IMPLEMENTATION OF DATA MANAGEMENT PORTION OF MIDAR PROJECT

By

Joshua J. Hornsby
B.S. University of Louisville, 2004

A Thesis
Submitted to the Faculty of the
University of Louisville
Speed School of Engineering
As Partial Fulfillment of the Requirements
For the Professional Degree

MASTER OF ENGINEERING

Department of Computer Engineering and Computer Science

May 2005

IMPLEMENTATION OF DATA MANAGEMENT PORTION OF MIDAR PROJECT

Submitted by: ______________________________
Joshua J. Hornsby

A Thesis Approved on

____
(Date)

by the Following Reading and Examination Committee:

____
Dr. Eric C. Rouchka, Thesis Director

____
Dr. Ahmed Desoky

____
Dr. Gail W. DePuy

____
Dr. Nigel G. F. Cooper

DEDICATION

This thesis is dedicated to my wife, Jessica Hornsby whose love and support made this possible, my mother and father who have given me thousands of reasons to be thankful for such great parents, and in memory of Mema, Aunt Marty, and Aunt Mona.

ACKNOWLEDGEMENTS

Many people have made this work possible. I would like to thank my wife, who has supported me throughout my education. Many thanks are owed to my parents, for making me the person I am today. I thank my Mom for forcing me to take a computer class in high school against my will. Many thanks are needed for Dr. Eric C. Rouchka, for the instruction and guidance provided during this long process. I would also like to thank the members of the thesis committee, Dr. Nigel G. F. Cooper, Dr. Ahmed Desoky, and Dr. Gail W. DePuy. Additional thanks go out to Tim Hardin, Elizabeth Cha, and the other members of the Bioinformatics Research Group. The work for this thesis was performed in the University of Louisville's Bioinformatics Laboratory. Equipment for the Bioinformatics lab is provided through the Kentucky Biomedical Research Infrastructure Network (NIH-NCRR grant #P20 RR16481; Nigel Cooper, PI)

ABSTRACT

Microarray analysis is an exciting new technology, developed at Stanford University in the early 1990s, which allows researchers to study the expression level of thousands of genes in a single experiment. Such experiments generate vast quantities of data that are not easily analyzed by hand. Adding to the complexity is a lack of a strict data standard as well as numerous formats that have been developed, such as Affymetrix, Agilent, Codelink, and even custom arrays. The need for data management and analysis tools is apparent when dealing with this new technology.

The design and implementation of a database and web interface is an integral portion of a data management system being developed by the Bioinformatics Research Group (BRG) at the University of Louisville as a collaborative effort between the J. B. Speed School of Engineering and the School of Medicine. The work in this thesis specifically addresses the portions of this project to create a data repository implemented in an Oracle database, and various tools needed to insert, extract, and view data in the system implemented through a web front end written in PHP and Perl. The Affymetrix technology is used as the format due to its complexity since it is widely used and well structured. A solid basis in this format will enable additional custom and commercial formats to be included in future development.

NOMENCLATURE[1]

Affymetrix: A company founded by Stephen P.A. Fodor, Ph.D. and others in the late 1980s with the revolutionary idea to use semiconductor manufacturing techniques to create GeneChips (an Affymetrix trademark) or generically DNA microarrays.

Agilent: Agilent Technologies has a robust research and development program as part of Agilent Laboratories, with active research in MEMS, nanotechnology, and Life Sciences.

Apache: An open source HTTP web server for Unix platforms (BSD, Linux, and UNIX systems), Microsoft Windows, and other platforms.

BASE: The BioArray Software Environment was developed at Department of Theoretical Physics, Lund University.

Bioinformatics: The use of techniques from applied mathematics, informatics, statistics, and computer science to solve biological problems.

CEL: An Affymetrix experiment result file.

CGI: An important World Wide Web technology that enables a client web browser to request data from a program executed on the Web server.

DBI: The most common database interface for the Perl programming language.

DNA: Deoxyribonucleic acid is a nucleic acid, which is capable of carrying genetic instructions for the biological development of all cellular forms of life and many viruses.

GD: A library by Thomas Boutell and others for dynamically manipulating images.

GeneChip: An Affymetrix trademark that refers to microarrays sold by that company (see Affymetrix).

LAD: The Longhorn Array Database is an open source version of the Stanford Microarray Database (see SMD).

[1] Most definitions are provided by www.wikipedia.org.

LIMS: Laboratory Information Management Software is computer software that is used in the laboratory for the management of samples, laboratory users, instruments, standards and other laboratory functions such as invoicing, plate management, stability lims, work flow automation.

MADAM: MicroArray Data Management is a software package available from The Institute for Genomic Research (see TIGR).

MAS 5.0: A statistical package for analyzing Affymetrix experiment results.

MD5: A widely used cryptographic hash function with a 128-bit hash value. As an Internet standard (RFC 1321), MD5 has been employed in a wide variety of security applications, and is also commonly used to check the integrity of files.

MIAME: The Minimal Information About a Microarray Experiment standard for describing a microarray experiment is being adopted by many journals as a requirement for the submission of papers based on microarray results.

Microarray: A piece of glass or plastic on which different molecules of DNA have been affixed at separate locations in an ordered manner thus forming a microscopic array.

mRNA: Messenger RNA is RNA that carries information from DNA to the ribosome sites of protein synthesis in the cell.

NAS: Network-attached storage systems are generally computing-storage devices that can be accessed over a computer network, rather than directly being connected to the computer (via a computer bus).

Object-Relational: Allows developers to integrate the database with their own custom data types and methods.

OCI: The Oracle Call Interface is a set of low-level APIs (Application programming interface calls) used to interact with Oracle databases.

Perl: A programming language released by Larry Wall on December 18, 1987 that borrows features from C, sed, awk, shell scripting (sh), and (to a lesser extent) from many other programming languages.

PHP: A widely used open-source programming language primarily for server-side applications and developing dynamic web content.

Probe: A DNA segment that has been affixed at separate locations in an ordered manner thus forming a microscopic array.

Query: A database query is often specified using the structured query language (see SQL) and is a specific request for information stored in the database.

Relational: A data model based on predicate logic and set theory.

SHA-1: The Secure Hash Algorithm family is a set of related cryptographic hash functions. The most commonly used function in the family, SHA-1, is employed in a large variety of popular security applications and protocols, including TLS, SSL, PGP, SSH, S/MIME, and IPSec.

SMD: The Stanford Microarray Database stores raw and normalized data from microarray experiments.

SQL: Structured Query Language is the most popular computer language used to create, modify and query databases.

SQL*LOADER: An oracle utility to quickly insert data into a table without using SQL statements.

SQL*PLUS: An oracle utility to directly query a database using the structured query language.

TIGR: The Institute of Genomic Research.

TLS: A cryptographic protocols, which provides secure communications on the Internet.

Transcription: Transcription is the process through which DNA is enzymatically converted into its complementary RNA.

Translation: Translation is the second process of protein biosynthesis (part of the overall process of gene expression). In translation, messenger RNA is decoded to produce a specific polypeptide according to the rules specified by the genetic code.

TABLE OF CONTENTS

LIST OF FIGURES

I. INTRODUCTION

Deoxyribonucleic acid (DNA) is the genetic blue print for life. DNA is found as a double helix in its most common form, where each strand is composed of four types of nucleotides, or bases chained together to form a polymer known as a DNA sequence. These bases are adenine, thymine, cytosine, and guanine abbreviated as A, T, C, and G. DNA can exist as a single strand of bases such as the following:

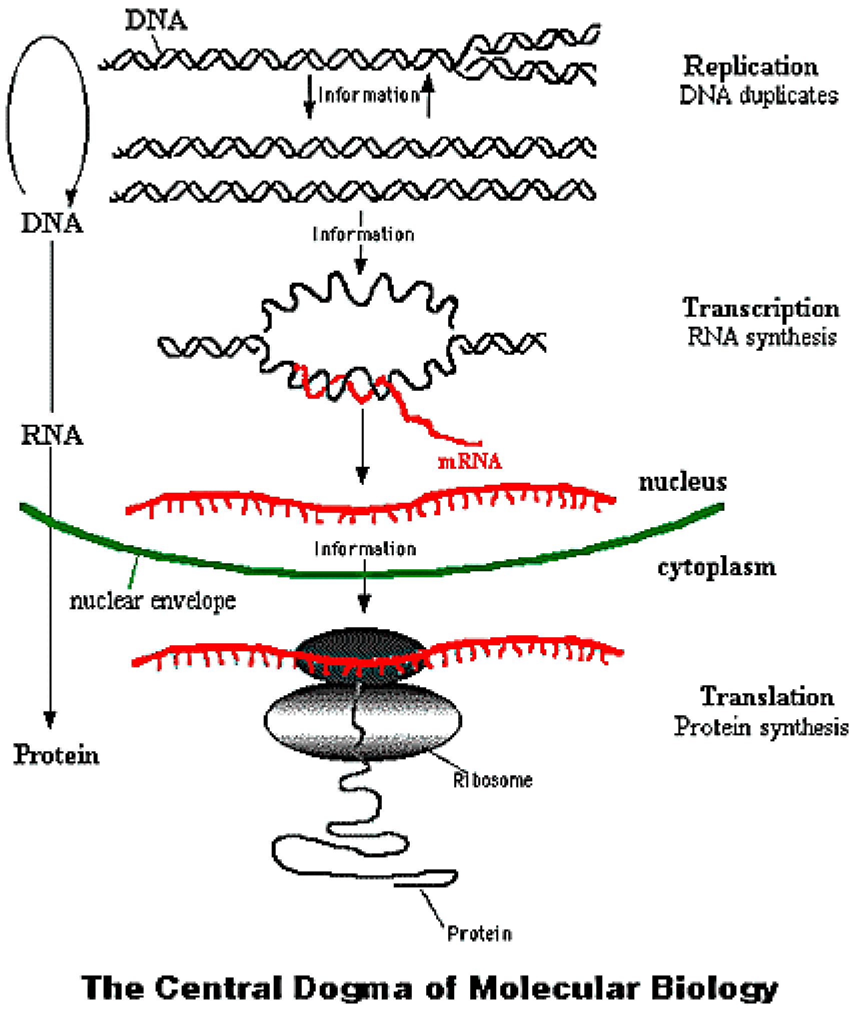
```
5′ G->T->A->A->A->G->T->C->C->C->G->T->T->A->G->C 3′
```

Each base has a compliment, with which it will hybridize or bond to form the complimentary strand. Base A will bond with a base T and vice versa, while bases C and G will also bond. DNA is double stranded by nature and will bond with available compliments if in a single stranded form. The resulting double stranded sequence for the above is as follows:

```
5′ G->T->A->A->A->G->T->C->C->C->G->T->T->A->G->C 3′
   |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
3′ C<-A<-T<-T<-T<-C<-A<-G<-G<-G<-C<-A<-A<-T<-C<-G 5′
```

The genome of an organism is the complete sequence of DNA contained within an individual cell. The human genome for instance contains approximately 3.2 billion base pairs spread out over 22 autosomes and 2 sex chromosomes.

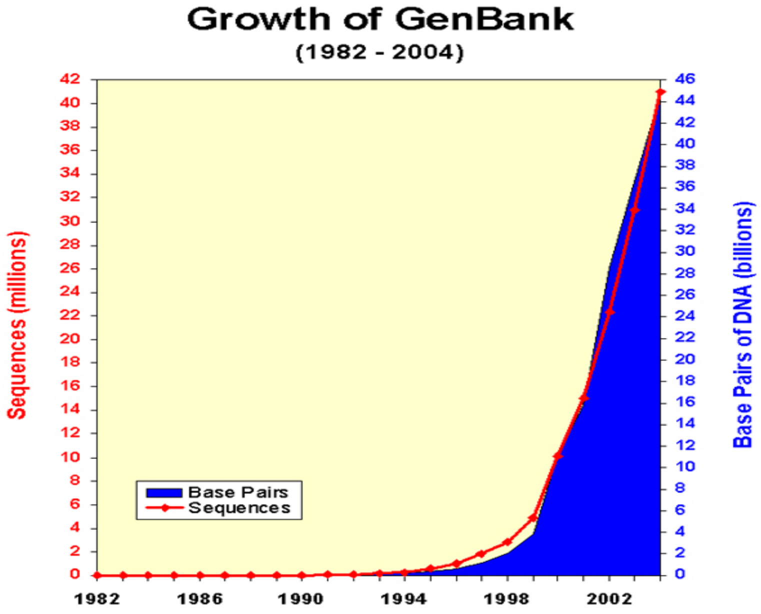
A small percentage of the human genome (approximately 2%) actually encodes genes. These genes are expressed through a process called the central dogma of molecular biology (see figure 1.1). The central dogma states simply that the region of a double stranded DNA molecule that corresponds to a gene is copied to a complementary single stranded mRNA molecule. The single stranded mRNA molecule then gets translated to a protein. If mRNA molecules can be identified, the expression level of the corresponding genes can be determined.



SOURCE: http://www.accessexcellence.org/RC/VL/GG/images/central.gif

FIGURE 1.1 – CENTRAL DOGMA OF MOLECULAR BIOLOGY

Bioinformatics or computational biology, as explained by Wikipedia (www.wikipedia.org), is the "use of techniques from applied mathematics, informatics, statistics, and computer science to solve biological problems". Some biological problems include sequence analysis, protein expression analysis, structure prediction, modeling biological systems, and gene expression analysis. The availability of biological data, including genomic data, has been growing exponentially over the past twenty years. For instance, various genome projects and improvements in sequencing technology have been implemented leading to a dramatic increase in DNA and protein sequences in GenBank[1] (see figure 1.2), which is a large central repository for biological data.



SOURCE: http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html

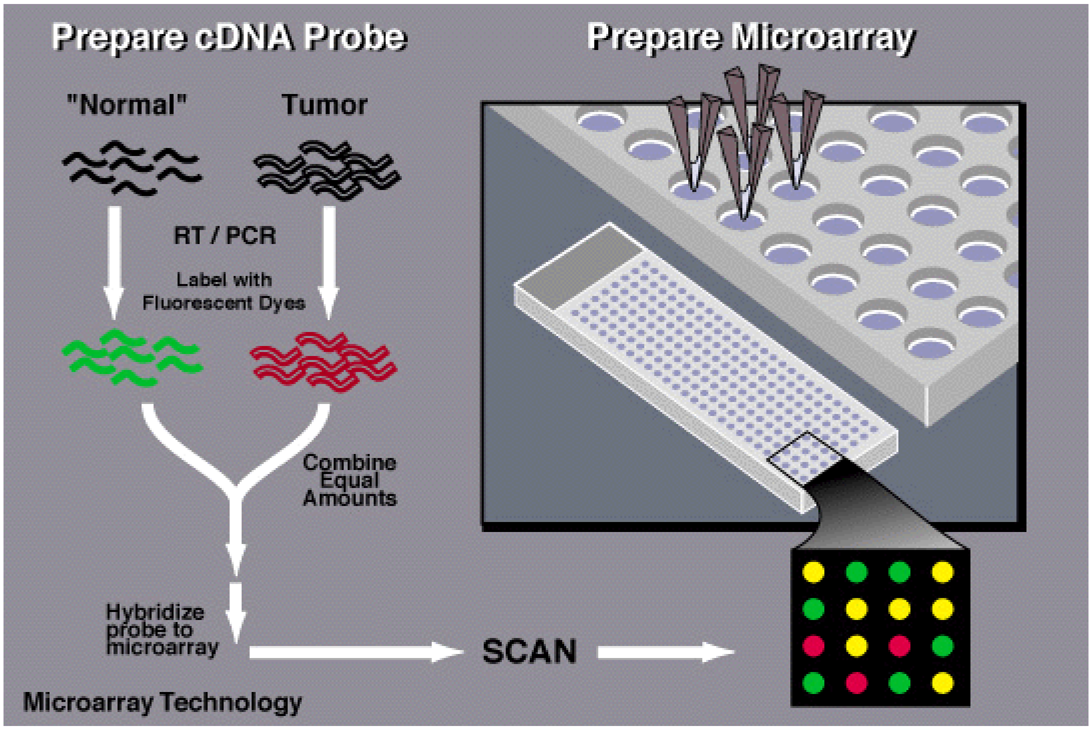
FIGURE 1.2 – GROWTH OF GENBANK

Due to the vast amounts of biological data available, the area of bioinformatics is closely related with computer engineering. New biological problems often mean new storage mechanisms, data structures, algorithms, or data mining techniques will be needed. At the University of Louisville, the Bioinformatics Research Group (BRG) is a joint collaboration between the Speed School of Engineering and the School of Medicine that has been set up to address bioinformatics issues. Researchers and students in the BRG are particularly interested in the analysis and storage of biological data resulting from a relatively new technology called microarrays[2] or gene chips. Microarrays were developed at Stanford University in the early 1990s[2]. This new technology allows researchers to study the expression level of thousands of genes in a single experiment.

A microarray is a slide, usually glass or plastic, on which potentially hundreds of thousands of sequences of DNA are affixed in a grid or array pattern. The DNA segments that are affixed to a microarray slide represent unique regions of genes. These sequences are single stranded, and are called probes. The single stranded probes on the microarray will hybridize, or bond, to their complementary mRNA sequences when they come in contact in order to form a double stranded sequence.

Samples of single-stranded mRNA molecules representing genes expressed under a certain condition are extracted from cells and labeled with dyes to enable detection with a microarray scanner. These mRNA samples are then washed over the microarray. When the labeled samples come into contact with the complementary sequences on the microarray, they will hybridize, causing the probe to also be bonded with the label, which will be detected when it is scanned with a microarray scanner.

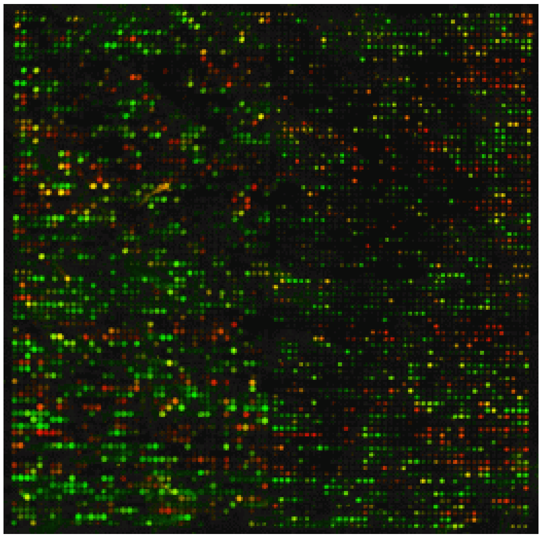
The intensity measured by the scanner indicates the amount of fluorescent dyed samples to which the probe has hybridized. This in turn is an indirect measure of the expression level of a gene under a particular condition. A high hybridization indicates a high expression level for the gene. A low hybridization means a low expression level. The conditions to be tested can be different types of tissue (brain versus liver), environments (presence or absence of light; healthy versus diseased), or time series analysis (such as the reaction after the administration of a drug at different time points).



SOURCE: http://www.accessexcellence.org/RC/VL/GG/microArray.html

FIGURE 1.3 – MICROARRAY PROCESS

Two different conditions may be studied by mixing the labeled mRNA from two different conditions, each with a different fluorescent color, usually red and green, on a single chip (see figures 1.3 and 1.4). With this technology, a researcher can determine the genes that are involved in a particular biological process. Figure 1.3 illustrates the process involved in analyzing a two-color microarray. Additionally, gene expression can be studied on one experimental condition per chip, where the results from two different chips are combined. The Affymetrix technology incorporates a single-color chip using biotin as the label. Members of the BRG are interested in using microarray analysis to discover the interaction of genes involved in birth defects and in neurological processes, such as aging (apoptosis) and Alzheimer's disease.



SOURCE: http://www.cs.unm.edu/~patrik/networks/microarray.gif

FIGURE 1.4 – SCANNED IMAGE OF A TWO-COLOR FLUORESCENT MICROARRAY

Vast amounts of data are being generated by microarray technologies. For example, a single Affymetrix (www.affymetrix.com) microarray chip can produce over 60 megabytes of raw gene expression data. In order to store and analyze this data, a data management system is needed. MiDaR (Microarray Database Resource) (Rouchka, et al, unpublished) is a multifaceted project that explores the implementation of a database management and analysis resource for a variety of custom and commercial microarrays. MiDaR will provide a secure system to store microarray experiments and ensure MIAME[3] compatibility. MiDaR is being developed at the University of Louisville by the Bioinformatics Research Group (BRG).

While the complete MiDaR system will include the ability to store and analyze microarray experiments from many custom and commercial packages, this portion of the project will focus on the storage and presentation of Affymetrix experiment results. Affymetrix gene chips represent each gene with a set of 25-mer perfect match oligonucleotides (oligos) and one-base mismatch oligo pairs. The mismatch oligo's middle (13th) position is complimentary to that of the perfect match oligo corresponding location. The use of mismatch oligos gives some indication of cross-hybridization that may occur with each match oligo. The set of oligos for each gene can then be combined to result in a single expression value for each gene (see figure 1.5). Affymetrix has a large number of chips including over twenty different genomes (www.affymetrix.com/products/arrays/index.affx).

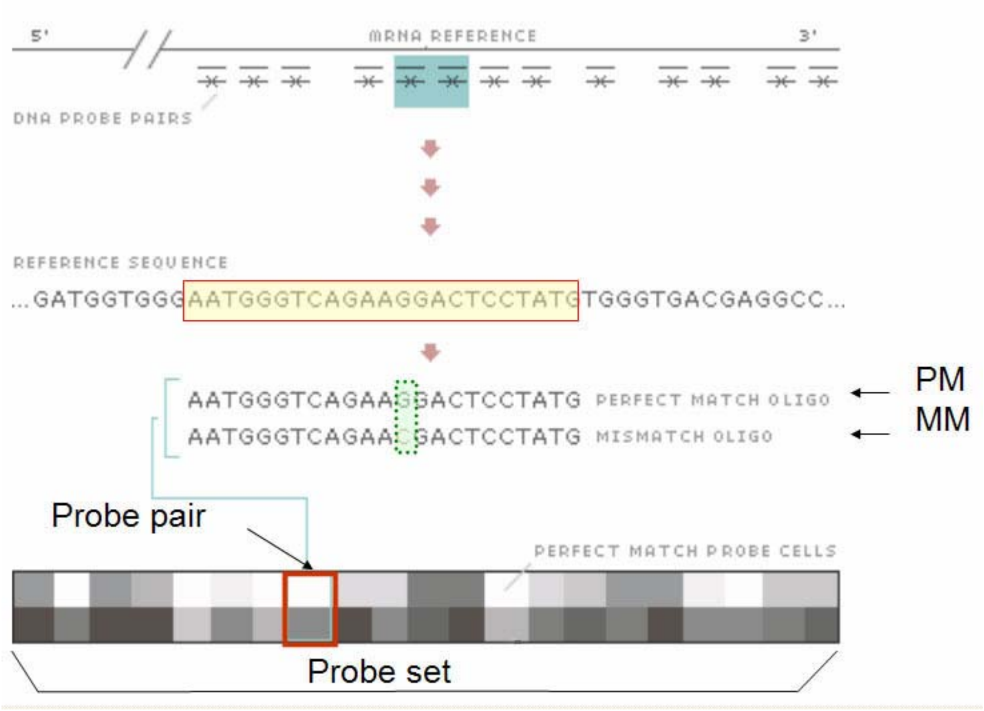
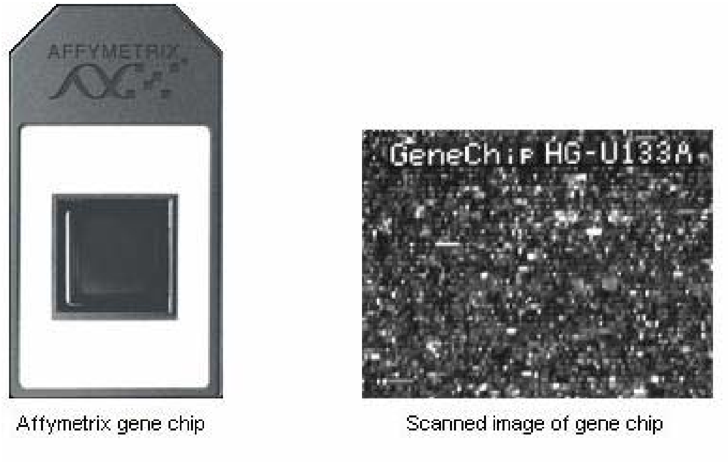

5'
MRNA REFERENCE
3'
DNA PROBE PAIRS
REFERENCE SEQUENCE
...GATGGTGGGAATGGGTCAGAAGGACTCCTATGTGGGTGACGAGGCC...
AATGGGTCAGAAGGACTCCTATG PERFECT MATCH OLIGO
AATGGGTCAGAACGACTCCTATG MISMATCH OLIGO
PM
MM
Probe pair
PERFECT MATCH PROBE CELLS
Probe set

FIGURE 1.5 – AFFYMETRIX PROBE PAIRS

A single Affymetrix chip is roughly the size of a glass microscope slide and contains over half a million spots, each of which represents a portion of a gene (see figure 1.6). After the chip is scanned and processed by specialized software that isolates spots and measures their intensity, experiment data is recorded into a large data file detailing the characteristics of each spot on the chip. The amount of data that is generated by one such experiment can be overwhelming to researchers who must analyze these large result sets. Many researchers attempt to analyze such data by hand, or by using common desktop tools such as Microsoft Excel. This process can be extremely prohibitive. An efficient data management system is needed to organize these important experiments and the vast amount of data that they generate. By using a data management system, data can be stored on a managed server with sufficient space for thousands of experiments and still provide easy access to specific information at a researcher's request.



SOURCE: http://www.genomenewsnetwork.org/articles/2004/07/23/sids3.jpg
SOURCE: http://www.rzpd.de/images/affy_genechip.jpg

FIGURE 1.6 – AFFYMETRIX CHIP AND IMAGE

Data presentation is an important aspect of a data management system and data analysis in general. Important data relationships can be obscured if too much or missed if too little of the data is presented. Allowing researchers to customize the data presentation gives them the opportunity to concentrate on specific data relationships. The data presentation is available over the web to allow multiple users to utilize the resources simultaneously. For those who wish to have more over the data, direct access should be provided so that SQL queries can be performed on the database.

II. LITERATURE REVIEW

2.1 Microarray Technology

Microarray chips from different technologies such as Agilent (www.chem.agilent.com) or Codelink (www.codelinkbioarrays.com) each contain different data formats that researchers must become familiar with in order to effectively use or compare experiments across multiple platforms. Even within a single technology platform, several data formats are often used to store the various types of data that are reported. Affymetrix experiment data is contained in several file types, each with different formats. A single Affymetrix experiment yields many files such as CHP, CEL, CDF, DAT, RPT, EXP, and others (www.affymetrix.com/support/developer/AffxFileFormats.zip). These files report information on experiment procedures, parameters, chip layout, and probe set analysis. Researchers must know where data is reported and in what format it will be stored. An efficient data management system is needed to organize these important experiments and the vast amount of data that they generate.

2.1.1 Efficient Data Management

An efficient data management system can enable researchers to access data in a number of important ways. What, How, and When are important considerations in a data management system. Controlling what data is retrieved and how it is presented gives flexibility to researchers who would otherwise have to filter the large datasets and then reformat them to obtain a similar result. When the results are returned is also important. Faster data retrieval can be translated into faster research. In addition to providing these features the system must be user friendly. Ease of use is an important feature if the system is to be fully utilized.

A relational database can be used to store microarray data results. Data retrieval will be customizable and fast. A database is only a part of a complete data management solution though. Additional applications are needed to interact with the database to accomplish all of the requirements of an efficient data management system that are not handled by the database alone. Database applications can be used to facilitate new experiment results being added to the database or to extract original experiment result files. Additional flexibility in displaying data can also be added. Most importantly ease of use can be greatly increased. Without database applications every researcher would be required to learn Standard Query Language (SQL)[4] as well as become intimately familiar with the database schema. A scientist wants to spend time on analyzing data and not on learning how to use the tools. Database applications can separate the user from the database and encapsulate the database access logic.

2.1.2 MIAME Standard

Due to differences in experimental conditions, variations in protocols, and differences in machinery, it is not sufficient to store only the resulting raw data from microarray experiments. Additional information pertaining to the experiment’s procedures allows other researchers to analyze the results in a more contextual way.

When the details of an experiment are provided with the results, researchers can make more meaningful conclusions about the data and can attempt to replicate experiments. In an effort to ensure the relevance of microarray data sets a standard has been proposed. The Minimal Information to Annotate a Microarray Experiment (MIAME) standard ensures that enough information is provided with experiment results to replicate the experiment at a later date. Several derivatives have been developed to handle different specialized data. MIAME/Tox (www.mged.org/MIAME1.1-DenverDraft.DOC) and MIAME/Env (envgen.nox.ac.uk/miame/MIAME1.6-envDraft-2.pdf) have been proposed as the standards for toxicology and environmental genomics experiments respectively. Currently the MIAME guidelines are still in draft form but many microarray software packages have already begun to conform to the preliminary guidelines. Once the standard is finalized and adopted, researchers will be able to compare experiment results with other similar experiments.

2.2 Microarray Databases

Since the introduction of microarray technology, several software packages have become available to address the need for data management systems. Commercial applications as well as open source solutions are available with a wide range of functionality. To determine which of these available solutions would best suit a researcher's particular needs, an analysis of their various features and limitations is needed. Of the many solutions available a few of them stand out as possible candidates and are used much more commonly than the rest. A brief review will be done on the requirements and three of the possible solutions, which include BioArray Software Environment (BASE)[5], MicroArray Data Manager (MADAM)[6], and the Longhorn Array Database (LAD)[7].

2.2.1 Data Management System Requirements

In order to fairly evaluate the software currently available a set of requirements needed must be specified in order to fulfill a researcher's needs. Measurements based on these criteria will make certain that any data management system chosen will meet all of the stated goals. The system must:

1. Straightforwardly handle Affymetrix data.
2. Have an interface that is uncomplicated and user friendly.
3. Support multiple users securely.
4. Be expandable for further feature additions and different microarray platforms.
5. Have an interface implemented as a web browser accessible application.

The first system requirement ensures that any system chosen will support Affymetrix data that researchers will generate during experiments. Secondly any system chosen must be easy to use for researchers who specialize in biological data and not computer science. Since multiple researchers working on many experiments will need the system, multiple users need to be supported. Additional features enabling groups and group level permissions would also prove useful. Collaboration among researchers in groups could be facilitated with virtual meeting places. Group administrators would be needed to allow self-management of individual groups. The ability to add additional functionality quickly and easily is also important in any scientific research field where new statistical methods are developed constantly. Finally it is desirable for the system interface to be implemented as a web based application accessible via a standard browser so as to be accessible from anywhere and on any computing platform. No software is required on the client side beyond the standard browser available on any platform. Updates to the system are instantaneously available via the Internet with no need for patches or new versions to be installed on every client.

With these requirements an examination will be done on currently available products. Three options will be studied as possible solutions. The BioArray Software Environment, MicroArray Data Management, and Longhorn Array Database are the open source systems we chose as possible solutions for further analysis.

2.2.2 BioArray Software Environment

The BioArray Software Environment was developed at Department of Theoretical Physics, Lund University[5]. BASE was developed using free software to reduce the total cost of ownership. Using open source solutions such as Linux OS, MySQL database, and the Apache web server means the only cost will be hardware related. BASE functions are written in PHP, Java, JavaScript, and C++. The user interface is web based enabling multiple users of the same system over the network or Internet. The web interface has user logins to control access to the system. BASE also touts a host of available features broken into four categories. User administration, array production LIMS, Biomaterials, and data analysis capabilities allow great control over experiment results. BASE allows data to be visualized in a variety of ways including plots, histograms and tables. BASE is compatible with one and two-color systems and on most types of array platforms and data types including Affymetrix.

2.2.3 MicroArray Data Management

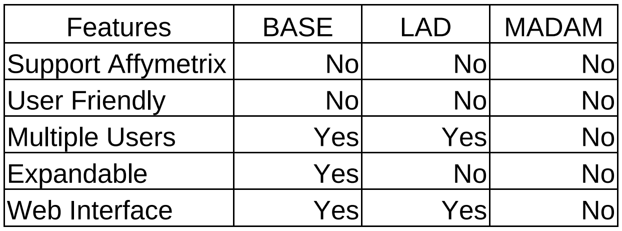
The MicroArray Data Management (MADAM) is a software package available from The Institute for Genomic Research (TIGR)[6]. MADAM is platform independent since it is written in Java and utilizes the open source database MySQL. MADAM has been tested on Microsoft Windows, Linux, Unix, and Mac OS X successfully. Additional tools are available to be integrated with the MADAM package, together making the TM4 suite. TIGR Spotfinder, Microarray Data Analysis System (MIDAS), and Multiexperiment Viewer (MeV) add many features to MADAM. Spotfinder is implemented in C/C++ and is only available on Windows systems, eliminating the advantage of platform independence the other three applications in the suite possess. Software tools included with TM4 were developed for spotted two-color custom microarrays, but can be adapted to work with single-color formats such as Affymetrix with some modifications. MADAM is also not network compatible and does not support multiple users.

2.2.4 Longhorn Array Database

The Longhorn Array Database (LAD) is an open source version of the Stanford Microarray Database (SMD)[7]. LAD is a MIAME compliant database that operates on Linux OS, and utilizes PostgreSQL, a robust open source database. LAD stores raw and normalized data from microarray experiments, as well as their corresponding image files. LAD uses a web interface which supports multiple users and gives researchers the freedom to access data, perform analysis, and visualization from anywhere on the Internet. LAD makes all of its functionality web browser accessible using Perl, GD, CGI, and DBI. In addition libgd, libjpg, libtiff, libxpm, libfreetype, zlib, libpng, netpbm, and ImageMagick packages are needed by the system to operate. Currently only Apache version 1.x is supported. A newer version, Apache 2.x, is not compatible at this time. LAD provides storage and analysis of two-color custom microarray data. The Affymetrix platform is not supported.

2.2.5 Custom In-House Solution

After reviewing the available solutions, it is apparent that none are a perfect match for the requirements stated. The BASE system supports multiple users and the addition of new features as well as being accessible from a web browser. The user interface is not very intuitive to most users and Affymetrix data is compatible but only in the technical sense, which only further convolutes its usability. MADAM is not browser accessible or usable by multiple users. Software tools must be adapted for use with Affymetrix data. LAD is a web based application but is only capable of storing and analyzing two-color microarray data. With no support for Affymetrix data LAD is not a viable solution either. Figure 2.1 indicates that none of the solutions reviewed completely satisfy all the requirements.



Features	BASE	LAD	MADAM
Support Affymetrix	No	No	No
User Friendly	No	No	No
Multiple Users	Yes	Yes	No
Expandable	Yes	No	No
Web Interface	Yes	Yes	No

FIGURE 2.1 – FEATURES COMPARISON

In order to meet all of the requirements put forth, a custom solution needs to be implemented in house. A system to meet these requirements has been designed and implemented. The design and implementation process will be discussed in detail in Chapter III and Chapter IV.

III. DESIGN

3.1 Database Package

Since the database will be the major component in the custom data management system, careful consideration should go into the database selection. A couple of options during the review of other data management systems have already been examined. A review will be made of the two open source options have previously discussed, MySQL and PostgreSQL, as well as the commercial package Oracle.

3.1.1 MySQL

MySQL is an open source database with the option for commercial support. MySQL is the database that is used in the MADAM and BASE systems. The database has been designed for speed, which would be useful in large transactions. Yahoo utilizes the MySQL database on its site for directory listings and financial services[8]. MySQL is currently the most widely installed database with about four million active installations[8], in part due to its inclusion in most Linux distributions. Some noticeable drawbacks with the MySQL database are its lack of features, though they have made in effort to address this by including support for stored procedures in their latest release, MySQL 5.0. Features found in other databases, which are not supported currently in MySQL, are constraints, sub-queries, views, cursors and objects.

3.1.2 PostgreSQL

Another open source database, PostgreSQL, was originally created as a research project at the University of California at Berkeley[8]. PostgreSQL is released under the Berkeley Software Distribution (BSD) license. The LAD system utilizes this database package. PostgreSQL is feature rich and has implemented most of the major features found in Oracle and other major database vendors. Despite its features, PostgreSQL falls short on performance in large transactions and heavy concurrent user loads. Development of PostgreSQL has shifted focus to improving performance, but currently cannot be used where performance is necessary on a large volume of data with large transactions.

3.1.3 Oracle

Oracle database is the most widely used commercial database package in the world[8]. Oracle supports a tremendous amount of advance features including stored procedures and triggers. The most recent versions of Oracle have incorporated regular expression pattern matching, which makes it very appealing to life science researchers[9]. Oracle is also an object relational database. Object relational databases are similar to the standard relational database, but provide support for complex object types that can be used in addition to the common data types such as ints, floats, and chars. Oracle's performance is also very high and easily supports large transactions and a high concurrent user load. Oracle's performance and features do carry a price tag, which like its capabilities are top among its peers. Such a high price can be prohibitive in some circumstances. An educational version of Oracle is available, although it is not the most recent release. The University of Louisville currently has an educational license to Oracle 9i and its installation is already locally available. Since Oracle offers such a large number of features and still maintains a high level of performance it is an obvious choice in a situation where it can be obtained and utilized at no charge. Having the database available on a central server eliminates the need for users to design and maintain separate databases each for their own personal use.

3.2 Database Design

Oracle 9i is the educational licensed version of Oracle that is available at the University of Louisville. Some advanced features of Oracle 9i that are of interest include objects and nested tables. Nested tables and objects allow a table to be stored in the record of another table as an object type. This type of object relational design can be used to store data encapsulation relationships. To determine if this approach should be taken, a comparison between the object relational design and solutions from the more traditional relational design is performed.

3.2.1 Relational Approach

The relational approach requires two tables. A parent table will contain details and information about an Affymetrix experiment in a record. Each record would contain a unique chip ID to identify the Affymetrix chip that was used in the experiment. The child table will contain actual measurement data from each spot on a chip using the same unique chip ID from the parent table as well as the X position and Y position to uniquely identify each record. This will be a one-to-many relationship where one parent record will be related to many child records, and each child record is related to only one parent record.

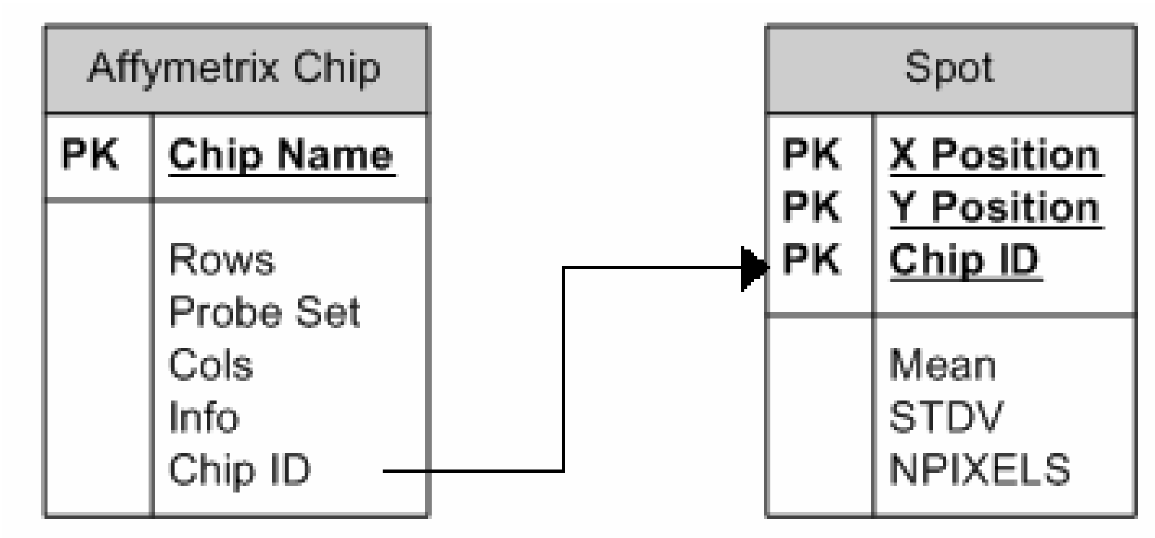
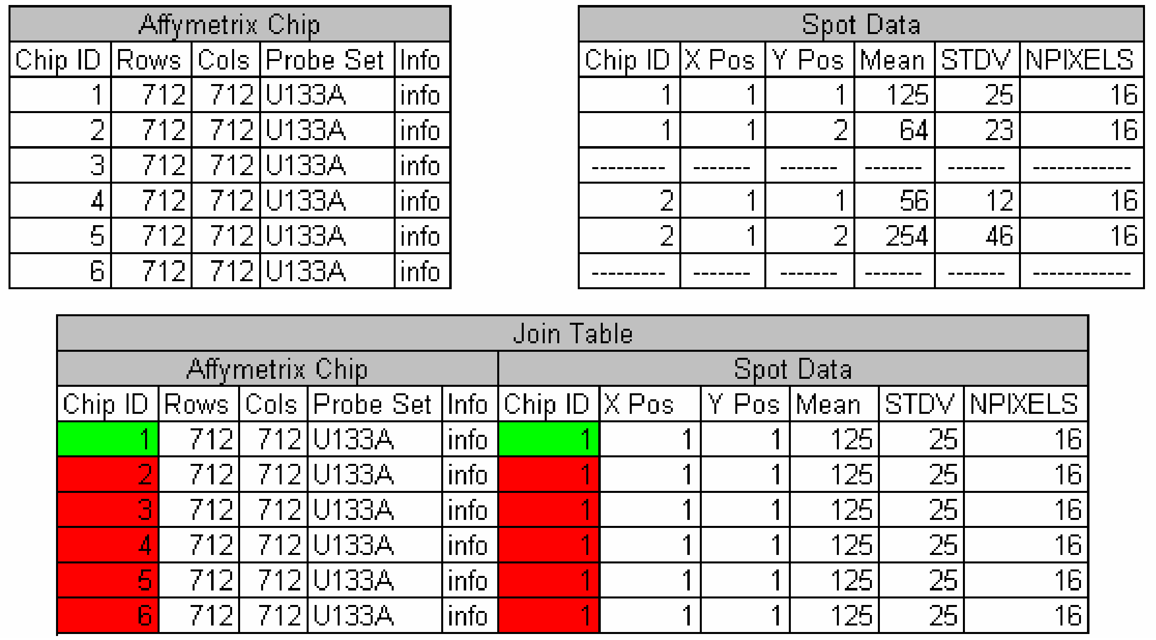


FIGURE 3.1 – RELATIONAL APPROACH

In this approach spot data can be related to its parent Affymetrix chip by performing a table join. A table join can be an expensive operation on large tables. When a table join is performed, every possible combination of rows in each table is made even though the combinations may not be valid associations. Invalid results are then discarded based on the where conditions provided by the query. This solution will not scale well in large tables due to the generation of invalid records. In the below example a table join would result in over 18 million record combinations, of which only about three million records are valid. The Affymetrix Chip table has 6 records. Each Affymetrix Human U133A chip contains 712 rows and 712 columns, which is equates to 506,944 individual spots per chip. The Spot Data table has been abbreviated here for space since it would contain 506,944 spots per chip for all six chips, or 3,041,664 records. The table join will incorrectly associate all three million plus of these records with all six chips even though a spot can only be on one chip. Some of the invalid records have been shown with the conflicting chip ID highlighted.



Affymetrix Chip				
Chip ID	Rows	Cols	Probe Set	Info
1	712	712	U133A	info
2	712	712	U133A	info
3	712	712	U133A	info
4	712	712	U133A	info
5	712	712	U133A	info
6	712	712	U133A	info

Spot Data					
Chip ID	X Pos	Y Pos	Mean	STDV	NPIXELS
1	1	1	125	25	16
1	1	2	64	23	16
---------	-------	-------	-------	-------	------------
2	1	1	56	12	16
2	1	2	254	46	16
---------	-------	-------	-------	-------	------------

Join Table										
Affymetrix Chip					Spot Data					
Chip ID	Rows	Cols	Probe Set	Info	Chip ID	X Pos	Y Pos	Mean	STDV	NPIXELS
1	712	712	U133A	info	1	1	1	125	25	16
2	712	712	U133A	info	1	1	1	125	25	16
3	712	712	U133A	info	1	1	1	125	25	16
4	712	712	U133A	info	1	1	1	125	25	16
5	712	712	U133A	info	1	1	1	125	25	16
6	712	712	U133A	info	1	1	1	125	25	16

FIGURE 3.2 – TABLE JOIN EXAMPLE

3.2.2 Object Relational Approach

The object relational approach will require one table that has another table nested in each of its rows. No table joins will be necessary since each record will already contain the associated spot data. Each spot on an Affymetrix chip is a single record in the nested table. The records contain the logical location in terms of an X position and Y position on the chip as well as the mean spot intensity, standard deviation of the intensity, and number of pixels used to scan each spot (typically 16). This table is then inserted as an object in the parent table, which describes the Affymetrix chip and experiment in more detail. Using an object-relational design will require the use of more advanced features of SQL and will extend the learning curve when new users learn to use the system. To some degree table views can be created to alleviate some of this burden by making the use of nested tables more transparent yet still utilize its benefits.

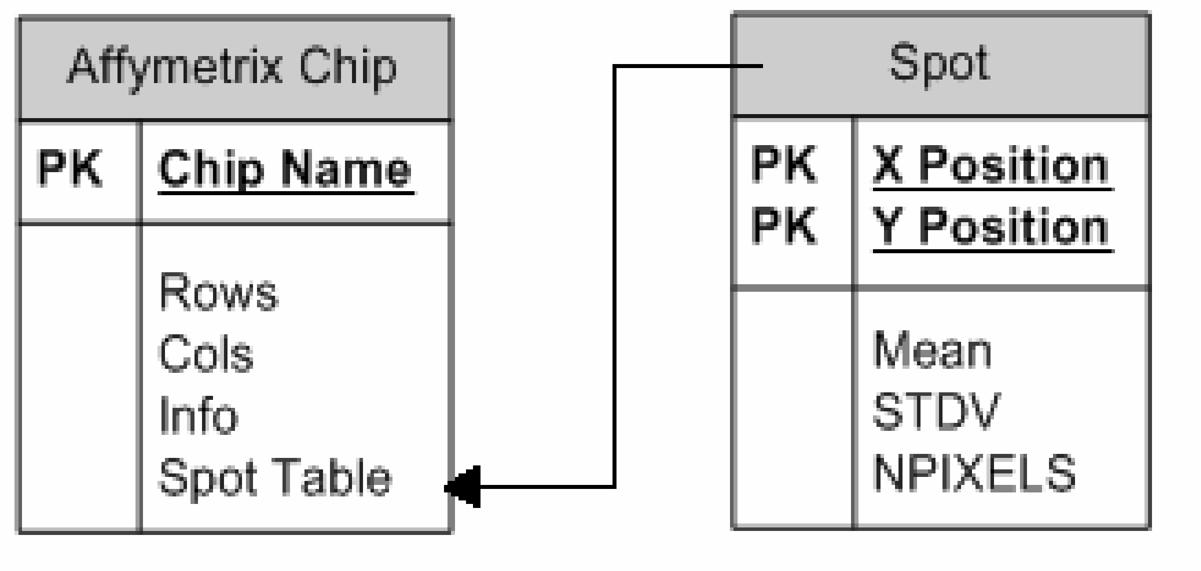


FIGURE 3.3 – NESTED TABLE

3.3 Database Schema

The database schema for this project takes the object relational approach due to its advantages over the traditional relational schema. In addition to the Affymetrix chip and its nested spot data table, another table is also used to identify which Affymetrix probe is located at each spot on a certain chip. This enables spot data to be correlated to an actual gene sequence. These tables will store a complete Affymetrix CEL file and the original CEL file can be recreated if it is needed in the future. This table will have a many-to-one relationship to the Affymetrix chip table, and therefore will not be nested. A many-to-one relationship means that each parent record will be associated with only child record, but each child record will be associated with many parent records. Nesting this relationship would require the child table to be replicated many times in each parent record and introduce many maintenance problems as well as redundant data wasting storage space.

3.4 Database Application Design

In order to meet our user-friendly ease of use guidelines some database applications will be necessary to handle the user's interaction with the data in the database. Applications will be needed to insert user's data, allow users to view and mine data, and extract or recreate original data formats. Common queries that will be performed by researchers must be made available with out the need for SQL

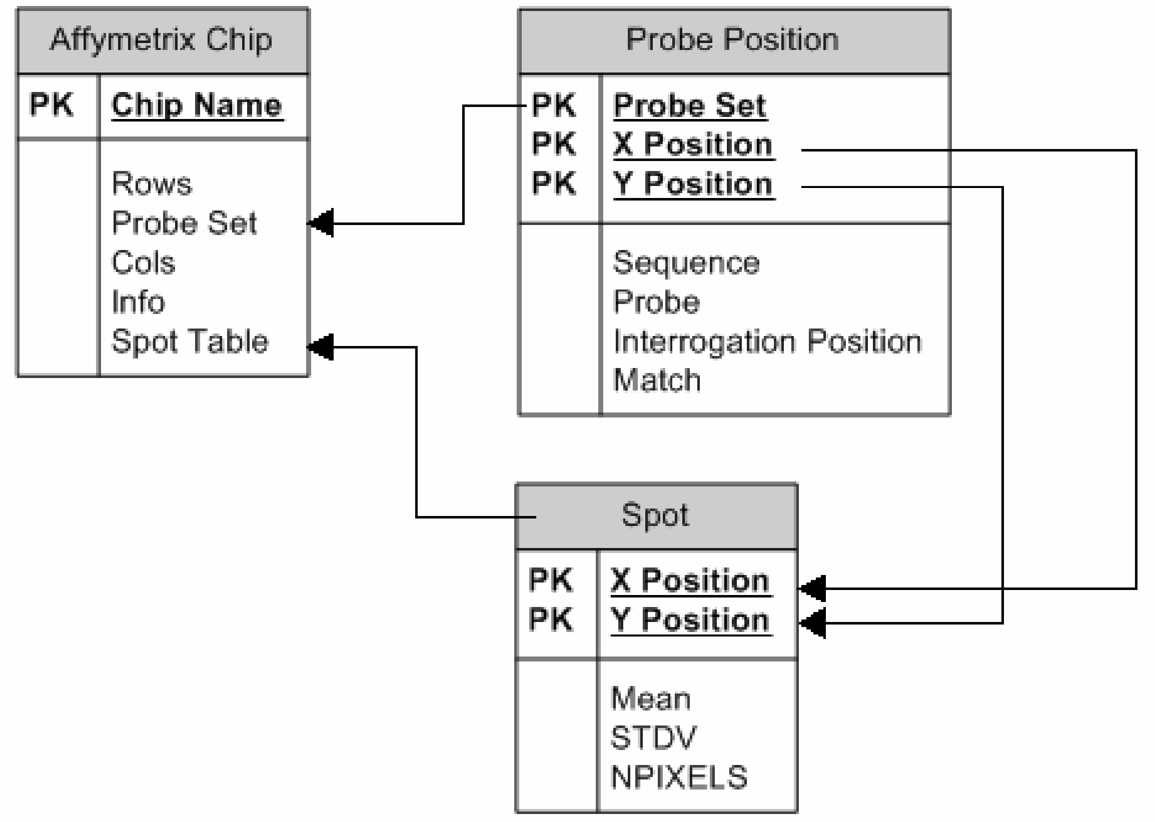


FIGURE 3.3 – INITIAL DESIGN SCHEMA

programming knowledge. These applications can also be used to pipeline data into other applications that already exist to analyze the data such as Bioconductor[10] and the MAS5.0 (www.affymetrix.com) statistical packages, which have recently been made open source. These tools should all be contained in one package for ease of use. Web based applications are used due to several advantages. Updates are available instantaneously instead of having patches and updates deployed. Computing power is centralized and can be maximized at a cheaper cost. Tech support can be done in one place rather than at each client's site.

3.5 Technology Choices

In addition to the choice of database package, it is necessary to determine additional tools, which will be needed to implement our data management system. Since Oracle is locally available on a Linux cluster with available computing power, we will utilize this system and must constrain our choices to Linux technologies. Using this cluster will provide plenty of computing power. Additionally a two-terabyte network attached storage (NAS) drive will provide the space required to store the large quantities of data that microarrays generate. At an average of less than twelve megabytes per Affymetrix chip, this system will hold over 83 thousand chips. During the design phase, it was decided, to implement the database applications as a web application. The Linux cluster has an Apache web server, which will be suitable to the end user's needs. The website will be written in PHP and Perl due to their capabilities in handling bioinformatics data. Additionally these languages are widely known by bioinformatics scientists. Using tools that are familiar to those in the field will allow for continued development in the future. Data manipulation tools will be written Perl while the website and flow control will be written in PHP.

IV. IMPLEMENTATION

4.1 Database Implementation

The Oracle database was implemented using SQL and an Oracle utility SQL*Plus. SQL*Plus is a simple command line connection utility that allows for direct interaction with the Oracle database using SQL syntax. The SQL commands that were used to create the database schema are as follows:

```
create table probe_position(
  filename varchar2(40),
  probe     varchar2(40),
  xpos number,
  ypos number,
  interrogation_pos number,
  sequence varchar2(30),
  strandedness   varchar2(10),
  match  varchar2(8)
  primary key(filename,xpos,ypos));

create or replace type cel_data as object(
  xpos  number,
  ypos  number,
  mean  float(126),
  stdv  float(126),
  npixels  number);

create type cel_data_nt as table of cel_data;

create table cel_info(
  filename  varchar2(30),
  cols  number,
  rows_num number,
  offsetx  number,
  offsety  number,
  cornerul  varchar2(20),
  cornerur  varchar2(20),
  cornerlr  varchar2(20),
  cornerll  varchar2(20),
  invertx  number,
  inverty number,
  swapxy number,
  datheader  varchar2(300),
  probeset   varchar2(30),
  algorithm varchar2(30),
  alg_param  varchar2(150),
  tail  clob,
  data cel_data_nt
  primary key(filename)
  nested table data store as cel_data_nt_tab);
```

While the first table, probe_position, is straightforward to create using the standard SQL create syntax, the second table, cel_info is more complicated. Since cel_info contains a nested table, the inner table must be created first. This is done by first creating an object, cel_data, which contains the data fields necessary. A table type with the name cel_data_nt is then created from the object. Once the inner table has been created the outer table, cel_info, can be created with a modified create statement. The nested table is listed as an ordinary column with the data type set to the table type previously created, cel_data_nt. Finally the nested table is stored physically as cel_data_nt_tab within the oracle database.

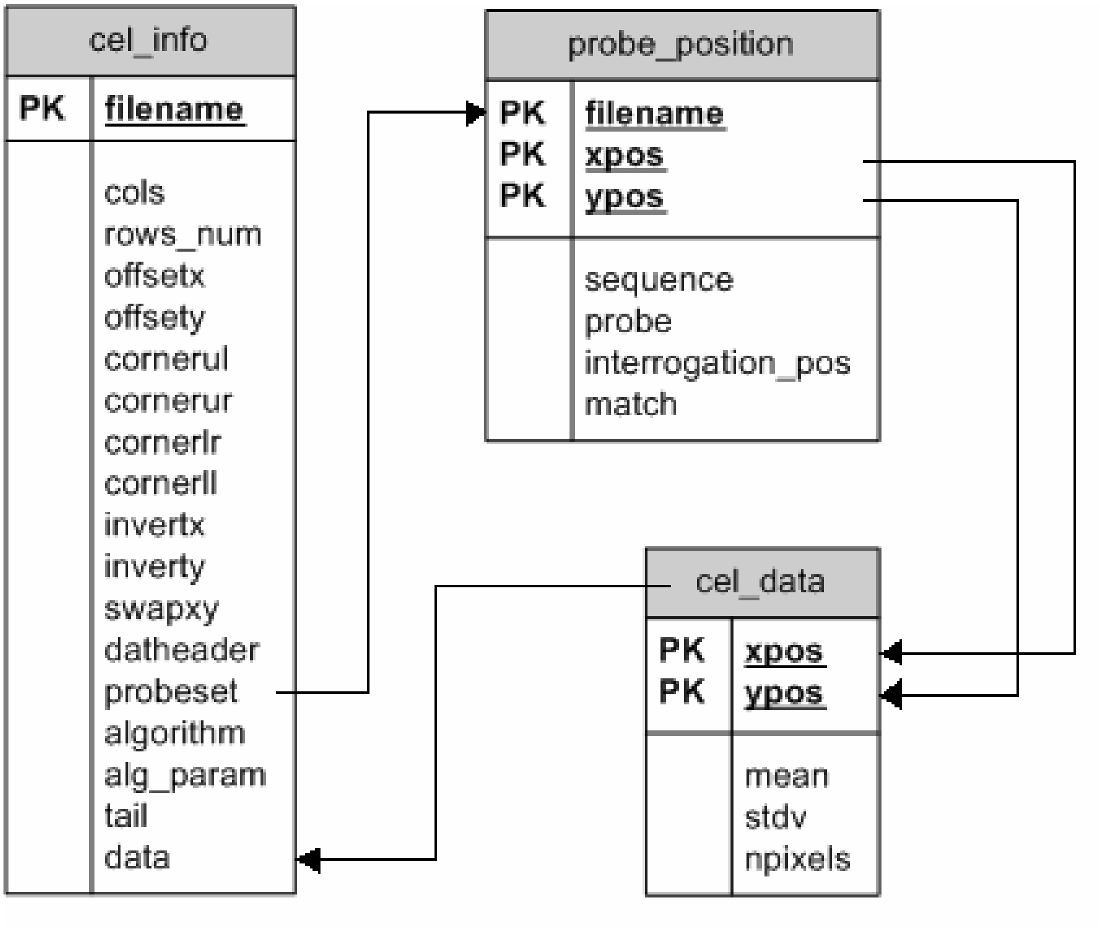


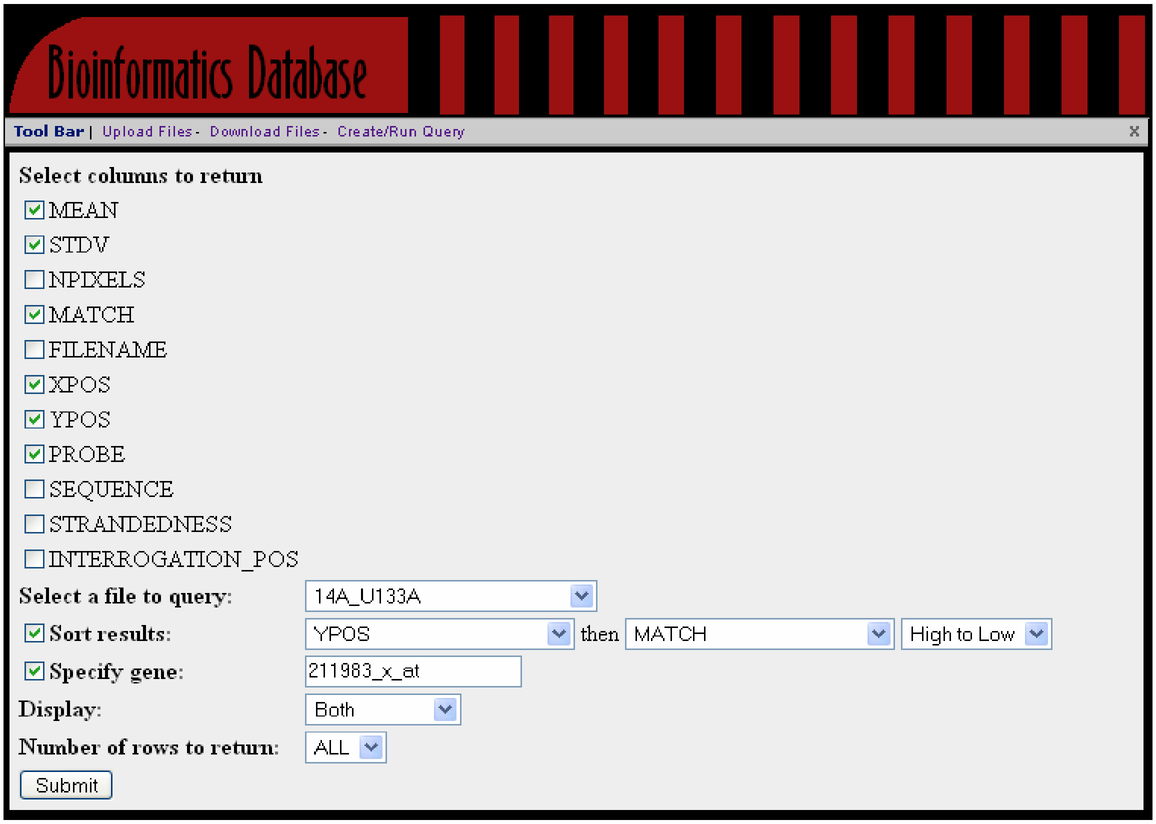
FIGURE 4.1 – DATABASE SCHEMA

4.2 Database Application Implementation

Once the database had been implemented some utilities were implemented to import data from Affymetrix CEL files and to recreate original CEL files from data already in the database. These applications are written in Perl and utilize the Oracle Call Interface (OCI). OCI handles connections to an Oracle database in Perl, and allows SQL statements to be executed. Another Oracle utility is also used to import data into the probe_position table. SQL*Loader (sqlldr) is a tool that performs high performance data loads. Using this utility in large data sets will perform much faster when compared to ordinary SQL insert commands. Data is read from a file and inserted into the table. A control file is used to specify the format of the data. A delimiter is used to separate data fields and the column order is indicated in the control file.

4.3 Web Access Front End

A user-friendly front end is implemented in PHP, an open source technology freely available on the net. The web site provides access to the data that researchers need without the need for knowledge of the SQL language. Those with knowledge of SQL, however, can still access the database using SQL*PLUS and have complete control over



Select columns to return

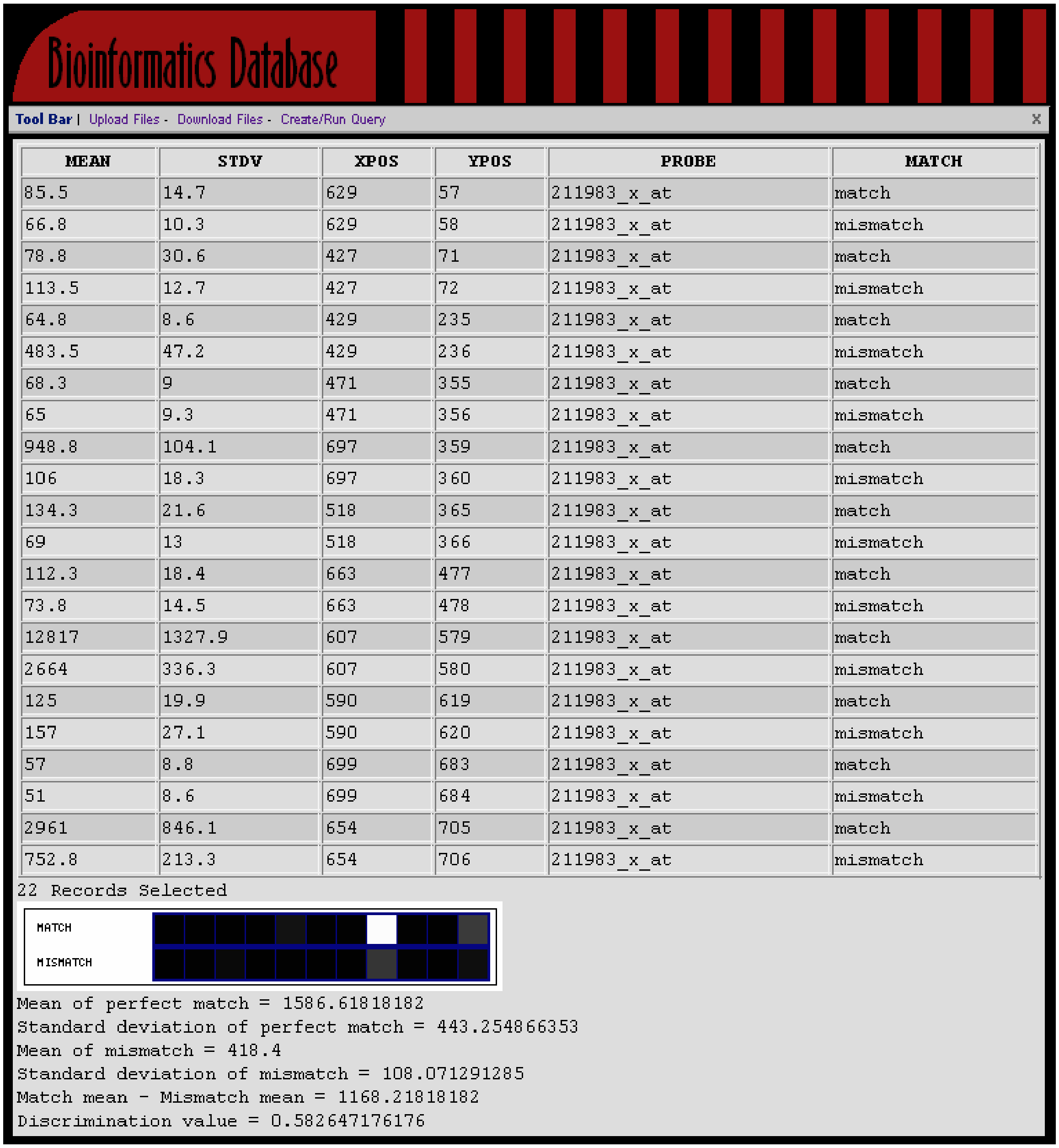
☑ MEAN
☑ STDV
☐ NPIXELS
☑ MATCH
☐ FILENAME
☑ XPOS
☑ YPOS
☑ PROBE
☐ SEQUENCE
☐ STRANDEDNESS
☐ INTERROGATION_POS

Select a file to query: 14A_U133A
☑ Sort results: YPOS then MATCH High to Low
☑ Specify gene: 211983_x_at
Display: Both
Number of rows to return: ALL

FIGURE 4.2 – CREATE QUERY SCREEN

the types of queries they would like to perform. After logging in on the website's home page users will have access to the database through a set of tools designed to build queries for the user. Queries can be built using the drop down selection boxes and check boxes that indicate the type of information and format a researcher desires. As a result of the sample query shown (figure 4.2), the selected columns will be chosen from each record in the database where the CEL file name is 14A_U133A and the Affymetrix probe name is 211983_x_at. The results will be sorted first on the YPOS column and then on its MATCH column from high to low or descending order. The resulting information will be for the gene actin gamma 1 from the Human Genome.

When a query specifies a specific gene as in the above example, statistical values are calculated with the results and are also reported with a graphic of the spot intensity values in addition to the data returned. The image is dynamically created using the GD package (graphics draw) (www.boutell.com/gd/) available in PHP.



MEAN	STDV	XPOS	YPOS	PROBE	MATCH
85.5	14.7	629	57	211983_x_at	match
66.8	10.3	629	58	211983_x_at	mismatch
78.8	30.6	427	71	211983_x_at	match
113.5	12.7	427	72	211983_x_at	mismatch
64.8	8.6	429	235	211983_x_at	match
483.5	47.2	429	236	211983_x_at	mismatch
68.3	9	471	355	211983_x_at	match
65	9.3	471	356	211983_x_at	mismatch
948.8	104.1	697	359	211983_x_at	match
106	18.3	697	360	211983_x_at	mismatch
134.3	21.6	518	365	211983_x_at	match
69	13	518	366	211983_x_at	mismatch
112.3	18.4	663	477	211983_x_at	match
73.8	14.5	663	478	211983_x_at	mismatch
12817	1327.9	607	579	211983_x_at	match
2664	336.3	607	580	211983_x_at	mismatch
125	19.9	590	619	211983_x_at	match
157	27.1	590	620	211983_x_at	mismatch
57	8.8	699	683	211983_x_at	match
51	8.6	699	684	211983_x_at	mismatch
2961	846.1	654	705	211983_x_at	match
752.8	213.3	654	706	211983_x_at	mismatch

```
22 Records Selected
```

```
Mean of perfect match = 1586.61818182
Standard deviation of perfect match = 443.254866353
Mean of mismatch = 418.4
Standard deviation of mismatch = 108.071291285
Match mean - Mismatch mean = 1168.21818182
Discrimination value = 0.582647176176
```

FIGURE 4.3 – RESULTS SCREEN

Researchers can quickly tell if a gene has a high or low expression level based on the graphic representation. Several examples below (figure 4.4) illustrate the differences between high and low expression levels for different genes found on the Affymetrix chip HG-U133A-2-121502.

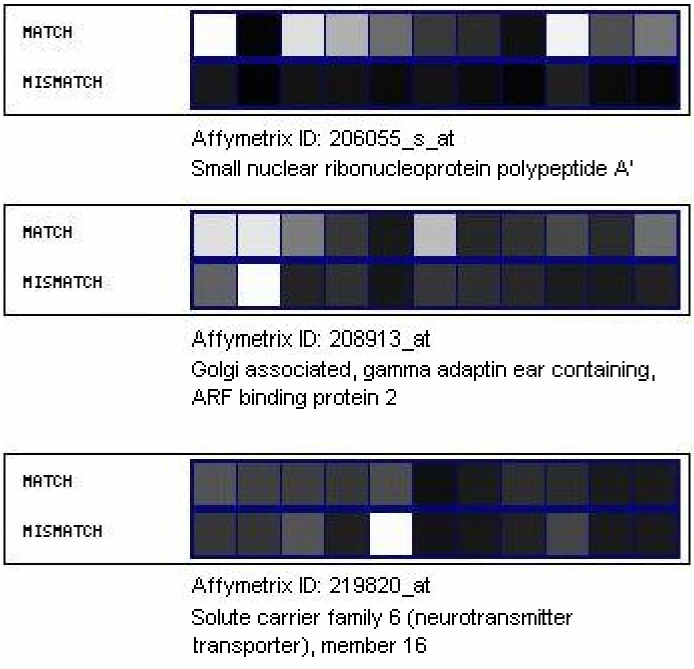


FIGURE 4.4 – EXPRESSION LEVEL IMAGES

V. RESULTS AND CONCLUSIONS

5.1 Project Results

This project has shown that data from Affymetrix experiments can successfully be stored in a data management system, and provide customizable access to researchers. The MiDaR system also meets the requirements that were originally set forth for a product of this nature. Affymetrix data is fully supported and the interface is user-friendly. Access to the system is available to multiple users simultaneously and is contingent upon user authentication. A web front end handles all user interaction, which is accessible from any web browser. The system is also expandable for further development.

Functionality currently being supported allows researchers to begin using the system while future development continues. Affymetrix experiment results can be uploaded and entered into the database. Common queries can be performed on all experiments that exist in the system. Experiments can also be extracted from the system as CEL files. This enables researchers to use other tools available to analyze data such as MAS 5.0, Bioconductor or other statistical packages. Additional features can be added, or support for different microarray platforms such as Agilent, Codelink or custom gene chip formats. Additional functionality will be realized with the continuing development and additions be done by other members of the Bioinformatics Research Group (BRG).

Some difficulties arose during the design and implementation of this project. Due to the relative newness of the object-relational approach versus the relational approach in database design, instruction and documentation were sparse in some areas. SQL insert and select commands become more complex with the use of objects. The use of database tools such as SQL*LDR also are much more difficult to implement. As more and more projects adopt the object-relational approach, due to its benefits, the knowledge base will increase, effectively eliminating this drawback.

5.2 Recommendations

With a solid foundation set, work can begin on more advanced features and improvements. Some additions and improvements that should be done include the capability to store additional microarray platforms, use of encryption for more security, implement the open source MAS 5.0 statistical package, and use SQL*LDR. Additional platforms can be added to the Affymetrix system that is currently in place, with relative ease since the framework has already been implemented. Security can be improved through the use of encryption. Available security measures include Transport Layer Security (TLS)[11], Message Digest Algorithm (MD5)[12], and Secure Hash Algorithm (SHA-1)[13]. The use of any of these or equivalent algorithms would be an improvement in security. The addition of groups and virtual meeting rooms can also be added. Group leaders can handle group level permissions and administration. Incorporating the MAS 5.0 statistical package into the system will provide researchers more analysis tools, and will ensure that all the tools that are needed can be found in one user-friendly system. The use of SQL*LDR to load the probe_position table is significantly faster than using the normal SQL command inserts. Although the structure for the cel_info table is more complicated and include the use of nested tables, the use of SQL*LDR may provide a similar performance gain if implemented.

BIBLIOGRAPHY

[1] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler, "GenBank," *Nucleic Acids Res.*, vol. 33 Database Issue, p. D34-D38, Jan.2005.

[2] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray," *Science*, vol. 270, no. 5235, pp. 467-470, Oct.1995.

[3] A. Brazma, P. Hingamp, J. Quackenbush, G. Sherlock, P. Spellman, C. Stoeckert, J. Aach, W. Ansorge, C. A. Ball, H. C. Causton, T. Gaasterland, P. Glenisson, F. C. Holstege, I. F. Kim, V. Markowitz, J. C. Matese, H. Parkinson, A. Robinson, U. Sarkans, S. Schulze-Kremer, J. Stewart, R. Taylor, J. Vilo, and M. Vingron, "Minimum information about a microarray experiment (MIAME)-toward standards for microarray data," *Nat. Genet.*, vol. 29, no. 4, pp. 365-371, Dec.2001.

[4] "INCITS/ISO/IEC 9075-1:2003. Information Technology - Database languages; American National Standard," 2003.

[5] L. H. Saal, C. Troein, J. Vallon-Christersson, S. Gruvberger, A. Borg, and C. Peterson, "BioArray Software Environment (BASE): a platform for comprehensive management and analysis of microarray data," *Genome Biol.*, vol. 3, no. 8, p. SOFTWARE0003, July2002.

[6] S. Dudoit, R. C. Gentleman, and J. Quackenbush, "Open source software for the analysis of microarray data," *Biotechniques*, vol. Suppl, pp. 45-51, Mar.2003.

[7] P. J. Killion, G. Sherlock, and V. R. Iyer, "The Longhorn Array Database (LAD): an open-source, MIAME compliant implementation of the Stanford Microarray Database (SMD)," *BMC. Bioinformatics.*, vol. 4, no. 1, p. 32, Aug.2003.

[8] David Schlosnagle, "Open Source Database Systems - Are They Ready For The Enterprise?" http://www.personal.psu.edu/users/d/c/dcs217/courses/engl202c/Open_Source_Databases.pdf 2003.

[9] S. M. Stephens, J. Y. Chen, M. G. Davidson, S. Thomas, and B. M. Trute, "Oracle Database 10g: a platform for BLAST search and Regular Expression pattern matching in life sciences," *Nucleic Acids Res.*, vol. 33 Database Issue, p. D675-D679, Jan.2005.

[10] 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. Yang, and J. Zhang, "Bioconductor: open software

development for computational biology and bioinformatics," *Genome Biol.*, vol. 5, no. 10, p. R80, 2004.

[11] Dierks T. and C.Allen, "The TLS Protocol Version 1.0," *RFC 2246*, Jan.1999.

[12] Rivest R., "The MD5 Message-Digest Algorithm," *RFC 1321*, Apr.1992.

[13] Eashlake D. and P.Jones, "US Secure Hash Algorithm 1 (SHA 1)," *RFC 3174*, Sept.2001.

VITA

Joshua J. Hornsby

Date of Birth	October 23, 1980
Place of Birth	Fort Sill, Oklahoma, USA
Undergraduate Study	University of Louisville, Louisville, Kentucky B.S. Computer Engineering and Computer Science, May 2004
Graduate Study	University of Louisville, Louisville, Kentucky M.Eng. Computer Engineering Computer Science, May 2005
Experience	**9/2000 5/2005** University of Louisville, Louisville, KY System Administrator.
	1/2001 5/2001 Thomson Multimedia Inc., Indianapolis, IN Software Developer (C++).
	8/2001 12/2001 Thomson Multimedia Inc., Indianapolis, IN Software Developer (C++), Hardware debugger.
	5/2002 9/2002 Thomson Multimedia Inc., Indianapolis, IN Software Developer (C++).
Activities	Association for Computing Machinery (ACM)
	Institute of Electrical and Electronics Engineers, Inc. (IEEE)
	Student Linux Users Group (SLUG)
	Student Macintosh Users Group (SMUG)
	IEEE South Eastern Conference Robotics Competition
Honors	Trustees’ Scholarship
	Speed School Alumni Scholarship
	Fort Knox Officers’ Wives’ Club Scholarship
	Fort Knox N.C.O. Wives’ Club Scholarship
	Jenny Rose Sipes Memorial Scholarship