Weighted Meta Analytic Procedure for Pooling High Throughput Biological Data

Xiaodong Zhou

Follow this and additional works at: https://digitalcommons.memphis.edu/etd

Recommended Citation
Zhou, Xiaodong, "Weighted Meta Analytic Procedure for Pooling High Throughput Biological Data" (2011). Electronic Theses and Dissertations. 393.
https://digitalcommons.memphis.edu/etd/393

This Dissertation is brought to you for free and open access by University of Memphis Digital Commons. It has been accepted for inclusion in Electronic Theses and Dissertations by an authorized administrator of University of Memphis Digital Commons. For more information, please contact khggerty@memphis.edu.
WEIGHTED META ANALYTIC PROCEDURE
FOR POOLING HIGH THROUGHPUT BIOLOGICAL DATA

by

Xiaodong Zhou

A Dissertation
Submitted in Partial Fulfillment of the
Requirements for the Degree of
Doctor of Philosophy

Major: Computer Science

The University of Memphis
December 2011
DEDICATION

This dissertation is dedicated to

the memory of my mother

Qianguí Wang
ACKNOWLEDGEMENTS

I would like to thank sincerely my advisor, Dr. E. Olusegun George for his guidance, understanding, and patience throughout the time it took me to complete this research and write the dissertation at University of Memphis. His mentorship was paramount in providing a well rounded experience consistent with my long-term career goals. He has enlightened me through his wide knowledge of statistics and his deep intuition about where it should go and what is necessary to get there. For everything you have done for me, Dr. George, I thank you.

Dr. Ramin Homayouni, Dr. Lan Wang, and Dr. Vinhthuy Phan deserve special thanks as my dissertation committee members. In particular, I would like to thank Dr. Ramin Homayouni for inspiring me on chapter 3 and kindly providing human gene-to-gene function similarity matrix. Many thanks to Dr. Lan Wang, and Dr. Vinhthuy Phan for their encouraging words, thoughtful criticism, and time and attention during busy semesters. Dr. Wang also revised chapter 4 of my dissertation.

I am grateful to my mentor at work, Dr. Robert Williams in Department of Anatomy and Neurobiology, University of Tennessee Heath Science Center for his constant encouragement of my pursuit of a Ph.D. degree. In addition to his mental support, Dr. Williams also generously provided all computation facilities needed for my dissertation work.

My thanks must go also to my colleague, Dr. Xusheng Wang, for sharing his enthusiasm for and comments on my work. My colleague, Lei Yan, set up the computation facilities for me. His help on my work is highly appreciated.
I owe my deepest gratitude to my parents for their endless love and support throughout my life. I am indebted to my mother, QianGui Wang, for her care and sacrifice. I have no suitable words to fully describe her everlasting love to me. Although she is no longer with us, she is forever remembered. I am sure she shares our joy and happiness in heaven. I thank my father, DianChuan Zhou, for his unwavering faith in me and encouragement to tackle challenges head on.

Finally, and most importantly, I would like to thank my wife Lijing Xu for her love and sacrifice. Without her encouragement and support, I would have given up long time ago under the pressure of taking full-time work and part-time study simultaneously. During the time of my research and writing the dissertation, despite her own busy work, Lijing took care of our home and our children so that I could spend all available time on my dissertation. This dissertation would simply be impossible without her. I cannot thank her enough. I also credit our children, Amanda Zhou and Andrew Zhou, for inspiring and amazing me every day.
ABSTRACT


High throughput technologies such as DNA microarray have been widely used to simultaneously measure the expressions of thousands of genes. Analysis of such data often yields to high false positive and false negative rates. When multiple datasets addressing the same scientific question are available, it is clear that integrating them can be more informative, with superior operating characteristics, than any individual study. Pooling the P-values of the statistical tests from the individual studies presents a feasible and proven solution for integrating heterogeneous datasets. However, pooling these P-values with equal weights may result in suboptimal statistical power because of the unequal sample size and experiment quality of the datasets. An approach which weighs dataset according to specified criteria would seem more efficacious.

In this dissertation, we developed a procedure for optimally pooling P-values of independent tests from several studies. We propose an approximation of the null distributions of weighted versions of three popular pooling statistics: the Fisher’s omnibus method, the Logit method and Z method. We use approximate null distribution to directly estimate the P-values of the weighted combination statistics, and compare our procedure with an existing procedure called Pointillist, in which the null distribution of the weighted combination procedure is simulated. We have found that the Pointillist software has several errors. We demonstrate the superiority of our procedure over the Pointillist algorithm by application to set of six experimental datasets. In addition, to pool evidence of gene expression data, one-sided P-value instead of two-sided P-values should
be used to avoid losing information. We construct an optimally weighted pooling procedure for pooling one-sided P-values.

Since the datasets to be pooled are those of gene expressions, a biological perspective can be used to assess the performance of the pooling methods. We developed optimally weighted combination procedure to pool gene expression data by maximizing functional coherence of top ranked genes. Testing with the sample datasets, the top ranked genes identified by this method has higher functional coherence than any single dataset.

We have developed a web tool for implementing the optimally weighted combination procedures proposed in this dissertation.
# TABLE OF CONTENTS

## LIST OF TABLES

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x</td>
</tr>
</tbody>
</table>

## LIST OF FIGURES

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>xii</td>
</tr>
</tbody>
</table>

## CHAPTER 1 INTRODUCTION

1.1 Gene expression  
1.2 DNA microarray  
1.3 Multiple hypothesis testing  
1.4 Meta-analysis of high throughput experiment data  
1.5 Pooling P-values from each dataset  
1.6 Equal weight pooling category  
1.6.1 Inverse $\chi^2$ method  
1.6.2 Z method  
1.6.3 Logit method  
1.7 Weighted pooling category  
1.7.1 Weighted inverse $\chi^2$ method  
1.7.2 Weighted Z method  
1.7.3 Weighted Logit method  
1.8 Weight vector optimization by Enhanced Simulated Annealing (ESA) method  
1.9 Objective  
1.10 Overview  

## CHAPTER 2 OPTIMALLY WEIGHTED COMBINATION PROCEDURES TO POOL GENE EXPRESSION DATA BY MAXIMIZING NUMBER OF SIGNIFICANT GENES

2.1 Introduction  
2.2 Experiment data  
2.2.1 The datasets  
2.2.2 Data preprocessing  
2.2.3 Reduction of probeset/spot level data to gene level data  
2.2.4 Generation of two-sided P-value for each gene  
2.3 Statistical errors in Pointillist and proposed corrections  
2.3.1 Introduction to Pointillist algorithm  

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>10</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>12</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>13</td>
</tr>
<tr>
<td></td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>17</td>
</tr>
<tr>
<td></td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>22</td>
</tr>
<tr>
<td></td>
<td>23</td>
</tr>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>24</td>
</tr>
<tr>
<td></td>
<td>26</td>
</tr>
<tr>
<td></td>
<td>26</td>
</tr>
</tbody>
</table>
2.3.2 Fixed number of simulated null hypothesis data points in Pointillist 29
2.3.3 Pointillist’s simulation of the null distribution 32
2.3.4 Pointillist’s fitted distribution 35

2.4 Estimate P-value of weight pooled statistics directly by its theoretical null hypothesis distribution 41
   2.4.1 Approximation of theoretical null hypothesis distribution 42
   2.4.2 Result of pooling two-sided P-values using approximated null distributions 47

2.5 one-sided P-values should be used to pool gene expression data 48
   2.5.1 Generatin of one-sided P-values for each gene 49
   2.5.2 Comparison of pooling two-sided P-values with pooling one-sided P-values 49

2.6 Procedure to pool gene expression data by one-sided P-values 53
   2.6.1 Pooling result with \( \alpha_{\text{up-regulation}} = \alpha_{\text{down-regulation}} \leq 0.05 \) 55
   2.6.2 Pooling result with \( \text{FWER}_{\text{up-regulation}} = \text{FWER}_{\text{down-regulation}} \leq 0.01 \) 58
   2.6.3 Pooling result with \( \text{FDR}_{\text{up-regulation}} = \text{FDR}_{\text{down-regulation}} \leq 0.01 \) 60

2.7 Conclusion and discussion 66

CHAPTER 3 OPTIMALLY WEIGHTED COMBINATION PROCEDURE TO POOL GENE EXPRESSION DATA BY MAXIMIZING FUNCTIONAL COHERENCE OF TOP RANKED GENES 68

3.1 Introduction 68
3.2 Method 75
3.3 Result 78
   3.3.1 Optimization result by maximizing functional coherence of top N genes 78
   3.3.2 Comparison of gene sets obtained by the two optimization procedures 80

3.4 Conclusion and discussion 82

CHAPTER 4 WEB TOOL 84

4.1 Introduction 84
4.2 User interface 84
   4.2.1 Input data type 84
   4.2.2 Input data format 84
   4.2.3 Interface for optimization by significant genes count 85
   4.2.4 Interface for optimization by gene set functional coherence 85

4.3 Architecture 86
4.4 Performance 86
4.5 License and availability 86
CHAPTER 5 CONCLUSIONS AND FUTURE WORK 92
REFERENCES 94
# LIST OF TABLES

<table>
<thead>
<tr>
<th>TABLE</th>
<th>PAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Microarray platform(s) and tissue samples in each dataset</td>
<td>26</td>
</tr>
<tr>
<td>2. Number of gene in each dataset</td>
<td>26</td>
</tr>
<tr>
<td>3. Original Pointillist result on different significance level</td>
<td>30</td>
</tr>
<tr>
<td>4. Pointillist’s empirical cumulative distribution function</td>
<td>31</td>
</tr>
<tr>
<td>5. Comparison of number of significant genes detected by different distributions</td>
<td>38</td>
</tr>
<tr>
<td>6. Corrected Pointillist’s performance and occupied memory at different significance levels</td>
<td>41</td>
</tr>
<tr>
<td>7. Comparison of result from exact null distribution with result from theoretical distribution</td>
<td>47</td>
</tr>
<tr>
<td>8. Comparison of pooling one-sided P-values with pooling two-sided P-values on dataset level</td>
<td>51</td>
</tr>
<tr>
<td>9. Comparison of one-sided-significant-genes and two-sided-significant-genes</td>
<td>52</td>
</tr>
<tr>
<td>10. In each dataset, count of genes whose one-sided P-values are less than 0.05</td>
<td>55</td>
</tr>
<tr>
<td>11. Result of pooling one-sided P-values with $\alpha_{\text{up-regulation}} = \alpha_{\text{down-regulation}} = 0.05$</td>
<td>57</td>
</tr>
<tr>
<td>12. In each dataset, the count of genes whose one-sided P-values are less than 4.88234E-7</td>
<td>58</td>
</tr>
<tr>
<td>13. Result of pooling one-sided P-values with threshold $\text{FWER}<em>{\text{up-regulation}} = \text{FWER}</em>{\text{down-regulation}} = 0.01$</td>
<td>59</td>
</tr>
<tr>
<td>14. Result of pooling one-sided P-values with equal weights when FDR = 0.01</td>
<td>61</td>
</tr>
<tr>
<td>15. $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$ and weight vector required to get optimal weighted procedure at FDR &lt; 0.01 level</td>
<td>65</td>
</tr>
</tbody>
</table>
16. Comparison of the weighted combination methods by gene counts when FDR_{up-regulation} = FDR_{down-regulation} \leq 0.01

17. Number of significant connections within top N genes in each dataset

18. Weight assigned to each dataset and count of significant connections in top N genes

19. Count of significant genes in the results generated by two optimization methods

20. Count of significant connections within top N genes in the results generated by two optimization methods
LIST OF FIGURES

FIGURE ........................................................................................................... PAGE

1 A contemporary view of gene expression. Courtesy of Orphanides and Reiberg (Orphanides and Reinberg, 2002) ........................................................................ 3
2 A schema of running microarray experiment .................................................................................. 6
3 Approach to pool evidence from multiple studies ........................................................................... 9
4 Process to pool evidence along with weight optimizer ..................................................................... 14
5 Process of Simulated Annealing to reach global optima ................................................................. 15
6 Schema of counting significant genes in one state of optimization proce ........................................ 20
7 Flow chart of Pointillist algorithm to count significant genes in one state of optimization process ................................................................. 27
8 Pointillist process to estimate significance of the optimally pooled P-values ................................. 28
9 Comparison of number of significant genes detected at various α values by different distributions for Logit method ................................................................. 39
10 Procedure to count significant genes based on raw two-sided P-values by theoretical distribution .......................................................................................................................... 42
11 Procedure to count significant genes based on raw one-sided P-values by theoretical distribution ................................................................................................. 54
12 One-sided P-value thresholds converging process ........................................................................ 63
13 The work flow of calculating the similarity between documents with LSI model .......................... 73
14 Procedure to evaluate functional coherence of one state .............................................................. 77
15 Web tool main page .......................................................................................................................... 87
16 Types of input P-values .................................................................................................................. 87
17 Input file format. ........................................................................................................... 88
18 Input interface for the option of optimization by count of significant genes............ 88
19 Output interface for the option of optimization by count of significant genes......... 89
20 Text format pooling result of all genes for the option of optimization by count of significant genes........................................................................................................... 89
21 Input interface for the option of optimization by gene set functional coherence. ...... 90
22 Output interface for the option of optimization by gene set functional coherence..... 90
23 Significant connections within top ranked genes for the option of optimization by gene set functional coherence........................................................................................................... 91
24 Software architecture. ...................................................................................................... 91
CHAPTER 1 INTRODUCTION

1.1 Gene expression

Deoxyribonucleic acid (DNA) exists in all known organisms, except in ribonucleic acid (RNA) viruses, to store genetic information for long term. A DNA molecule consists of two long reverse complimentary strands. The unit of DNA strand is nucleotide. Each nucleotide consists of a ribose sugar and phosphate, which represent the strand’s backbone, and one of four different nitrogenous base units, adenine (A), guanine (G), thymine (T) and cytosine (C). The sequence of nucleotides, referred to as a DNA sequence, stores genetic information which is essential for constructing the other cell components, such as RNA and protein. Practically all the cells in the same organism have the same DNA. A gene is DNA segments that code for one specific type of protein or RNA.

RNA is similar to DNA except the thymine (T) bases are replaced by uracil (U) bases. However, most RNA molecules are single-stranded. There are three main types of RNA: messenger RNA (mRNA), transfer RNA (tRNA) and ribosomal RNA (rRNA). mRNA is synthesized from a gene segment of DNA, and carries the information of the amino acids sequence in a protein to ribosomes. tRNA is responsible for carrying amino acids to the ribosome where rRNA assembles amino acids to protein by decoding mRNA.

Proteins are macromolecules made of amino acids which are assembled in linear polypeptide chains. The sequence of amino acids in protein is defined by genes. Proteins are involved in virtually all cell functions. They act as enzymes to catalyze chemical reactions. Proteins also play important roles in the processing of cell signaling, ligand
transportation. Structural proteins are responsible for providing stiffness and rigidity to cell components, and generating mechanical forces for movement.

A gene expression is the process of synthesizing a biologically functional molecule of gene product, either protein or functional RNA, such as tRNA and rRNA. Most genes synthesize proteins. These genes are called protein-coding genes. The few genes whose products are functional RNA are known as RNA-coding genes. There are two major steps by which genes are expressed. The first step, called transcription, is the process by which RNA is synthesized from DNA. For RNA-coding genes, this step produces their final products (e.g., tRNA, rRNA). For a protein-coding gene, the transcription results in mRNA, the intermediate product which carries coding information from the gene to a protein. In a second major step called translation, the coding information on mRNA is decoded by ribosomes to synthesize protein. A gene expression can be regulated at different levels: transcription, post-transcriptional modification, translation, and post-translational modification. A contemporary view of eukaryotic gene expression is shown in Figure 1, which gives simplified schematics for all the steps involved in a gene expression.
Figure 1. A contemporary view of gene expression. Courtesy of Orphanides and Reinberg (Orphanides & Reinberg, 2002)

A gene expression is affected by factors such as tissue type, genetic background, the development stage of the organism, and the physiological state of the cell. Since changes in the physiology of an organism are precipitated by changes in gene expressions, the analysis of gene expressions is an important tool for gaining insights into the complex functioning of the cell. Virtually all major differences in cell state or type are related to changes in the mRNA levels of many genes.
1.2 DNA microarray

Gene expression is the quantity by which a genetic code is interpreted. Ideally, a gene expression is measured by the final gene product (e.g., protein, functional RNA), but technically it is easier to measure mRNA, and use it to infer the gene expression level. Traditional molecular biological methods, such as Northern blotting and Reverse Transcription quantitative Polymerase Chain Reaction (RT-qPCR), can measure only the mRNA of few genes at once. With such low-throughput approaches, it is hard to study complex biological processes in which many genes work in concert, and comprehend the “whole picture” of genes interactions. In recent years, several high-throughput techniques have been developed to measure the mRNA levels of thousands of genes simultaneously, so as to create a global picture of gene expression. These techniques include Serial Analysis of Gene Expression (SAGE), DNA microarray, and next generation sequencing. These techniques have been used on large scale to study the effects of treatments, identify genes involved in complex diseases, study the interactions between genes, identify biological pathway/network, study gene regulation at different developmental stages, and predict patient prognosis. Among all the techniques for measuring gene expression, DNA microarray is the most widely used technique.

A DNA microarray is a collection of probes on a solid support. Each probe is specific to one particular gene. The major steps involved in running a DNA microarray experiment are to isolate mRNA from a sample, reverse transcribe the mRNA into cDNA (or cRNA), label cDNA (or cRNA) with fluorescent dyes, hybridize labeled cDNA (or cRNA) to microarray, wash the microarray, and scan it with laser scanner.
There are essentially two types of microarrays: the two-color cDNA microarray and the one-color oligonucleotides microarray. For the two-color microarray, the probes are usually Expressed Sequence Tags (ESTs) from a cDNA library representing a subset of genes. To run experiment with two-color microarray, cDNA from two samples are prepared and labeled with two different dyes, typically Cy3 (green color) and Cy5 (red color). The two differently labeled cDNA are then mixed and hybridized to the same microarray. The ratio of intensities of each dye reflects the ratio of mRNA levels in the two samples. One-color microarray only measures the mRNA of one sample. In comparison with the one-color microarray, the two-color microarray is less specific, and expression measurements based on this platform are more noisy and less reproducible (Lee et al., 2005). A schema of running microarray experiment is shown in Figure 2.

Microarray experiments generate enormous amount of data, producing a mix of both information and noise. For example, the Affymetrix Human Genome U133 2.0 plus array contains almost 45,000 probe sets representing approximately 20,000 well-substantiated human genes. Thus, a single experiment with the U133 chip will generate 45,000 data points. In a typical data analysis of one microarray experiment, each probe is analyzed separately to identify genes differentially expressed under different conditions. Further down-stream analyses then are performed with the selected genes. These analyses are usually performed on each gene. This results in multiple hypothesis testing.
1.3 Multiple hypothesis testing

High-through techniques such as microarray involve the simultaneous measurements of the expression of thousands of genes. Performing hypothesis testing on each gene results in a multiplicity of tests, with the undesirable consequence of significant increase in the chance of falsely rejecting some of the null hypotheses of no gene differential expression. The challenge of adjusting for the resulting inflation in the overall Type I error rate, or alternatively, controlling the false discovery rates has been a focus of considerable statistical research in the last 10 years.

*Figure 2.* A schema of running microarray experiment
Commonly, the overall type I error rate is defined as the probability of at least one false positive declaration of a gene to be differentially expressed, under the condition that none of the genes is differentially expressed. This error rate is referred to as the family wise error rate (FWER). Several procedures have been proposed to control FWER (Hochberg, 1988; Holm, 1979; Sidak, 1967; Westfall & Young 1993). The Bonferroni correction, which assigns a level $\alpha/n$ to each of $n$ tests in order to guarantee a size $\alpha$ overall is the simplest and most conservative approach to control FWER.

An alternative approach is to control the expected proportion of falsely rejected hypothesis, or false discovery rate (FDR). Benjamini and Hochberg (1995) proposed to control the FDR by a step-down procedure. This procedure proceeds as follows. Suppose that $m$ null hypotheses $H_1,\ldots,H_m$ are tested and $q^*$ is the target FDR. Let $P_{(1)} \leq P_{(2)} \leq \ldots \leq P_{(m)}$ are the P-values of the tests arranged in order of magnitude. Let $P_{(i)}$ denote the P-value corresponding to the test of hypothesis $H_{(i)}$ ($1 \leq i \leq m$). Find the largest $i$ (let it be $k$) for which $P_{(i)} \leq \left(\frac{i}{m}\right) q^*$ and reject all $H_{(i)}$ where $i = 1, 2, \ldots, k$.

In this dissertation, we will use Bonferroni’s procedure to control FWER and Benjamini’s step-down procedure to control FDR. Other procedures can be easily adapted for use with the meta-analytic method proposed in this dissertation.

1.4 Meta-analysis of high throughput experiment data

It has increasingly become common to find several research centers performing microarray experiments on the same genomic study. Typically, such centers do not always use the same microarray platform. However, it is clear that integrating the multiple datasets from these sources could be more informative, and have superior
operating characteristics than any individual study. Such a collection of datasets can be pooled using meta-analysis. Meta analysis is a statistical technique for combining results of related but independent studies (Normand, 1999). It has been shown that meta-analysis not only improve reproducibility, hence improve reliability of analyses (Hong et al., 2006), but also enhance the statistical power for detecting small but consistent effects that might be classified as false negatives in individual studies (Choi, 2003). In the context of pooling microarray datasets, meta-analysis has become as a valuable tool pooling pertinent information from multiple experiments and performing pathway analysis.

1.5 Pooling P-values from each dataset

Although it is very common for different research centers to perform experimental studies on same or highly related hypothesis, it is usually quite difficult to integrate the dataset from these studies. This is because the procedures used to generate these data sets are usually heterogeneous. The types of data points between studies might range from ratio to absolute value, and the platform on which they are generated might be incompatible. Even when datasets are generated by the same technique or same platform, it is very rare that they can be combined at raw level. Complicated experimental variables embedded in array experiments usually make direct comparison among multiple microarray datasets impossible (Kuo et al., 2002; Irizarry et al., 2005). Pooling P-values from the various studies presents a feasible solution for integrating statistical inference from each individual dataset. A schema of this approach is illustrated in Figure 3.
In the field of bioinformatics, several researchers have implemented pooling of P-values in the analysis of genomic datasets from multiple independent sources. Bailey and Gribskov (1998) applied a variant of Fisher’s inverse $\chi^2$ method to solve the sequence homology search problem. They also used the method for protein classification and scoring motifs (Bailey & Grundy, 1999). Oliva et al. (2005) used a method based on Z-transform method to pool P-values from three microarray datasets of the fission yeast Schizosaccharomyces pombe.

In this dissertation, we introduce optimally weighted versions of several commonly used pooling methods that combine P-values of independent datasets. These methods fall into two categories according to the weights assigned to each dataset.
1.6 Equal weight pooling category.

Let $P_1, P_2, \ldots, P_k$ be the P-values from K independent datasets. Let $H_0, \ldots H_{0k}$ be the corresponding null hypotheses. For the methods in this category, each sample is assigned the same weight. In other words, each dataset is treated as equally contributing to the pooled evidence. For instance, if there are N datasets, we can assign 1 to each dataset or $1/N$ to each dataset. The P-values from each dataset cannot simply be averaged. They are usually transformed by some function $F$ and summed of to get the pooled statistic $T=\sum_{i=1}^{K} F(P_i)$. The P-value of the combined statistics can be calculated using the distribution of $T$ under the overall null hypothesis. There are three commonly used methods: (1) Fisher’s inverse $\chi^2$ method (1932), (2) Stouffer’s Z method (1949), (3) George and Mudholkar’s Logit method (1983).

1.6.1 Inverse $\chi^2$ method

The Fisher procedure for pooling the P-values is performed by computing the statistic

$$F = \sum_{i=1}^{K} -2\ln(P_i)$$

Under the null hypothesis $H_{0i}$, $P_i$ has a uniform distribution in the range of [0, 1] if the component test statistics are continuous, $i=1,\ldots,k$. Consequently, $-2\ln(P_i)$ follows a $\chi^2$ distribution with two degrees of freedom, and from the additivity of $\chi^2$ distribution, $F$ follows a $\chi^2$ distribution with $2K$ degrees of freedom. Thus the P-value associated with the F-score is given by $P_i=P(X \geq F)$, where $X$ is a $\chi^2$-score with $2K$ degrees of freedom.
1.6.2 Z method

The Z-transform procedure (Stouffer et al., 1949) is performed by computing the value

\[ Z = \sum_{i=1}^{K} \left( \frac{Z_i}{\sqrt{R}} \right), \]

where

\[ Z_i = \Phi^{-1}(1 - p_i). \]

Here \( \Phi \) is the standard cumulative normal distribution function and \( Z \) has a standard normal distribution under null hypothesis.

1.6.3 Logit method

The Logit procedure (George & Mudholkar, 1983) is performed by computing the statistic

\[ L = -\sum_{i=1}^{K} \ln \left( \frac{p_i}{1-p_i} \right). \]

Under the null hypothesis, \( L \) is a convolution of \( K \) logistic random variables. George and Mudholkar gave the exact distribution of \( L \) and proposed that it can be approximated by a \( t \) distribution with \( 5K + 4 \) degrees of freedom. That is,

\[ L \sim \pi \sqrt{\frac{K(5K+2)}{3(5K+4)}} (t_{5K+4}). \]

The P-value associated with \( L \) score is given by
where $X$ is a t-score with $5K + 4$ degrees of freedom.

1.7 Weighted pooling category

For the equal weight methods in the last category, it is assumed that all tests to be combined or the associated P-values are equally important. However, each high throughput technology has its own systematic bias (Mrowka et al., 2001; von Mering & Bork, 2002) and data from different technologies have different degrees of reliability (Hwang et al., 2005). Even if the multiple datasets are generated by the same technology, their sample sizes, underlying error variances and sensitivities might be quite different. Under this consideration, a weighted combination statistic seems to be more desirable to get larger statistical power. Without loss of generality, the positive weights are assigned so that their sum is 1. The weights are expressed as a vector (called weight vector in this dissertation) in order of datasets to simplify further computation. For instance, if there are three datasets to be pooled, and the weights 0.5, 0.2, 0.3 are assigned to datasets 1, 2 and 3 respectively, the weight vector is [0.5, 0.2, 0.3]. The weighted versions of the three pooling statistics described in last section are presented below.

1.7.1 Weighted inverse $\chi^2$ method

The weighted Fisher’s inverse $\chi^2$ method is in the form of

$$F_W = \sum_{i=1}^{K} -2w_i \ln(p_i)$$
where \(w_i\) is the weight assigned to test \(i\).

Hou (2005) approximated the distribution of \(F_w\) by that of \(c \chi_f^2\), where

\[
c = \frac{\sum_{i=1}^{K} w_i^2}{\sum_{i=1}^{K} w_i^2},
\]

and \(\chi_f^2\) is a chi-square random variable with \(f\) degree of freedom where

\[
f = 2 \left( \frac{\sum_{i=1}^{K} w_i}{\sum_{i=1}^{K} w_i^2} \right)^2.
\]

1.7.2 Weighted Z method

The weighted Liptak-Stouffer’s Z-method is defined by the statistic

\[
Z_W = \frac{\sum_{i=1}^{K} w_i Z_i}{\sqrt{\sum_{i=1}^{K} w_i^2}}.
\]

where \(w_i\) is the weight assigned to test \(i\). The distribution of \(Z_W\) is standard normal distribution with mean 0 and standard deviation 1.

1.7.3 Weighted Logit method

The weighted Logit method is in the form of

\[
L_W = -\sum_{i=1}^{K} w_i \ln\left(\frac{P_i}{1-P_i}\right),
\]

where \(w_i\) is the weight assigned to test \(i\). However, the approximation of the distribution of \(L_W\) had not been derived before this study.

1.8 Weight vector optimization by Enhanced Simulated Annealing (ESA) method.

If there is only one overall null hypothesis to test, it is relatively easy to assign weight to individual studies. Whitlock (2005) suggested that the weight assigned to each
P-value should be proportional to the inverse of its error variance. But with microarray data, there are many more factors to consider. In addition to sample size, there are variations in experiment quality, technology difference—cDNA platform tends to generate noisy data, and oligonucleotide platform tends to be less sensitive, platform difference—even the platforms from the same manufacturer are evolving. Many of these factors are hard to be quantitatively analyzed. Thus, it is hard to empirically decide the weight vector. We will use optimization method rather than exhaustive numeration to get the optimal weights. The process for calculation the optimal weight is illustrated in Figure 4.

![Figure 4](image)

*Figure 4. Process to pool evidence along with weight optimizer*

Annealing is metallurgic term used to describe the heating of a material and gradually cooling it to remove structural defect. While hot, atoms are more mobile and become unstuck from their initial positions so as to have a chance to wander randomly through higher energy states. As the temperature cools, they settle into their lowest energy states (ideally lower than their initial one). Kirkpatrick et al. (1983) described simulated annealing method, an adaptation of the Metropolis-Hastings algorithm
(Metropolis et al., 1953) by analogy with this physical process. The method illustrated by Figure 5 can be summarized in the following steps:

1. Start from a state $A$, compute its energy $E_A$.
2. Pick a nearby state $B$ at random, compute its energy $E_B$ (typically as a small change from $E_A$).
3. If $E_B < E_A$, accept the new state, since it has lower energy (a desirable thing).
4. If $E_B > E_A$, accept the new (higher energy) state with probability $p = e^{-(E_B - E_A)/T}$.

This means that when the temperature is high, “uphill” moves are allowed with higher probability because they will save the method from becoming stuck at local optima. But as the temperature is lowered, more “downhill” moves will be made to force the system to settle into the lowest configuration in neighborhood.

![Figure 5. Process of Simulated Annealing to reach global optima](image)

Siarry et al. (1997) presented an Enhanced Simulated Annealing (ESA) method for functions of many continuous variables. The ESA method not only closely approximates
known global optima, but also significantly reduces the number of function evaluations. Since the purpose of our work is not to develop or improve an optimization algorithm, we will use ESA for optimization in this dissertation.

1.9 Objective

The primary objective of this dissertation is to develop optimally weighted combination procedures to pool high throughput gene expression data. This can be described in terms of three specific goals and summarized as follows:

The key question for pooling high throughput gene expression data along with weight optimizer is how to quantitatively evaluate the pooled evidence in one state which is determined by the input weight vector, so that the system can determine if the state (weight vector) should be accepted.

**Goal 1:** The pooled evidence in one state can be evaluated from statistical perspective. Since it is desirable to identify more statistically significantly differentially expressed genes when using the pooled datasets, the number of significant genes can be used to quantitatively evaluate the state. Thus our first goal is to develop optimally weighted combination procedure to pool gene expression data by maximizing number of significant genes.

**Goal 2:** Since the datasets to be pooled are those of gene expressions, the pooled evidence in one state can be evaluated from biological perspective. Then the top ranked genes identified by the optimal result should have the most biological relevance. For this purpose, we will use gene set functional coherence analysis as a robust method for evaluating the biological relevance of the given gene set. Hence, the second goal is to
develop optimally weighted combination procedure to pool gene expression data by maximizing functional coherence of top rank genes.

**Goal 3:** Our third goal is to develop a web tool for implementing the optimally weighted combination procedures proposed in this dissertation for the benefit of other scientific researchers.

1.10 Overview

The rest of the dissertation is organized as follows: Chapter 2 develops in detail the computation of optimal weights for pooling P-values of gene expressions from multiple datasets by maximizing number of significant genes. In section 2.1, we introduce the idea of evaluating each state in optimization process by number of significant genes, as initially proposed by Hwang et al. (2005). We also give a brief introduction to Pointillist, the software package to implement Hwang’s procedure using simulated null hypothesis data. In section 2.2 we introduce the datasets that will be used throughout this dissertation for illustration of our procedures. In section 2.3 we point out several serious statistical errors in Hwang’s Pointillist algorithm and describe correction to amend these errors. Other shortcomings of Pointillist’s procedure, such as computational inefficiency, are also described. In section 2.4, we describe our procedure for counting significant genes in each state: the P-value of the weighted pooled statistics is directly estimated by using its theoretical null distribution. We then describe how to approximate the distribution of weighted Fisher’s inverse $\chi^2$ statistics by gamma distribution, and the distribution of weighted Logit statistics by t-distribution. Finally, using two-sided raw P-values we
compare our procedure with our corrected version of Pointillist procedure to highlight the improvement on the enhanced computational efficiency and statistical power of our procedure relative to Pointillist method. In section 2.5 we use examples to show that it is necessary to pool one-sided $P$-value rather than two-sided $P$-value for gene expression studies. In section 2.6 we illustrate our procedure to pool one-sided $P$-values for gene expression data. In section 2.7 we conclude the chapter with a discussion.

In Chapter 3 we present the computation of optimally weights by maximizing gene set functional coherence. We show that the set of top rank genes obtained by the use of optimally pooled $P$-values has significantly higher functional coherence than the set of top rank genes from any single study or those obtained by using equal weight pooling.

In Chapter 4 we present a web tool that we have built for the implementation of the procedures developed in Chapters 2 and 3. The web tool is accessible to the general public. Chapter 5 discusses future research directions.
CHAPTER 2 OPTIMALLY WEIGHTED COMBINATION PROCEDURES TO POOL GENE EXPRESSION DATA BY MAXIMIZING NUMBER OF SIGNIFICANT GENES

2.1 Introduction

Hwang et al. (2005) proposed a procedure which uses the number of “significant genes” to find optimal weight vector for combining P-values in high throughput data analysis. Specifically, using simulated annealing for optimization, the number of “significant genes” can be used to evaluate each state, determined by its input weight vector. In this dissertation, “significant genes” are defined as genes for which the P-values of the weighted combination procedure is less than level of significance $\alpha$. A schema for evaluation of significant genes based this definition is illustrated in Figure 6. With the input weight vector and raw P-values from experiments, a combination procedure will be used to pool the weighted raw P-values for each gene. Then the P-values of the combination procedure are compared with the input significant level $\alpha$ (for instance, $\alpha < 0.05$) to determine the number of significant genes. At each state, negative number of significant genes will serve as “energy” of that state. The more significant genes of one state, the less the energy of the state. After optimization process, the optimal state should have maximal number of significant genes, and the corresponding weights will be optimal weight vector.
Hwang et al. (2005) proposed a pooling procedure based on this strategy and developed the software package Pointillist (v2.1.2) in Matlab. Pointillist uses simulation to estimate the null distribution of combination statistics in each state during optimization process and provides three options for pooling weighted P-values: Fisher’s weighted inverse $\chi^2$ method, Mudholkar-George’s Logit method and Liptak-Stouffer’s weighted $Z$-method. After the optimization process ends, Pointillist takes the optimal weight vector, fits the weighted pooled statistics to a gamma or a normal distribution, then uses the fitted distribution to evaluate the P-values of optimally pooled experiment data.
In this chapter we identify several statistical errors in Hwang’s Pointillist algorithm, and we propose corrections to these errors. However, since Pointillist’s procedure is based on use of a simulated null distribution, even after our correction, it still has inherent shortcomings such as computational inefficiency and lack of precision at low P-values. We provide an alternative procedure in which we directly estimate the P-value of each weighted pooling statistics by using the exact theoretical distribution. In addition, in Hwang’s procedure, two-sided P-values of gene regulation are pooled with the result that information is lost. We propose that only one-sided P-values should be pooled to keep the complete information of gene regulation. We then illustrate our optimally weighted pooling procedure to pool one-sided P-values for gene expression data.

The rest of the chapter is organized as follows: in section 2.2 we describe the experimental datasets and their processing. In section 2.3 we identify severe errors in Pointillist and describe our correction of them. We then discuss inherent shortcomings of the Pointillist’s procedure. In section 2.4, we approximate the distribution of weighted Fisher’s inverse \( \chi^2 \) statistics by gamma distribution, and weighted Logit statistics by t-distribution. Finally, for comparison, proposes with Pointillist, we completed our procedure using two-sided P-values. In section 2.5 we argue that one-sided P-value rather than two-sided P-value should be used to pool gene expression datasets, and in section 2.6 we illustrate our procedure in this context. In section 2.7 we conclude the chapter with a discussion.
2.2 Experiment data

2.2.1 The datasets

In recent years, several research groups have independently used the gene expression data from microarray to study the biology of prostate cancer. These multiple datasets can be pooled using meta-analysis to obtain statistically more powerful information about the differential gene expression profile of primary prostate tumor in comparison with normal prostate tissue, than could be obtained from each individual research group. In this dissertation, we illustrate the power of such meta-analytic pooling datasets by using the research output from six groups: Chandran, 2005; Lapointe, 2004; Liu, 2006; Singh, 2002; Varambally, 2005; Welsh, 2001; and Yu, 2004. These research groups examined both primary prostate tumor samples and normal prostate tissue samples, and made the raw data of their microarray experiments available online. In this dissertation, the datasets from the groups of Lapointe, Liu, Singh, Varambally and Welsh are identified by their authors’ names. Chandran’s and Yu’s datasets are the subsets of the datasets published in NCBI Gene Expression Omnibus (GEO) series GSE6919. We took all primary prostate tumor and normal prostate tissue samples in this GEO series as our sixth dataset, and identify it by Monzon, the name of the contactor of this GEO series.

The tissue samples in these datasets were run on different microarray platforms. These include customized cDNA microarray made by Stanford University and commercial oligonucleotide platform made by the Affymetrix company, The probeset/spot numbers on each platform are quite different, varying from 12,558 to 54,613. In addition, the sample size of each dataset varies dramatically from 9 in Varambally’s dataset to 66 in Monzon’s dataset. Considering the variance in platform,
sample size and other factors, we assign different weights to the P-values of tests from different datasets rather than weighing each dataset equally. The platform and sample information of each dataset are summarized in Table 1.

2.2.2 Data preprocessing

For all the datasets which use Affymetrix oligonucleotide microarray except Welsh’s dataset, we downloaded the raw image files (CEL file) and re-analyzed them. For Welsh’s dataset, only text format gene expression values are available. Hence we used them directly in our work. We re-analyzed the downloaded Affymetrix oligonucleotide microarray image files with “affy” package in the Bioconductor (version 2.6) software run in R. We used the MAS5 algorithm to process all image files for consistency. For the Lapointe’s cDNA dataset, we downloaded the text files for each sample. The spot value is log (base 2) of the ratio of the median of channel 2 (usually 635 nm) to channel 1 (usually 532 nm) (Lapointe et al., 2004). Although the samples were run on 3 cDNA microarrays, because the difference of cDNA spots on them are small (less than 10%), they can be analyzed together (Lapointe et al., 2004). We essentially followed the normalization method described by Lapointe et al. (2004) and Perou et al. (2000). In brief, the spot values were mean centered by each microarray. On each array, there are approximately 1,000 redundant spots. We removed the redundancy by averaging the gene expression values of duplicate spots. We also removed the spots that have no associated GenBank ID annotation from further analysis. Each spot value was then mean centered across all microarrays of the same platform to remove platform bias. Finally, all remaining spots of each platform were joined together for further analysis (44,639 spots).
2.2.3 Reduction of probeset/spot level data to gene level data

To facilitate cross-platform data integration, we reduced the probeset / spot level data to gene level data. For all Affymetrix platforms, we used the annotation packages maintained by Biocore Data Team of R Bioconductor project to map the probesets to NCBI Entrez GeneIDs. For the custom cDNA platforms, in the companion platform annotation files, the spots are mapped to GenBank IDs. We utilized the “org.Hs.eg.db” package in Bioconductor to map GenBank IDs to GeneIDs.

On a microarray platform, there might be several probesets/spots mapped to the same GeneID. For Affymetrix platforms, we chose the probeset with highest mean value across all samples in one study to represent the gene. For cDNA platforms, we averaged the normalized data of the multiple spots of the same gene.

2.2.4 Generation of two-sided P-value for each gene

A student t-test was performed on each gene with “t.test” function in R (version 2.11.1) to identify differentially expressed genes in the prostate tumor group relative to the normal prostate tissue group. Although in this dissertation, we have recommended to use one-sided P-values to avoid losing information (see section 2.5 and 2.6), however since the Pointillist is coded for pooling two-sided P-values, we will input two-sided P-values to both the Pointillist and our procedure, in order to be able to make a direct comparison (see section 2.3 and 2.4). Two-sided P-values of each t-test statistics is obtained by using “two.sided” as value of the “alternative” parameter in the R “t.test” function.
Each dataset includes different number of genes because of the platform difference (see Table 2 for details). The union of genes from all datasets gave 20,482 genes. The genes and their corresponding P-values were put into a matrix of dimension 20,482 x 6, with rows corresponding to the P-values of genes and columns to datasets. Missing P-values we set 0.5 to avoid bias.
Table 1. Microarray platform(s) and tissue samples in each dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Platform type</th>
<th>Number of probe/spot</th>
<th>Prostate tumor sample</th>
<th>Normal prostate sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lapointe</td>
<td>custom cDNA microarray</td>
<td>44,639</td>
<td>62</td>
<td>41</td>
</tr>
<tr>
<td>Varambally</td>
<td>U133 2.0 plus</td>
<td>54,613</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Liu</td>
<td>U133 A, U133 B</td>
<td>44,792</td>
<td>44</td>
<td>13</td>
</tr>
<tr>
<td>Monzon</td>
<td>U95Av2, U95B, U95C</td>
<td>37,690</td>
<td>66</td>
<td>18</td>
</tr>
<tr>
<td>Singh</td>
<td>U95Av2</td>
<td>12,558</td>
<td>52</td>
<td>50</td>
</tr>
<tr>
<td>Welsh</td>
<td>U95A</td>
<td>12,558</td>
<td>25</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 2. Number of gene in each dataset

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Lapointe</th>
<th>Varambally</th>
<th>Liu</th>
<th>Monzon</th>
<th>Singh</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td># of genes</td>
<td>14980</td>
<td>19738</td>
<td>17917</td>
<td>15574</td>
<td>8799</td>
<td>8799</td>
</tr>
</tbody>
</table>

2.3 Statistical errors in Pointillist and proposed corrections

2.3.1 Introduction to Pointillist algorithm

Pointillist uses simulated null hypothesis data to count significant genes in each state during optimization process and provides three options for pooling weighted P-values: weighted Fisher’s inverse $\chi^2$ method, weighted Mudholkar-George’s Logit method and weighted Liptak-Stouffer’s Z-method. Before optimization, and for each input study, the acceptance-rejection method (Neumann, 1951) is used to generate data from the null distribution of the chosen pooling procedure. For instance, if there are 6 studies to be pooled and user chooses weighted Fisher’s inverse $\chi^2$ method option, then 6 sets of random numbers are generated from the $\chi^2$ distribution with 2 degree of freedom. These are used in each state to build an empirical null distribution of weighted combination.
statistics. Hwang’s procedure for counting significant genes in one state of the optimization process is illustrated in Figure 7. An empirical distribution function based on the weighted combination statistics is first constructed, and used to calculate the 100(1- \( \alpha \)) percentile, the critical value \( c \) corresponding to the input significant level \( \alpha \).

With \( K \) studies and \( G \) genes per dataset, the raw P-values from all the \( K \) studies are transformed each by the chosen pooling method and the transformed P-values are weighted, using “weights” generated by simulated annealing algorithm. The significance of weighted combination statistics is evaluated by comparison with the critical value \( c \).

Negative value of the count of significant genes serves as the “energy” of current state of optimization process.

**Figure 7.** Flow chart of Pointillist algorithm to count significant genes in one state of optimization process.
After the optimization process ends, Pointillist takes the optimal weight vector, fits the weighted pooled statistics to gamma or normal distribution, then uses the fitted distribution to evaluate the P-values of weighted pooled statistics. Specifically, Pointillist uses the optimal weight vector to combine independent random variables generated under the null hypothesis. The resulting combined statistics are then fitted by Matlab to a gamma distribution using “gamfit” function if Fisher’s weighted inverse \( \chi^2 \) method option was chosen, or the normal distribution by “normfit” function if Mudholkar-George’s Logit method or Liptak-Stouffer’s weighted Z-method was chosen. Finally, the raw P-values from studies are transformed and combined using optimal weight vector. The P-values of the pooled statistics are estimated by the fitted distribution. The procedure is illustrated in Figure 8.

\[
\begin{align*}
S_g &= \sum_{i=1}^{K} w_i F_n(p_{1g}), g = 1, \ldots, G \\
S_b &= \sum_{i=1}^{K} w_i S_{ib}, b = 1, \ldots, B \\
\end{align*}
\]

Figure 8. Pointillist process to estimate significance of the optimally pooled P-values
In this dissertation we point out the statistical errors in Hwang’s Pointillist algorithm, and we propose corrections to these errors in this section. We provide an alternative procedure to Hwang’s method in next section. Note that for comparison purpose, we have used two-sided P-values in both Pointillist and our procedure.

2.3.2 Fixed number of simulated null hypothesis data points in Pointillist

The Pointillist algorithm is based on essentially constructing an empirical distribution of the null distribution of weighted combination statistics. One of the consequences of using an empirical distribution of the combination statistic is that it could fail to estimate the tail probabilities accurately if the number of data elements generated to build it is not adequate. This is crucial in microarray experiments which involve the testing of thousands of hypothesis corresponding to thousands of genes. Specifically, in order to adjust for inflated probability of type I error, P-values must therefore be adjusted for multiple hypotheses testing to maintain the size of family wise error rates, (FWER) and/ or false discovery rates (FDR).

For the purpose of illustration, suppose we wish to test at FWER of 0.01. Using Bonferroni correction to control for multiple testing (Neyman & Pearson, 1928), each test has to be controlled at level \( \alpha = 0.01 / 20482 \approx 4.88E-7 \). The optimization process needs to maximize the number of genes whose pooled P-values are less than 4.88E-7. Hence, the approximation of the null distribution by an empirical distribution function would require that a huge sample size should be generated so that the total probability can be accurately approximated.
For illustration, if we start testing Pointillist with significance level 0.05, and then gradually reduce $\alpha$ to 4.88E-7, as the significance level tightens, the number of significant genes is supposed to reduce. But due to the fixing of number of samples of simulated null random variables at an inadequate size, Pointillist procedure is incapable of capturing such a low tail probability and reports the same number of significant genes for any $\alpha$ values under certain level. The result of Fisher’s weighted inverse $\chi^2$ method is illustrated in Table 3. The results of Mudholkar-George’s Logit method and Liptak-Stouffer’s weighted $Z$-method are similar and not shown.

Table 3. Original Pointillist result on different significance level

<table>
<thead>
<tr>
<th>$\alpha$</th>
<th># of significant</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>genes</td>
<td>Lapointe</td>
<td>Varambally</td>
<td>Liu</td>
<td>Monzon</td>
<td>Singh</td>
<td>Welsh</td>
</tr>
<tr>
<td>5.0000E-2</td>
<td>8,038</td>
<td>0.2153</td>
<td>0.0482</td>
<td>0.1903</td>
<td>0.2674</td>
<td>0.1536</td>
<td>0.1252</td>
</tr>
<tr>
<td>1.0000E-2</td>
<td>5,971</td>
<td>0.2481</td>
<td>0.0375</td>
<td>0.1799</td>
<td>0.205</td>
<td>0.1758</td>
<td>0.1537</td>
</tr>
<tr>
<td>1.0000E-3</td>
<td>4,321</td>
<td>0.2493</td>
<td>0.0345</td>
<td>0.1358</td>
<td>0.2093</td>
<td>0.1707</td>
<td>0.2005</td>
</tr>
<tr>
<td>1.0000E-4</td>
<td>3,446</td>
<td>0.2683</td>
<td>0.0373</td>
<td>0.2118</td>
<td>0.0789</td>
<td>0.2111</td>
<td>0.1926</td>
</tr>
<tr>
<td>1.9989E-5</td>
<td>3,228</td>
<td>0.2194</td>
<td>0.0425</td>
<td>0.2251</td>
<td>0.1585</td>
<td>0.2201</td>
<td>0.1344</td>
</tr>
<tr>
<td>1.9988E-5</td>
<td>2,921</td>
<td>0.1655</td>
<td>0.1509</td>
<td>0.1513</td>
<td>0.1851</td>
<td>0.2248</td>
<td>0.1224</td>
</tr>
<tr>
<td>1.0000E-5</td>
<td>2,921</td>
<td>0.1655</td>
<td>0.1509</td>
<td>0.1513</td>
<td>0.1851</td>
<td>0.2248</td>
<td>0.1224</td>
</tr>
<tr>
<td>1.0000E-6</td>
<td>2,921</td>
<td>0.1655</td>
<td>0.1509</td>
<td>0.1513</td>
<td>0.1851</td>
<td>0.2248</td>
<td>0.1224</td>
</tr>
<tr>
<td>4.8800E-7</td>
<td>2,921</td>
<td>0.1655</td>
<td>0.1509</td>
<td>0.1513</td>
<td>0.1851</td>
<td>0.2248</td>
<td>0.1224</td>
</tr>
</tbody>
</table>

With weighted Fisher’s method, Pointillist reports the same weight vector and stabilizes at 2,921 genes for all significance levels $< 1.9989E-5$. This is because with our experiment data (one 20,482 x 6 matrix), Pointillist is coded to use the same fixed number of 25,015 random variables from the null distribution of statistic elements to build the empirical cumulative distribution function, regardless of the input $\alpha$ value. However, the sample size of 25,015, is inadequate in this asymptotic approximation of the true
distribution function. To build cumulative distribution function, the statistics of these data
elements are sorted in ascending order, and the corresponding cumulative density array of
the same size is built with interval of \( 1 / 25,015 \approx 3.9976 \times 10^{-5} \). The schema of its
cumulative distribution function is shown in Table 4.

### Table 4. Pointillist’s empirical cumulative distribution function

<table>
<thead>
<tr>
<th>Sorted statistics</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>…</th>
<th>S25014</th>
<th>S25015</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cumulative density</td>
<td>0</td>
<td>3.9976E-5</td>
<td>2*3.9976E-5</td>
<td>…</td>
<td>1-3.9976E-5</td>
<td>1</td>
</tr>
</tbody>
</table>

To find the critical value \( c \) of input \( \alpha \), at first, from the cumulative distribution array,
Pointillist finds the element closest to \( 1 - \alpha \), then finds its corresponding statistics and
uses it as critical value \( c \). For all significance level \(< 1.9989E-5 \) (half of the interval), the
closest elements are always the same, i.e., the last element in the cumulative distribution
array, and the critical values are the same too, i.e., the largest statistics. Since the same
critical value is used to count significant genes for all significance levels \(< 1.9989E-5 \),
Pointillist always reaches the same “optimal” state and reports the same amount of
significant genes even the significance levels are different.

A simple argument can be used to estimate the needed number of data points. Let
\( N_{\alpha_0} \) = \# of P-values needed until one of them is \( \leq \alpha \), then \( N_{\alpha} \) has a geometric distribution
with probability of success \( \alpha_0 \) and \( E(N_{\alpha}) = 1/\alpha \). Hence based with a Bonferroni correction
for FWER, the number simulated data points should be at least \( |G|/\alpha_0 \) where \( |G| \) is the
number of genes and \( \alpha_0 \) is the required FWER.
2.3.3 Pointillist’s simulation of the null distribution

To compute the critical value, the Pointillist approach generates a statistic with the null distribution of the transformed P-value $P_i$, $1 \leq i \leq K$. These are weighted and pooled, and then used to build an empirical distribution function of which is then used to find critical value $c$, for level of significance $\alpha$. A fundamental source of inaccuracy in the Pointillist algorithm is in its use of unduly complicated simulation procedures to approximate null distributions that can easily be obtained exactly by the use of a probability integral transform, namely $F^{-1}(U)$, where $F$ is the target null distribution function. For example, the exact null distribution of $\log(P/(1-P))$ in the Logit method should have been generated, not by the t-distribution with 9 degrees of freedom, but as $\log(U/(1-U))$, where $U$ is the Uniform(0,1) random variable.

Pointillist provides three options for weighted pooling statistic: Fisher’s weighted inverse $\chi^2$ method, weighted Mudholkar-George’s Logit method and Liptak-Stouffer’s weighted Z-method. Fisher’s weighted inverse $\chi^2$ method is correctly implemented by pulling random null hypothesis statistics from $\chi^2$ distribution with 2 degree of freedom, and then pulling with weights to get

$$S_w = \sum_{i=1}^{K} w_i(S_i).$$

This can then be used to build empirical distribution of combined weighted null statistics, and compute the critical value $c$. The P-values from datasets are pooled using the weighted statistic,

$$F_w = \sum_{i=1}^{K} -2w_i \ln(P_i)$$

and compared with critical value $c$ to count number of significant genes.
Similarly, Liptak-Stouffer’s weighted Z-method is correctly implemented by pulling random null hypothesis statistics from standard normal distribution with mean 0 and standard deviation 1. Then in each state, random samples of null hypothesis statistics are pooled with weights, \( S_w = \frac{\sum_{i=1}^{K} w_i S_i}{\sqrt{K}} \) can then be generated. The P-values from datasets are pooled using the weighted statistic,

\[
Z_w = \frac{-\sum_{i=1}^{K} w_i \Phi^{-1}(P_i)}{\sqrt{K}}.
\]

Note that in the original Liptak-Stouffer’s weighted Z-method, the constant should be

\[
\frac{1}{\sqrt{\sum_{i=1}^{K} w_i^2}}
\]

so that the significance of weighted pooled statistics can be evaluated by standard normal distribution. However, Pointillist uses constant

\[
\frac{1}{\sqrt{K}}
\]

which is only true for equal weight. However, the constant does not have effect since the same constant is applied to both simulated null hypothesis data and real experiment data. Actually, while using an empirical distribution, it is not necessary to apply any constant to normalize the statistics.

However, in the formulation of the Logit method, Hwang et al. (2005) made a number of statistical errors. First of all, rather than approximating the weighted sum of Logit statistics by treating each unweighted component as a logistic random variable, Pointillist generates independent and identically distributed copies of student-t\(_9\) random variables, and the distribution of a convolution of the weighted version of these student-t\(_9\)
variables was approximated. The distribution of a convolution of $K$ unweighted logistic random variables can be approximated by student-t distribution with $5K + 4$ degrees of freedom with assumption of $K \geq 2$ (George & Mudholkar, 1983). By replacing each logistic variate by a student-$t_9$ random variable, Hwang erroneously replaced a standardized convolution of weighted logistic random variables by a convolution of weighted normalized student-$t_9$ random variable. In the process, Pointillist used a wrong normalizing constant to standardize the weighted Logit statistics of experiment data,

$$L_W = -\sqrt{\frac{15K + 12}{K(5K + 2)}} \sum_{i=1}^{K} w_i \ln\left(\frac{p_i}{1 - p_i}\right)$$

The error in Pointillist algorithm is then compounded by using the statistic,

$$S_W = \sqrt{\frac{15K + 12}{K(5K + 2)}} \sum_{i=1}^{K} w_i S_i$$

where $S_i$ is random variate of student-$t$ distribution with 9 degrees of freedom, to estimate the critical value for the weighted Logit statistics. We found that this approximation of an approximation, standardized as given above, leads to an overestimation of the number of significant genes, especially in case of small $\alpha$ (see Table 5).

We describe a statistically correct version of the Pointillist method for the weighted Logit method by recognizing that each null hypothesis $S_i$ can be generated by $\log(U_i/\left(1-U_i\right)$ where $U_i$’s are independent uniform(0,1) random variables. Specifically, generate a set of $K$ logistic variates $L_{1b}, \ldots, L_{Kb}, b = 1, \ldots, B$ where $L_{ib} = \log(U_{ib}/\left(1-U_{ib}\right), i = 1, \ldots, K, b = 1, \ldots, B$ and $U_{ib}$ are uniform (0, 1) random variables. The weighted statistic $S_w = \sum_{i=1}^{K} w_i S_i$ is then being used to find the critical value $c$. The P-values from datasets
are pooled using the weighted statistic: \( L_W = - \sum_{i=1}^{K} w_i \log \left( \frac{p_i}{1-p_i} \right) \), and they are compared with critical value \( c \) to count number of significant genes. We can also apply a similar method to generate the exact null distribution of the Fisher’s inverse \( \chi^2 \) method and Liptak-Stouffer’s Z-method: For weighted inverse \( \chi^2 \) method, each null hypothesis \( F_{ib} \) can be generated by \(-2*\log(U_{ib})\) and for weighted Z-method, each null hypothesis \( Z_{ib} \) can be generated by \( \Phi^{-1}(1-P_{ib}) \) where \( \Phi \) is the standard normal cumulative density distribution. Our procedure replaces, with improvement in accuracy, the acceptance-rejection method that Pointillist uses. It is clear that while the acceptance-rejection procedure is very useful, the simulation here can be achieved more accurately, with a less complicated algorithm.

2.3.4 Pointillist’s fitted distribution

An empirical distribution is a weak tool in evaluating significance of statistics since its smallest P-value is 1/number of elements used to build empirical distribution. To better evaluate the P-values of optimally pooled experiment data, after the optimization process ends, Pointillist fits the optimally pooled null hypothesis statistics to an approx of the null of distribution, then uses the fitted distribution to evaluate the optimally pooled statistics of experiment data. For Fisher’s weighted inverse \( \chi^2 \) method, Pointillist correctly fits the weighted pooled null hypothesis statistics as a \( \chi^2 \) distribution. For Liptak-Stouffer’s weighted Z-method, Pointillist also correctly fits the weighted pooled null hypothesis statistics as a normal distribution. However, for weighted Logit method, in addition to the errors discussed above, Pointillist also fits the pooled weighted null hypothesis statistics incorrectly. It approximates the null distribution by a normal distribution instead of student-t distribution. We found that the fitted normal distribution
overestimates the P-value of optimal pooled Logistic statistics, since the process leads to lighter tail distribution which makes approximation at the tails of the distribution inaccurate, with consequence of underestimating of the magnitude of the P-value of the pooled statistics, and a corresponding overestimation of the number of significant genes (see Table 5). The rational for the use of the normal approximation by Pointillist is that the t-distribution, which is used to approximate the null distribution of the Logit statistics can be approximated by normal distribution. However, the approximation of the t-distribution by normal distribution is reasonable, only for large degrees of freedom. For small to moderate degrees of freedom, this approximation is poor at the tails of the distribution. It is well known that the t-distribution has heavier tail than the normal distribution. The significance of optimally pooled logistic statistics cannot be accurately evaluated unless the approximation of convolution of weighted logistic variables is derived, which will be shown in next section.

We demonstrate Pointillist’s errors by comparing its test result with the corrected version. To better illustrate the error of empirical distribution built by weighted Logit method in Pointillist at low significance level, the original Pointillist software is slightly modified to increase the number of null hypothesis data points to 100/α if 100/α > 25,015. In our corrected version, for all three pooling methods (weighted inverse $\chi^2$ method, weighted Logit method and weighted Z method), the null hypothesis statistics are generated by transforming uniform random variables with range of (0 1) as described before. The results are summarized in Table 5.

Since the number of null hypothesis data elements is determined by input α instead of a fixed number, the built empirical distribution is always accurate enough to evaluate
input $\alpha$. For instance, with weighted inverse $\chi^2$ method, for all significance level $< 1.9989E-5$, Pointillist always reaches the same “optimal” state and reports the same number of 2,921 significant genes (see Table 3). With empirical distribution based on larger sample size, 2,359 significant genes are reported instead at $\alpha = 1E-5$ level.

Although, there are statistical issues with Pontillist implementation of the weighted Fisher’s $\chi^2$ method and weighted Z method, the difference between the number of significant genes estimated by exact null distribution, Pointillist’s simulated null distribution and the fitted distribution is negligible. There is however substantial difference in power between Pontillist’s simulated null distribution and our exact null distribution when the weighted Logit method is implemented, as can be seen from Table 5 and Figure 9. This difference increases as $\alpha$ decreases, becoming quite substantial when $\alpha$ drops to 1E-3 level and less. The difference in counts of significant genes is 6.2%, 12.8% respectively at 1E-3 and 1E-5 level. Such big differences indicate the error of using student-t variables to build empirical distribution.

Using the normal distribution instead of Student-t to approximate the distribution of the weighted combination statistic produces an inflated count of significant genes. At $\alpha = 1E-3$ level, the over-estimation increases from 6.2% to 10%. At $\alpha = 1E-4$ level, the over-estimation increases from 8.28% to 16.7%. At $\alpha = 1E-5$ level, the over-estimation increases from 12.8% to 27.7%. Although we did not perform the comparisons for $\alpha$ levels less than 1E-5 because of the prohibitive computation expensiveness of Pointillist algorithm (see section 2.3.5), it is clear that there is an increasing trend in the inflated count rate as $\alpha$ drops when the weighted Logit method is implemented by Pointillist.
Table 5. Comparison of number of significant genes detected by different distributions.


<table>
<thead>
<tr>
<th>alpha</th>
<th>Distribution</th>
<th>Gamma</th>
<th>Z</th>
<th>Logit</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.00E-02</td>
<td>D1</td>
<td>7,899</td>
<td>7,450</td>
<td>7,749</td>
</tr>
<tr>
<td></td>
<td>D2</td>
<td>8,038</td>
<td>7,463</td>
<td>7,916</td>
</tr>
<tr>
<td></td>
<td>D3</td>
<td>7,884</td>
<td>7,452</td>
<td>7,783</td>
</tr>
<tr>
<td>1.00E-02</td>
<td>D1</td>
<td>5,801</td>
<td>5,269</td>
<td>5,653</td>
</tr>
<tr>
<td></td>
<td>D2</td>
<td>5,971</td>
<td>5,211</td>
<td>5,799</td>
</tr>
<tr>
<td></td>
<td>D3</td>
<td>5,848</td>
<td>5,185</td>
<td>5,805</td>
</tr>
<tr>
<td>1.00E-03</td>
<td>D1</td>
<td>4,078</td>
<td>3,619</td>
<td>3,953</td>
</tr>
<tr>
<td></td>
<td>D2</td>
<td>4,120</td>
<td>3,575</td>
<td>4,199</td>
</tr>
<tr>
<td></td>
<td>D3</td>
<td>4,206</td>
<td>3,473</td>
<td>4,348</td>
</tr>
<tr>
<td>1.00E-04</td>
<td>D1</td>
<td>3,043</td>
<td>2,692</td>
<td>2,969</td>
</tr>
<tr>
<td></td>
<td>D2</td>
<td>3,103</td>
<td>2,661</td>
<td>3,214</td>
</tr>
<tr>
<td></td>
<td>D3</td>
<td>3,196</td>
<td>2,610</td>
<td>3,466</td>
</tr>
<tr>
<td>1.00E-05</td>
<td>D1</td>
<td>2,359</td>
<td>1,973</td>
<td>2,303</td>
</tr>
<tr>
<td></td>
<td>D2</td>
<td>2,349</td>
<td>1,989</td>
<td>2,598</td>
</tr>
<tr>
<td></td>
<td>D3</td>
<td>2,476</td>
<td>1,929</td>
<td>2,940</td>
</tr>
</tbody>
</table>
Figure 9. Comparison of number of significant genes detected at various \( \alpha \) values by different distributions for Logit method. Distribution 1 (D1): Exact null distribution. Distribution 2 (D2): Pointillist simulated null distribution. Distribution 3 (D3): Pointillist normal distribution approximating its simulated null distribution.

These inflated counts produced by the Pointillist algorithm for the weighted Logit procedure are statistically meaningless because neither the representation of the convolution of logistic variables by a convolution of student-\( t \) variables nor the use of a normal percentile to estimate the critical value produces a test that maintains size \( \alpha \). For instance, at \( \alpha = 1E-3 \) level, Pointillist finds the optimal weights [0.3074 0.0468 0.1379 0.187 0.1528 0.1682] and \( c \) is 0.59513162. Since in Pointillist, the P-values from datasets are pooled using the weighted statistic,
We apply the same constant (which is 0.232 while K=6) and weights to logistic variables to build exact null distribution. According to exact null distribution, this $c$ value actually corresponds to P-value of 0.00166. Pointillist further fits the null hypothesis data elements into normal distribution, ends with one normal distribution of mean= -1.38656e-03 and standard deviation= 1.8671643e-01. By this normal distribution, $c$ value corresponds to $\alpha = 1E-3$ is 0.5755. According to empirical distribution built correctly by logistic variables, this $c$ value actually corresponds to P-value of 0.00224. In other words, Pointillist actually reports number of significant genes at P-value of 0.00166 by its simulated null distribution and at P-value of 0.00224 by its fitted normal distribution.

2.3.5 Computational inefficiency in Pointillist

Correcting the distributional errors in Pointillist, but maintaining its empirical distribution approach does not remove the computation inefficiency in the algorithm. As $\alpha$ drops, more null hypothesis data points are needed to build empirical distribution, so more physical memory is required to load all data into memory. The limit of physical memory will quickly be reached if $\alpha$ is small. Similarly, as number of null hypothesis data elements increases rapidly, the CPU time gets prohibitively long. We execute the corrected Pointillist procedure on one Dell PowerEdge 860 server (2.66 GHz CPU, 8 GB memory). The occupied memory and CPU time are illustrated in Table 6. At $\alpha = 1E-5$, the occupied memory is 3 GB and the computation takes almost one day. It is clear that the empirical distribution method is not a practical procedure for performing the
optimally weighted pooling procedure at FWER of 0.01 level \((\alpha = 4.88E-7)\) since it will require 150 GB physical memory and the expected computation time of about 50 days.

Table 6. Corrected Pointillist’s performance and occupied memory at different significance level

<table>
<thead>
<tr>
<th>alpha</th>
<th># of elements</th>
<th>memory</th>
<th>Gamma (min)</th>
<th>Z (min)</th>
<th>Logit (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.00E-02</td>
<td>25,015</td>
<td>6 MB</td>
<td>&lt; 1</td>
<td>2</td>
<td>3.5</td>
</tr>
<tr>
<td>1.00E-02</td>
<td>25,015</td>
<td>6 MB</td>
<td>&lt; 1</td>
<td>1.5</td>
<td>7</td>
</tr>
<tr>
<td>1.00E-03</td>
<td>100,000</td>
<td>32 MB</td>
<td>3</td>
<td>4.8</td>
<td>5</td>
</tr>
<tr>
<td>1.00E-04</td>
<td>1,000,000</td>
<td>300 MB</td>
<td>42</td>
<td>83</td>
<td>75</td>
</tr>
<tr>
<td>1.00E-05</td>
<td>10,000,000</td>
<td>3 GB</td>
<td>1050</td>
<td>816</td>
<td>816</td>
</tr>
</tbody>
</table>

2.4 Estimate P-value of weight pooled statistics directly by its theoretical null hypothesis distribution.

Here we propose to a computationally efficient procedure, based on the use of an approximation of the theoretical null distribution of the weighted combination statistics, to estimate P-values. The proposed method is efficient in the sense that it minimizes the physical memory required while keeping the computation time constant for all significance levels. Our process for counting the significant genes in one state is shown in Figure 10. At each state, the pooling method transforms the raw P-values to statistics and sums up them with weights in the input weight vector. The weighted combination statistics follows the theoretical distribution built on the input weight vector. So the P-values of all pooled statistics can be directly estimated by the distribution. The P-values are then compared with \(\alpha\) value to count the number of significant genes.
Figure 10. Procedure to count significant genes based on raw two-sided P-values by theoretical distribution

2.4.1 Approximation of theoretical null hypothesis distribution.

The key to this strategy is to derive the distribution of weighted pooled statistics based on input weight vector. The strategy we have adopted is to use the method of moments for this purpose. For weighted Fisher’s inverse $\chi^2$ method, the weighted pooled statistics is in form of
\[ F_w = \sum_{i=1}^{K} -2w_i \ln(P_i) \]

where \( w_i \) is the weight assigned to study \( i \).

We approximated the distribution of \( F_w \) with Gamma distribution by method of moment. The result Gamma distribution has shape parameter

\[ a = \frac{\left( \sum_{i=1}^{K} w_i \right)^2}{\sum_{i=1}^{K} w_i^2} \]

and scale parameter

\[ b = \frac{2 \sum_{i=1}^{K} w_i^2}{\sum_{i=1}^{K} w_i} \]

which is essentially the chi-square distribution reported by Hou (2005).

For weighted Z-method, the weighted pooled statistics is given by

\[ Z_w = \frac{\sum_{i=1}^{K} w_i Z_i}{\sqrt{\sum_{i=1}^{K} w_i^2}} \]

where \( w_i \) is the weight assigned to study \( i \). The distribution of \( Z_w \) is standard normal distribution with mean 0 and standard deviation 1 (Liptak, 1958).

For weighted Logistic method, the weighted pooled statistics is in form of

\[ L_w = \sum_{i=1}^{K} w_i \ln\left( \frac{p_i}{1-p_i} \right) \]
where $w_i$ is the weight assigned to study $i$. We derived the approximate distribution of $L_w$ with $t$ distribution as follows.

1. Computation of the approximating student-$t$ degree of freedom

Let

\begin{align*}
(1.1) \quad L_w &= \sum_{i=1}^{K} w_i l_i \\
(1.2) \quad L_w^* &= \frac{\sqrt{3} \sum_{i=1}^{K} w_i l_i}{\pi \sqrt{\sum_{i=1}^{K} w_i^2}} \\
(1.3) \quad t_v^* &= \frac{\sqrt{\frac{v-2}{v}} t_v}{\sqrt{\frac{\nu}{v}}}
\end{align*}

where $t_v$ is a Student $t$-random variable with $v$ degrees of freedom and $t_v^*$ is the standardized version.

Since Kurtosis of $L_w$ and $L_w^*$, $\beta_2(L_w) = \frac{E(L_w^4)}{(E(L_w^2))^2} = \frac{E(L_w^*4)}{(E(L_w^*2))^2} = \beta_2(L_w^*)$, we want a Student-$t$ random variable, $t_v$ with $v$ degrees of freedom, such that $\beta_2(t_v) = \beta_2(L_w)$.

First,

\begin{align*}
E(L_w^4) &= \sum_{i=1}^{k} w_i^4 E(l_i^4) + \binom{4}{2} \sum_{1 \leq i<j \leq k} w_i^2 w_j^2 E(l_i^2) E(l_j^2) \\
&= \pi^4 \left[ \frac{7}{15} \sum_{i=1}^{k} w_i^4 + \frac{2}{3} \sum_{1 \leq i<j \leq k} w_i^2 w_j^2 \right]
\end{align*}

Next:
\[ E(L_{w}^2) = \frac{\pi^2}{3} \sum_{i=1}^{k} w_i^2 \]

Hence,

\[ \beta_2(L_w) = \frac{\pi^4}{15} \frac{\sum_{i=1}^{k} w_i^4}{\left( \frac{\pi^2}{3} \sum_{i=1}^{k} w_i^2 \right)^2} \left[ \frac{7}{\sum_{i=1}^{k} w_i^4} + \frac{2}{3} \sum_{1 \leq i < j \leq k} w_i^2 w_j^2 \right] \]

\[ = \frac{\frac{21}{5} \sum_{i=1}^{k} w_i^4 + 6 \sum_{1 \leq i < j \leq k} w_i^2 w_j^2}{\left( \sum_{i=1}^{k} w_i^2 \right)^2} \]

\[ = \frac{3 (\sum_{i=1}^{k} w_i^4 + 2 \sum_{1 \leq i < j \leq k} w_i^2 w_j^2)}{5 \sum_{i=1}^{k} w_i^4} \]

\[ = \frac{3 (\sum_{i=1}^{k} w_i^2)^2 + 6 \sum_{i=1}^{k} w_i^4}{5 (\sum_{i=1}^{k} w_i^2)^2} \]

\[ = 3 + \frac{6}{5} \frac{\sum_{i=1}^{k} w_i^4}{(\sum_{i=1}^{k} w_i^2)^2} \]

We want \( v \) such that \( \beta_2(L_w) = \beta_2(t_v) \), that is, \( 3 + \frac{6}{5} \frac{\sum_{i=1}^{k} w_i^4}{(\sum_{i=1}^{k} w_i^2)^2} = 3 + \frac{6}{v-4} \), this lead to

\[ (1.4) \]

\[ \frac{6}{v-4} = \frac{6}{5} \frac{\sum_{i=1}^{k} w_i^4}{(\sum_{i=1}^{k} w_i^2)^2} \]

That is,

\[ (1.5) \]

\[ v = 4 + 5 \left( \frac{\sum_{i=1}^{k} w_i^2}{\sum_{i=1}^{k} w_i^4} \right)^2 \]
CHECK

If \( w_1 = \ldots = w_k = 1, \sum_{i=1}^{k} w_i^2 = \sum_{i=1}^{k} w_i^4 = K \)

Hence,

\[
(1.6) \quad v = 4 + 5 \frac{K^2}{K}
\]

\[
(1.7) \quad v = 4 + 5K
\]

2. Approximation

\[
(2.9) \quad P(L^*_W \leq x) = P(t^*_v \leq x)
\]

\[
(2.10) \quad = P(t_v \leq \frac{\sqrt{v}}{\sqrt{v-2}} x)
\]

Therefore,

\[
P(L_W \leq x) = P \left( L^*_W \leq \frac{x}{\pi} \sqrt{\frac{3}{\sum_{i=1}^{k} w_i^2}} \right)
\]

\[
= P \left( t^*_v \leq \frac{x}{\pi} \sqrt{\frac{3}{\sum_{i=1}^{k} w_i^2}} \right)
\]

\[
= P \left( t_v \leq \frac{x}{\pi} \sqrt{\frac{3v}{(v-2) \sum_{i=1}^{k} w_i^2}} \right)
\]
2.4.2 Result of pooling two-sided P-values using approximated null distributions

We apply our procedure with three combination methods at different $\alpha$ levels, and compare with the number of significant genes of optimal state with the one from corrected Pointillist empirical distribution procedure. The results are illustrated in Table 7. For all three pooling methods, the results of theoretical distribution are consistently close to the results of corrected empirical distribution. With theoretical distribution, optimization can be performed at any $\alpha$ level, for instance, $\alpha = 4.88E-7$ for FWER of 0.01 level. In addition, the P-value of optimally pooled statistics can be estimated without any limitation. In contrast, with empirical distribution, the best estimated P-value is $1 / \text{number of null hypothesis data elements}$. From computation perspective, the memory requirement and computation time is independent of $\alpha$. For all pooling methods and all $\alpha$ levels of our experiment data, the CPU time is less than 5 minutes.

Table 7. Comparison of result from exact null distribution with result from theoretical distribution

<table>
<thead>
<tr>
<th>alpha</th>
<th>Distribution</th>
<th>Gamma</th>
<th>Z</th>
<th>Logit</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.00E-02</td>
<td>Exact null distribution</td>
<td>7,899</td>
<td>7,450</td>
<td>7,749</td>
</tr>
<tr>
<td></td>
<td>Theoretical null distribution</td>
<td>7,934</td>
<td>7,415</td>
<td>7,762</td>
</tr>
<tr>
<td>1.00E-02</td>
<td>Exact null distribution</td>
<td>5,801</td>
<td>5,269</td>
<td>5,653</td>
</tr>
<tr>
<td></td>
<td>Theoretical null distribution</td>
<td>5,915</td>
<td>5,232</td>
<td>5,643</td>
</tr>
<tr>
<td>1.00E-03</td>
<td>Exact null distribution</td>
<td>4,078</td>
<td>3,619</td>
<td>3,953</td>
</tr>
<tr>
<td></td>
<td>Theoretical null distribution</td>
<td>4,208</td>
<td>3,564</td>
<td>3,913</td>
</tr>
<tr>
<td>1.00E-04</td>
<td>Exact null distribution</td>
<td>3,043</td>
<td>2,692</td>
<td>2,969</td>
</tr>
<tr>
<td></td>
<td>Theoretical null distribution</td>
<td>3,217</td>
<td>2,618</td>
<td>2,913</td>
</tr>
<tr>
<td>1.00E-05</td>
<td>Exact null distribution</td>
<td>2,359</td>
<td>1,973</td>
<td>2,303</td>
</tr>
<tr>
<td></td>
<td>Theoretical null distribution</td>
<td>2,539</td>
<td>1,965</td>
<td>2,215</td>
</tr>
<tr>
<td>4.88E-07</td>
<td>Exact null distribution</td>
<td>N/A</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td></td>
<td>Theoretical null distribution</td>
<td>1,947</td>
<td>1,426</td>
<td>1,589</td>
</tr>
</tbody>
</table>
2.5 one-sided P-values should be used to pool gene expression data.

It is common in biomedical studies to report two-sided P-values when the direction of the alternative hypothesis is irrelevant or ambiguous. When the direction of alternative is important, many investigators take the recommendation of Gibbons and Pratt (1975) and report a one-sided P-value along with the direction of the rejection. When dealing with applications in which the formulation of the alternative intrinsically two-sided, George and Mudholhar (1990) have proposed a general procedure for computing a two-sided P-value. Their definition reduces to the usual doubling of the one-sided P-value when the null density is symmetric. In gene expression studies, it is not only important to know if a gene is differentially expressed between different conditions (for instance, tumor versus normal tissue, treatment versus control), but also necessary to know the direction of differentiation (for instance, if a gene is up regulated or down regulated in tumor compared with normal tissue). So it is widely recommended that a one-sided P-value should be used because it contains the information of direction whereas a two-sided P-value does not (Marot et al., 2009; Whitlock, 2005; Zaykin, 2011). Using one-sided P-value is particular important when pooling evidence of microarray data from multiple studies. Since the microarray data are inherently noisy, a gene could be observed both up-regulated in one study and down-regulated in another study. Accordingly, in order to extract unambiguous and informative decision from multiple studies, the procedure for pooling of evidence from each gene should not discard information of the genes direction of regulation. In practice, the testing of hypothesis related to up-regulation should be separated from the hypotheses related to down-regulation. This practice will increase the power of the pooling tests of genes being observed to be regulated in the same direction.
2.5.1 Generation of one-sided P-values for each gene

For the pre-processed data described in section 2.2.2, we test gene regulation in both directions by the t.test function in R. To simplify further analysis and interpretation, we consistently pass the normal tissue group values to the parameter ‘x’, and the tumor group values to parameter ‘y’. For each gene, at first, we test the null hypothesis that mean expression of normal tissue group is not less than its mean in tumor group against the alternative hypothesis that it is less by choosing the “less” option of parameter “alternative” in t.test function. In this dissertation we will refer to the P-values as up-regulation P-values. If the expression measurement in normal tissue group is significantly less than expression measurement in tumor group, the up-regulation P-value < 0.05. Similarly, the calculation of down-regulation P-value is choosing the “greater” option of parameter “alternative” in t.test function. If the expression measurement in the normal group is significantly greater than the expression in tumor group, down-regulation P-value will be < 0.05. Since t-distribution is symmetric, up-regulation P-value = 1 – down-regulation P-value.

Finally, the up-regulation P-values of genes from all datasets were put into a matrix of dimension 20,482 x 6, the up-regulation P-values matrix. The down-regulation P-values were put into one matrix of the same dimension, called down-regulation P-values matrix. We set the missing P-values to 0.5 to avoid bias.

2.5.2 Comparison of pooling two-sided P-values with pooling one-sided P-values

We show the two advantages of pooling one-sided P-values. 1. To exclude the “noisy” genes that are detected differentially expressed, but in opposite directions in different studies. 2. To increase the power to find the genes that are detected differentially
expressed in the same direction, but with weak statistical significance in individual datasets.

2.5.2.1 Gene examples

We selected two genes to show the advantages of pooling one-sided P-values. We pooled their P-values from individual datasets with equal weight Logit method. The result is in Table 8. The first gene, with GeneID 25875, shows the increased power to detect gene showing the same regulation direction across multiple studies. Although its up-regulation P-values are not significant in most studies, it shows the same direction of differentiation, and its pooled up-regulation P-value is significant (P=0.00516805). In contrast, its pooled two-sided P-value is only 0.15642249. The second example shows that pooling one-sided P-value helps to exclude genes with opposite directed differentiation across studies. For GeneID 3195, it shows differentiation in one direction in half studies, and differentiation in the opposite direction in another half studies. So neither its pooled up-regulation P-value nor pooled down-regulation P-value is significant. In contrast, its pooled two-sided P-value is significant (P=3.5031E-06), which will lead to wrong conclusion.
Table 8. Comparison of pooling one-sided P-values with pooling two-sided P-values on dataset level

<table>
<thead>
<tr>
<th>GeneID</th>
<th>P-value type</th>
<th>Pooled P-value</th>
<th>Lapointe</th>
<th>Varambally</th>
<th>Liu</th>
<th>Monzon</th>
<th>Singh</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td>25875</td>
<td>up-regulation P-value</td>
<td>0.00516805</td>
<td>0.029467</td>
<td>0.4606943</td>
<td>0.276916</td>
<td>0.010728</td>
<td>0.258464</td>
<td>0.180576</td>
</tr>
<tr>
<td></td>
<td>down-regulation P-value</td>
<td>0.99483195</td>
<td>0.970533</td>
<td>0.5393057</td>
<td>0.723084</td>
<td>0.989272</td>
<td>0.741536</td>
<td>0.819424</td>
</tr>
<tr>
<td></td>
<td>two-sided P-value</td>
<td>0.15642249</td>
<td>0.058935</td>
<td>0.9213887</td>
<td>0.553832</td>
<td>0.021457</td>
<td>0.516927</td>
<td>0.361151</td>
</tr>
<tr>
<td>3195</td>
<td>up-regulation P-value</td>
<td>0.74033745</td>
<td>0.320773</td>
<td>0.0226876</td>
<td>0.000291</td>
<td>0.998955</td>
<td>0.999491</td>
<td>0.734386</td>
</tr>
<tr>
<td></td>
<td>down-regulation P-value</td>
<td>0.25966255</td>
<td>0.679227</td>
<td>0.9773124</td>
<td>0.999709</td>
<td>0.001045</td>
<td>0.000509</td>
<td>0.265614</td>
</tr>
<tr>
<td></td>
<td>two-sided P-value</td>
<td>3.50E-06</td>
<td>0.641547</td>
<td>0.0453752</td>
<td>0.000582</td>
<td>0.002091</td>
<td>0.001018</td>
<td>0.531229</td>
</tr>
</tbody>
</table>
2.5.2.2 Comparison of significant genes detected by pooling one-side P-values and by pooling two-sided P-values

To further indicate the benefit of pooling one-sided P-values, we compare the significant genes detected by their pooled one-sided P-values with the significant genes detected by their pooled two-sided P-values. We pooled the P-values with equal weight Logistic method. Since t-distribution is symmetric, we set $\alpha$ as 0.025 for one-sided P-value and 0.05 for two-sided P-value so as for fair comparison. By pooling up-regulation P-values, we got 3,512 significantly up-regulated genes. By pooling down-regulation P-values, we got 3,181 significantly down-regulated genes. We pooled these 6,693 genes together as genes that exhibit significant expression difference between prostate tumor and normal prostate tissue. Using two-sided P-values, we got 7,456 significant genes. We compared these two lists of genes. The result is in Table 9. The overlap is 5,712 genes. There are 1,744 genes that are only detected by their pooled two-sided P-values. They are the “noisy” genes which we want to exclude. There are 981 genes that are detected only by their pooled one-sided P-values. All of these results show the increased power by pooling one-sided P-values.

Table 9. Comparison of one-sided-significant-genes and two-sided-significant-genes

<table>
<thead>
<tr>
<th>$\alpha &lt; 0.05$ for two-sided P-value</th>
<th>$\alpha &lt; 0.025$ for one-sided P-value</th>
<th>Common</th>
<th>two-sided only</th>
<th>one-sided only</th>
</tr>
</thead>
<tbody>
<tr>
<td>8315</td>
<td>3,512 up, 3,181 down</td>
<td>5,712</td>
<td>1,744</td>
<td>981</td>
</tr>
</tbody>
</table>
2.6 Procedure to pool gene expression data by one-sided P-values

Our pooling procedure for one-sided P-values is shown in Figure 11. For each weight vector, at first, apply the weight vector to up-regulation P-values matrix, calculate the pooled up-regulation P-value with theoretical distribution for each gene, count the number of genes whose pooled up-regulation P-values are less than the input $\alpha_{\text{up-regulation}}$, and save this count as N1. Then apply the same weight vector to down-regulation P-value matrix, calculate the pooled down-regulation P-value with theoretical distribution for each gene, count the number of genes whose pooled down-regulation P-values are less than the input $\alpha_{\text{down-regulation}}$, and save this count as N2. Finally, N which is the sum of N1 and N2, will serve as “energy” of current state. Note that $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$ might be different (see section 2.6.3).
Figure 11. Procedure to count significant genes based on raw one-sided P-values by theoretical distribution

\[ N = N^\uparrow + N^\downarrow \]
We apply our procedure to the experiment datasets at different significance levels to test its performance. We also run each procedure with equal weight vector for comparison purpose. All three pooling methods (weighted inverse chi-square method, weighted Z method and weighted Logit method) are performed respectively.

2.6.1 Pooling result with $\alpha_{\text{up-regulation}} = \alpha_{\text{down-regulation}} \leq 0.05$

We first try the non-conservative significance level of $\alpha_{\text{up-regulation}} = \alpha_{\text{down-regulation}} = 0.05$. In Table 10, we illustrate that in each dataset, the count of genes whose raw one-sided P-values are less than 0.05 to compare with those using optimal weights.

Table 10. In each dataset, count of genes whose one-sided P-values are less than 0.05

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Lapointe</th>
<th>Varambally</th>
<th>Liu</th>
<th>Monzon</th>
<th>Singh</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td>up-regulation</td>
<td>3,387</td>
<td>2,214</td>
<td>2,975</td>
<td>3,249</td>
<td>1,397</td>
<td>1,901</td>
</tr>
<tr>
<td>down-regulation</td>
<td>3,403</td>
<td>1,100</td>
<td>2,146</td>
<td>2,218</td>
<td>2,713</td>
<td>1,892</td>
</tr>
<tr>
<td>total</td>
<td>6,790</td>
<td>3,314</td>
<td>5,121</td>
<td>5,467</td>
<td>4,110</td>
<td>3,793</td>
</tr>
</tbody>
</table>

The pooling results are illustrated in Table 11. For each pooling method, the weighted pooling strategy yields more significant genes than equal weight pooling strategy does. For instance, when the raw P-values are pooled by unweighted Logit method, the number of significant genes is 7,808. In contrast, when the raw P-values are pooled by weighted Logit method, the number of significant genes increases to 8422.

We found that the optimal weight for each dataset is proportional to the number of significant genes in the set. (see Table 10 and 11). Lapointe’s, Liu’s and Monzon’s datasets have most significant gene numbers, and they do get most weights.
With Fisher’s inverse $\chi^2$ method, the count of significant genes (8,993) is more than the counts of significant genes from other two methods (8,009 from Z, and 8,422 from Logit), because the underlying distribution is asymmetric and favors small P-values. On the other hand, the Z method yields the smallest count. The count obtained by weighted Logit method is smaller than that of Fisher’s method, but significantly more than that obtained by the weighted Z-method.
Table 11. Result of pooling one-sided P-values with $\alpha_{\text{up-regulation}} = \alpha_{\text{down-regulation}} = 0.05$

<table>
<thead>
<tr>
<th>Method</th>
<th>Vector</th>
<th>Count of significantly up-regulated genes (N1)</th>
<th>Count of significantly down-regulated genes (N2)</th>
<th>Total count of significant genes (N1+N2)</th>
<th>Lapointe</th>
<th>Varambally</th>
<th>Liu</th>
<th>Monzon</th>
<th>Singh</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma</td>
<td>Equal</td>
<td>4,508</td>
<td>4,040</td>
<td>8,548</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>4,996</td>
<td>3,997</td>
<td>8,993</td>
<td>0.232</td>
<td>0.127</td>
<td>0.237</td>
<td>0.246</td>
<td>0.059</td>
<td>0.1</td>
</tr>
<tr>
<td>Z</td>
<td>Equal</td>
<td>3,830</td>
<td>3,446</td>
<td>7,276</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>4,397</td>
<td>3,612</td>
<td>8,009</td>
<td>0.265</td>
<td>0.161</td>
<td>0.25</td>
<td>0.176</td>
<td>0.034</td>
<td>0.114</td>
</tr>
<tr>
<td>Logistic</td>
<td>Equal</td>
<td>4,114</td>
<td>3,694</td>
<td>7,808</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>4,670</td>
<td>3,752</td>
<td>8,422</td>
<td>0.283</td>
<td>0.145</td>
<td>0.248</td>
<td>0.187</td>
<td>0.011</td>
<td>0.124</td>
</tr>
</tbody>
</table>
2.6.2 Pooling result with $\text{FWER}_{\text{up-regulation}} = \text{FWER}_{\text{down-regulation}} \leq 0.01$

Using the Bonferroni to control FWER, the $\alpha$ corresponding to FWER at 0.01 level is $0.01/20482=4.88234\times10^{-7}$. So we set $\alpha_{\text{up-regulation}} = \alpha_{\text{down-regulation}} = 4.88234\times10^{-7}$. In Table 12, we illustrate that in each dataset, the count of genes whose one-sided P-values are less than $4.88234\times10^{-7}$ to compare with the count corresponding to the case of optimal weights.

Table 12. In each dataset, the count of genes whose one-sided P-values are less than $4.88234\times10^{-7}$

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Lapointe</th>
<th>Varambally</th>
<th>Liu</th>
<th>Monzon</th>
<th>Singh</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td>up-regulation</td>
<td>333</td>
<td>0</td>
<td>29</td>
<td>119</td>
<td>60</td>
<td>129</td>
</tr>
<tr>
<td>down-regulation</td>
<td>691</td>
<td>0</td>
<td>5</td>
<td>2</td>
<td>36</td>
<td>17</td>
</tr>
<tr>
<td>total</td>
<td>1,024</td>
<td>6</td>
<td>34</td>
<td>121</td>
<td>96</td>
<td>146</td>
</tr>
</tbody>
</table>

The pooling results are illustrated in Table 13. Again, for each pooling method, the weighted pooling strategy yields more significant genes than unweighted pooling strategy does. Lapointe's dataset have much more significant gene number than any other datasets (see Table 12), and it does receive the most weight (see Table 13). Weighted Fisher’s method yields most significant genes (2,253), weighted Z method yields least significant genes (1,713) and the number of significant genes by weighted Logit method is in the middle (1,882).
Table 13. Result of pooling one-sided P-values with threshold $\text{FWER}_{\text{up-regulation}} = \text{FWER}_{\text{down-regulation}} = 0.01$

<table>
<thead>
<tr>
<th>Method</th>
<th>Vector</th>
<th>Count of significantly up-regulated genes (N1)</th>
<th>Count of significantly down-regulated genes (N2)</th>
<th>Total count of significant genes (N1+N2)</th>
<th>Lapointe</th>
<th>Varambally</th>
<th>Liu</th>
<th>Monzon</th>
<th>Singh</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma</td>
<td>Equal</td>
<td>958</td>
<td>905</td>
<td>1,863</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>1,072</td>
<td>1,181</td>
<td>2,253</td>
<td>0.33</td>
<td>0.085</td>
<td>0.134</td>
<td>0.153</td>
<td>0.123</td>
<td>0.175</td>
</tr>
<tr>
<td>Z</td>
<td>Equal</td>
<td>722</td>
<td>662</td>
<td>1,384</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>784</td>
<td>929</td>
<td>1,713</td>
<td>0.321</td>
<td>0.111</td>
<td>0.174</td>
<td>0.127</td>
<td>0.129</td>
<td>0.137</td>
</tr>
<tr>
<td>Logistic</td>
<td>Equal</td>
<td>869</td>
<td>822</td>
<td>1,691</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>909</td>
<td>973</td>
<td>1,882</td>
<td>0.249</td>
<td>0.105</td>
<td>0.186</td>
<td>0.146</td>
<td>0.143</td>
<td>0.171</td>
</tr>
</tbody>
</table>
2.6.3 Pooling result with $\text{FDR}_{\text{up-regulation}} = \text{FDR}_{\text{down-regulation}} \leq 0.01$

Since the FDR is estimated from data, its value is not fixed for a given level of significance $\alpha$. Rather it is determined by the P-values, and like the P-values, the FDR may be considered a random variable. Consequently, it is not meaningful to compare the count of significant genes in a dataset by a given FDR with the counts obtained by the same FDR in other datasets. Thus, when using FDR to control for multiple testing, it is not possible to use the count of significant genes to determine the optimal weights. For this reason, pooling one-sided P-values with equal weights, we had $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$, the significance levels at which FDR for the tests for up-regulated genes ($\text{FDR}_{\text{up-regulation}}$) equals to the FDR of the tests for down-regulated genes ($\text{FDR}_{\text{down-regulation}}$). In Table 14 we show that when $\text{FDR}_{\text{up-regulation}}$ and $\text{FDR}_{\text{down-regulation}}$ are equal, the corresponding $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$ required to achieve these FDRs are not necessarily equal.
Table 14. Result of pooling one-sided P-values with equal weights when FDR = 0.01

<table>
<thead>
<tr>
<th>Method</th>
<th>$\alpha_{\text{up-regulation}}$</th>
<th>$\alpha_{\text{down-regulation}}$</th>
<th>count of significantly up-regulated genes</th>
<th>count of significantly down-regulated genes</th>
<th>Total count of significant genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma</td>
<td>0.001137</td>
<td>0.001025</td>
<td>2,229</td>
<td>2,100</td>
<td>4,329</td>
</tr>
<tr>
<td>Z</td>
<td>0.000851</td>
<td>0.000806</td>
<td>1,745</td>
<td>1,653</td>
<td>3,398</td>
</tr>
<tr>
<td>Logistic</td>
<td>0.001022</td>
<td>0.00094</td>
<td>2,094</td>
<td>1,927</td>
<td>4,021</td>
</tr>
</tbody>
</table>
Since the optimal weight vector changes as $\alpha$ changes as Hwang et al. pointed out (2005) (also see optimal weight vectors at different $\alpha$ level illustrated in Table 11 and Table 13), we have developed an algorithm to compute the effective level of significance for a fixed FDR (see Figure 12).

To address the problem of obtaining optimal weights using the FDR, suppose that we desire to allow for $\text{FDR} = \text{FDR}_{\text{up-regulation}} = \text{FDR}_{\text{down-regulation}} \leq \alpha_0$. We start with a value of significance $\alpha_{\text{up}}^{(1)}$ and $\alpha_{\text{down}}^{(1)}$, (say $\alpha_{\text{up}}^{(1)} = \alpha_{\text{down}}^{(1)} = 0.05$), and obtain optimal weights $w^{(1)}$. Based on the weighted pooled statistic $\varphi(w^{(1)}, P_{\text{up}})$, and $\varphi(w^{(1)}, P_{\text{down}})$, we calculate $\alpha_{\text{up}}^{(2)}$ and $\alpha_{\text{down}}^{(2)}$ so that $\text{FDR}_{\text{up}}^{(1)} = \text{FDR}_{\text{down}}^{(1)} \leq \alpha_0$. Using $\alpha_{\text{up}}^{(2)}$ and $\alpha_{\text{down}}^{(2)}$, we had the set of weights $w^{(2)}$ to maximize gene counts and then pooled to get statistic $\varphi(w^{(2)}, P_{\text{up}})$, and $\varphi(w^{(2)}, P_{\text{down}})$. We now use these to get $\alpha_{\text{up}}^{(3)}$ and $\alpha_{\text{down}}^{(3)}$ so that $\text{FDR}_{\text{up}}^{(2)} = \text{FDR}_{\text{down}}^{(2)} \leq \alpha_0$. So at the $j$th iteration ($2 \leq j \leq 5$), Using $\alpha_{\text{up}}^{(j)}$ and $\alpha_{\text{down}}^{(j)}$, are used to find $w^{(j)}$, the weight vector to maximize gene counts and use it to get statistic $\varphi(w^{(j)}, P_{\text{up}})$, and $\varphi(w^{(j)}, P_{\text{down}})$. These are used to get $\alpha_{\text{up}}^{(j+1)}$ and $\alpha_{\text{down}}^{(j+1)}$ such that $\text{FDR}_{\text{up}}^{(j)} = \text{FDR}_{\text{down}}^{(j)} \leq \alpha_0$. When $|\alpha_{\text{up}}^{(j+1)} - \alpha_{\text{up}}^{(j)}| / \alpha_{\text{up}}^{(j)} < 0.05$ and $|\alpha_{\text{down}}^{(j+1)} - \alpha_{\text{down}}^{(j)}| / \alpha_{\text{down}}^{(j)} < 0.05$, we declare that the proceeding has converged and set $\alpha_{\text{up-regulation}} = \alpha_{\text{up}}^{(j+1)}$ and $\alpha_{\text{down-regulation}} = \alpha_{\text{down}}^{(j+1)}$. If the $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$ have not both converged after five rounds, we will stop the process and alert that this FDR threshold is not applicable to the input data.
Figure 12. One-sided P-value thresholds converging process
The whole $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$ converging process at FDR = 0.01 level is summarized in Table 15. We take Fisher’s inverse $\chi^2$ method as example to explain the whole $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$ converging process. In the first round, we set a very loose criterion $\alpha_{\text{up-regulation}} = \alpha_{\text{down-regulation}} = 0.05$. After optimization, we calculate FDR$_{\text{up-regulation}}$ based on pooled up-regulation P-values, and find the $\alpha_{\text{up-regulation}}$ corresponding to FDR$_{\text{up-regulation}}$ at 0.01 level is 0.0012. It serves as input $\alpha_{\text{up-regulation}}$ in next round. Similarly, we find $\alpha_{\text{down-regulation}}$ for next round is 0.00105.

In the second round, the input $\alpha_{\text{up-regulation}}$ is 0.0012 and $\alpha_{\text{down-regulation}}$ is 0.00105. After optimization, we calculate new FDR$_{\text{up-regulation}}$ based on pooled up-regulation P-values in this round, and find the new $\alpha_{\text{up-regulation}}$ corresponding to FDR$_{\text{up-regulation}}$ at 0.01 level is 0.00125. Similarly, we find the new $\alpha_{\text{down-regulation}}$ corresponding to FDR$_{\text{down-regulation}}$ at 0.01 level is 0.00114. The difference between input $\alpha_{\text{up-regulation}}$ (0.0012) and new $\alpha_{\text{up-regulation}}$ (0.00125) is less than 5%, but the difference between input $\alpha_{\text{down-regulation}}$ (0.00105) and new $\alpha_{\text{down-regulation}}$ (0.00114) is more than 5%. So it is necessary to go through the second round.

In the third round, after optimization, we repeated the same process. The difference between input $\alpha_{\text{up-regulation}}$ (0.00125) and new $\alpha_{\text{up-regulation}}$ generated in this round (0.00126) is less than 5%, and the difference between input $\alpha_{\text{down-regulation}}$ (0.00114) and new $\alpha_{\text{down-regulation}}$ generated in this round (0.00114) is also less than 5%. Since both $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$ have converged, so the whole process stops. For weighed Z method and weighted Logit method, both $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$ also converge after the third round.
Table 15. $\alpha_{\text{up-regulation}}$ and $\alpha_{\text{down-regulation}}$ and weight vector required to get optimal weighted procedure at FDR < 0.01 level

<table>
<thead>
<tr>
<th>Method</th>
<th>Round</th>
<th>$\alpha_{\text{up-regulation}}$</th>
<th>$\alpha_{\text{down-regulation}}$</th>
<th>Lapointe</th>
<th>Varambally</th>
<th>Liu</th>
<th>Monzon</th>
<th>Singh</th>
<th>Welsh</th>
<th>$\alpha_{\text{up-regulation}}$ for next round</th>
<th>$\alpha_{\text{down-regulation}}$ for next round</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma</td>
<td>1</td>
<td>0.05</td>
<td>0.05</td>
<td>0.232</td>
<td>0.127</td>
<td>0.238</td>
<td>0.247</td>
<td>0.056</td>
<td>0.1</td>
<td>0.0012</td>
<td>0.00105</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.0012</td>
<td>0.00105</td>
<td>0.271</td>
<td>0.095</td>
<td>0.157</td>
<td>0.196</td>
<td>0.119</td>
<td>0.162</td>
<td>0.00125</td>
<td>0.00114</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.00125</td>
<td>0.00114</td>
<td>0.272</td>
<td>0.108</td>
<td>0.154</td>
<td>0.217</td>
<td>0.117</td>
<td>0.132</td>
<td>0.00126</td>
<td>0.00114</td>
</tr>
<tr>
<td>Z</td>
<td>1</td>
<td>0.05</td>
<td>0.05</td>
<td>0.2654</td>
<td>0.161</td>
<td>0.25</td>
<td>0.176</td>
<td>0.0337</td>
<td>0.114</td>
<td>0.000953</td>
<td>0.000872</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.000953</td>
<td>0.000872</td>
<td>0.343</td>
<td>0.11</td>
<td>0.183</td>
<td>0.12</td>
<td>0.097</td>
<td>0.145</td>
<td>0.000972</td>
<td>0.000976</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.000972</td>
<td>0.000976</td>
<td>0.351</td>
<td>0.099</td>
<td>0.178</td>
<td>0.126</td>
<td>0.083</td>
<td>0.162</td>
<td>0.000977</td>
<td>0.00097</td>
</tr>
<tr>
<td>Logistic</td>
<td>1</td>
<td>0.05</td>
<td>0.05</td>
<td>0.283</td>
<td>0.145</td>
<td>0.248</td>
<td>0.187</td>
<td>0.114</td>
<td>0.124</td>
<td>0.00108</td>
<td>0.00078</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>0.00108</td>
<td>0.00078</td>
<td>0.262</td>
<td>0.13</td>
<td>0.162</td>
<td>0.158</td>
<td>0.116</td>
<td>0.171</td>
<td>0.001085</td>
<td>0.00104</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>0.001085</td>
<td>0.00104</td>
<td>0.259</td>
<td>0.104</td>
<td>0.178</td>
<td>0.194</td>
<td>0.112</td>
<td>0.153</td>
<td>0.00111</td>
<td>0.001035</td>
</tr>
</tbody>
</table>
Table 16 summarizes the result of implementing algorithm using the weighted Fisher’s inverse chi squared method, Z method and Logit method. For each method, more significant genes are identified by weighted pooling method than by equal weight pooling method (see Table 14 and 16).

Table 16. Comparison of the weighted combination methods by gene counts when

\[
\text{FDR}_{\text{up-regulation}} = \text{FDR}_{\text{down-regulation}} \leq 0.01
\]

<table>
<thead>
<tr>
<th>Method</th>
<th>( \alpha_{\text{up-regulation}} )</th>
<th>( \alpha_{\text{down-regulation}} )</th>
<th>count of significantly up-regulated genes</th>
<th>count of significantly down-regulated genes</th>
<th>Total count of significant genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gamma</td>
<td>0.001256</td>
<td>0.001141</td>
<td>2,583</td>
<td>2,337</td>
<td>4,920</td>
</tr>
<tr>
<td>Z</td>
<td>0.000977</td>
<td>0.00097</td>
<td>2,002</td>
<td>2,000</td>
<td>4,002</td>
</tr>
<tr>
<td>Logistic</td>
<td>0.00111</td>
<td>0.001035</td>
<td>2,275</td>
<td>2,131</td>
<td>4,406</td>
</tr>
</tbody>
</table>

2.7 Conclusion and discussion

In this chapter, we have proposed a correction to the optimally weighted combination procedure upon which Pointillist algorithm (Hwang et al., 2005) is built. Our procedure is to replace simulation of null distributions of combination weights by approximation of the exact distributions and correcting statistical errors in the Pointillist algorithm. We also developed an algorithm for computing optimal weights for weighted pooling of one-sided P-values when controlling for multiple testing by FDR.

In our test of gene expression experiment data, the weighted pooling methods consistently yield more significant genes than the equal weight pooling method. It’s useful tool to extract more information whenever the multiple datasets are available.
Although we have focused on microarray data, our procedure is applicable to other experiments in which high dimensional data are collected from several studies. Also we feel that our method can be extended to pooling dependent datasets.
CHAPTER 3 OPTIMALLY WEIGHTED COMBINATION PROCEDURE TO POOL GENE EXPRESSION DATA BY MAXIMIZING FUNCTIONAL COHERENCE OF TOP RANKED GENES

3.1 Introduction

In chapter 2, we proposed and illustrated the use of optimal weighted combination procedures for combining P-values from several independent microarray data. The criterion for optimality in the pooling method is to maximize the number of significant genes identified. In this chapter, we optimize weights through the use of gene set functional coherence of top rank genes. The goal is to find an ‘optimal state’ in which the top ranked genes ordered by the P-values of their weighted combination statistics have maximal gene set functional coherence.

In a typical microarray study, gene specific statistical tests are performed and genes are ordered according to the strength of significance of their differential expression, as indicated by the P-values of the test statistic. Following this, an investigator will focus on the top ranked genes to establish gene functional relationship / network, biological pathway, or other microbiologically ramifications of the gene’s selection. However, such lists of top ranked genes are not always stable. A list can be heavily affected by many factors such as the microarray platform design, sample size, experiment quality, and the procedures for data processing (e.g. normalization method, feature selection method). The application of different feature selection methods on the same dataset may yield different top ranked genes. Jeffery et al. (2006) tested 10 popular feature selection methods and found very little overlap in a list of top 100 genes produced by each method. Only 21% of genes were common to all 10 lists. Agreement in the lists of the top ranked
genes from different studies is even rarer. For example, using the six prostate tumor datasets discussed in the previous chapters of this dissertation, we ordered genes according to the strength of the significance of their two-sided P-values based t-test, and compared the lists of the top 100 genes from each dataset. Surprisingly, only one gene (GeneID: 3249) was common to all six lists. Only three genes (GeneID: 5831, 10606, and 65108) were present in five lists. Such a lack of consensuses in lists of top ranked genes motivates the desire to evaluate genes based on biological significance of different gene sets. Therefore, we did the evaluation of gene sets by Latent Semantic Indexing Cohesive (LSIC) Analysis.

Latent Semantic Indexing (LSI) is a statistical procedure that represents terms, documents, abstracts etc, as elements of a high dimensional vector space by using singular value decomposition and may be regarded as an optimal version of multidimensional scaling (Bartell et al., 1992). Homayouni et al. (2005) utilized LSI, to discover conceptual relationships among genes based on titles and abstracts in MEDLINE citations, and illustrated the robustness of the method for identifying both explicit and implicit gene relationships by mining biomedical literature.

The implementation of LSI is well illustrated by Berry and Browne (1999). In that work, terms and documents are represented as vectors, and a term-by-document matrix $M$ with $n$ documents and $m$ unique terms were created. This matrix can be defined as:

$$M = [a_{ij}],$$

$$a_{ij} = l_{ij}g_id_j,$$

and where $l_{ij}$ is the local weight for term $i$ appearing in document $j$, $g_i$ is the global weight for term $i$ across the collection, $d_j$ is the normalization factor for document $j$. 

69
Let $f_{ij}$ the frequency of term $i$ occurring in document $j$ and $p_{ij} = f_{ij} / \sum_i f_{ij}$,

$$b = \chi(f_{ij}),$$

$$l_{ij} = \log(1 + f_{ij}),$$

$$n = (\chi(f_{ij}) + (f_{ij} / \max_k f_{kj}))/2,$$

$$t = f_{ij}.$$  

where $n$ is the accumulated normalized term frequency and $t$ is the term frequency.

Let

$$\chi(s) = \begin{cases} 1 & \text{if } s > 0, \\ 0 & \text{if } s = 0, \end{cases}$$

Then an appropriate weighting scheme can be defined as:

$$a_{ij} = \frac{\log(f_{ij} + 1) \log(n / \sum_j \chi(f_{ij}))}{\sqrt{\sum_i \left( \log(f_{ij} + 1) \log(n / \sum_j \chi(f_{ij})) \right)^2}}$$

The expression for local term weights ($g_i$) are given by:

$$g_i = 1 + (\sum_j [p_{ij} \log_2(p_{ij})]) / \log_2 n,$$

$$f = \log(n / \sum_j \chi(f_{ij})), $$

$$g = (\sum_i f_{ij}) / \sum_j \chi(f_{ij}).$$

$$n = 1 / \sqrt{\sum_i f_{ij}^2},$$

$$p = \log((n - \sum_j \chi(f_{ij}))/\sum_j \chi(f_{ij})).$$

Here, $f$ is the inverse document frequency (IDF), $g$ is global frequency inverse document frequency (GfIdf) and $p$ is the probability inverse.
In general $m$ and $n$ are quite large and matrix $M$ can be very sparse. Each cell of matrix $M$ contains the frequency of a term in a document (Figure 13A). After constructing the matrix, each entry in this matrix is normalized to weighted frequencies which are the product of local and global weight (Figure 13B). Using the term-by-document matrix $M$, a truncated SVD of that matrix is performed to create three factor matrices:

$$M = X\Sigma Y^T$$

where $X$ represents the $m \times b$ matrix of eigenvectors of $MM^T$, $Y^T$ represents the $b \times n$ matrix of eigenvectors of $M^TM$. $\Sigma$ is the $b \times b$ diagonal matrix, with $b$ non-negative singular values $\sigma_1 \geq \sigma_2 \cdots \geq \sigma_{\text{min}(m,n)}$ of $M$ by the order of the diagonal (Golub and Loan 1996). The first $c$ columns of them are then used to create one new matrix $M_c$ where $c$ is much smaller than $b$ (Figure 13C):

$$M_c = X\tilde{\Sigma}Y^T.$$

Here $\tilde{\Sigma}$ contains only the top $k$ largest singular values and all others are set to zero. This new matrix $M_c$ is the closest approximation to matrix $M$ according to the Eckart–Young theorem (Eckart & Young, 1936) which gives the minimal difference between $M$ and $M_c$ as

$$\|M - M_c\|_F = \min_{\text{rank}(N) \leq k} \|M - N\|_F = \sqrt{\sigma_{k+1}^2 + \cdots + \sigma_{r_A}^2},$$

where $r_A$ is the number of nonzero diagonal elements of $R$.

We can compare a given keyword vector $d$ with the document vectors using
\[ d_j = \sum y^T e_j \]

where \( e_j \) is a vector consisting of zero elements except for 1 in position \( j \). The document vectors’ relationship to the keyword query can be evaluated by a similarity score, defined as the cosine value of the two document vectors (Figure 14D) (Berry & Browne, 1999). The similarity scores can be calculated as

\[
\cos \theta_j = \frac{d_j^T (X^T q)}{\|d_j\| \|q\|}, \quad j = 1, \ldots, n.
\]

Here, \( q \) (treated as pseudo-documents) can be calculated as:

\[ q = q_0^T X_c \Sigma^{-1}. \]

\[
\cos \theta_j = \frac{(A_k e_j)^T q}{\|A_k e_j\| \|q\|} = \frac{(X_k \Sigma_k Y_k^T e_j)^T q}{\|X_k \Sigma_k Y_k^T e_j\| \|q\|} = \frac{e_j^T Y_k \Sigma_k (X_k^T q)}{\|\Sigma_k Y_k^T e_j\| \|q\|}, \quad \text{for } j = 1, 2, \ldots, n.
\]

The entries of matrix \( M_c \) are components that reflect the associations of the terms with the corresponding document. The components in one document can be regarded as a concept derived from the word usage patterns in the document, so the similarity of documents is measured on conceptual level. Therefore, LSI can be used to find not only explicit (literal match) but also implicit (similar word usage pattern) relationships between relevant documents which could be collected from MEDLINE citations.

MEDLINE is the premier bibliographic database for biomedicine, supported by the National Library of Medicine. It contains greater than 19 million references, most of which have abstracts. Besides, MEDLINE covers over 4,800 journals, in over 30 languages. The citations of MEDLINE can date back to 1966. In Homayouni’s study
(2005), genes were treated as documents. Each gene-document was generated by concatenation of all titles and abstracts of the MEDLINE citations cross-referenced in the mouse, rat and human. After the term-by-gene document matrix was built, Latent Semantic Indexing (LSI) was utilized to extract both explicit and implicit gene relationships from the literature. Then the cosine of the angle between two gene-document vectors is calculated to construct the gene-to-gene similarity matrix.

Figure 13. The work flow for calculating the similarity between documents with LSI model.

(A) Each document is represented by a corpus of terms and frequencies of the terms are calculated. (B) A term-by-document matrix was created that contains the weighted
frequencies of each term. (C) A truncated SVD of that matrix is performed to create one lower dimension matrix. (D) Similarity between documents is derived from the cosine angle between vectors.

Xu et al. (2011) have developed a statistical method for literature semantic indexing cohesive (LSI-c) analysis, to evaluate the functional coherence of gene sets using LSI derived gene-to-gene similarities. Within the given gene set, the total number of the gene-to-gene similarities above the pre-determined similarity threshold is counted, and then Fisher’s exact test is performed to test if the observed number of similarities above the threshold is significantly different from what would be expected with a random set of genes. They conducted a large scale evaluation of their method in many known different functions in Gene Ontology (GO), and found that most of the functional gene sets in GO have statistically significant literature support, indicating that their method provides a very robust statistical way to evaluate the functional cohesiveness of a gene set. They applied the literature semantic indexing cohesive (LSI-c) analysis to evaluate the gene sets generated by different algorithms so as to help biologists in selecting appropriate algorithm that yields the most biologically relevant gene set.

Since the top ranked genes play important role in downstream analysis, it is desirable to find the list of top rank genes with maximal biological relevance. In this chapter, we propose a procedure based on an enhanced simulated annealing (ESA) and literature semantic indexing cohesive (LSI-c) analysis to assign optimal weights to datasets so as to maximize the functional coherence of the top rank genes. In the optimization process, in each state, all genes are ordered by their P-values. Then we apply the LSC-c analysis to the list of top rank genes. Its functional coherence will serve as the energy of the state.
The state in which the top genes have maximal gene functional coherence will be the optimal state.

3.2 Method

In Xu’s original LSI-c analysis (2011), they performed Fisher’s exact test to evaluate the statistical significance of functional coherence of the given gene set. In this dissertation, the genes in a given gene set are considered fully connected. The connections between genes whose functional similarity score is above a certain threshold are called significant connection. The count of significant connections reflects the functional coherence of the given gene set. The more significant connections within one gene set, the more functional coherence this gene set is considered to have. Xu et al performed Fisher’s exact test by comparing the observed significant connection and insignificant connection from the given gene set with the expected ones from the random gene set of the same size. In our work, our purpose is to compare the functional coherence of many gene sets (one gene set from one state) in order to find the gene set with highest functional coherence, rather than evaluating the statistical significance of functional coherence of one particular gene set. In addition, for fair comparison purpose, in each state, the number of top rank genes selected for comparing functional coherence is fixed (for example, always select top 100 or top 500 genes). Hence, we can reduce the work to simply compare the count of significant connections from the fixed number of top rank genes in each state. In the optimal state, the top ranked genes should have more significant connections than in any other states.

The human gene functional similarity matrix contains 17,451 genes (Courtesy of Dr. Ramin Homayouni). We examined the similarity scores of all gene pairs (17451x17450/2)
and found out that the score of 0.6 is the threshold for top 5% gene pair connections. We use this number as similarity score threshold in our work. The connection between gene pair whose similarity score is larger than this threshold is considered significant connection in our study. During the optimization process, in each state, at first we rank order all input genes. Then from the top, we take the first N genes that also exist in human gene functional similarity matrix (called top N genes in this dissertation), and count the number of significant connections within these genes. For the 20,482 genes in our prostate datasets, there are 14,112 genes overlapping with the genes in the human gene functional similarity matrix. They are the genes actually being used to evaluate the functional coherence of top rank genes.

In the gene functional similarity matrix, the similarity scores of gene pairs are always positive values no matter the gene pairs are positive or negative correlated. In other words, the gene similarity scores do not reflect regulation direction. So in each state of the optimization process, all highly differentially expressed genes must get high rank, no matter whether they are under-regulated or up-regulated. This could be achieved by assigning each gene one of its pooled one-sided P-values whichever is close to 0, in other words, min(pooled up-regulation P-value, pooled down-regulation P-value). Then sort genes by the assigned pooled P-values in ascending order.
**Figure 14.** Procedure to evaluate functional coherence of one state.

1. **Raw up-regulation P-values**
2. **Weight vector**
3. **Raw down-regulation P-values**

- Transform up-regulation P-values to statistics, and weighted pool them.
- Derive the theoretical distribution.
- Transform down-regulation P-values to statistics, and weighted pool them.

- **Pooled up-regulation P-value**
- **Pooled down-regulation P-value**

- For each gene, assign min (pooled up-regulation P-value, pooled down-regulation P-value)

- Sort genes by the assigned P-value in ascending order

- N, Sorted genes, Gene functional similarity matrix

- From the top, take the first N genes which are also in gene functional similarity matrix

- Gene functional similarity threshold, Top N genes, Gene functional similarity matrix

- Count the significant connections among the top N genes
Our procedure of optimization by gene functional coherence is illustrated in Figure 14. The input consists of up-regulation P-value matrix and down-regulation P-value Matrix. In each state, we apply the current weight vector to each matrix and calculate the pooled weighted up-regulation P-value and the pooled weighted down-regulation P-value separately for each gene. Each gene is then assigned with min(pooled up-regulation P-value, pooled down-regulation P-value). The genes are then sorted by their assigned pooled P-values in ascending order. The highly differentially expressed genes are at the top. Count the significant connections among the top ranked N genes. Its negative value serves as the “energy” of the state.

3.3 Result

In this chapter, we only present the results from the raw P-values which are pooled by Mudholkar-George’s Logit method. The results generated by pooling P-values with Fisher’s weighted inverse $\chi^2$ method or Liptak-Stouffer’s weighted Z-method are very similar to the result produced from Mudholkar-George’s Logit method.

3.3.1 Optimization result by maximizing functional coherence of top N genes

We count the significant connections within top N genes in each dataset to compare with the optimal weights. For each dataset and each gene, we first assign min(raw up-regulation P-value, pooled down-regulation P-value) to it, then sort genes in ascending order. Finally we count the significant connections within top N genes. The result is illustrated in Table 17.
Table 17. Number of significant connections within top N genes in each dataset

<table>
<thead>
<tr>
<th>Top N genes</th>
<th>Lapointe</th>
<th>Varambally</th>
<th>Liu</th>
<th>Monzon</th>
<th>Singh</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>402</td>
<td>386</td>
<td>667</td>
<td>720</td>
<td>871</td>
<td>649</td>
</tr>
<tr>
<td>500</td>
<td>9,755</td>
<td>8,482</td>
<td>9,082</td>
<td>10,713</td>
<td>14,629</td>
<td>14,082</td>
</tr>
</tbody>
</table>

We then optimize the weight vector by comparing the counts of significant connections within the top N genes of each state. The result is in Table 18, where we also list the count of significant connections within the top N genes when we assign equal weight to each dataset for comparison.

Table 18. Weight assigned to each dataset and count of significant connections in top N genes

<table>
<thead>
<tr>
<th>N</th>
<th>weight</th>
<th>connections</th>
<th>Lapointe</th>
<th>Varambally</th>
<th>Liu</th>
<th>Monzon</th>
<th>Singh</th>
<th>Welsh</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>equal</td>
<td>625</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
<tr>
<td></td>
<td>optimal</td>
<td>1,134</td>
<td>0.01</td>
<td>0.17</td>
<td>0.084</td>
<td>0.389</td>
<td>0.494</td>
<td>0.007</td>
</tr>
<tr>
<td>500</td>
<td>equal</td>
<td>11,987</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
<td>1/6</td>
</tr>
<tr>
<td></td>
<td>optimal</td>
<td>18,041</td>
<td>0.0019</td>
<td>0.157</td>
<td>0.025</td>
<td>0.433</td>
<td>0.369</td>
<td>0.0147</td>
</tr>
</tbody>
</table>

When we assign equal weights to datasets, the functional coherence of the combined evidence is not an improvement over that of the single dataset with highest functional coherence (Singh dataset). This is different from the results from measuring statistical power, in which case the statistical power of unweighted pooling procedure is better than any single datasets. When we use optimal weights, determined by functional coherence, the functional coherence of the combined evidence is always significantly higher than that of the single dataset with highest functional coherence (Singh dataset). Hence,
choosing weights to optimize functional coherence is a biological meaningful procedure for pooling datasets.

3.3.2 Comparison of gene sets obtained by the two optimization procedures

The use of the two criteria, gene counts and functional coherence, to calculate optimal weights for pooling datasets, can clearly be expected to produce different gene sets. How different the sets are? To answer this question, we compare the results of optimization by functional coherence of top N genes with the result of optimization by count of significant genes with Logit method.

Table 19 illustrates a comparison of the count of significant genes using these two optimization methods. It shows that the use of gene set functional coherence produces smaller number of significant genes than optimized pooling by count of significant genes. For instance, in using functional coherence of top 500 genes, there are 2,908 genes whose up-regulation pooled P-values are less than 0.05, and there are 2,582 genes whose down-regulation pooled P-values are less than 0.05. In contrast, optimizing weights by count of genes whose pooled P-values are less than 0.05 produces 4,670 up-regulated genes and 3,753 down-regulated genes whose pooled P-values meet this threshold.

Table 19. Count of significant genes in the results generated by two optimization methods

<table>
<thead>
<tr>
<th>optimization method</th>
<th>P &lt; 0.05</th>
<th>FWER &lt; 0.01</th>
<th>FDR &lt; 0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>By functional coherence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Top 100</td>
<td>5,246</td>
<td>234</td>
<td>1,178</td>
</tr>
<tr>
<td>Top 500</td>
<td>5,490</td>
<td>260</td>
<td>1,241</td>
</tr>
<tr>
<td>By count of significant genes</td>
<td>8,423</td>
<td>1,882</td>
<td>4,406</td>
</tr>
</tbody>
</table>

In Table 20, we summarize the results of two optimization methods in terms of the count of significant connections within top N genes. We found that optimization by count of
significant genes produces fewer significant connections within top N genes than by gene set functional coherence. For instance, with optimization by count of significant genes (P < 0.05), there are 488 significant gene connections within the top 100 genes and 11,165 significant connections within the top 500 genes. In contrast, optimizing by gene set functional coherence of top 500 genes produces 1,030 significant gene connections within top 100 genes and 18,041 significant connections within top 500 genes.
Table 20. Count of significant connections within top N genes in the results generated by two optimization methods

<table>
<thead>
<tr>
<th>Optimization method</th>
<th>Significant Level</th>
<th>In top 100 genes</th>
<th>In top 500 genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>By count of significant genes</td>
<td>P &lt; 0.05</td>
<td>488</td>
<td>11,165</td>
</tr>
<tr>
<td></td>
<td>FWER &lt; 0.01</td>
<td>573</td>
<td>11,254</td>
</tr>
<tr>
<td></td>
<td>FDR &lt; 0.01</td>
<td>530</td>
<td>11,365</td>
</tr>
<tr>
<td>By functional coherence</td>
<td>Top 100 genes</td>
<td>1,134</td>
<td>13,971</td>
</tr>
<tr>
<td></td>
<td>Top 500 genes</td>
<td>1,030</td>
<td>18,041</td>
</tr>
</tbody>
</table>

While it is clear that these two procedures are substantively different, the results of the comparisons are consistent with the optimization criteria. Optimizing weighted pooling by gene count does not necessarily incorporate connections within top ranked gene while the coherence criterion does not depend on the number of genes, but connection of genes within the targeted top N genes.

3.4 Conclusion and discussion

We have proposed an approach to select weights for pooling P-values from datasets so that the functional coherence of the top rank genes is maximized. The use of functional coherence with top ranked genes for estimating optimal weight vector produced genes that have much more significant connections than the single dataset that has highest functional coherence. The top ranked genes obtained by this optimization method have more biological information in terms of gene function and other related biological properties. We also showed that optimization of weights by count of significant genes produced a different gene set than is obtained through optimization by maximizing functional coherence with top ranked genes. A future research plan is to develop an
approach that integrates these two methods using a Bayesian procedure to calculate posterior probability of gene’s significance with prior distributions elicited from information about gene-to-gene functional similarity.
CHAPTER 4 WEB TOOL

4.1 Introduction

We have built a web tool, POOLHIT, to implement the weighted pooling methods described in the previous chapters. The users will not need to install any languages/packages. The user will only need to upload two matrices and set a few parameters. Results will be returned within minutes. Beginning from December 10, 2011, the web tool can be accessed at 128.169.4.89 (Figure 15).

4.2 User interface

4.2.1 Input data type

Although one-sided P-values should be used to pool gene expression datasets, we have created an option for pooling two-sided P-values, in when only two-sided P-values are available or for comparison purpose (Figure 16). In the rest of this chapter, we illustrate the web tool with one-sided P-value sample. The interface for pooling two-sided P-values is very similar.

4.2.2 Input data format

We provide the prostate experiment datasets used in the previous chapters as the sample data (Figure 17). The input data should be two matrices. One contains the raw up-regulation P-values, and the other contains the raw down-regulation P-values. Each matrix should be saved as a text file in tab delimited format. In each file, the first row contains column header, the remaining rows contains one-sided P-values for the genes. The first column contains GeneIDs, and each of the remaining columns contains the P-values with one column per experiment. The columns and rows in up-regulation P-value
file and down-regulation P-value file must be in the same order. Missing values should be set as 0.5 to avoid bias.

We provide three options for computing weights: (a) equal weight (b) optimal weights by significant genes counts and (c) optimal weights by gene set functional coherence. Here we introduce the interface of the optimal weights procedures.

4.2.3 Interface for optimization by significant genes count.

The files containing the raw $P_{\text{up-regulation}}$ values and the $P_{\text{down-regulation}}$ values should be uploaded (Figure 18). The user can choose different pooling statistics (Fisher’s, Logit, Z), error control (Bonferroni FWER, BH 95 FDR) and significance levels (0.01, 0.05). After submission, the user will get the optimal weight vector and list of significant genes (Figure 19). Each gene is shown with both of its optimally pooled one-sided P-values, along with the corresponding Bonferroni FWER and BH 95 FDR. The text format result of all input genes is also available (Figure 20).

4.2.4 Interface for optimization by gene set functional coherence.

The files containing the raw $P_{\text{up-regulation}}$ values and $P_{\text{down-regulation}}$ values should be uploaded. The user can set the parameter of pooling statistics (Fisher’s, Logit, Z), N, the number of top ranked genes for evaluating functional coherence (Figure 21). The optimal weight vector and list of top rank genes will be returned (Figure 22). The text format result of all input genes and the list of significant connections within the top ranked genes will also be available (Figure 23).
4.3 Architecture

The web service is implemented using Apache HTTP server, Python, Matlab, R and Java (Figure 24). Python is used to handle HTTP request, pass parameters, wrap Matlab and R scripts, and return dynamically built dynamic webpage. The R multtest package is used to perform multiple hypothesis testing. Matlab scripts and Java programs are written to pool raw one-sided P-values and evaluate functional coherence. One Matlab script (esa.m) in Pointillist (v2.1.2) is used to perform enhanced simulated annealing.

4.4 Performance

The web service is currently hosted on a Dell PowerEdge 860 server (2.66 GHz CPU, 8 GB memory). A sample program using the six sample prostate datasets takes 1 to 15 minutes computation time depending on the optimization method and parameters selected.

4.5 License and availability

The source code of this work is under the GNU Lesser General Public License. It is available upon request.
Figure 15. Web tool main page.

Figure 16. Input P-values
Figure 17. Input file format

Figure 18. Input interface for optimization by count of significant genes
**Figure 19.** Output interface for optimization by significant genes count

**Figure 20.** Text format pooling result of all genes for optimization by significant genes count
Figure 21. Input interface for optimization by gene set functional coherence

Figure 22. Output interface for optimization by gene set functional coherence
**Figure 23.** Significant connections within top ranked genes for optimization by gene set functional coherence

<table>
<thead>
<tr>
<th>Gene ID</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>32101</td>
<td>SPARC</td>
<td>Tissue factor 1</td>
</tr>
<tr>
<td>32102</td>
<td>IL1B</td>
<td>Interleukin 1</td>
</tr>
<tr>
<td>32103</td>
<td>TNF</td>
<td>Tumor necrosis factor</td>
</tr>
</tbody>
</table>

**Figure 24.** Software architecture

- **Apache web server**
- **Python scripts to handle web-related business, and wrap R, Matlab scripts**
- **R script to perform multiple hypothesis testing**
- **Matlab scripts for optimization, and pooling raw one-sided P-values**
- **Java programs to evaluate functional coherence**
CHAPTER 5 CONCLUSIONS AND FUTURE WORK

5.1 Accomplishments

In this dissertation, we have developed two meta-analytic procedures for optimal weighted pooling of high throughput biological data. The first procedure computes optimal weights to pool P-values so as to maximize the number of differentially expressed genes identified. The second procedure uses a biological perspective in which optimal weights are computed so as to optimize the functional coherence of the top ranked genes. Finally, we provide a free web tool implementing our procedures.

We show that our procedure has better operating characteristics than a commonly used procedure called Pointillist.

5.2 Future Directions

The approximation of the distributions in this dissertation is under the assumption the significance tests are independent. While this is true when combining evidence from multiple independent studies, it cannot be applied to pool data from correlated studies. In such case, the approximation of the distributions and the selection of weights would have to take into account the dependency between datasets. Hou (2005) presented the approximation for the distribution of weighted correlated inverse chi-square statistics. The approximation for the distribution of weighted correlated Z-score or logistic statistics have not been studied. In future, we will study the approximation for these distributions and extend our procedure to correlated studies so that our procedure can be applied in broader range.
In the evaluation of pooled data by the functional coherence of top ranked genes, we sought to optimize the biological information of the top ranked genes. Gene set functional coherence analysis is based on the functional similarity of all known genes. In some areas of biology research, domain-specific databases are available. For instance, KEGG PATHWAY is a collection of pathway maps of known molecular interaction and reaction networks for metabolism, environmental information processing, cell processing etc. Cancer Gene Index is a collection of 6,995 human genes identified from literature having association with cancer. In these areas of research, genes in domain-specific databases might provide more accurate evaluation of pooled evidence. Genes in these databases can serve as “benchmark genes”, and Gene Set Enrichment Analysis (Subramanian et al., 2005) can be performed to evaluate the extent of enrichment of “benchmark genes”, represented by Enrichment Score (ES), in pooled evidence. We could define optimality in terms of maximal enrichment of “benchmark genes”.

In this dissertation, we have developed procedures to optimally pool evidence from both purely statistical perspective and a combination of biological perspective. It is desirable to analyze gene expression data using a combination of the biological information to maximize gene counts. Xu et al. (2010) have built a Bayesian model which incorporates literature information into the analysis of microarray data. In their study, a prior distribution based on gene relationships derived from the biomedical literature using Latent Semantic Indexing was used to identify differentially expressed genes. They showed that this procedure increased the statistical power while producing more biological significant genes. Further work is needed to develop an optimal pooling procedure in this context of gene expression data analysis.
REFERENCES


