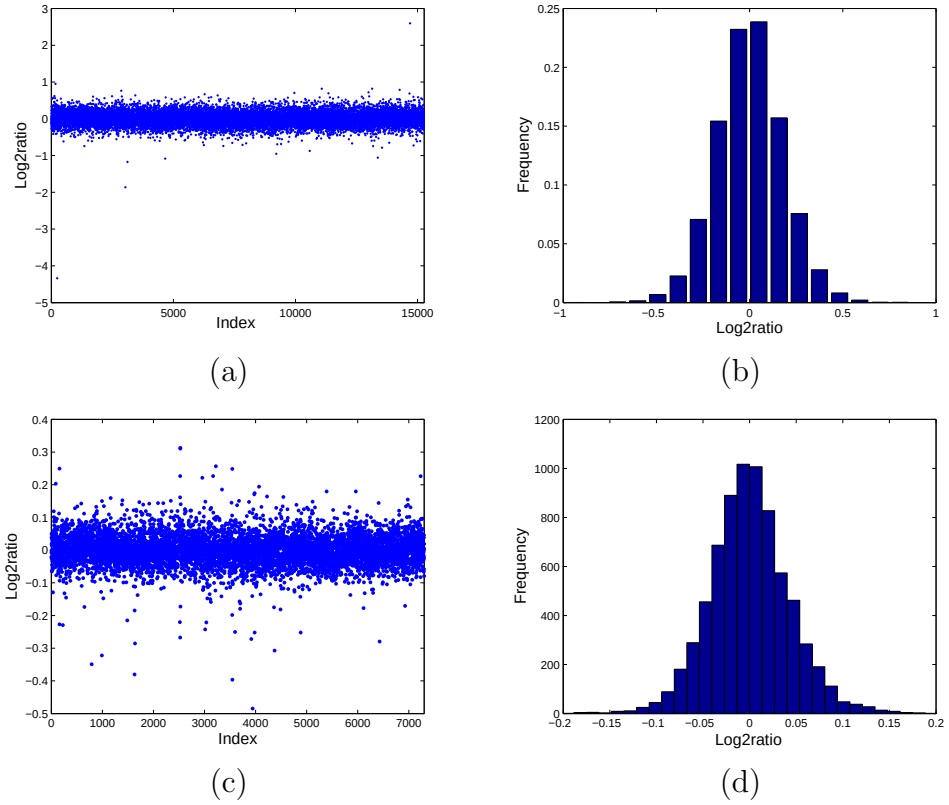


Figure 3.1. Examples of array CGH and their empirical histograms: (a) Chromosome 15 of GSM232967, (b) Empirical histogram of chromosome 15 of GSM232967, (c) Chromosome 13 of GSM215042, (d) Empirical histogram of chromosome 13 of GSM215042.

Table 3.1. Five datasets which are used to analyze noise in array CGH with many platforms

Dataset	Number of Arrays	Platform
Lee 2008 array [36]	40	Nimblegen Macaque Whole genome CGH 385K array
Snijders 2001 array [37]	15	
Bredel 2005 array [38]	26	
Smith 2007 array [39]	69	Agilent-015366 Custom Human 244K CGH Microarray
Nicolas 2009 array [40]	23	Custom Nimblegen array CGH chip targeted to canFam2 segmental duplications

log2-transformed ratios (CH1/CH2) with some ten-thousand probes. Totally there are 40 (2 samples $\times$ 20 chromosomes) chromosomes, whose true segmentation results are zero everywhere, to analyze real noise in this data source. The Snijders 2001 array [37] is from Stanford University with 15 human cell lines. One chromosome in this data just contains around one hundred of probes. The Bredel 2005 array [38] data is from Harvard Medical School. This data includes 26 samples, and each sample has thousands of probes. Many normal chromosomes of the same sample are combined together, because the number of probes of one chromosome in these data is not enough for noise analysis (fitting noise model). The Smith 2007 data, whose platform is Agilent-015366 Custom Human 244K CGH Microarray, includes three control self-self hybridization samples, and each sample has twenty-three chromosomes. From this data, there are 69 chromosomes with ten-thousand probes each to study real noise model because their true signals are expected to be zero everywhere. In the Nicolas 2009 [40] whose platform is Custom Nimblegen array CGH chip targeted to canFam2 segmental duplications, there are 23 chromosomes in self-test sample (GSM334824) for noise analysis.

3.2.2 Distribution Noise Candidates in Array CGH

After these five datasets are analyzed, the noise distribution of array CGH is bell-shaped and symmetric. Two samples are shown in Fig. 3.1. They are chromosome 15 of GSM232967 in Fig. 3.1(a) from the Lee 2008 array, and chromosome 13 of GSM215042 in Fig. 3.1(c) from the Smith 2007 array. Fig. 3.1(b) and Fig. 3.1(d) are empirical histograms of two above signals. These two histograms are bell-shaped and symmetric. There are four probability distribution candidates for these noise models such as Gaussian distribution, generalized Gaussian distribution (GGD), Student's

t distribution, and Cauchy distribution. There is another bell-shaped distribution, extreme value distribution but it is not symmetric.

Fig. 3.2 shows four distribution candidates. Gaussian distribution with zero mean, shown in Fig. 3.2(a), has PDF as follows

$$p(x;\sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-x^2/(2\sigma^2)}, \tag{3.1}$$

where σ is the standard deviation. Fig. 3.2(c) represents the PDF of Student's t distribution as follows

$$p(x;\nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}(1+\frac{x^2}{\nu})^{-(\frac{\nu+1}{2})}, \tag{3.2}$$

where ν is the number of degrees of freedom and Γ is the Gamma function. The last distribution candidate is Cauchy distribution, shown in Fig. 3.2(d), with PDF as follows

$$p(x;\gamma) = \frac{1}{\pi\gamma[1+(\frac{x^2}{\gamma})]}, \tag{3.3}$$

where γ is the scale parameter.

GGD, shown in Fig. 3.2(b), will be discussed next. The probability density function (PDF) of a generalized Gaussian random variable x, with zero mean, is defined as

$$p(x;\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} e^{-(|x|/\alpha)^\beta}, \tag{3.4}$$

where $\Gamma(.)$ is the Gamma function, $\Gamma(z) = \int_0^\infty e^{-t}t^{z-1}dt$, with $z > 0$. Here α is the standard deviation, while β is inversely proportional to the decreasing rate of the peak. α is referred to the scale parameter and β is called the shape parameter. The Gaussian and Laplacian PDF are only special cases of GGD at $\alpha = 2$ and $\alpha = 1$, respectively. If compared to three other distributions, GGD can fit the data with sharp peak and heavy-tail better. Thus, GGD is proposed to use to capture noise of array CGH data. The parameters α and the β can be estimated as in [41].

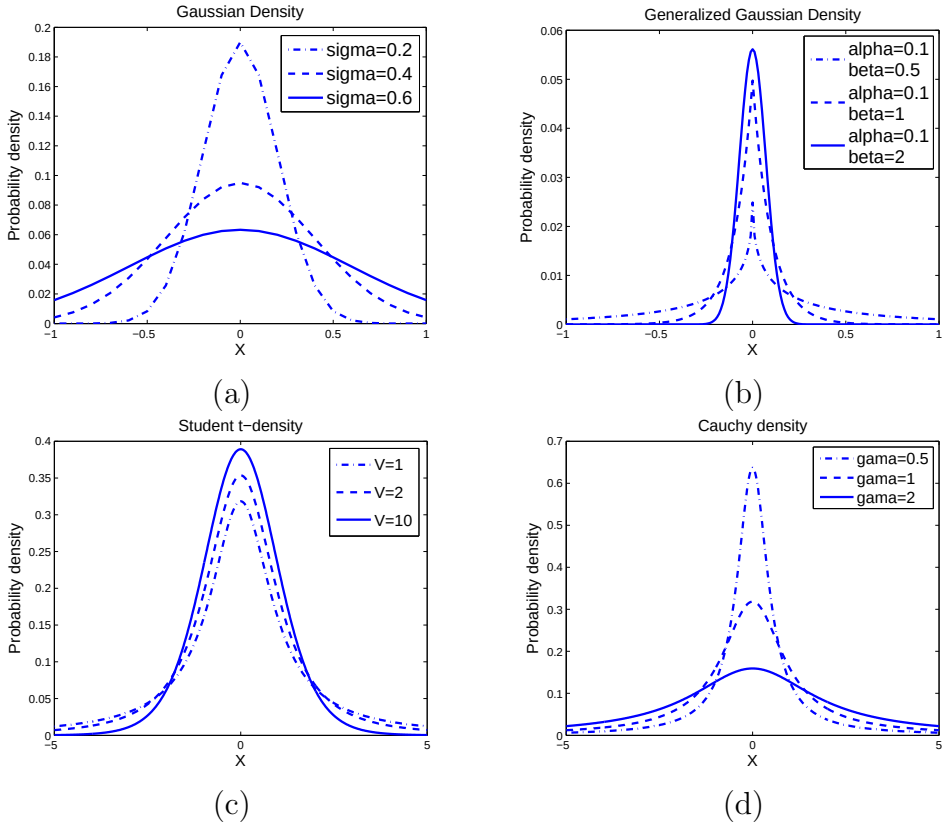


Figure 3.2. Four probability density candidates with zero mean: (a) Gaussian density with $\sigma = 0.2, 0.4$ and 0.6; (b) Generalized Gaussian density with $\alpha = 0.1, \beta = 0.5, 1,$ and 2; (c) Student's t density with $\nu = 1, 2$ and 10; (d) Cauchy density with $\gamma = 0.5, 1$ and 2.

In probability theory or information theory, Kullback-Leibler Divergence (KLD) is commonly used to measure the difference between two probability distributions. The following definition is used to calculate KLD between real probability distribution P and estimated probability distribution Q, .

$$KLD(P||Q) = \sum_i P(i) \log_{10} \frac{P(i)}{Q(i)}. \tag{3.5}$$

The entropy of distribution P can be calculated by:

$$H(P) = \sum_i P(i) \log_{10}(P(i)). \tag{3.6}$$

From Eq. (3.5) and Eq. (3.6):

$$\frac{KLD(P||Q)}{H(P)} = \frac{H(P,Q) - H(P)}{H(P)} = \frac{\Delta H}{H}, \tag{3.7}$$

where $\Delta H = H(P,Q) - H(P)$. $\frac{KLD(P||Q)}{H}$ or $\frac{\Delta H}{H}$ can be used to see how the estimated probability distribution Q fits to real probability distribution P. $\frac{\Delta H}{H}$ is also called as relative entropy. The fitting between P and Q is better when $\frac{\Delta H}{H}$ is smaller.

3.2.3 Validation of New Array CGH Data Noise Model

In this part, four candidates including Gaussian, GGD, student's t, and Cauchy are employed to fit noise distribution. The real noise of array CGH signal is obtained from five sources: Lee 2008 array [36], Snijders 2001 array [37], Bredel 2005 array [38], Smith 2007 array [39] and Nicolas 2009 array [40]. To estimate the parameters of Gaussian, Student's t, and Cauchy models, the nonlinear curve-fitting method is used. $\Delta H/H$ (relative entropy) between each model and empirical noise PDF can be calculated by Eq. (3.7). This $\Delta H/H$ value represents the difference between two PDFs. A model fits an empirical PDF better than another one if its $\Delta H/H$ is smaller.

Two examples of fitting models are shown in Figs. 3.3 and 3.4. Histogram of GSM232967's chromosome 15 and fitting results in Fig. 3.3 illustrate that the difference between GGD model and empirical noise PDF is much less than that of other models. KLD between GGD model and empirical PDF is 0.0061, while KLDs between Gaussian, student's t, Cauchy model and empirical PDF are 0.0155, 0.0135 and 1.7583, respectively. This result also agrees with chromosome 13 of GSM215042 in Fig. 3.4. Relative entropy $\Delta H/H$ of GGD is 0.0108, smallest, while KLDs of Gaussian, Student's t and Cauchy are 0.0339, 0.0251 and 0.4692, respectively. From the above fitting results, the relative entropy $\Delta H/H$ between GGD and noise histogram is always smallest. GGD outperforms other distributions in fitting evaluations. Both

Table 3.2. Average $\Delta H/H$ of five distributions. Lee 2008 array includes 60 samples. Snijders 2001 array includes 15 samples. Bredel 2005 array includes 26 samples. Smith2007 array includes 69 samples. Nicolas2009 array has 23 samples

Data	*Gaussian*	*GGD*	*Student t*	*Cauchy*
Lee 2008	0.0200	0.0083	0.0172	0.8846
Snijders 2001	0.0471	0.0216	0.0252	0.3154
Bredel 2005	0.0846	0.0227	0.0588	0.5770
Smith 2007	0.0298	0.0184	0.0259	0.7997
Nicolas 2009	0.0311	0.0243	0.0461	0.4238

student's t distribution and GGD fit better than Gaussian model. Therefore, noise in array CGH is really heavy-tailed. This result also agrees with the conclusion of paper [33] in which array CGH data noise is a highly non-Gaussian with heavy tail.

Relative entropy $\Delta H/H$ is also calculated between each model and individual empirical PDF of 40 arrays from Lee 2008 array, 15 arrays from Snijders 2001 array, 26 arrays from the Bredel 2005 array, 69 arrays from the Smith 2007 array and 23 arrays from Nicolas 2009. Then the average KLDs between each model and each data source are calculated as shown in Table. 3.2. The difference between the GGD model and noise PDF is the smallest in all data sources with many platforms. Compared with Gaussian, student't and Cauchy models, GGD model is more accurate and sufficient for fitting empirical noise PDF in the array CGH data than the others. It can capture behaviors of the real noise PDF. Therefore, GGD model is proposed as a new noise model assumption in the array CGH data. Then, a smoothing algorithm will be developed based on this GGD noise model.

3.3 Proposed Methods

How to reduce heavy-tailed noise in array CGH and how to detect breakpoints of array CGH data are two problems which will be solved in this section. Generalized

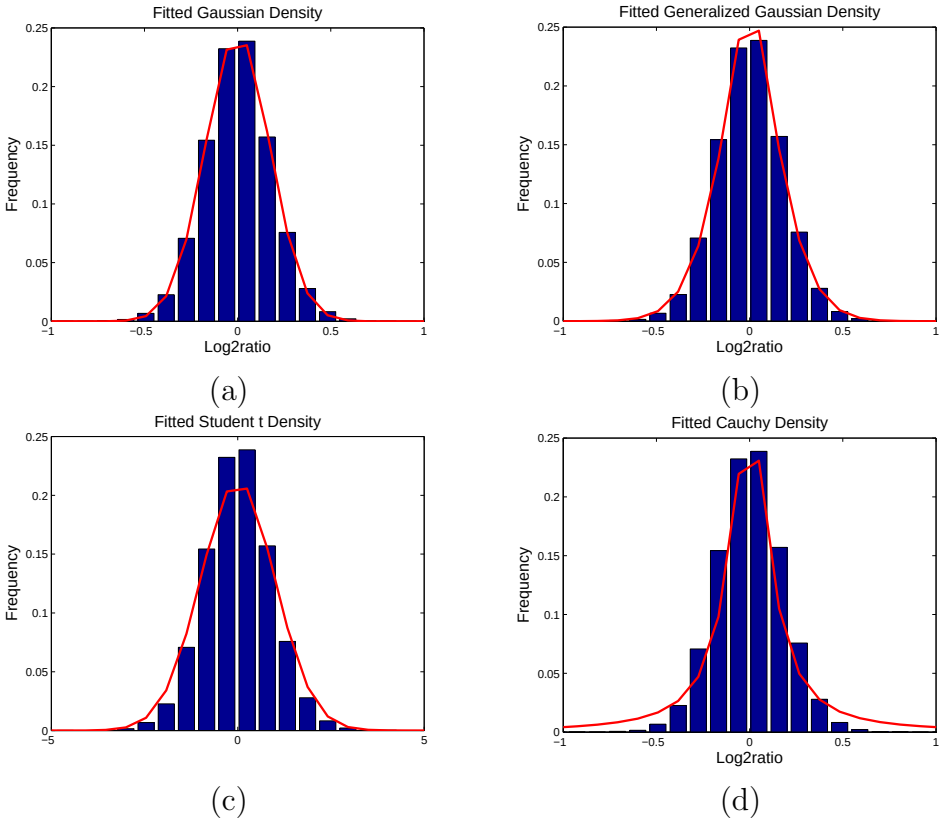


Figure 3.3. The relative entropy between the histogram of the chromosome 15 of GSM232967 and four distribution candidates such as (a) Gaussian: $\Delta H/H = 0.0155$, (b) Generalized Gaussian: $\Delta H/H = 0.0061$, (c) student's t: $\Delta H/H = 0.0135$, and (d) Cauchy: $\Delta H/H = 1.7583$.

Gaussian bivariate shrinkage function based de-noising procedure in wavelet domain will be mentioned in the first sub-section. In the second sub-section, wavelet derivative scalogram in 1-D will be defined to detect breakpoints which mark changing points of segments in array CGH. In the last sub-section, the main method which is combination of heavy-tailed noise suppression and wavelet pattern-matching for breakpoint detection will be proposed.

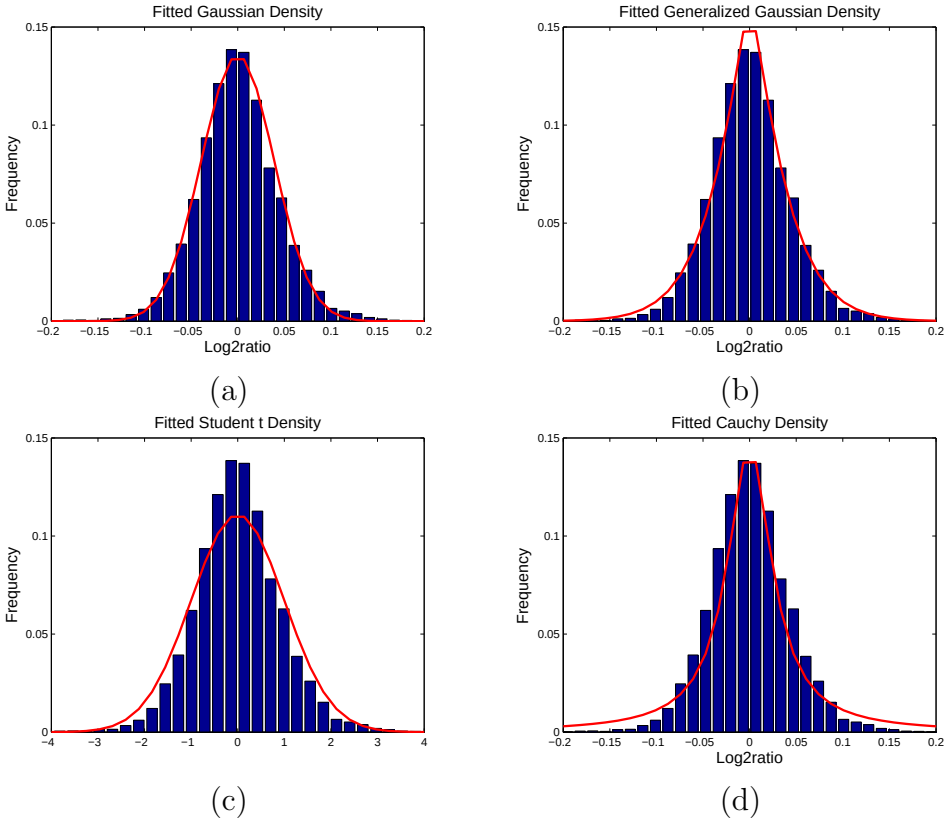


Figure 3.4. The relative entropy between the histogram of the chromosome 13 of GSM215042 and four distribution candidates such as (a) Gaussian: $\Delta H/H = 0.0339$, (b) Generalized Gaussian: $\Delta H/H = 0.0108$, (c) student's t: $\Delta H/H = 0.0251$, and (d) Cauchy: $\Delta H/H = 0.4692$.

3.3.1 Heavy-Tailed Noise Suppression

As discussed in section "Array CGH Noise Characteristic", generalized Gaussian should be a better noise assumption than Gaussian or the others. With this new noise assumption, de-noising becomes a challenging problem. According to comparison in [8], with Gaussian noise assumption, Wave [9] in which SWT and hard thresholding are used is a really good method. So, SWT will be continued to use for noise reducing in this chapter. However, the hard threshold based estimator is replaced by a new estimator which is designed to operate with heavy-tailed noise.

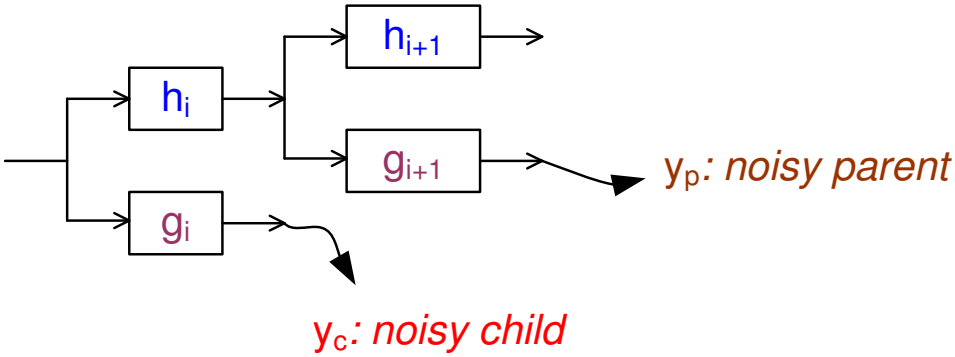


Figure 3.5. The position of child and parent coefficients in stationary wavelet domain.

Huang *et al.*[42] and Nguyen *et al.*[43] applied that function in the complex wavelet transform domain to recover array CGH data successfully and got promising results. However, noise here is assumed as generalized Gaussian. So, the idea in [1] will be imitated to build new algorithm for new noise in SWT. Generally the array CGH data Y can be obtained by:

$$Y = D + \mathcal{N}, \tag{3.8}$$

where D is the deterministic signal and $\mathcal{N}$ represents for generalized Gaussian noise which has distribution as Eq. (3.4). The coefficients $\mathbf{y}_k$ can be obtained by decomposing the data Y with the SWT and those coefficients can be formulated as

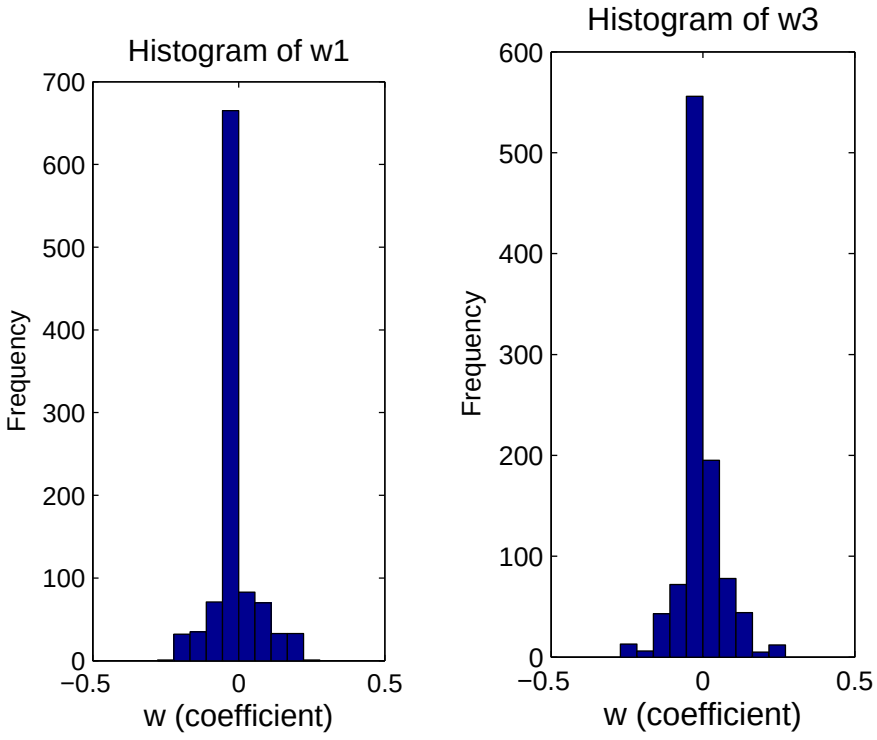
$$\begin{aligned} \mathbf{y_1} &= \mathbf{w_1} + \mathbf{n_1}, \\ \mathbf{y_2} &= \mathbf{w_2} + \mathbf{n_2}, \end{aligned} \tag{3.9}$$

where $\mathbf{y_1}$ and $\mathbf{y_2}$ are noisy wavelet coefficients, $\mathbf{w_1}$ and $\mathbf{w_2}$ are true coefficients. The noise pdf should be followed as

$$p_{\mathbf{n}}(\mathbf{n}) = K(\alpha,\beta)\exp(-\frac{|n_1|^\beta + |n_2|^\beta}{\alpha^\beta}), \tag{3.10}$$

where $K(\alpha,\beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)}$. The standard MAP estimator [1] of $\mathbf{w}$ from $\mathbf{y}$ is followed as

$$\widehat{\mathbf{w}}(\mathbf{y}) = \arg\max_{\mathbf{w}}[\log_e(p_{\mathbf{n}}(\mathbf{y}\text{-}\mathbf{w})) + \log_e(p_{\mathbf{w}}(\mathbf{w}))]. \tag{3.11}$$



(a) (b)

Figure 3.6. The histograms computed from true array CGH signals. (a) Histogram of w_1 , (b) Histogram of w_2.

Fig. 3.6 illustrates the histogram of $\mathbf{w_1}$ (child) and $\mathbf{w_2}$ (parent). The $\mathbf{w_1}$ and $\mathbf{w_2}$ are computed from array CGH data without noise by using the SWT. Fig. 3.7 shows the joint distribution of $\mathbf{w_1}$ and $\mathbf{w_2}$. Two pdfs are being proposed to fit the joint distribution.

The imitated idea [1] is used to propose a non-Gaussian bivariate pdf for $\mathbf{w_1}$ and $\mathbf{w_2}$ as

$$p_{\mathbf{w}}(\mathbf{w}) = \frac{3}{2\pi\sigma^2}\exp(-\frac{\sqrt{3}}{\sigma}\sqrt{|w_1|^2+|w_2|^2}). \tag{3.12}$$

This pdf (3.12) is sketched in Fig. 3.8. With this pdf, two variables $\mathbf{w_1}$ and $\mathbf{w_2}$ are really dependent. Let us define

$$f(w) = \log_e(P_w(w)) = \log_e(\frac{3}{2\pi\sigma^2}) - \frac{\sqrt{3}}{\sigma}\sqrt{|w_1|^2+|w_2|^2}. \tag{3.13}$$

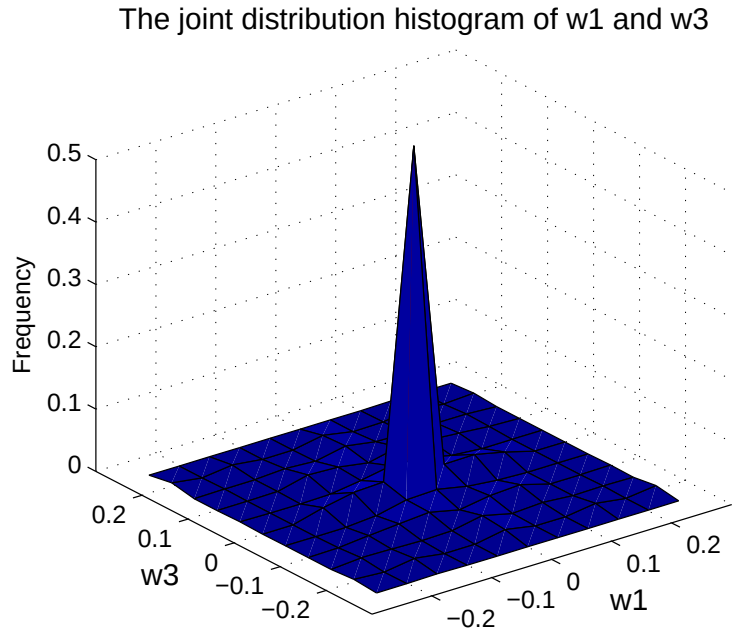


Figure 3.7. The joint distribution of w_1 and w_2 which are created from decomposition of true array CGH signal.

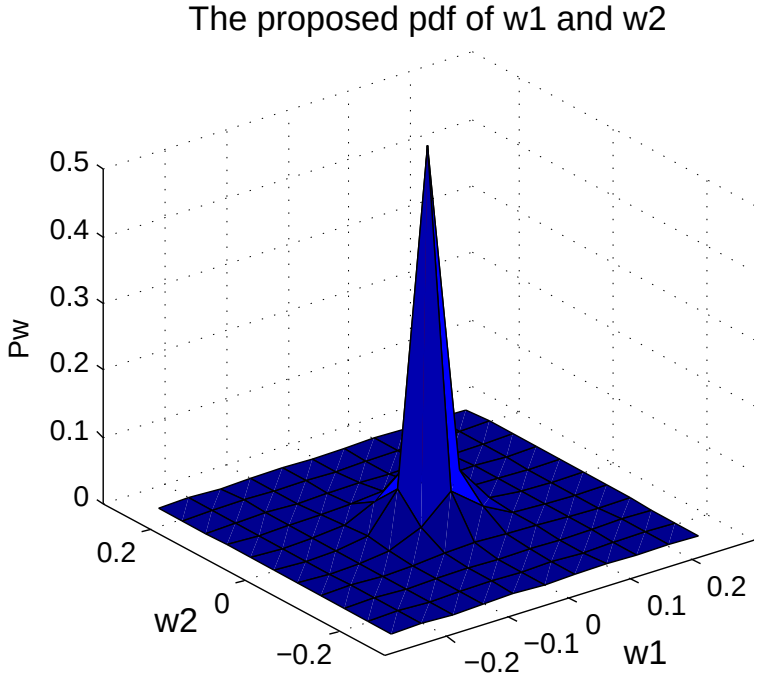


Figure 3.8. The proposed pdf with two variables: w_1 and w_2.

By using Eq.(3.10), Eq.(3.11) becomes:

$$\widehat{\mathbf{w}}(\mathbf{y}) = \arg\max_{\mathbf{w}}[\log_e(K(\alpha,\beta)) - \frac{|y_1 - w_1|^\beta + |y_2 - w_2|^\beta}{\alpha^\beta} + f(w)]. \tag{3.14}$$

Solving Eq.(3.14) is equivalent to solving two following equations:

$$sign(y_1 - w_1) \times \beta \times \frac{|y_1 - w_1|^{\beta-1}}{\alpha^\beta} = \frac{\sqrt{3}w_1}{\sigma\sqrt{|w_1|^2 + |w_2|^2}}, \tag{3.15}$$

$$sign(y_2 - w_2) \times \beta \times \frac{|y_2 - w_2|^{\beta-1}}{\alpha^\beta} = \frac{\sqrt{3}w_2}{\sigma\sqrt{|w_1|^2 + |w_2|^2}}. \tag{3.16}$$

If this is Gaussian noise ($\beta = 2$ and ${\sigma_n}^2 = \frac{\alpha^2}{2}$), according to [1], the solutions can be formulated as

$$\hat{w_1}(\beta = 2) = \frac{(\sqrt{|y_1|^2 + |y_2|^2} - \frac{\sqrt{3}{\sigma_n}^2}{\sigma})_+}{\sqrt{|y_1|^2 + |y_2|^2}}.y_1, \tag{3.17}$$

$$\hat{w_2}(\beta = 2) = \frac{(\sqrt{|y_1|^2 + |y_2|^2} - \frac{\sqrt{3}{\sigma_n}^2}{\sigma})_+}{\sqrt{|y_1|^2 + |y_2|^2}}.y_2, \tag{3.18}$$

where $()_+$ is defined by

$$(u)_+ = \begin{cases} 0 & \text{if } u < 0, \\ u & \text{otherwise.} \end{cases} \tag{3.19}$$

In Eq. (3.17) and Eq. (3.18), σ can be estimated by

$$\hat{\sigma} = \sqrt{(\hat{\sigma}_y^2 - \hat{\sigma}_n^2)_+}, \tag{3.20}$$

where $\hat{\sigma}_n$ is the noise deviation which is estimated from the finest scale wavelet coefficients by using a robust median estimator [24] as follows

$$\hat{\sigma}_n^2 = \frac{\text{median}(|y_i|)}{0.6745}. \tag{3.21}$$

$\hat{\sigma}_y$ is the deviation of observation signal estimated by

$$\hat{\sigma}_y^2 = \frac{1}{M}\sum_{y_i \in N(i)} |y_i|^2, \tag{3.22}$$

where M is the size of the neighborhood $N(i)$.

The successive substitution method can be used to get solution in general case of β.

Step 1 Initialize $\hat{w_1}^{[0]} = \hat{w_1}(\beta = 2)$ and $\hat{w_2}^{[0]} = \hat{w_2}(\beta = 2)$ at $k = 0$

Step 2 Calculate ${r_1}^{[k]}$ and ${r_2}^{[k]}$ using

$$ {r_1}^{[k]} = \sqrt{(\hat{w_1}^{[k]})^2 + (\hat{w_2}^{[k]})^2} \times |y_1 - \hat{w_1}^{[k]}|^{(\beta-1)} \times sign(y_1 - \hat{w_1}^{[k]}). \tag{3.23} $$

$$ {r_2}^{[k]} = \sqrt{(\hat{w_1}^{[k]})^2 + (\hat{w_2}^{[k]})^2} \times |y_2 - \hat{w_2}^{[k]}|^{(\beta-1)} \times sign(y_2 - \hat{w_2}^{[k]}). \tag{3.24} $$

Step 3 Find $\hat{w_1}^{[k+1]} = \dfrac{\beta\sigma {r_1}^{[k]}}{\alpha^\beta\sqrt{3}}$ and $\hat{w_2}^{[k+1]} = \dfrac{\beta\sigma {r_2}^{[k]}}{\alpha^\beta\sqrt{3}}$

Step 4 Find the differences $\epsilon_1 = \hat{w_1}^{[k+1]} - \hat{w_1}^{[k]}$ and $\epsilon_2 = \hat{w_2}^{[k+1]} - \hat{w_2}^{[k]}$

Step 5 If both ϵ_1 and ϵ_2 are small, then terminate the iteration. Otherwise, set $k = k+1$ and go to step 2.

3.3.2 One-Directional Derivative Wavelet Scalogram

After noise suppression step, beakpoints in processed array CGH will be detected by 1D scalogram whose theory is studied as the following section. True array CGH is considered as a mixture of step functions as follows

$$ f(t) = \sum_i^N f_i(t) = \sum_i^N A_i \times u(t - t_{0i}). \tag{3.25} $$

The continuous wavelet transform can be written as a convolution product in Eq. 3.26:

$$ Wf(u,s) = \int_{-\infty}^{+\infty} f_i(t) \frac{1}{\sqrt{s}} \Psi^\star(\frac{t-u}{s}) dt, \tag{3.26} $$

where $\star$ is the conjugate. According to [44], the wavelet transform in Eq. 3.26 can be rewritten as a multi-scale differential operator in Eq. 3.27

$$ W_n f(u,s) = s^n \frac{d^n}{du^n}(f_i * \bar{\theta}_s(t))(u), \tag{3.27} $$

where $*$ is convolution. In HaarSeg method [45], the simple derivative wavelet (Haar filters) has been used. The result in HaarSeg method is really promising because of not only segment result but also algorithm speed. However, Haar wavelet is so sensitive with heavy-tailed noise. In this paper, Gaussian wavelet is used instead of Haar wavelet to make sure that our method is robust with noise. So, $\bar{\theta}_s(t)$ can be written as follows

$$\bar{\theta}_s(t) = \frac{1}{\sqrt{s}} exp(-\frac{t^2}{s^2}). \tag{3.28}$$

Taking convolution $f_i * \bar{\theta}_s$, one gets

$$Wf(u,s) = A_i \times \int_{t_{0i}}^{+\infty} \frac{1}{\sqrt{s}} e^{\frac{-(t-u)^2}{s^2}} dt, \tag{3.29}$$

$$W_1 f(u,s) = -A_i \times \sqrt{s} \times e^{\frac{-(u-t_{0i})^2}{s^2}}. \tag{3.30}$$

$W_1 f(u,s)$ gets maximum at $u = t_{0i}$. The scalogram in 2D is obtained by:

$$WS(u,s) = 100 \times \frac{(\frac{W_1 f(u,s)}{\sqrt{s}})^2}{\sum_{i=1}^{N} (\frac{W_1 f(u,s)}{\sqrt{s}})^2}. \tag{3.31}$$

However, breakpoint detection using wavelet pattern-matching could not be finished easily in 2D scalogram. So, scalogram in 2D will be transformed into 1D by using two following steps. In the first step, ridge lines [44] are identified by linking the local maxima of 2-D scalogram at each scale level. L_R and $\mathcal{U}(u)$ represent linking line length and a vector including linked maxima position with position u at scale one. Also in this step, ridge line whose length is smaller than a certain threshold should be reset to zero. Step one can be formulated as

$$\mathcal{U} = \begin{cases} 0 & \text{if } L_R < \text{threshold}, \\ u_1 \ u_2 \ ...u_{s_{max}} & \text{otherwise.} \end{cases} \tag{3.32}$$

In this chapter, threshold is selected as 32. In the second step, 1-D scalogram is built as follows

$$WS_{1D}(\overline{u}) = \begin{cases} 0 & \text{if } \mathcal{U} = 0, \\ \sum_{u \in \mathcal{U}} WS(u,s) & \text{otherwise,} \end{cases} \tag{3.33}$$

where $\overline{u} = \overline{\mathcal{U}(u)} = \frac{u_1 + u_2 + ... + u_{s_{max}}}{s_{max}}$

3.3.3 Derivative Wavelet Scalogram based Segmentation Method

Derivative wavelet scalogram based segmentation (DWSS) proposed in this chapter is based on two steps as follows

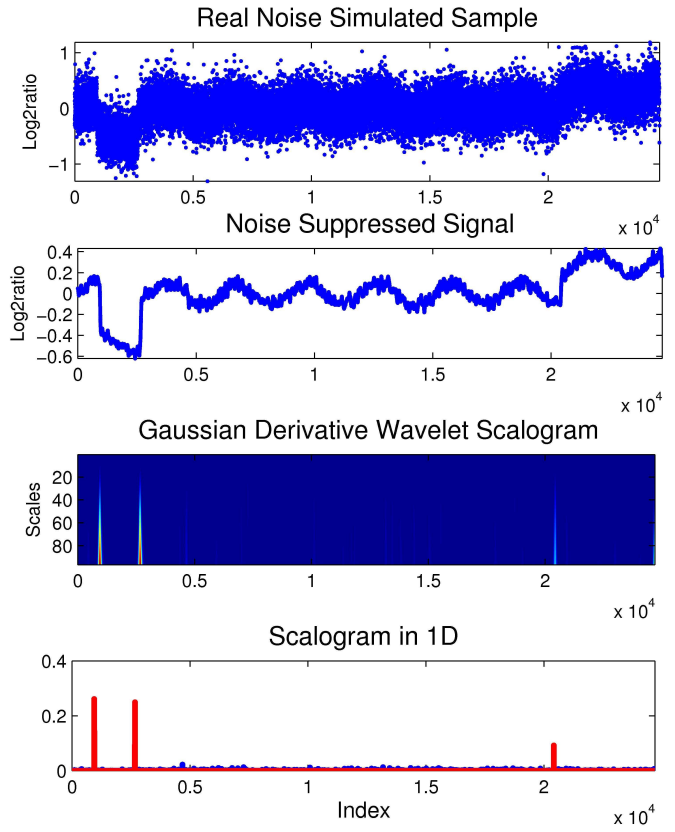
Step 1: Noise Suppression

First, heavy-tailed noise in array CGH signal will be removed by generalized Gaussian bivariate shrinkage function in stationary wavelet domain. After array CGH is decomposed by SWT, noise suppression should be done by five steps in Section "Heavy-Tailed Noise Suppression". Just high frequency scales should be applied to remove noise. Approximation scales are kept to make sure that true signal should not be removed in de-noising process. After that, noise suppressed array CGH signal is obtained back by taking inverse SWT.

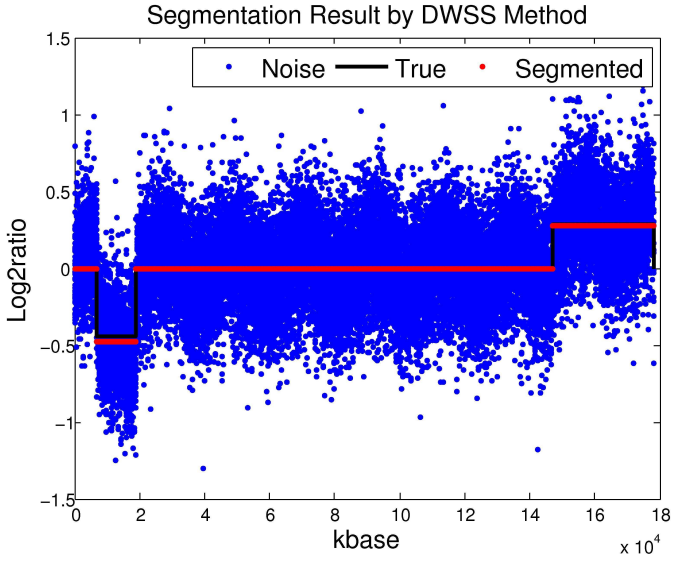
Step 2: Breakpoint Detection

1-D Gaussian derivative wavelet scalogram is used to detect breakpoints in noise suppressed array CGH. Mean value of processed signal in each segment will be considered as log2ratio of that segment.

Fig 3.9(a) illustrates two steps of DWSS. This real noise simulated sample is just removed outlier. Noise in that sample is also reduced but there is still noise inside signal to keep true signal and make algorithm run faster. Gaussian derivative wavelet scalogram in 2D is built from denoised signal. 96 scales are used to build



(a)



(b)

Figure 3.9. One example of DWSS method. (a) Demo of two steps in DWSS: this sample is real noise simulated array CGH with bias of 0.2. Noise suppression step and breakpoint detection step are illustrated. (b) Segment Result by DWSS on one sample: DWSS detects exactly four segments (two normals, one deletion and one gain) in this sample.

this scalogram. Even though footprints of breakpoints are visible in 2-D scalogram, breakpoints could not be detected easily. So, scalogram in 1D is defined and used to detect breakpoints. 1D scalogram has large value at positions corresponding to breakpoints in array CGH in the fourth sub-figure in Fig 3.9(a). The final segmentation result of DWSS in this example is shown in Fig 3.9(b) in red points. The black line represents true signal and blue points are noise simulated data. Evaluation of the proposed method will be discussed in next section.

3.4 Results

In this section, how to improve array CGH synthetic data model will be introduced. Next, a comparison of the proposed method to previous works by using RMSE and ROC curves will be mentioned. In conclusion, DWSS works robustly and accurately with heavy-tailed noise and probe hybridization bias.

3.4.1 Improved Synthetic Data Model

Synthetic array CGH data is very important for array CGH study and algorithm evaluation. Because the ground truth of array CGH aberration regions is known in synthetic array CGH data, the different smoothing or segmentation algorithms' performance can be measured. However if the synthetic array CGH data model cannot correctly represent the natural properties of real array CGH data, the evaluation results based on them will mislead the array CGH studies. So far, the most commonly used synthetic array CGH data model was proposed by Willenbrock and Fridlyand [15] in 2005. They segmented a primary tumor dataset of 145 samples using DNA copy number levels from the empirical distribution of segment mean values. They got results such as copy number probabilities and the distributions of segment length. The expected $log2ratio$ for each clone was computed as $\log_2(\frac{cP_t+2(1-P_t)}{2})$ where c was

Table 3.3. Estimated values for the parameters α, β of real array CGH noises

Data Source	α	β
Lee 2008 (40 samples)	$0.1998 \rightarrow 0.3032$	$1.165 \rightarrow 1.9109$
Smith 2007 (69 samples)	$0.1221 \rightarrow 0.2010$	$1.0538 \rightarrow 1.7342$
Nicolas 2009 (23 samples)	$0.2547 \rightarrow 0.3032$	$1.7841 \rightarrow 2.3764$

the assigned copy number with P_t is a proportion of tumor cells whose value is from a uniform distribution between 0.3 and 0.7. The probe hybridization bias proposed in [34] is also added to true signal.

$$Y = D + \mathcal{R} + \mathcal{N}, \tag{3.34}$$

where D is true signal, $\mathcal{R}$ is hybridization bias and $\mathcal{N}$ is noise. b can be used as a ratio of from zero to one to change bias value in simulated data.

$$\mathcal{R} = b \times (0.5 \sin(2\pi 0.001m) + \mathcal{N}(0, 0.25)), \tag{3.35}$$

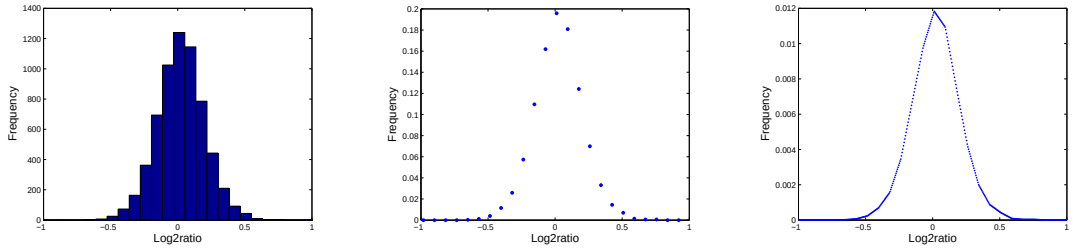
Where m is the length of simulated signal, b is the bias value. Following this standard model, true array CGH signal without noise can be used. In order to improve their model [15] in which Gaussian noise is used, two different types of noises (GGD noise and real noise) are proposed to add to true signal to create two new synthetic array CGH data models.

3.4.1.1 GGD Noise

As discussed in the previous section, GGD fits very well noise PDF in the array CGH data. Parameters α and β are estimated as shown in table 3.3. From table. 3.3, the parameter α ranges from 0.12 to 0.3 and the parameter β ranges from 1.05 to 2.38. Therefore, GGD noise model with α and β values as Table. 3.3 should be used for synthetic array CGH data generation.

3.4.1.2 Real Noise

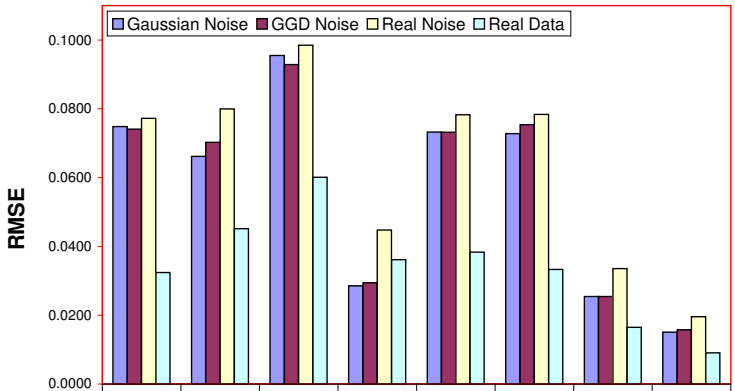
The real noise is extracted from array CGH data as following steps. First, histogram of a real array CGH data containing noise only is shown in Fig. 3.10 (a). From this histogram, a discrete PDF with number of bins of 64 is formed as Fig. 3.10 (b). Then the 64 bins-PDF is interpolated and new PDF is normalized to get with some thousands of bins as Fig. 3.10 (c). Finally a new random noise vector will be yielded from this PDF. In the experiments, 40 arrays of the Lee 2008 array [36], 69 arrays of the Smith 2007 array [39] and 23 arrays of Nicolas 2009 array [40] are used with ten thousands probes which contain noise only. Therefore, there are 132 high standard deviation PDFs to create thousands of random noise vector which are added to true signal to create real noise simulated array CGH data. One example to illustrate how to get real noise is shown in Fig. 3.10.



(a) (b) (c)

Figure 3.10. The procedure to create real noise from chromosome 19 of GSM232967 (a) Histogram with number of bins of 64 (b) PDF number of bins of 64 (c) PDF number of bins of 1024.

After noise is added, unequally spaced probes [13] are created. The intuition of this step is that the distances between probe k and probe $k+1$ are randomly and the best way to get these distances from the real array CGH data, such as Lee 2008 array [36] for high resolution. Then unequally spaced probes on chromosomes are



	Lowess	Wave	SmoothSeg	HaarSeg	CBS	GADA1	GADA2	DWSS
Gaussian Noise	0.0749	0.0662	0.0955	0.0285	0.0733	0.0728	0.0254	0.0151
GGD Noise	0.0741	0.0703	0.0929	0.0294	0.0732	0.0754	0.0254	0.0157
Real Noise	0.0772	0.0800	0.0985	0.0448	0.0783	0.0784	0.0336	0.0196
Real Data	0.0324	0.0452	0.0601	0.0361	0.0383	0.0333	0.0165	0.0091

Figure 3.11. Summary result: average RMSEs of all testing methods on three simulated data-sets and real data set are shown in this table. Eight methods in this table are Lowess, Wave, SmoothSeg, HaarSeg, CBS, GADA1, GADA2 and DWSS methods. Different bias levels are tested with each simulated data-set and average RMSEs are obtained. Three real data-sets are used and also average RMSEs are obtained.

placed. Number of probes can be low, high and very high. Now, there are many artificial chromosomes of length 200 Mbase with three resolutions and two kinds of noise including generlized Gaussian noise and real noise.

3.4.2 Performance Evaluation of DWSS Method

DWSS method is compared to other most commonly used methods in literature, such as Lowess [8], Wave [9], Smoothseg [23], HaarSeg [45], CBS [32], GADA1 [35] and GADA2 [34]. R package DNAcopy version 1.14 for CBS, R package smoothseg for smoothseg, and R package waveslim for Wave are used. With Lowess, HaarSeg, GADA1, GADA2 and the proposed method, Matlab has been used. HaarSeg, GADA1 and GADA2 's implement are downloaded from sharing link in [45] and in [34].

3.4.2.1 RMSE Comparison

One thousand high resolution simple chromosomes are created with three kinds of noise such as Gaussian, GGD and real noise. The root mean square errors (RMSEs) of all methods are calculated and shown in Table. 3.11. With these simulated data, the DWSS method has the best performance. The DWSS outperforms averagely the Lowess by 78%, the Wave by 77%, the Smoothseg by 82%, the HaarSeg by 51%, the CBS by 78%, the GADA1 by 78% and the GADA2 by 40% in terms of the RMSEs. For all noises, the DWSS consistently achieves much better results than the others.

Real array CGH data is also used to evaluate performances of eight methods. Lee 2008 array [36] array including 40 samples, Smith2007 array including 69 samples and Nicolas 2009 including 23 samples are three real data with the known ground truth. In table.3.11, the performance of DWSS method is much better than that of the others. Average RSME of the proposed method is smaller than SmoothSeg by 5.7, Lowess, CBS and GADA1 by 4.5 times, Wave by 4.3 times, HaarSeg by 2 times and GADA2 by 1.7 times.

In general, when both simulated and real data are considered, the proposed method improved 41% to 77% when compared to previous methods.

3.4.2.2 ROC Curve Comparison

A comparison of array CGH detection algorithms was studied by Lai *et. al.* [8]. They used the ROC curve to evaluate 11 algorithms with aberration widths of 5, 10, 20 and 40, and signal-to-noise ratios (SNRs) of 1, 2, 3 and 4. Many synthetic chromosomes consisting of 100 probes are created from four templates with Gaussian noise and square-wave signal at the center of chromosome . In 2007, Huang *et al.* [23] improved this setting by decreasing the width of the center square-wave and increasing

the noise level. In [45], 2008, Ben and Eldar proposed using very high resolution data and real noise to improve the quality of evaluation. In this section, the performance of all methods will be evaluated not only at the middle of signal but also at the border of signal. Therefore, four templates with the aberration at the center and four more templates with the aberration at the border are used. The aberration widths used in this section are 5%, 10%, 15% and 20% of whole chromosome length. Both Gaussian and real noise are used to evaluate all methods. The real noise from forty self-self test arrays of Lee 2008 array [36] is also used to add to these templates. In all cases, bias of 0.8 will be added to make problem harder. Using all eight genomic templates, 100 noise array CGH arrays are generated with unequally spaced probes. Segmentation performances of all methods are tested on three true segment amplitudes of $log_2\frac{3}{2}$, $log_2\frac{4}{2}$ and $log_2\frac{5}{2}$. In Fig. 3.12, when real noise simulated data which has gain segmentation amplitude of $log_2\frac{5}{2}$ is used, the performances of DWSS and GADA2 are the best. HaarSeg and CBS work well and their ROC curves are very close together. Wave method also work well. The next ones are GADA1, SmoothSeg and Lowess. With Gaussian noise (not shown here), except for Lowess' performance, all methods' ROC curves are very close together and perfect. Simulated data makes more difficult to segment in Fig. 3.13. The gain segment amplitude is reduced to $\log_2\frac{4}{2}$ and $\log_2\frac{3}{2}$. With copy of four in Fig. 3.13 (b)(d), DWSS and HaarSeg still work well. Performance of GADA2 gets worse fast and is worse than DWSS and HaarSeg's ones. With copy of three in Fig. 3.13 (a)(c), most of methods get worse. However, DWSS is still the best method. The next ones are HaarSeg and GADA2. These results confirmed above results when RMSEs are used.

When both RMSEs and ROC curves are used, DWSS has the best performance. GADA2 and HaarSeg are good methods being robust with bias. GADA2 is more robust with heavy-tailed than HaarSeg. HaarSeg detects segments which have small

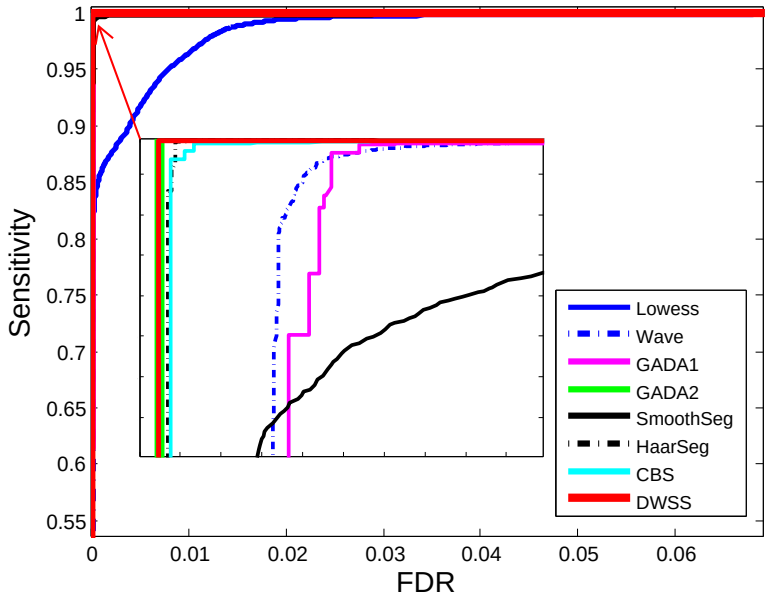


Figure 3.12. One hundred real noise simulated samples which true abnormal segments have amplitude of $\log_2 \frac{5}{2}$ are used. Bias of 0.8 is used in this case. ROC curves are obtained from arrays which are generated from 8 genomic templates and real noise source. DWSS is compared to other most common used array CGH algorithms such as Lowess, Wave, SmoothSeg, HaarSeg, CBS, GADA1 and GADA2. The performances of DWSS and GADA2 are the best. HaarSeg and CBS also work well and their ROC curves are very close together.

Signal-Noise ratio better than GADA2. GADA1 and CBS are comparable. About algorithm speed, DWSS runs faster than GADA2 by 1.6 times and CBS by 7.4 times.

3.5 Discussion

In this section, eight methods will be discussed more. Two concepts which will be used to study these methods are robustness with heavy-tailed noise and sensitivity with hybridization bias. Pros and cons of seven previous methods will be studied. First, based on table. 3.4 and two figures (Fig. 3.12 and Fig. 3.13), eight methods are tested with Gaussian and heavy-tailed noise. In this testing, biases of 0.2, 0.4 and 0.8 are used.

Lowess method [8] is robust with heavy-tail but so sensitive with bias. In general, performance of Lowess is not really good. Wave method [9] should be used with Gaussian noise and without bias because of its' nice result and fast speed. Wave method is designed for Gaussian noise so it is not robust with heavy-tailed noise and it is also sensitive with bias. With small SNR, Wave method's performance is comparable with that of CBS and GADA1 methods. SmoothSeg method [23] is designed for heavy-tail noise so it operates well with generalized Gaussian noise in which β is smaller than 2. However, if compared to other methods, SmoothSeg method's performance is the worse. HaarSeg method [45] uses wavelet based pattern matching so it is so robust with bias. However, Haar filter is used in stationary wavelet domain, so it is so sensitive with outlier or it is not robust with heavy-tailed noise. However, this method gives a promising result. HaarSeg is better than denoising methods and two segmentation methods such as CBS and GADA1. If compared by ROC curves, HaarSeg works even better than GADA2 in case of small SNR. CBS method [32] is the best in case without bias. That means that CBS is really robust with heavy-tailed noise. However, it is so sensitive with bias. That is the reason it get worse when signal has bias. GADA1 method [35] can be comparable with CBS. However robust property with heavy-tailed noise is not good as CBS. This method runs much faster than CBS. It still has a problem with bias as CBS. GADA2 method [34] is designed to operate with probe hybridization bias. It is also robust with heavy-tailed noise. This method is better than GADA1, CBS, HaarSeg and all denoising methods. Becuase GADA2 is designed to be robust with hybridization, it will lost some segment with small energy. This is its disadvantage. With small SNR segments, GADA2 works worse than DWSS and HaarSeg. However, except for that disadvantage, this method was the best method so far.

With Gaussian noise and no bias, there are three methods whose RMSEs are smaller than 0.001. They are Wave, CBS, GADA1 and DWSS. With this case, Wave method is good enough because its' RMSE is small enough and it's speed is fast. With heavy-tailed noise and no bias, Wave and GADA1 methods get worse. Two methods which still work well with heavy tailed noise are CBS and DWSS. So, CBS is best method with simulated data (Gaussian and GGD) without bias. However, with bias, CBS gets worse so fast. Just DWSS is robust with hybridization bias and heavy tailed noise. Some methods being sensitive with bias are Lowess, Wave, SmoothSeg and GADA1. With real noise without bias, DWSS is the best, CBS is the second and GADA1 is the third. In both Gaussian noise and real noise, DWSS is on the top, GADA2 is the second one, HaarSeg is the third one. In real data, GADA2 is the second, DWSS is the best. HaarSeg gets worse in this data because of heavy-tailed noise.

Two samples in Fig. 3.14 and Fig. 3.15 are used to illustrate how heavy-tailed noise and bias effect to performances of Wave, HaarSeg, CBS, GADA1, GADA2 and DWSS. Wave represents denoising methods. The rest methods are segmentation methods. In Fig. 3.14, a real noise simulated sample without bias is used to show that Wave is sensitive with outlier or heavy-tailed noise. Also in this example, HaarSeg and GADA2 lose the second segment (deletion), while CBS, GADA1 and DWSS segmented enough three segments. In Fig. 3.15, also with real noise simulated sample, however, bias of 0.2 (see Eq.(3.35)) has been added to this sample to test how hybridization bias effect to testing methods. Wave is smoothing method so it is so sensitive with hybridization bias. CBS and GADA1 are not designed for bias problem so their performances get worse with bias. In this sample, HaarSeg, GADA2 and DWSS are robust with hybridization bias. However, only DWSS detects exact three segments(two normal and one deletion). Through this sample, HaarSeg is not

really robust with outlier. Heavy-tailed noise suppression step helps DWSS be robust with outlier.

3.6 Conclusion

In this chapter, noise distribution has been studied in array CGH data using five real datasets in many platforms with different resolutions. As discussed, almost all previous array CGH data processing and analysis methods assumed that the noise PDF is Gaussian. However the recent work and experimental results show that array CGH noise is heavy-tailed noise. When compared with other distributions used in previous research such as Gaussian and Student's t distributions, generalized Gaussian distribution fits very well noise PDF in the array CGH data. Therefore GGD has been proposed for modeling noise assumption in the array CGH data and developed a novel smoothing-segmentation method based on this generalized Gaussian noise. Bivariate shrinkage's theory in SWT is built with an approach to suppress heavy-tailed noise in array CGH. One-directional Gaussian wavelet derivative scalogram is defined and proposed to detect breakpoints in array CGH. Because the ground truth aberration regions are not clear in real array CGH datasets, synthetic array CGH data plays an important role in array CGH analysis algorithm evaluation. By generalized Gaussian noise and real noise are used, the synthetic array CGH data models which are closer to the real array CGH data than the most commonly used standard [15] and [34] have been improved. Both synthetic data and real data are used to evaluate the performance of the proposed method, DWSS. New method outperforms other most commonly used algorithms in array CGH literature both in terms of RMSE and ROC curve.

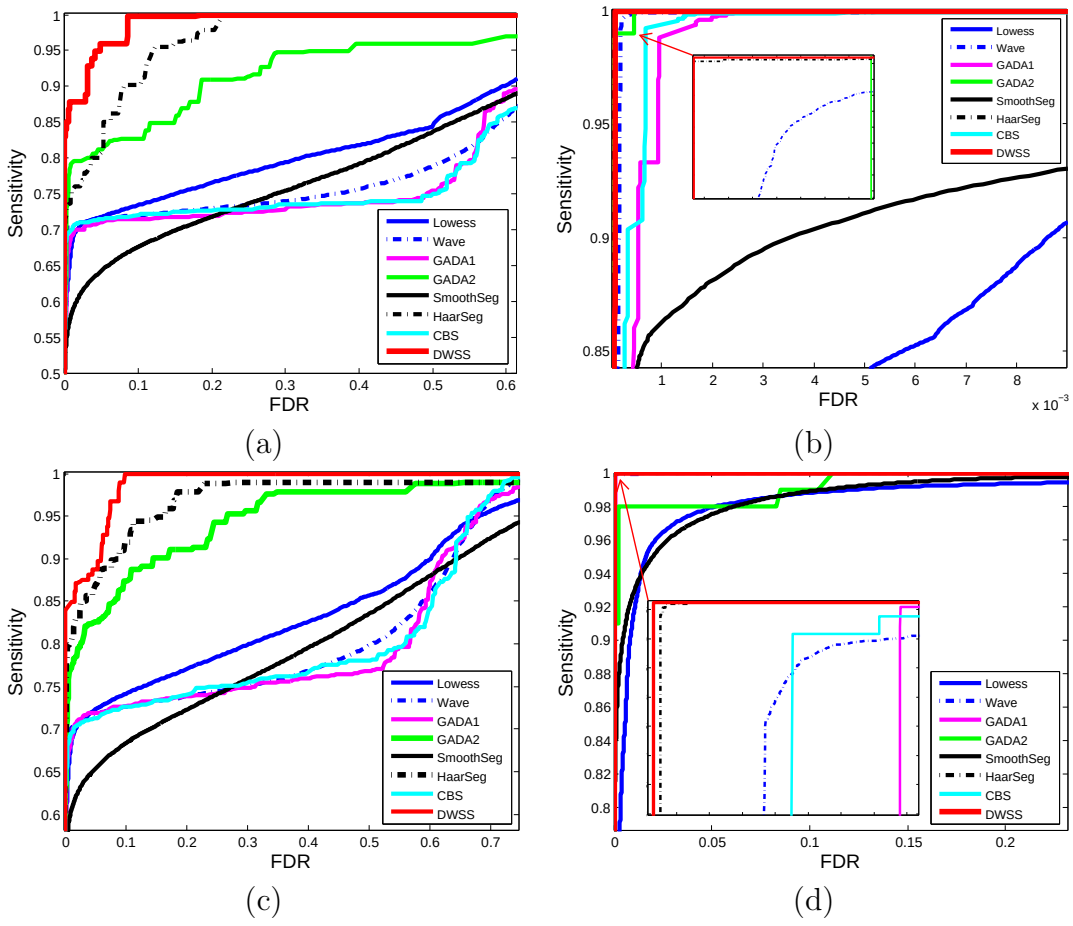


Figure 3.13. One hundred simulated samples whose true abnormal segments have amplitudes of $\log_2 \frac{3}{2}$ and $\log_2 \frac{4}{2}$ are used. Bias of 0.8 is also used in these figures. ROC curves are obtained from arrays which are generated from 8 genomic templates and both Gaussian as well as real noise source. DWSS is compared to other most common used array CGH algorithms such as Lowess, Wave, SmoothSeg, HaarSeg, CBS, GADA1 and GADA2. In four cases, DWSS still yields the best performance. The second one is HaarSeg. (a)Gaussian Noise-Segment gain of $\log_2 \frac{3}{2}$, (b)Gaussian Noise-Segment gain of $\log_2 \frac{4}{2}$, (c)Real Noise-Segment gain of $\log_2 \frac{3}{2}$, (d)Real Noise-Segment gain of $\log_2 \frac{4}{2}$.

Table 3.4. Comparison of average RMSEs obtained from six simulated data-sets with 1,000 arrays each will be shown in this table. Gaussian noise and GGD noise representing for heavy-tail noise will be added to true signal. Bias of 0, 0.2 and 0.4 (see Eq.(3.35)) are used in each kind of noise. Eight tested methods are Lowess(L), Wave(W), SmoothSeg(S), HaarSeg(H), CBS(C), GADA1(G1), GADA2(G2) and DWSS(D). Gauss(.2) means Gaussian($\sigma = 0.2$). GG(.2,1.2) means GGD($\alpha = 0.2$,$\beta = 1.2$).

Noise	*Bias*	*L*	*W*	*S*	*H*	*C*	*G*1	*G*2	*D*
Gauss(.2)	0	.0492	.0071	.0448	.0124	.0041	.0063	.0128	.0058
Gauss(.2)	0.2	.0677	.0610	.0867	.0244	.0721	.0703	.0189	.0113
Gauss(.2)	0.4	.1077	.1305	.1550	.0488	.1436	.1418	.0446	.0281
GG(.2,1.2)	0	.0473	.0179	.0386	.0125	.0048	.0145	.0118	.0057
GG(.2,1.2)	0.2	.0679	.0624	.0850	.0261	.0714	.0700	.0189	.0122
GG(.2,1.2)	0.4	.1071	.1305	.1550	.0497	.1434	.1417	.0456	.0293

Table 3.5. Comparison of average RMSEs obtained from three real noise simulated data-sets with 1,000 arrays each. Biases of 0, 0.2 and 0.4 (see Eq.(3.35) are used in this case. Eight tested methods are Lowess(L), Wave(W), SmoothSeg(S), HaarSeg(H), CBS(C), Gada1(G1), Gada2(G2) and DWSS (D).

Bias	*L*	*W*	*S*	*H*	*C*	*G*1	*G*2	*D*
0	.0522	.0345	.0492	.0330	.0161	.0199	.0239	.0107
0.2	.0706	.0710	.0896	.0425	.0742	.0725	.0286	.0168
0.4	.1089	.1344	.1567	.0588	.1446	.1427	.0482	.0312

Table 3.6. Comparison of average RMSEs obtained from Lee 2008 array including 40 samples, Smith2007 array including 69 samples and Nicolas2009 array including 23 samples. Six tested methods are Lowess(L), Wave(W), SmoothSeg(S), HaarSeg(H), CBS(C), GADA1(G1), GADA2(G2) and DWSS(D).

Data Source	*L*	*W*	*S*	*H*	*C*	*G*1	*G*2	*D*
Lee 2008	.0257	.0436	.0555	.0352	.0286	.0301	.0156	.0017
Smith 2007	.0330	.0581	.0551	.0515	.0425	.0390	.0154	.0127
Nicholas 2008	.0386	.0338	.0697	.0217	.0439	.0308	.0185	.0128

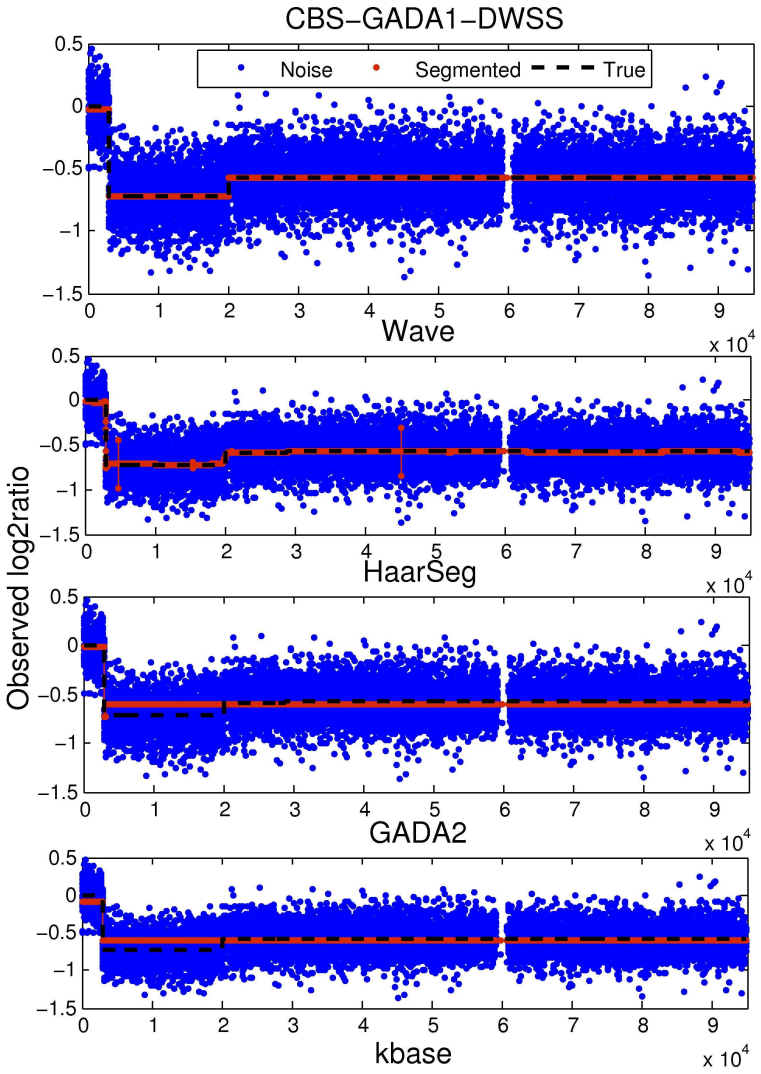


Figure 3.14. With real noise simulated data without bias, CBS, GADA1 and DWSS get good results. Wave is sensitive with heavy-tailed noise. HaarSeg and GADA2 miss some segments.

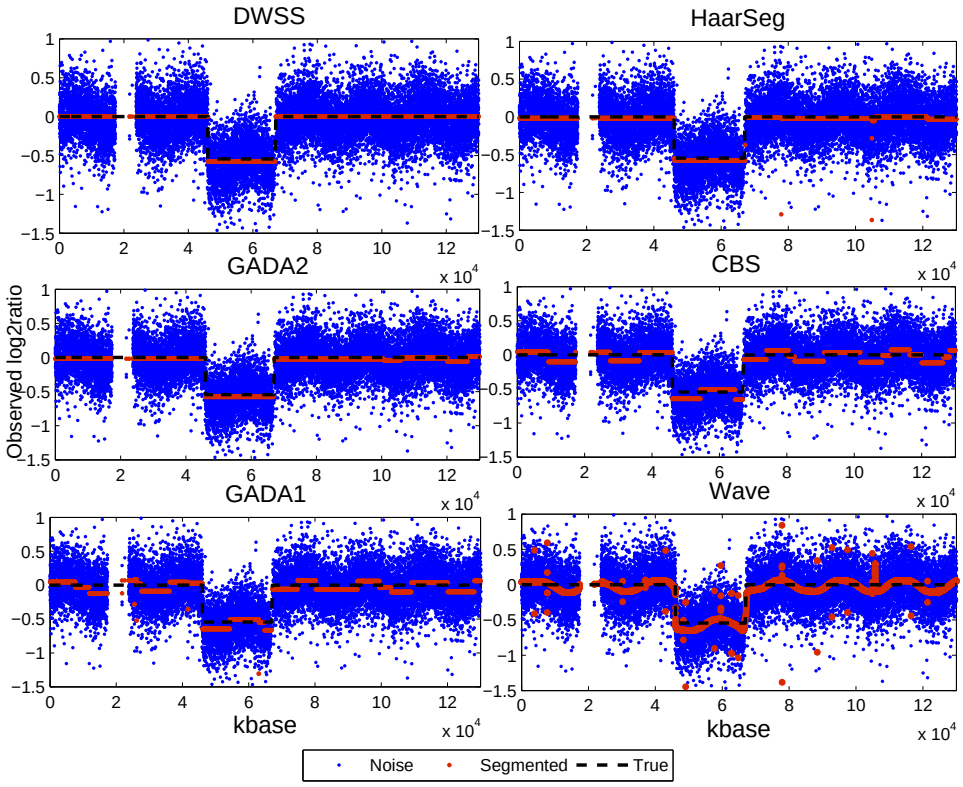


Figure 3.15. With real noise simulated data with bias of 0.2 (see Eq.(3.35)), just DWSS provides good result. CBS and GADA1 are so sensitive with bias. GADA2 is better but it cannot give good result. HaarSeg and Wave are not robust with heavy-tailed noise.

CHAPTER 4
GABOR FILTERS AND ENVELOPE ANALYSIS BASED MASS SPECTROMETRY PEAK DETECTION

4.1 Introduction

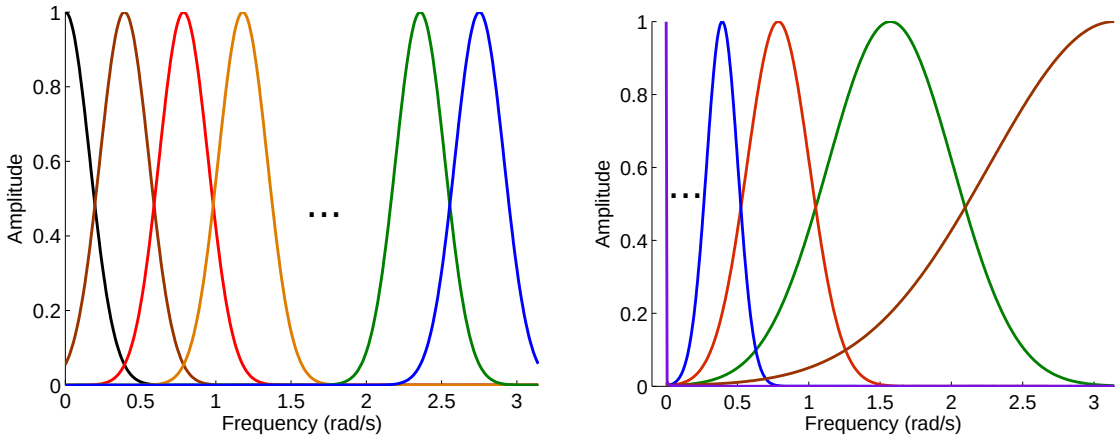
Mass Spectrometry (MS) is an analytical technique that has been widely used to discover diseases related proteomic patterns. From these proteomic patterns, researchers can identify bio-markers, make an early diagnosis, observe disease progression, response to treatment and so on. Peak detection is one of the most important steps in the analysis of mass spectrum because its performance directly affects the other processing steps and final results such as profile alignment [46], bio-marker identification [47], and protein identification [48].

There are two peak detection approaches: denoising [49, 50] and non-denoising (or decomposing) [51, 52] methods. There are several similar steps between these two approaches such as baseline correction, alignment of spectrograms, and normalization. They also use local maxima to detect peak positions and use some rules to quantify peaks. Specially, both approaches use the signal to noise ratio (SNR) to remove the small energy peaks whose SNR values are less than a threshold. However, in the denoising approach, before peak detection, a denoising step is used to reduce the noise of mass spectrum data. In the non-denoising approach, a decomposition step is used to analyze mass spectrum into different scales before the peak detection by local maxima. When the smoothing step is applied into denoising approach, it possibly removes both noise and signal. If the real peaks are removed by smoothing step, they can never be recovered in the other processing steps. As a result, some important information is lose and error is introduced into MS data analysis. Thus, the way of

decomposistion a signal into many scales without denoising is a really better approach with great potentials.

The SNR is used to identify peaks in both denoising and non-denoising methods. Du *et al* [51] estimated the SNR in the wavelet space and got much better results than the previous work. However, they still failed to detect some peaks with small SNRs [51]. This problem came from the SNR value estimation and all previous methods estimated the SNR values by using the relationship between the peak amplitude and the surrounding noise levels. Since some sources of noise can also have high amplitudes, the high amplitude peak does not always guarantee to be a real peak. On the other hand, some low amplitude peaks can also be real peaks. It is clear that the way using SNR to quantify peaks is not efficient and not accurate.

In this chapter, two novel robust MS peak detection approaches such as GaborLocal and GaborEnvelop are proposed. First the Gabor filters are used to create many scales from the original signal without smoothing. The Gaussian local maxima is exploited to detect peaks in the GaborLocal method instead of the local maxima that is less robust to the noise of mass spectrum. Furthermore, the envelope analysis is also proposed and applied to detect peaks in the GaborEnvelop method. Finally, the peak rank (PR) is used to remove some false peaks instead of the SNR. The real SELDI-TOF spectrum with known polypeptide composition and position is used to evaluate the proposed methods. The experimental results show that new approaches can detect both high amplitude and small amplitude peaks with a low false discovery rate and are much better than the previous methods. Two proposed methods are also compared in section 4.5.3.



(a) (b)

Figure 4.1. Frequency supports of complex Gabor filters. (a) uniform filters, and (b) non-uniform filters.

4.2 Complex Gabor Filters

The Gabor filters are developed to create Gaussian transfer functions in the frequency domain. The Gabor filters have been shown to have optimal combined localization in both spatial and spatial-frequency domains ([53, 54]). A generic one-dimensional Gabor function and its Fourier transform are given by:

$$\begin{aligned} h(t) &= \frac{1}{\sqrt{2\pi}\sigma}\exp(-\frac{t^2}{2\sigma^2})\exp(j2\pi F_i t), \\ H(f) &= \exp(-\frac{(f-F_i)^2}{2\sigma_f^2}), \end{aligned} \tag{4.1}$$

where $\sigma_f = 1/(2\pi\sigma)$ represents the bandwidth of the filter and F_i is the central frequency. In certain applications, this filtering technique has been demonstrated to be robust and fast ([55]) and the recursive implementation of 1D Gabor filtering has been shown in [56]. This recursive algorithm for the Gabor filter possibly achieves the fastest implementation. For a signal consisting of N samples, this implementation requires $O(N)$ multiply-and-add operations.

The Gabor filter can be viewed as a Gaussian modulated by a complex sinusoid (with center frequencies F_i). This filter responds to some frequency, but only in a

localized part of a signal. The coefficients of Gabor filters are complex. Therefore, the Gabor filters have one-side frequency support as shown in Fig. 4.1.

4.3 Envelope Analysis

In this section, a novel analysis method, envelope analysis which is based on the Gaussian local maxima, minima, and interpolation algorithms is proposed. A signal can be decomposed into three envelopes including maximal envelope, minimal envelope, and median envelope that will be defined and described below.

Gaussian local maxima and minima: Local maxima and local minima of $y(t)$ will be found. Two steps should be followed such as computing derivative of $y(t)$ and finding zero crossing. The derivative of $y(t)$ is approximated by the finite difference as follows:

$$\frac{d(y(t))}{dt} = \lim_{h->0} \frac{y(t+h) - y(t)}{h} \approx y(t+1) - y(t). \tag{4.2}$$

At $t = t_0$, if the derivative of $y(t)$ equals to zero and has a change from positive to negative or from negative to positive, there is zero-crossing. If the derivative of $y(t)$ changes from positive to negative at t_0, there is local maxima at t_0. Otherwise, if the derivative of $y(t)$ changes from negative to positive at t_0, there is local minima at t_0. With discrete signal, Eq. 4.2 can be rewritten as follows

$$\frac{d(y(n))}{dn} = y(n+1) - y(n) = y(n) * [1 \quad -1]. \tag{4.3}$$

Unfortunately, MS data always have noise. Thus, Gaussian filter $g(t, \sigma)$ is used to make the proposed methods more robust to noise in MS data. This is not a denoising step because the noise is not removed. Finally, derivative of $y(t) * g(t, \sigma)$ will replace the derivative of $y(t)$ as follows

$$\frac{d(y(t) * g(t, \sigma))}{dt} = \frac{d(\int_{-\infty}^{\infty}(y(\tau).g(t-\tau, \sigma)d\tau))}{dt}$$

Table 4.1. The values of vector $v(k)$ with different lengths.

length	k = 1	2	3	4	5	6	7	8	9
5	0.0007	0.2824	0	-0.2824	-0.0007				
6	0.0007	0.1259	0.7478	-0.7478	-0.1259	-0.0007			
7	0.0007	0.0654	0.6572	0	-0.6572	-0.0654	-0.0007		
8	0.0007	0.0388	0.4398	0.6372	-0.6372	-0.4398	-0.0388	-0.0007	
9	0.0007	0.0254	0.2824	0.7634	0	-0.7634	-0.2824	-0.0254	-0.0007

$$= \int_{-\infty}^{\infty} (y(\tau).\frac{d(g(t-\tau,\sigma))}{dt}d\tau) = y(t) * \frac{d(g(t,\sigma))}{dt}, \tag{4.4}$$

where

$$g(t,\sigma) = \exp(-\frac{t^2}{2\sigma^2}). \tag{4.5}$$

Taking the derivative of $g(t,\sigma)$ in Eq. 4.5,

$$\frac{d(g(t,\sigma))}{dt} = \frac{-t}{\sigma^2}\exp(-\frac{t^2}{2\sigma^2}). \tag{4.6}$$

From Eq. 4.4 and Eq. 4.6,

$$\frac{d(y(t)*g(t,\sigma))}{dt} = y(t) * \frac{d(g(t,\sigma))}{dt} = y(t) * (\frac{-t}{\sigma^2}\exp(-\frac{t^2}{2.\sigma^2})). \tag{4.7}$$

Instead of zero crossing of $\frac{d(y(t))}{dt}$, zero-crossing of $\frac{d(y(t)*g(t,\sigma))}{dt}$ is found by Eq. 4.7. With discrete signal, Eq. 4.7 can be rewritten as follows

$$\frac{d(y(n)*g(n,\sigma))}{dn} = y(n) * v(n), \tag{4.8}$$

where $v(n)$ is listed in Table 4.1. Using Gaussian filters makes the Gaussian local maxima and minima method more robust with noise.

Definition 1 *Maximal envelope (MAX) of a signal $y(t)$ is a signal of the same length containing all Gaussian local maxima of $y(t)$ and all interpolation points between them.*

Maximal envelope is created from $y(t)$ by two following steps:

- Step 1: Finding Gaussian local maxima of $y(t)$ and their indices;
- Step 2: Performing interpolation of the maximal points obtained from step 1 so that the lengths of maximal envelope and $y(t)$ are the same.

Definition 2 *Minimal envelope (MIN) of a signal $y(t)$ is a signal of the same length containing all Gaussian local minima of $y(t)$ and all interpolation points between them.*

Minimal envelope is created from $y(t)$ by two following steps:

- Step 1: Finding Gaussian local minima of $y(t)$ and their indices;
- Step 2: Performing interpolation of the minima points obtained from step 1 so that the lengths of minimal envelope and $y(t)$ are the same.

Definition 3 *Median envelope (MED) of a signal $y(t)$ is a signal of the same length containing non-maxima and non-minima of $y(t)$ and all interpolation points between them.*

Median envelope is created from $y(t)$ by two following steps:

- Step 1: Finding points of y(t) which are not maximal and minimal as well and their indices;
- Step 2: Performing interpolation of the median points obtained from step 1 so that the lengths of median envelope and $y(t)$ are the same.

Fig. 4.2(a) shows the flowchart of envelope analysis. Any finite energy signal $y(t)$ can be analyzed into three envelope signals including MAX, MIN and MED envelopes at the first level. Each of these envelopes can be considered as a signal and will be decomposed into three envelopes. Therefore, there are $3^2 = 9$ envelopes at the second level. This process is iterated and there are 3^i envelopes at the i^{th} level. In general, the envelope analysis can be formulated as follows

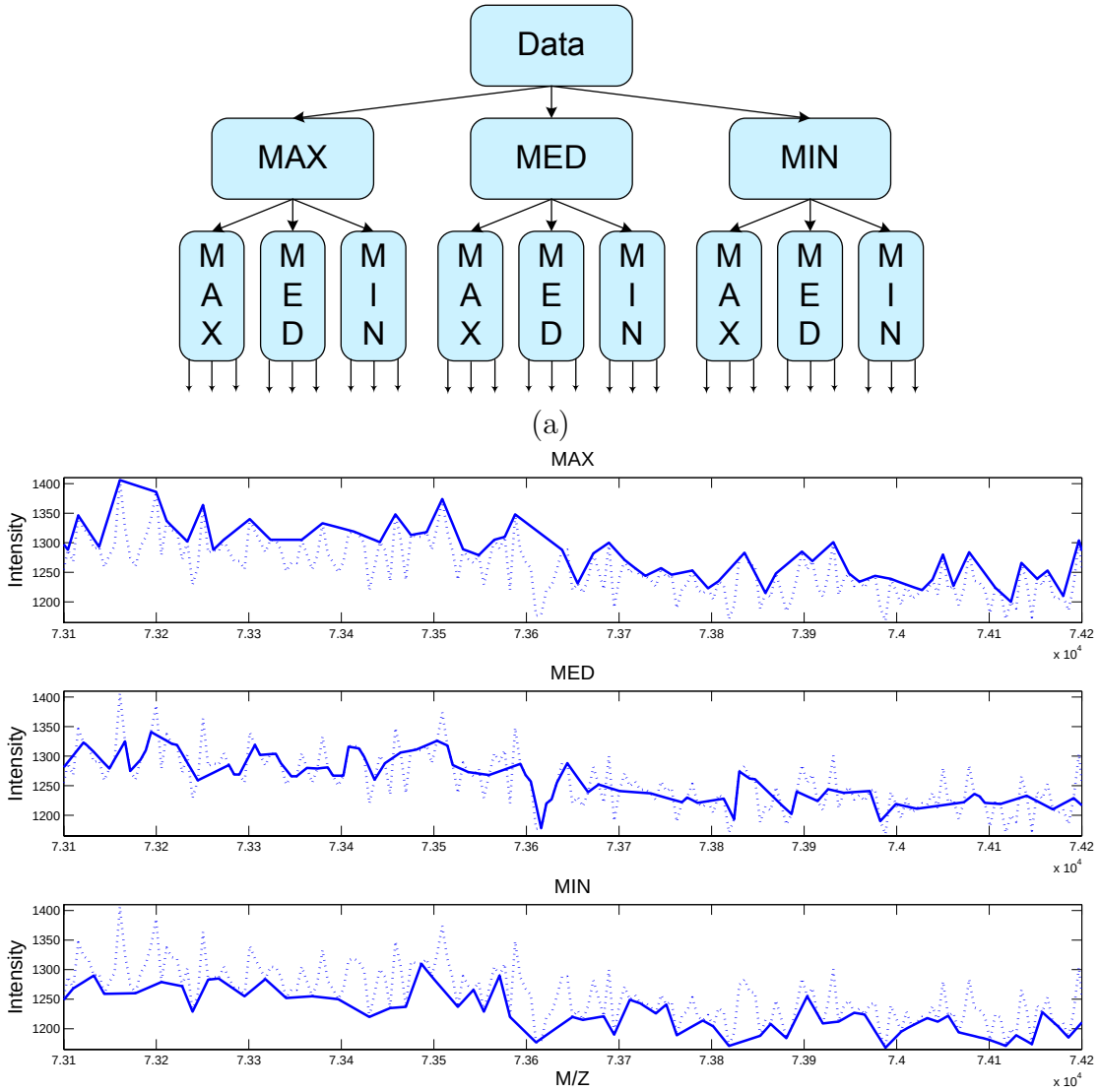


Figure 4.2. (a) General flowchart of envelope analysis. (b) An example of envelopes. The solid lines represent MAX, MED and MIN envelopes of the 6^{th} MS signal of CAMDA 2006 ([57]) at the first level. The dash lines here show the real MS signal.

$$\begin{aligned} Level_1 &= [M_{11}, M_{12}, M_{13}], \\ Level_2 &= [M_{21}, M_{22}, ..., M_{29}], \\ &... \\ Level_i &= [M_{i1}, M_{i2}, ..., M_{i3^i}], \end{aligned} \tag{4.9}$$

where M_{ij} represents the j^{th} envelope at level i. With $k \geq 0$, when $j = 1 + 3k$, M_{ij} is the maximal envelope. And when $j = 2 + 3k$ or $j = 3 + 3k$, M_{ij} is the median envelope or the minimal envelope, respectively. Fig. 4.2(b) shows an example of the envelope analysis.

In some cases, a signal can be analyzed into some specific M_{ij} envelopes. This option depends on some different applications. In this chapter, MAX and MED envelopes are used to detect peaks because MIN envelopes contain no peak. Eq. (4.10) describes the structure of envelope analysis used in the proposed method.

$$\begin{aligned} Level_1 &= [M_{11}], \\ Level_2 &= [M_{21}, M_{22}], \\ Level_3 &= [M_{31}, M_{32}, M_{34}, M_{35}], \\ &... \\ Level_n &= [M_{n1}, M_{n2}, M_{n4}, M_{n5}], \end{aligned} \tag{4.10}$$

where n is the number of levels chosen to satisfy the threshold of number of peaks. How to choose the value of n will be discussed in next section. In this peak detection application, the original signal is decomposed into one MAX envelope at level 1, one MAX and one MED envelope at level 2 and four envelopes which comprises two MAX envelopes and two MED envelopes at level $n > 2$ as shown in Fig. 4.3.

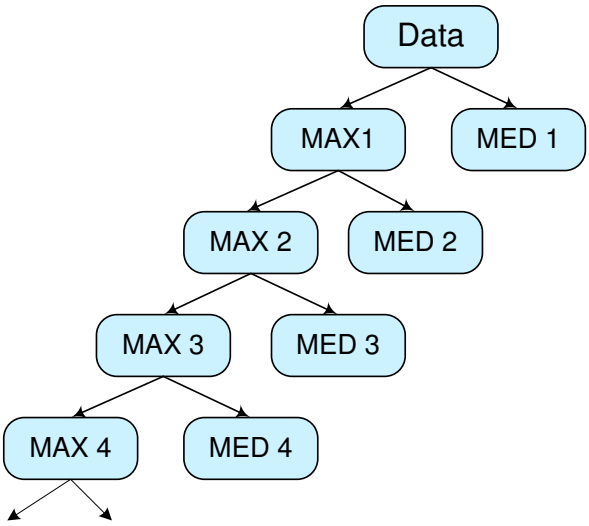


Figure 4.3. Proposed envelope analysis for peak detection.

4.4 GaborLocal and GaborEnvelop Methods

The main idea is to amplify the true signal and compress the noise of mass spectrum by using the Gabor filter bank. After that, the Gaussian local maxima is used to detect peaks and the peak rank which will be defined later to quantify peaks. This method is named as Gabor filter - Gaussian local maxima (GaborLocal). Envelope analysis can be also used to detect and quantify peaks and this method is called as Gabor filter - envelope analysis (GaborEnvelop). Fig. 4.4 and Fig. 4.5 are the flowchart of GaborLocal and GaborEnvelop methods. Each method can be detailed into the four steps including the full frequency MS signal generation, the peak detection, the peak quantification, and the intersection. Both methods have the same first step (full frequency MS signal generation) and the same last step (intersection). They are different at the peak detection and the peak quantification steps.

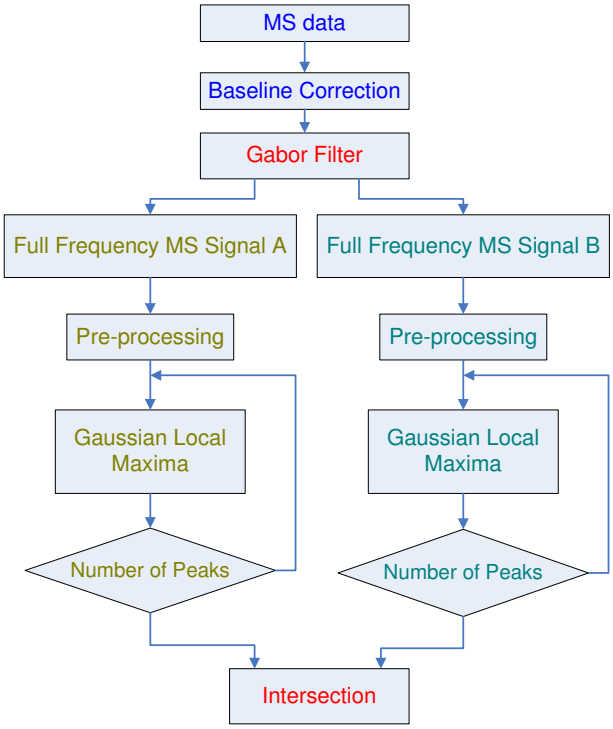


Figure 4.4. Flowchart of GaborLocal method in the MS peak detection.

4.4.1 Full Frequency MS Signal Seneration

Mass spectrum is decomposed to many scales by using the Gabor filters after the baseline correction. The purpose is to emphasize some hidden peaks buried by noise. When 60 MS signals of the CAMDA 2006 are analyzed in the frequency domain, the valuable information of these signals locate from zero to around 0.06 (rad/s) and the noises locate from 0.06 to π (rad/s).

Therefore, the bandwidth σ_f of the Gabor filters which enhances peaks must be less than 0.06. In experiments, $\sigma_f = 0.01$ is used. If the uniform Gabor filter is used, the number of scales must be

$$N = \frac{\pi}{0.01} \approx 314 \; scales. \tag{4.11}$$

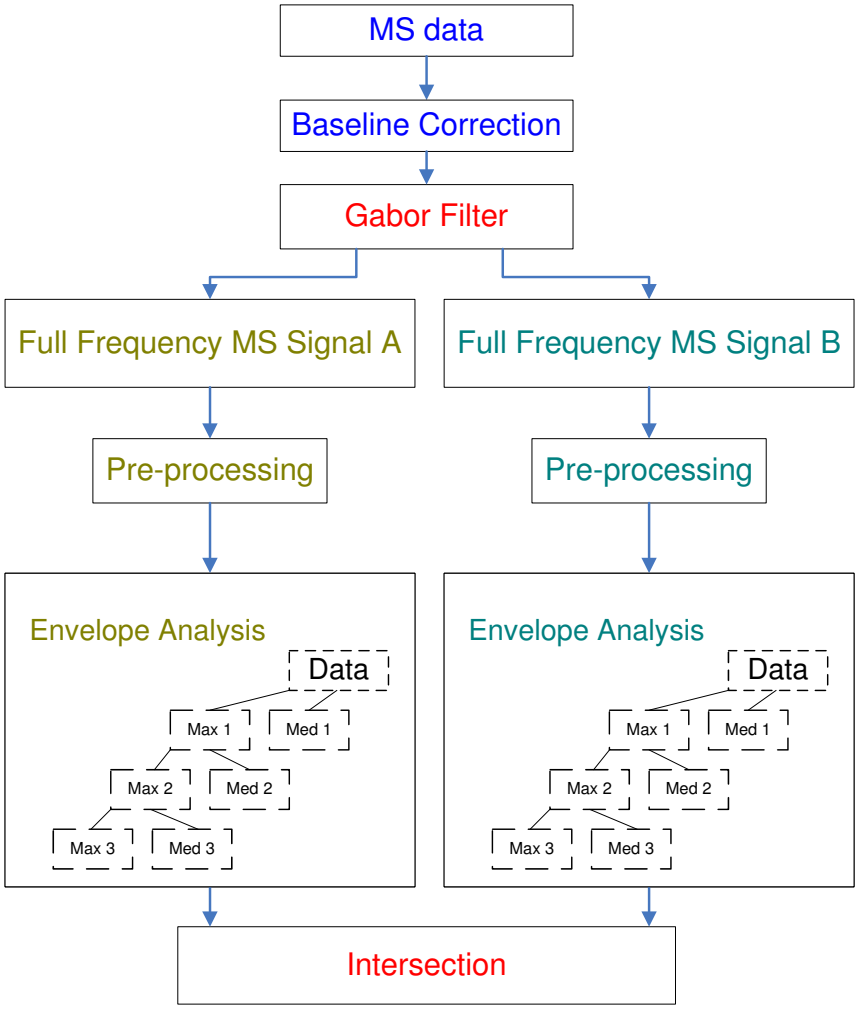


Figure 4.5. Flowchart of GaborEnvelop method in the MS peak detection.

With 314 scales in Eq. 4.11, the uniform Gabor filter is not efficient. If the non-uniform Gabor filter is used, the number of scales should be calculated as follows:

$$\sigma_f = \frac{\pi}{2^N},$$

$$N = log_2(\frac{\pi}{\sigma_f}),$$

$$N \approx 8.3 \; scales \; with \; \sigma_f = 0.01. \tag{4.12}$$

Based on Eq. 4.12, the non-uniform Gabor filters are used with 9 scales to decompose the MS data (CAMDA 2006 data [57] is used for experiments). If $y_i(t),\ h_i(t)$ and $x(t)$ are transformed into the frequency domain

$$Y_i(f) = X(f) \cdot H_i(f), \tag{4.13}$$

where $X(f)$ is the frequency response of the raw MS signal, $H_i(f)$ is the frequency response of the i^{th} Gabor filter, and $Y_i(f)$ is the frequency response of the i^{th} scale. After we get 9 signals according to 9 frequency sub-bands in complex values, the full frequency signal A will be created by summing above signals in complex values first and taking their absolute values at the final. To create the full frequency signal B, the absolute values are taken for each sub-band and then sum all these sub-bands. After this step, there are two full frequency signals A and B. Let us denote $y(t)$ and $Y(f)$ as the full frequency signal in time domain and frequency domain, respectively.

$$Y(f) = \sum_{i=N_i} Y_i(f), \tag{4.14}$$

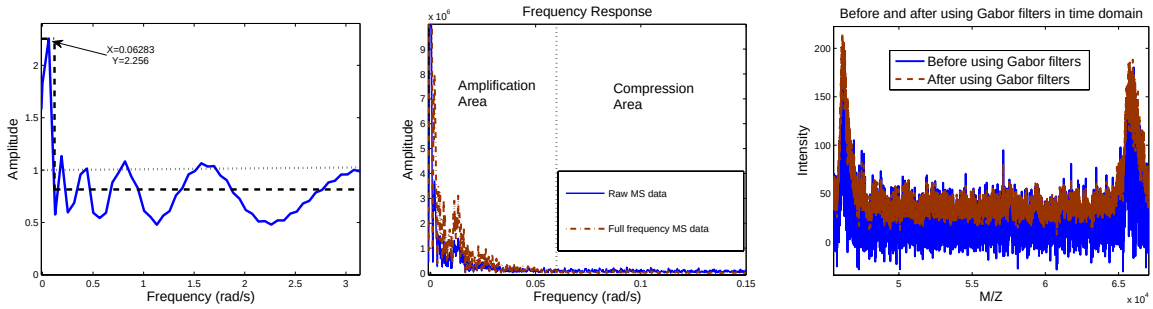
where N_i are the scales which are used to create the full frequency signal. From Eq. 4.13 and 4.14

$$\begin{aligned} Y(f) &= \sum_{i=N_i} X(f)H_i(f) \\ &= X(f) \sum_{i=N_i} H_i(f) = X(f)H_s(f), \end{aligned} \tag{4.15}$$

where $H_s(f) = \sum_{i=N_i} H_i(f)$ is called the summary filter. The summary filter can be formulated as follows

$$H_s(f) = \sum_{i=N_i} \exp(-\frac{(f - F_i)^2}{2\sigma_f^2}). \tag{4.16}$$

Illustration: Intuition using Gabor filters The purpose of this step is to amplify the true signal and to compress the noise. The black line in Fig. 4.6(a) is $H_s(w)$ which can amplify the true signal from 0 to 0.06 $\frac{rad}{s}$ and compress noise from 0.06 to



(a) (b) (c)

Figure 4.6. (a) The frequency response of the summary filter. (b) The frequency response of one MS data before and after using the summary filter. In amplification area, amplitude of full frequency MS signal is higher than raw MS signal. In compression area, amplitude of full frequency MS signal is smaller than raw MS signal.(c) One example is used to show how Gabor filters to affect MS signal in time domain. The intensity values of peaks are gained and noise is compressed after using Gabor filters. Raw MS data which is used in (b) and (c) is the 19^{th} MS signal of CAMDA 2006.

π. In this case, if using $N_i = [1\ 2\ ...\ 9]$ the summarized filter can be represented by the blue line in Fig. 4.6(a). Fig. 4.6(b) shows the frequency response of the 19^{th} raw MS signal (blue line) and that of full frequency signal (red line). The signal from 0 to 0.06 is amplified and the noise from 0.06 to π is compressed. In Fig. 4.6(c), after using Gabor filters, the intensity values of true peaks have increased and the standard deviations of noise have decreased in time domain. Therefore, in both full frequency MS signal A and B, all peaks have been emphasized to help the next peak detection step. In this step, baseline correction is also used before applying Gabor filters and is detailed as follows

Baseline correction: The chemical noise or the ion overloading is the main reason causing a varying baseline in mass spectrometry data. Baseline correction is

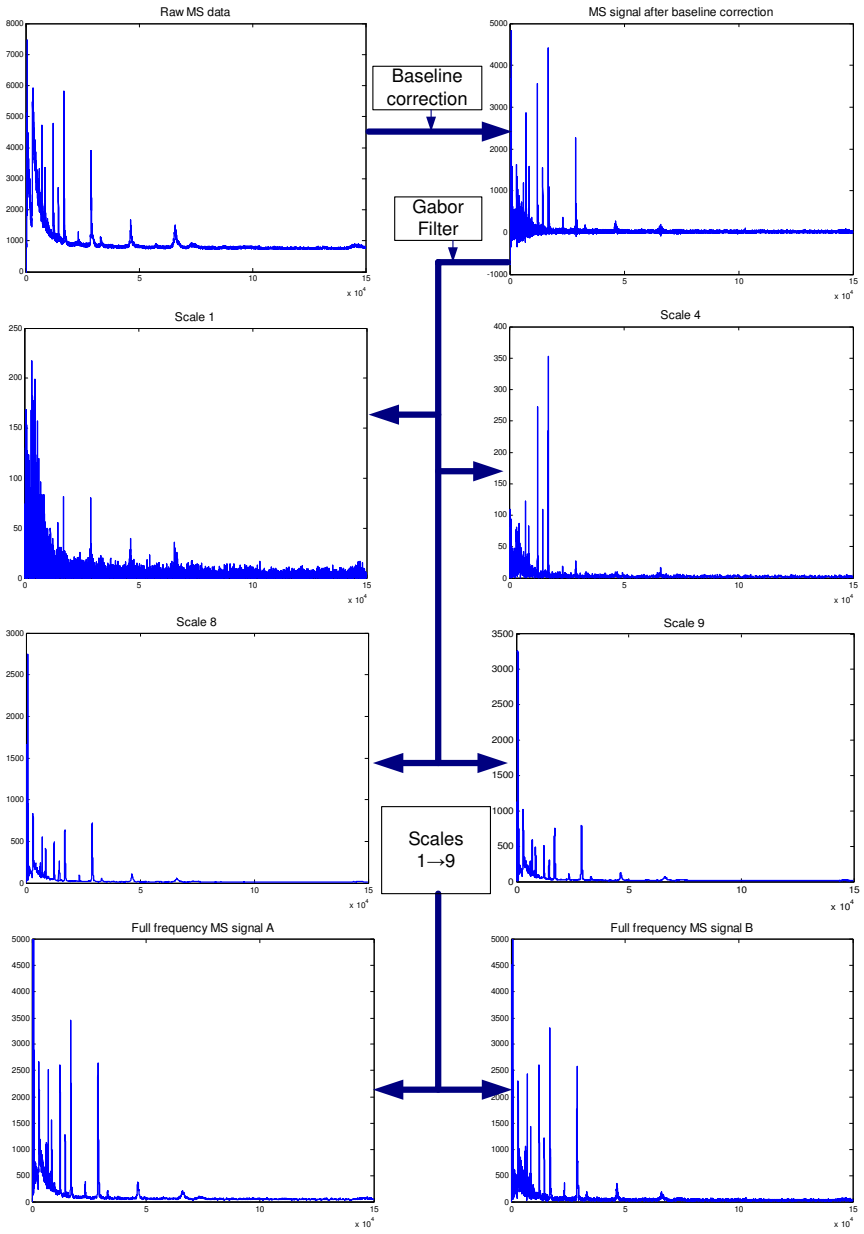


Figure 4.7. One example of the step named full frequency MS signal generation. Raw MS data is the 19^{th} MS signal of CAMDA 2006.

an important step before using Gabor filter to get the full frequency MS signals. The raw MS signal x_{raw} includes some real peaks x_p, the baseline x_b, and the noise x_n.

$$x_{raw} = x_p + x_b + x_n. \tag{4.17}$$

The baseline correction is used to remove the artifact x_b. In this chapter, 'msbackadj' function of MATLAB is used to remove baseline. The msbackadj function estimates a low-frequency baseline first which is hidden among the high-frequency noise and the signal peaks and then subtracts the baseline from the spectrogram. This function follows the algorithms in Andrade *et al.*'s paper [58].

Illustration: In order to understand this step easier, one example of the way to create full frequency MS signal is shown in Fig. 4.7. In this example, the 19^{th} MS signal of CAMDA 2006 is chosen as raw MS data. After the baseline correction, MS signal is used as the input of the Gabor filters. A Gabor filter bank with 9 non-uniform sub-bands is employed to create 9 MS signals with 9 different frequency sub-bands. In Fig. 4.7, the signals of scale 1, 4, 8 and 9 are visualized. Some noises in high frequency are separated from the MS signal of the scale 1, 2, ..., 5. In the MS signal under the scales 6, ..., 9, all high intensity peaks are still kept. After the MS signals of all scales are combined in two ways, the full frequency MS signal A and B are created. The comparison between the raw MS and full frequency signal in frequency and time domain is shown in Fig. 4.6(a)(b)(c). These figures show how to amplify the important signal and compress the noise. This is just a compression of noise instead of removing noise. As the outputs, two full frequency MS signal A and B will be used to detect peaks in the next step instead of raw MS data.

Table 4.2. Definition of peak rank in GaborLocal. Y means that the peak can be detected at that loop. N means that the peak cannot be detected at that loop. The peak with the peak rank equaling to 1 is able to be detected at all of the loops. The peak with the peak rank equaling to n only appeared at the first loop.

Peak Rank	*Loop* 1	*Loop* 2	*Loop* 3	*Loop* 4	... *Loop* $(n-1)$	*Loop* n
1	Y	Y	Y	Y	... Y	Y
2	Y	Y	Y	Y	...Y	N
...	...	...	...	...	...	...
n	Y	N	N	N	...N	N

4.4.2 Peak Detection and Peak Quantification in GaborLocal

All peaks are detected as many as possible by using Gaussian local maxima with the full frequency MS signal A as well as the full frequency MS signal B. The Gaussian local maxima is used instead of local maxima because Gaussian local maxima is robust with noise in peak detection. Before peak detection, pre-processing step is also applied such as peak elimination in the low-mass region.

After many peaks are detected in full frequency MS signals, a new signal is obtained from these peaks. This new signal will be the input of the next peak detection loop where the Gaussian local maxima method is also applied. Then, many loops are repeated until the number of peaks obtained is less than a threshold. Now, the peak rank of peaks is defined as follows:

Peak rank in GaborLocal: n loops are used and get m_1 peaks at the loop 1, m_2 peaks at loop 2,...and m_n peaks at the loop n. There are $m_1 > m_2 > ... > m_n$. Peak rank (PR) is defined as Table 2.

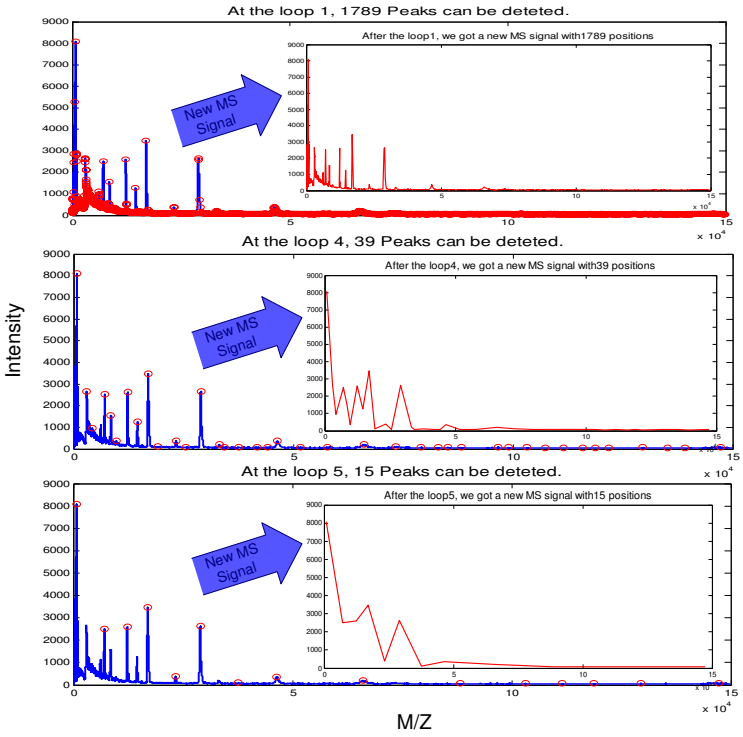
There are m_n peaks with $PR = 1$, $m_{n-1} - m_n$ peaks with $PR = 2$,...and $m_1 - m_2$ peaks with $PR = n$. In the proposed algorithm, the probability of the true peaks with $PR = i$ is higher than with $PR > i$.

Demonstration: Fig. 4.8(a) shows an example of the step named the peak quantification by using the peak rank. First, the full frequency MS signal A is used to detect peaks by using Gaussian local maxima. At loop 1, 1789 peaks can be detected. From these 1789 peaks, a new signal is created with 1789 positions. At the next loops 2, 3, 4, 509, 143, 39 peaks can be detected, respectively. At loop 5, 15 peaks can be detected. If a threshold of 16 and *number of peaks* of 15 are selected, the algorithm will stop at loop 5. Actually, the threshold can be selected from 38 to 16 and also get 15 peaks at the final loop. Now, there are 15 peaks with $PR = 1$, $39-15 = 24$ peaks with $PR = 2$, $143-39 = 104$ peaks with $PR = 3$, $509-143 = 366$ peaks with $PR = 4$ and $1789-509 = 1280$ peaks with $PR = 5$. In this case, 15 peaks are only kept with $PR = 1$. The same on the full frequency MS signal B is done and 12 peaks can be detected with $PR = 1$ at the last loop.

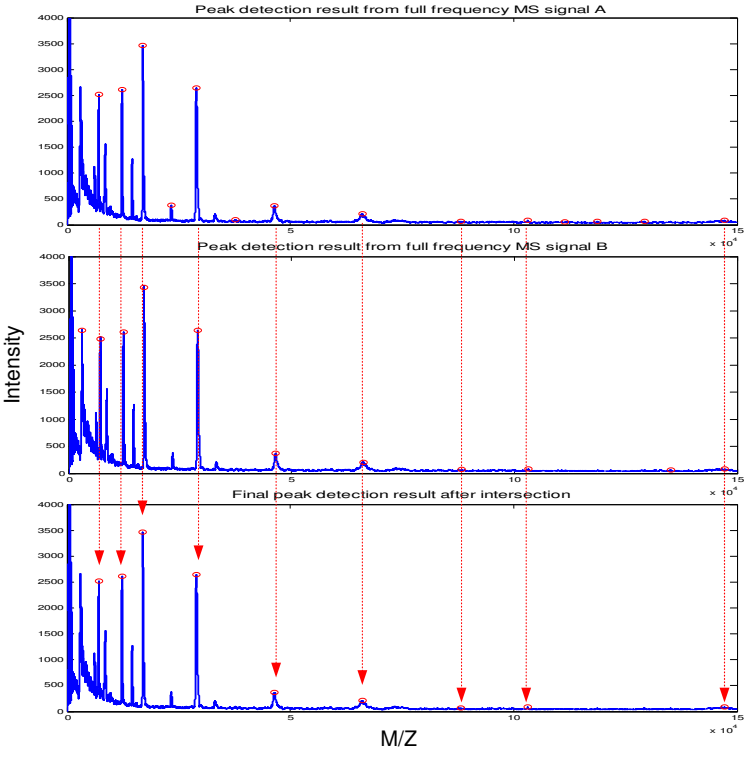
4.4.3 Peak Detection and Peak Quantification in GaborEnvelop

After using Gabor filter, we have visible peaks in the full frequency MS signal A and the full frequency signal B. Envelope analysis of both signal A and signal B is used to detect MS peak instead of Gaussian local maxima. The MAX and the MED are used at the final level to find peaks. Fig. 4.5 shows that MAX and MED are used instead of MAX, MED, and MIN because peaks of data do not appear at MIN envelope. Of course, before envelope analysis of signal A and B are obtained, peak elimination in low-mass region is also applied.

Peak rank in GaborEnvelop: envelope analysis of data is taken at level n. Because only MAX is used at the first level, there are $2n-1$ groups of peaks corresponding $2n-1$ thresholds of peak. Let us assume there are m_1 peaks at the MAX1, m_2 peaks at group of MAX2 and MED2,...and m_n peaks at the MAXn. We have $m_1 > m_2 > ... > m_n$. Peak rank (PR) is defined in Table 3.



(a)



(b)

Figure 4.8. One example of GaborLocal in peak detection (a) peak detection and quantification step, (b) intersection step.

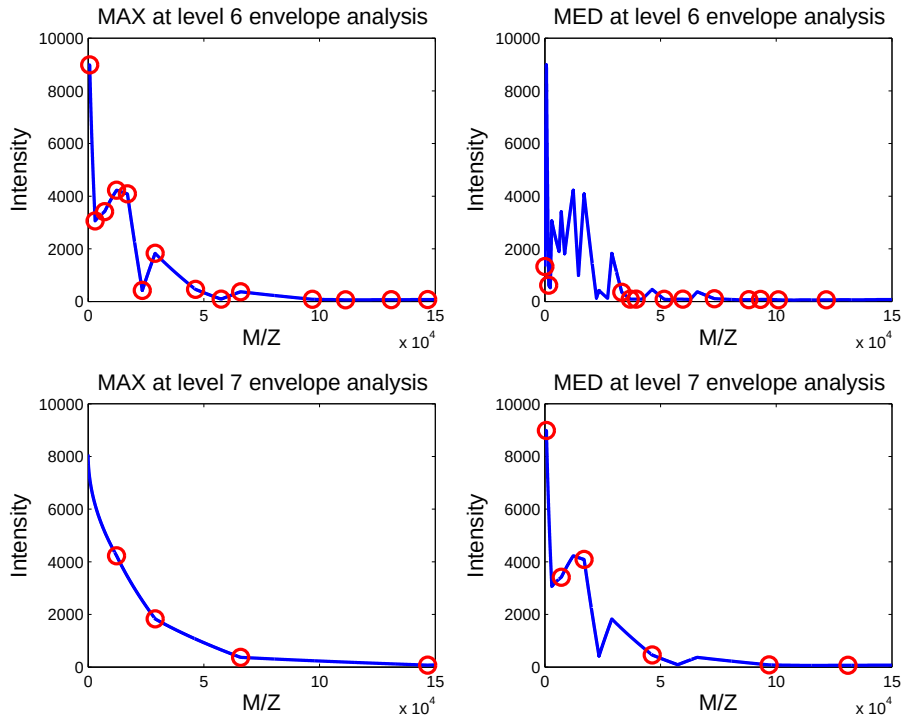


Figure 4.9. One example of envelope analysis. Maximum envelope, median envelope, and minimum envelope of a MS data at level 6 and level 7. The input MS data is the 39^{th} MS signal of CAMDA 2006. The circle sign represents peaks which are from the MAX1 (peaks of the input MS signal) and are the identified peaks.

Table 4.3. Definition of peak rank (PR) in GaborEnvelop. MAXi column means that peaks of MAX envelope are taken at level i. If peaks are detected in both of MAX and MED envelope at level i, that column is named as MAXi, MEDi. "Y" is the peak can be detected. N means that the peak cannot be detected. The peak with the peak rank equaling to 1 is able to be detected at all of envelopes. The peak with the peak rank equaling to n only appeared at the envelope of level 1.

PR	$MAX1$	$MAX2, MED2$	$MAX2$	$MAX3, MED3$	... $MAXn, MEDn$	$MAXn$
1	Y	Y	Y	Y	... Y	Y
2	Y	Y	Y	Y	...Y	N
...	...	...	...	...	...	...
n	Y	N	N	N	...N	N

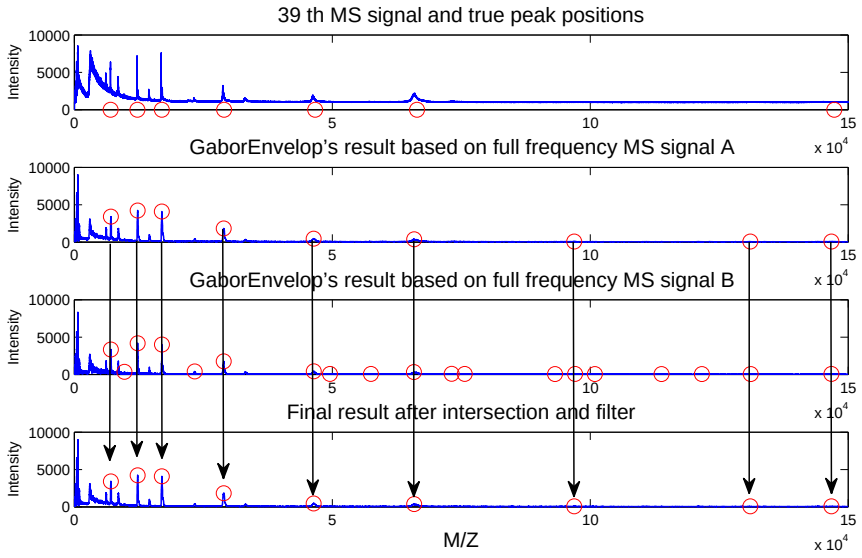


Figure 4.10. One example of GaborEnvelop at intersection step. Raw MS data is the 39^{th} MS signal of CAMDA 2006.

There are m_n peaks with $PR = 1$, $m_{n-1} - m_n$ peaks with $PR = 2$,...and $m_1 - m_2$ peaks with $PR = n$. In the proposed algorithm, the probability of the true peaks with $PR = i$ is higher than with $PR > i$.

Demonstration: Fig. 4.9 shows an example of envelope analysis of the 39^{th} MS signal at level 6 and 7. Only MAX and MED are used in this case. The input signal of envelope analysis is full frequency MS signal A without pre-processing. At the MAX scale of level 6, 14 peaks are detected. If continuing level 7, 4 peaks are obtained at the MAX scale and 6 peaks at the MED scale. If pre-processing is applied to full frequency MS signal A, 5 peaks are just obtained at the MED scale of level 7. Finally, there are $4 + 5 = 9$ peaks from signal A and 19 peaks from signal B.

4.4.4 Intersection

Now, there are two results of peak detection from two full frequency MS signals. The intersection of two above results will be the final result. For example, Fig. 4.8(b) shows how to do the intersection of two results. There are 15 peaks in signal A and 12 peaks in signal B, but there are only 9 peaks as the final result. With this result, there are 7 true peaks and 2 false peaks. With example in Fig. 4.10, after intersection, there are 9 peaks. There are only 7 true peaks and 2 false peaks. These results show that the true position rate (or sensitivity) equals to $\frac{7}{7} = 1$ and the false discovery rate equals to $\frac{2}{9} \approx 0.22$.

In general, the GaborEnvelop includes the GaborLocal. In envelope analysis, if MAX envelope signals are just used, the GaborEnvelop will become the GaborLocal method which uses many loops to quantify peaks. The GaborEnvelop uses both MAX and MED envelopes to keep the number of true peaks (TPR) and decrease the number of false peaks (FDR).

4.5 Experiments and Discussions

In this section, GaborLocal and GaborEnvelop methods will be compared to two other most commonly used methods: the Cromwell [49, 50] and the CWT [51]. The performance of those methods will be evaluated by using the ROC curve that is the standard criterion in this area.

4.5.1 Cromwell Method

Cromwell method is implemented as a set of MATLAB scripts which can be downloaded from [59]. The algorithms and the performance of the Cromwell were described in [50, 49]. The main idea of the Cromwell method can be summarized as follows

- Denoise the individual spectrum using the undecimated discrete wavelet transform. The hard thresholding method was used to reset small wavelet coefficients to zero. In these papers, the authors used the median absolute deviation (MAD) to estimate the thresholding.
- Estimate and remove the baseline artifact by using a monotone local minimum curve on the smoothed signal.
- Normalize the spectrum by dividing the total ion current, defined to be the mean intensity of the denoised and baseline corrected spectrum.
- Identify peaks by using local maxima and signal to noise ratio (SNR).
- Match peaks across spectrum and quantify peaks using either the intensity of the local maximum or computing the area under the curve for the region defined to be the peaks.

4.5.2 CWT Method

The algorithm of CWT method has been implemented in R ('MassSpecWavelet') and the Version 1.4 can be downloaded from [60]. This method was proposed by Pan Du *et al.* [51] in 2006 and can be summarized as follows:

- Identify the ridges by linking the local maxima. Continuous wavelet transform (CWT) is used to create many scales from one mass spectrum. The local maxima at each scale is detected. The next step is to link these local maxima as lines.
- Identify the peaks based on the ridge lines. There were three rules to identify the major peaks. They are the scale with the maximum amplitude on the ridge line, the SNR being larger than a threshold and the length of ridge being larger than a threshold. Notice that the SNR is estimated in the wavelet space. This is a nice motivation of this method.

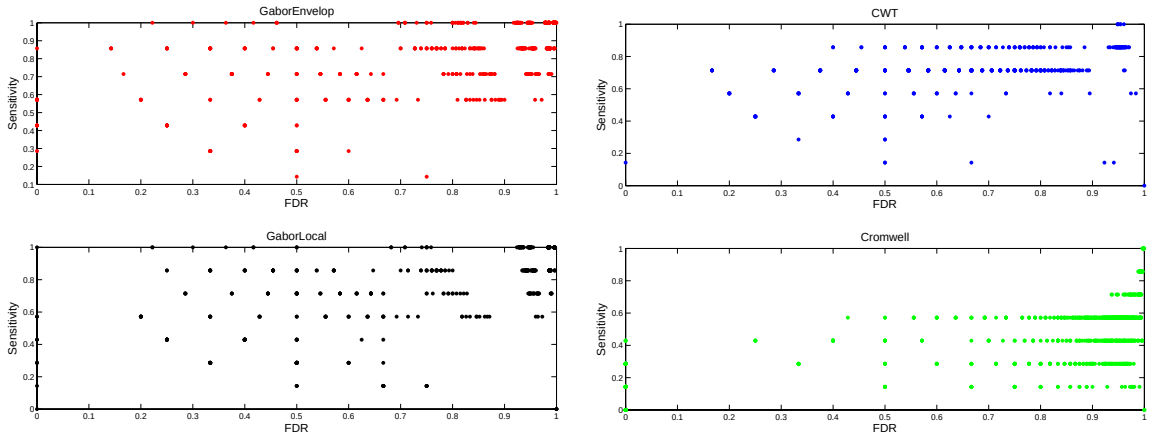


Figure 4.11. Detailed ROC curves obtained from 60 MS signals using Cromwell, CWT, and GaborLocal and GaborEnvelop methods. The sensitivity is the true position rate.

- Refine the peak parameter estimation.

4.5.3 Evaluation Using ROC Curve

The CAMDA 2006 dataset [57] of all-in-1 Protein Standard II (Ciphergen Cat. # $C100-007$) is used to evaluate four algorithms: the Cromwell, the CWT, and two proposed methods. Because polypeptide composition and position are known, the true position rate (TPR or sensitivity) and the false discovery rate (FDR) can be estimated. Another advantage of this dataset is that it is real data and better than the simulated data in evaluation.

The TPR is defined as the number of identified true peaks divided by the total number of true peaks. The FDR is defined as the number of falsely identified peaks divided by the total number of identified peaks. An identified peak is called as true peak if it is located within the error range of 1% of the known m/z value of true peaks. There are seven polypeptides which create seven true peaks at 7034, 12230, 16951, 29023, 46671, 66433 and 147300 of the m/z values. Fig. 4.11

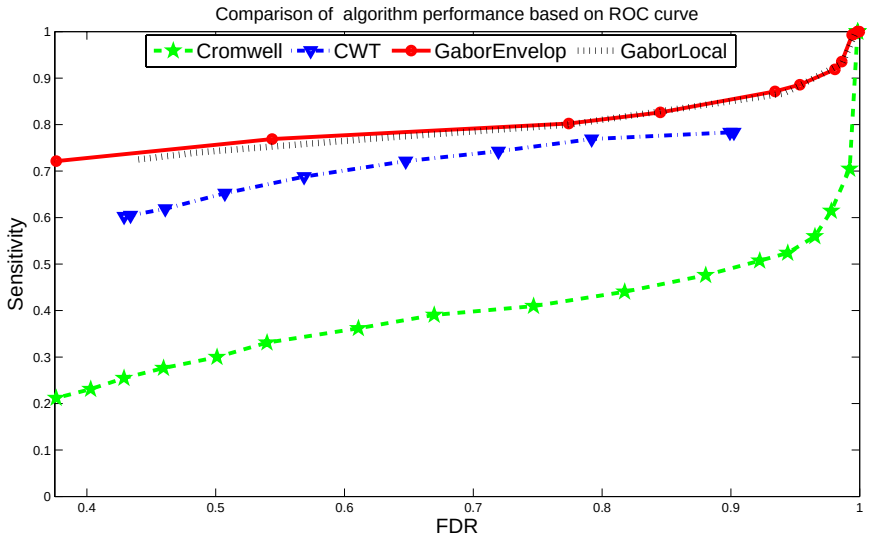


Figure 4.12. Average receiver operating characteristic (ROC) curves obtained from 60 MS signals using Cromwell, CWT, and GaborLocal and GaborEnvelop methods. The sensitivity is the true position rate.

shows the TPR and the FDR of four above methods with an assumption that there is only one charge. To calculate the ROC curve of Cromwell and CWT methods, the SNR thresholding values are changed. The SNR thresholding values are chosen from 0 to 20 for Cromwell method, from 0 to 65 for CWT method. In GaborLocal method, the threshold for the number of peaks is changed from 2000 to 10 to create the ROC curve. In GaborEnvelop method, the level is changed from seven to one to build the ROC curve. In Fig. 4.11, the performance of Cromwell method is much worse than CWT and GaborLocal and GaborEnvelop methods. Most of the ROC points of Cromwell method locate at the bottom of right corner and most of ROC points of CWT, GaborLocal, and GaborEnvelop methods are well placed on the top regions. In the proposed methods, some ROC points appear at the top line with $TPR = 1$ and some ROC points go with $TPR = 1$ and $FDR = 0$. However, it does not happen to the CWT. Therefore, GaborLocal and GaborEnvelop are better than CWT and Cromwell in peak detection.

If the average of those detailed ROC results is taken in Fig. 4.11, there is the average ROC curve as in Fig. 4.12. Notice that average of all ROC points is taken with the same SNR threshold (for Cromwell and CWT) and with the same peak threshold and the same level (for the proposed methods). From Fig. 4.12, the results of the proposed methods and CWT are much better than the Cromwell's one. Therefore, the decomposing approach without smoothing (SWT, GaborLocal and GaborEnvelop) is more efficient than the denoising approach (like Cromwell). At the same FDR, the TPRs of the proposed methods are consistently higher than the TPRs of CWT. Because the peak rank was used to identify peaks in the GaborLocal and GaborEnvelop methods instead of the SNR. It is clear that the utilizing peak rank to identify peak gives out valuable results. These methods have a significant contribution to detect both high energy and small energy peaks. Another advantage of these methods is that the threshold for the number of peaks can be created easier than the SNR. Therefore, the GaborLocal and GaborEnvelop method are more efficient and accurate methods for real MS data peak detection.

As shown in Fig. 4.12, GaborEnvelop is slightly better than GaborLocal in ROC curve. With the same TPR, GaborEnvelop gives out smaller FDR than GaborLocal. However, peak quantification step using many loops in GaborLocal method is simpler than using envelope analysis in GaborEnvelop one. If a simple method which can also detect almost true peaks is necessary, GaborLocal is a good option. During more complicated analysis, GaborEnvelop can be employed to improve the results of peak detection in MS signal. Since the number of detected peaks increase gradually when the peak rank increases, GaborEnvelop is useful for many applications, e.g. protein identification. They are also the reasons that two methods are proposed in this chapter.

4.6 Conclusion

In this chapter, two new approaches including GaborLocal and GaborEnvelop to solve peak detection problem in MS data with promising results have been proposed. GaborLocal method is a combination of the Gabor filter and Gaussian local maxima approach. The integration of Gabor filter and envelope analysis is further developed as GaborEnvelop method. The peak rank method is presented and used at the first time to replace the previous SNR method to identify true peaks. With real MS dataset, the proposed method gave much better performance in the ROC curve comparison with two other most common used peak detection methods.

CHAPTER 5
GAUSSIAN DERIVATIVE WAVELET BASED MASS SPECTROMETRY DATA PROCESSING

5.1 Introduction

Most peak detection methods employed denoising step by removing noise in each scale of wavelet, such as commonly used Cromwell ([49, 50]) and CWT ([51, 61]) methods. However, true peaks in MS could have large frequency response and be removed by denoising step. As a result, these true peaks cannot be detected. Bivariate shrinkage model, which considers relationship of two neighbor scales, is proposed to remove noise in stationary wavelet domain. Utilizing relationship between two neighbor coefficients or two scales of wavelets can keep high frequency true signal ([62]). Stationary wavelet transform (SWT) utilizes spatial information of signals and suppress artifacts by redundant representations.

Baseline removal step was widely used in peak detection methods, but it often got rid of true peaks and creates new false peaks. To avoid removing baseline, the CWT-based pattern-matching algorithm was introduced in [51]. Using Mexican Hat wavelet in multi-scale, this method gave good results in peak detection with high sensitivity and low false discovery rate (FDR). However, the more important property of multi-scale in wavelet domain was not used in this method ([44]). Instead of considering peaks as the sum of delta functions, more generally, MS peaks are considered as a mixture of Gaussian in which each peak corresponds to one Gaussian. Gaussian derivative wavelet is proposed to use instead of Mexican Hat wavelet which is only the second derivative of Gaussian wavelet. Zero-crossing lines which are robust to

noise are also introduced to replace Ridge-lines in [51]. The zero-crossing lines are studied in multi-scale wavelet and new theoretical analysis is presented.

In most of the peak detection methods, signal to noise ratio (SNR) was used to remove the small energy peaks with SNR values less than a threshold. But MS noise cannot be correctly estimated in either time domain or wavelet domain. Thus, in this chapter, instead of SNR, frequency response, height, and standard deviation of Gaussian peaks calculated by zero-crossing in Gaussian derivative wavelet domain are used to remove false peaks. In order to improve sensitivity, the Envelope analysis ([63]) is also used to save some important peaks which have small energy.

In this chapter, a new Gaussian derivative wavelet (GDWavelet) based peak detection method is proposed for Surface-Enhanced Laser Desorption/Ionization Time-Of-Flight (SELDI-TOF) spectrum. Both simulated and real spectrum with known polypeptide compositions and positions are used to evaluate proposed method. With simulated data, different peak detection algorithms are compared by both Gaussian noise and real noise. All experimental results show that new approach can detect more peaks (in both high amplitude and low amplitude) with a lower false discovery rate than state-of-the-art methods.

In this chapter, new Gaussian derivative wavelet based method (GDWavelet) will be introduced. In GDWavelet, bivariate smoothing model, Gaussian derivative wavelet, and envelope analysis are used. First, bivariate shrinkage estimator in SWT domain will be used to reduce noise and to keep whole true signal. Second, how to detect peaks using Gaussian derivative wavelet through peak properties such as frequency response, standard deviation and height will be introduced. Finally, envelope analysis is performed to save true small energy peaks which will be missed if only peak properties are used.

5.2 Smoothing by Bivariate Shrinkage Function

Noise smoothing in MS is an important step which should remove noise and keep true peaks. In Myers *et al.* (2004), they tried to remove noise as much as possible. However, in that case, some true peaks are also removed. Utilizing bivariate shrinkage estimator in SWT domain is proposed to reduce noise and keep whole true signal. More precisely, the noise level is decreased without removing most of them. SWT is chosen due to its fast speed and redundant representations. The later step will further handle the remain noise.

The MAP estimator of w_1 ([1]) is written as

$$\hat{w_1} = \begin{cases} 0 & \text{if } \sqrt{|y_1|^2 + |y_2|^2} < \frac{\sqrt{3}{\sigma_n}^2}{\sigma}, \\ \dfrac{\sqrt{|y_1|^2 + |y_2|^2} - \dfrac{\sqrt{3}{\sigma_n}^2}{\sigma}}{\sqrt{|y_1|^2 + |y_2|^2}}.y_1 & \text{otherwise.} \end{cases} \tag{5.1}$$

where y_1 is noisy child coefficient, y_2 is noisy parent coefficient.This estimator is a bivariate shrinkage function. It has been used to smooth many kinds of signals such as image ([1]), DNA copy number ([42, 43]), *etc.* In this chapter, bivariate shrinkage estimator is used to smooth MS signals. An example of denoising result is shown in Fig. 5.3(a). This example will be discussed in § 5.5.

5.3 Peak Detection by Gaussian Derivative Wavelet

In previous work ([49, 50]), MS peaks were considered as the sum of delta functions. That means only heights of peaks have been used for peak detection throughout SNR. Du *et al.* (2006) utilized width of peaks to improve peak detection results a lot. MS peaks are considered as a mixture of Gaussian in which each peak corresponds to one Gaussian:

$$f(t) = \sum_{i=1}^{N} f_i(t) = \sum_{i=1}^{N} A_i \; exp(-\frac{(t-\mu_i)^2}{2\sigma_i^2}). \tag{5.2}$$

With this assumption, four parameters provide intrinsic differences between true peaks and noise. They are peak position, standard deviation, height, and frequency response of peak. To find these parameters of a peak, using zero-crossing lines in multi-scale of Gaussian derivative wavelet is proposed instead of ridge-lines in multi-scale of Mexican hat wavelet that was used by [51].

5.3.1 Theory of Zero-Crossing Lines in Multi-Scale

Scaling theory for zero-crossings has been studied and applied to many applications. Yuille *et al.* ([64]) assumed that signal is the sum of delta functions. Another similar assumption of signal, bandlimited signal, has been studied in [65]. However, studying zero-crossing of signals with Gaussian mixture assumption still is a new and challenging problem. New theory of zero-crossing lines is built in multi-Scale in following sections. Through proposed theory, parameters (position, standard deviation, height, and frequency response) of a Gaussian peak can be accurately estimated.

The first derivative of $f_i(t)$ is used to locate local maxima corresponding to peak position: $f_i'(t_0) = 0$ with $t_0 = \mu_i$. The second derivative and third derivative of $f_i(t)$ continue to be used to estimate height and standard deviation of Gaussian peak: $f_i''(t_0) = 0$ with $t_0 = \mu_i \pm \sigma_i$, $f_i''(t_0) = 0$ with $t_0 = \mu_i$ and $t_0 = \mu_i \pm \sqrt{3}\sigma_i$.

Because smoothing was performed in denoising step just to reduce noise and to keep small true peaks, multi-scales of Gaussian derivative wavelet are used to make local maxima and minima more robust to noise instead of only one Gaussian filter in [63]. The wavelet transform can be written as convolution product in Eq. 5.3:

$$Wf(u,s) = \int_{-\infty}^{+\infty} f_i(t) \frac{1}{\sqrt{s}} \Psi^{\star}(\frac{t-u}{s}) dt, \tag{5.3}$$

where $\star$ is the conjugate. According to chapter 6 in [44], the wavelet transform in Eq. 5.3 can be rewritten as a multi-scale differential operator in Eq. 5.4

$$W_n f(u,s) = s^n \frac{d^n}{du^n}(f_i * \bar{\theta}_s(t))(u), \tag{5.4}$$

where * is convolution. In this paper, the Gaussian wavelet is used. So, $\bar{\theta}_s(t)$ can be followed as Eq. 5.5:

$$\bar{\theta}_s(t) = \frac{1}{\sqrt{s}} exp(-\frac{t^2}{s^2}). \tag{5.5}$$

Convoluting $f_i(t)$ and $\bar{\theta}_s(t)$, one gets result in Eq. 5.6

$$(f_i * \bar{\theta}_s)(u) = K_1 \ exp(-K_2(u-\mu_i)^2), \tag{5.6}$$

where $K_1 = A\sqrt{\frac{1}{2\pi\sigma_i^2 s^3}}$ and $K_2 = \frac{1}{s^2+2\sigma_i^2}$.

Remark: The zero-crossing points of $W_1 f(u,s)$ and $W_2 f(u,s)$ belong to connected curves that are never interrupted when the scale decreases.

Proof: With the first derivative, Eq. 5.4 can be rewritten as Eq. 5.7

$$W_1 f(u,s) = 2sK_1K_2(u-\mu_i) \ exp(-K_2(u-\mu_i)^2) \tag{5.7}$$

if $W_1 f(u,s) = 0$, it got $u_0 = \mu_i$, then $u_0(s+1) - u_0(s) = 0$ with any scale s. With the second derivative, Eq. 5.4 can be rewritten as Eq. 5.8

$$W_2 f(u,s) = 2s^2K_1K_2[-2K_2(u-\mu_i)+1] \ exp(-K_2(u-\mu_i)^2). \tag{5.8}$$

If $W_2 f(u,s) = 0$, one gets $u_0 = \mu_i \pm \sqrt{\sigma_i^2 + \frac{s^2}{2}}$, then $0 < u_0(s+1) - u_0(s) <= 1$ with any scale s.

With the third derivative, Eq. 5.4 can be rewritten as Eq. 5.9

$$W_3 f(u,s) = -2s^3K_1K_2(u-\mu_i)[2K_2(u-\mu_i)^2 - 3] \ exp(-K_2(u-\mu_i)^2) \tag{5.9}$$

If $W_3 f(u,s) = 0$, one gets $u_0 = \mu_i$ *or* $u_0 = \mu_i \pm \sqrt{3}\sqrt{\sigma_i^2 + \frac{s^2}{2}}$. If $s = 100$ and $\sigma_i = 0.1$ are selected then $u_0(100+1) - u_0(100) = 1.2247$. In conclusion, $0 <= u_0(s+$

$1) - u_0(s) <= 1$ with the first and second derivative and zero-crossing lines belong to connected curves. Another conclusion is that zero-crossing lines is discontinuous lines if the third derivative Gaussian wavelet is used. Thus, only the first and second derivative Gaussian wavelets should be used in peak detection.

If f_i is a discrete signal, Eq. 5.3 can be rewritten as follows:

$$Wf(u,s) = \sum_k f_i(k) \int_k^{K+1} \frac{1}{\sqrt{s}} \Psi^\star(\frac{t-u}{s}) dt. \tag{5.10}$$

One gets $f(k)$ by sampling $f_i(t)$ with T_s:

$$f_i(k) = f_i(kT_s) = A_i \ exp(-\frac{(k-\frac{\mu_i}{T_s})^2}{2(\frac{\sigma_i}{T_s})^2}). \tag{5.11}$$

If $W_2 f(u,s) = 0$, it gets $u_0 = \mu_i \pm \sqrt{\sigma_i^2 + \frac{(s \times T_s)^2}{2}}$. if $W_3 f(u,s) = 0$, it gets $u_0 = \mu_i \ or \ u_0 = \mu_i \pm \sqrt{3}\sqrt{\sigma_i^2 + \frac{(s \times T_s)^2}{2}}$.

Note: Zero-crossing line is more robust to noise than ridge-line. This conclusion is illustrated in an example in Fig. 5.1. Fig. 5.1(c)(e) show that zero-crossing lines are detected easier than ridge lines linking local maxima or local minima points.

5.3.2 Application of Zero-Crossing to Peak Detection

From section 5.3.1, parameters of a Gaussian peak could be estimated as follows

Estimation of Peak Position: There are three ways to estimate peak positions throughout zero-crossing of three kind Gaussian derivative wavelets.

1. The First Gaussian Derivative Wavelet (Gaus1): a zero-crossing line corresponds to a peak position. In multi-scale, this zero-crossing line is a continuous line with length N. Peak position should be estimated by

$$\mu_i = \frac{1}{N} \sum_{s=1}^{N} u_0(s). \tag{5.12}$$

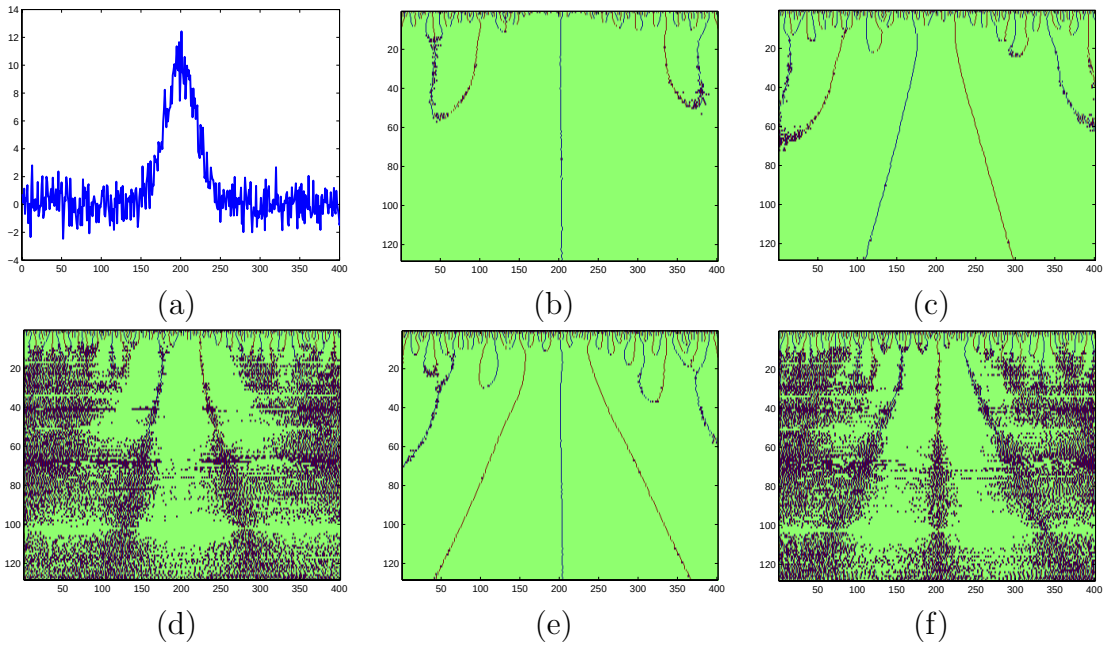


Figure 5.1. An illustration of zero-crossing line and ridge line comparison: (a) A peak sample whose shape follows ($10\ exp(-\frac{(t-5)^2}{2\times 0.5^2})$) with Gaussian noise (std=1); (b) By using $Gaus1$, the zero-crossing line corresponds to peak position, $t = 5$; (c) By using $Gaus2$, two zero-crossing lines correspond to two peak edges whose distances to peak position are $\sigma_i = 0.5$; (d) By using $Gaus1$, two ridge lines are corresponding to two peak edges whose distances to peak position are $\sigma_i = 0.5$; (e) By using $Gaus3$, three zero-crossing lines are corresponding to one peak position and two peak edges whose distances to peak position are $\sqrt{3}\sigma_i = 0.866$; (f) By using $Gaus2$, three ridge lines are corresponding to one peak position and two peak edges whose distances to peak position are $\sqrt{3}\sigma_i = 0.866$.

2. The Second Gaussian Derivative Wavelet (Gaus2): There are two zero-crossing lines corresponding to two edges of Gaussian peak. They are u_{0left} and u_{0right}. Because two zero-crossing lines are symmetric at peak position, peak position should be estimated by

$$\mu_i = \frac{1}{N}\sum_{s=1}^{N}\frac{u_{0left}(s) + u_{0right}(s)}{2}. \tag{5.13}$$

3. The Third Gaussian Derivative Wavelet (Gaus3): we get three zero-crossing lines if using the third Gaussian derivative wavelet. They are u_{0left}, u_{0midle} and

u_{0right}. Because u_{0left} and u_{0right} are non-continuous lines, they should not be used to estimate peak position. From u_{0midle}, peak position can be found by

$$\mu_i = \frac{1}{N}\sum_{s=1}^{N} u_{0midle}(s). \tag{5.14}$$

Estimation of Peak's Standard Deviation: Another parameter of Gaussian peak is standard deviation σ_i. There are two ways to estimate σ_i as follows

1. The Second Gaussian Derivative Wavelet (Gaus2): From Remark, σ_i at scale s can be calculated by

$$\sigma_{i-left}(s) = \sqrt{(u_{0left}(s) - \mu_i)^2 - \frac{s^2}{2}}, \tag{5.15}$$

$$\sigma_{i-right}(s) = \sqrt{(u_{0right}(s) - \mu_i)^2 - \frac{s^2}{2}}. \tag{5.16}$$

After $\sigma_{i-left}(s)$ and $\sigma_{i-right}(s)$ are calculated at all scales, σ_i should be estimated by

$$\sigma_i = \frac{\frac{1}{N_l}\sum_{s=1}^{N_l}\sigma_{i-left}(s) + \frac{1}{N_r}\sum_{s=1}^{N_r}\sigma_{i-right}(s)}{2}, \tag{5.17}$$

where N_l and N_r are length of left and right zero-crossing lines.

2. The Third Gaussian Derivative Wavelet (Gaus3): Also from Remark, σ_i at scale s can be calculated by

$$\sigma_{i-left}(s) = \sqrt{\frac{1}{3}(u_{0left}(s) - \mu_i)^2 - \frac{s^2}{2}}, \tag{5.18}$$

$$\sigma_{i-right}(s) = \sqrt{\frac{1}{3}(u_{0right}(s) - \mu_i)^2 - \frac{s^2}{2}}. \tag{5.19}$$

After $\sigma_{i-left}(s)$ and $\sigma_{i-right}(s)$ are calculated at all scales, σ_i should be estimated by

$$\sigma_i = \frac{\frac{1}{N_l}\sum_{s=1}^{N_l}\sigma_{i-left}(s) + \frac{1}{N_r}\sum_{s=1}^{N_r}\sigma_{i-right}(s)}{2}, \tag{5.20}$$

where N_l and N_r are the lengths of left and right zero-crossing lines. However, zero-crossing lines at left and right sides of the third Gaussian derivative wavelet

are disconnected lines, so it is not easy to estimate σ_i through Eqs. 5.18, 5.19 and 5.20.

Estimation of Peak Height: Finally, a way to estimate real height of Gaussian peak is developed. With Gaussian peak $f_i(t) = A_i\ exp(-\frac{(t-\mu_i)^2}{2\sigma_i^2})$, it has

$$A_i = \frac{f_i(\mu_i) - f_i(\mu_i - \sigma_i)}{0.3935}. \tag{5.21}$$

Eq. 5.21 can be used to estimate height of Gaussian peak after μ_i and σ_i are calculated.

An Example: To illustrate the above theory, a Gaussian peak is assumed as follows

$$x(t) = A_x\ exp(-\frac{(t-\mu_x)^2}{2 \times {\sigma_x}^2}), \tag{5.22}$$

where $A_x = 10$, $\mu_x = 5$ and $\sigma_x = 0.5$. Gaussian noise and baseline are added as follows

$$f(t) = x(t) + G(\sigma, \mu) + b, \tag{5.23}$$

where b is a constant, a representation of base line, $\mu = 0$ and $\sigma = [0.25; 0.5; 0.75; 1]$. With each σ value, 200 signals, $f(t)$, have been created. One sample $f(t)$ is shown in Fig. 5.1(a). μ_x, σ_x and A_x are estimated by using above zero-crossing theory. Error rate which is defined in Eq. 5.24 will be used to compare accuracy of different estimation methods:

$$error\ rate = \frac{|true\ value - estimated\ value|}{true\ value} \times 100. \tag{5.24}$$

Fig. 5.1(b), (c), and (e) show zero-crossing lines in 128 scales using $Gaus1$, $Gaus2$ and $Gaus3$. These zero-crossing lines will be used to estimate μ_x, σ_x and A_x. Table. 5.1 lists error rates of four methods to estimate peak position μ_x. With $Gaus1$, $Gaus2$, and $Gaus3$ method, μ_x values are calculated by Eq. 5.12, Eq. 5.13, and Eq. 5.14 correspondingly. The term "with denoise" means bivariate shrinkage

Table 5.1. Error of Peak Position Estimation. By using zero-crossing lines in multi-scale of Gaussian derivative wavelet, there are three ways to estimate peak position as in Eq. 5.12, Eq. 5.13 and Eq. 5.14. Errors of these estimations and CWT's estimation ([51, 61]) are compared. The error rate is defined by Eq. 5.24. In each Gaussian noise level, σ, two hundred signals have been created. Error value shown in this table is average value.

σ in Eq. 5.23	Gaus1 without Denoise	Gaus1 with Denoise	Gaus2 without Denoise	Gaus2 with Denoise	Gaus3 without Denoise	Gaus3 with Denoise	Mexh ([51, 61])
0.25	0.0519	0.0365	0.1533	0.1434	0.4890	0.2652	1.979
0.50	0.1319	0.0809	0.2253	0.1943	0.6918	0.3851	2.0170
0.75	0.1658	0.1034	0.3382	0.2353	0.7008	0.4855	2.1137
1.00	0.2118	0.1469	0.4630	0.2672	0.8681	0.5874	2.1618

estimator is used to denoise Gaussian noise in signal $f(t)$. The Mexh, Mexican hat wavelet, which corresponds to $Gaus2$, is used as core part to detect peak in CWT method ([51, 61]) and peak tree method ([66]). Based on the table. 5.1's result, the error rate when using Mexh wavelet ([51, 61]) is the largest. Note that Package "MassSpecWavelet" ([60]) which uses denoising with DWT ([61]) and finds peak position using ridge lines ([51]) with Mexh wavelet is used. With "Gaus1 with denoise", error rate is the smallest. However, error rates in Gaus1 without denoising and in Gaus2 are still acceptable and much better than in Mexh wavelet.

The σ_x can be estimated by Eq. 5.17 or Eq. 5.20. However, with $Gaus3$, zero-crossing lines are not continuous lines (see Remark in section 5.3.1). Thus, estimation of zero-crossing in 128 scales of $Gaus3$ is a problem. This problem causes a larger error in calculating the σ_x. From result of Table. 5.2, $Gaus2$ with denoising should be used to estimate σ_x because its' error rate is the smallest.

By using Eq. 5.21 with zero-crossing lines of both $Gaus2$ and $Gaus3$, the height of Gaussian peak is estimated. In this case, baseline b which is used in Eq. 5.23 is

Table 5.2. Error of Peak 's Standard Deviation Estimation. σ_x can be estimated by Eq. 5.17 with $Gaus2$ or Eq. 5.20 with $Gaus3$. Error rate here is defined by Eq. 5.24. These error values are average values which are gotten from 200 signals with each added Gaussian noise level, σ.

σ in Eq. 5.23	Gaus2 without Denoise	Gaus2 with Denoise	Gaus3 without Denoise
0.25	1.6560	1.3829	2.3371
0.50	2.5626	2.3392	3.7318
0.75	3.3841	2.5087	4.7881
1.00	3.9726	2.8529	5.9220

Table 5.3. Error of Peak 's Height Estimation. Peak height A_x can be calculated by Eq. 5.21. Error rate here is defined by Eq. 5.24. These error values are average values which are gotten from 200 signals, with each added Gaussian noise level, σ.

σ in Eq. 5.23	Gaus2 without Denoise	Gaus2 with Denoise	Gaus3 without Denoise
0.25	4.1032	1.7544	4.8886
0.50	7.8084	2.6869	8.2126
0.75	11.0612	2.8954	14.3860
1.00	13.6141	3.0502	16.9405

a constant. From Table. 5.3, $Gaus2$ with denoising gives the smallest error rate and should be used to calculate A_x.

From above example, the best way to estimate peak position μ_x is from the first Gaussian derivative wavelet, $Gaus1$. The second Gaussian derivative wavelet, $Gaus2$, should be used to estimate standard deviation σ_x and height A_x of a Gaussian peak. Fig. 5.1(d)(f) show Ridge lines which correspond to zero-crossing lines in Fig. 5.1(c)(e). It is clearly that detecting Ridge-lines is more difficult than detecting zero-crossing lines. Ridge-lines in Du *et al.* (2006) are similar to Ridge-lines in Fig. 5.1(f), corresponding to zero-crossing line in $Gaus3$ which should not be used because of its high error in calculating parameters of peaks.

5.4 Saving Small Energy Peaks by Envelope Analysis

Envelope analysis has been introduced by [63]. Any finite energy signal $y(t)$ can be analyzed into three envelope signals including MAX, MIN, and MED envelops at the first level. Each of these envelops can be considered as a signal and will be decomposed into three envelopes. In this chapter, MAX and MED envelops are used to detect peaks because MIN envelops contain no peak. The original signal is decomposed into one MAX envelop at level 1, one MAX and one MED envelops at level 2 and four envelops which comprise two MAX envelops and two MED envelops at level $n > 2$.

5.5 Gaussian Derivative Wavelet based Method (GDWavelet)

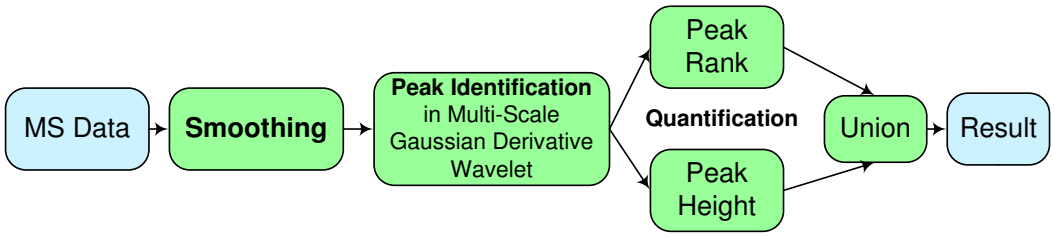


Figure 5.2. GDWavelet Method's Flowchart: Raw MS data is smoothed by bivariate shrinkage estimator in SWT domain to keep true signal and to reduce noise. Without removing baseline, smoothed signal is used in next step to estimate parameters of peaks by zero-crossing lines in multi-scale Gaussian derivative wavelet domain. After removing peaks whose zero-crossing line's length and width are less than a threshold, all peak candidates are obtained. All peak candidates will be quantified by peak rank in envelop analysis and peak height. Union results are final output.

The framework of the proposed GDWavelet method is shown in Fig. 5.2. First, raw MS data is smoothed by bivariate shrinkage estimator (Eq. 5.1) in SWT domain to keep true signal and to reduce noise. Note that, the lowest frequency detail scale and approximate scale which may include true signal should not be applied with any

estimator, so that true signal will not be removed. As a result, noise in signal is reduced and smoothed signal still has a little noise. Second, without being applied with baseline removal that often discards true peaks and creates some new peaks, smoothed signal is used to estimate frequency response, position, height, and standard deviation of Gaussian peak by zero-crossing lines in multi-scale Gaussian derivative wavelet domain. Zero-crossing lines are more robust to noise than ridge lines ([51]). Frequency response of Gaussian peak is proportional to the length of zero-crossing line if the first derivative Gaussian (Gaus1) is used. Peak position, μ_i, is estimated by Eq. 5.12. Standard deviation, σ_i, of Gaussian peak is calculated by Eq. 5.17. Result of Eq. 5.21 with $Gaus2$ will give heights of peaks. Using the first and the second derivative Gaussian wavelet, all parameters of a Gaussian peak can be estimated. After peaks whose frequency response and standard deviation are less than a threshold are removed, all peak candidates are obtained. Third, in peak quantification step, two rules are used to remove false peaks: 1) All peak candidates are quantified by peak rank (PR) ([63]) in envelop analysis. Peaks with $PR = 1$, even small peaks, are important peaks. 2) Peak height is used to remove peaks with height smaller than threshold. Peak height is used to substitute SNR that was used by Morris *et al.* (2005) and Du *et al.* (2006), because noise cannot be exactly estimated in either time domain or wavelet domain. Finally, the union results of two quantifying rules are the final detected peaks.

The 19^{th} sample of [57] is selected randomly to illustrate how GDWavelet method operates to detect peaks in MS signal. In Fig. 5.3(a), blue signal represents raw signal and red one is signal smoothed by using Eq. 5.1. A zoom in subfigure draws the peak which is used to show its' zero-crossing lines in Fig. 5.3(b). Using one zero-crossing line in multi-scale of the $Gaus1$ and two zero-crossing lines in multi-scale of the $Gaus2$, position, height, standard deviation, and frequency response of

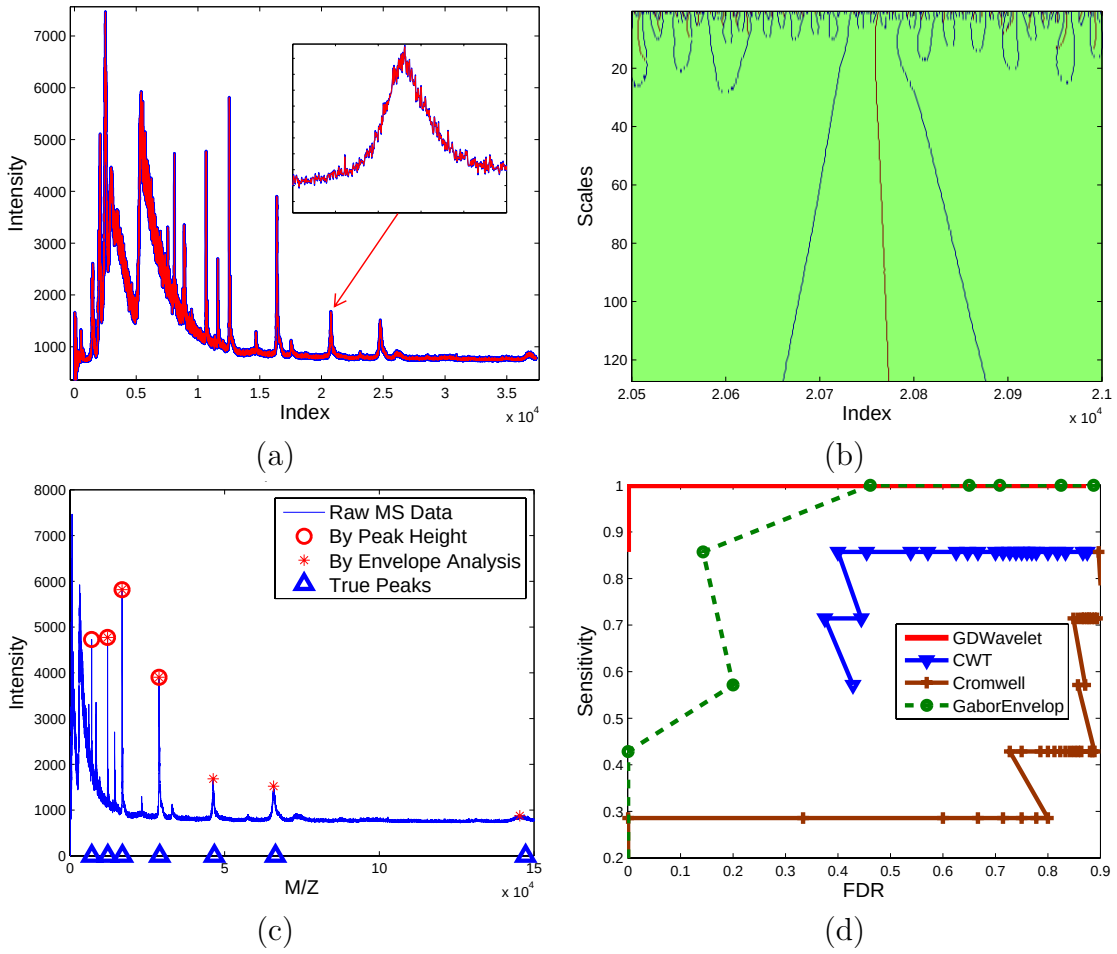


Figure 5.3. An Illustration of GDWavelet: the 19^{th} sample of [57] dataset is selected to illustrate how GDWavelet method operates to detect peak in MS signal. (a) Blue signal line represents for a raw signal and red one is a signal smoothed by bivariate shrinkage estimator in wavelet domain. (b) Parameters of peaks are estimated by zero-crossing lines. This figure shows zero-crossing lines of only one zoom peak in figure (a). (c) Peaks are quantified by two rules: peak height and peak rank (in Envelope analysis). Union results include peaks whose heights are larger than a threshold and peak ranks equal to one. (d) ROC curves of four methods' performance. With this sample, GDWavlet yields the best performance.

this peak are estimated. In Fig. 5.3(c), peaks are quantified by two rules: peak height and peak rank (in envelope analysis). The circles are results from peak height based quantification. The stars are from peak rank based quantification. Union results include all peaks with heights larger than a threshold or peak rank one. Fig. 5.3(d)

shows ROC curves of four methods' performance that will discussed in next section. GDWavlet gives the best performance.

5.6 Experiments and Discussions

5.6.1 Experimental Setup

Cruz-Marcelo *et al.*(2008) and Emanuele *et al.* (2009) presented the results of an extended and up-to-date study that compares the performance of current popular methods for SELDI data pre-processing, including CWT ([51]), Cromwell ([49, 50]), PROcess ([67]), Ciphergen and SpecAlign ([68]). They concluded that CWT ([51]) has the best performance in term of peak detection. Another method which also works well is Cromwell ([49, 50]). In this section, GDWavelet method will be compared to the Cromwell ([49, 50]), the CWT ([51]), and the previous method, GaborEnvelop ([63]). These methods are state-of-the-art MS peak detection methods. Cromwell method is implemented by MATLAB and can be downloaded from [59]. The CWT method ([51]) has been implemented in R (called 'MassSpecWavelet') and Version 1.12 can be downloaded from [60]. GaborEnvelop ([63]) is implemented in MATLAB.

The performances of above methods are evaluated by the ROC curve that is the standard validation criterion. Both simulated and real data are used in this chapter. The first simulated data was proposed by Morris *et al.* (2005) and Coombes *et al.* (2005) and are available for download at [69]. In this data, hundreds of mean spectrum samples with hundreds of proteomics datasets are generated.

Based on the simulation engine developed by [49] and code (R and MATLAB) to generate simulated data proposed by Cruz-Marcelo *et al.* (2008) and Zhang *et al.* (2009), two simulated datasets are also used to investigate noise affection on different algorithms. The 100 spectrums with $20 - 30$ true peaks are created first,

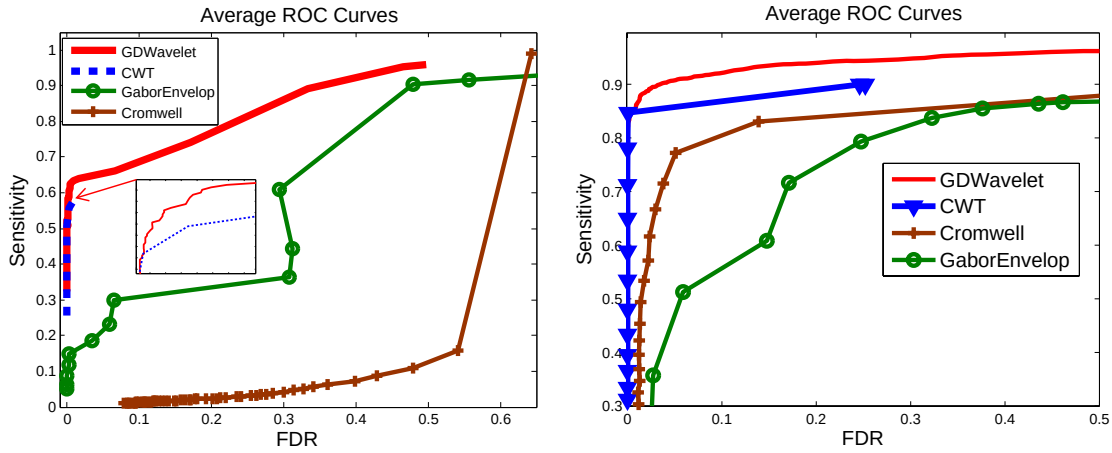
and Gaussian and real noise are added separately to get two datasets. When Gaussian noise is added, each sample includes 20% of protein peaks which are below three of SNR. Real noise is extracted from real data in which there is not any true peaks. There is only noise from 26000 (index) to end in the 1^{st} sample of [57]. Real noise probability density function is built. Using this function, noise with different standard deviation will be created. Based on this configuration, $20 - 30$ true peaks and more small energy peaks are created in simulated data.

The CAMDA dataset (2006) of all-in-1 Protein Standard II (Ciphergen Cat. # $C100 - 007$) is the real dataset used in this chapter. Because polypeptide composition and position are known in this dataset, the sensitivity and the false discovery rate (FDR) can be estimated. Another advantage of this dataset is that they are real data and better than the simulated data in evaluation. There are seven polypeptides which create seven true peaks at 7034, 12230, 16951, 29023, 46671, 66433, and 147300 of the m/z values.

The sensitivity and FDR of four methods are calculated for 60 real MS signals and three simulated MS datasets with 100 signals each. The SNR thresholding values are increased gradually to calculate the ROC curves of Cromwell and CWT methods. The SNR thresholding values are chosen from 0 to 20 for Cromwell method and from 0 to 120 for CWT method. In GDWavelet method, the peak height ratio, which is defined as the ratio of current peak height over average height of peaks, is changed from zero to ten to build the ROC curve. The average ROC curves are plotted in Fig. 5.4 and Fig. 5.5. Notice that average of all ROC points is taken with the same SNR threshold (for Cromwell and CWT) and the same peak height rate (for GDWavelet method).

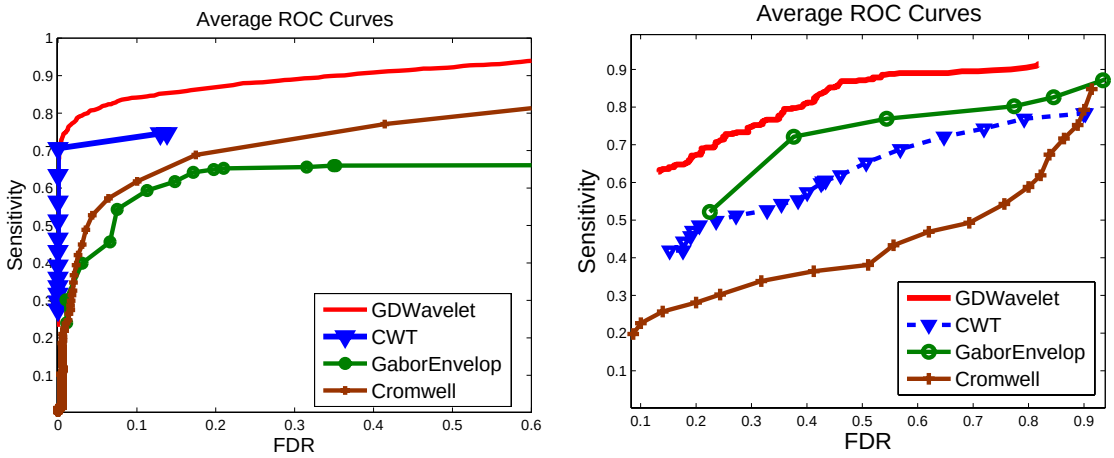
5.6.2 Experimental Results

Three simulated datasets and one real SELDI-TOF dataset are used to create ROC curves in Fig. 5.4 and Fig. 5.5. In all four datasets, the performance of Cromwell is not stable and gets worse than CWT and GDWavelet. Between GaborEnvelop used envelope analysis and CWT used ridge lines, GaborEnvelop is better than CWT in real data in Fig. 5.4 (b). However, CWT is better than GaborEnvelop in simulated data. In all cases, GDWavelet method has much better performance than GaborEnvelop and CWT methods. At the same FDR, the sensitivity of proposed method is consistently higher than GaborEnvelop's and CWT's sensitivity. It is clear that utilizing both of envelope analysis and Gaussian derivative wavelet in peak quantification made a significant contribution to detect both high energy and small energy peaks. Note that GDWavelet is designed from three nice techniques such as wavelet denoising, multi-scale wavelet, and envelop analysis ([63]). Bivariate shrinkage estimator in wavelet domain guarantees that denoising step in the proposed method saves true signal much better than [49]. Zero-crossing lines based peak parameters estimations in this chapter are more accurate and robust to noise than ridge lines in [51]. Envelope analysis is more efficiently used in GDWavelet than in GaborEnvelop. Therefore, the GDWavelet has better peak detection results than Cromwell, GaborEnvelop, and CWT. Thus, it is an efficient and accuracy method to detect peaks in both real and simulated MS data. In Fig. 5.4 and 5.5, CWT's ROC curves is limited in small FDR because two thresholds of the length of ridge lines and the scale corresponding to the maximum amplitude on the ridge line are used as default in MassSpecWavelet ([60]).



(a) (b)

Figure 5.4. ROC Curves - Simulated data with Gaussian noise: Average ROC curves of four methods (Cromwell, CWT, GaborEvelop, and GDWavelet). (a) Obtained from 100 mean simulated MS signals which can be downloaded from [69]. There are 149 true peaks in this data. (b) Obtained from 100 simulated MS signals in which Gaussian noise is added. There are $20-30$ true peaks in this data.



(a) (b)

Figure 5.5. ROC Curves - Simulated data with real noise and real data: Average ROC curves of four methods (Cromwell, CWT, GaborEvelop, and GDWavelet). (a) Obtained from 100 simulated MS signals in which real noise is used. There are $20-30$ true peaks in this data. (b) Obtained from 60 MS signals ([57]). There are seven true peaks in this data.

5.7 Conclusion

In this chapter, new zero-crossing line theory in multi-scale of Gaussian derivative wavelet has been proposed to estimate parameters of peaks in mass spectrometry

which has been assumed as a mixture of Gaussian. A novel GDWavelet method was proposed to efficiently and accurately detect MS peaks by integrating bivariate shrinkage model, Gaussian derivative, and envelope analysis. The bivariate shrinkage estimator in SWT domain was used to reduce noise and still keep true peaks. All parameters of a Gaussian peak, estimated by multi-scale in Gaussian derivative wavelet and envelope analysis, have been used to remove false peaks. The peak height and peak rank were introduced as a nice substitution of the previous SNR method to identify true peaks. Both simulated data and real MS data are used to evaluate GDWavelet method. Simulated data was created with both Gaussian noise and real noise. The GDWavelet method gave out a much better performance in the ROC curves than three other state-of-the-art peak detection methods. Based on GDWavelet method, many MS data related applications will be improved, such as protein identification, biomarker discovery, cancer classification, *etc.*

CHAPTER 6

CONCLUSION

6.1 Array-CGH

In this dissertation, at the chapter 2, the stationary wavelet packet transform method has been explored with the new bivariate shrinkage estimators in array CGH data denoising study. The dependent Laplacian bivariate shrinkage estimator has been proposed to improve the SWPT method in aCGH data denoising study. In experiments, the denoising results of SWPT-LaBi method are much better than the previous methods in terms of the root mean squared error measurement and the ROC curve at different Gaussian noise levels. The denoising results from SWPT-AdaBi are much better than previous methods. Furthermore, real noise is also used to improve the traditional aCGH synthetic data generation. Since new synthetic data generation model is a better approximation of real aCGH data, it can more accurately evaluate the aCGH smoothing algorithms. This new synthetic aCGH with real noise is also exploited in evaluation, and proposed method still outperforms others. Meantime, the real aCGH data is also used to demonstrate proposed approach is better than other most common used smoothing methods.

In chapter 3, noise distribution has been studied in array CGH data using five real datasets in many platforms with different resolutions. As discussed, almost all previous array CGH data processing and analysis methods assumed that the noise PDF is Gaussian. However the recent work and experimental result show that array CGH noise is heavy-tailed noise. When compared with other distributions used in previous research such as Gaussian and Student's t distributions, generalized Gaus-

sian distribution fits very well noise PDF in the array CGH data. Therefore, using GGD for modeling noise assumption is used in the array CGH data and a novel smoothing-segmentation method based on this generalized Gaussian noise has been proposed. Bivariate shrinkage function's theory in SWT is built with an approach to suppress heavy-tailed noise in array CGH. One-directional Gaussian wavelet derivative scalogram is defined and proposed to detect breakpoints in array CGH. Because the ground truth aberration regions are not clear in real array CGH datasets, synthetic array CGH data plays an important role in array CGH analysis algorithm evaluation. By using generalized Gaussian noise and real noise, the synthetic array CGH data models which are closer to the real array CGH data than the most commonly used standard [15] and [34] have been improved. Both synthetic data and real data are used to evaluate the performance of proposed method, DWSS. New method outperforms other most commonly used algorithms in array CGH literature both in terms of RMSE and ROC curve.

6.2 Mass Spectrometry

Mass spectrometry data analysis has been discussed in chapter 4. In the results of traditional MS peak detection algorithms, there exist massive true negative and false positive, *i.e.*, the real peaks with small amplitude are easily missed and the false peaks with high amplitude are always detected. In this dissertation, a new complex Gabor-Envelope approach has been proposed to solve these problems with promising results. Gaborlocal and GaborEnvelop methods are a combination of the complex Gabor filter and envelope analysis approaches. Envelope analysis is proposed as a new theory to discover important signals from a large range of frequency. Most of all peaks in mass spectrometry are visible after using complex Gabor filters. Envelope analysis is employed to classify peaks into many groups corresponding their importance. The

peak ranking method is introduced and used at the first time to replace the previous SNR method to identify true peaks. The real MS dataset, the CAMDA 2006 data, was used in validations. The proposed method gives out a much better performance in the ROC curve comparison with two other state-of-the-art peak detection methods.

In chapter 5, new zero-crossing line theory in multi-scale of Gaussian derivative wavelet has been proposed to estimate parameters of peaks in mass spectrometry which has been assumed as a mixture of Gaussian. A novel GDWavelet method was proposed to efficiently and accurately detect MS peaks by integrating bivariate shrinkage model, Gaussian derivative, and envelope analysis. The bivariate shrinkage estimator in SWT domain was used to reduce noise and still keep true peaks. All parameters of a Gaussian peak, estimated by multi-scale in Gaussian derivative wavelet and envelope analysis, have been used to remove false peaks. The peak height and peak rank were introduced as a nice substitution of the previous SNR method to identify true peaks. Both simulated data and real MS data are used to evaluate GDWavelet method. Simulated data was created with both Gaussian noise and real noise. GDWavelet method gave out a much better performance in the ROC curves than three other state-of-the-art peak detection methods. Based on GDWavelet method, many MS data related applications will be improved, such as protein identification, biomarker discovery, cancer classification, *etc.*

APPENDIX A

ABBREVIATION LIST

1-D	One-Directional
aCGH	Array Comparative Genomic Hybridization
array CGH	Array Comparative Genomic Hybridization
BAC	Bacterial Artificial Chromosome
CBS	Circular Binary Segmentation Method
cDNA	complementary DNA
CWT	Continuous Wavelet Transform
DNA	Deoxyribonucleic acid
DTCWT	Dual-Tree Complex Wavelet Transform
DTCWTi	Dual-Tree Complex Wavelet Transform and Interpolation Method
DTCWTi-Bi	Dual-Tree Complex Wavelet Transform, Interpolation and Bivariate Shrinkage Function Method
DWPT	Discrete Wavelet Packet Transform
DWSS	Derivative Wavelet Scalogram Segmentation Method
DWT	Discrete Wavelet Transform
FDR	False Discovery Rate
FPR	False Position Rate
GaborEnvelop	Gabor filters and Envelope Analysis based Method
GaborLocal	Gabor filters and Gaussian Local Maxima based Method
GADA	Genome Alteration Dectection Analysis
Gaus1	First Derivative Gaussian Wavelet
Gaus2	Second Derivative Gaussian Wavelet
Gaus3	Third Derivative Gaussian Wavelet
GDWavelet	Gaussian Derivative Wavelet based Method

GGD	Generalized Gaussian Distribution
HaarSeg	Haar Wavelet - Segmentation Method
IID	Independent Identically Distributed
KLD	Kullback-Leibler Divergence
MAP	Maximum A Posteriori Probability
MAX	Maximal Envelope
MED	Median Envelope
MIN	Minimal Envelope
MS	Mass Spectrometry
PDF	Probability Density Function
Quantreg	Quantile Regression Method
RMSE	Root Mean Squared Error
ROC curve	Receivever Operating Characteristic curve
SELDI-TOF	Surface-enhanced laser desorption/ionization-time-of-flight
SmoothSeg	Smooth Segmentation Method
SNP	Single-Nucleotide Polymorphism
SNR	Signal to Noise Ratio
SWPT	Stationary Wavelet Packet Transform
SWPT-AdaBi	SWPT and Adaptive Bivariate Shrinkage based Method
SWPT-LaBi	SWPT and Laplacian Bivariate Shrinkage based Method
SWT	Stationary Wavelet Transform
SWTi	Stationary Wavelet Transform
	and Interpolation based Method
TPR	True Position Rate

REFERENCES

[1] L. Sendur and I. Selesnick, "Bivariate shinkage function for wavelet-based denoising exploiting interscale dependency," *IEEE Transactions on Signal Processing*, vol. 50, no. 11, pp. 2744–2756, November 2002.

[2] J. Pollack, T. Sorlie, C. Perou, C. Rees, *et al.*, "Microarray analysis reveals a major direct role of DNA copy number alteration in the transcriptional program of human breast tumors," *Proceeding of National Academy of Sciences*, vol. 99, pp. 12 963–12 968, 2002.

[3] D. Pinkel *et al.*, "High resolution analysis of DNA copy number variation using comparative genomic hybridization to microarrays," *Nature Genetics*, vol. 20, pp. 207–211, 1998.

[4] A. M. Snijders, N. Nowak, R. Segraves, S. Blackwood, *et al.*, "Assembly of microarrays for genome-wide measurement of DNA copy number," *Nature Genetics*, vol. 29, no. 3, pp. 263–264, 2001.

[5] C. Brennan, Y. Zhang, C. Leo, B. Fenga, *et al.*, "High-resolution global profiling of genomic alterations with long oligonucleotide microarray," *Cancer Research*, vol. 64, pp. 4744–4748, 2004.

[6] J. Pollack *et al.*, "Genome-wide analysis of DNA copy-number changes using cDNA microarrays," *Nature Genetics*, vol. 23, pp. 41–46, 1999.

[7] S. Bilke, Q. R. Chen, C. C. Whiteford, and J. Khan, "Detection of low level genomic alterations by comparative genomic hybridization based on cDNA microarrays," *Bioinformatics*, vol. 21, no. 7, pp. 1138–1145, 2005.

[8] W. Lai, M. Johnson, R. Kucherlapati, and P. Park, "Comparative analysis of algorithms for identifying amplifications and deletions in array CGH data," *Bioinformatics*, vol. 21, pp. 3763–3770, 2005.

[9] L.Hsu, SG.Self, D.Grove, T.Randolph, K.Wang, JJ.Delrow, L.Loo, and P.Porter, "Denoising array-based comparative genomic hybridization data using wavelets," *Biostatistics(Oxford,England)*, vol. 6, no. 2, pp. 211–226, 2005.

[10] P. Eilers and R. de Menezes, "Quantile smoothing of array CGH data," *Bioinformatics*, vol. 21, pp. 1146–1153, 2005.

[11] B. Beheshti, I. Braude, P. Marrano, P. Thorner, M. Zielenska, and J. Squire, "Chromosomal localization of DNA amplifications in neuroblastoma tumors using cDNA microarray comparative genomic hybridization," *Neoplasia*, vol. 5, pp. 53–62, 2003.

[12] R. Coifman and D. Donoho, "Translation-invariant de-noising," *Wavelets and Statistics*, vol. 103 of Lecture Notes in Statistics, pp. 125–150, 1995.

[13] Y. Wang and S. Wang, "A novel stationary wavelet denoising algorithm for array-based DNA copy number data," *International Journal of Bioinformatics Research and Applications*, vol. 3, no. 2, pp. 206 – 222, 2007.

[14] N. Nguyen, H. Huang, S. Oraintara, and A. Vo, "A new smoothing model for analyzing array CGH data," *IEEE International Conference on Bioinformatics and Bioengineering*, pp. 1027–1034, 2007.

[15] H. Willenbrock and J. Fridlyand, "A comparison study: applying segmentation to array CGH data for downstream analyses," *Bioinformatics*, vol. 21, no. 22, pp. 4084–4091, 2005.

[16] S. University, "Assembly of microarrays for genome-wide measurement of DNA copy number," *http://www.nature.com/ng/journal/v29/n3/suppinfo/ng754_S1.html*.

[17] C. Genomics, "Supplement material of lai w, 2005." [Online]. Available: http://compbio.med.harvard.edu/Supplements/Bioinformatics05b.html

[18] N. C. for Biotechnology Information, "Gene expression omnibus-GEO." [Online]. Available: http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE9220

[19] N. G. Kingsbury, "Image processing with complex wavelets," *Philosophical Transactions of the Royal Society A*, vol. 357, no. 1760, pp. 2543–2560, Sept 1999.

[20] ——, "Complex wavelets for shift invariant analysis and filtering of signals," *Journal of Applied and Computational Harmonic Analysis*, vol. 10, no. 3, pp. 234–253, May 2001.

[21] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury, "The dual-tree complex wavelet transform," *IEEE Signal Processing Magazine*, vol. 22, no. 6, pp. 123–151, Nov 2005.

[22] Y. Li and J. Zhu, "Analysis of array CGH data for cancer studies using fused quantile regression," *Bioinformatics*, vol. 23, pp. 2470–2476, 2007.

[23] J. Huang *et al.*, "Robust smooth segmentation approach for array CGH data analysis," *Bioinformatics*, vol. 23, pp. 2463–2469, 2007.

[24] D. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[25] D. Donoho and I. Johnstone, "Ideal spatial adaptation by wavelet shrinkage," *Biometrika*, vol. 81, pp. 425–455, 1994.

[26] I. Johnstone and B. Silverman, "Wavelet threshold estimators for data with correlated noise," *Journal of the Royal Statistical Society*, no. 59, pp. 319–351, 1997.

[27] S. Chang, B. Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on image processing*, vol. 9, pp. 1532–1546, Sept.2000.

[28] G. Strang and T. Nguyen, *Wavelets and filter banks*. Wellesley-Cambridge Press, 1996.

[29] R. Coifman and M. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Transactions on Information Theory*, vol. 38, pp. 713–718, 1992.

[30] A. Bruce and H. Gao, "Understanding waveshrink: Variance and bias estimation," *Biometrika*, vol. 83, pp. 727–745, 1996.

[31] N. Nguyen, H. Huang, S. Oraintara, and Y. Wang, "Denoising of array-based DNA copy number data using the dual-tree complex wavelet transform," *IEEE International Conference on Bioinformatics and Bioengineering*, pp. 137–144, 2007.

[32] A. Olshen, E. Venkatraman, R. Lucito, and M. Wigler, "Circular binary segmentation for the analysis of array-based DNA copy number data," *Biostatistics*, vol. 5, pp. 557–572, 2004.

[33] J. Hu *et al.*, "Exploiting noise in array CGH data to improve detection of DNA copy number change," *Nucleic Acids Research*, vol. 35, pp. e35–e35, 2007.

[34] R. Pique-Regi, A. Ortega, and S. Asgharzadeh, "Joint estimation of copy number variation and reference intensities on multiple DNA arrays using GADA," *Bioinformatics*, vol. 25, pp. 1223–1230, 2009.

[35] R. Pique-Regi, J. Monso-Varona, A. Ortega, R. Seeger, T. Triche, and S. Asgharzadeh, "Sparse representation and bayesian detection of the genome copy number alterations from microarray data," *Bioinformatics*, vol. 24, pp. 309–318, 2008.

[36] A. Lee *et al.*, "Analysis of copy number variation in the rhesus macaque genome identifies candidate loci for evolutionary and human disease studies." *Human Molecular Genetics*, vol. 17, no. 8, pp. 1127–36, 2008.

[37] A. Snijders *et al.*, "Assembly of microarrays for genome-wide measurement of DNA copy number," *Nature Genetics*, vol. 29, pp. 263 – 264, 2001.

[38] M. Bredel *et al.*, "High-resolution genome-wide mapping of genetic alterations in human glial brain tumors," *Cancer Research*, vol. 65, pp. 4088–4096, 2005.

[39] A. Smith *et al.*, "Array CGH analysis of copy number variation identifies 1284 new genes variant in healthy white males: implications for association studies of complex diseases." *Human Molecular Genetics*, vol. 16, pp. 2783–2794, 2007.

[40] T. Nicholas, Z. Cheng, M. Ventura, K. Mealey, E. Eichler, and J. Akey, "The genomic architecture of segmental duplications and associated copy number variants in dogs," *Genome Research*, vol. 19, pp. 491–499, 2009.

[41] M. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance," *IEEE Transactions on Image Processing*, vol. 11, pp. 146–158, 2002.

[42] H. Huang, N. Nguyen, S. Oraintara, and A. Vo, "Array CGH data modeling and smoothing in stationary wavelet packet transform domain," *BMC Genomics*, vol. 9, p. S2:S17, 2008.

[43] N. Nguyen, H. Huang, S. Oraintara, and A. Vo, "Stationary wavelet packet transform and dependent Laplacian bivariate shrinkage estimator for array-CGH data smoothing," *Journal of Computational Biology*, vol. 17, pp. 139–152, 2010.

[44] S. Mallat, *Wavelet Tour of Signal Processing - The Sparse Way.* Elsevier, 2009.

[45] E. Ben and Y. Eldar, "A fast and flexible method for the segmentation of aCGH data." *Bioinformatics*, vol. 24, pp. 139–145, 2008.

[46] N. Jeffries, "Algorithms for alignment of mass spectrometry proteomic data," *Bioinformatics*, vol. 21, pp. 3066–3073, 2005.

[47] J. Li *et al.*, "Independent validation of candidate breast cancer serum biomarkers identified by mass spectrometry," *Clinical Chemistry*, vol. 51, pp. 2229–2235, 2005.

[48] T. Rejtar *et al.*, "Increased identification of peptides by enhanced data processing of high-resolution MALDI TOF/TOF mass spectra prior to database searching," *Analytical Chemistry*, vol. 76, pp. 6017–6028, 2004.

[49] J. Morris, K. Coombes, J. Koomen, K. Baggerly, and R. Kobayashi, "Feature extraction and quantification for mass spectrometry in biomedical applications using the mean spectrum," *Bioinformatics*, vol. 21, no. 9, pp. 1764–1775, 2005.

[50] K. Coombes *et al.*, "Improved peak detection and quantification of mass spectrometry data acquired from surface-enhanced laser desorption and ionization by denoising spectra with the undecimated discrete wavelet transform," *Proteomics*, vol. 5, no. 16, pp. 4107–4117, 2005.

[51] P. Du, W. Kibble, and S. Lin, "Improved peak detection in mass spectrum by incorporating continuous wavelet transform-based pattern matching," *Bioinformatics*, vol. 22, no. 17, pp. 2059–2065, 2006.

[52] E. Lange *et al.*, "High-accuracy peak picking of proteomics data using wavelet techniques," *Proceedings of Pacific Symposium on Biocomputing*, pp. 243–254, 2006.

[53] J. Kamarainen, V. Kyrki, and H. Kalviainen, "Invariance properties of Gabor filter-based features-overview and applications," *IEEE Transactions on Image Processing*, vol. 15, no. 5, pp. 1088–1099, May 2006.

[54] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America A*, vol. 2, pp. 1160–1169, 1985.

[55] C. L. D. Tsai, "Fast defect detection in textured surfaces using 1D Gabor filters," *The International Journal of Advanced Manufacturing*, vol. 20, no. 9, pp. 664–675, Oct 2002.

[56] I. Young and M. G. L. Vliet, "Recursive Gabor filtering," *IEEE Transactions on Signal Processing*, vol. 50, no. 11, pp. 2798–2805, Nov 2002.

[57] CDC-Chronic-Fatigue-Syndrome-Research-Group, "Camda 2006 conference contest datasets." [Online]. Available: http://camda.duke.edu/camda06/datasets/index.html

[58] L. Andrade and L. Manolakos, "Signal background estimation and baseline correction algorithms for accurate DNA sequencing," *Journal of VLSI, special issue on Bioinformatics*, vol. 35, pp. 229–243, 2003.

[59] UT-MD-Anderson-Cancer-Center, "The new model processor for mass spectrometry data." [Online]. Available: http://bioinformatics.mdanderson.org/cromwell.html

[60] P. Du, "Mass spectrum processing by wavelet-based algorithms." [Online]. Available: http://bioconductor.org/packages/2.5/bioc/html/MassSpecWavelet.html

[61] P. Du *et al.*, "Application of wavelet transform to the MS-based proteomics data preprocessing," *IEEE International Conference on Bioinformatics and Bioengineering*, pp. 680–686, 2007.

[62] I. W. Selesnick, "Hilbert transform pairs of wavelet bases," *IEEE Signal Processing Letters*, vol. 8, no. 6, pp. 170–173, June 2001.

[63] N.Nguyen, H.Huang, S.Oraintara, and A.Vo, "Peak detection in mass spectrometry by Gabor filters and envelope analysis," *Journal of Bioinformatics and Computational Biology*, vol. 7, pp. 547–569, 2009.

[64] A. Yuille and T. Poggio, "Scaling theorems for zero crossings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, pp. 46–54, 1990.

[65] A. Vo, Y. Ji, and T. Hung, "Scaling theorems for zero crossings of bandlimited signals," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, pp. 309–320, 1996.

[66] P. Zhang, H. Li, S. Wang, and X. Zhou, "Peak tree: A new tool for multiscale hierarchical representation and peak detection of mass spectrometry data," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2009, accepted.

[67] X. Li, R. Gentleman, X. Lu, Q. Shi, J. Iglehart, L. Harris, and A. Miron, *SELDI-TOF mass spectrometry protein data*. Springer New York, 2005.

[68] J. W. Wong, G. Cagney, and H. M. Cartwright, "SpecAlign processing and alignment of mass spectra datasets," *Bioinformatics*, vol. 21, pp. 2088–2090, 2005.

[69] UT-MD-Anderson-Cancer-Center, "Simulated proteomics spectra." [Online]. Available: http://bioinformatics.mdanderson.org/Supplements/Datasets/Simulations/

BIOGRAPHICAL STATEMENT

Nha Nguyen received his B.S and M.S degrees in Electrical Engineering from HCMC University of Technology, Viet Nam, in 1996 and 2000, respectively. He worked at Sai Gon Technology University, Viet Nam, as a lecturer in the Department of Electrical Engineering from 2001 to 2007. He is currently a Ph.D student at the University of Texas at Arlington, USA.