DISCRETE METHODS FOR MICROARRAY ANALYSIS

By

Humberto Ortiz-Zuazaga

A thesis submitted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER ENGINEERING

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS

May, 2008

Approved by:

Dorothy Bollman, Ph.D
Member, Graduate Committee
Date

Carlos Corrada Bravo, Ph.D
Member, Graduate Committee
Date

Luis Pericchi, Ph.D
Member, Graduate Committee
Date

Jaime Seguel, Ph.D
Member, Graduate Committee
Date

Oscar Moreno, Ph.D
President, Graduate Committee
Date

Jose E. García Arrarás, Ph.D
Representative, Graduate Studies
Date

Nestor Rodriguez, Ph.D
Chairperson of the Department
Date

Abstract of Dissertation Presented to the Graduate School
of the University of Puerto Rico in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

DISCRETE METHODS FOR MICROARRAY ANALYSIS

By

Humberto Ortiz-Zuazaga

May 2008

Chair: Oscar Moreno
Major Department: Computer and Information Sciences and Engineering

Microarrays allow researchers to simultaneously measure the expression of thousands of genes. They give invaluable insight into the transcriptional state of biological systems, and can be important in understanding physiological as well as diseased conditions. However, the analysis of data from many thousands of genes, from only a few replications is very difficult.

We have devised a novel method of correcting errors in microarray experiments, that also clusters genes into groups, and categorizes their measurements into coarse divisions, suitable for discrete techniques for reverse engineering. These techniques are based on finite fields and algebraic coding theory. We test these new techniques on a data set obtained from behavioral training experiments on rats, and identify two novel genes that may be involved in learning and memory.

We extend this method to work with “probe level” microarray data, where each gene is represented by multiple probes. We have applied the error correction procedure to two data sets, one Affymetrix, one NimbleGen, having either 14 (Affymetrix) or approximately 10 (NimbleNen) probes per gene, derived from an odor avoidance experiment on *Drosophila.* The experiment is designed to validate analysis procedures by examining the degree of concordance the procedures produce across the data sets.

For this data we devise a method to measure the concordance quantitatively. We have developed a technique based on mutual information to compare results obtained across the two data sets. Our results show that our error correction techniques result in a greater amount of shared information between data sets than traditional approaches based on averaging of probes and gene expression levels across repetitions.

We show how our results can be extended to sets with finer gradations in expression values, and present the analysis of the *Drosophila* data discretized to 5 separate expression values. Finally, we present some future applications, such as using finite fields to encode expression values, allowing us to use the algebraic properties of finite fields to perform reverse engineering of gene regulatory networks.

Resumen de Disertación Presentado a la Escuela Graduada
de la Universidad de Puerto Rico Como Requisito Parcial de los
Requerimientos para el Grado de Doctorado en Filosofía

METODOS DISCRETOS PARA EL ANALISIS DE MICROARREGLOS

Por

Humberto Ortiz-Zuazaga

Mayo 2008

Consejero: Oscar Moreno
Departamento: Ciencias e Ingeniería de la Información y la Computación

Microarreglos de material genetico permiten medir niveles de expresión de miles de genes en un solo experimento. Presentan un cuadro de el estado transcripcional de una muestra biológica, y pueden ser de gran valor en elucidar mecanismos de acción de procesos fisiológicos o patológicos. El análisis de datos de estos experimentos, sin embargo, se hace difícil por la gran cantidad de genes medidos, y la carencia de replicados.

Hemos desarollado un método novedoso de analizar estos datos. Nuestra técnica agrupa genes en categorías gruesas, permite corregir errores experimentales, y sirve para producir datos discretos de expresión para luego utilizar técnicas discretas para más análisis. Nuestras técnicas se basan en representar valores de expresión genéticas como elementos de cuerpos finitos, y utilizan propiedades algebraicas de tales cuerpos. Hemos demostrado nuestras técnicas en un conjunto de datos provenientes de un experimento conducual en ratas, e identificamos dos genes que parecen estar involucrados en memoria y aprendizaje.

Extendimos nuestras técnicas para trabajar con datos de sondas individuales, donde multiples sondas de material genético diferentes miden la expresión de un solo gen. Esta nueva técnica fue demostrada en dos conjuntos de datos provenientes de experimentos iguales hechos en dos tecnologias de microarreglos distintos. El experimento fue diseñado para probar y validar técnicas de análisis, midiendo el grado de concordancia entre los dos tipos de microarreglos.

Para este experimento diseñamos una metodología para cuantificar la concordancia entre los resultados en ambos tipos de microarreglos. Esta metodología utiliza el concepto de información mutua para asignar un valor cuantitativo al grado de concordancia. Nuestros resultados demuestran que nuestra metodología de discretización y corrección de errores resulta en mayor concordancia, determinado por un aumento en la información mutua, cuando la comparamos con las técnicas usuales de análisis que promedian la información de las distintas sondas y de las repeticiones.

También aprovechamos modelos sobre conjuntos finitos para producir un modelo con mayor número de niveles de expresión, que puede capturar diferencias más sutiles entre los niveles de expresión de un gen. Por último, demostramos applicaciones futuras utilizando propiedades algebraicas de cuerpos finitos para encontrar una solución algebraica a el problema de determinar una función que explique la relación entre genes.

To my family, nuclear and extended, for their encouragement and support.

ACKNOWLEDGMENTS

I owe a great debt of gratitude to my advisor, for knowing when to push and when to leave me space, and his guidance throughout this work.

Sandra Peña de Ortiz and Tim Tully provided the microarray data and collaborated on the analisis.

I received partial support from a SCORE grant (S06GM08102) and an INBRE grant (P20RR16470) from the National Institutes of Health.

TABLE OF CONTENTS

page

LIST OF TABLES

LIST OF FIGURES

Figure page

LIST OF ABBREVIATIONS

DNA	Deoxyribonucleic acid, genetic material in cells that encodes genes.
mRNA	Messenger ribonucleic acid, transcribed from the DNA of a gene, later translated into amino acids to make a polypeptide, the components of proteins.
cDNA	Complementary DNA, DNA reverse transcribed from an RNA molecule. In nature, some viruses use complementary DNA to insert mRNA from viral genes into a host cell for production of viral proteins. In molecular biology, cDNA is used to clone eucaryotic genes into procaryotes, or as probes for genes, such as on microarrays
SWAMI	Summed Weighted Averaged Mutual Information, a measure of information content and similarity of several sequences.

CHAPTER 1
INTRODUCTION

1.1 A microarray primer

Microarrays are a technique for measuring the abundance of messenger RNA from many thousands of genes simultaneously in an inexpensive experiment. They are one of the first techniques in high throughput genomics, first described in [40].

They work by immobilizing short sequences of DNA, known as "probes" onto known locations on some fixed substrate. The microarrays described in [40] used full length cDNA clones spotted using hand tools onto nylon membranes. Further refinements have included automated spotting using robotics, and direct synthesis of short oligonucleotide probes onto glass slides using photolithography or ink-jet technologies. Modern microarrays pack hundreds of thousands of probes onto a single slide.

In an experiment, mRNA is extracted from a sample and reverse transcribed in the presence of a labeling agent, producing labeled cDNA called the "target". Since spotted arrays have such great variability, many experiments measured relative abundance of two targets labeled with different fluorescent dyes, in so called "two-color" microarray experiments. Many microarray experiments are still performed in this manner, although *in situ* synthesised arrays have a low slide to slide variation, and can be hybridized with a single target in "single-color" experiments, such as Affymetrix chips.

In the single color experiment, the labeled targets are hybridized to the immobilized probes, and complementary molecules bind to one another. Unbound

DNA is washed away, and the amount of bound target at each probe is measured by scanning and image quantification. Each microarray is hybridized and scanned separately, and the results combined.

Microarray images come from a variety of sources, from scanned autoradiographs to confocal microscopes. The processing of these images is a complex process in it's own right, but for the most part, outside of the scope of the current thesis. I would like to point out, however, that all these microarray images are digitized in some manner or other prior to processing in the computer. In addition, the image analysis is trying to count the number of hybridized targets on each probe. This is also a fundamentally discrete process.

The analysis of microarray data, however, is a difficult task, proving a fruitful area of research in numerous fields. An extensive review is available in [9]. This section will attempt to review the literature most relevant to the proposed work.

1.2 Stages of microarray analysis

The analysis of microarray data is a complex, multi-stage process that typically involves the following steps:

1. microarray image analysis
2. normalization
3. detection of differential expression
4. clustering
5. biological network analysis

1.3 Clustering

Clustering of gene expression measurements is an important step in many analysis, most early microarray work performed hierarchical clustering, where genes are successively agglomerated into groups by selecting the two clusters whose average expression values are closest [11]. It is typical to first cluster genes before trying to determine the gene regulatory network by reverse engineering. Clustering helps reduce the computational resources required to analyze microarray data sets by grouping together many separate genes that demonstrate similar patterns of expression [2]. It also can help in determining common functionality or common regulatory elements of genes which cluster together [10].

1.4 Genetic network models

As early as 1969 Stuart A. Kauffman [21] (see [22] for a detailed review) proposed the far-reaching and important idea of using Boolean logic, the logic of computers, to produce and gain insight into the logic of genes. The invention of cDNA microarrays brought a resurgence of interest in these Boolean genetic network models.

1.5 Boolean models and the reverse engineering problem

A series of papers in 1998, 1999 and 2000 defined Boolean network models, reverse engineering, and proved interesting results on the number of experiments required to completely define a Boolean network. Taking the model definition from [17], for example, we can describe a genetic network as a directed graph consisting of N nodes numbered $1, 2, \ldots, N$, such that for each node n there is an associated Boolean function f_n. An edge from a node to another represents an influence of the first node on the expression of the second. We understand that the following is a formalization of the model presented in [17].

Definition 1.5.1. *A Boolean variable assumes the values 0,1.*

Definition 1.5.2. *A Boolean function is a function involving Boolean variables and the operations* $\wedge$, $\vee$, $\neg$ *with the following definitions:*

X	Y	$X \wedge Y$	$X \vee Y$	$\neg X$
1	1	1	1	0
1	0	0	1	0
0	1	0	1	1
0	0	0	0	1

Definition 1.5.3. *A dBnm with n genes $G_1, \ldots, G_n$ is a set of n Boolean variables $(x_1, \ldots, x_n)$, and a set of n Boolean functions $(f_1, \ldots, f_n)$. The Boolean variables represent the current expression of each gene or stimuli, and the Boolean function f_i represents how the gene G_i is updated given the current values of all the other genes.*

Lemma 1. *Given n Boolean variables $(x_1, \ldots, x_n)$ and define $f(x_1, \ldots, x_n)$ for all possible values. Then there is a Boolean function that coincides with f as defined.*

Proof: See any book on computer architecture (c. f. [36]) for realizing a Boolean function as sums of products and products of sums.

We will additionally define:

An *expression matrix* is a set of measurements (such as those which result from microarray experiments) over the genetic network. From this expression data, the challenge is to reconstruct or reverse engineer the genetic network.

A *gene perturbation experiment* is an expression matrix where some entries correspond to measurements taken when the value of one gene or more are forced to a known state.

Akutsu *et al.* proved lower and upper bounds on the number of gene perturbation experiments required to completely determine a gene network in [1]. The results are discouraging, since in the general case, the problem is shown to be NP-complete. However, in [32], an efficient algorithm for determining the gene network from a set of input-output pairs is developed, assuming that each gene has an indegree in the directed graph that is at most three. This restriction corresponds to saying that at most three genes have an influence on the expression of the target gene. Further research proceeds on the assumption that this indegree is bounded by a small constant. In [2] it is shown that a gene network will be recovered with high probability in only $O(\log n)$ experiments if the indegree is at most two. An iterative procedure for selecting genes to perturb while determining a genetic network such that the uncertainty in the specification of the model is reduced is described in [17]. After this series of papers, work on these Boolean models was mostly discontinued, biologists objected to the simplicity of the Boolean representation of genes.

It is also important to note that all of these Boolean network papers leave unspecified the manner in which gene expression measurements are converted to Boolean values. For example, [17] simply says that gene values will be approximated as high or low and represented by the values 1 or 0.

1.6 Partial enumeration

In the Boolean network models reverse engineering via partial enumeration of functions as described in [1, 32] requires limiting the number of inputs to each genetic function, usually assuming that between 2 to 4 genes affect the expression of a given gene. This requirement for computational tractability directly conflicts with the evidence that transcriptional networks for higher organisms are significantly more complex [30, 34], with even yeast having up to 10 or more transcription factors influencing the expression of a single gene [29].

1.7 Finite Dynamical Systems

Boolean networks thus have 2 limitations: they can only represent genes as “on” or “off”, and they limit the nature of the gene interaction network to ensure computational tractability.

Finite dynamical systems (FDS) are a broad class of models that consist of a pair (X, f). FDS represent the state of a system as a value over the finite set X, and the state of the system evolves over time by iterating a fixed function $f : X \to X$. The function f applied to the current state yields the new state of the system. A particular type of finite dynamical system where the set X is a finite field was developed in [26] for the simulation of computer systems, and later adapted for genetic regulatory networks [28].

These models allow for a richer variation of gene expression levels, and remove the restrictions on the degree of the genes. Several alternative representations and techniques for polynomial models over finite fields have been developed [3, 14, 35], and [4] demonstrates that these polynomial models are equivalent to those described in [26, 28].

This research lead to a series of techniques for error-correction, clustering for any finite set, and reverse engineering [5] based on finite fields.

1.7.1 Finite fields

A *finite field* $\{F, +, \cdot\}$ is a finite set F, and two operations $+$ and $\cdot$ that satisfy the following properties:

- $\forall a, b \in F,\ a + b \in F,\ a \cdot b \in F$
- $\forall a, b \in F,\ a + b = b + a,\ a \cdot b = b \cdot a$
- $\forall a, b, c \in F,\ a + (b + c) = (a + b) + c,\ (a \cdot b) \cdot c = a \cdot (b \cdot c)$
- $\forall a, b, c \in F,\ a \cdot (b + c) = (a \cdot b) + (a \cdot c)$
- $\exists 0, 1 \in F,\ a + 0 = 0 + a = a,\ a \cdot 1 = 1 \cdot a = a$
- $\forall a \in F,\ \exists (-a) \in F$ s.t. $a + (-a) = (-a) + a = 0$

  $\forall a \neq 0 \in F, \exists a^{-1} \in F$ s.t. $a \cdot a^{-1} = a^{-1} \cdot a = 1$

The field is closed under both operations, both operations are commutative and associative, and the distributive law holds. There are additive and multiplicative identities and inverses.

The real and rational numbers are fields with an infinite number of elements. A finite field has the same properties as the rational numbers, over a finite set. In particular, we can add, subtract, multiply and divide any element by any other.

1.7.2 The world's smallest finite field

The integers 0 and 1, with integer addition and multiplication modulo 2 form the finite field $Z_2 = \{\{0,1\},+,\cdot\}$.

The operators $+$ and $\cdot$ are defined as follows:

$+$	0	1
0	0	1
1	1	0

$\cdot$	0	1
0	0	0
1	0	1

1.7.3 Boolean operators and Z_2

We can realize any Boolean operator as an expression over Z_2:

$$\begin{aligned} X \wedge Y &= X \cdot Y \\ X \vee Y &= X + Y + X \cdot Y \\ \neg X &= 1 + X \end{aligned}$$

Note that $+$ corresponds to the exclusive or (**xor**) Boolean function, so all Boolean operators can be realized with **and** and **xor**.

CHAPTER 2
ERROR CORRECTION AND CLUSTERING GENE EXPRESSION DATA USING MAJORITY LOGIC DECODING

2.1 Introduction

2.1.1 Microarray experiments

The microarray studies described here focused on one cognitive task, conditioned taste aversion (CTA), as a model system for gene expression profiling. CTA is an associative aversive conditioning paradigm in which pairing gastrointestinal malaise (induced by lithium chloride, LiCl, the unconditioned stimulus) with prior exposure to a novel taste (the conditioned stimulus) may create a strong and long lasting aversion to the novel taste.

CTA lends itself as an excellent model system to study the dynamics of gene regulation in learning and memory because it is a single trial associative learning paradigm, which involves discrete regions in the brain, including selected amygdala nuclei [46, 47].

Behavioral training

Behavioral training of rats in the CTA task prior to collection of the microarray data used for our experiments was done as described in [12].

Microarray measurements

The gene profiling experiment was replicated five times. Four animals were used per condition for each replicate. Thus, a total of twenty rats were used per condition. Animals were sacrificed by decapitation at 1, 3, 6, and 24 hours after conditioning and amygdala enriched tissue punches were obtained for RNA isolation. Hybridization, image capture and analysis was similar to the procedures described in [39]. The data set thus obtained (CTA data set) is described in [7]. In summary, the data has two controls, the pre-treatment group and the one hour saline group, and four time points, 1, 3, 6, and 24 hours after conditioning. Each array has 1185 genes, and we have 5 biological replicates of each array.

2.2 Methods

The methods described here were developed for the purpose of analyzing the CTA data set, but are sufficiently general to analyze any equivalent data set.

2.2.1 Error correction and clustering

We have devised a scheme for detecting and correcting errors using discretized data.

Here we apply our technique to data from gene A01a in the CTA data set described in Section 2.1, to illustrate the method:

Pre	Sal	1 h	3 h	6 h	24h
0.172	0.099	0.176	0.142	0.062	0.152
0.274	0.168	0.126	0.114	0.104	0.276
0.003	0.119	0.552	0.178	0.193	0.114
0.114	0.139	0.6	0.311	0.179	0.181
0.04	0.006	0.172	0.103	0.036	-0.047

Each row is a repetition of the microarray experiment. Columns represent the measurements of the genes. Pre and Sal are the pretreatment (time 0) and injection with saline solution controls.

2.2.2 Averaging

The first step in the analysis is to average the expression across repetitions.

average	0.12	0.11	0.32	0.17	0.12	0.13

We also average our control columns to obtain a control value of 0.115.

We compute an epsilon value, such that either the 1 h or 24 h columns are within the range of control +/- epsilon. In this case, the epsilon is 0.022.

2.2.3 Discretization

We proceed to discretize each repetition by comparing each column to the control +/- epsilon. We illustrate for repetition 1:

Pre	Sal	1 h	3 h	6 h	24h
0.172	0.099	0.176	0.142	0.062	0.152

The control for this repetition is $(0.172 + 0.099)/2 = 0.1355$, epsilon is fixed for all our tests at 0.022. We now call a column “+” if its value is greater than the control + epsilon, “-” if is is less than control - epsilon, and “0” otherwise.

Pre	Sal	1 h	3 h	6 h	24h
+	-	+	0	-	0

Repeating for the remaining repetitions yields;

Pre	Sal	1 h	3 h	6 h	24h
+	-	+	0	-	0
+	-	-	-	-	+
-	+	+	+	+	+
0	0	+	+	+	+
0	0	+	+	0	-

2.2.4 Majority logic decoding

We now obtain a consensus for each column by majority logic decoding, 3 or more occurrences of the same symbol in a column indicate that symbol is the consensus. If no consensus is obtained, we indicate “?”.

	Pre	Sal	1 h	3 h	6 h	24h
consensus	?	?	+	+	?	+

2.2.5 Discretizing against averaged controls

The above procedure is very sensitive to the value of the controls. Errors in the controls can skew the entire set of calls. We devised an alternate method of discretization that replaces the control value for each row by the average of the control value for all the rows. In our case this average control is 0.113. The discretization of the repetitions using this average control yields the following values, which we summarize with this consensus versus average control (cvac):

	Pre	Sal	1 h	3 h	6 h	24h
	+	0	+	+	-	+
	+	+	0	0	0	+
	-	0	+	+	+	0
	0	+	+	+	+	+
	-	-	+	0	-	-
cvac	?	?	+	+	?	+

2.2.6 Discretizing the average

We also compute the discretization of the average values of each column, using the control 0.113 and the epsilon 0.022:

	Pre	Sal	1 h	3 h	6 h	24h
average	0.12	0.11	0.32	0.17	0.12	0.14
calls	0	0	+	+	0	0

2.2.7 Error correction

We now enter an error correction phase, we seek out outliers in the data of the columns and remove them, and recompute the average, controls, and epsilon.

Pre	Sal	1 h	3 h	6 h	24h
—	0.099	0.176	0.142	—	0.152
—	—	0.126	0.114	0.104	—
0.003	0.119	—	—	0.193	0.114
0.114	0.139	—	—	0.179	0.181
0.04	—	0.172	0.103	—	—

With these outliers deleted from our data we now have new averages, control and epsilon values:

	Pre	Sal	1 h	3 h	6 h	24h
average	0.052	0.119	0.158	0.12	0.159	0.149
control	0.086					
epsilon	0.063					
calls	0	0	+	0	+	0

2.2.8 Consistent calls

We are now ready to produce a consistent set of calls for the gene. A set of calls is consistent if the following conditions are met:

1. at least two of the above set of calls agrees in the last 4 columns of data (1 h, 3 h, 6 h, and 24h)
2. either the 1 h or the 24 h columns is a “0”
3. across the last 4 columns of data, the column exhibits the consecutive zeros property (*i.e.,* values do not oscillate between “0” and “+” or “-”)

As an example, the set of calls for A01a are:

	1 h	3 h	6 h	24h
consensus	+	+	?	+
cvac	+	+	?	+
average calls	+	+	0	0
new calls	+	0	+	0

These calls are not consistent, and this gene is removed from further examination. Together, the procedures we developed and the consistency criteria try to capture biologist's intuitions on the nature of gene expression changes.

2.3 Results

We have performed the analysis described above on the CTA data set described in Section 2.1.1. In this data set, there are 127 consistent genes, which we divide into clusters by grouping together the genes that have the same set of calls in the 1 hour through 24 hour time points. This results in the 23 clusters shown in Table 2–1.

Table 2–1: Consistent genes clustered by the error correction procedure.

Cluster	Gene coordinate
- 0 0 0	A05f, A12k, A14g, B07m, B07n
0 + + +	B01n, B04i, B05l, B06j, B06l, C10l, D01j, D10l, E01k, E03j, E09c, E10e, E13i, E13l, F01e
+ 0 0 0	B07e, D04f, D09l
- - 0 0	A02m, A14e, B03i, C08i, F08c, F13e
- - + 0	C14k, E03l
0 0 + +	B13n, D14g, E06i, E10m, E11l, E13e, E14d
- + + 0	B06m, E02i
0 0 - -	A05n, D02a, F12e
0 - + +	B04j, E10l, F11j
+ - - 0	C01f, F01a
0 - + -	A11d, C09m
- + - 0	D12a
0 + - +	D09i
- - - 0	A02l, A03h, A09c, B10l, C02m, C04d, C04f, C06e, D02b, F02b, F03c, F09n, F11e
+ - + 0	B13a, F02a
0 0 0 -	A10l, C08a, C14g, C14l, D13e
+ + 0 0	A08l, B08b, C08j, F11k
+ + - 0	D04l
0 0 0 +	A07i, B09h, C10c, D08n, E03m, E04i, E13h, E14f
0 - - +	C01e, F12j
0 - - -	B05b, C13a, C13i, C14n, E02e, F04l, F06a, F11l
- + 0 0	A12m
+ + + 0	B01i, B07f, B10c, B10d, B14h, D08f, D09e, E03i, E14k, F02n, F05k, G15

A particular focus of interest in our studies was the identification of genes regulated by the transcription factor CREB (cAMP Responsive Element Binding protein), which is known to play important roles in memory formation [25]. We

Table 2–2: Genes that bind CRE.

Coord.	Accession	Description
A05h	U38938	activating transcription factor 2 (ATF2); cAMP response element DNA-binding protein 1 (CREBP1)
A05l	X14788	cAMP-responsive element-binding protein 1 (CREB1)

focused on the expression of both CREB and other genes with similar patterns of expression in order to detect changes in gene expression paralleling CREB's expression. CREB binds to a DNA element called cAMP-response element (CRE) in the promoter region of its target genes, and in conjunction with a co-activator promotes the initiation of their transcription [33].

There are two genes in our data set that bind CRE, A05h and A05l, these genes are described in Table 2–2. The coordinate column is a unique identifier for the spot on the Clontech arrays, the accession number is Clontech's assignment of a gene in the Genbank nucleotide database to this entry. Of the two CREB genes on the arrays, the gene most associated with learning and memory processes is A05l, or Creb1. The discretization of the average expression of A05l yields "000+", therefore we focused on the cluster labeled "000+". The calls for these genes represents no change over the 1, 3, and 6 hour time points, followed by upregulation at the 24 hour time point. This cluster consists of genes whose expression most closely matches the expression profile of Creb1. We investigated the genes in this cluster in depth, retrieving the gene information and sequence from the Ensembl Genome Browser version 32 [16].

From Ensembl we obtained genomic sequence for each of these genes, 1020 base pairs starting 800 base pairs upstream of the transcription start site. These sequences were then submitted to TESS [41] to search for transcription factor binding sites. We look for the CRE element, a DNA sequence that is the target site for CREB. Genes that have CRE in their upstream region are potential targets of regulation by CREB.

Table 2–3: Members of the gene cluster 000+.

Coord.	Accession	Description
A07i	L24388	galactosyltransferase-associated protein kinase (GTA); CDC2-related protein kinase (CDC2L1)
B09h	L10362	synaptic vesicle protein 2B
C10c	L33869	ceruloplasmin (CERP; CP); ferroxidase
D08n	X63255	N-methyl-D-aspartate receptor subtype 1 (NMDAR1; NR1); glutamate receptor subunit zeta 1 (GRIN1)
E03m	M29712	melanin-concentrating hormone (PMCH; MCH)
E04i	V01228	calcitonin
E13h	M20713	guanine nucleotide-binding protein G(K) alpha 3 subunit (G(I) alpha 3 (GNAI3)
E14f	X06890	ras-related protein RAB4A

Based on our findings we focused on two specific genes: E03m or Pmch (pro-melanin-concentrating hormone) and E04i or Calca (calcitonin/calcitonin-related polypeptide, alpha). Both genes have CRE elements in their upstream regions. According to the Rat Genome Database [38], Pmch is a cyclic neuropeptide that induces hippocampal synaptic transmission. Pmch also seems to have an effect on appetite or metabolism [37] and anxiety [23], and promotes synaptic transmission in the hippocampus [45]. Calca is principally a vasodilator, but seems to have a role in axonal regeneration or synaptogenesis [31]. Thus, these genes exhibit a pattern of expression consistent with the expression of Creb1, have CRE elements upstream of their transcription start site, and seem to have a role in strengthening or creating new synapses.

2.4 Discussion

We have developed a method for error correction of microarray experiments. The technique produces a clustering of genes and describes each gene as unchanged, upregulated, or downregulated, in accordance to biologists natural description of expression levels. We applied these techniques to a microarray data set derived from a CTA experiment in rats, looking for genes that may be important in learning and memory processes. We found two genes, Pmch and Calca, that share an expression pattern with CREB, contain CRE in their upstream regions, and have demonstrated function related to synaptic plasticity. Pmch and Calca are strongly implicated as important genes for the formation of memories. We are now actively seeking confirmation of these genes' role in CTA and of their regulation by CREB as a result of CTA training.

CHAPTER 3
MAJORITY LOGIC DECODING FOR PROBE-LEVEL MICROARRAY DATA

3.1 Introduction

In Chapter 2 we described a method for error correction of microarray data. That method produces a coarse characterization of gene expression levels, based on majority logic decoding of thresholded genes from multiple repetitions. Many microarray experiments use multiple probes per gene. This is typical of Affymetrix style gene chips, but is also seen on oligonucleotide arrays. In the analysis such data, a probe summarization step is performed. There exists a great variety of probe summarization techniques, many are compared in [8, 20].

Section 3.2.2 describes an extension of our discretization and error correction procedure to deal with multiple probes per gene. Section 3.2.3 describes our principal contribution, a technique based on mutual information to compare the degree of concordance of results obtained from two data sets. Mutual information had been used previously to perform reverse engineering of gene expression networks from microarray data [32] or cluster microarray data [6], but not to measure correlation across two data sets. Section 3.2.1 describes two data sets, obtained on two different microarray technologies that we use to validate our analysis procedures.

We have applied the discretization and error correction procedure to two *Drosophila* data sets described in Section 3.2.1, one Affymetrix and one NimbleGen, having multiple probes per gene. The experimental technique is designed to compare analysis procedures by measuring the degree of concordance in the two separate data sets. Analysis procedures that produce good concordance across the data sets are thus validated empirically. Our results show that our new technique results in a greater amount of shared information between data sets than traditional approaches based on averaging of probes and gene expression levels across repetitions (Section 3.3). We find much more correlation between the data sets than that detected by earlier techniques. This increased concordance is principally due to the recovery of many false negatives eliminated by the prior analysis techniques.

3.2 Methods

3.2.1 Microarray data

The data sets were produced in experiments comparing gene expression levels at different times after odor avoidance training of *Drosophila melanogaster* [44, 48]. The experiments were run on drosgenome1 chips from Affymetrix (Santa Clara, CA, USA), and a set of custom arrays from NimbleGen (Madison, WI, USA). A data set consists of 10 repetitions of each condition (massed training, spaced training) at 3 separate time points, 0 (no training), 6, and 24 hours after training. The Affymetrix arrays have 14 probes for each gene, and 14010 probe sets, including controls. The NimbleGen arrays have a set of probes with around 10 probes for each probe set, and 12240 probe sets, including controls.

These experiments were designed to test the degree of concordance between genes produced by different analysis techniques. Analyses that produce the same results across the two data sets should be detecting some ground truth of the biological system, and are less likely to be detecting spurious signal from the particular experimental technique.

3.2.2 Error correction of probe-level data

Chapter 2 described an error correction method for replicate microarray data sets. We assumed each gene was represented by a single probe, as is typical of cDNA arrays. To extend our method to multiple probes, we first run our prior method on the data, treating each probe as a separate entity. This summarizes the repetitions, resulting in a set of calls for each probe. A call is “+” if a probe is upregulated compared to the control, “-” if the probe is downregulated compared to the control, “0” if the probe is within epsilon of the control, and “?” if the results are ambiguous. We then perform majority logic decoding on the set of probes corresponding to each gene as described in Section 2.2.4. Briefly, in the Affymetrix data set, each gene is represented by 14 probes. If a set of probes has more than 7 symbols in agreement, we use that consensus symbol, otherwise we use “?” to denote an ambiguous call. Figure 3–1 illustrates the majority logic decoding results for an example probe set in the Affymetrix data. There are 14 rows and 4 columns, and in the result each column is set to the symbol occurring more than 7 times in the data.

```
[[’0’, ’-’, ’0’, ’+’],
 [’0’, ’-’, ’+’, ’+’],
 [’0’, ’-’, ’0’, ’+’],
 [’0’, ’-’, ’+’, ’+’],
 [’0’, ’-’, ’+’, ’+’],
 [’-’, ’-’, ’0’, ’0’],
 [’0’, ’-’, ’0’, ’+’],
 [’0’, ’-’, ’0’, ’+’],
 [’-’, ’-’, ’-’, ’0’],
 [’0’, ’-’, ’0’, ’+’],
 [’-’, ’-’, ’0’, ’0’],
 [’0’, ’-’, ’0’, ’+’],
 [’0’, ’-’, ’0’, ’+’],
 [’0’, ’-’, ’+’, ’+’]]

 ’mld’: [’0’, ’-’, ’0’, ’+’]
```

Figure 3–1: Majority logic decoding of an example probe set.

3.2.3 Sorting genes by weighted mutual information

Because we have two experiments performed on different microarray technologies, we wish to discover which genes demonstrate the same patterns of expressions in the two data sets. We developed a program to compare the expression in the two data sets using weighted mutual information. Mutual information can measure positive (same expression patterns) and negative (inverted expression patterns) correlation, but in our case, we want to select genes that show only very similar patterns of expression, not opposite patterns. Thus, we use a weighted variant of mutual information [15].

$$I(X, Y) = \sum_{y \in Y} \sum_{x \in X} w(x, y) p(x, y) \log \left(\frac{p(x, y)}{p(x)p(y)} \right)$$

Where $w(x, y)$ is the weight assigned to the combination of symbol x and y, described below, $p(x, y)$ is the probability of the combination of symbol x and y, and $p(x)$ is the frequency of symbol x in the sequence X.

We set the weights such that similar patterns of expression are given higher weights, and opposite expression is given lower weight:

$$w(x, y) = \begin{cases} 1.0 & \text{if } x = y,\ x, y \neq \text{?} \\ 0.5 & \text{if } x = \text{?} \text{ or } y = \text{?} \\ 0.1 & \text{otherwise} \end{cases}$$

Ambiguous calls are given an intermediate weight.

Figure 3–2 illustrates the calls in each of the data sets for the example probe set. Equation (3.1) shows the computation of the weighted mutual information (WMI) for these two sequences.

```
ac = [’0’, ’-’, ’0’, ’+’]
nc = [’0’, ’-’, ’-’, ’-’]
```

Figure 3–2: Example calls for a single probe set in both data sets.

$$\begin{aligned}
I(\texttt{ac},\texttt{nc}) &= w(\texttt{-},\texttt{-})p(\texttt{-},\texttt{-})\log\left(\frac{p(\texttt{-},\texttt{-})}{p(\texttt{-})p(\texttt{-})}\right)+ \\
&\quad w(\texttt{0},\texttt{-})p(\texttt{0},\texttt{-})\log\left(\frac{p(\texttt{0},\texttt{-})}{p(\texttt{0})p(\texttt{-})}\right)+ \\
&\quad w(\texttt{+},\texttt{-})p(\texttt{+},\texttt{-})\log\left(\frac{p(\texttt{+},\texttt{-})}{p(\texttt{+})p(\texttt{-})}\right)+ \\
&\quad w(\texttt{0},\texttt{0})p(\texttt{0},\texttt{0})\log\left(\frac{p(\texttt{0},\texttt{0})}{p(\texttt{0})p(\texttt{0})}\right) \\
&= 1\cdot 1/4\cdot\log\left(\frac{1/4}{1/4\cdot 3/4}\right)+0.1\cdot 1/4\cdot\log\left(\frac{1/4}{2/4\cdot 3/4}\right)+ \\
&\quad 0.1\cdot 1/4\cdot\log\left(\frac{1/4}{1/4\cdot 3/4}\right)+1\cdot 1/4\cdot\log\left(\frac{1/4}{2/4\cdot 1/4}\right) \\
&= 0.35
\end{aligned} \tag{3.1}$$

We obtained from Affymetrix a file with the sequence annotations for every probe on the drosgenome1 chips, `DrosGenome1.na21.annot.csv`. We used the “Probe Set ID” and “Ensembl” columns to construct a map from the ID used by Affymetrix to the IDs used in the NimbleGen arrays. Several Affymetrix Probe Set ID have more than one Ensembl ID listed. Because of this we average the WMI for all NimbleGen probe sets that map to the same Affymetrix probe set.

We sum the average WMI over all Affymetrix probe sets, and obtain a single score for a particular analysis method, the summed weighted averaged mutual information or SWAMI.

With the SWAMI score, we can perform many analyses, and compare the SWAMI score obtained to determine which analysis technique produces the best agreement between the two data sets.

In addition, we sort the probe set list by the weighted averaged mutual information, this produces a list of probe sets ranked according to how informative and how similar they are between data sets.

3.2.4 Normalization and summarization tests

We set up a series of analyses to test the effect of different transformations and summarization algorithms on the concordance between the two data sets, as measured by the SWAMI score. We use the `affyPLM` package from BioConductor [13]. We set up a comparison of “log2”, “sqrt” and “cuberoot” transformations on the expression values, and “Huber” “fair” and “Cauchy” methods of robust regression of the probe values. Once we produce the summarized data, we use the `limma` package from bioconductor to produce a discretization using the `decideTests` function [42, 43]. These discretizations are compared between the two data sets using the SWAMI score, just as we compared the error correction methods above. We also ran our error correction and clustering procedures on the data summarized using the `rma` command from BioConductor [18].

3.3 Results

Table 3–1 summarizes the total SWAMI scores obtained for several different transformation and regression methods on the *Drosophila* data. The defaults for `affyPLM` are log2 transformation and Huber regression, but sqrt transformation and fair regression yielded much better SWAMI scores on our data.

Table 3–1: SWAMI scores for several transformation and regression methods.

Transformation	Regression	SWAMI
log2	Huber	182
log2	fair	186
log2	Cauchy	169
sqrt	Huber	212
sqrt	fair	230
sqrt	Cauchy	200
cuberoot	Huber	207
cuberoot	fair	216
cuberoot	Cauchy	202

Table 3–2 presents the SWAMI scores for our error correction techniques on the *Drosophila* data. All these scores are more than an order of magnitude higher

than the scores for the `affyPLM` based methods. The highest score is the "trimmed mean" method, which discards repetitions which deviate most from the mean. In our case we discard 2 and keep 8 repetitions for each probe.

Table 3–2: SWAMI scores for error correction methods.

Method	SWAMI
trimmed mean	3657
mean	2535
consensus	3058
consensus vs mean control	1525

For comparison, Table 3–3 presents the SWAMI scores for our prior error correction scheme, using standard RMA to summarize probes [18]. The additional level of error correction afforded by the probes results in an increase of the SWAMI score.

Figures 3–3 and 3–4 show the frequency of the individual WMI scores per probe set for two representative methods, the sqrt-fair method, which obtained the best SWAMI score in Table 3–1, and the trimmed mean, the best performer in Table 3–2. The error correction methods show a wider distribution, whereas the `affyPLM` methods are very narrowly distributed around 0.

Figures 3–5 and 3–6 illustrate the distribution of nonzero calls in each probe set in the data. In binary vectors this would be called the "weight" of the vector. The sqrt-fair method produces calls distributed normally with mean around 5, whereas the trimmed mean method produces calls with nearly all entries nonzero.

3.4 Discussion

We have presented a novel method of error correction for probe-level microarray data, such as that generated by Affymetrix chips. We have also developed a novel

Table 3–3: SWAMI scores for RMA summarized data for several methods.

Method	SWAMI
trimmed mean	2753
mean	2610
consensus	1920
consensus vs mean control	1888

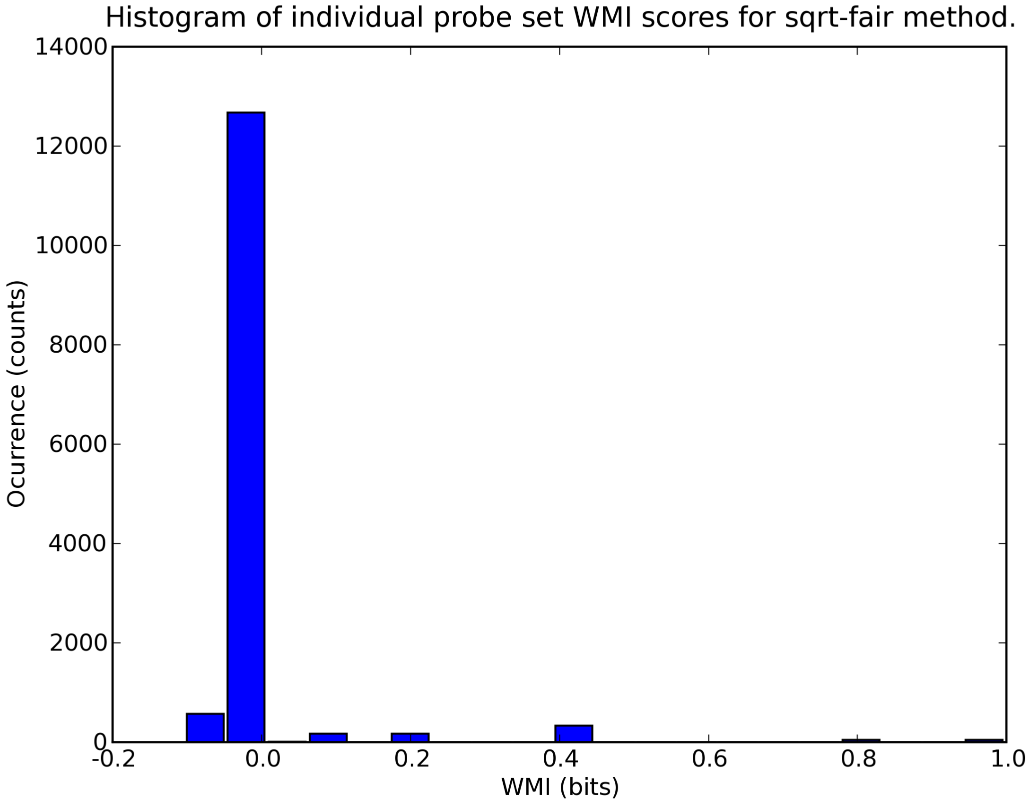

Figure 3–3: Distribution of weighted mutual information scores for individual probe sets in the sqrt-fair method.

scoring method for measuring the degree of agreement between two independent data sets representing the same or similar genes. The SWAMI score measures mutual information between data sets, but is weighted by a score to produce biologically meaningful correlation, two sequences cannot be inversely correlated and still have a high SWAMI score.

We have applied our methodology to a large data set obtained from an odor-avoidance training experiment with *Drosophila melanogaster.* This experiment was designed to test different data analysis techniques by measuring concordance between the results on both data sets. The results indicate that our error correction procedure results in much higher SWAMI scores between two different data sets than other more common analysis techniques (Tables 3–1 and 3–2). The first reason

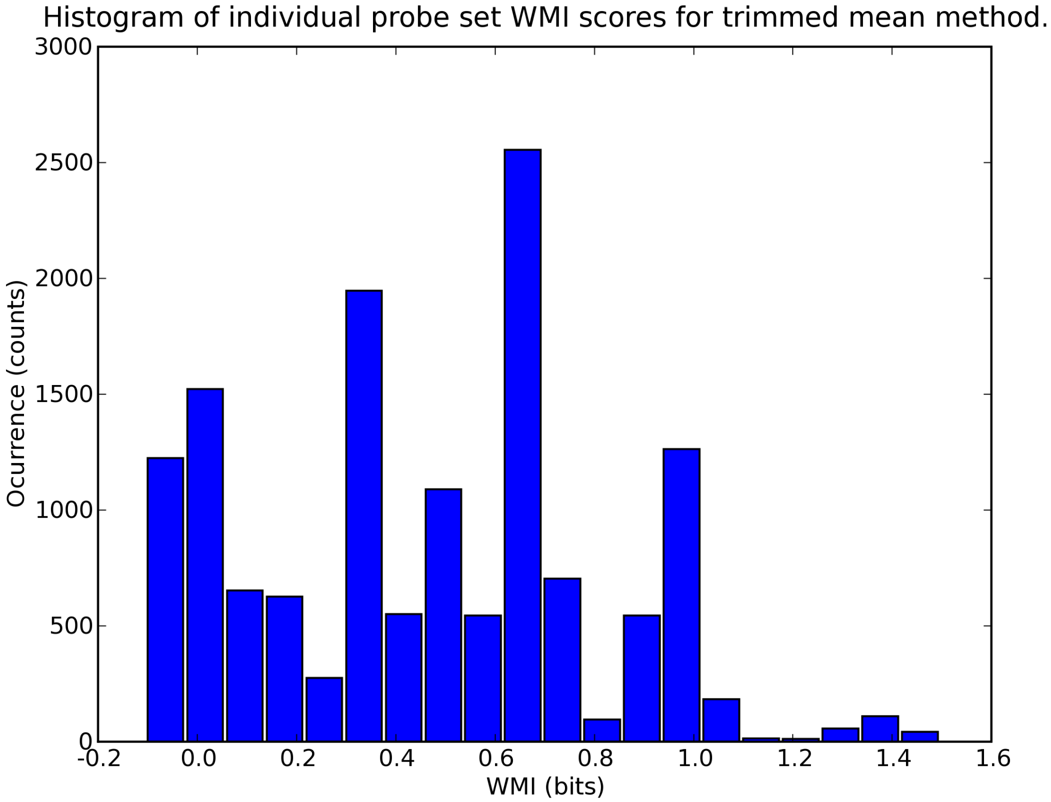

Figure 3–4: Distribution of WMI scores for individual probe sets in the trimmed mean method.

is that the set of calls produced by the other techniques have more “0” calls, no significant change in expression, as seen in Figure 3–5 where the weight peaks around 5. The majority logic decoding results in calls with more “+” and “–” values, the weight illustrated in Figure 3–6 is much higher. If these calls were not in agreement between data sets, however, the SWAMI scores would not be high. Thus the second improvement is that the error correction procedure increases the degree of concordance between the data sets, as measured by the SWAMI score. Prior studies of concordance across microarray technologies have demonstrated poor results, the authors concluded diverse array technologies cannot be compared [24]. However, newer studies have shown large variability between labs, and that the best labs have low variability, even using different technologies[19].

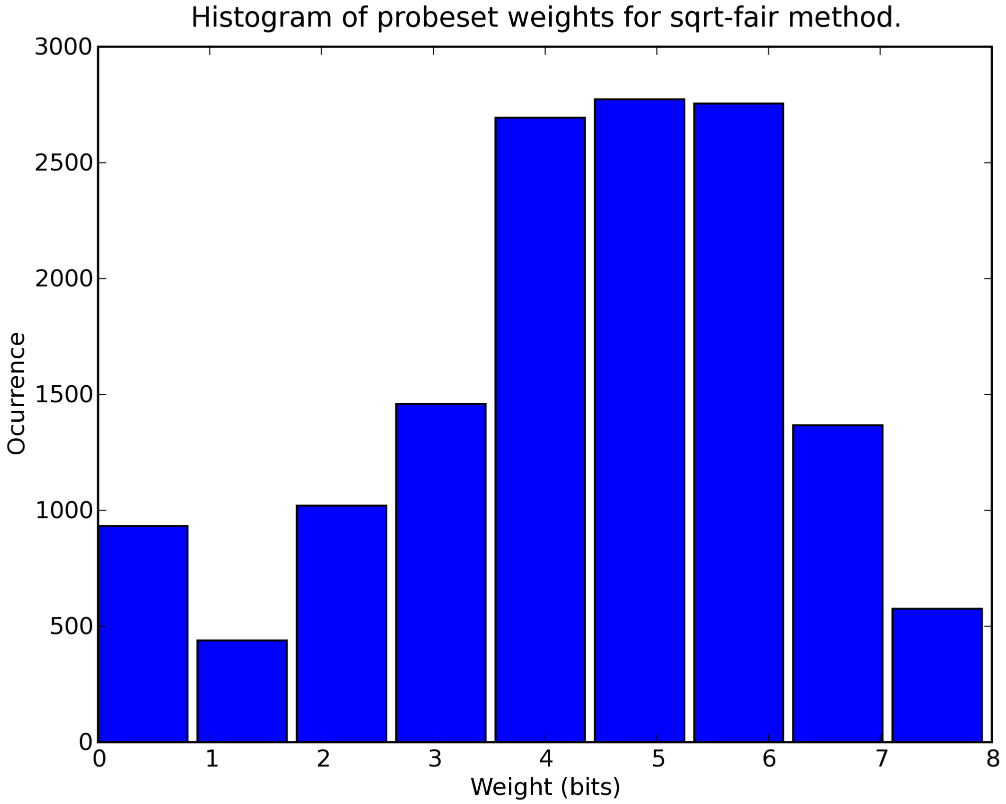

Figure 3–5: Distribution of nonzero calls for individual probe sets in the sqrt-fair method.

The SWAMI score leads to a list of probe sets that in some sense optimize two biologically relevant criteria: the probe sets must be informative in the sense that they take on a range of values, and the probe sets must be consistent between the two data sets. These are precisely the kind of genes we seek to understand the molecular changes underlying the conditions we are studying.

The error correction and SWAMI procedures produce a list of probe sets sorted by the WMI score, there are a small number of probe sets with maximal scores. Future work should include examining these candidate probe sets to confirm their role in learning and memory processes.

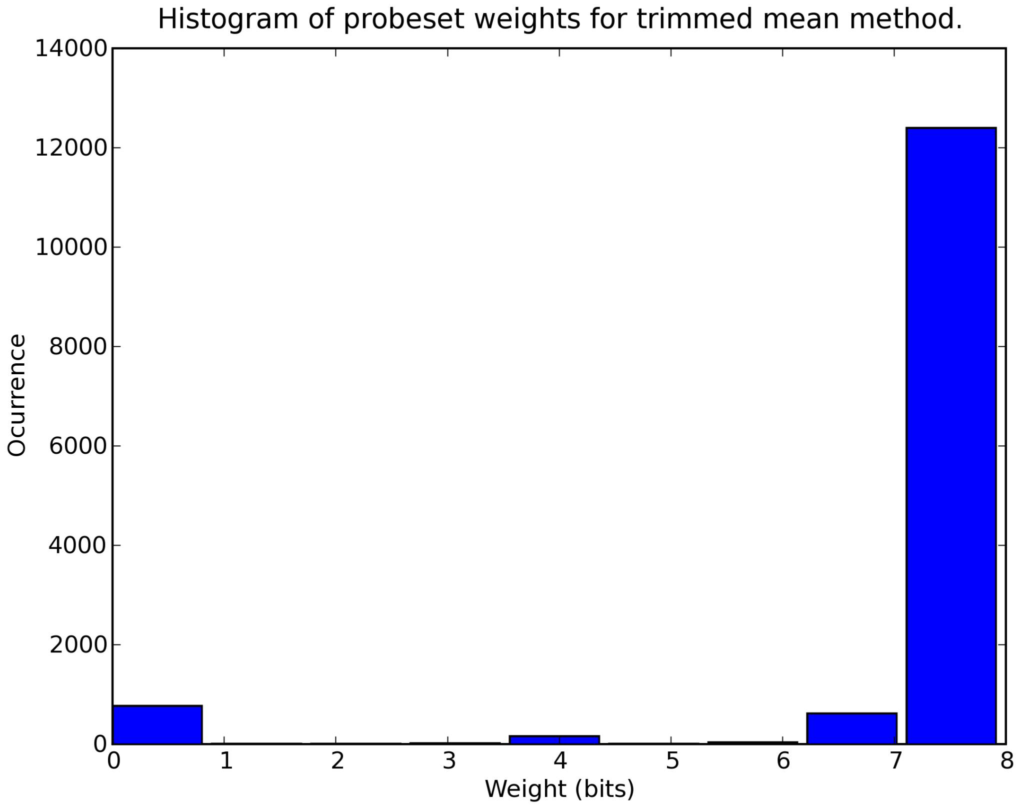

Figure 3–6: Distribution of nonzero calls for individual probe sets in the trimmed mean method.

CHAPTER 4
DISCRETE METHODS IN MICROARRAY ANALYSIS

4.1 Introduction

We previously described finite models of gene expression networks, and developed techniques to discretize gene expression measurements. We illustrated these techniques on data sets from rats and fruit flies, using a model with 3 expression values for each gene, $X(3) = \{\texttt{0, +, -}\}$ representing genes unchanged, upregulated or downregulated.

This chapter extends our discretization and error correction methods to sets with more elements, while preserving useful qualities of the data, such as the SWAMI score.

4.2 Methods

4.2.1 Discretizing to a set with more elements

We have modified our procedure for error correction on probe level data to produce gene expression levels in a set with 5 elements. We label these expression levels as $X(5) = \{\texttt{*, +, 0, -, =}\}$ with the same meaning as before, adding `*` for highly upregulated genes (two `+` symbols), and `=` for strongly downregulated genes (two `-` symbols).

The procedure for discretizing to the set with 5 elements is as follows: let x be the expression of gene minus control and ϵ be the discretization threshold as in Section 3.2.2.

- if $x > 2\epsilon$ encode as $+2$ (`*`)

- if $1\epsilon < x \leq 2\epsilon$ encode as $+1$ (`+`)
- if $-1\epsilon \leq x \leq 1\epsilon$ encode as `0`
- if $-2\epsilon \leq x < -1\epsilon$ encode as -1 (`-`)
- if $x < 2\epsilon$ encode as -2 (`=`)

4.2.2 Selecting consistent genes

In this section, we again devise a new test for consistency. Our concern is that genes discretized to the set with 5 elements should agree with those already chosen for the set with 3 elements.

- Rank genes by WMI score for $X(3)$ and $X(5)$
- Pick top 1000 genes from each method
- Take intersection of top genes from both methods

4.2.3 Searching for a better epsilon

The mutual information for a sequence depends on ϵ: if epsilon is too high, all calls are `0`, which would give good concordance, but low informativeness. At the other extreme, if epsilon is too low, all calls are double negative `=`, or double positive `*`, yielding a high information content, but low concordance between the experiments. We examine the relationship between ϵ and the SWAMI score by choosing ϵ, and testing SWAMI score for different multipliers, to see effect of changes in the discretization threshold on the mutual information.

4.3 Results

4.3.1 Clusters

Table 4–1 presents the clusters formed by discretizing to $X(3)$, the set with 3 elements, and the clusters into which those genes fall when discretized to the new set with 5 elements $X(5)$. The 21 clusters formed by discretizing genes to $X(3)$ are split into 72 clusters when discretizing genes to $X(5)$

Table 4–1: Gene clusters resulting from discretization to the set with 3 or 5 elements.

$X(3)$	$X(5)$
+++0	***0 *++0 +**0 +*+0 +++0
++−0	**=0 ++=0
++0+	**0* **0+ *+0+ +*0+ ++0+
+−+0	*=*0
+0++	*0** *0++ +0** +0*+ +0+* +0++
+0−−	*0== +0=− +0==
−++0	=+*0
−+−0	=*=0 =+−0 =+=0
−+0−	=*0=
−−+0	−=*0 ==*0
−−−0	−−−0 −−=0 −=−0 −==0 =−−0 ==−0 ===0
−−0+	=−0+ ==0* ==0+
−−0−	−−0− −−0= −=0− =−0− ==0=
−0++	−0** =0** =0++
−0−+	=0=*
−0−−	−0−− −0−= −0=− −0== =0−− =0==
0+++	0*** 0*+* 0*++ 0+** 0+*+ 0++* 0+++
0++−	0*+=
0+−−	0*== 0+==
0−++	0=++
0−−−	0−−− 0−=− 0−== 0=−− 0=−= 0==− 0===

Table 4–2: SWAMI scores for genes discretized to sets with 3 and 5 elements.

Method	GF(3)	GF(5)
trimmed mean	2753	3229
mean	2610	3226
consensus	1920	1876
consensus vs mean control	1888	1996

4.3.2 SWAMI scores for discretizing to a set with 5 elements

Table 4–2 presents the SWAMI scores for our discretization methods producing calls with 3 or 5 elements on the *Drosophila* data. Discretizing to the set with 5 elements yields better SWMI scores except in the case of the consensus calls.

Figure 4–1 shows the effect of the size of epsilon, the discretization threshold, on the SWAMI scores for error correction and summarization methods on the *Drosophila* data discretized according to the procedure described in Section 4.2.1.

4.4 Discussion

This chapter presented some results that use sets with more elements to represent finer distinctions in gene expression levels. We applied these techniques to the *Drosophila* data sets.

Discretizing to the set with 5 elements yields a slight improvement in the SWAMI scores for the data, except for the consensus method, which showed a small decrease. This method uses majority logic decoding to produce a call, and the increased input symbols resulting from the chage to the set with 5 elements may be causing the majority logic decoding to produce different calls in the two experiments (*i.e.* produce a `+` in one experiment and a `++` in the other), thus decreasing the consistency and the SWAMI score. Revising the weighting function to give a higher score to `-, =` and `+, *` pairs may result in better scores.

Our results show that these techniques can provide insight into biological data sets, using terminology familiar to the biologist, an important consideration in multidisciplinary research environments.

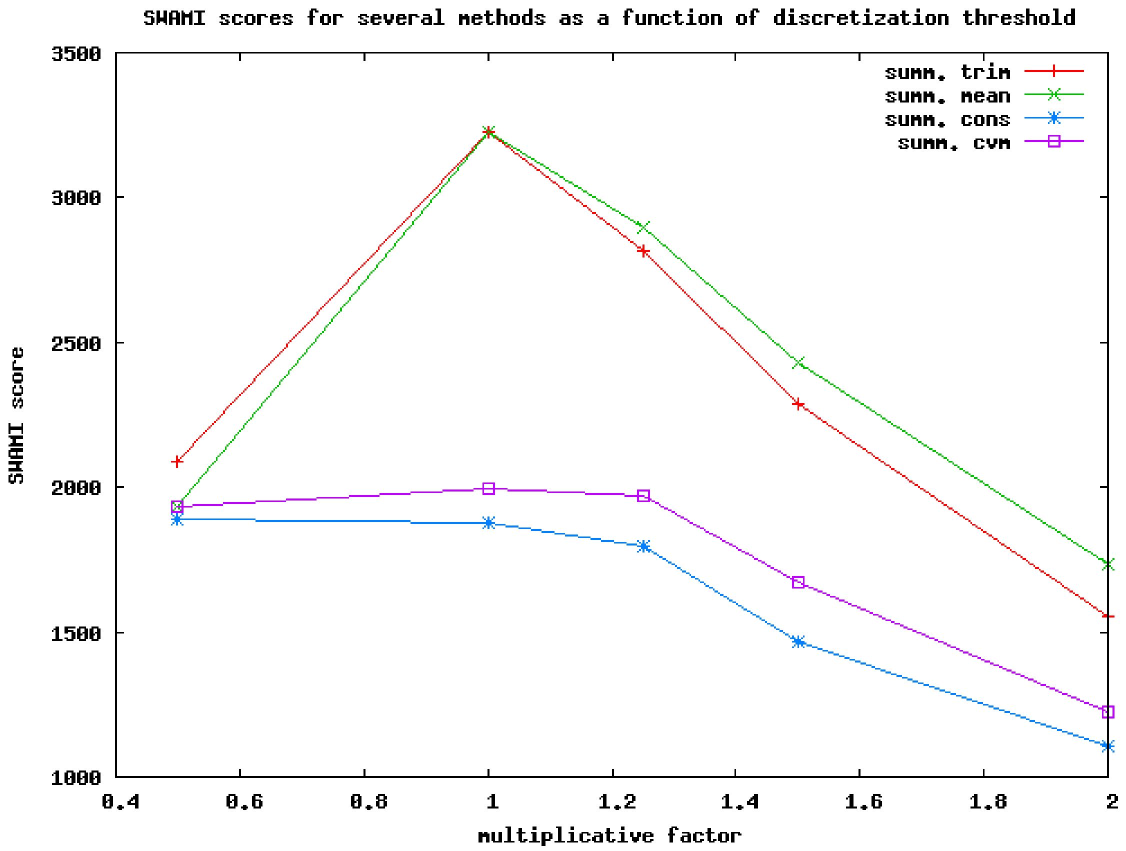

Figure 4–1: SWAMI score for several methods as a function of epsilon. The SWAMI score is plotted for our error correction techniques using summarized probes.

CHAPTER 5
CONCLUSION AND FUTURE WORKS

5.1 Summary

We have presented several new techniques for the analysis of microarray data. These techniques group gene expression values into coarse descriptions, similar to those used by biologists when describing gene expression changes (“upregulated”, “downregulated”, “unchanged”). We encode these coarse characterizations into symbols, and can then use techniques such as majority logic decoding to perform error correction on multiple repetitions of microarray experiments.

We applied these techniques to a time-course experiment performed on rats trained in a conditioned test aversion task. The error correction and clustering technique produced 23 clusters of genes in this task, and we selected candidate genes from a cluster containing the transcription factor CREB, known to be required for learning of this task.

Sequence analysis and a literature search of the candidate genes strongly suggest that two genes, Pmch and Calca play an active role in learning and memory in this task. Research into their role in CTA is still ongoing.

We also extended our error correction technique to microarray data where each gene has more than one probe on the array, called a probe set. Examples of these arrays include Affymetrix arrays, widely used in human and animal research.

We obtained from our collaborators a unique data set from odor avoidance training experiments on fruit flies. The data set contains multiple replicates of a time course done under two different training conditions on two different microarray technologies. We extended our error correction and clustering technique to work with the multiple probe data for each probe set. We also devised a new technique, the SWAMI score, based on mutual information, to measure concordance between genes on the different microarray technologies.

We applied this mutual information technique to validate different microarray analysis techniques by measuring the concordance between the two microarray technologies of the results of several different analysis techniques. We include in the comparison traditional microarray analysis techniques such as 'affyPLM' and our own error correction and clustering techniques.

The results clearly indicate that our error correction and clustering technique results in a much higher SWAMI score than the other analysis techniques tested. In addition, the use of probe-level data in a probe set increased the SWAMI score as well.

The *Drosophila* microarray experiment was designed as an empirical test of different analysis techniques, that our technique produced the highest SWAMI score demonstrates the validity of the methodology.

The discretization procedures developed in Chapters 2 and 3 result in clusters of genes with expression values in the set {`-, 0, +`}. In Chapter 4 we show how we ca construct finite fields with 3 or 5 elements, and use algebraic properties of finite fields to reverse engineer a genetic network for the CTA data.

5.2 Future studies

Future experimental work includes experimental validation of the role of Pmch and Calca in learning and memory. These experiments are being carried out in collaboration with the laboratory of Dr. Sandra Peña de Ortiz (UPR-RRP). In addition, Dr. Tim Tully of Dart Neuroscience LLC has expressed an interest in confirming results of the *Drosophila* analysis using other biological experiments.

On the algorithmic side, we are most interested in exploring if multiple probes can offset the need for multiple repetitions. We have seen that probe-level information can increase SWAMI scores in microarray experiments, if the number of repetitions can be decreased while preserving the SWAMI scores, we may be able to dramatically decrease the cost of microarray experiments.

Another area of interest is augmenting or annotating gene expression data with partial information on gene regulation. As more sequence and functional information is deposited in repositories of biological data, it is important to incorporate this information in the analysis of expression data.

5.2.1 Interpolation

Another example of future work involves exploiting algebraic structure in gene nwtwork models. One such example is reverse engineering gene networks when genes take values from a finite field. This is a special case of our discrete genetic network models.

For a network of n genes, each having values in GF(p), measured at k time points. A time series $S = S_1, S_2, ..., S_k$, where $S_i = (s_{i1}, s_{i2}, ..., s_{in})$, and $s_{ij} \in \mathrm{GF}(p)$ is the expression of the jth gene at time i.

The approach taken in [27] for the reverse engineering problem is to give a procedure using Gröbner basis to find all functions f such that given a time series $S_1, S_2, \ldots, S_k$, the function f has the property that $f(S_1) = S_2$, $f(f(S_1)) = S_3$, and so on. In general, $f^{i-1}(S_1) = S_i$.

We will give an alternative procedure to do this using univariate polynomials over GF(p^n).

The Lagrange Interpolation Formula says that for $n \geq 0$, let $a_0, \ldots, a_n$ be $n+1$ distinct elements of a finite field F, and let $b_0, \ldots, b_n$ be $n+1$ arbitrary elements of F. Then there exists exactly one polynomial $f \in F[x]$ of degree $d \leq n$ such that

$f(a_i) = b_i$ for $i = 0, \ldots, n$. This polynomial is given by

$$f(x) = \sum_{i=0}^{n} b_i \prod_{k=0,k\neq i}^{n} (a_i - a_k)^{-1}(x - a_k).$$

Using this we can therefore give a closed-form solution for the reverse engineering problem as follows:

If we want to find the network function f for a gene network with time series $S_1, S_2, \ldots, S_i$ then f is given by:

$$f = \sum_{j=2}^{i} S_j \prod_{k=1,k\neq j}^{i-1} (S_j - S_k)^{-1}(x - S_k).$$

Proof: We use Lagrange interpolation with $a_0 = S_1, a_1 = S_2, \ldots, a_{i-2} = S_{i-1}$ and $b_0 = S_2, b_1 = S_3, \ldots, b_{i-1} = S_i$, obtaining:

$$f = \sum_{j=2}^{i} S_j \prod_{k=1,k\neq j}^{i-1} (S_j - S_k)^{-1}(x - S_k).$$

5.2.2 Preliminary results

Table 2–1 presented 23 clusters of genes using a ternary model of expression. We can treat these expression values as values in $GF(3)$. Reading down each column of the table, we can assign a value to the four time points as follows:

$$\begin{aligned}
S_1 &= -a^0 + a^2 - a^3 - a^4 - a^6 + a^9 - a^{11} \\
&\quad -a^{13} + a^{14} + a^{16} + a^{17} - a^{21} + a^{22} \\
S_2 &= a - a^3 - a^4 + a^6 - a^8 - a^9 - a^{10} + a^{11} \\
&\quad +a^{12} - a^{13} - a^{14} + a^{16} + a^{17} - a^{19} - a^{20} + a^{21} + a^{22} \\
S_3 &= a + a^4 + a^5 + a^6 - a^7 + a^8 - a^9 + a^{10} - a^{11} \\
&\quad -a^{12} - a^{13} + a^{14} - a^{17} - a^{19} - a^{20} + a^{22} \\
S_4 &= a + a^5 - a^7 + a^8 - a^{10} \\
&\quad +a^{12} - a^{15} + a^{18} + a^{19} - a^{20}
\end{aligned}$$

The top row of the table `- 0 0 0` represents the coefficients of the a^0'th component of the field, where a is the generator of the field GF(3^{23}), thus the first time point S_1 starts with $-1 \cdot a^0$, and the time points S_2, S_3, S_4 start with $0 \cdot a^0$. Reading the remaining rows of the table gives us successively the coefficients of larger terms.

Using the Lagrange interpolation, we can then define a function that interpolates the values in the above time series:

$$\begin{aligned}
f(x) &= (-a^{22} + a^{21} - a^{20} - a^{18} + a^{17} + a^{16} + a^{15} + a^{14} - a^{13} + a^{12} - a^{11} - a^{10} \\
&\quad +a^8 - a^7 - a^6 - a^5 + a^2 - a - 1)x^2 \\
&\quad +(a^{20} + a^{18} - a^{16} - a^{15} - a^{14} + a^{13} + a^{12} - a^{11} + a^{10} \\
&\quad +a^9 + a^8 + a^7 - a^5 + a^4 - a^3 - a^2 + a)x \\
&\quad -a^{22} - a^{21} - a^{19} - a^{18} + a^{16} - a^{15} + a^{14} - a^{11} \\
&\quad -a^6 + a^3 + a^2 - 1
\end{aligned}$$

CHAPTER 6
ETHICAL ISSUES

Our expectation is that this research will yield generally positive ethical outcomes. The purpose of the research is to develop tools for extracting biological insight, such as knowledge of transcriptional regulatory networks from microarray experiments. Better knowledge of transcriptional proceses and memory formation will help us understand the molecular mechanisms underlying learning and memory, and help combat disease and psychiatric disorders.

One of the experiments described in this proposal rely on a data set derived from experiments on rats, and will likely involve further experiments prior to publication of a peer reviewed paper in the area. Some people consider experimentation on animals to be unethical. The performed and proposed experiments follow the established protocols for ethical use of the animal subjects, and the experiments are designed to eliminate unncessesary suffering and pain for the animals. Biological systems are extremely complex, and no other way of obtaining the understanding of the biological processes underlying memory and learning disorders exists, except for animal experimentation.

The second experiment, performed on fruit flies, is less troublesome, since the animals involved are invertebrates, but good laboratory practices were also employed.

In any case, accurate models of transcriptional processes, such as we wish to construct using the methods developed in this thesis proposal, could eventually allow *in-silico* simulations of complete experiments, reducing the number of real animals used in research. Note that this simulation technique is already utilized to perform nuclear arms testing, so it is not as infeasible as it sounds at first blush.

Two additional controversial topics related to the research above are genetic testing and genetic engineering. Microarrays are already used or proposed to diagnose certain conditions. The accumulation of more microarray data and advances in analysis techniques would allow for screening for susceptibility to certain conditions. Access to private genetic information needs to be strictly protected, and analysis routines have to be designed to neither cause undue distress through false positives, nor a false sense of security due to false negative results. The reverse engineering problem in microarrays intends to allow us to model cell processes leading to a condition. This knowledge could be used to target specific points in a regulatory network to prevent disease, or to produce a desired outcome including inducing disease or death.

The other major ethical issue is that efficient algorithms for reverse engineering genetic networks would likely be equally efficient at reverse engineering binary networks. Thus reverse engineering digital hardware or “black-box” software could be sped up by using our techniques. Similarly, many cryptographic protocols such as those protecting online commercial transactions are based on the difficulty of determining a specific key used in a cryptographic function to transform inputs into outputs. An efficient reverse engineering algorithm could be used to decrypt sensitive communications.

REFERENCE LIST

[1] T. Akutsu, S. Kuahara, O. Maruyama, and S. Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. *Proceedings of the 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[2] T. Akutsu, S. Miyano, and S. Kuhara. Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. *Pacific Symposium on Biocomputing*, 4:17–28, 1999.

[3] M. Aviñó, E. Green, and O. Moreno. Applications of finite fields to dynamical systems and reverse engineering problems. *Proceedings of the 19th ACM Symposium on Applied Computing - SAC*, 2004.

[4] D. Bollman, O. Colon-Reyes, and E. Orozco. Fixed points in discrete models for regulatory genetic networks. *EURASIP J Bioinform Syst Biol*, page 97356, 2007.

[5] D. Bollman, E. Orozco, and O. Moreno. A parallel solution to reverse engineering genetic networks. In *Lecture Notes in Computer Science*, volume 3045, pages 490–497. Springer-Verlag, 2004. Part III.

[6] A. J. Butte and I. S. Kohane. Mutual information relevance networks: functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput*, pages 418–429, 2000.

[7] R. Chiesa, H. G. Ortiz-Zuazaga, H. Ge, and S. Peña de Ortiz. Gene expression profiling in emotional learning with cDNA microarrays. In *40th meeting of the American Society for Cell Biology*, San Francisco, California, December 2000.

[8] L. M. Cope, R. A. Irizarry, H. A. Jaffee, Z. Wu, and T. P. Speed. A benchmark for affymetrix genechip expression measures. *Bioinformatics*, 20(3):323–331,

2004.

[9] H. de Jong. Modeling and simulation of genetic regulatory systems: A literature review. *Journal of Computational Biology*, 9(1):67–103, 2002.

[10] P. D'haeseleer, S. Liang, and R. Somogyi. Genetic network inference: from co-expression clustering to reverse engineering. *Bioinformatics*, 16(8):707–726, Aug 2000.

[11] M. Eisen, P. Spellman, D. Botstein, and P. Brown. Cluster analysis and display of genome-wide expression patterns. *Proceedings of National Academy of Science*, 95:14863–14867, 1998.

[12] H. Ge, R. Chiesa, and S. Peña de Ortiz. Hzf-3 expression in the amygdala after establishment of conditioned taste aversion. *Neuroscience*, 120(1):1–4, 2003.

[13] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, K. Hornik, T. Hothorn, W. Huber, S. Iacus, R. Irizarry, F. Leisch, C. Li, M. Maechler, A. J. Rossini, G. Sawitzki, C. Smith, G. Smyth, L. Tierney, J. Y. H. Yang, and J. Zhang. Bioconductor: Open software development for computational biology and bioinformatics. *Genome Biology*, 5:R80, 2004.

[14] E. L. Green. On polynomial solutions to reverse engineering problems. Preprint, 2004.

[15] S. Guiasu. *Information Theory with Applications.* McGraw-Hill, New York, 1977.

[16] T. Hubbard, D. Andrews, M. Caccamo, G. Cameron, Y. Chen, M. Clamp, L. Clarke, G. Coates, T. Cox, F. Cunningham, V. Curwen, T. Cutts, T. Down, R. Durbin, X. M. Fernandez-Suarez, J. Gilbert, M. Hammond, J. Herrero, H. Hotz, K. Howe, V. Iyer, K. Jekosch, A. Kahari, A. Kasprzyk, D. Keefe, S. Keenan, F. Kokocinsci, D. London, I. Longden, G. McVicker, C. Melsopp,

P. Meidl, S. Potter, G. Proctor, M. Rae, D. Rios, M. Schuster, S. Searle, J. Severin, G. Slater, D. Smedley, J. Smith, W. Spooner, A. Stabenau, J. Stalker, R. Storey, S. Trevanion, A. Ureta-Vidal, J. Vogel, S. White, C. Woodwark, and E. Birney. Ensembl 2005. *Nucleic Acids Res.*, 33:D447–D453, Jan 1 2005. Database issue.

[17] T. E. Ideker, V. Thorsson, and R. M. Karp. Discovery of regulatory interactions through perturbation: Inference and experimental design. *Pacific Symposium on Biocomputing*, 5:302–313, 2000.

[18] R. A. Irizarry, B. M. Bolstad, F. Collin, L. M. Cope, B. Hobbs, and T. P. Speed. Summaries of affymetrix genechip probe level data. *Nucleic Acids Res*, 31(4):e15, 2003.

[19] R. A. Irizarry, D. Warren, F. Spencer, I. F. Kim, S. Biswal, B. C. Frank, E. Gabrielson, J. G. N. Garcia, J. Geoghegan, G. Germino, C. Griffin, S. C. Hilmer, E. Hoffman, A. E. Jedlicka, E. Kawasaki, F. Martinez-Murillo, L. Morsberger, H. Lee, D. Petersen, J. Quackenbush, A. Scott, M. Wilson, Y. Yang, S. Q. Ye, and W. Yu. Multiple-laboratory comparison of microarray platforms. *Nat Methods*, 2(5):345–350, 2005.

[20] R. A. Irizarry, Z. Wu, and H. A. Jaffee. Comparison of Affymetrix GeneChip expression measures. *Bioinformatics*, 22(7):789–794, 2006.

[21] S. A. Kauffman. Metabolic stability and epigenesis in randomly constructed genetic nets. *J. Theor. Biol.*, 22:437–467, 1969.

[22] S. A. Kauffman. *The Origins of Order.* Oxford University Press, New York, Oxford, 1993.

[23] J. Kela, P. Salmi, R. Rimondini-Giorgini, M. Heilig, and C. Wahlestedt. Behavioural analysis of melanin-concentrating hormone in rats: evidence for orexigenic and anxiolytic properties. *Regul. Pept.*, 114(2–3):109–114, Jul 15 2003.

[24] W. P. Kuo, T.-K. Jenssen, A. J. Butte, L. Ohno-Machado, and I. S. Kohane. Analysis of matched mRNA measurements from two different microarray technologies. *Bioinformatics*, 18(3):405–412, 2002.

[25] R. Lamprecht, S. Hazvi, and Y. Dudai. cAMP response element-binding protein in the amygdala is required for long- but not short-term conditioned taste aversion memory. *J. Neurosci.*, 17:"8443–8450", 1997.

[26] R. Laubenbacher and B. Pareigis. Equivalence relations on finite dynamical systems. *Advances in Applied Mathematics*, 26:237–251, 2001.

[27] R. Laubenbacher and B. Stigler. Dynamic networks. *Adv. in Al. Math.*, 26:237–251, 2001.

[28] R. Laubenbacher and B. Stigler. A computational algebra approach to the reverse engineering of gene regulatory networks. *J. Theor. Biol.*, 229:523–537, 2004.

[29] T. Lee, N. Rinaldi, F. Robert, D. Odom, Z. Bar-Joseph, G. Gerber, N. Hannett, C. Harbison, C. Thompson, I. Simon, J. Zeitlinger, E. Jennings, H. Murray, D. Gordon, B. Ren, J. Wyrick, J. Tagne, T. Volkert, E. Fraenkel, D. Gifford, , and R. Young. Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 298:799–804, 2002.

[30] B. Lemon and R. Tjian. Orchestrated response: a symphony of transcription factors for gene control. *Genes and Development*, 14(20):2551–2569, October 2000.

[31] X. Q. Li, V. M. Verge, J. M. Johnston, and D. W. Zochodne. CGRP peptide and regenerating sensory axons. *J. Neuropathol. Exp. Neurol.*, 63(10):1092–1103, Oct 2004.

[32] S. Liang, S. Fuhrman, and R. Somogyi. REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Pac Symp Biocomput*, pages 18–29, 1998.

[33] H. Lodish, A. Berk, L. Zipursky, P. Matsudaira, D. Baltimore, and J. Darnell. *Molecular Cell Biology.* W. H. Freeman, 2000. Section 20.8.

[34] M. Merika and D. Thanos. Enhanceosomes. *Curr Opin Genet Dev*, 11(2):205–208, April 2001.

[35] O. Moreno, D. Bollman, and M. Aviñó. Finite dynamical systems, linear automata and finite fields. *2002 WSEAS Int. Conf. on System Science Alied Mathematics & Computer Science and Power Engineering Systems*, pages 1481–1483, 2002. Also to appear in the International Journal of Computer Research.

[36] D. A. Patterson and J. L. Hennessy. *Computer Organization and Design.* Morgan Kaufmann Publishers, San Francisco, 1997.

[37] M. Pereira-da Silva, M. Torsoni, H. Nourani, V. Augusto, C. Souza, A. Gasparetti, J. Carvalheira, G. Ventrucci, M. Marcondes, A. Cruz-Neto, M. Saad, A. Boschero, E. Carneiro, and L. Velloso. Hypothalamic melanin-concentrating hormone is induced by cold exposure and participates in the control of energy expenditure in rats. *Endocrinology*, 144(11):4831–4840, Nov 2003.

[38] Rat Genome Database Web Site. Rat genome data. Medical College of Wisconsin, Milwaukee, Wisconsin. World Wide Web (URL: `http://rgd.mcw.edu/`), Aug 2005.

[39] Y. Robles, P. E. Vivas, H. G. Ortiz-Zuazaga, Y. Felix, and S. Peña de Ortiz. Hippocampal gene expression profiling in spatial learning. *Neurobiology of Learning and Memory*, 80(1):80–95, Jul 2003.

[40] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235):467–470, Oct 20; 1995.

[41] J. Schug and G. C. Overton. Tess: Transcription element search software on the www. Technical report, Computational Biology and Informatics Laboratory,

School of Medicine, University of Pennsylvania, 1997. Technical Report CBIL-TR-1997-1001-v0.0.

[42] G. K. Smyth. Linear models and empirical Bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol*, 3:Article3, 2004.

[43] G. K. Smyth. Limma: linear models for microarray data. In R. Gentleman, V. Carey, S. Dudoit, R. Irizarry, and W. Huber, editors, *Bioinformatics and Computational Biology Solutions using R and Bioconductor*, pages 397–420. Springer, New York, 2005.

[44] T. Tully, T. Preat, S. C. Boynton, and M. Del Vecchio. Genetic dissection of consolidated memory in *Drosophila*. *Cell*, 79(1):35–47, 1994.

[45] M. Varas, M. Perez, O. Ramirez, and S. de Barioglio. Melanin concentrating hormone increase hippocampal synaptic transmission in the rat. *Peptides*, 23(1):151–155, Jan 2002.

[46] T. Yamamoto, T. Shimura, N. Sako, Y. Yasoshima, and N. Sakai. Neural substrates for conditioned taste aversion in the rat. *Behav. Brain Res.*, 65:1231–137, 1994.

[47] Y. Yasoshima, T. Shimura, and T. Yamamoto. Single unit responses of the amygdala after conditioned taste aversion in conscious rats. *Neuroreport*, 6:2424–2428, 1995.

[48] J. C. Yin, J. S. Wallach, M. Del Vecchio, E. L. Wilder, H. Zhou, W. G. Quinn, and T. Tully. Induction of a dominant negative CREB transgene specifically blocks long-term memory in *Drosophila*. *Cell*, 79(1):49–58, 1994.