Bioinformatics framework for genotyping microarray data analysis

Kai Zhang
New Jersey Institute of Technology

Follow this and additional works at: https://digitalcommons.njit.edu/dissertations

Part of the Computer Sciences Commons

Recommended Citation
https://digitalcommons.njit.edu/dissertations/765

This Dissertation is brought to you for free and open access by the Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Dissertations by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.
Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement.

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation.

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen.
The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.
ABSTRACT

BIOINFORMATICS FRAMEWORK FOR GENOTYPING MICROARRAY DATA ANALYSIS

by
Kai Zhang

Functional genomics is a flourishing science enabled by recent technological breakthroughs in high-throughput instrumentation and microarray data analysis. Genotyping microarrays establish the genotypes of DNA sequences containing single nucleotide polymorphisms (SNPs), and can help biologists probe the functions of different genes and/or construct complex gene interaction networks. The enormous amount of data from these experiments makes it infeasible to perform manual processing to obtain accurate and reliable results in daily routines. Advanced algorithms as well as an integrated software toolkit are needed to help perform reliable and fast data analysis.

The author developed a MATLAB™ based software package, called TIMDA (a Toolkit for Integrated Genotyping Microarray Data Analysis), for fully automatic, accurate and reliable genotyping microarray data analysis. The author also developed new algorithms for image processing and genotype-calling. The modular design of TIMDA allows satisfactory extensibility and maintainability. TIMDA is open source (URL: http://timda.SF.net), and can be easily customized by users to meet their particular needs. The quality and reproducibility of results in image processing and genotype-calling and the ease of customization indicate that TIMDA is a useful package for genomics research.
BIOINFORMATICS FRAMEWORK FOR GENOTYPING MICROARRAY DATA ANALYSIS

by
Kai Zhang

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Computer Science

Department of Computer Science

January 2006
APPROVAL PAGE

BIOINFORMATICS FRAMEWORK FOR GENOTYPING MICROARRAY DATA ANALYSIS

Kai Zhang

Dr. Frank Y. Shih, Dissertation Advisor
Professor of Computer Science, NJIT

Dr. Marc Q. Ma, Co-Dissertation Advisor
Assistant Professor of Computer Science, NJIT

Dr. Barry Cohen, Committee Member
Assistant Professor of Computer Science, NJIT

Dr. Usman W. Roshan, Committee Member
Assistant Professor of Computer Science, NJIT

Dr. Hui-Yun Wang, Committee Member
Research Associate of Bionomics Research & Technology EOHSI, Rutgers
BIOGRAPHICAL SKETCH

Author: Kai Zhang

Degree: Doctor of Philosophy

Date: January 2006

Undergraduate and Graduate Education:

- Doctor of Philosophy in Computer Science, New Jersey Institute of Technology, Newark, NJ, 2005

- Master of Science in Computer Science, New Jersey Institute of Technology, Newark, NJ, 2000

- Bachelor of Science in Physics, Tsinghua University, Beijing, P. R. China, 1997

Major: Computer Science

Publications:

F. Y. Shih and Kai Zhang,

F. Y. Shih and Kai Zhang,
"Inter-frame interpolation by snake model and greedy algorithms,"

F. Y. Shih, Kai Zhang and Y. Fu,
"A hybrid two-phase algorithm for face recognition,"

F. Y. Shih, Y. Fu and Kai Zhang,
"Multi-View face identification and pose estimation using B-spline interpolation,"
F. Y. Shih and Kai Zhang,
“Support vector machine networks for multi-class classification,”
International Journal of Pattern Recognition and Artificial Intelligence, Vol. 19,
no. 6, pp. 775-786, 2005.

Kai Zhang, M. Ma, H. Wang, Y. Wang, T. Wang, F. Shih and P. Soteropoulos,
“Microarray image segmentation and quantification using extended local
background,”
In Proceedings of the 8th Joint Conference on Information Sciences, pp. 587-590,
Salt Lake City, July 21-26, 2005.

Kai Zhang, M. Ma, H. Wang, Y. Wang, M. Banerjee, A. Karmaker, F. Shih, J. Wang and
H. Li,
“SNP auto-calling using support vector machines,”
In Proceedings of the 8th Joint Conference on Information Sciences, pp. 1199-
1202, Salt Lake City, July 21-26, 2005.

Kai Zhang, Marc Q. Ma, Hui-Yun Wang, Malini Banerjee, Avik Karmaker, Damien
Spivak, Yu Wang, I-Jen Yeh, Li Jia and Frank Shih,
“TIMDA, A Toolkit for Integrated Analysis of Data from Genotyping
Microarrays,”
The long river always flows to the east. The flow of the river always goes forward. Glorious heroes will always appear.

------ Shi Su
ACKNOWLEDGMENT

The author would like to express his deepest appreciation to Dr. Frank Shih and Dr. Marc Ma, who not only served as his research supervisors, providing valuable and countless resources, insight, and intuition, but also constantly gave me support, encouragement, and reassurance. Special thanks are given to Dr. Hui-yun Wang, Dr. Jason Wang, and Dr. Barry Cohen and Dr. Usman W. Roshan for once or actively participating in his proposal or dissertation committee.

Many of his fellow graduate students, Camel Fu, Shouxian Chen, Yi-Ta Wu, Chaofa Zhuang, and Ming Qu in the Computer Vision Laboratory, Yu Wang, Tongsheng Wang, Malini Banerjee, Avik Karmaker, Damien Spivak, I-Jen Yeh and Li Jia are deserved of recognition for their invaluable support.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>1.1 Genomics and Bioinformatics</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Microarray</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Objective</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Others</td>
<td>5</td>
</tr>
<tr>
<td>2 MICROARRAY</td>
<td>6</td>
</tr>
<tr>
<td>2.1 Microarray Experiments</td>
<td>7</td>
</tr>
<tr>
<td>2.2 Procedures of Microarray Experiments</td>
<td>8</td>
</tr>
<tr>
<td>3 MICROARRAY IMAGE PROCESSING</td>
<td>11</td>
</tr>
<tr>
<td>3.1 Microarray Image Processing</td>
<td>11</td>
</tr>
<tr>
<td>3.1.1 Gridding</td>
<td>13</td>
</tr>
<tr>
<td>3.1.2 Segmentation</td>
<td>14</td>
</tr>
<tr>
<td>3.1.3 Quantification</td>
<td>14</td>
</tr>
<tr>
<td>3.2 Gridding</td>
<td>15</td>
</tr>
<tr>
<td>3.2.1 Automatic Threshold Selection</td>
<td>16</td>
</tr>
<tr>
<td>3.2.2 Rotation Detection and Correction</td>
<td>18</td>
</tr>
<tr>
<td>3.2.3 Fine-Tuning</td>
<td>20</td>
</tr>
<tr>
<td>3.3 Segmentation</td>
<td>21</td>
</tr>
<tr>
<td>3.3.1 ELB</td>
<td>22</td>
</tr>
<tr>
<td>3.3.2 ELB on Simulated Noise Data</td>
<td>26</td>
</tr>
<tr>
<td>3.3.3 More Results on ELB Model</td>
<td>29</td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

(Continued)

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.4 Summary and Future Work</td>
<td>32</td>
</tr>
<tr>
<td>4 GENOTYPE DETERMINATION</td>
<td>34</td>
</tr>
<tr>
<td>4.1 Procedure of Genotype Determination Using Channel Intensities</td>
<td>34</td>
</tr>
<tr>
<td>4.2 A Special Background Subtraction Method</td>
<td>35</td>
</tr>
<tr>
<td>4.3 Normalization</td>
<td>36</td>
</tr>
<tr>
<td>4.4 Pattern Recognition</td>
<td>38</td>
</tr>
<tr>
<td>4.4.1 Support Vector Machine (SVM)</td>
<td>40</td>
</tr>
<tr>
<td>4.4.2 Neural Networks</td>
<td>47</td>
</tr>
<tr>
<td>4.5 Cutoff Method</td>
<td>48</td>
</tr>
<tr>
<td>4.6 Iterative Learning (GenoIterSVM and GenoIterANN)</td>
<td>49</td>
</tr>
<tr>
<td>5 A TOOLKIT FOR INTEGRATED GENOTYPING MICROARRAY DATA ANALYSIS, TIMDA</td>
<td>52</td>
</tr>
<tr>
<td>5.1 Modular Design</td>
<td>52</td>
</tr>
<tr>
<td>5.2 Image Processing (IP) Module</td>
<td>54</td>
</tr>
<tr>
<td>5.3 Intermediate Data Conversion (IDC) Module</td>
<td>59</td>
</tr>
<tr>
<td>5.4 Genotype Calling (GC) module</td>
<td>61</td>
</tr>
<tr>
<td>5.5 Loss-of-Heterozygosity (LOH) Module</td>
<td>66</td>
</tr>
<tr>
<td>5.6 OneInt Module</td>
<td>70</td>
</tr>
<tr>
<td>5.7 Implementation</td>
<td>71</td>
</tr>
<tr>
<td>5.7.1 Rotation Detection and Correction</td>
<td>71</td>
</tr>
<tr>
<td>5.7.2 Customized Format Loading</td>
<td>72</td>
</tr>
<tr>
<td>5.7.3 ID Loading and Association</td>
<td>73</td>
</tr>
<tr>
<td>Chapter</td>
<td>Page</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>6 SUMMARY AND FUTURE RESEARCH</td>
<td>74</td>
</tr>
<tr>
<td>APPENDIX A EFFICIENT CONTOUR DETECTION BASED ON IMPROVED SNAKE MODEL</td>
<td>75</td>
</tr>
<tr>
<td>APPENDIX B INTER-FRAME INTERPOLATION BY SNAKE MODEL AND GREEDY ALGORITHMS</td>
<td>77</td>
</tr>
<tr>
<td>APPENDIX C A HYBRID TWO-PHASE ALGORITHM FOR FACE RECOGNITION</td>
<td>81</td>
</tr>
<tr>
<td>APPENDIX D SUPPORT VECTOR MACHINE NETWORKS FOR MULTI-CLASS CLASSIFICATION</td>
<td>84</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>89</td>
</tr>
</tbody>
</table>
**LIST OF TABLES**

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.1</td>
<td>ELB Results for that Spot with Very High Background Value</td>
<td>30</td>
</tr>
<tr>
<td>3.2</td>
<td>Processing Time Used for Different ELB Configurations</td>
<td>31</td>
</tr>
<tr>
<td>5.1</td>
<td>Example Format of an ID Association File</td>
<td>60</td>
</tr>
<tr>
<td>5.2</td>
<td>Genotyping Methods</td>
<td>63</td>
</tr>
<tr>
<td>5.3</td>
<td>Genotyping Results with Different Normalization and Genotype Calling Methods</td>
<td>63</td>
</tr>
<tr>
<td>5.4</td>
<td>Genotyping Results for Manipulated Data</td>
<td>64</td>
</tr>
<tr>
<td>5.5</td>
<td>Example Format of a LOH Report</td>
<td>69</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>A microarray printer</td>
<td>7</td>
</tr>
<tr>
<td>2.2</td>
<td>A small glass slide of a microarray experiment</td>
<td>8</td>
</tr>
<tr>
<td>2.3</td>
<td>Hybridization</td>
<td>9</td>
</tr>
<tr>
<td>3.1</td>
<td>A sample microarray image</td>
<td>11</td>
</tr>
<tr>
<td>3.2</td>
<td>A sample image taken from microarray for both green and red channels</td>
<td>12</td>
</tr>
<tr>
<td>3.3</td>
<td>The basic procedure of gridding</td>
<td>16</td>
</tr>
<tr>
<td>3.4</td>
<td>Automatic threshold determination</td>
<td>16</td>
</tr>
<tr>
<td>3.5</td>
<td>The layout template is determined based on the parameters</td>
<td>18</td>
</tr>
<tr>
<td>3.6</td>
<td>Global rotation detection</td>
<td>19</td>
</tr>
<tr>
<td>3.7</td>
<td>a) The original image; (b) the used filter; (b) the filtered image</td>
<td>20</td>
</tr>
<tr>
<td>3.8</td>
<td>The final fine-tuned gridding</td>
<td>21</td>
</tr>
<tr>
<td>3.9</td>
<td>Extended local background</td>
<td>24</td>
</tr>
<tr>
<td>3.10</td>
<td>Different allowable ELB configurations</td>
<td>25</td>
</tr>
<tr>
<td>3.11</td>
<td>Background region definition used in other analysis tools</td>
<td>26</td>
</tr>
<tr>
<td>3.12</td>
<td>(a) Gaussian noise image (500 by 500 pixels) for ideal experiment; (b)</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td>Intensity level of a sub-image from (a)</td>
<td></td>
</tr>
<tr>
<td>3.13</td>
<td>(a) Distributions of mean values for varying size of local regions used</td>
<td>28</td>
</tr>
<tr>
<td></td>
<td>for calculating the means. (b) the histogram and fitted distribution for a</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 by 10 local region</td>
<td></td>
</tr>
<tr>
<td>3.14</td>
<td>The standard deviation of pixel intensities</td>
<td>29</td>
</tr>
<tr>
<td>3.15</td>
<td>The close look of (a) the spot with high background value and (b) one of</td>
<td>30</td>
</tr>
<tr>
<td></td>
<td>adjacent pots</td>
<td></td>
</tr>
<tr>
<td>3.16</td>
<td>background distribution comparison between ELB and GenePix</td>
<td>31</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
<td>Page</td>
</tr>
<tr>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>3.17</td>
<td>The background values calculated using different ELB configurations</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>MA-plot</td>
<td>38</td>
</tr>
<tr>
<td>4.2</td>
<td>Intensity projections from red and green channels</td>
<td>39</td>
</tr>
<tr>
<td>4.3</td>
<td>Logarithm intensity projection from red and green channels</td>
<td>39</td>
</tr>
<tr>
<td>4.4</td>
<td>Basic SVM</td>
<td>40</td>
</tr>
<tr>
<td>4.5</td>
<td>One-against-all SVM</td>
<td>44</td>
</tr>
<tr>
<td>4.6</td>
<td>One-against-one SVM</td>
<td>45</td>
</tr>
<tr>
<td>4.7</td>
<td>Architecture of a layered neural network</td>
<td>47</td>
</tr>
<tr>
<td>4.8</td>
<td>The linear cutoff based method in a two-color genotyping microarray</td>
<td>48</td>
</tr>
<tr>
<td>4.9</td>
<td>SVM classifiers using (a) the linear kernel and (b) those with iterative adjustment</td>
<td>50</td>
</tr>
<tr>
<td>4.10</td>
<td>The procedure of iterative learning of GenoIterSVM / GenoIterANN</td>
<td>50</td>
</tr>
<tr>
<td>5.1</td>
<td>(a) TIMDA framework; (b) OneInt module</td>
<td>54</td>
</tr>
<tr>
<td>5.2</td>
<td>The gridding sub-module in the IP module</td>
<td>55</td>
</tr>
<tr>
<td>5.3</td>
<td>The interface of ELB sub-module</td>
<td>56</td>
</tr>
<tr>
<td>5.4</td>
<td>The interface of the ELB pixel-wised definition</td>
<td>57</td>
</tr>
<tr>
<td>5.5</td>
<td>The distribution plot notifies the ELB process is done</td>
<td>58</td>
</tr>
<tr>
<td>5.6</td>
<td>IDC interface</td>
<td>60</td>
</tr>
<tr>
<td>5.7</td>
<td>The logarithm distribution plot notifies the IDC process is done</td>
<td>61</td>
</tr>
<tr>
<td>5.8</td>
<td>GC interface</td>
<td>65</td>
</tr>
<tr>
<td>5.9</td>
<td>Genotype layout</td>
<td>65</td>
</tr>
<tr>
<td>5.10</td>
<td>LOH logarithm comparison plot</td>
<td>67</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS
(Continued)

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.11</td>
<td>The graphical output for index-based analysis of LOH module</td>
<td>68</td>
</tr>
<tr>
<td>5.12</td>
<td>The interface of OneInt module</td>
<td>70</td>
</tr>
<tr>
<td>5.13</td>
<td>(a) An original image; (b) the rotated image with 5°</td>
<td>71</td>
</tr>
<tr>
<td>5.14</td>
<td>The sub-array numbering</td>
<td>73</td>
</tr>
<tr>
<td>B.1</td>
<td>Inter-frames by using linear interpolation</td>
<td>78</td>
</tr>
<tr>
<td>D.1</td>
<td>Decision function of a 3-class SVM</td>
<td>85</td>
</tr>
<tr>
<td>D.2</td>
<td>The architecture of our framework</td>
<td>85</td>
</tr>
<tr>
<td>D.3</td>
<td>Hyper-planes divide the space</td>
<td>86</td>
</tr>
<tr>
<td>D.4</td>
<td>The overall procedures</td>
<td>87</td>
</tr>
</tbody>
</table>
CHAPTER 1

INTRODUCTION

Genomics study is very important in understanding the genetic basis for many biological phenomena. Using the state-of-the-art of microarray technology, biological and biomedical researchers are able to simultaneously monitor the expression level or genotypes of thousands of genes, which can be used to comprehend gene functions and/or construct complex gene interaction networks. The amount of data generated from this type of microarray experiment is enormous, so that it is not feasible to use manual processing for accurate and reliable analysis in daily routine. According to the requirement of microarray data analysis, the author developed TIMDA (Toolkit for Integrated Microarray Data Analysis) for spotted SNP genotyping microarray data analysis with some novel algorithms.

1.1 Genomics and Bioinformatics

Genomics is the branch of biology that studies the structure and function of genes [1]. For all living creatures, their organisms consist of small cells. There are estimated about $6 \times 10^{13}$ cells in a human body, of about 320 different types. In a typical cell, there are one or several long double stranded DNA molecules organized as chromosomes [2]. A human has 23 pairs of chromosomes, which are believed to encode almost all the functional heredity information. Determining the four letter (A, T, G and C) sequence for a given a DNA molecule is known as the DNA sequencing. A gene [3] is the structural unit of
inheritance in living organisms. A gene is, in essence, a segment of DNA that has a particular purpose, i.e., that codes for (contains the chemical information necessary for the creation of) a specific enzyme or other protein. The strands of DNA on which the genes occur are organized into chromosomes. The nucleus of each eukaryotic (nucleated) cell has a complete set of chromosomes and therefore a complete set of genes. Each gene provides a blueprint for the synthesis (via RNA) of enzymes and other proteins and specifies when these substances are to be made (see nucleic acid). Genes govern both the structure and metabolic functions of the cells, and thus of the entire organism and, when located in reproductive cells, they pass their information to the next generation. It is also a determined factor to make a particular type of a protein or a few different proteins. DNA is the carrier of genes. It is considered that there must be at least 40,000 – 50,000 genes in the human genome, in which there are about 6700 experimentally confirmed. Particularly important variations in individual genomes are the single nucleotide polymorphisms (SNPs). SNPs are DNA sequence variations which occur when a single base (A, T, G and C) is altered so that different individuals may have different letters in these positions. Particular nucleotides in SNP positions within genes can influence the gene's protein product. SNP variation may indicate the predisposition to a genetic disease. An important question in biology is how genes are regulated. Microarrays and computational methods are playing a major role in attempts to reveal the gene networks.

*Bioinformatics* [4] derives knowledge from computer analysis of biological data. These can consist of the information stored in the genetic code, but also experimental results from various sources, patient statistics, and scientific literature. Research in bioinformatics includes method development for storage, retrieval, and analysis of the data.
Bioinformatics is a rapidly developing branch of biology and is highly interdisciplinary, using techniques and concepts from informatics, statistics, mathematics, chemistry, biochemistry, physics, and linguistics. It has many practical applications in different areas of biology and medicine [5].

1.2 Microarray

Functional genomics study is a complex enterprise and it is made tractable by the technological breakthroughs in microarrays (Schena et al. [6], DeRisi et al. [7] and Hegde et al. [8]). By using recent technological breakthroughs in high-throughput instrumentation and microarray data analysis [1, 6, 7, 9, 10], microarray experiments establish gene expression and/or genotypes of the SNP-containing DNA sequences.

Gene expression microarray is used to measure gene expression on a large scale basis. It tries to find out how the expression of a gene is, or what patterns of gene expression cause a specific disease. Genotyping microarray establish the genotypes of DNA sequences containing single nucleotide polymorphisms (SNPs), and can help biologists probe the functions of different genes and/or construct complex gene interaction networks.

1.3 Objective

The purpose of microarray experiments is to accurately and fast measure the gene expression level and determine the correct genotypes. The amount of data generated from these types of microarray experiments is enormous, so that it is not feasible to use manual processing for accurate and reliable analysis in daily routine. Without reliable, robust and
fast data analysis which yields critical subsequent results on gene expression, statistics and/or genotypes, it is impossible to infer genetic changes due to drug action, genetic differences during development, gene interaction networks or biological pathways. Enormous volume of raw data is generated from microarray experiments, which precludes manual processing and strongly calls for automated data analysis. New advanced algorithms for image processing and genotype-calling are developed to improve the accuracy of microarray data analysis and genotype determination.

As technology advances, the density of spots on microarray chips continues to increase. With more pixels used in the local background, it can make significant statistical sense and result in more reliable estimates, so the author proposed a new robust segmentation method based on a larger region to estimate the local background to achieve better results. New methods applying machine learning approaches with iterative learning for genotype determination are also proposed, which can achieve better results than traditional simple cutoff value based methods.

According to the requirement of microarray data analysis, the author developed a MATLAB-based software framework, TIMDA (Toolkit for Integrated Microarray Data Analysis), for spotted SNP genotyping microarray data analysis. The framework consists of modules of, image processing, intermediate data conversion, genotype-calling and Loss-of-Heterozygosity (LOH) with graphical user interface (GUI). The image processing module employs efficient and automatic gridding techniques as well as a novel extended local background (ELB) noise model for better foreground/background segmentation. The genotype-calling module consists of Support Vector Machine (SVM) and Artificial Neural Network (ANN) based methods with iterative learning. The LOH module can be used to
study the genetic changes in diseases such as cancers. The modular design of the framework achieves satisfactory extensibility and maintainability. TIMDA can be publicly accessed from internet (URL: http://timda.SF.net).

1.4 Others
The author's other research will be listed in the Appendix. Appendix A: Efficient contour detection based on improved snake model [11]; Appendix B: Inter-frame interpolation by snake model and greedy algorithms [12]; Appendix C: A hybrid two-phase algorithm for face recognition [13]; Appendix D: Support vector machine networks for multi-class classification [14].
CHAPTER 2

MICROARRAY

Functional genomics is a flourishing field of study in which the function and behavior of genes in a genome is investigated. It tends to function collectively in pathways as coordinated sequences of genetic and molecular activities. Functional genomics study is a complex enterprise and it is made tractable by the technological breakthroughs in microarrays. Using the state of the art of high-throughput instrumentation microarray technology and microarray data analysis, the expression level of thousands of genes can be monitored simultaneously. The oligonucleotide genotyping microarray probes the sequence variation in the genome and has drawn increasing attention in the biomedical community. It has been used in analyzing the loss of heterozygosity (LOH) in human bladder [15], breast cancer [16] and lung cancer [17].

There are mainly two types of microarrays, which are gene expression microarray and genotyping microarray. DNA microarray is widely used as general means of monitoring the expression patterns of large number of genes. Genotyping microarray amplifies thousands of DNA sequences containing SNPs in a single multiplex *Polymerase Chain Reaction (PCR)*, which Polymerase chain reaction (PCR) has rapidly become one of the most widely used techniques in molecular biology and for good reasons: it is a rapid, inexpensive and simple means of producing relatively large numbers of copies of DNA molecules from minute quantities of source DNA material—even when the source DNA is
of relatively poor quality. PCR involves preparation of the sample, the master mix and the primers, followed by detection and analysis of the reaction products. And then the genotypes of the SNPs can then be determined by microarray assay. These SNPs can be used as markers to detect genetic changes in different genomes including those from patients with diseases such as cancer to understand the mechanisms underlying various biological and disease processes.

This chapter will briefly introduce the microarray experiment and microarray data analysis.

2.1 Microarray Experiments

A microarray is a small device, about the size of a microscope slide, with thousands of different known DNA sequences immobilized at different addresses on the surface. Each of these DNA sequences can participate in a hybridization reaction. Because there are several thousands of spots in a single microarray, it allows the monitoring of expression levels for thousands of genes simultaneously. Figure 2.1 [18] shows the microarray printer used in microarray experiments.
Microarray experiments have been applied to a wide variety of fields in bioinformatics. Microarrays can be conducted in different types of research, such as, tissue-specific gene expression, genetic disease and complex diseases etc.

2.2 Procedures of Microarray Experiments

The five basic steps are conducted in a typical microarray experiment.

1. Preparing purified DNA or oligonucleotide samples;
2. Preparing the labeled samples;
3. Hybridization;
4. Scanning the microarray image;
5. Analyzing the scanned image.
In brief, a typical microarray experiment proceeds as follows: take a small glass slide as shown in Figure 2.2 [18]. Suppose that the surface of the slide has been divided into serials of imaginary squares cells to form a rectangular grid. Onto each square cell, stick a tiny amount of liquid that contains purified DNA corresponding to a gene of known sequence. Different cells may express different genes. The separately prepared solution contains a mixture of mRNA whose sequences are unknown. Add to this solution a substance that fluoresces when excited by light. Pour the solution onto the slide. The mRNA molecules will diffuse over the slide and, wherever they find a matching (i.e., complementary) DNA sequence, such as the one taken from the gene from which the
mRNA was transcribed, they will hybridize to each other and the solution will stick to the slide as shown in Figure 2.3. Without a match, the solution will not stick to the slide and can be washed away. Use a laser scanner to detect and measure the fluorescent signal being emitted at each cell. By comparing the intensity levels of the fluorescent signals (The experiment uses the fluorescent Cy3-ddNTP (red) and Cye5-ddNTP (green) to label the probes.) across the multiple mRNA samples, scientists will be able to understand how the expression profile of a set of genes differs across the different mRNA samples.

After obtained the scanned image, how to accurately measure the true signal intensity levels is the core problem for microarray image processing. For the genotyping microarray, there are two types of approaches to determine genotypes. One is using measured the two-channel intensity level; the other is biological approach including the Restriction Fragment Length Polymorphism (RFLP) [19] and direct sequencing [20, 21].
Images from microarray experiments contain digitized images scanned from two channels, red and green. Image processing has been a bottleneck for reproducibility, accuracy and efficiency of microarray data analysis due to the intrinsic difficulties such as bad grid layouts, contaminations, background estimations, noises, irregular spot shapes, dense layouts and etc. [22].

3.1 Microarray Image Analysis

Figure 3.1 A sample microarray image. The experiment uses the fluorescent Cy3-ddNTP (red) and Cye5-ddNTP (green) to label the allelic probes.
Figure 3.1 shows the raw data of a typical two-color SNP genotyping microarray using the new high-throughput genotyping system [10]. After the microarray images are scanned at the end of the experiment, the images must be converted into spot intensities for analysis. The purpose of this process is to assign every DNA sequence that was spotted on the microarray an intensity measure, called the spot intensity, reflecting the amount of labeled sample that hybridized to it. The main goal of array image processing is to measure the intensity of the spots and quantify the gene expression values based on these intensities [23].

In some of experiments, the data for each gene is expressed by two fluorescence intensity measurements differentially, \((R, G)\), representing the expression level of the gene in the red (Cy5) and green (Cy3) respectively. Sample images taken from the red channel are shown in Figure 3.2. Two-channel information will be digitized for subsequent image analysis to reveal its biological meanings.

Figure 3.2 A sample image taken from microarray for both green and red channels.
The procedure of microarray data is generally divided into three steps:

1. Gridding (addressing);
2. Segmentation;
3. Reduction.

### 3.1.1 Gridding

Gridding is trying to index the spots in microarray images. Generally, every spot has a special ID which indicates its biological meaning. Researchers can use this ID to indicate its biological meaning. If the arraying process arranged the spots in a perfect rectangular grid, as it should, the task of indexing the spots by assigning coordinates to the center of each circular spot should be a simple matter: just overlay an appropriately sized template on the microarray image and move it around. In practice, however, the arraying process is not perfect; the overlaid grid will need further fine-tuning.

Jain et al. [24] describe a system for microarray gridding and quantification. Their gridding algorithm is based on axis projections of image intensity. This approach is not robust to misalignments of different grids and rotations. They make strong assumptions on the distribution of image intensity, but allow projective distortions of the printed grids. Yang et al. [25] use template matching and seeded region growing methods for semiautomatic gridding.
3.1.2 Segmentation

Because the process of the experiments, the glass slide also reflects the light, when using laser to scan the microarray, a certain intensity level will be displayed in the background region of the scanned image. To reveal the true intensity value of each spot, the background value should be subtracted from the actual intensity reading.

In estimation of microarray background, there are several methods being used. Steinfath et al. [26] describe methods for segmentation of filter array images. Adaptive shape (AS) segmentation offers more flexible answers to irregular shapes, but it cannot give robust estimation for the foreground or background when big local variation of intensities exists. Several authors propose morphological methods for grid segmentation [27, 28]. Since these approaches employ axis projections as a central component, irregular and overlapping grid layouts cause problems. BlueFuse [22, 29], which is one of the histogram-based methods, uses a Bayesian model to generate a confidence measure for each spot. In comparison to the spatial-based approaches, histogram-based methods do not spend processing time in analyzing spatial distribution for each spot. Instead, they directly analyze the histogram distribution of local spot regions, in which the pixels are categorized into foreground and background based on some criteria. However, the quantization in histogram-based methods is unstable when a large target mask is set to compensate for spot size variation [30].

3.1.3 Quantification

After the intensity level of each spot is measured, the quality of the spot can be assessed by other spot-related statistics [31-33], which include spot intensity, spot background, the
number of foreground pixels, the number of background pixels, pixel intensity distribution and spot morphology.

3.2 Gridding

The purpose of gridding is to index microarray layouts despite of possible rotations residing in scanned microarrays images. After this step, each spot in the microarray image will have a sub-array number, row number and column number associated with it. These indices will combine with certain IDs to refer to specific biological meanings in the subsequent analysis. Figure 3.1 and 3.2 show the scanned digitized microarray images. The basic procedure of gridding is illustrated in Figure 3.3. The input is the digitized image and the output will be the locations for each indexed spot in the scanned image.

The purpose of finding a suitable threshold to binary the input image is to precisely find the spot layout template according to the intensity projection of the binary image. After the rotation of the image has been adjusted, the new threshold should be reset, otherwise the layout can’t be measured correctly. Finally, the center of each indexed spot is determined for subsequent analysis.
Find a suitable threshold to binary the input image

Rotation Detection

Rotation Correction

Recalculate the threshold to binary the adjusted image

Calculate the layout template of the adjusted image

Calculate the layout template of the original image

Fine-tuning the location for each spot

Figure 3.3 The basic procedure of gridding.

3.2.1 Automatic Threshold Selection

Figure 3.4 Automatic threshold determination. Left (a): The horizontal projection with a random threshold. Middle (b): It shows such a projection with the best threshold. Right (c): The distribution of variances of vertical intensity projection based on different thresholds.
The approach for automatic threshold determination is illustrated in Figure 3.4. First, it can set a threshold selection range, for example, \([1,50]\). For a certain threshold \(\sigma\), it resets the intensities of the image as 0’s and 1’s, interpreted as black and white. The image size is \(m \times n\).

\[
I_{\text{binary}}(i, j, \sigma) = \begin{cases} 
0 & \text{if } I(i, j) \leq \sigma \\
1 & \text{if } I(i, j) > \sigma 
\end{cases}
\]

Then, it projects the intensity summation along the \(x\) and \(y\) axes as follows:

\[
f(i, \sigma) = \sum_{j=1}^{n} I_{\text{binary}}(i, j, \sigma).
\]

Figure 3.4(a) shows the horizontal projection with a random threshold \(\sigma\). From it, you cannot observe the periodical intensity distribution. The algorithm selects the variance (or standard deviation (STD)) to measure its periodical distribution. Figure 3.2(c) illustrates the relation between the threshold \(\sigma\) and its variance. The higher variance, the more periodical properties the projection distributes. The algorithm chooses the threshold corresponding to the highest variance in Figure 3.2(c), which is used in Figure 3.2(b). It should display the best periodical distribution result. According to the (horizontal and vertical) projections, layout parameters, which include the number of columns and rows, the interval between adjacent columns and rows, the interval between adjacent sub-array
and approximate spot radius, are calculated. Based on them, a binary layout template can be determined. From the experiences, usually, optimal threshold values for horizontal and vertical axes may be slightly different. The calculation of layout parameters will be based on the different thresholds along the $x$ and $y$ axes. Figure 3.5 shows the layout template determined by those parameters.

Figure 3.5 The layout template is determined based on the parameters.

### 3.2.2 Rotation Detection and Correction

The rotation may exist on microarray images. Without rotation correction, the misleading layout may be applied to subsequent analysis. For rotation detection, it still applies intensity projections and variance (or $STD$) to measure the possible rotation. The idea is
illustrated in Figure 3.4, which is also mention in [34]. If there exists rotation (Figure 3.6(c)), its horizontal and/or vertical intensity projection should be disordered (Figure 3.6(d)); otherwise, its projection should be periodically distributed (Figure 3.6(c)). It calculates the variance of the horizontal and vertical intensity projection and use it as the indicator of possible rotations. The input image is rotated within certain successive bounds (-5° to 5°) with 1° for each step, and then the variance of intensity projection for each rotation degree is calculated. Figure 3.6(e) and 3.6(f) show the variance distributions of horizontal and vertical projections based on the skewed image, Figure 3.6(e, f), in which the peak values indicate the rotation correctly. The further fine-tuning can deal with smaller rotations (<1°). That is the reason that 1° for each step is chosen. A spot layout template is constructed after the rotation correction.

![Figure 3.6](image)

**Figure 3.6** Global rotation detection. (a): A sample un-skewed image; (b): a skewed image; (c, d): The intensity projection respects x-axis; (e, f): The variances changes respect to rotation and the peaks indicate the rotation degrees of (b).
After the rotation degree is detected, the original microarray image will be adjusted according to that degree. Then, a new threshold will be recalculated for both axes. The layout parameters can be determined. The indexes and locations of all spots are obtained according to layout parameters in the adjusted image. The next step is to convert the locations into the original image.

3.2.3 Fine-Tuning

In practice, the gridding somehow may not be perfectly aligned. In such cases, the fine-tuning is needed. Based on an assumption that the correct signal region covers more intensities than the other regions locally, the center for each spot can be determined by finding local maximum intensity. For speeding up the process, it uses a filter to find local maxima, which is shown in Figure 3.7(b). The filter reassigns the pixel value to be the intensity summation inside the local filter region. From the centers decided from last step, each center will be reassigned to the location with the local maximum value of Figure 3.7(c). The final layout is displayed in Figure 3.8, in which some of spots have been locally tuned from the perfect locations.

Figure 3.7 (a) The original image; (b) the used filter; (b) the filtered image.
3.3 Segmentation

Because the process of the experiments, the glass slide also reflects the light, when using laser to scan the microarray, a certain intensity level will be displayed in the background region of the scanned image. Segmentation of microarray image is trying to distinguish the background and foreground signals. In estimation of microarray background, there are several methods being used. Global background (GB) estimation method calculates the average intensity level of all the pixels not belonging to signal regions is calculated. Thus, GB ignores the spatial background variation across the whole slide. Several other spatial or histogram-based techniques have been proposed for analyzing microarray images to overcome this limitation. Fixed circle (FC) segmentation fits circles with a constant
diameter to all the spots in the image [25]. Adaptive circle (AC) segmentation estimates the circle’s diameter individually for each spot [35] so that some of the limitations of FC segmentation method are overcome and may generate more reliable estimates. However, AC does not handle irregular shapes such as donuts properly. Adaptive shape (AS) segmentation offers more flexible answers to irregular shapes, but it cannot give robust estimation for the foreground or background when big local variation of intensities exists.

Morphological approach [27, 28, 36], seeded region growing (SRG) [37] and Markov Random Field (MRF) [38] Model also attract researchers’ attention. BlueFuse [22, 29], which is one of the histogram-based methods, uses a Bayesian model to generate a confidence measure for each spot. In comparison to the spatial-based approaches, histogram-based methods do not spend processing time in analyzing spatial distribution for each spot. Instead, they directly analyze the histogram distribution of local spot regions, in which the pixels are categorized into foreground and background based on some criteria. However, the quantization in histogram-based methods is unstable when a large target mask is set to compensate for spot size variation [30].

3.3.1 ELB

As technology advances, the density of spots on microarray chips continues to increase. With more pixels used in the local background, it can make significant statistical sense and result in more reliable estimates. The author focus on a new robust segmentation method based on a larger region to estimate the local background, which is termed as extended local background (ELB). ELB is essentially a histogram-based method. Before processing ELB for each spot, it uses the spot gridding template constructed from the last step to
estimate the global background first. All pixels belonging to the background region of the whole chip are collected, and then 20% high intensity pixels are eliminated because of irregular spot shapes, saturated and contaminated spots, which probably reside in the background region of the layout template from the last step, and then calculates the average global background level \( \bar{v} \) and global standard deviation \( \sigma_{GB} \). Then, the global cut-off threshold value \( V_{GB} \) is computed as

\[
V_{GB} = \bar{v} + \sigma_{GB},
\]

which will be considered to be larger than local background intensities levels. The reason is that those considered pixels include many pixels with very low grey level (lots of pixels with 0 intensity level) and they will enlarge the actual global standard deviation \( \sigma_{GB} \). That guarantees \( V_{GB} \) larger than local background intensities. \( V_{GB} \) will be further applied to the local background estimation.
Next, it uses local histogram information on a suitable (larger) population to generate better and more robust estimates for local background noise. Figure 3.9 illustrates an allowable ELB configuration. Pixels in the shaded area with intensities below $V_{GB}$ are initially treated as the candidate background pixels and their mean, median, standard deviation $\sigma_{BG}$ and other statistics values are calculated. It chooses the median intensity $\bar{v}_{local}$ as the background intensity associated with the individual spot. Then, a new local foreground cutoff threshold $V_{foreground\_cutoff}$ is calculated as

$$V_{foreground\_cutoff} = \bar{v}_{local} + 2\sigma_{BG},$$

which is applied to the further foreground estimation. In signal region, only pixels with the intensities above this threshold will be classified as foreground pixels from which it computes their median value, which is assumed to be the combination of true signal intensity and background intensity, $v_{(signal+noise)}$. True signal intensity is revealed using the following equation in which $V_{ELB} = \bar{v}_{local}$

$$v_{true\_signal} = v_{(signal+noise)} - V_{ELB}.$$
Because ELB already considered a larger region for local background estimation and a few more pixels will not make a big difference but increase complexity, it does not take pixels with intensity larger than $\overline{V}_{local}$ in the signal region back into consideration of $\overline{V}_{local}$.

Figure 3.10 Different allowable ELB configurations.

ELB also allows flexible configurations as shown in Figure 3.10, which includes square, circle, rectangle and ellipse. The configurations can be defined by pixel-wise or spot-wise. For a pixel-wise definition, to set the width with 100 and the height with 100, users will get the similar configuration shown in Figure 3.10(a), in which the shaded areas are the initial ELB regions; for a spot-wise definition, to set a rectangular ELB with size of 5 by 3 in terms number of spots, users will get the configuration shown in Figure 3.10(c).

Figure 3.11 Background region definition used in other analysis tools.

Figure 3.11 shows the local background region definitions [39] used in some popular microarray image processing software packages including ScanAlyze™ [40],...
QuantArray™ [41], ImaGene™ [42] and GenePix™ [43]. To estimate local background noise, ScanAlyze™ allows user to define a size of a square region, which is similar to the ELB method, however its choice is limited. QuantArray™, ImaGen™ and GenePix™ use fixed regions. One obvious disadvantage for fixed regions is that if the specific spot is defective for any reasons, the background estimates will significantly be affected. Involving more pixels for estimation, ELB guarantees to make more statistic sense. Because of the pre-distinguished global cut-off threshold value $V_{\text{global\_cutoff}}$ and local cut-off threshold value $V_{\text{foreground\_cutoff}}$ used, the ELB can generate more robust and accurate estimation. On the other hand, since it is histogram-based background estimation method, it can tolerate more errors in the gridding stage, such as false grid locations. ELB is a novel, viable and feasible method for robust microarray image segmentation and quantification.

3.3.2 ELB on Simulated Noise Data

The author performs a simple numerical experiment to establish the statistical basis of ELB and validate that larger population of pixels yield more robust statistical estimates [44]. He generates a square Gaussian random noise image with 500 pixels on each side. The probability distribution function is defined in:

$$f(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}},$$

in which it sets the mean value as $\mu = 300$ and the standard deviation as $\sigma = 200$. The random image and the noise levels are shown in Figure 3.12.
225 grid points are selected evenly spaced on the random noise image, Figure 3.10 (a) and calculate the mean value and standard deviation of the pixel intensities at each one of these locations with varying size of the local regions. The size of each local region is determined such that no overlapping of adjacent local regions will occur to help isolate factors that may influence the statistics. Figure 3.13(a) shows the distributions of mean values for varying size of local regions used for calculating the means. You can observe that as the local region becomes larger, the variance becomes smaller (the bell curve gets “thinner”). This indicates that the estimates approach the expected value (the preset mean value) more robustly. Figure 3.13(b) shows the details of histogram distribution corresponding to local regions of size of 10 by 10 pixels, from which you can observe that a large number of estimates of mean values are far off from the expected mean value, which renders these estimates less robust. Figure 3.12 shows that the larger the population used in local noise estimation, the more robust these estimates become.
Figure 3.13 (a) Distributions of mean values for varying size of local regions used for calculating the means. (b) the histogram and fitted distribution for a 10 by 10 local region.

Figure 3.14 The standard deviation of pixel intensities.

While computing the mean value for a local region, it also obtains the standard deviation of the pixel intensities. For each size of the local region to be examined, it picks 10 grid points and plot the standard deviation across these grid points. As shown in Figure 3.14, you see the standard deviation estimates becomes more stable as the size of the local
region becomes larger, which implies that the threshold value, $v_{\text{cutoff}}$, for each local region becomes more reliable.

It can use the above procedure to fine-tune the size of the local region such that the standard deviation of a random subset of grid points becomes stable and approaches the preset value. Figure 3.12(d) shows that local region size of is already reasonably good in terms of the standard deviation (4.8798) of the standard deviations computed for all 225 local regions.

### 3.3.3 More Results on ELB Model

![Figure 3.15](image)

**Figure 3.15** The close look of (a) the spot with high background value and (b) one of adjacent spots.

Figure 3.16 compares the scatter plot of the logarithmic values of foreground intensity vs. the corresponding background intensities obtained using the ELB method in the IP module (left) and GenePix (right) for the same microarray image. Intuitively, the variance of the background intensity values should be bounded in a certain range, which is proved by the random noise model. There are several unusual spots whose intensities are either too small or out of the range. Figure 3.15(a) gives a close look of the spot with a very high background value (214) in Figure 16. After checking the result from the GenePix
report, you can find the problem is that its diameter defined for that spot is relatively small (60) comparing to most of spots (such as Figure 3.15(b): 150). According to the GenePix background definition (Figure 3.11) the tentative background region of this spot will locate in the real signal region, in that dramatically increases the background estimation value. That illustrates the limitation of the spatial-based segmentation, which is that accurate quantification results highly rely on the correct segmentation and it also shows the advantage of applying the global background cutoff value. Table 3.1 illustrates the results for this spot based on different ELB configurations with more reliable results as more pixels involves in the estimation.

<table>
<thead>
<tr>
<th>ELB Results for that Spot with Very High Background Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>num. of BG pixels</td>
</tr>
<tr>
<td>-------------------</td>
</tr>
<tr>
<td>Spot-based 1X1</td>
</tr>
<tr>
<td>Spot-based 1X2</td>
</tr>
<tr>
<td>Spot-based 2X1</td>
</tr>
<tr>
<td>Spot-based 2X2</td>
</tr>
</tbody>
</table>
Figure 3.16 The background distribution comparison between ELB and GenePix.

Figure 3.17 The background values calculated using different ELB configurations.

Table 3.2 Processing Time Used for Different ELB Configurations

<table>
<thead>
<tr>
<th>Sizes of ELB</th>
<th>Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>3×3</td>
<td>7.40</td>
</tr>
<tr>
<td>5×5</td>
<td>15.27</td>
</tr>
<tr>
<td>7×7</td>
<td>27.02</td>
</tr>
</tbody>
</table>
The ELB model usually results in lower background variance, due to larger estimation regions used as expected. Figure 3.16 illustrates the background distribution comparison (one-channel) between ELB and GenePix. x-axis denotes the index of spots in 1-D and y-axis denotes corresponding background intensity levels. For GenePix there are several relatively large background values which are likely the result of misclassification of some foreground pixels; there are several very small background values which are affected by the gap of adjacent sub-arrays. ELB may avoid such errors, because it applies the global background estimation before the ELB. Figure 3.17 illustrates the background distributions using different ELB configurations (3×3, 5×5 and 7×7), which becomes smoother as the ELB definition size is larger. Users can easily define the ELB to meet their needs. Table 3.2 lists the processing time used for different ELB configurations.

### 3.4 Summary and Future Work

The author has presented ELB, a novel and viable method for robust microarray image segmentation and quantification based on using a larger local region. He has done preliminary validation of ELB approach using a numerical experiment and the comparisons against GenePix’s results. Meanwhile, the ELB-based image quantification still gives signal values in the traditional sense. In contrast, the “true signal values” estimated by the BlueFuse [22, 29], approach can neither be easily associated with mean nor median of pixel intensities.

What the author have not discussed is the role of different shapes of ELB configuration in segmentation and quantification. The shape of the ELB may also be an important factor in capturing the spatial difference in localization and quantification of
local noise levels. For example, for a rectangular sub-array shape, it might be a good idea to use shape configurations like those in Figure 3.10(c) or (d).

The future work will include optimizing the size of the ELB region on-the-fly for low computational cost and reasonable variation of variance in intensity estimates for all spots as mentioned in the previous the section. The author will also perform a more systematic comparison of results by ELB with different shape configurations and size definitions against those by other popular microarray image processing software.
In two-color SNP genotyping microarray experiments on diploid samples, there are three possible genotypes for each SNP locus: homozygotes “C/C”, “T/T” and heterozygote “C/T” on one strand. The core objective of all genotyping microarray experiments is to generate accurate genotypes. The “true” genotype for each data point is determined using two independent traditional biological methods including the Restriction Fragment Length Polymorphism (RFLP) [19] and direct sequencing [20, 21], which are the most reliable experimental methods for determining the genotypes. They all involve complicated procedures and very time-consuming. Logarithm ratio cutoff method [10] has also been applied in genotype determination.

4.1 Procedure of Genotype Determination Using Channel Intensities

Using traditional biological methods to determine genotypes involve complicated procedures and very time-consuming. It is not feasible in daily routine of research laboratories. For fast and reliable genotype determination, scientists directly use channel intensities as alternatives. Before performing any methods, the difference among experiment configuration should be eliminated. Several steps will be carried out:

1. Background extraction;
2. Channel normalization;
Although most of microarray image processing package offers background subtraction functionality, scientists still apply their own background subtraction methods for special purpose. Red and green channels may be imbalanced in digitization of the fluorescent signals, e.g., red signals are systematically stronger than green signals [45], which needs to be corrected using normalization techniques.

4.2 A Special Background Subtraction Method

This section will introduce a special background subtraction method [10], which is trying to avoid non-specific hybridization. Basically, it is a global background (GC) method, which determines the generalized background intensity for each channel, and then calculates the “true intensities” for each spot from channel actual intensity readings and background intensity.

It sorts the spots in descending order in terms of the ratio of intensities, $r/g$. The leading and trailing spots are tentatively classified as homozygotes “C/C” and “T/T”. The background level of green channel is computed as

$$v_{background\_green} = \text{mean}\{v_i | P_1, P_2, \ldots, P_n\}.$$  

And the background level of green channel is computed as

$$v_{background\_red} = \text{mean}\{v_i | P_{m+1}, P_{m+2}, \ldots, P_{m+n}\}.$$
(P_1, P_2, P_3, ..., P_n) is the heading spots and (P_{m+1}, P_{m+2}, P_{m+3}, ..., P_{m+n}) is the trailing spots. In other words, it calculates the green channel background from the brightest red signals and calculates the red channel background the brightest green signals.

4.3 Normalization

Each channel may have bias in digitization of the fluorescence signals, e.g., red signals may be systematically stronger than green signals [45]. Yang et al. [46], Quackenbush [47] and Bilban et al. [48] compared some normalization methods.

This section presents two effective preprocessing methods. The first one is used in Ref. [10]. It is similar to the background subtraction method mentioned in the last section. It also sorts the spots in descending order in terms of the ratio of intensities, r/g. The leading and trailing spots are tentatively classified as homozygotes “C/C” and “T/T”. The ratio,

\[ \varepsilon = \frac{\bar{r}}{\bar{g}} , \]

is computed as the normalization ratio, in which \( \bar{r} \) is the mean intensity of leading spots \( (P_1, P_2, P_3, ..., P_n) \) and \( \bar{g} \) is the mean intensity of the trailing spots \( (P_{m+1}, P_{m+2}, P_{m+3}, ..., P_{m+n}) \). In other words, it uses the brightest red signals and the brightest green signals to normalize two channel imbalances. Then, the normalized red intensity will be computed as
\[ R' = R / \sqrt{\varepsilon}, \]

and the normalized green intensity will be computed as

\[ G' = G \times \sqrt{\varepsilon}. \]

It increases the weaker channel intensity and decreases the stronger channel intensity.

The second normalization method is using heterozygotes to find the normalization ratio. It roughly finds heterozygotes, and then calculate the normalization ratio as

\[ \varepsilon = \sum r_i / \sum g_i, \]

in which \( \{ i \mid i \in \text{heterozygote} \} \). The idea is based on the MA-plot as shown in Figure 9. The blue points denote heterozygotes. If without the channel imbalance, blue points should be balanced in the line \( y = 0 \). The author uses this line to find out the normalization ratio to balance red and green channels.
4.4 Pattern Recognition

For two-color genotyping microarray, genotype determination is a pattern recognition problem. The actual intensities of red and green channels are projected in Figure 4.2. From it, you can observe that in the bottom-left, there classes are overlapped, so researchers often use logarithm projection to deal with such a classification problem as shown in Figure 4.3, in which three classes are more separable. The other advantage is that it does not involve large number calculation. Nowadays, Support Vector Machine (SVM) and Neural Network (ANN) attract researchers’ great attention in pattern recognition.
Figure 4.2 Intensity projections from red and green channels.

Figure 4.3 Logarithm intensity projection from red and green channels.
4.4.1 Support Vector Machine (SVM)

![Image of SVM](image)

**Figure 4.4** Basic SVM.

In 1995, Cortes et al. [49] introduced the two-group SVM into classification. For a given set of labeled training patterns \((\tilde{x}_1, y_1), (\tilde{x}_2, y_2), \ldots, (\tilde{x}_i, y_i), \ldots, (\tilde{x}_j, y_j)\), where \(y_i \in \{-1, 1\}\), the objective of SVM classifier is to find an optimal hyper-plane: \(\tilde{w}_0 \cdot \tilde{x} + b_0 = 0\), where \(\tilde{w}_0 \in R^N\) and \(b_0 \in R\), that separates two classes by maximizing the margin as shown in Figure 4.4, in which the two green lines are the margin of hyper-planes. It corresponds to the decision function

\[
    f(\tilde{x}) = \text{sign}(\tilde{w}_0 \cdot \tilde{x} + b_0).
\]

To find the solution, they used a standard optimization technique, namely the Lagrange multiplier. You can construct the Lagrangian function as

\[
    L(\tilde{w}, b, \Lambda) = \frac{1}{2} \tilde{w} \cdot \tilde{w} - \sum_{i=1}^{i} \alpha_i [y_i (\tilde{x}_i \cdot \tilde{w} + b) - 1].
\]
where $\Lambda^T = (\alpha_1, \ldots, \alpha_i)$ denotes the Lagrange multiplier under the constraint $y_i(\hat{\omega}_0 \cdot \bar{x} + b) \geq 1$. To find the minimum, it differentiates the Lagrange multiplier with respect to $\hat{\omega}$ and $b$. You obtain

$$\hat{\omega}_0 = \sum_{i=1}^{l} \alpha_i y_i \bar{x}_i \quad \text{and} \quad \sum_{i=1}^{l} \alpha_i y_i = \Lambda^T Y = 0.$$  

The Lagrangian multiplier can be changed into

$$W(\Lambda) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \bar{x}_i \cdot \bar{x}_j = \Lambda^T \frac{1}{2} \Lambda^T D \Lambda,$$

where $D_{ij} = y_i y_j \bar{x}_i \cdot \bar{x}_j$. According to $\partial W / \partial \Lambda = 0$, it can find $\Lambda$, and then take it into

$$\hat{\omega}_0 = \sum_{i=1}^{l} \alpha_i y_i \bar{x}_i$$

to obtain the desired $\hat{\omega}_0$. Finally, using Kuhn-Tucker theory, an equality

$$\alpha_i [y_i (\bar{x}_i \cdot \hat{\omega}_0 + b_0) - 1] = 0$$
will hold at the saddle point to find the intercept $b_0$. Therefore, the complete optimal hyper-plane is decided. Note that according to this equality, only those points, which satisfy
\[ y_i(\mathbf{x}_i \cdot \mathbf{w}_0 + b_0) - 1 = 0, \]
correspond to non-zero $\alpha_i$. It means that only the samples on margins are effective for the optimal solution. They are called support vectors; others take no effect. That is the basic difference between SVM and other commonly used classification techniques, such as Principle Component Analysis (PCA) [50, 51], Linear Discriminant Analysis (LDA) [50-52], and Neural Networks (NN) [53], in which the distribution and density functions play important roles.

Most of cases, input data cannot be separated linearly in input space. You can map it to high dimensional feature space through the mapping function $\phi(\cdot)$, in which
\[ D_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j \]
maps into
\[ D_{ij} = y_i y_j \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j), \]
while classes can be separated linearly in this new feature space. In practice, both situations involve computational complexity. Generally, it is equivalent that replacing the mapping
by kernel functions $k(\tilde{x}_i, \tilde{x}_j)$ [54]. Commonly used kernel functions are radial basis function (RBF):

$$k(\tilde{x}_i, \tilde{x}_j) = \exp\left(-\frac{|\tilde{x}_i - \tilde{x}_j|^2}{2\sigma^2}\right)$$

and polynomial kernel function

$$k(\tilde{x}_i, \tilde{x}_j) = (\tilde{x}_i \cdot \tilde{x}_j + \beta)^d.$$  

Applying kernel functions reduces the running time from $O(n^2)$ to $O(n)$. The main advantage for using kernel function is to distinguish non-linear separable data in input space non-linearly.

SVM is originally designed as a binary (or two-class) classifier. Researchers have proposed extensions to multi-class classification by combining multiple SVMs, such as one-against-all (Figure 4.5), one-against-one (Figure 4.6) and DAG-SVM. One-against-all [55] method is to learn $k$ binary SVMs during the training stage. For the $i$-th SVM, it labels all training samples by

$$y(j)_{j=1} = +1 \text{ and } y(j)_{j=1} = -1$$

to obtain the hyper-plane
for the $i$-th SVM. For finding the $i$-th hyper-plane, the optimization problem

$$
(w^i)^T \phi(\bar{x}) + b^i = 0
$$

is involved, where $\xi$ is a penalty that can reduce training errors. In the testing, the decision function is obtained by

$$
\text{Class of } \bar{x} = \arg \max_{i=1,2,...,k} ((w^i)^T \phi(\bar{x}) + b^i)
$$

It was shown that you can solve the $k$-group problem simultaneously [56].

Figure 4.5 One-against-all SVM.
One-against-one method \cite{57} (Figure 4.6) is to construct \(k(k-1)/2\) SVMs in a tree structure. Each SVM represents a distinguishable pair-wise classifier from different classes in the training stage. The optimization problem

\[
\min \left( \frac{1}{2} w^{ij^T} w^{jj} + C \sum_{i=1}^{l} \xi_{ij} \right)
\]

is involved. In the testing, each leaf SVM pops up one desirable class label, which propagates to the upper level in the tree until it processes the root SVM of the tree.

The Directed Acyclic Graph (DAG) SVM \cite{58} is to combine multiple binary one-against-one SVM classifiers as same as one-against-one fashion. It also has \(k(k-1)/2\) internal SVM nodes, but in the testing phase, it starts from the root node and moves to the left or right sub-tree depending on the output value of the binary decision function until a leaf node is reached, which indicates the recognized class. There are two main differences between one-against-one and DAG methods. One is that one-against-one needs to evaluate all \(k(k-1)/2\) nodes, while DAG only needs to evaluate \(k\) nodes due to
the different testing phase schema. The other is that one-against-one uses bottom-up approach, but DAG uses top-down approach.

Taskar, et al. [59] which combined both the advantages of the graphical models (maximum margin Markov (M3-net) networks that can be triangulated tractably) and the kernel SVM to solve the problem multi-class supervised classification. They triangulated graph with the advantage of the sequential minimal optimization. Lee et al. [60] proposed MSVM. The motive is to design an optimal multi-category SVM, which continues to deliver the efficiency of the binary SVM. With this intent, they devise a loss function with suitable class codes for the multi-category classification problem. C Angulo et al. [61] introduced the “Support Vector Classification-Regression” machine for K-class classification purposes (K-SVCR), the new training algorithm with ternary outputs \{-1; 0; +1\} based on Vapnik’s Support Vector theory. Shih and Kai developed a new kind of SVM multi-class recognition system, support vector machine networks [62], in which the author combines the SVM and ANN and take advantages from both of them to solve the classification problem.

In this genotyping classification problem, in which there are only three classes and two dimensions for the training and testing data, the author uses the simple one-against-one extension of SVM with linear kernel.
4.4.2 Neural Networks

A layered neural network is a network of neurons organized in the form of layers. The first layer is the input layer. In the middle, there are one or more hidden layers. The last one is the output layer. The architecture of a layered neural network is shown in Figure 4.7. The function of the hidden neurons is to intervene between the external input and the network output. The output vector $\bar{Y}(\bar{x})$ is

$$\bar{Y}(\bar{x}) = \sum_{i=1}^{l} f_i(\bar{x})w_{ik} = \bar{F} \cdot W,$$

where $\bar{F}$ is the hidden layer’s output vector corresponding to a sample $\bar{x}$, $W$ is the weight matrix with dimensions of $l \times k$, $l$ is the number of hidden neurons and $k$ is the number of output nodes. In such a structure, all neurons cooperate together to generate a single output vector. In other words, the output does not only depend on each individual neuron, but on all of them.
There are two categories in neural networks. One is supervised learning and the other is unsupervised learning. For supervised learning, the target responses are known for the training set, but for unsupervised learning, you do not know the desired responses. In this work, the author uses supervised learning.

### 4.5 Cutoff Method

To solve the genotype-calling problem, one can use a simple linear cutoff-based method [10] based on the MA-plot as shown in Figure 4.8. In the cutoff-based method, good global thresholds must be chosen as empirical cutoff values (±1.5). Spots with log ratios within a certain range will be given a genotype.

**Figure 4.8** The linear cutoff based method in a two-color genotyping microarray. The x-axis is \( \log(R \times G) \) and y-axis is \( \log(R \div G) \). ±1.5 is the cutoff values.
4.6 Iterative Learning (GenoIterSVM and GenoIterANN)

Classification accuracy could be improved by introducing iterative refinement in the learning and testing stages. The resulting algorithm is termed as GenoIterSVM [63], which consists of a basic SVM as the core and an iterative procedure for improving classification accuracy. With this algorithm, the data are iteratively re-normalized during the learning and the blinded testing stages. Iterative leaning is aim at eliminating variations among different experiments with different configurations.

In SNP genotyping microarrays, usually there are plenty of data points representing heterozygous alleles at DNA loci. Ideally, these heterozygous data should be balanced along the bisector of the first quadrant of the coordinate system, while this is not always true. It can impose an artificial regularity constraint: the optimal curve fitting for these heterozygote data using linear regression analysis must be collinear with \( y = x \). If this regularity constraint is not met, the author considers that a systematic bias is inherent and must be corrected. Figure 4.6(a) shows that the collinear constraint is not met for the SVM classifiers obtained after the initial learning stage. The author corrects this systematic bias iteratively by shifting and rotating the coordinate system, until this constraint is met. This iterative learning process results in a system of canonical classifiers, as shown in Figure 4.9(b). The author uses a similar iterative approach in the blinded testing step. That means all testing data also need to iteratively adjust to respect the collinear constraint. He found that in most cases it is enough to achieve convergence after one iteration. Using such a procedure, you are able to eliminate the human intervention and system difference, such as setting any empirical thresholds, and obtain high accuracy, reliability and efficiency [63]. The procedure is illustrated in Figure 4.10. The author also applies ANN as the core
classification kernel to perform the same iterative learning, which is termed as GenolterANN.

**Figure 4.9** SVM classifiers using (a) the linear kernel and (b) those with iterative adjustment.

**Figure 4.10** The procedure of iterative learning of GenolterSVM/ GenolterANN.
In principal, you could determine the genotypes for all DNA loci using other independent genotyping methods such as RFLP and direct sequencing, however, in reality it is not feasible to perform such methods since they are very time-consuming. To validate the classification accuracy of the auto-calling algorithms, it calculates the concordance rate, and link it to classification accuracy. Experiments are designed to genotype from both sense and anti-sense directions, and it measures the concordance rate of results from both directions. The concordance rate is defined as

\[ R_{\text{concordance}} = \frac{|A \cap B|}{|A|}, \]

in which \(A\) and \(B\) represent the genotypes from sense and anti-sense directions, respectively. Ideally, \(A\) and \(B\) should present the same genotypes for the same location (SNPs). Higher concordance rate means higher recognition rate. The experiments include about 4,600 SNPs covering 12 chromosomes in at least 24 genomic DNA samples and 30 human single sperms. Experimental results will be reported in the next chapter. From the experiments, GenoIterANN and GenoIterSVM are the best genotyping methods, which perform best. The results of ANN and SVM without iterative adjustment are slight lower than the iterative learning methods, but they still outperform the simple cutoff methods.
Genotyping microarrays establish the genotypes of DNA sequences containing single nucleotide polymorphisms (SNPs), and can help biologists probe the functions of different genes and/or construct complex gene interaction networks. An enormous amount of data from these experiments makes it infeasible to perform manual processing to obtain accurate and reliable results in daily routine. Advanced algorithms as well as an integrated software toolkit are needed to help perform reliable and fast data analysis. The author developed a MATLAB™ [64] based software package, called TIMDA (a Toolkit for Integrated Genotyping Microarray Data Analysis), for fully automatic, accurate and reliable genotyping microarray data analysis. He also developed new algorithms for image processing and genotype-calling. The modular design of TIMDA allows satisfactory extensibility and maintainability. TIMDA is an open source (URL: http://timda.SF.net), and can be easily customized by users to meet their particular needs.

5.1 Modular Design

High-throughput genotyping microarray systems generate a large volume of raw data. It is not feasible to manually process these data and yield accurate and reliable results in daily routine. Advanced algorithms as well as an integrated software tool are needed to be developed to help performing reliable and fast analysis. The author developed a MATLAB based software package, TIMDA (a Toolkit for Integrated Genotyping Microarray Data
Analysis), for fully automatic, accurate and reliable genotyping microarray data analysis based on the new algorithms for image processing (IP) and genotype-calling (GC). TIMDA consists of several modules performing different tasks. The schematic representation of the modular design of TIMDA is shown in Figure 5.1. The IP module employs several novel and effective techniques for gridding, segmentation and quantification to obtain the correct gene expression level from microarray images. Since all subsequent analysis relies on accurate results of IP, IP is a critical step in microarray data analysis. The genotype-calling (GC) module is used to perform auto-calling of genotypes using the novel machine-learning based approaches including support vector machines (SVMs) [49] and artificial neural networks (ANNs) [53]. The loss-of-heterozygosity (LOH) module is used to establish the LOH pattern using microarray data for normal cells and diseased cells, which can be used for investigating the genetic changes. Inside the LOH module, TIMDA also integrates the replication checking, which probably happens in one direction or two directions (sense direction or anti-sense direction).

All modules are launch from the GUI interface. After users set the basic configurations, (like which image or which data file(s) you want to analyze) TIMDA will start to process and generate the text reports or graphic outputs when the tasks are completed.
5.2 Image Processing (IP) Module

The image processing module performs automatic gridding, segmentation of foreground and background and quantification of intensities. Currently, IP module supports the TIFF image format, which contains digitized images scanned from two channels, red and green and also is a commonly used image format in microarray printers. Image processing has been a bottleneck for reproducibility, accuracy and efficiency of microarray data analysis due to the intrinsic difficulties. The design of IP module and the use of the ELB model improve the quality of image processing with some other novel features as well.

The first step of IP module is gridding which is trying to index microarray layouts despite of possible rotations of the scanned microarray images. IP module uses the proposed the methods to performs automatic image rotation detection and correction as an
option after loading the input image as shown in Figure 5.2. IP module also can perform automatic image rotation detection and correction on each sub-array. Because the rotation detection and correction take considerable time, TIMDA doesn’t suggest user to perform them on all images if the situation of rotation is not severe.

![Image of gridding sub-module in the IP module](image)

**Figure 5.2** The gridding sub-module in the IP module.

Traditionally, a manually selected threshold value is used to convert the original image to a binary image for constructing a spot layout template more easily. For automatic processing purpose, IP module uses the proposed new non-parametric method to find the optimal threshold. From the experiences, usually, optimal threshold values for horizontal and vertical axes may be slightly different. According to the projections, such as Figure 3.4(b), layout parameters, which include the number of columns and rows, the interval between adjacent columns and rows, the interval between adjacent sub-array and approximate spot radius, are calculated. If the gridding result is not satisfied, users are
allowed to sliding bar to manually adjust the threshold to achieve the best result. Then, a binary layout template is determined and then using pre-defined the filter (Figure 3.7(b)) the fine-tuning step is carried out to finally determine the location for each spot. If the image has been rotational corrected in the last step, the layout should be converted back in the original image.

In internal implementation, TIMDA uses MATLAB’s workspace (which consists of the set of variables built up during a MATLAB session and stored in memory) to pass the results to the next step (or module), which is the ELB sub-module. The location of each spot will be passed by as the moist important parameter. The ELB sub-module is shown in Figure 5.3.

![Figure 5.3 The interface of ELB sub-module.](image)

Before performing the ELB sub-module, IP module ask users to configure the ELB definition first. If users choose define ELB spot-wised, they can define ELB row and ELB
column in Figure 5.4. If they want to define it pixel-wised, they can either define the ELB shape rectangle or ellipse as shown in Figure 5.4. Figure 3.10 illustrates an allowable ELB configuration. The center of ELB is the location of each spot. The covered area excluding the signal region will be the considered background region.

![ELB pixel-wised definition](image)

**Figure 5.4** The interface of the ELB pixel-wised definition.

It is commonly believed that the pixel intensity of the foreground reflects the joint effects of the hybridization level of the gene and the reflected background intensity of the glass surface. According to the methods mentioned in Chapter 2, the background and foreground intensity levels will be estimated.

As technology advances, the density of spots on microarray chips continues to increase. With fewer pixels used in the local background estimation of traditional approaches, they cannot make significant statistical sense and result in unreliable
estimates. To overcome this limitation, ELB is a very good choice which is a novel, viable and feasible method for robust microarray image segmentation and quantification.

The interface of the ELB sub-module also allows users to monitor each individual spot in the microarray image, e.g., background/foreground intensity, standard deviation, local cut-off values, the number of pixels in the foreground region, etc. It is a very useful functionality when users want to find out information about specific spots instantly. A plot of the background distribution (Figure 5.5) over the whole image will be popped out to notify users the complete gene expression report having been generated.

Figure 5.5 The distribution plot notifies the ELB process is done.
5.3 Intermediate Data Conversion (IDC) Module

IDC module can be used to convert analysis results from other software tools into a generalized MATLAB’s workspace for further processing, such as genotype determination and LOH studies. All text reports of TIMDA are highly structured and could be opened by any text editors or spreadsheet application such as Microsoft™ Excel™. It is necessary to construct a middle layer which integrates different text outputs into MATLAB for subsequent analysis. Meanwhile, it is very important for researchers to associate IDs with their microarray experiments, so that they can identify each spot in the microarray images. Some software has already offered ID association functionality. IDC will inherit its ID association when they need to be further processed. If the loaded gene expression results without such an association, such as IDC will help user generate it from an ID association file. Table 5.1 shows the example format of an ID association file. IDC module then helps users to convert data into the identical format. Figure 5.6 shows the interface of IDC module. IDC collects data from loaded text files, which include block, column, row, ID, x and y coordinates, red and green intensities, red and green background intensities. Some known formats have been stored in IDC to enhance efficiency. A generalized MATLAB’s workspace file (.mat file) will be created at the end of IDC session. Logarithm plot (Figure 5.7) will be shown to notify users the process having been completed.
Figure 5.6 IDC interface.

Table 5.1 Example Format of an ID Association File

<table>
<thead>
<tr>
<th>Block</th>
<th>Column</th>
<th>Row</th>
<th>Reserved</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
<td>01T027</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1</td>
<td></td>
<td>01T027</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td></td>
<td>01T030</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1</td>
<td></td>
<td>01T030</td>
</tr>
<tr>
<td>1</td>
<td>5</td>
<td>1</td>
<td></td>
<td>01T032</td>
</tr>
</tbody>
</table>
Figure 5.7 The logarithm distribution plot notifies the IDC process is done.

5.4 Genotype Calling (GC) module

In two-color SNP genotyping microarray experiments on haploid samples, there are three possible genotypes for each SNP locus: homozygotes “C/C”, “T/T” and heterozygote “C/T” on one strand. The core objective of all genotyping microarray experiments is to generate accurate genotypes. GC module is designed to help users to obtain correct genotypes by applying different background subtraction, normalization and genotype calling methods.

For background processing, GC module recommends users to use the true intensity levels from the last module. However, it still preserves two background processing methods for comparison purpose. One is the global background subtraction, which was
mentioned in Chapter 3, and the other directly uses the intensity readings without processing background. These two options are rarely used.

For normalization methods, GC module also offers three options. The first option is finding the normalization ratio from the brightest spots of both channels according to their ratios; the second one is finding the normalization ratio from the heterozygote; the third one is no normalization applied for the comparison purpose. The recommendation is option 2.

GC module integrates a set of algorithms listed in Table 5.2 for genotype-calling. The recommendation is options 4 and 5 of Table 5.2, which use GenoInterSVM and GenoIterANN maps with training data for iterative adjustment. The results are reported in Table 5.3.
Table 5.2 Genotyping Methods.

<table>
<thead>
<tr>
<th>Options</th>
<th>Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MA-plot; using ±1.5 as cutoff values</td>
</tr>
<tr>
<td>2</td>
<td>Using GenoInterSVM map; but without training data iterative adjustment</td>
</tr>
<tr>
<td>3</td>
<td>Using GenoInterANN map; but without training data iterative adjustment</td>
</tr>
<tr>
<td>4</td>
<td>Using GenoInterSVM map with training data iterative adjustment</td>
</tr>
<tr>
<td>5</td>
<td>Using GenoInterANN map with training data iterative adjustment</td>
</tr>
<tr>
<td>6</td>
<td>Using $y = x \pm 2$ as the hyper planes to separate three classes in the logarithm intensity map</td>
</tr>
<tr>
<td>7</td>
<td>Using user defined map to do classification in the logarithm intensity map</td>
</tr>
</tbody>
</table>

Table 5.3 Genotyping Results with Different Normalization and Genotype Methods.

<table>
<thead>
<tr>
<th>Concordance Rate</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without normalization</td>
<td>90.56%</td>
<td>90.97%</td>
<td>91.64%</td>
<td>91.22%</td>
<td>91.86%</td>
<td>90.56%</td>
</tr>
<tr>
<td>Using the brightest spots to find the normalization ratio</td>
<td>90.95%</td>
<td>91.26%</td>
<td>91.93%</td>
<td>91.51%</td>
<td>92.14%</td>
<td>90.95%</td>
</tr>
<tr>
<td>Using the heterozygote spots to find the normalization ratio</td>
<td>90.99%</td>
<td>91.27%</td>
<td>92.09%</td>
<td>91.53%</td>
<td>92.12%</td>
<td>90.99%</td>
</tr>
</tbody>
</table>

Table 5.3 compares the results of using different normalization and genotype calling methods. From the table, the concordance rates show the worst results if not applying any normalization methods. That means the channel unbalance does exist, and it
affects the meaningful gene expression levels. In this table, the best normalization method is the one uses heterozygote spots to find the normalization ratio. For genotype calling methods, GenoIterANN and GenoIterSVM perform the best. The results of ANN and SVM without iterative adjustment are slightly lower than the iterative testing methods, but they still outperform the simple cutoff methods.

For illustrating the effect of normalization, the author also performs the manually manipulated data on the same test. The result is listed in Table 5.4. The author manually increases the intensity of the red channel in sense direction and decreases the intensity of the green channel in anti-sense direction. From it, you can observe the normalization process is really needed to eliminate the channel imbalance.

**Table 5.4 Genotyping Results for Manipulated Data.**

<table>
<thead>
<tr>
<th>Concordance Rate</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Without normalization</td>
<td>78.4%</td>
<td>88.1</td>
<td>87.0%</td>
<td>90.3%</td>
<td>87.3%</td>
<td>78.4%</td>
</tr>
<tr>
<td>Using the brightest spots to find the normalization ratio</td>
<td>80.1%</td>
<td>87.4%</td>
<td>88.3%</td>
<td>91.3%</td>
<td>88.7%</td>
<td>80.1%</td>
</tr>
<tr>
<td>Using the heterozygote spots to find the normalization ratio</td>
<td>90.5%</td>
<td>90.5%</td>
<td>90.4%</td>
<td>91.5%</td>
<td>89.7%</td>
<td>90.5%</td>
</tr>
</tbody>
</table>

The learning algorithms are better methods for the classification purpose than simple cutoff methods. Iterative learning eliminates the system bias and further improves the recognition rate. In practice, users can apply multiple microarray experiments with the same configuration, and then manually select a set of training data to construct GenoIterANN and GenoIterSVM. In such a manner, GenoIterANN and GenoIterSVM could generate better classifiers for subsequent analysis. In the testing stage, users can apply replications in a single batch or use sense and anti-sense directions to avoid errors.
Figure 5.8 GC interface.

Figure 5.9 Genotype layout.
Figure 5.8 shows the GC interface, in which users can easily configure the genotype calling options. When the process is completed, the genotype layout will be displayed as shown in Figure 5.9.

5.5 Loss-of-Heterozygosity (LOH) Module

Loss-of-heterozygosity (LOH) patterns reveal genetic off/on regulation in tumor samples in comparison with the normal samples. TIMDA’s LOH module is designed to help researchers to analyze the results between normal and disease samples.

The LOH module offers two approaches to analyze LOH: index-based and ID-based. The index-based approach is based on the indexed spot layout, while ID-based approach is based on ID associations. The LOH experiments are designed to determine genotypes from both normal and disease samples, which are also normally used in other laboratories. If the ID-based approach is used before processing LOH comparison, the ID associations for both directions are determined. Furthermore, replications are also supported to minimize system errors. Replication analysis is only supported in the ID-based approach. For example, for double-replication, ID associations are determined for replicating genes first. If both genes are shown as the same genotypes, they will be preserved under the same ID; otherwise, the ID will be discarded. In the comparison step, two properties are concerned. One is the real loss of heterozygosity, i.e., the heterozygotes change to homozygotes after genotype determination; the other is that the change of the logarithmic ratios of green to red intensities for heterozygotes in both samples is beyond a certain threshold.
\{p(i) \mid \lambda_i = \left| \log\left( \frac{r_{\text{normal}_i}}{g_{\text{normal}_i}} \right) - \log\left( \frac{r_{\text{disease}_i}}{g_{\text{disease}_i}} \right) \right| > \sigma \}

The graphic output of LOH ID-based analysis is shown in Figure 5.10. Index-based comparison completes the same comparison with the graphic output as shown in Figure 5.11. The text report is shown in Table 5.4. The marked spots are heterozygosities in the normal samples. The solid stars mean, these genes are heterozygosities for both samples and the logarithm ratio changes are below the threshold; while the hollow stars mean they are still heterozygosities, but their logarithm ratio changes are above the threshold. The other hollow shapes indicate genotypes have changed from heterozygosities in the disease sample.

Figure 5.10 LOH logarithm comparison plot. Black dots are heterozygosities (C/T) in the original samples; red triangles are “C/C” genotypes changed from original (C/T) genotypes; green quadrangles are “T/T” genotypes changed from original (C/T) genotypes; yellow
disks are still “C/T” genotypes whose logarithm ratio change above a threshold. Three regions indicate the classifying map.

Figure 5.11 The graphical output for index-based analysis of LOH module. The graphic output indicates the LOH changes. All marks means in the location, they are heterozygosities (C/T) in the normal samples. Circles means they change to “C/C” in the tumor sample; quadrangles are “T/T”; hollow stars mean they are still “C/T”, but their ratio change over a certain threshold; solid stars means “C/T” ratio changes below the certain threshold.
Table 5.5 Example Format of a LOH Report

<table>
<thead>
<tr>
<th>ID</th>
<th>normal_R</th>
<th>normal_G</th>
<th>ratio</th>
<th>genotype</th>
<th>Disease_R</th>
<th>disease_G</th>
<th>ratio</th>
<th>genotype</th>
</tr>
</thead>
<tbody>
<tr>
<td>02T224</td>
<td>798.3037</td>
<td>349.2579</td>
<td></td>
<td>C/T</td>
<td>4801.4242</td>
<td>625.0781</td>
<td></td>
<td>C</td>
</tr>
<tr>
<td>02T489</td>
<td>6997.132</td>
<td>3170.264</td>
<td>0.79169</td>
<td>C/T</td>
<td>1125.5331</td>
<td>5717.9437</td>
<td>-1.625</td>
<td>C/T</td>
</tr>
<tr>
<td>01T267</td>
<td>4855.682</td>
<td>12622.18</td>
<td></td>
<td>C/T</td>
<td>80.956</td>
<td>992.5691</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>01T247</td>
<td>10870.9</td>
<td>62777.6</td>
<td>-1.7535</td>
<td>C/T</td>
<td>54273.854</td>
<td>11909.848</td>
<td>1.5167</td>
<td>C/T</td>
</tr>
<tr>
<td>01T213</td>
<td>49436.95</td>
<td>63894.23</td>
<td></td>
<td>C/T</td>
<td>114.8103</td>
<td>8253.0918</td>
<td></td>
<td>T</td>
</tr>
<tr>
<td>01T564</td>
<td>153.6735</td>
<td>342.2727</td>
<td></td>
<td>C/T</td>
<td>6434.2813</td>
<td>576.9952</td>
<td></td>
<td>C</td>
</tr>
</tbody>
</table>
5.6 OneInt Module

TIMDA minimizes human intervention. In the gridding stage of IP module, maximizing variances achieves non-parametric thresholding and automatic rotation detection and correction. Applying the global background and local background estimation in histogram-based segmentation makes the ELB processing fully automatic. Using machine learning algorithms can avoid preset thresholds without paying accuracy cost. Considering users’ convenience, the OneInt module can generate genotype results from input images directly. The OneInt interface is shown in Figure 5.12.

![Figure 5.12](image)

**Figure 5.12** The interface of OneInt module.
5.7 Implementation

In this section, the author will introduce some of implementations of TIMDA in detail.

5.7.1 Rotation Detection and Correction

Previous sections have introduced the algorithm for rotation detection and correction in Chapter 3. To calculate the intensity projections according to different rotation angles, the original image should be rotated in a certain degree. After the image has been rotated, the image appears larger than the original size. In Figure 5.13, the size of the original image is $100 \times 100$ and the size of the rotated image is $109 \times 109$. The boundary of the rotated image also affects the calculation of intensity projections.

![Figure 5.13](image)

(a) An original image; (b) the rotated image with $5^\circ$.

In the actual implementation, it uses a mask with the same size of the original image, which is rotated with the original image simultaneously. The boundary effect could be eliminated using the mask “AND” the rotated image as following:

$$I' = I_{rotated} \cap I_{mask\_rotated}.$$
The intensity levels of the boundary in the rotated mask are “0”. After “AND” relation, the intensity levels of the boundary in the rotated image are also “0”, which won’t contribute to the intensity projections along both axes (horizontal and vertical).

The rotation determination will be based on the rotated image (which is large than the original image). The locations of all spots are measured in the rotated image. They have to be converted back in the original image. Those locations are stored in a 2-D array, in which the first dimension is the index of vertical direction, the second dimension is the horizontal direction followed by the MATLAB’s convention. In the ELB sub-module, TIMDA allows users to monitor and fetch the statistical information of each individual spot. At that time, the 2-D array will be converted to a 4-D array. The first dimension is the sub-array index of vertical direction; the second dimension is the horizontal direction; the third dimension is the index of vertical direction in each sub-array; the fourth dimension is the horizontal direction in the sub-array. From such a structure, TIMDA can easily access each individual spot. In the reporting process, TIMDA follows the analysis report of GenePix [43], which uses 3 dimensions. The first dimension is the index of each sub-array; the third dimension is the index of vertical direction in each sub-array; the third dimension is the horizontal direction in the sub-array.

5.7.2 Customized Format Loading

In IDC module, TIMDA allows users to load some commonly used file format convention to convert the data analysis report into a generalized MATLAB workspace. For such known report formats, know the exact locations for the needed information. After setting the format, IDC module will fetch the useful data from those customized location in the report.
5.7.3 ID Loading and Association

Without ID association, those microarray spots do not have any biological meanings, so IDC module requires the ID association loading. In this module, users can load their corresponding ID association file to give the biological meaning to each spot. The sample ID association file is shown in Table 5.1. You can find the indices that only have three dimensions instead of four dimensions. It follows the GenePix convention. Its sub-array numbering is defined in Figure 5.14.

![Sub-array numbering](image)

**Figure 5.14** The sub-array numbering.
CHAPTER 6

SUMMARY AND FUTURE WORKS

The previous chapters briefly introduce microarray technology and present new approaches for microarray image processing and genotype determination. They also introduce the new integrated software package, TIMDA, for genotyping microarray analysis.

For the microarray image processing, the ELB concept is used to enlarge the background for segmentation and quantification. The ELB is a novel and viable method. It not only uses a flexible larger local region, but also applies the global background and local background thresholds as cutoff values for estimation, that can eliminate some segmentation errors and make more statistical sense.

For genotype determination, the author applies iterative learning to approaches eliminate human intervention and errors, and the learning theories are used to find the optimal classifiers.

In the future research, the author intends to further validate the ELB concept and compare the results with other microarray image processing software packages as well as the methods of different normalization and genotyping methods. For TIMDA, it also needs to be further developed. It includes modular design, method development, and the user-friendly interface.
APPENDIX A

EFFICIENT CONTOUR DETECTION BASED ON IMPROVED SNAKE MODEL

A traditional snake [65] is a contour \( v(s) = [x(s), y(s)] \), where the parameter \( s \in [0, 1] \), moves through the spatial domain of an image to minimize the energy function given by

\[
E_{\text{snake}} = \sum_{i=1}^{n} \{ \alpha(s) \times E_{\text{cont}}[v_i(s)] + \beta(s) \times E_{\text{curv}}[v_i(s)] + \gamma(s) \times E_{\text{image}}[v_i(s)] \},
\]

where \( \alpha, \beta \) and \( \gamma \) are weighting coefficients that control the snake’s tension, rigidity and attraction, respectively. The first and second terms are correspondingly the first- and second-order continuity constraints. The third term measures the edge strength (i.e., the image force).

After analyzing the original snake model, find that there are some problems that cannot be solved properly. The first one is that the snake cannot move toward the object if it is too far away from the actual boundary. The second one is that the snake cannot move into boundary concavities. The third one is that the snake is attracted to local strongest edges and suffered from a great disadvantage such that when an object resides in a complex background, the strongest edges are often not the edges of the object are looking for.

In physics, the objects attract each other with a force, which is proportional to the mass product and inversely proportional to the distance between mass centers. The new gravitation external energy field, called the gravitation energy field, can be represented as
where $\vec{r}$ is the position vector, $\hat{r}$ is the unit vector, and $g(\vec{r})$ denotes the first- or second-order derivative. Note that the pixels having the maximum values of the first- or second-order derivative are edges. They make the attractive force point toward the object.

This new technique [11] combines the gravitation force field with edge preserving smoothing to move the snake toward the object boundary including concavity. Our gravitation force field uses gradient values as particles to construct force field in the whole image. This force field will attract the contour toward the edge boundary. The locations of the initial contour are very flexible, such that they can be very far away from the objects and can be inside, outside, or the mixture. The improved snake can converge toward the object boundary in a fast pace.

$$E_{gravitation} = \int_{\Omega} \frac{g(\vec{r})}{||\vec{r}||} \hat{r} d\vec{r},$$
A traditional Image metamorphosis deals with the fluid transformation from one digital image into another. This technique, commonly referred to as morphing, has found widespread use in the entertainment industry to achieve stunning visual effects. Smooth transformations are realized by coupling image warping [66, 67].

In computer animation, are frequently faced with the problem of knowing that an object is located at point $P_1(x_1, y_1, z_1)$ of frame $F_1$, at point $P_2(x_2, y_2, z_2)$ of frame $F_2$, and the object has to be moved from $P_1$ to $P_2$ as the frame count proceeds from $F_1$ to $F_2$. Computing these inbetween values is called interpolating. There are some widely-used interpolation methods: linear, non-linear and space curves interpolations. Linear interpolation (Figure B.1), also called lerping, is the simplest way to generate inbetween frames. Although lerping generates continuous motion, it does not generate continuous derivatives, so there may be abrupt changes in velocity when lerping is used to interpolate positions. Because of the drawback of lerping, non-linear interpolation has been used. It can be used to vary any parameter smoothly as a function of time. Non-linear interpolation methods can make individual points (or individual objects) move smoothly in space and time, but they are vertex-oriented. It means they consider each point under the time constraint, but do not consider the shape of objects. In this paper, present a new interpolation method by using energy-minimization function. From the energy function, can consider both the time constraint and the shape of the object in each frame simultaneously. In an animation system, the image warping is completed through feature
points of objects. The feature points which explicitly describe the object contours can be represented by the control points.

Shih and Kai present a new energy function \((E(t, p))\) to measure the energy of such a framework and find the minimum energy to be the optimal solution.

![Figure B.1 Inter-frames by using linear interpolation.](image)

\[
E(t, p) = \sum_{i=2}^{N-1} \sum_{j=2}^{M-1} \left[ \alpha(t) E_{\text{time}}(t_i, p_{ij}) + \beta_c E_{\text{curve}}(t_i, p_{ij}) \right],
\]

where \(\alpha(t)\) is the time-controlling factor and \(\beta_c\) is the curve-controlling factor. Let \(E_{\text{time}}\) denote the vertex-oriented energy term, which measures the corresponding control points in each inter-frame. It can be represented as
\[
E_{\text{time}}(t, p_y) = \frac{|\bar{d} - |p_{i,j} - p_{i-1,j}||}{\max\{|\bar{d} - |p_{i,j} - p_{i-1,j}||\}},
\]

where \( \bar{d} \) is the average of lengths between each pair of adjacent control points on the time related frame, represented as

\[
\bar{d} = \frac{\sum_{i=1}^{N} |p_i - p_{i-1}|}{N-1}.
\]

The time-controlling factor \( \alpha(t) \) controls the intervals between each adjacent pair of frames. If \( \alpha(t) = \text{constant} \), the factor will generate linear time-related interpolations. If \( \alpha(t) = f(t) \), the factor will generate non-linear time-related interpolations.

\( E_{\text{curve}} \) is the curve-oriented energy term, which measures the energy of the curves in each frame. It can be represented as

\[
E_{\text{curve}} = \frac{|p_{i,j-1} - 2p_{i,j} + p_{i,j+1}|}{\max\{|p_{i,j-1} - 2p_{i,j} + p_{i,j+1}|\}}.
\]

Note that the denominator is a constant and the numerator can be rearranged as

\[
p_{i,j-1} - 2p_{i,j} + p_{i,j+1} = (p_{i,j-1} - p_{i,j}) + (p_{i,j+1} - p_{i,j}).
\]
That means while the $j$-th vertex is pushed toward the midpoint of two neighboring vertices along the curve, the value $E_{\text{curv}}$ is minimum, i.e., the shape of the curve will be maintained as $C^2$ continuity.

The curve-controlling factor $\beta_c$ controls the shape of the curves in each frame. If $\beta_c = \text{constant}$, the factor will generate a kind of force to maintain $C^2$ continuity of curves; if $\beta_c = g(c)$, the factor will generate a kind of force to control the shape of curves.

The new method is not only vertex-oriented but also curve-oriented. It can easily generate linear or non-linear interpolations and maintain the continuity of the curve itself as well. You can generate different kinds of animation by using the same energy-minimization function. It offers great flexibility in inter-frame interpolations.
APPENDIX C

A HYBRID TWO-PHASE ALGORITHM FOR FACE RECOGNITION

Scientists developed many classifiers in pattern recognition field, while applying single technique is very difficult to achieve high recognition rate for very large database. Decomposing the problem into two parts and solving each of them individually make it easier. In face databases, group the images of same person as one class, while they are different from poses or lightening conditions aspects. As know, in general, all faces are stored as gray level view-based images, which is in high dimensional space. Before apply classification, need to reduce the dimensionality of the original images first. As know, \textit{PCA} (Principal Component Analysis) method is a successful and commonly used dimensionality reduction technique in computer vision. It can also be a classification method, but because it does not consider the difference among classes, it is not very good in classification sense. \textit{LDA} (Linear Discriminate Analysis) overcame the drawback of PCA and became a very graceful classification method. You can use it in the first place. Misclassification still exists, but it can restrict our further processing into a limited size database.

Traditional \textit{SVM} (Support Vector Machine) are formulated to solve a classical two-class pattern recognition problem. Many researchers adopted the traditional two-class SVM to multi-class SVM. In the second step of the restricted problem, choose the tree-based SVM to perform the refinery classification to achieve high recognition rate.

For face recognition, FERET database is used in our experiments. The training set of face images includes 100 persons, each having 9 poses: $\pm 60^\circ$, $\pm 40^\circ$, $\pm 25^\circ$, $\pm 15^\circ$ and $0^\circ$. The testing set excludes $\pm 60^\circ$ because they are the fringe poses in the training set.
and can not be approximated by two consecutive pose angles. Note that when the test image of a pose is probed, the eigenvectors associated with that pose will be removed from the eigenspaces.

Geometric normalization is performed by cropping the face images in dimensions of 32 by 32 to include eye brows, eyes, nose and mouth. Shih and Kai apply the conventional PCA to select the closest $K$ persons as well as the closest 3 poses to create the $3K$ candidates for further refinement. Then, the matrix-based PCA is performed on these $3K$ candidates. Finally this feature space of the testing image is compared with all the feature spaces of the $3K$ images to obtain the matching by using the nearest neighbor matching. Here L1 is used since it performs extraordinary well in the matching process in high dimensional space according to the empirical results. The L1 distance between two matrices, $A$ and $B$, is computed as

$$d = \sum_i \sum_j (a_{ij} - b_{ij}) ,$$

where $A = \{a_i\}$ and $B = \{b_i\}$.

Our experimental results show that two-phase pattern recognition using PCA/SVM can achieve a 100% recognition rate in both face recognition and optical character recognition. It is also efficient. With a little more time trading off the accuracy, the two-phase recognition scheme can be applied to real-time applications. Since SVM is not sensitive to the input space which has already left out the problem of the dimension of feature space, the key to the high recognition rate therefore is the fact that the inclusion of the corrected candidate as the output of the first phase task.
For completing a two-phase pattern recognition task, we have already shown how PCA and SVM are coordinated as well as in what condition a high recognition rate can be guaranteed. Since SVM has promising generalization ability, our logical future direction will be probing the tradeoff between PCA and ICA (Independent Component Analysis) in terms of the feature extraction ability.
APPENDIX D

SUPPORT VECTOR MACHINE NETWORKS FOR MULTI-CLASS CLASSIFICATION

The support vector machine (SVM) has recently attracted growing interests in pattern classification due to its competitive performance. It is originally designed for two-class classification, and many researchers have been working on extensions to multi-class. In this paper, present a new framework that adapts the SVM with neural networks and analyze the source of misclassification in guiding our pre-processing for optimization in multi-class classification. Shih and Kai perform experiments on the ORL database and the results show that our framework can achieve high recognition rates.

The decision function of SVM provides not only the class label but also useful information, such as class density and distribution knowledge.

\[ f_d(\bar{x}) = \bar{w}_0 \cdot \bar{x} + b_0, \]

where \( f_d(\bar{x}) \) represents the distance between the sample \( \bar{x} \) and the optimal hyper-plane \( \bar{w}_0 \cdot \bar{x} + b_0 = 0 \). As shown in Fig. D.1, the distances \( f_{d12}^2, f_{d13}^2 \) and \( f_{d23}^2 \) compose a vector \( \bar{f}_d^2 = (f_{d12}^2, f_{d13}^2, f_{d23}^2)^\top \) to identify the position of class \( C_2 \), and \( y_j \) denotes the hyper-plane between classes \( i \) and \( j \). Therefore, can use \( \bar{f}_d^i \) to illustrate the distribution of class \( i \).
The author extends this idea to a layered neural network to construct a new framework as shown in Figure D.2. For the $k$-class problem, create $k$ hidden neurons, each of which is a one-against-all SVM having the decision function $f_d(\mathbf{x}) = \mathbf{w}_0 \cdot \mathbf{x} + b_0$, and $k$ output nodes, which compose an output vector $\mathbf{Y} = (y_1, y_2, ..., y_k)^T$. 

**Figure D.1** Decision function of a 3-class SVM.
For a training sample $\bar{x}$ from class $i$, it serves all the SVMs. You can obtain a new distance vector $\bar{f}_d(\bar{x}) = (f_d^1, f_d^2, ..., f_d^k)^T \in R^k$ to represent the sample $\bar{x}$ in $R^k$ space. Finally, the desired output vector $\bar{Y}(\bar{x}) = (y_1, y_2, ..., y_k)^T = \bar{f}_d(\bar{x}) \cdot W$, where $W$ is a $k \times k$ weight matrix, and $y_i = 1$ and $y_{j \neq i} = 0$. Because $\bar{Y}(\bar{x})$ and $\bar{f}_d(\bar{x})$ are known, the weight matrix can be obtained by

$$W = F_d^* Y = (F_d^T F_d)^{-1} F_d^T Y,$$

where $F_d$ and $Y$ are derived by all training samples, and $F_d^*$ denotes the pseudo-inverse of $F_d$. Besides, can imagine that for a $k$-class problem, the whole space is divided by $k$ hyper-planes through such a framework as shown in Figure D.3.

![Figure D.3 Hyper-planes divide the space.](image)

Traditional transformation maps the original image vectors to another high dimensional space associated with a fixed origin point. But our proposed framework maps them to a moving coordinate system whose degree is the number of classes. There are three
phases in our framework. The first is the constructing phase, the second is the training phase, and the third is the testing phase. The constructing phase is to build all SVMs. The dimensionality of all samples is reduced from $\mathbb{R}^m \rightarrow \mathbb{R}^n$ through the PCA procedure. The $\tilde{x} \in \mathbb{R}^n$ will serve as the input vector into our framework. Supposedly, there are $l$ samples for each class in the training set. For the $i$-th SVM $M_i^1$, select all training samples in class $i$, and any one sample from all other classes $j \neq i$ to obtain the training set $T_i^1$. After $M_i^1$ is trained, test the complete training set. There exist some misclassified samples. The author adds them to $T_i^1$ with negative labels as a new training set $T_i^2$, so can obtain a new trained SVM called $M_i$. If no misclassified samples are found, then $M_i^1 = M_i$. Through such a procedure, $M_i$ can enhance its generalization. The SVM is trained by all training samples just once. After all of $M_i$’s are established, the target outputs will be created. For a training sample $\tilde{x}$ from the class $i$, the desired output vector is $\bar{Y}(\tilde{x}) = (y_1, y_2, \ldots, y_k)^T$, where $y_i = 1$ and $y_{j \neq i} = 0$. The overall procedures are given in Fig. 5.

![Figure D.4 The overall procedures.](image)

The objective of the training phase is to compute the weight matrix. For a complete training set, the system generates $F_d W = Y$, where $F_d = (\tilde{f}_d^{1}, \tilde{f}_d^{2}, \ldots, \tilde{f}_d^{k_l})$ and $Y = [\bar{Y}(\tilde{x}_{i,1}), \bar{Y}(\tilde{x}_{i,2}), \ldots, \bar{Y}(\tilde{x}_{i,k})]$. The output vector only contains a single “1” in the corresponding output node to identify the desired class number. You have $\bar{Y}(\tilde{x}_{i,m}) = (y_1, y_2, \ldots, y_k)$, where $y_i = 1$ and $y_{j \neq i} = 0$. It expects that the output vector only
contains a single “1” in the corresponding output node to identify the desired class. Once $F_d$ and $Y$ are known, the weight matrix can be obtained by eq. (12). Finally, the whole system is established.

In the testing phase, for an unknown sample $\tilde{x}$, the system outputs a vector $\tilde{Y}(\tilde{x})$, where the location of the maximum of output nodes, and then indicates the recognized class. In other words, it is the most likely class.

$$\text{Class of } \tilde{x} = \arg \max_{i=1,2,...,A} y_i(\tilde{x}) .$$
REFERENCES


