# THE USE OF MICROARRAY DATA INTEGRATION TO IMPROVE CANCER PROGNOSIS

Zhe Zhang

A dissertation submitted to the faculty of the University of North Carolina at Chapel Hill in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Biomedical Engineering.

Chapel Hill

2006

Approved by

Advisor: David Fenstermacher

Advisor: Henry Hsiao

Reader: Dechang Chen

Reader: Jeffrey MacDonald

Reader: David Threadgill

# ABSTRACT

ZHE ZHANG: The Use of Microarray Data Integration to Improve Cancer Prognosis

(Under the direction of David Fenstermacher)

Microarray is a high-throughput technology used to simultaneously measuring the expression of thousands of genes in each sample. Therefore, it has the potential to benefit the treatment of complicated diseases like cancer. This study made efforts to improve the application of microarray technologies to clinical medicine with two separate, but related phases. The first phase dealt with the generation of clinically valuable expression profiles from microarray data. By re-analyzing several published cancer datasets, we first confirmed that microarray data presented extra information about prognosis of cancer patients beyond currently used indexes such as tumor size. At the same time, it was noticed that those indexes generally confounded the correlation between gene expression and cancer outcome, so the contents of expression profiles were highly dependent on the clinical background of sample patients. Consequently, integrating multiple datasets was revealed by this study to obtain more general and reproducible cancer expression profiles. A novel data analysis procedure incorporating bootstrap re-sampling and training/testing validation was performed to impartially compare strategies of expression profiling. The results illustrated that after two independent datasets were integrated, the resultant expression profiles more correctly differentiated cancer patients in terms of disease outcome.

The second phase of this study was to develop MAMA (Meta-Analysis of MicroArray), a data mining platform for conveniently collecting, managing, and analyzing multiple microarray datasets altogether. The complete MAMA system included three components: a relational database storing microarray cancer datasets; a web server providing the access to the database; and a client-side application implementing data manipulation and analysis methods. MAMA had an open-source framework allowing other developers to plug in their own data analysis methods. Moreover, it made cross-dataset analysis possible by standardizing annotation of samples and sequences in microarray datasets.

# ACKNOWLEDGEMENTS

To my family (father Guozhong Zhang, mother Ming Zhu, sister Wei Zhang, and wife Yanye Maggie Li in particular), without who I could not finish this work.

My special gratitude goes to Dr. David Fenstermacher, who has always been there to advice and inspire me all these years. I also thank Dr. Lauren Gollahon for giving precious supporting during my first two years in US.

# TABLE OF CONTENTS

Page

APPENDICES

# LIST OF TABLES

Page

Table

# LIST OF FIGURES

Page

Figure

# CHAPTER I
# INTRODUCTION

While many biomedical researchers agree that we are now at the dawn of a ‘Genomic Age’, there are still short of clear visions about how to fit high-throughput technologies, such as microarray, into the diagnosis, prognosis and treatment of diseases. A large number of experiments using these technologies have been carried out to generate information-rich datasets. Analysis of these datasets, however, barely brought us results applicable in medical practice. A quick response to this problem is probably that the technologies are still immature. However, before the technologies become ideal, researchers can still work on fundamental questions such as:

- Does high-throughput data really include helpful information for clinical decisions? If so, how valuable are they and what are their advantages and limitations?
- Have we dug out as much as possible meaningful information from experimental data having been generated? If we have not, what data analysis techniques can we use to fully take advantage of these data?
- What kind of evidences will we need to justify that information presented by high-throughput technologies is clinically informative, and what data analysis methods and protocols should be applied to collect these evidences?

- How much will these technologies change our current way of evaluating and treating disease?

The current study gave its answer to above questions by investigating the relationship between microarray data and cancer. As a complex disease, cancer is an ideal subject of genomic research because its initiation, progression, and metastasis have been related to a series of genomic disorders. A large number of studies have been carried out in recent years to find out gene expression patterns in tumors or tumor subtypes. This study applied its own strategies to re-analyze published microarray datasets about cancer with the purposes different from those of the original studies. The process of identifying the expression patterns of samples is known as ‘gene expression profiling’. Identified patterns can be used to classify tumor tissues. For example, expression patterns obtained in this study were used to distinguish breast cancer patients having good or poor prognosis. A strategy of gene expression profiling can be evaluated by the quality of expression patterns identified. An expression pattern of good quality should classify samples precisely and consistently. The theme of this study is how to achieve more accurate, reproducible and efficient expression profiling of tumor tissues. Two independent but related phases were carried out, both taking about the same efforts to accomplish.

In the first phase, published cancer microarray datasets were analyzed to verify two hypotheses:

1. Microarray data present extra clinical information that is not available via currently used methods.
2. It is feasible to perform gene expression profiling across multiple datasets to increase the overall sample size and quality of acquired profiles.

The first hypothesis was verified by proving that microarray data classified patients better than currently used clinical indexes. The confirmation of this hypothesis was the basis of all following steps, because if it is wrong, applying microarray technology to medical practice is not necessary. Although all similar studies should assume the correctness of this hypothesis too, few have verified it in their reports.

It was observed in early stage of this study that the contents of gene expression patterns were highly dependent on the clinical scenario of sample patients. The expression patterns of the same features, such as the recurrence outcome of cancer, varied substantially when they were obtained from different subpopulations of patients. Consequently, the usefulness of these patterns to general population is limited. The major cause of this observation, ironically, was that the size of most microarray datasets was too small to give reproducible results of expression profiling. This suggestion led to the hypothesis about multi-dataset microarray analysis. The confirmation of this hypothesis will allow researchers to comfortably reuse and combine existing datasets, so information unable to be obtained from individual datasets can be discovered. Furthermore, multi-dataset analysis is a potential solution to the issue about medical application of high-throughput technologies since it can provide stronger statistical evidences by covering various patient subpopulations.

The second phase of this study is the development of a software system called MAMA (Meta-Analysis of MicroArray). This project was motivated by the experience of the data analysis phase, which demonstrated that multi-dataset expression profiling was not technically straightforward. Systematic variation caused by experimental protocol to data annotation exists everywhere in independent microarray studies and datasets. Datasets collected from individual studies can be integrated together only after they have been

consistently annotated, processed and formatted. For biologists and statisticians whose primary interest is high-level data analysis, dealing with these issues is distracting and time-consuming. The MAMA system was developed to provide users a software environment within which they could simultaneously and conveniently investigate multiple microarray datasets. Collected cancer microarray datasets were stored in a relational database after they were re-processed and re-annotated according to pre-defined guidelines. This database was made accessible on the web by a server program that can handle concurrent requests of multiple clients. The MAMA client program was a software application for users to manipulate and analyze microarray datasets. It has a graphical user interface. Using this program, users can selectively download data from the server or directly import their own data, and work with these data on their local disk. High flexibility was a priority of developing MAMA. For example, users were provided with programming interfaces to plug in their own data analysis methods. On the other hand, data objects were formally and consistently annotated. Popular sequence databases, such as GenBank and Unigene, were used to annotate nucleotide sequences; and controlled vocabularies, such as MGED ontology and NCI thesaurus, were used to describe biological samples. Implemented data analysis methods were focused on the correlation between genes and features of cancer samples, or other genes. Meta-analysis methods, such as combined tests and measures of effect size, were made available too.

This dissertation is organized into following chapters, which usually illustrate the data analysis and MAMA projects separately:

- Chapter 1. Introduction.
- Chapter 2. Background:

    - high-throughput technologies;
    - literature review about gene expression profiling in cancer research;
    - summary of standards, databases, and software used in microarray research.
- Chapter 3: Methods:
    - description of analyzed datasets;
    - statistical methods and procedures used for data analysis;
    - software and standards used to develop MAMA.
- Chapter 4: Results and Discussions:
    - pilot studies and the implications of their results;
    - verification of hypothesis about clinical value of microarray technology;
    - verification of the advantages of multi-dataset gene expression profiling;
    - a novel strategy of performing and evaluating gene expression profiling;
    - sensitivity vs. specificity of reporter gene selection;
    - vision, design and data model of MAMA system;
    - data analysis functions implemented in MAMA.
- Chapter 5: Conclusion.
- Chapter 6: References.
- Chapter 7: Glossary.
- Chapter 8: Appendices:
    - step-by-step data analysis procedure;
    - reviews of software development tools and standards;
    - review of meta-analysis methods;
    - supplementary results of data analyses;

- design documents of MAMA project, such as use cases and database schema;
- user guides of MAMA, such as software installation and method plug-in;
- source codes and deliverables of MAMA.

# CHAPTER II
# BACKGROUND

## 2.1 Gene Expression Profiling of Cancer Tissues

Cancer threatens human life by destroying normal tissues with uncontrollable proliferation of malignant cells. While the disease is usually curable by surgery as long as the primary tumors are locally restrained, most cancer-related deceases are caused by metastasis: malignant cells escape from the primary tumor, enter lymphatic or vascular circulation, and establish inoperable secondary tumors at distant locations. Clinical decision about cancer treatment is mostly based on pathological observations of tumor status. The most referred to cancer clinical index is TNM classification, which categorizes cancer patients jointly according to tumor size (T), local lymph node metastasis (N), and distant metastasis (M) [1]. Other common indexes, such as histological grade and angioinvasion, have been used as supplements of TNM system [2, 3].

The search for a general cure of cancer has become the biggest dilemma of biomedical research. On one side, the knowledge about the molecular basis of cancer has been substantially improved after decades of enormous research effort. It is already known that cancer is caused by a series of genetic disorders, from point mutations to insertion/deletion to chromosome rearrangement [4-7]. Genetic defects destroy the balance between cell growth

and cell death, and transform normal cells into malignant. Therefore, the investigation of cancer-related genes, such as tumor-suppressor genes and oncogenes, has been the focus of cancer research for more than 10 years [8, 9]. Some of these genes have been used as molecular markers in cancer clinics, such as p53 [10] and HER-2/neu [11]. On the other hand, discoveries in molecular biology have hardly benefited more cancer patients. Pathological indexes still dominate the diagnosis of cancer in clinics. Although the list of cancer-related genes is continuously growing, the addition of most new members was unable to improve the comprehensive image of malignancy transformation, advocating high genetic variability of cancer [12]. Regular laboratory approaches, which laboriously investigate a single gene or several closely linked genes at a time, have been proven ineffective to this complex and genomic-level disease. To improve cancer treatment, cancer patients should be classified based on genomic information produced by more efficient technologies.

Development of high-throughput technologies in recent years is about to bring biomedicine into a genomic age. These technologies have promising potentials in cancer research by allowing investigators examine cancer from a systematic perspective. CGH (Comparative Genomic Hybridization) is able to detect chromosomal gains and losses through the entire genome [13]. Chromosomal DNA obtained from both cancer and normal tissues is labeled with two fluorescent colors, and the difference in signal intensity indicates sequence deletion or amplification. Although array-based CGH has been developed to improve the resolution of this technology [14], CGH is not suitable for detecting mutations of short sequences. Since point mutations are the most common chromosomal alteration, genomic sequencing is a thorough solution for identifying genes mutated in cancer. However, the effort and cost of whole-genome screening are still impractical for most research facilities.

Technologies such as heteroduplex analysis and CSGE (Conformation Sensitive Gel Electrophoresis) have been used as alternatives of sequencing for mutation detection [15-17]. While sequence information is mostly static, expression profiling technologies can describe the dynamic status of cancer cells at gene transcription level. SAGE (Serial Analysis of Gene Expression) and microarray have been used to generate source data of expression profiling. By assuming that 10-12 bp nucleotide sequences will uniquely represent most human transcripts, SAGE concatenates short cDNA fragments into sequencing vectors to constitute a SAGE library [18, 19]. The library will be sequenced to comprehensively quantify gene expression. SAGE constitutes an open system, so it can measure the expression of unknown or rarely-expressed genes in cells under investigation. Microarray is the most applied high-throughput technology in recent years. By fixing probes (synthesized oligonucleotides or sequences extracted from cDNA libraries) of transcripts onto a solid surface as tiny spots, microarray technology is able to simultaneously and efficiently measure the expression of thousands of genes [20-22]. Microarray is not as sensitive as SAGE because background noise on arrays makes the measurement of low abundance mRNA inaccurate, but it is less laborious and expensive. Individual microarray studies are usually able to measure expression of genes in 20 to 200 samples. Therefore, in cancer research, microarray datasets are often used for classifying samples into categories such as cancer subtypes [23-26]. The protein-level genomic technology is proteomics, which relies on high-throughput platforms, such as array slides or 2-D electrophoresis gels, to simultaneously analyze a large number of proteins [27-29]. Proteomics has been used to classify biological samples as microarray, and to discover, identify, and quantify cancer biomarkers [30, 31]. Since proteins are the

functional units in cells, proteomics could be considered as the bridge connecting genomics and clinical medicine.

As a relatively mature high-throughput technology, microarray has been used to produce gene expression datasets about a variety of types of cancer. A common application of microarray datasets is to identify gene expression patterns corresponding to the sample features under investigation, such as the recurrence of cancer. The process of identifying gene expression patterns is often referred to as ‘gene expression profiling’, which is characterized by the major trends in biomedical research: large datasets, computer-assisted data analysis, and integration of scientific domains including biomedicine, statistics, and information sciences. Because of its clinical prospects, expression profiling of cancer tissues has been extensively explored [32-35].

Statistical analysis plays a critical role in gene expression profiling. Clustering and reporter gene selection are two extensively applied techniques [36-38]. Clustering is relatively more visual and straightforward, while reporter selection is usually more complex and has more variations. Methods having been used to reporter selection include simple hypothesis testing methods, such as Student or permutation t test [39, 40], ANOVA [41], and advanced models, such as Bayesian Networks [42] and Principal Component Analysis [43]. Selection of expression profiling methods should be determined according to the purposes of research and the characteristics of dataset. Complicated methods are not necessarily better than simple ones, and powerful methods usually make stricter assumptions on data than methods that are less powerful but more robust.

Analysis of microarray data also presents a challenge to traditional statistics because the number of variables (genes) in a microarray dataset is usually much larger than sample size.

Repeating a hypothesis test on a large number of variables will falsely reject the null hypothesis for some variables, an issue usually referred to as ‘multiple hypothesis testing’ [44]. For example, when a test is performed on each of 10,000 genes, averagely 100 genes will get a significant p-value less than 0.01 just because of the random distribution of data. Therefore, reporter gene lists derived from microarray datasets often include false positives. Decreasing the sample size or increasing the genes of a dataset will generally augment the rate of false positives in a reporter list. Reducing the size of the list will lift the specificity of reporter selection, but lessen the sensitivity at the same time. The ‘q-value’ index has been suggested to control the false positive rate of reporter selection by adjusting the test p-values by the number of tests performed [45, 46].

The following gives a quick literature review about cancer microarray studies using breast cancer as an example. Breast cancer is aroused by an accumulation of genetic mutations, and its initiation, invasion, progression, and metastasis are related, but distinct diseases. The complexity of breast cancer makes it an ideal subject of genomic-level research [47]. Gene expression profiling about breast cancer has drawn many research interests in recent years and a large number of studies have been reported. The focuses of microarray studies about breast cancer ranged from biological differentiation of different cell lines [48] or normal/tumor tissues [49], to clinical classification of tumors into subtypes using expression profiles [49-52], to discovery of molecular markers and drug targets [53-56]. Since most breast cancer-related deaths are caused by recurrence and/or metastasis of disease, profiling of disease endpoints is clinically more valuable. It is a challenging topic at the same time because of the difficulty of predicting cancer outcome.

Practitioners usually measure and integrate a number of prognostic indexes to predict the outcome of breast cancer. The prediction guides the making of disease treatment decision, which is crucial for the quality of patients' post-diagnosis life. Besides TNM classification, common prognostic indexes of breast cancer include age, ethnicity, grade, vascular invasion, and so on [57]. Moreover, molecular markers, such as HER2/neu and ER (Estrogen Receptor) status, have a growing utilization in the prognosis of breast cancer [57, 58]. For example, an extensive study (n>37,000) concluded that ER-negative patients were 7-times more likely to develop recurrent diseases than ER-positive patients after 5 years or more of adjuvant tamoxifen treatment [59]. However, currently used indexes often misclassify patients in terms of disease outcome because they are unable to describe the integral status of tumors with enough details. Microarray data, on the other hand, has the potential to characterize subtypes of cancer more specifically by making a census about gene expression in tumors.

Microarray studies have reported expression profiles related to outcome of breast cancer. Sørlie *et al* collected 85 tissue samples from breast cancer patients (78 cancers, three benign tumors, and four normal tissues), and categorized them with hierarchical clustering of their microarray data [49]. The categorization was supervised by post-diagnosis survival of patients. All samples were classified into 6 sub-groups and Kaplan-Meier survival curves of these sub-groups were significantly separated ($p<0.01$). Van't Veer *et al* applied different approaches to identify gene expression profiles associated to recurrence of breast cancer, during which the correlation between expression of each gene and 5-year prognosis was evaluated with statistical tests across 78 breast cancer patients [60]. Genes having the most significant correlation were selected as reporters of recurrence outcome. 'Leave-one-out' cross-validation was then applied to optimize the length of reporter gene list. The 70-gene

profile achieved optimal validation results. When the expression profile including these genes was used to classify patients, observed recurrence outcome of 65 out of 78 patients (83%) was correctly matched by the classification. Furthermore, when this profile was applied to a testing group of 19 patients, 17 patients (89%) was correctly classified. Although previous study acquired inspiring results, its weakness should also be noticed. The self-validation result might be an overfitting and the size of the testing group was too small to give enough statistical power. Furthermore, all sample patients were selected from a specific sub-population of breast cancer (lymph node negative, tumor size less than 5 cm, and age under 55 at diagnosis), so the expression profile obtained from these patients might not be generally applicable.

The dependence of gene expression profiles on clinical background of sample patients has been observed in microarray datasets. Gruvberger *et al* noticed that 165 out of 231 top-ranked genes in the study of van ‘t Veer had significant correlation with ER status of patients [61], and suggested that expression profiling should be performed separately on ER-positive and -negative patients. It has been reported that some clinical indexes, such as ER status and tumor size, ubiquitously influence the expression of genes in tumor cells [62, 63]. Therefore, these indexes are confounders of gene expression profiling. Since microarray data are expected to provide extra clinical value beyond currently used indexes, expression profiles will be more valuable if the intervening effect of clinical scenarios is controlled. The suggestion of Gruvberger, though, is not an ultimate solution to this issue because there are other confounders of expression profiling besides ER status and it is not feasible to further split sample patients into subgroups corresponding to all confounders.

Both of the sample size and confounder issues discussed above put doubts on the general usability of expression profiles identified from individual datasets about cancer. Consequently, recent microarray studies have been trying to give a solution by integrating multiple datasets. The advantages of data integration are apparent because it increases the overall sample size of data analysis. Furthermore, since research questions often need to be re-defined for data integration analysis, information not discovered by individual studies can be uncovered from integrated data. One of the most common techniques of data integration is meta-analysis [64, 65]. Meta-analysis is often referred to as 'analysis of analyses'. It takes the results of individual studies as its inputs to carry out secondary analysis, and how those studies got their results is irrelevant to the process of meta-analysis.

Meta-analysis and other data integration strategies have been used in cancer microarray studies [66-69]. Ghosh *et al* investigated the consistence of four prostate cancer microarray datasets using a meta-analysis process [68]. They concluded that the profiles derived from these independent datasets shared significant similarity and proposed candidate gene pathways with the results. Microarray datasets of various cancer types were meta-analyzed by Rhodes *et al* [69]. They applied a comparative meta-profiling method to 40 published datasets with an overall sample size greater than 3,700. Gene expression profiles identified from these data were mapped to several characteristics of cancer tissues. For example, comparison of gene expression in cancer and normal tissues across 21 independent datasets and 12 cancer types recognized 67 genes that had significantly high possibility to be selected as reporters from individual datasets. The expression profile including these genes was proposed as a general gene expression signature of neoplastic transformation. Altogether,

previous results shed light on the possibility to combine multiple microarray datasets for more precise profiling.

Rhodes' study implied that samples of disparate subpopulations might still share considerable commonness in terms of expression profiling results, which advocated the feasibility of integrated profiling of cancer outcomes. If a gene is highly correlated to a disease outcome in one patient subpopulation but not the others, its role in the integrated data will fade out as disparate patient cohorts are involved. On the other hand, genes whose correlation to an outcome is independent of clinical background of samples will have a bigger chance to be identified in integration analysis. Nevertheless, for multi-dataset expression profiling, it should be assumed that a common expression profile corresponding to the investigated output variable does exist regardless of the experimental design of individual studies and the substantial systematic variations between microarray datasets. Therefore, it is often necessary to re-define the output variables to make them proper for integration analysis. As a result, sample patients of the original studies may need to be filtered and re-categorized accordingly.

Categorizing cancer patients into prognosis groups is not as straightforward as it looks. In ideal situation, sample patients will be followed up until their disease endpoint is reached, so they can be unquestionably categorized. In the case of breast cancer, patients are considered as being cured if they keep recurrence-free long enough (usually 15 to 20 years) until their hazard to the disease is not greater than that of the general population. In clinics, however, outcome of cancer patients is often censored by short follow-up or incomplete medical record. Survival analysis is the most common method dealing with censored data [70]. It builds a Kaplan-Meier curve with follow-up data of each patient group and calculates test statistics

about the separation of curves. Survival analysis is statistically powerful since it takes all available follow-up information into account. However, its assumption about the constant effect of predictive index on output variable is not the fact in the case of cancer outcomes because survival and recurrence rates of cancer patients usually change from time to time. Alternatively, classifying patients into pre-defined prognosis groups would simplify the subsequent statistical analyses. The clinical convention is to categorize patients according to their 5-year follow-up. Patients who survive and keep disease-free for at least five years will be thought as having good prognosis and who die or recur within five years after diagnosis will be thought as having poor prognosis. Such a classification will exclude some sample patients from the following data analyses if they have not been followed up long enough to be categorized. Meanwhile, it will make the generating and validating of expression profiles more convenient, and if the cutoff value for classification is properly chosen, the corresponding expression profiling could also be powerful. Nevertheless, it should be pointed out that the 5-year convention was established by usage rather than by any biological basis. Classifying sample patients arbitrarily regardless of their intrinsic difference in genetic background will considerably reduce the statistical power of gene expression profiling.

Retsky *et al.* investigated the follow-up data of 1,173 breast cancer patients and discovered that their recurrence rate had two peaks [71, 72]. The summits of these peaks were located at about 18 and 60 months after mastectomy, separated by a nadir around 50 months. It was also concluded that the appearance of this double-peaked distribution was independent of tumor size, number of positive nodes, and menopause status of patients. Computer simulation of tumor progression implied that this distribution might be caused by disparate dynamics of secondary tumor growth. It was proposed that early recurrence was

caused by mastectomy-initiated accelerated growth of secondary tumor, while the second peak was the result of steady stochastic transitions of tumor progression phases. It was further suggested that different treatment strategies should be applied to patients located in different peaks.

Since Retsky's recurrence model is independent of clinical indexes, it could be considered as applicable to general population of breast cancer. If the computer simulation model about growth of secondary tumors is true, patients respectively recurred during those two peaks have a good chance to be distinguished by their gene expression profiles. Therefore, as an output variable for expression profiling, 3-year prognosis of breast cancer has better grounds than 5-year prognosis.

The medical application of microarray technology is at its very early stage. Researchers are actively bringing up new topics and methods in this field. The current study tried to verify that integration of microarray data from independent sources will improve expression profiling of cancer prognosis. A novel strategy of generating and validating gene expression profiles was developed and applied to two published microarray datasets about breast cancer. The two-peak recurrence model discovered by Retsky was adopted to make results more biologically meaningful and clinically beneficial. It was revealed after two datasets were integrated, not only selection of reporter genes had higher specificity, but also the expression profiles acquired were more predictive. Furthermore, when the same strategy was applied to four public microarray datasets about lung cancer, similar results were observed. It was also demonstrated that microarray data provide extra prognostic value besides commonly used indexes.

## 2.2 Microarray Standards, Databases and Software

Handling of microarray data is challenging for most biomedical researchers since it involves many aspects of information technologies. First, the structure of microarray datasets is complicated and has many variations. To describe a reproducible microarray dataset, metadata from experiment design to data processing strategy should be completely and unambiguously given. Consequently, standards of formally describing microarray datasets are necessary. Secondly, microarray datasets include slide images, raw measurements, processed data and other related data types, which make their structure complicate and their size big. Storage of microarray datasets in centralized repositories, such as databases, will give researchers quick and convenient data access. Finally, methods for presenting, processing and analyzing microarray data are various and often sophisticated. Packing these methods into computer software will save researchers from the trouble of implementing these methods by themselves.

MGED (Microarray Gene Expression Data) Society is the most active organization that creates microarray standards [73]. A series of standards established by MGED for annotating and exchanging microarray data are being widely used in microarray community. The following is a brief introduction to these standards.

MIAME (Minimum Information About a Microarray Experiment) is the first standard proposed by MGED [74]. It is recommended to authors and editors of microarray publications as a set of information necessary to reproduce microarray datasets. Particularly, MIAME requires for description of array and experiment design, samples, experiment protocols, and measured data. These requirements are summarized in MIAME checklist. MIAME is mostly conceptual. It neither structuralizes the contents of microarray datasets nor

specifies the standard vocabularies to describe them, which are respectively the goals of MGED MAGE (MicroArray and Gene Expression) and Ontology working groups.

MAGE has established two standards for structure of microarray data: MAGE-OM (Object Model) and MAGE-ML (Markup Language) [75]. MAGE-OM is a complex data model that defines over than 150 data types related to microarray experiments and the inter-relationship of these types. Its development follows the principles of UML (Unified Modeling Language, Object Management Group, Inc.). MAGE-ML defines a set of XML (eXtensible Markup Language, World Wide Web Consortium) elements, which are automatically derived from MAGE-OM. Hence, XML documents tagged with these elements can be exchanged between MAGE-compliant data systems. Although MAGE standards have covered most aspects about microarray datasets, they are mostly focused on the generation of microarray data and do not strongly support high-level data analysis. Furthermore, the complexity of these standards makes them difficult to be implemented and may cause low performance of data processing programs.

The aim of MGED Ontology is to provide a set of defined and tree-structured terms for description of microarray-related concepts [76]. It has two major branches: core and extended. The core ontology is limited to the description of data objects covered in MAGE-OM while the extended one has a wider scope. Although a major part of MGEG Ontology has been put on the features of biological samples used in microarray experiments, the supplement of other controlled vocabularies is necessary to make unambiguous description of samples because the high diversity of biological entities. For example, NCBI (National Center for Biotechnology Information) Taxonomy database provides the official names of species and their categorization [77], and NCI (National Cancer Institute) Metathesaurus is a

resource summarizing cancer-related vocabularies. The probes or nucleotide sequences included in array designs also need to be systematically annotated. Major sequence databases, such as GenBank [78] and Ensemble [79], have been used to annotate sequences in microarray datasets. However, these databases often store multiple sequence records belonging to the same genes and assign them different identifiers. Redundant appearance of a gene in an expression profile will reduce its sensitivity and artificially increase the weight of the gene, so it is better practice to condense data of redundant genes together for high-level analysis. Consequently, many researchers prefer annotating nucleotide sequences with systems developed for naming genes or gene products. NCBI Unigene database, which clusters GenBank sequences into a non-redundant set of genes, is such a system having been commonly used [80, 81]. Since individual studies use a variety of sequence annotation systems, mapping annotations between systems has became an ordinary data processing step in microarray studies.

Databases play a crucial role in the storage, distribution, and standardization of microarray datasets. Public microarray databases are usually accessible via the web, so they have web-based interface for users to query for sequences, samples, experiments, and other data types. Some microarray databases also provide tools for data processing and analysis. Microarray data models such as MAGE-OM and RAD [82] have been taken as basis of database schemas.

SMD (Stanford Microarray Database) is the first major microarray database whose source codes were released [83-85]. Although it was initially developed to serve human and yeast researches at Stanford University, its application was extended to a much larger scope

and has been adopted by other institutes. SMD was not MAGE-compliant originally, but its later version supported the exchange of data formatted with MAGE-ML.

As more and more microarray studies are reported and their source data are available, the demand for centralized repository of published microarray datasets is increasing in research community. Two major bioinformatics organizations, NCBI and EBI (European Bioinformatics Institute), discretely provided their solution to this requirement. GEO (the Gene Expression Omnibus) is a database established by NCBI [86]. It currently stores about half a billion gene expression measurements generated by microarray or SAGE. The microarray database established by EBI is called ArrayExpress [87, 88]. The schema of ArrayExpress is consistent with MAGE-OM and other MGED standards including MIAME and MGED Ontology are also adopted by this database for data submission and description.

Data-mining functions are provided by some microarray databases as their supplement. For example, ONCOMINE is a system combining a microarray database with a data-mining platform for discovering gene expression patterns in cancer [89]. Establishment of this system dramatically accelerated subsequent data analyses. Studies based upon ONCOMINE database have been carried out successfully and reported [69, 90-92].

Computer software is a requisite part of microarray studies. To support their products, microarray hardware vendors such as Affymetrix, Inc. usually provide software tools for upstream handling of experimental results, such as image acquisition and in-chip normalization [93-95]. On the other hand, a large number of computer programs are available for high-level microarray analysis. These programs have been developed as simple desktop tools to powerful enterprise systems. Popular statistical programming language including R

(www.r-project.org), MatLab (MathWorks, Inc.), and SAS (SAS Institute, Inc.) are also extensively applied in microarray researches.

GeneSpring GX (Aligent technologies, Inc.) is one of the most popular commercial microarray software. Like most of other business products, it has an attractive user interface and is relatively user-friendly. Besides rendering data with graphics, it implements various types of statistical methods, such as ANOVA and clustering, for identifying reporter genes or expression patterns. Furthermore, it provides programming interface that allows users to incorporate third party applications for data visualization or analysis. Although GeneSpring GX is gaining its popularity, sophisticated users may still feel that its functionality cannot fulfill their demand because of the rapid updating of data analysis techniques, which is probably the reason why most commercial microarray software did not succeed.

Allowing users to modify or extend the source codes to meet their special requirements, open-source software has become a major driving force of microarray research [96-100]. BioConductor is an R-based open source project for the analysis of microarray and other genomic data [100-103]. Its newest version includes over 100 software packages implementing annotation, documentation, statistical analysis, and many other functions about genomic data. It should be pointed out that the quality of open source software varies. Unsophisticated programs may mistakenly implement statistical methods or inaccurately interpret results. Therefore, open source programs should only be recommended to experienced users.

In spite of various options of microarray software, researchers often find it necessary to write programs by themselves, especially when they are developing new data analysis

methods or procedures. For example, the current study coded most programs used in its data analysis phase.

The MAMA (Meta-Analysis of MicroArray) system presented in this dissertation is an open-source platform supporting data-mining in cancer microarray datasets. It has three major components: a relational database, a server program, and a data analysis package. The MAMA database provides a centralized storage of microarray datasets about cancer and the server program made this database web-accessible. Unlike ONCOMINE, the focus of the MAMA project is its data analysis package that could be run as a desktop application. This application supported basic operations about microarray dataset such as data import/export and re-processing. Furthermore, it provided an open-source framework to satisfy diverse user requirements on its functionality. Similar to GeneSpring GX, MAMA allows users to plug in their own methods by implementing specified programming interface. A highlight of MAMA project was the availability of meta-analysis functions, which was realized by using MGED Ontology and other controlled vocabularies to describe data from independent sources. The MAMA system was expected to provide microarray researchers an easy-to-use and extensible data mining platform with functions not fulfilled by other microarray software.

# CHAPTER III
# METHODS

## 3.1 Data Analysis

### 3.1.1 Datasets and Data Pre-processing

This study analyzed published microarray datasets, including two from breast cancer patients [49, 60] and four from lung cancer patients [104-107]. All datasets provided clinical data about patients, such as disease follow-up and tumor size, in addition to microarray data. Cancer patients in each dataset were re-sampled and classified into two prognostic groups. Breast cancer patients who developed secondary tumors within three years after mastectomy were put into the poor prognosis group while patients who were followed up for at least three years and had no observed recurrence were classified as having good prognosis. Patients inappropriate to either group were excluded from this study. (See Appendix A.1 for demo of patient classification.) In the case of lung cancer, patients were classified according to their two-year survival outcome, and only adenocarcinoma patients were selected. Disease outcome of patients in all datasets was denoted as a dichotomous variable for all statistical analyses (0: good prognosis and 1: poor prognosis). The resulting sample sizes of all breast and lung datasets are summarized separately in Table 1A and 1B. Before analyzing these datasets, several data pre-processing steps were carried out. Sequences of all datasets were mapped to Unigene clusters and expression levels of redundant entries were averaged to

generate a set of genes without redundancy. In the case of breast cancer, there were 5,569 non-redundant Unigene clusters presented in both datasets. (See Appendix A.2 for demo of mapping sequences to Unigene.) Expression measurements having low quality and sequences unable to be mapped to Unigene were filtered out of the datasets. Ratio expression data of cDNA datasets were $\log_{10}$-transformed. Furthermore, expression measurements in each dataset were normalized for each patient and then for each gene, making the median expression level of each patient or gene equal to 0.0 and the standard deviation equal to 1.0. (See Appendix A.3 for demo of expression data pre-processing.)

**Table 1:**
**Microarray Datasets Used in This Study**

**Table 1A: Two Breast Cancer Datasets**

Poor prognosis: patients recurred within three years after diagnosis; good prognosis: patients had no observed recurrence and were followed up for at least three years. Patients who were in the original datasets, but could not be categorized into either group were removed from this study.

| Dataset [reference] / Platform | Poor Prognosis | Good Prognosis | Total |
|---|---|---|---|
| **Rosetta Breast [60] / cDNA** | 31 | 51 | 82 |
| **Stanford Breast [49] / cDNA** | 37 | 25 | 62 |
| **Total** | 68 | 76 | 144 |

**Table 1B: Four Lung Cancer (Adenocarcinoma) Datasets**

Poor prognosis: patients died within two years after diagnosis; good prognosis: patients survived after at least two years of follow-up. Only adenocarcinoma patients were selected. Patients who were in the original datasets, but could not be categorized into either group were removed from this study.

| Dataset [reference] / Platform | Poor Prognosis | Good Prognosis | Total |
|---|---|---|---|
| **Harvard Lung [106] / Oligo** | 30 | 33 | 63 |
| **Michigan Lung [105] / Oligo** | 17 | 43 | 60 |
| **Stanford Lung [104] / cDNA** | 10 | 9 | 19 |
| **Ontario Lung [103] / cDNA** | 3 | 8 | 11 |
| **Total** | 60 | 93 | 153 |

### 3.1.2 SEP: Score for Expression Profile

A designed variable, Score for Expression Profile or SEP, was defined as a qualifier of gene expression profiles. Given a profile with N reporter genes, the SEP of each patient was calculated as:

$$\textbf{(1).} \quad \mathbf{SEP = \Sigma^{N} [w_i * (X_i - E_i)]}$$

In Formula (1), $w_i$ was the weight of *i*th gene in the profile. The sign of $w_i$ corresponded to a positive or negative correlation between *i*th gene and the output variable under investigation while the magnitude of $w_i$ indicated its relative importance in a profile. $X_i$ was the expression measurement of *i*th gene in the patient while $E_i$ was its expected expression level. Since the output variable was dichotomous, $E_i$ was the expression level that had equal probability to be found in either sample group and could be denoted as $E(X_i \mid p_+ = p_- = 0.5)$. The $E_i$ of each gene was empirically estimated from the training data. By using Formula (1), the difference between $X_i$ and $E_i$ of each gene in the profile was weighted and then be linearly summarized to obtain a SEP score. According to this process of calculating SEP, patients with poor prognosis were expected to have lower SEP scores than patients with good prognosis in general. SEP demonstrated its advantage in the current study as a numeric variable appropriate for common quantitative methods, such as chi-square test and 2-group comparison. Consequently, it was treated as a potential prognostic index, representing the information provided by microarray data. (See Appendix A.8 for calculation of SEP score.)

### 3.1.3 Correlation Analysis

Genes whose expression is highly correlated to an investigated sample feature can be considered as the reporters of that feature. Identifying reporter genes from a genome involves

procedure called ‘feature selection’, during which one or more statistical tests are applied to evaluate gene-feature correlation. Among all the methods used to evaluate correlation between two variables, Pearson’s correlation is most common and straightforward [70]. It has the best performance when data follows linear distribution. Pearson’s correlation reports an r statistic as its result. After this r statistic is further transformed into a normally distributed t statistic, a p-value corresponding to a hypothesis testing about correlation can be acquired. In this study, reporters of an output variable were defined as those genes having the most significant p-values. (See Appendix A.5 for more details about correlation analysis.)

### 3.1.4 Partial Correlation Analysis

Partial correlation is a statistical technique used to control the effect of confounders out of the correlation between two variables [108]. In the case of cancer microarray analysis, it was observed that the correlation between gene expression and disease outcome varied with some clinical indexes (see Fig. 2), such as Estrogen Receptor (ER) status of breast tumors. This confounding effect causes the dependence of cancer expression profiles on the clinical scenarios of sample patients, and then reduces the reproducibility of profiles. A partial correlation analysis was proposed by this study, during which the gene-outcome correlation is re-evaluated after a confounding variable is controlled. The first step of this analysis is to transform each gene expression measurement to a residual using:

$$\textbf{(2).} \quad X_{residue} = X - E\,(X \mid \text{controlled variable})$$

In Formula (2), X was the original expression measurement of a gene and E was the expected X given a known value of the controlled variable, such as positive or negative ER status. Patients were classified into groups according to the controlled variable and E values

of each gene were estimated by averaging X of all patients in each group. Subsequently, the partial correlation coefficient (r') of each gene to the output variable was calculated using the residuals. Once each measurement in the original data matrix was transformed to a residual, the gene-outcome correlation will be re-calculated, with X replaced by $X_{residual}$, to get a partial correlation coefficient (r'). The r' statistic was considered the same as r through the subsequent analyses. Theoretically, Formula (2) could be reiterated until all the confounders were controlled. (See Appendix A.6 for more details about partial correlation analysis.)

### 3.1.5 Rank Sum Test

Since samples had only two outcome categories in this study, two-group comparison methods were potentially appropriate for calculating gene-output correlation. Wilcoxon Rank Sum Test (RST) [70] was used as the main method to evaluate differential expression of genes between opposite patient groups. As a non-parametric method, RST does not assume the normality of data as parametric methods, such as commonly used Student's t test. Large portions of the genes in analyzed microarray datasets do not satisfy this assumption of normality. Although non-parametric methods have less statistical power, such a disadvantage is insignificant if reporter genes are selected based on relative ranks of genes instead of their p-values. RST reports a Z statistic as its result. It first transforms all expression measurements of a gene into ranks, and then calculates the Z statistic with the ranks assigned to the compared groups. When both groups have no less than eight observations, Z statistic follows standard normal distribution, so a corresponding p-value can be obtained. With the procedure used in this study, the Z statistic of a gene would be positive if it is generally over-expressed in good prognosis patients; otherwise, it would be negative. No matter which

statistical test was used, the resultant test statistics were used to rank genes. In this study, genes having the highest magnitude of Z statistics or the smallest p-values were selected as reporter genes. (See Appendix A.7 for guide of calculating RST Z statistic.)

### 3.1.6 Logistic Regression Model

Logistic regression is a statistical technique used to evaluate the predictive ability of independent variable(s) on a dependent variable having dichotomous outputs [108]. Building logistic regression models is process during which the best estimation of the parameters of a regression formula is achieved based on input data. The resultant model has a statistic called -2 Log Likelihood (-2LL), which is used to compare fitness of models to actual observation of an output variable. For a fixed sample size, a smaller -2LL represents better model fitness. A model is uni-variate if it has only one independent variable and is multi-variate if it includes more than one independent variable. Models generated by this study utilized available prognostic indexes including SEP as independent variables and disease outcome as the dependent variable. Multi-variate models were built using a forward stepwise procedure, during which independent variables were added into a model one by one in the sequence of their significance. The resultant –2LL of each step was recorded to trace the changing of model fitness. All models were generated using SAS System for Windows, Release 8.02 (SAS Institute, Inc.).

### 3.1.7 ROC Curve

ROC (Receptor Operating Characteristic) curve is a type of plot used to evaluate the accuracy of a clinical test [70]. It shows the tradeoff of sensitivity (true positive rate) and

specificity (true negative rate) when the test result is at each of its cutoff points. From a ROC curve, one can determine the false positive rate that needs to be tolerated to guarantee a certain sensitivity of a test. The curve is usually drawn from the lower left corner to the upper right corner in a 1.0X1.0 scale, so its AUC (Area Under the Curve) ranges from 0.0 to 1.0; the larger the AUC, the higher accuracy of a corresponding clinical test. A test will be ideal if its ROC curve has AUC equal to 1.0. In this study, the SEP scores of patients were considered as the results of a clinical test based on microarray experiment, and ROC curves built with these scores were used to evaluate the clinical value of expression profiles.

### 3.1.8 Bootstrap Re-sampling Strategy

Some reporter genes selected into expression profiles could be false positives because of the issue of multiple hypothesis testing. Consequently, validating a profile with the same data used to generate it will cause overfitting in results. To avoid self-adaptive overfitting, patients of each dataset were randomly re-sampled into training and testing subgroups. Thereafter, the expression profiles were generated from training data and validated with testing data. Although this strategy eliminated overfitting, it still had a major drawback. The random re-sampling process introduced bias into the profiling and validating results, especially when the sample size of a dataset was small. A bootstrap strategy was applied to remove sampling bias by repeating the sampling-profiling-validating process a large number of times. Each bootstrapping repeat created an expression profile from the training data, which was used to calculate SEP scores of testing patients. SEP scores were used to classify patients and build an ROC curve, insulting in classification accuracy and AUC as test statistics to indicate the quality of the expression profile. These statistics obtained from all

bootstrapping repeats were summarized to get their median and 90% Confidence Interval (CI) values. Hence, this bootstrap re-sampling strategy allowed the objective and unbiased comparison of gene expression profiles and the approaches used to generate them. Every bootstrapping repeat also assigned a Z statistic and a rank to each gene. These results were summarized to make a final ranking of all genes for entire dataset. Genes consistently getting significant Z values or top-ranked were selected as reporters. (See Appendix A.4 for demo of patient re-sampling.)

### 3.1.9 Gene Categorization According to Gene Ontology

Gene Ontology (GO) is an infrastructure of controlled vocabularies supporting unambiguous description of genes and their products [109]. All vocabularies of GO are organized as a tree-like structure with three roots: ‘Biological Processes’, ‘Cellular Components’, and ‘Molecular Functions’. GO allows researchers consistently and conveniently query for attributes of a given gene, genes of a specific category, and even the associations between genes. In this study, reporter genes were categorized into the ‘Biological Processes’ domain of GO, which includes sub-categories such as Cell Cycle and Signal Transduction. The route of mapping Unigene clusters into GO categories was: Unigene ID $\rightarrow$ Entrez Gene Symbol [110] $\rightarrow$ International Protein Index [111] $\rightarrow$ GO ID.

## 3.2 MAMA Project

### 3.2.1 Developmental Stages

The development of the MAMA (Meta-Analysis of MicroArray) project followed the common criteria of software engineering. It started with vision and requirement analyses, followed by use case analysis, system architecture design and data modeling. The database of MAMA was designed and implemented before developing a software application of data analysis. The sequential developmental stages of this application were software architecture design, package and class design, coding and testing. Other efforts involved in this project included loading data into database, User Interface (UI) design, and documentation.

### 3.2.2 Data Models

MAMA project used two data models to describe microarray data objects and their relationship. The data model adopted by the MAMA database is MAGE-OM (MicroArray Gene Expression – Object Model) [75]. MAGE-OM is a complex data model developed by MGED (Microarray Gene Expression Data) Society to facilitate the sharing of microarray data. It defines the concepts about most aspects of microarray-based experiments and their associations. (See Appendix B for more details about MAGE-OM classes.) Although the MAMA database is fully MAGE-compliant, only a minor portion of its tables have data loaded into them because the current project only dealt with the high-level analysis aspect of microarray data. Despite of the complexity of MAGE-OM, it is focused on the description of static data, but not the data analysis procedures. Therefore, the data analysis application needs its own data model. Since this application was coded with Java (J2SE, v1.4.2 Sun Microsystems Inc.), an object-oriented programming language, its data model has an object-

oriented tree structure. Specifically, its root class is called ‘Workspace’, within which various data manipulation and analysis operations can be performed by end users. Each Workspace includes various types of data objects, such as ‘Query’, ‘Experiment’, and ‘Analysis’. Each of these objects has its own contents and associations to other types of data. For example, a ‘Query’ object has attributes including its identifier, subtype, created date, and selection limits, and it can be related to a ‘Query Result’ object.

### 3.2.3 Relational Database

The database schema of MAMA included the schema of ArrayExpress [87, 88], a MAGE-OM-based public database. Denormalization tables were added to improve query performance. The MAMA database was implemented into an Oracle 9i (Release 9.0.1, Oracle Corporation) database system located on a Sun 280R server (Sun Microsystems Inc.). In the current version of MAMA, a server program interacts with this database using Java JDBC package to load or retrieve data. The open architecture of MAMA allows other developers to integrate other methods for these tasks. The MAMA database provides a centralized repository of public microarray datasets about cancer. End users have free, but limited, access to this database. They will be able to freely query about the stored microarray datasets or directly download complete datasets, but cannot modify existing data or load data into the database, which are the tasks of data curator and administrator.

### 3.2.4 Server Program

The MAMA server program is running as a Java servlet (J2EE Servlet Specification 2.3, Sun Microsystems Inc.) deployed in a Tomcat container (Apache Tomcat Version 4.1,

Apache Software Foundation). A servlet uses threads to handle concurrent requests sent by different clients and send back responses. (See Appendix C for more details about Servlet/Tomcat server.) The server program interacts with the MAMA database with Java JDBC package to query or load data, and the client program accesses data in the database through the server. The server and client programs communicate with each other through a pre-defined protocol. Beyond the client program provided by the current build, other developers can write their own as long as this protocol is implemented.

### 3.2.5 Client Program

The MAMA client program is a data analysis application executable on any computer system running JVM (Java Virtual Machine). Although it requires network connection to retrieve data from MAMA database through the server program, this program can be used as a stand-alone application. End users can load microarray datasets into the client program either by downloading them from the MAMA database or by directly importing them from text files. Afterward, they will be able to save or work with loaded datasets on their local machine. The client program has a Graphical User Interface (GUI) programmed with Java Swing to improve its user-friendliness. The software development followed the MVC (Model-View-Controller) design pattern. The ‘Model’ package defined the Java classes for data objects and maintained them in a hierarchical structure, the ‘View’ package included GUI components, such as List and Table, to render data objects, and the ‘Controller’ package implemented handlers of user events that might modify the data objects and/or the GUI components. These packages encapsulate the functions of the client program and interact with each other through software interface. (See Appendix D for more details about MVC

design.) The MAMA client also includes a data analysis package, which implements the statistical methods of microarray analysis, such as Pearson's correlation analysis or Student's t test. These methods are called by the 'Controllers' in response to the initiation of data analysis operations. The Eclipse Platform (version 3.0.1, Eclipse contributors and others, http://www.eclipse.org), an open-source product, was used for the creation, organization, and compilation of Java source codes.

### 3.2.6 File Formats

The MAMA client program accepts and processes two file formats. The first one is tab-delimited text. It is the only data format accepted by the current build for data importing, and is also used for saving matrixes of expression measurements in 'Workspaces'. The other format is XML (eXtensible Markup Language). XML documents organize data in hierarchical structure and label them with defined tags. They are machine-readable files and proper for data exchange between different computer programs. The MAMA client uses XML for the storage of all data objects in 'Workspaces' except matrixes of expression measurements, whose amount is usually too large to be processed as XML documents. The current project did not define any schema or DTD (Data Type Definition) for XML documents. Instead, it utilized the XML data-binding functions provided by Castor XML (version 0.9.6, Exolab Group, Intalio Inc., and Contributors), which could automatically map Java objects to XML documents or *vice versa*. The mapping rules were defined in an XML document, which can be downloaded together with the client program. (See Appendix L for Java-XML mapping with Castor.)

### 3.2.7 Open Source Framework

MAMA is an open-source project. Its source code will be freely downloadable. Furthermore, the three components of MAMA project: database, server, and client, are independent of each other, which means other researchers can develop their own programs to interact with any of these components as long as those proper interfaces to existing components are implemented. The current version also provides a mechanism for users to plug in their own data analysis methods into the client program. The plug-in of a method includes two steps. The first step is to create a Java class that realizes the method. This class will be able to activate a procedure to run the method. It should also implement the API (Application Program Interface), which is designed for the method category belonged to by the method. Methods sharing the same API will have the same types of inputs and outputs. For example, all methods evaluating correlation between two genes will have two arrays of expression measurements that have the equal length as its inputs, and the value of a test statistic and its corresponding p-value as its outputs. The second step of method plug-in is to register the new method by providing information about its type, name, and path of Java class. A file including all registration information can also be downloaded together with the client program. (See Appendix O for more description of method plug-in.)

### 3.2.8 Meta-Analysis Methods

A key feature of MAMA data analysis software is the availability of meta-analysis methods. Meta-analysis is often referred to as ‘Analysis of Analyses’, a statistical technique that reviews the results from multiple individual studies to draw integrated conclusions. MAMA implemented two major types of meta-analysis methods: ‘combined tests’ and

‘measures of effect size’ [64]. Combined tests are applied to the results from individual studies, such as p-values, t and z test statistics, to obtain a combined test statistic. Examples of these tests are Fisher, Winer, and Stouffer combined tests, all of which were adopted by MAMA. Compared to combined tests, which provide only the statistical significance of hypothesis tests, measures of effect sizes are more informative because ‘effect size’ represents ‘the degree to which the null hypothesis is false’. Many meta-analysis methods have been developed to deal with two types of effect size: correlation coefficients (r) and standardized mean differences between two groups (d). These methods usually utilize r or d statistics obtained from individual studies to generate a summary statistic. (See Appendix E for more description of meta-analysis methods.)

# CHAPTER IV
# RESULTS AND DISCUSSIONS

## 4.1 Data Analysis

### 4.1.1 Pilot Studies

The pilot studies analyzed two published breast cancer datasets (Table 1A), mostly the Rosetta dataset. The purposes of these studies are to:

- Evaluate the feasibility of using SEP as a prognostic index of cancer.
- Verify the confounding effect of clinical indexes on expression profiling of cancer outcome.
- Try partial correlation analysis to control the effect of confounders.
- Confirm the prognostic value of microarray data on cancer outcome.

#### 4.1.1.1 Confounding Effect of Clinical Indexes

The original study of Rosetta dataset selected 78 breast cancer patients from it to carry out expression profiling. 44 patients who did not develop recurrence and were followed up for at least five years were categorized into a good prognosis group, while the other 34 patients who recurred within five years after diagnosis were considered as having poor prognosis. This study calculated the Pearson's correlation between the expression of each gene and the recurrence outcome of patients, and 231 genes obtained significant correlation

coefficient ($|r| > 0.3$, $p < 0.01$). The current study used these genes to calculate a SEP for the same 78 patients using Formula (1) within which the correlation coefficient r of each reporter genes was taken as its weight w.

The density distribution of all 78 SEP scores was plotted in Fig. 1A and an unexpected 3-peak mode was observed. Fig. 1B, on the other hand, separately plotted the density distributions of the scores of good and poor prognosis patients. Comparison of Fig. 1A and 1B showed that the most of right and middle peaks were correspondingly composed of good and poor prognosis patients while the left peak was a mixture of patients from both prognosis groups. Shapiro-Wilk test [112] was used to test the normality of the two curves in Fig. 1B. It rejected the normality of good prognosis curve with p-value of 0.0022, but failed to reject it for the poor prognosis curve with p-value of 0.49.

The existence of the left tails of both curves in Fig. 1B suggested that although all reporter genes were identified because of their observed significant correlation to recurrence outcome, some of them might more significantly correlated to other variable(s). This suggestion could be verified if the patients located in the left peak of Fig. 1A shared some common attributes that were not possessed by the patients in the middle and right peaks. This interpretation was consistent with the observations that most patients in the left tails were ER-negative and the majority of the reporter genes (165 of 231) were also significantly correlated to ER status of patients ($p < 0.01$).

Chi-square tests were performed to evaluate the dependence of SEP scores on common clinical indexes. All 78 scores were artificially separated into two groups using -5 (left valley in Fig. 1A) as the cutoff. Clinical indexes were categorized according to the criteria used by the original study (e.g. ER: positive and negative; Grade: 1, 2, 3, and 4). Test result of all six

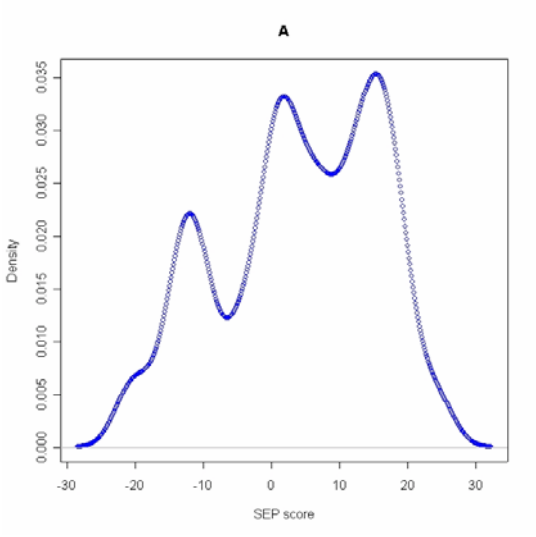



Figure 1A Density Distribution of SEP Scores of All 78 Breast Cancer Patients

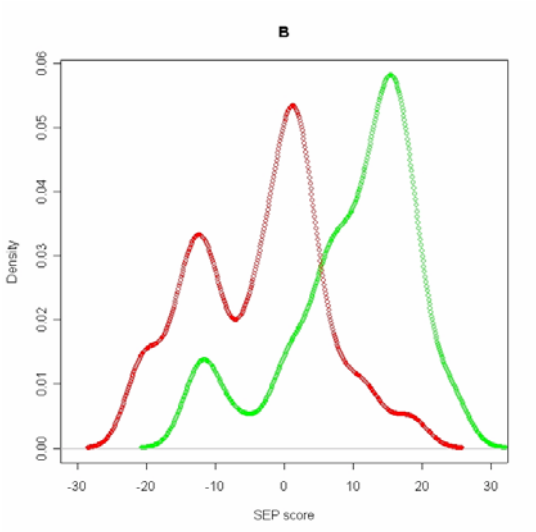



Figure 1B Density Distribution of SEP Scores of Both Prognosis Groups

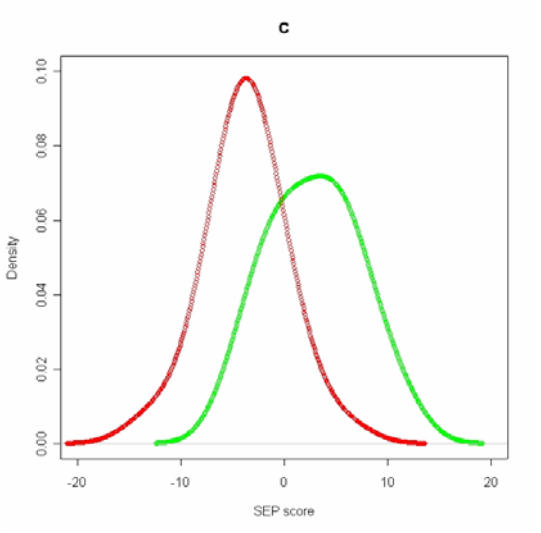



Figure 1C Density Distribution of SEP Scores of Both Prognosis Groups after Partial Correlation

**Figure 1 Density Distribution of SEP Scores**

**Correlation or partial correlation analysis was based on microarray data of 78 breast cancer patients in Rosetta dataset. (1A) The distribution of all 78 scores were plotted together. SEP was calculated with 231 reporter genes identified by the original study. All reporters had significant correlation ($p < 0.01$) to 5-year recurrence outcome of breast cancer patients. Correlation coefficient r of each reporter was used as its weight to calculate SEP. (1B) The distributions of 44 scores from good prognosis patients and 34 scores from poor prognosis patients were separately plotted. SEP score were calculated with the same 231 genes and their weight. In general, good prognosis patients were expected to have higher SEP scores. (1C) The distributions of scores from good and poor prognosis patients were plotted again after partial correlation analysis. SEP was calculated with 127 reporters that had significant partial correlation ($p < 0.01$) to 5-year recurrence after ER status of patients was controlled. Partial correlation coefficient r' was used as weight of reporters.**

available indexes were summarized in Table 2, which showed that the value of SEP was significantly dependent on tumor size, histological grade, ER and PR (Progesterone Receptor) status. Therefore, the confounding effect of these variables might have considerable influence on the expression profiling of breast cancer recurrence.

**Table 2:**
**Chi-square Tests on SEP Scores and Clinical Indexes**

d.f.: degree of freedom, number of categories of each clinic index minus 1; $\chi^2$: chi-square test statistic, measurement of the association between two variables: SEP and a given clinical index; Grade: degree of morphological abnormality of cancer cells; Angioinvasion: invasion of cancer cells into blood or lymph vessels; ER: Estrogen Receptor; PR: Progesterone Receptor.

| Clinical Index | d.f. | $\chi 2$ | p-value |
|---|---|---|---|
| **Age** | 2 | 0.51 | 0.776 |
| **Tumor size** | 1 | 10.24 | 0.014 |
| **Grade** | 3 | 13.10 | 0.014 |
| **Angioinvasion** | 1 | 0.41 | 0.520 |
| **ER status** | 1 | 42.55 | $<10^{-8}$ |
| **PR status** | 1 | 24.61 | $7X10^{-7}$ |

Fig. 2 presents a general causal model about the interrelationship of gene expression, disease outcome, and clinical indexes. In the model, both gene expression and clinical indexes are causal variables of cancer outcome while they are correlated to each other. Thus, when the gene-outcome correlation between gene and outcome is under investigation, clinical indexes are potential confounders. In Fig. 2, if both $r_{23}$ and $r_{13}$ are significant, $r_{12}$ will not represent the 'intrinsic' correlation of a gene to disease outcome. When an expression profile includes many confounded genes, the value of this profile as an independent prognostic index will be reduced because its predictive ability varies with the clinical background of patient cohorts. To avoid acquiring expression profiles with inconsistent

performance, Gruvberger suggested that expression profiling should be carried out separately for ER-positive and -negative patients [61]. However, this suggestion did not give an ultimate solution to the problem. Since ER status is not the only confounding index according to previous chi-square tests, patients need to be further sub-grouped to control other indexes. Consequently, sample size of subgroups will be too small to produce statistically meaningful results. More sophisticated statistical techniques are required to generate more general and independent prognostic index of cancer from microarray data.

This section demonstrated the advantage of using SEP to summarize expression profile into a numeric value. Common statistical analyses could be applied to this variable to provide powerful and straightforward results. Therefore, the rest part of this study would use SEP as a potential prognostic index of cancer outcome and compare it to other indexes in terms of their clinical value.

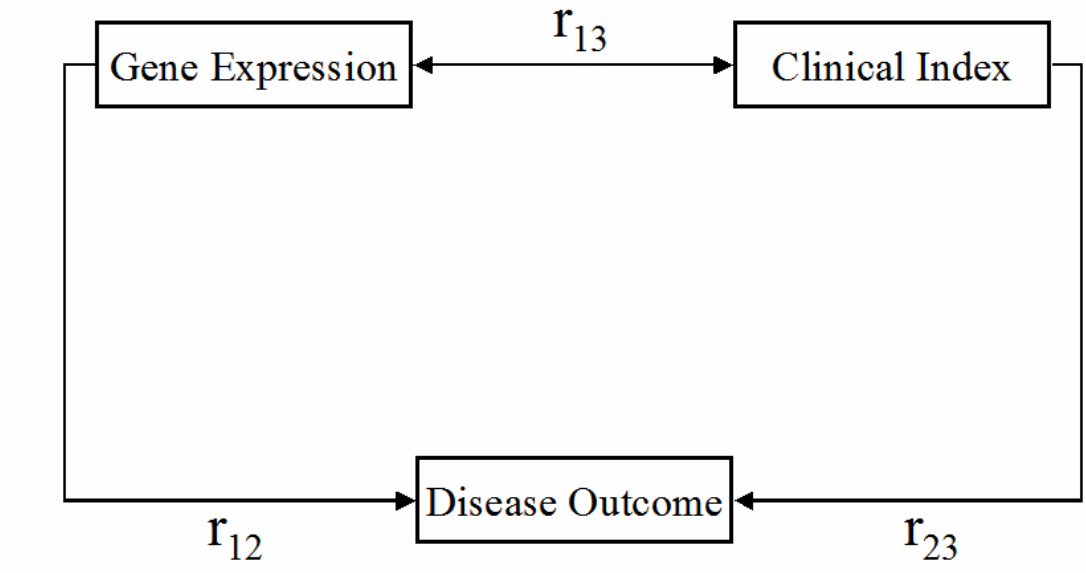



**Figure 2 Confounding Effect of Clinical Index on Gene-Outcome Correlation**

Observed correlation between gene expression and disease outcome ($r_{12}$) is intervened by the clinical index because of the gene-index ($r_{13}$) and index-outcome ($r_{23}$) correlations. Direction of arrowheads indicates causal relationships.

#### 4.1.1.2 Partial Correlation Analysis

This analysis controlled the confounding effect of ER status, which has the highest correlation to SEP according to Table 2, from the calculation of gene-recurrence correlation. The conditional expected expression levels (E) of each gene in ER-positive and -negative patients were separately estimated by calculating the group averages (Appendix A.6). Thereafter, Formula (2) was used to subtract E from the original measurements to obtain residuals. These steps were repeated for all genes to generate a data matrix composed of the residuals. This matrix was used to replace the original data matrix in subsequent steps.

A partial correlation coefficient (r') of each gene to recurrence outcome was then obtained from the residuals. Among all 19,174 Unigene clusters, 127 had significant partial correlation ($|r'| > 0.3$, $p < 0.01$), including Cyclins (B2, E2, etc.), kinases (PK428, PGK1, etc.), transcription factors (FOXM1, GTF3C1, etc.), growth factors (TGFB3, FGF18, etc.), and genes related to cytokinesis (KIF3B, PRC1, etc.). This list of reporter genes had only about half the size of the previous 231-gene list, which was not unexpected because many genes in the first list were highly correlated to ER status, the controlled variable.

SEP scores of all 78 patients were re-calculated with Formula (1) while r and L were replaced with r' and $L_{residue}$. The density distributions of the new scores corresponding to two prognosis groups are separately plotted in Fig. 1C. Both curves are bell-shaped and Shapiro-Wilk tests failed to reject their normality ($p = 0.50$ and 0.79 respectively). Although Student's t test rejected the equality of group means with $p < 0.0001$, these two curves shared noteworthy overlapping. Classification of patients had the best fit to actual observations when cutoff of SEP was -2.2. In particular, nine poor prognosis and four prognosis patients were incorrectly classified, giving an overall accuracy of 83.3%.

Logistic regression models were used to compare the expression profiles derived from regular and partial correlation analyses. Table 3 summarized the -2LL (-2 Log Likelihood) fitness and classification accuracy of various models. All models had SEP as independent variable and multi-variate models also included ER status and other clinical indexes (PR status, tumor size, grade, angioinvasion, and age of patients). Results in Table 3 suggested that:

- When SEP was acquired from partial correlation analysis, models had improved fitness.
- The model including all available indexes and SEP obtained from partial correlation had the best fitness (51.7) and accuracy (84.6%). Nevertheless, the difference of classification accuracy among models was not significant.
- Multi-variate models combining the clinical indexes and SEP had better fitness than uni-variate models of SEP.

**Table 3:**
**Comparison of SEP and Clinical Indexes Using Logistic Regression Models**

Regular Correlation: reporter genes and their weight were obtained by calculating the Pearson's correlation between expression measurements and breast cancer recurrence; Partial Correlation: reporter genes and their weight were obtained by calculating the Pearson's correlation between expression measurements and breast cancer recurrence after the effect of ER status was controlled; -2LL: -2 log likelihood, indicating the fitness of models to actual observations; Accuracy: accuracy of patient classification using the model; Intercept: initial model having no independent variable and including the constant term only; ER: Estrogen Receptor; all indexes: Age, Angioinvasion, Grade, Tumor Size, and ER/PR statuses.

| Independent Variable(s) | Regular Correlation | | Partial Correlation | |
|---|---|---|---|---|
| | –2 LL | Accuracy | –2 LL | Accuracy |
| **Intercept** | 106.8 | 56.4% | 106.8 | 56.4% |
| **SEP only** | 71.6 | 83.3% | 66.5 | 78.2% |
| **SEP + ER** | 66.7 | 79.5% | 63.8 | 80.8% |
| **SEP + all indexes** | 54.1 | 80.8% | 51.7 | 84.6% |

Although the superiority of partial correlation analysis was supported by Table 3, these results were obtained from self-adaptive processes and might include overfitting. For example, when both expression profiles were cross-validated with the Stanford dataset, the 127-gene profile did not perform better than the 231-gene profile. This observation revealed a critical drawback of partial correlation analysis. Since the conditional expected expression level (E) in Formula (2) was estimated from experimental data, extra variance and overfitting was introduced into the analyses and results. Furthermore, there existed more confounders other than ER status as showed in Table 2. Chi-square tests showed that the SEP calculated with the 127-gene profile was not dependent on ER status any more ($p = 0.67$), but still significantly correlated to PR status ($p = 0.039$), tumor size ($p = 0.019$), and histological grade ($p = 0.0002$). Although Formula (2) could be iteratively used to control all confounders, such a process would introduce even more variance and overfitting. Therefore, partial correlation analysis was not recommended by this study.

#### 4.1.1.3 Case Study: a Gene Regulatory Pathway

In addition to expression profiling of sample features, an important application of microarray data is to discover or confirm gene regulatory pathways by revealing correlated expression of genes. Partial correlation analysis can be used for this purpose because gene-gene correlation is also influenced by confounders. For example, if two genes are both highly correlated to ER status, they are very likely to have an observed correlation with each other too even they are functionally irrelevant.

The residuals obtained from controlling ER status were used to perform a 2-way hierarchical clustering of 78 breast cancer patients and 127 reporter genes (Fig. 3). Patient

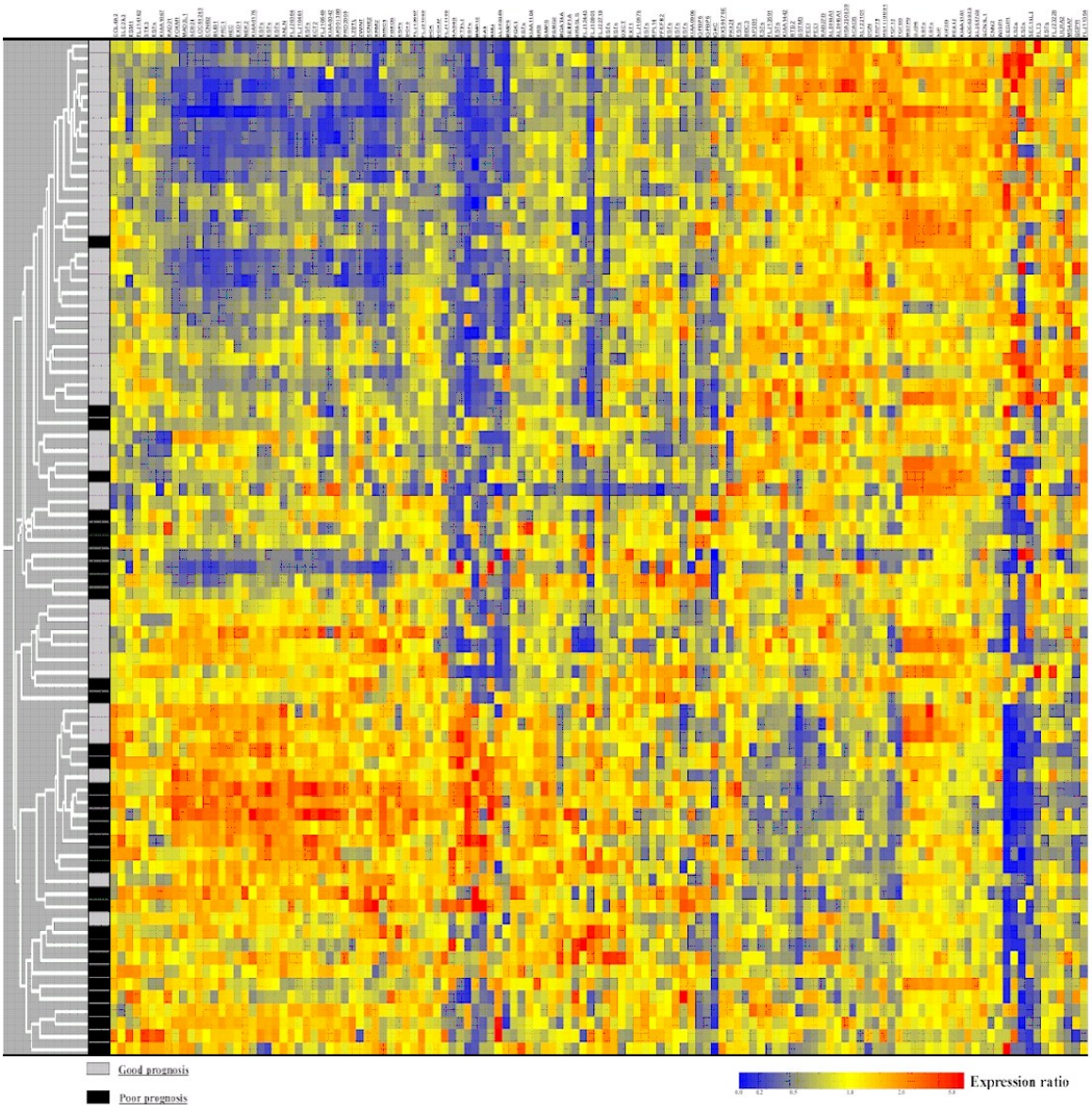



**Figure 3 Clustering of 127 Reporter Genes and 78 Breast Cancer Patients**

Results were obtained from partial correlation analysis that controlled the ER status of patients from original expression measurements of Rosetta breast dataset. Identified 127 reporter genes were horizontally clustered into two major branches, corresponding to positive (right) and negative (left) correlation to disease outcome. Similarly, 78 breast cancer patients were vertically clustered into two major branches, corresponding to good (up) and poor (down) prognosis. Each of these branches also included two sub-branches. In the case of the branch of good prognosis patients, the bigger sub-branch included 33 patients, but only four of them had poor prognosis while seven of eight patients in the smaller sub-branch had poor prognosis. The branch of poor prognosis patients had the similar pattern.

clustering deviated in two first-level branches, corresponding to two prognosis groups. There was a large sub-branch of 29 patients including only two poor prognosis ones. Genes were clustered into two first-level branches too, corresponding to positive and negative correlation to recurrence outcome. A small sub-branch of eight reporter genes was further investigated because it demonstrated a very strong gene co-expression pattern. Literature searches indicated that six of these genes have functions directly or indirectly related to cell cycle regulation. Over-expression of CCNB2 (cyclin B2) will block the exit of mitosis [113]. MAD2L1 is located upstream of CCNB2 in a known gene pathway [114, 115]. BUB1 is a direct regulator of MAD2L1 [116] and HEC is required by kinetochore recruitment of MAD1-MAD2 complex [117]. Furthermore, FOXM1 gene is a transcription factor regulating several cyclins [118] while PRC1 is a substrate of several cyclin-dependent kinases [119]. Fig. 4 presented a diagram of regulatory relationship between these genes. The expression of all six genes was highly correlated to recurrence of 78 breast cancer patients in Rosetta dataset (Table 4), which implied that this pathway played an important role in breast cancer recurrence.

The correlation between the genes described above and breast cancer recurrence was validated with testing data including 19 patients from Rosetta dataset and 48 patients from Stanford dataset. The resulting correlation coefficients (r) and corresponding p-values were listed in Table 4. As in training data, all genes had negative correlation to recurrence outcome. However, most of them did not get a significant correlation coefficient (Table 4), probably because of the relatively small sample size of testing data. Therefore, two meta-analysis methods, Fisher and Stouffer combined tests, were applied to the results obtained from two testing datasets. The right side of Table 4 gave the p-values of each gene obtained

from both tests. All p-values were significant or marginally significant while Fisher combined test was more conservative than Stouffer in general.

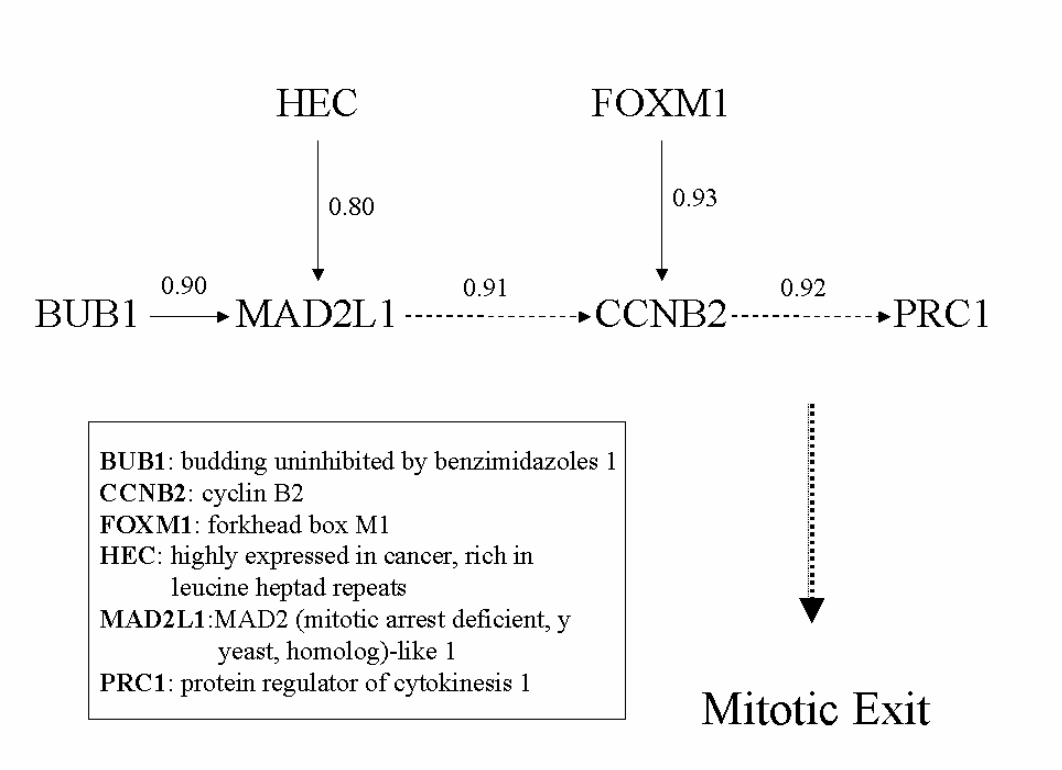



**Figure 4 A Cell Cycle-related Pathway Revealed by Partial Correlation Analysis**

All Genes were related to a spindle checkpoint pathway according to literature search. Results were obtained from the microarray data of 78 breast cancer patients in Rosetta datasets. Direction of arrowheads indicated the regulatory relationship (solid line: direct regulation; dashed line: indirect regulation). Label numbers were the Pearson's correlation coefficient (r) between each pair of genes.

Although the focus of this study was not gene-gene relationship, this case study gave an interesting example about the application of microarray data to research of gene pathway. Moreover, cross-validation results suggested that independent datasets might share some common information. Meta-analysis demonstrated its value in microarray analysis by successfully combining such information from two testing datasets.

## Table 4:

### Correlation of Genes in a Cell Cycle Pathway to Breast Cancer Recurrence

Combined test: a type of meta-analysis test combining the test statistics of individual tests; n: number of patients having non-missing measurements of the gene; r: Pearson's correlation coefficient, negative r indicated that gene was over-expressed in poor prognosis patients; Fisher: Fisher's combined tests, using p-values of individual tests as its inputs; Stouffer: Stouffer's combined test, using z test statistics of individual tests as its input.

| Gene Symbol | Unigene ID | Training Data | | | Testing Data | | | | | | Combined test | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 78 patients (Rosetta) | | | 19 patients (Rosetta) | | | 48 patients (Stanford) | | | Fisher | Stouffer |
| | | n | r | p-value | n | r | p-value | n | r | p-value | p-value | p-value |
| BUB1 | Hs.98658 | 77 | -0.41 | 0.0002 | 19 | -0.36 | 0.130 | 47 | -0.21 | 0.157 | 0.064 | 0.038 |
| CCNB2 | Hs.194698 | 78 | -0.37 | 0.0009 | 19 | -0.44 | 0.059 | N/A | N/A | N/A | 0.059 | 0.059 |
| FOXM1 | Hs.239 | 77 | -0.37 | 0.0009 | 19 | -0.30 | 0.212 | 48 | -0.29 | 0.046 | 0.034 | 0.022 |
| HEC | Hs.58169 | 78 | -0.39 | 0.0004 | 19 | -0.14 | 0.568 | 7 | -0.66 | 0.107 | 0.158 | 0.123 |
| MAD2L1 | Hs.79078 | 78 | -0.35 | 0.0017 | 19 | -0.41 | 0.080 | 47 | -0.13 | 0.384 | 0.090 | 0.064 |
| PRC1 | Hs.344037 | 78 | -0.37 | 0.0009 | 19 | -0.45 | 0.053 | N/A | N/A | N/A | 0.053 | 0.053 |

The pathway in Fig. 4 functioned as a block of mitosis exit, and the over-expression of CCNB2 and other genes was expected to slow down cell division and tumor growth, which would lead to good prognosis of patients. However, according to Table 4, the observed correlation coefficients between good prognosis and genes were all negative. This conflict might be explained by the fact that actively growing tumors had a larger portion of cycling cells than latent ones. Therefore, since above genes were known as being over-expressed during mitosis, quickly growing (poor prognosis) tumors would contain more mRNA of those genes. This observation occurred when the influence of cell cycle-dependent gene expression overwhelmed the functions of genes. This interpretation indicated the importance of biomaterial components on expression profiling.

### 4.1.2 Expression Profiling using Multiple Datasets

Although the pilot studies confirmed the confounding effect of clinical indexes on expression profiling of cancer outcome, they did not propose practical approaches to generate more generally applicable profiles. Therefore, the following studies were carried out on more than one microarray dataset to:

- Derive expression profiles from multiple independent microarray studies.
- Combine training/testing validation and bootstrap strategies to make unbiased estimation about the quality of expression profiles.
- Objectively compare SEP to currently used prognostic indexes by cross-validating of datasets.
- Verify the recurrence model of breast cancer proposed by Retsky [71], and generate an expression profile corresponding to this model.

#### 4.1.2.1 Analysis of Individual Datasets

In this section, expression profiles were generated separately from two breast cancer datasets, Rosetta and Stanford. Identical steps were applied to both datasets during the process. Patients were classified using 3-year recurrence outcome as the cutoff, according to the recurrence model of Retsky (Appendix A.1). The size of the resultant prognosis groups was given in Table 1A. Source expression data were pre-processed and filtered as described in Chapter 3.

The first step was to split patients into training/testing subgroups. About two-thirds patients of each dataset were randomly selected into a training subgroup, leaving the rest for testing the expression profile derived from the training data (Appendix A.4). The testing

results would be critically influenced by sampling bias since the sample size of the current datasets could not provide satisfying statistical power. Consequently, a 'no replacement bootstrap' approach was performed to eliminate the sampling bias from the results. This approach repeatedly re-sampled patients into training/testing subgroups and executed identical analyses on each combination. The results obtained from all re-samplings were collected and summarized to draw an unbiased final conclusion.

The differential expression of each gene in patients having good and poor prognosis was assessed using the data of each re-sampled training subgroup. A hypothesis test about gene-outcome correlation was performed on every gene. Since it was a two-group comparison problem, Wilcoxon Rank Sum Test (RST) was used for the hypothesis test. As a non-parametric method, RST was less powerful than Student's t test, but it did not assume the normality of expression measurements, which was violated by many genes in microarray data. RST calculated a Z test statistic for each gene. When there were at least eight measurements in each prognosis group, RST Z followed standard normal distribution N (0, 1). A gene would have positive Z if it was over-expressed in patients having good prognosis (Appendix A.7). Given the results of RST tests, all genes were ranked according to the magnitude of their Z values. Genes having the highest magnitude of Z statistic were top-ranked and selected into an expression profile as reporters. The number of reporters in a profile was denoted as N. Some reporters might be false positives because of the problem of multiple hypothesis tests. Increasing N would improve the sensitivity of reporter selection, but reduce the specificity at the same time. Instead of arbitrarily setting the value of N, the current study applied a stepwise procedure to find an N that would optimal balance the sensitivity and specificity. This procedure increased the value of N one by one from 1 to 100,

and at each step, the SEP scores of testing patients were calculated with the top-ranked N genes. The SEP scores were calculated using Formula (1), while the weight w of each gene was its RST Z statistic and the expected expression level E was estimated from the training data (Appendix A.8). The resultant SEP scores were used to classify testing patients with cutoff = 0. Patients with positive or negative scores would be classified into good or poor prognosis group respectively. The accuracy of classification was obtained by comparing SEP-classification to actual patient outcomes. SEP scores of a testing subgroup were also used to build an ROC curve. The area under the curve (AUC) was an index indicating the ability of SEP to differentiate good and poor prognosis patients. Since ROC curve took the relative quantity of each score into account, it was more informative and powerful than dichotomous classification.

Both datasets had totally 10,000 bootstrap re-samplings, each of which identically went through the above steps. Consequently, unbiased bootstrapping estimation of test statistics was concluded from all re-samplings. The upper half of Table 5A shows the median and 90% Confidence Interval (CI) of SEP-classification accuracy when N was 100. Bootstrapping statistics of two datasets were close to each other, although Rosetta dataset generally had better results. The left column in this table was the size-weighted averages summarized from both datasets. To calculate these values, the test statistics of both datasets were weighted by the size of the corresponding testing subgroups and averaged at each re-sampling. The final bootstrapping statistics were concluded from the size-weighted averages of all re-samplings. As in Table 5A, the median size-weight average accuracy was just above 70% and a symmetric 90% CI ranged from 61% to 80%. Eight of the 10,000 re-samplings got size-weighted average accuracy lower than 50%, giving a 0.0008 bootstrapping p-value in favor

of that SEP of testing patients suggested their recurrence outcome. Table 5B presented the median and 90% CI of AUC when N was 100. The median of size-weighted average AUC was 0.767. Rosetta dataset got better results than Stanford data again, probably because of its relatively larger sample size and/or less diverse clinical background of patients.

**Table 5:**
**Bootstrapping Test Statistics Collected from 10,000 Re-samplings**

10,000 bootstrapping re-samplings were performed on both individual breast cancer datasets and their combination. At each re-sampling, patients of each dataset were split into training/testing subgroups. Each dataset ranked genes based on RST Z statistic applied to training data and selected 100 top-ranked genes as reporter. These reporters and their weight were used to calculate SEP of testing patients. Resultant scores were used to classify testing patients and build ROC curves. Both classification accuracy and AUC were adjusted by size of testing subgroups to get the size-weighted averages. Bootstrapping median and 90% CI of these statistics were listed in tables.

Table 5A: Classification Accuracy of SEP Scores

| Training Dataset | Bootstrapping Statistic | Testing Dataset | | |
|---|---|---|---|---|
| | | Rosetta | Stanford | Size-weighted Avg. |
| **Individual** | 5% high | 84.00% | 83.33% | 80.00% |
| | **Median** | **71.43%** | **70.00%** | **70.59%** |
| | 5% low | 58.06% | 55.00% | 60.98% |
| **Combined** | 5% high | 83.33% | 85.00% | 80.43% |
| | **Median** | **71.43%** | **71.43%** | **71.11%** |
| | 5% low | 58.33% | 56.52% | 61.54% |

Table 5B: Area of ROC Curves (AUC) Built with SEP Scores

| Training Dataset | Bootstrapping Statistic | Testing Dataset | | |
|---|---|---|---|---|
| | | Rosetta | Stanford Set | Size-weighted Avg. |
| **Individual** | 5% high | 0.895 | 0.902 | 0.860 |
| | **Median** | **0.775** | **0.764** | **0.767** |
| | 5% low | 0.640 | 0.604 | 0.668 |
| **Combined** | 5% high | 0.903 | 0.933 | 0.877 |
| | **Median** | **0.786** | **0.799** | **0.789** |
| | 5% low | 0.654 | 0.636 | 0.689 |

The changing of size-weighted average accuracy and AUC with N were separately plotted in Fig. 5A and 5B. The middle curve in each figure corresponded to the medians while the other two curves parenthesized the ranges of 90% CI. Although all curves generally ascended with the increasing of N, they were not linear. They went up dramatically at the beginning and reached a plateau when N was around 60, suggesting that the classification and differentiation ability of expression profiles were about to get to their maximum. Consequently, it was empirically decided that the sensitivity and specificity of reporter selection were optimally balanced at N = 60. Furthermore, the width of 90% CI in both figures had no noticeable change in both figures since N was larger than 5, indicating that increasing N would improve the performance stability of SEP as a classifier.

In the next step, each gene was assigned a final rank from each dataset by counting how many times it was ranked top-100 by the dataset through all 10,000 re-samplings. The 60 genes having the most counts were selected to make an expression profile of breast cancer 3-year recurrence. The weight of each reporter was its RST Z statistic calculated with the data of all patients in the dataset. Both datasets got a 60-gene profile. The complete lists are presented in Appendix F. Both 60-gene profiles were precise classifiers when they were self-validated by the same datasets generating them. SEP scores had 79.3% classification accuracy and 0.89 AUC in Rosetta dataset, and 82.3% accuracy and 0.93 AUC in Stanford dataset. These results clearly had overfitting because of the existence of false positives in expression profiles.

The microarray analysis procedure developed in this section was able to provide unbiased estimation about the quality of expression profiles. Therefore, this procedure could be used to compare different strategies of expression profiling. Bootstrapping test statistics obtained

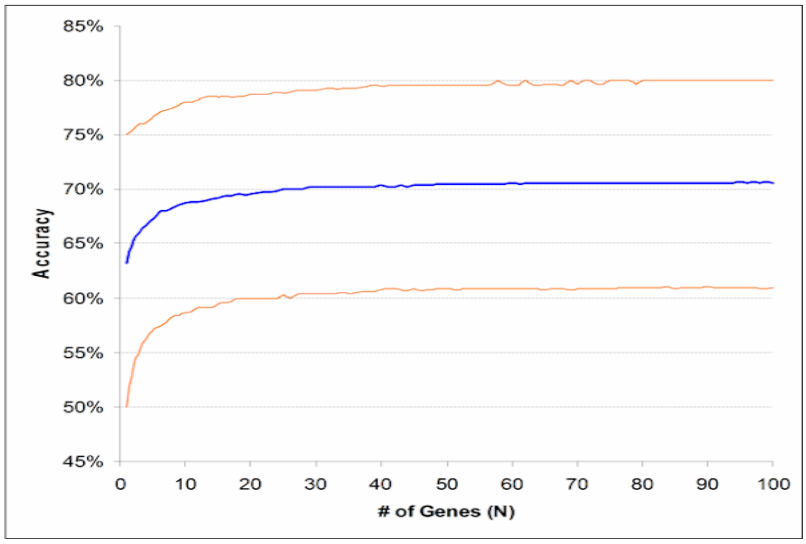



Figure 5A Classification Accuracy of SEP Scores

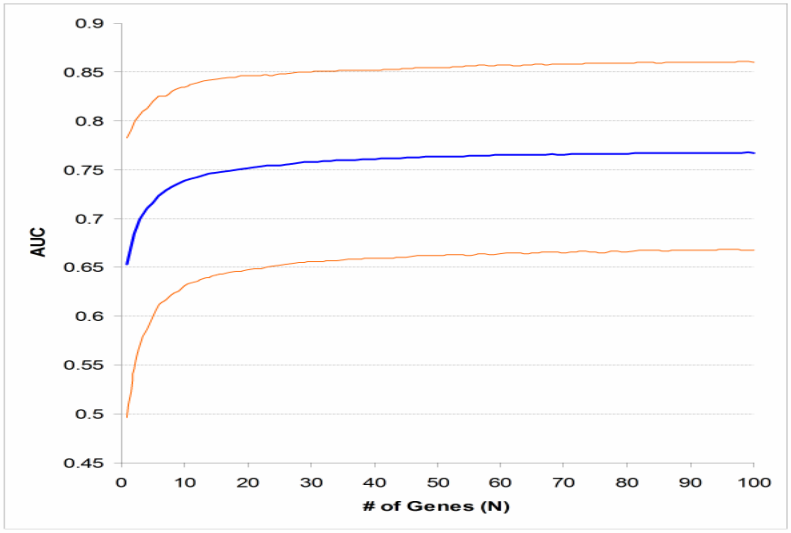



Figure 5B Area of ROC Curves (AUC) Built with SEP Scores

## Figure 5 Bootstrapping Statistics Separately Obtained from Breast Cancer Datasets

**Changing of average test statistics was traced with number of reporter genes in expression profiles (N). At each of 10,000 re-samplings, SEP scores of testing patients were calculated with N reporters and their weight derived from the training patients, separately in Rosetta and Stanford breast cancer datasets. The averages were adjusted by the size of testing subgroups. Three bootstrapping statistics were reported in each figure. The blue line represents the bootstrapping median of size-weighted averages while the orange lines parenthesize the bootstrapping 90 CI of size-weighted averages. (5A) Testing patients were classified according to their SEP scores. The accuracy of classification was obtained by comparing SEP-classification to actual events. (5B) SEP scores of testing patients were used to build ROC curves. The area under ROC curve represented the ability of SEP to differentiate patients.**

from two datasets had no significant difference, suggesting that they had similar quality and were potentially suitable for multi-dataset analysis. Furthermore, stepwise procedure denied the necessity of high sensitivity of reporter selection. The influence of false positives on quality of expression profiles will be discussed in last section of this chapter.

#### 4.1.2.2 Cross-validation of Two Datasets

In last section, two breast cancer datasets had similar results in terms of bootstrapping test statistics. However, the 60-gene profiles generated from these datasets barely overlapped with each other (Appendix F) although reporters of both profiles were selected from the same 5,569 Unigene clusters. Only two genes, BUB1 and LRP8, appeared in both profiles. The following steps cross-validated the 60-gene profile of each dataset using the data of the other dataset.

SEP of validating patients was calculated with Formula (1), using the 60 reporters and their weights obtained from the validated dataset. Expected expression level E of each gene was set as 0 by default, assuming that patients of both datasets were sampled from the same population. SEP scores of validating patients were used to classify patients with cutoff = 0. Consequently, 48 Stanford patients were correctly classified by Rosetta profile, giving an accuracy of 77.4% ($p < 0.0001$, 90% CI [69-86%]); and 58 Rosetta patients were correctly classified by Stanford profile, giving an accuracy of 70.3% ($p = 0.0002$, 90% CI [62-79%]). SEP scores were also used to build ROC curves. Classification accuracy and AUC results were listed in Table 6. The size-weighted averages were 73.6% and 0.795 respectively, which are impressive results considering the difficulty of cancer prognosis.

**Table 6:**
**Cross-validation of Expression Profiles Derived from Breast Cancer Datasets**

60 reporters top-ranked by each validated dataset and their weight were used to calculate a SEP for each patient in the validating dataset. Resultant SEP scores were used to classify validating patients with cutoff equal to 0 and to build ROC curves. Subsequent classification accuracy and AUC were listed. Average of these statistics was adjusted by the sample size of validating datasets.

| Validated Dataset | Validating Dataset | Accuracy | AUC |
|---|---|---|---|
| **Rosetta** | **Stanford** | 77.42% | 0.808 |
| **Stanford** | **Rosetta** | 70.73% | 0.786 |
| **Size-weighted Average** | | 73.61% | 0.795 |

The SEP of validating patients was compared to known prognostic indexes of breast cancer using logistic regression models. Each uni-variate model was built with an individual index as the independent variable. The fitness of models to actual observations of recurrence outcome was listed in Table 7A, which showed that the models of both datasets had the smallest -2LL and the largest AUC when SEP was the independent variable. This result suggested that SEP was superior to all the other indexes. Histological grade had the best performance except for SEP and surprisingly, models using tumor size as the independent variable were among those having the worst fitness. Multi-variate models jointly including all available indexes were also built since it was better practice to synthesize multiple prognostic indexes to make clinical decisions. A forward stepwise procedure was applied to generate these models. At each step, one independent variable, which would improve model fitness better than all the remaining indexes, was added into the model. Table 7B listed the sequential addition of the indexes and the consequent statistics of the models. When SEP was used as an independent variable, it was always the one to be added first. The SEP-included multi-variate models of two datasets had AUC equal to 0.858 and 0.837, the best results in this study. When SEP was excluded from the models, other indexes were added into models

in similar sequence. The model fitness difference between models with and without SEP was marginally significant. In the case of Stanford dataset, the model without SEP reduced the final AUC by 0.04 and increased the -2LL by 3.3. This result supported the first hypothesis of the current study: microarray data provides extra information about cancer outcome beyond currently used clinical indexes. This conclusion was objective since overfitting in results was avoided by cross-validation process.

**Table 7:**
**Comparison of Prognostic Indexes using Logistic Regression Models**

ER: Estrogen Receptor status; PR: Progesterone Receptor status; Size: tumor size; Node: lymph node metastasis; Grade: degree of morphological abnormality of cancer cells -2LL: -2 log likelihood, smaller -2LL indicated better model fitness. Each uni-variate model (Table 6A) had only one independent variable. Multi-variate models (Table 7B) were built with forward stepwise procedure, which added independent variables into a model one by one. Table 7B also shows the comparison of multi-variate models with or without SEP.

**Table 7A: Fitness (-2LL) of Uni-variate Models**

| Dataset | Null | SEP | ER | PR | Size | Node | Grade | Age |
|---|---|---|---|---|---|---|---|---|
| **Rosetta** | 108.7 | 92.2 | 98.0 | 103.5 | 101.2 | N/A* | 94.5 | 95.5 |
| **Stanford** | 83.6 | 66.5 | 79.1 | N/A | 80.9 | 79.7 | 70.1 | 83.5 |

**Table 7B: Fitness (-2LL) of Multi-variate Models**

| Dataset | | Step 1 | Step 2 | Step 3 | Step 4 | Step 5 | Step 6 |
|---|---|---|---|---|---|---|---|
| **Rosetta** | **With SEP** | **SEP** | **Age** | **Grade** | **Size** | **ER** | **PR** |
| | | 92.2 | 82.7 | 79.9 | 78.6 | 78.2 | 76.9 |
| | **Without SEP** | **Age** | **Grade** | **ER** | **Size** | **PR** | |
| | | 95.5 | 84.8 | 81.5 | 79.8 | 79.2 | |
| **Stanford** | **With SEP** | **SEP** | **ER** | **Grade** | **Node** | **Size** | **Age** |
| | | 63.5 | 61.3 | 60.2 | 59.5 | 59.3 | 58.9 |
| | **Without SEP** | **Grade** | **ER** | **Size** | **Node** | **Age** | |
| | | 70.6 | 65.2 | 63.2 | 62.5 | 62.2 | |

Cross-validation results suggested that Rosetta and Stanford datasets share common information about prognosis of breast cancer. Therefore, such common information would constitute more general expression profiles if it could be extracted from multiple datasets. It was further implied that high sensitivity of reporter selection was unnecessary. Both 60-gene profiles performed well on the validating patients although the little overlapping between them suggested that both lists missed many true positives.

#### 4.1.2.3 Combination of Individual Datasets

A straightforward strategy was used in this study for multi-dataset expression profiling. It directly combined the training subgroups of two datasets after each bootstrapping re-sampling. No extra data processing was necessary for this combination as long as both datasets had been normalized to consistent scales. This assumed that patients of independent datasets were sampled from the same general population, which was also required by other cross-dataset analyses including meta-analysis.

Reporter genes were selected from the combined subgroups with the same procedure performed on the individual subgroups. Thereafter, reporters and their weight were separately verified by both testing subgroups. The upper half of Table 5A and 5B gave the median and 90% CI of SEP-classification accuracy and AUC of 10,000 re-samplings when N was 100. Bootstrapping test statistics of size-weighted averages were also listed in these tables, which were generally higher than those obtained from the individual datasets. The median size-weighted accuracy and AUC were raised by 0.52% and 0.022 respectively. However, the difference was not significant. After 10,000 re-samplings, the comparison of results obtained from these two strategies showed that the difference in AUC had a one-tailed

bootstrapping p-value equal to 0.26. SEP scores incorrectly classified more than half testing patients only in five re-samplings when the cutoff was 0 (bootstrapping p = 0.0005).

Fig. 6A and 6B compared the median size-weighted average accuracy and AUC obtained from individual datasets (blue lines) and the combined dataset (red lines). Blue curves were higher than the corresponding red curves at the beginning, which indicated that individual datasets were good at consistently identifying a few 'true' positive reporters having the highest ranks. These reporters and their weights were dataset-specific, so they did not perform on the combined dataset as well as on the individual ones. However, red lines grew up faster in both figures and they were generally above the corresponding blues lines after N was about 10.

Same as the individual datasets, the combined dataset gave each gene a final rank according to how many times it was ranked within top-100 across the 10,000 re-samplings. The counts of top-300 genes obtained from both individual datasets and the combined dataset were plotted in Fig. 7. It illustrated that the specificity of reporter selection was low with given data. For example, the $100^{th}$ genes of all three datasets had less than one-third probability to be selected into top 100 by any re-sampling. This observation explained why two re-samplings could generate fairly different expression profiles. Nevertheless, the combined dataset selected reporters, especially those finally ranked between 30 and 150, more specifically than the individual ones. Theoretically, increased sample size accounted for this improvement.

The 60 genes top-ranked by the combined dataset and their weight defined an expression profile associated to 3-year recurrence of breast cancer. This was an optimal profile achievable with the current data and methods. Table 8 listed some of these genes by giving

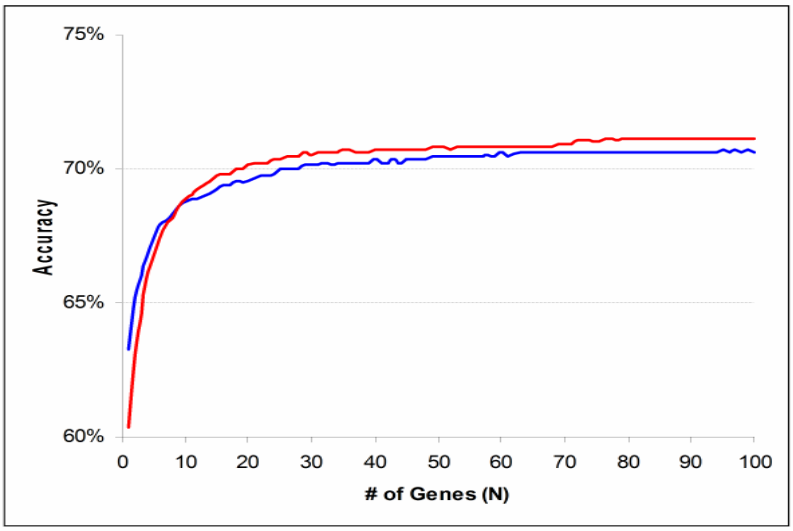



Figure 6A Comparison of Classification Accuracy between Single-dataset and Multi-dataset Profiling

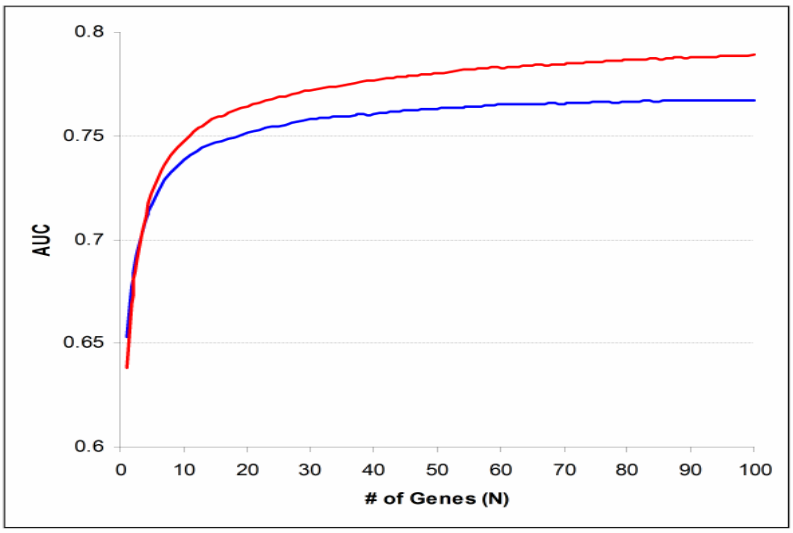



Figure 6B Comparison of ROC Curve Area (AUC) between Single-dataset and Multi-dataset Profiling

## Figure 6 Comparison of Expression Profiling Strategies

Changing of average test statistics was traced with number of reporter genes in expression profiles (N). At each of 10,000 re-samplings, SEP scores of testing patients were calculated with N reporters and their weight derived from the training patients of individual datasets or their combination. The averages were adjusted by the size of testing subgroups. Both blue lines in the figures were the same bootstrapping medians as in Figure 5A and 5B. The red lines were made of bootstrapping medians derived after the integration of two datasets. The two lines in each figure intercept when N was less than 10, indicating that the data integration strategy became superior afterwards. (6A) Expression profiling using individual datasets or combined dataset was compared in terms of patient classification accuracy of SEP scores. The combined dataset increased bootstrapping median of classification accuracy by 0.52% when N was 100. (6B) Expression profiling using individual datasets or combined dataset was compared in terms of area of ROC curves built with SEP scores. The combined dataset increased bootstrapping median of AUC by 0.022 when N was 100.

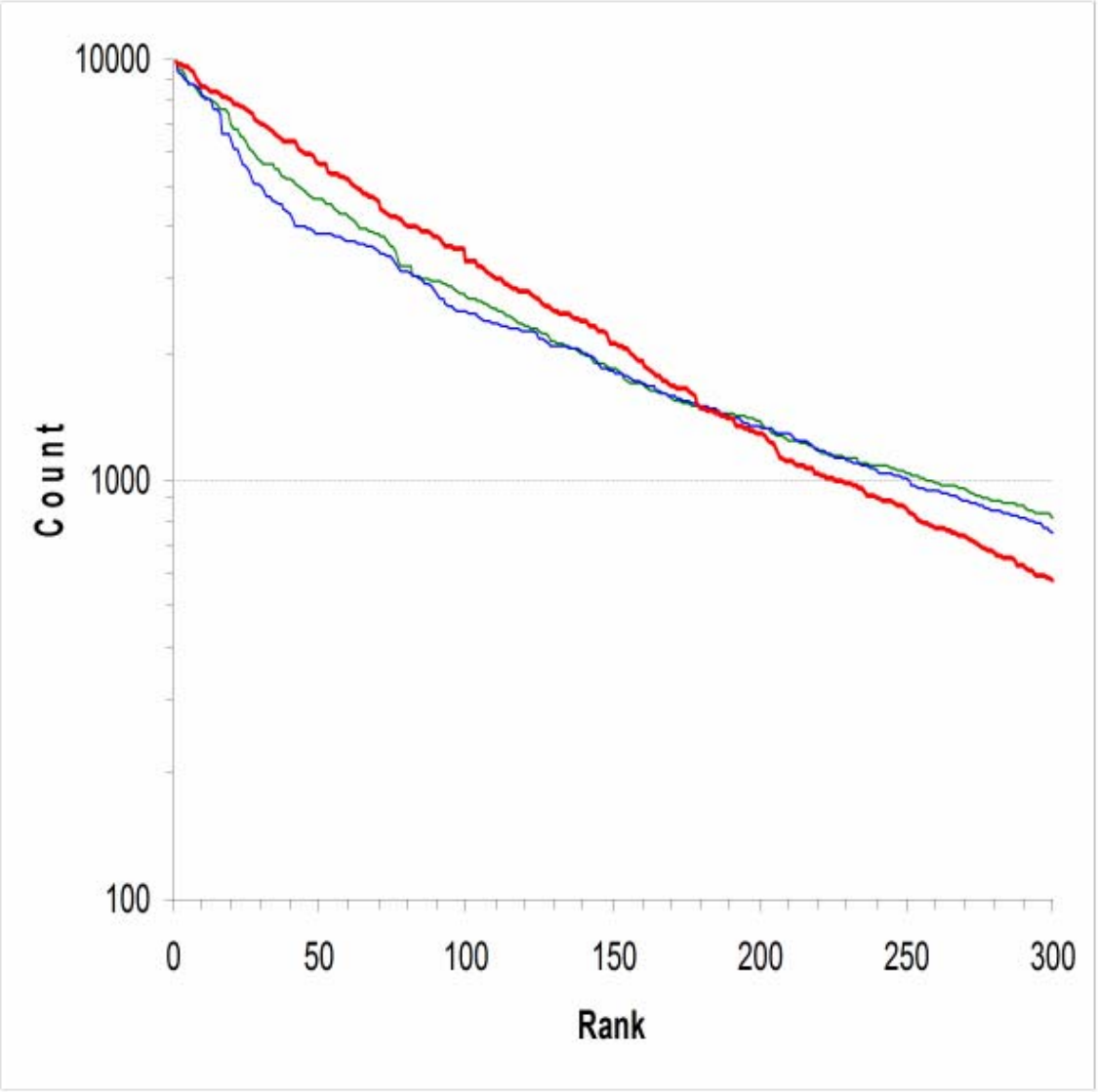



**Figure 7 Comparison of Reporter Selection Consistence**

Two individual breast cancer datasets and their combination ranked genes based on how many times they were selected into the top-100 reporter lists across 10,000 bootstrapping re-samplings. The counts of the top-300 genes of each dataset are plotted in this figure. (blue line: Stanford dataset, green line: Rosetta dataset, red line: the combined dataset)

**Table 8: Examples: Reporter Genes of 3-year Breast Cancer Recurrence**

Count: how many times a gene was ranked within top-100 across 10,000 bootstrap re-samplings by the dataset; Rank: the final rank assigned to the gene based on its count.

| Unigene ID | Gene Symbol | Full Name | Dataset (Rank / Count) | | |
|---|---|---|---|---|---|
| | | | Combined | Rosetta | Stanford |
| Hs.287472 | BUB1 | BUB1 budding uninhibited by benzimidazoles 1 homolog (yeast) | 3 / 9,681 | 3 / 9,681 | 3 / 9,681 |
| Hs.410784 | LRP8 | low density lipoprotein receptor-related protein 8 | 24 / 6,547 | 24 / 6,547 | 24 / 6,547 |
| Hs.171834 | PCTK1 | PCTAIRE protein kinase 1 | 11 / 8,047 | 11 / 8,047 | 11 / 8,047 |
| Hs.435861 | SCUBE2 | signal peptide, CUB domain, EGF-like 2 | 13 / 8,428 | 13 / 8,428 | 13 / 8,428 |
| Hs.411509 | GSTP1 | glutathione S-transferase pi | 60 / 4,211 | 60 / 4,211 | 60 / 4,211 |
| Hs.1657 | ESR1 | estrogen receptor 1 | 50 / 3,858 | 50 / 3,858 | 50 / 3,858 |
| Hs.85137 | CCNA2 | cyclin A2 | 1 / 9,862 | 1 / 9,862 | 1 / 9,862 |
| Hs.169946 | GATA3 | GATA binding protein 3 | 5 / 8,768 | 5 / 8,768 | 5 / 8,768 |
| Hs.82906 | CDC20 | CDC20 cell division cycle 20 homolog (S. cerevisiae) | 120 / 2,261 | 120 / 2,261 | 120 / 2,261 |
| Hs.267659 | VAV3 | vav 3 oncogene | 2 / 9,732 | 2 / 9,732 | 2 / 9,732 |
| Hs.12272 | BECN1 | beclin 1 (coiled-coil, myosin-like BCL2 interacting protein) | 1 / 9,991 | 1 / 9,991 | 1 / 9,991 |
| Hs.153752 | CDC25B | cell division cycle 25B | 3,615 / 1 | 3,615 / 1 | 3,615 / 1 |
| Hs.9589 | UBQLN1 | ubiquilin 1 | 228 / 989 | 228 / 989 | 228 / 989 |
| Hs.79241 | BCL2 | B-cell CLL/lymphoma 2 | >3,257 / 0 | >3,257 / 0 | >3,257 / 0 |
| Hs.69771 | BF | B-factor, properdin | 1 / 9,866 | 1 / 9,866 | 1 / 9,866 |
| Hs.432750 | HPN | hepsin (transmembrane protease, serine 1) | 4 / 9,647 | 4 / 9,647 | 4 / 9,647 |

the counts and ranks of each reporter obtained from three datasets. (See Appendix F.1 for complete lists of reporter genes.) Two genes, BUB1 and LRP8, were presented in all three 60-gene profiles while 15 others (CDC20, BECN1, etc.) were only within the profile of the combined dataset. These 17 genes got higher ranks from the combined dataset because of their low inter-dataset variance. Two well-known molecular markers of breast cancer, BCL2 [120] and ESR1 [121], were ranked $4^{th}$ and $30^{th}$ by the combined dataset.

To explore the functions of identified reporters, all genes in the final top-60 lists were mapped to the 'Biological Process' domain of Gene Ontology. Fig. 8 illustrated the categorization of these genes using GO and the numbers of genes in each category. According to this figure, most reporters have been related to important cellular processes such as cell cycle and transcription. Some sample reporters identified from the combined dataset and their GO categories were:

- GO:0007049 || Cell cycle || CCNA2, CDC20, KIFC1, etc.
- GO:0007165 || Signal transduction || ESR1, LRP8, EXT1, etc.
- GO:0006350 || Transcription || TFDP1, GATA3, TLE1, etc.
- GO:0050896 || Response to stimulus || BECN1, ACTL6A, BCL2, etc.
- GO:0044267 || Protein modification || UBQLN1, BUB1, CDC25B, etc.

Analyses in this section verified the second hypothesis of the current study: expression profiling across multiple datasets improved the quality of expression profiles. According to Fig. 7, this improvement was probably the consequence of higher reporter selection specificity. In the last section of this chapter will give more discussion on this topic. The 60-gene profile derived from the combined dataset was recommended by this study as a valuable prognostic index of breast cancer. However, no more published datasets were available to

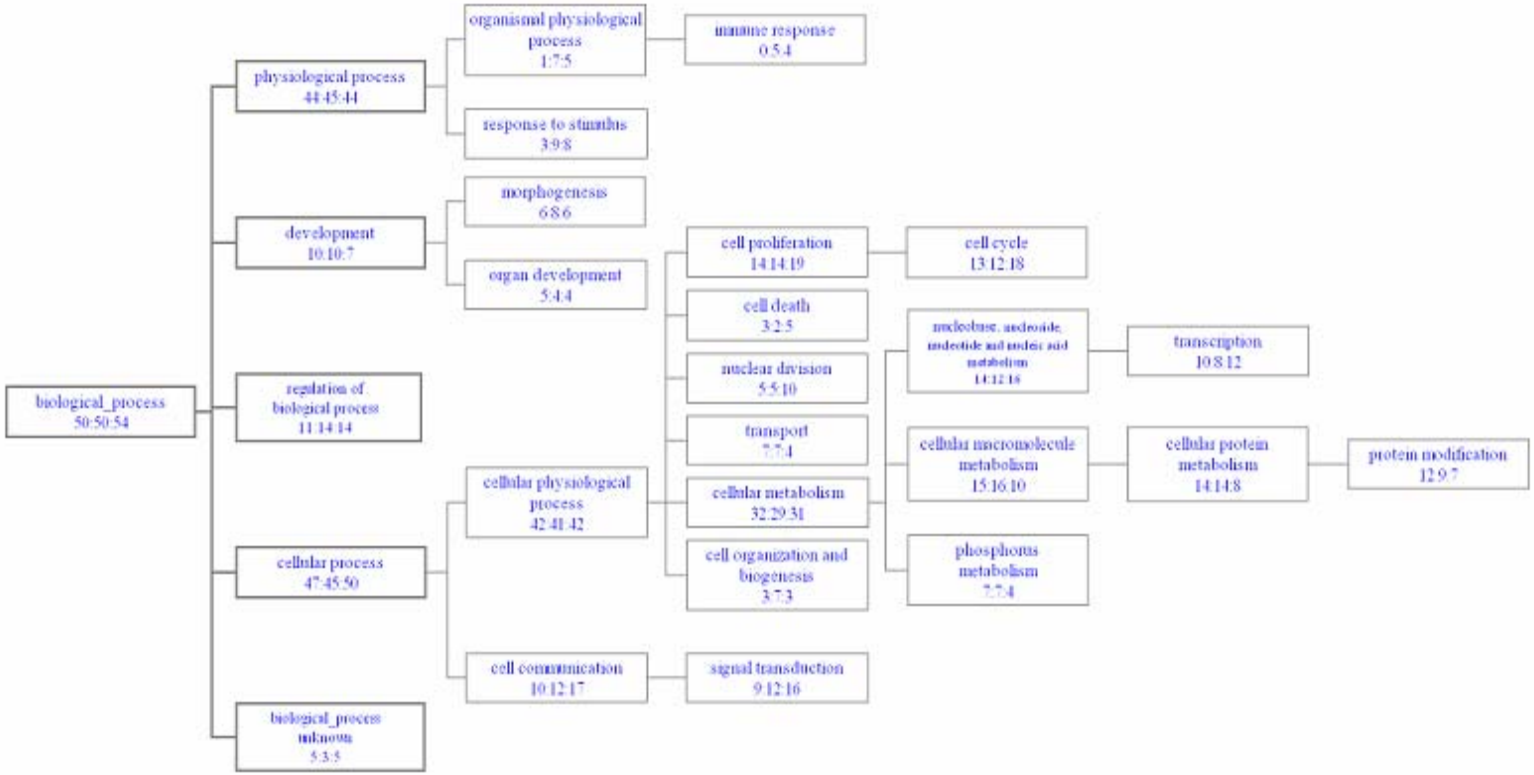



**Figure 8 Mapping of Reporter Genes to Gene Ontology**

Top-ranked 60 reporters by both individual datasets and the combined dataset were categorized. To simplify this diagram, only major categories under 'biological process' domain were presented. The numbers of reporters in each category were listed in the boxes, sequentially corresponding to the Rosetta, Stanford, and combined datasets.

verify or further improve this profile. Up till now, all microarray analyses have been performed on breast cancer datasets. To verify the methods and results of previous studies, microarray datasets about lung cancer were also investigated in the following section.

#### 4.1.2.4 Results from Lung Cancer Datasets

The microarray analyses of this section involved four published datasets about lung cancer (Table 1B). Because of their small sample sizes, Stanford and Ontario datasets were only used to validate the expression profiles derived from the other two datasets. Harvard and Michigan datasets were both generated from oligonucleotide gene chips produced by Affymetrix Inc. All genes were mapped to unique Unigene clusters. 4,036 clusters presented in both Harvard and Michigan datasets were used for expression profiling. Only adenocarcinoma patients were analyzed due to the high diversity of lung cancer subtypes. All four datasets provided clinical data and survival outcome of patients, but the recurrence information was incomplete. Literature searches failed to find a temporal model about lung cancer outcome similar to the model proposed by Retsky about breast cancer. Alternatively, lung cancer patients were classified according to their 2-year survival outcome. This classification was based on the fact that about 60% of invasive lung cancer patients did not survive more than two years after diagnose [122].

Similar to what was applied to the breast cancer datasets, each lung dataset was analyzed with the following steps:

- About two-thirds patients were randomly selected into a training subgroup.
- Genes were ranked according to Wilcoxon RST performed on data of individual training subgroups and their combination.

- Top-ranked N genes and their RST Z were used to calculate SEP of the testing patients.
- Resultant SEP scores were used as a 2-year prognostic index of lung cancer survival.
- Above steps were repeated for 1,000 times.
- Both individual datasets and their combination assigned a final rank to each gene according to how many times it was ranked within top 100 by all re-samplings.

Fig. 9 plots the size-weighted median and 90% CI of AUC summarized from 1,000 bootstrapping re-samplings. The blue and read curves respectively correspond to individual and the combined datasets. All curves grew up in the same pattern as what was observed in Fig. 5B when N was increased from 1 to 100. The red lines were generally above corresponding blue lines, confirming that the combined dataset generated more discriminative profiles. The difference of median AUC was 0.022 at N = 100. However, comparing to Fig. 6B, the curves of median AUC were located at an obviously lower level. Specifically, the median AUC of the combined dataset dropped from 0.789 to 0.685 at N = 100. The classification accuracy of SEP had similar results and it bootstrapping p-value was 0.029, in favor of rejecting the null hypothesis of 50% accuracy.

The availability of Ontario and Stanford lung datasets made it possible to validate the expression profile derived from the combination of Michigan and Harvard datasets. Because of inconsistent microarray design, many identified reporter genes were not presented in the validating datasets. N of expression profile was increased from 60 to 100. Respectively, 53 and 78 of 100 reporters identified from the combined dataset were found in Ontario and Stanford datasets. These genes and their weight were used to calculate SEP of validating patients, and patients were classified into two prognosis groups with the cutoff equal to 0.

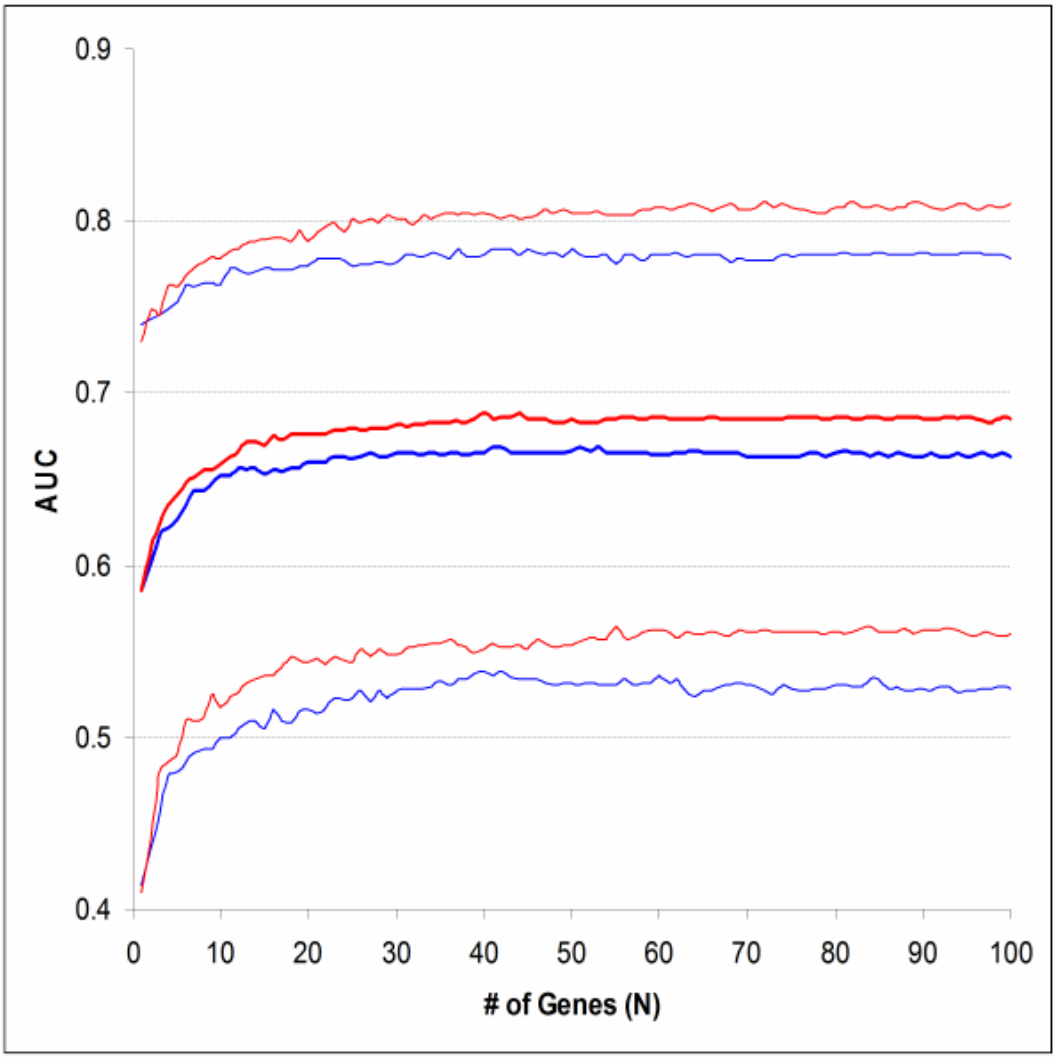



**Figure 9 Bootstrapping ROC Curve Statistics Obtained from Lung Cancer Datasets**

Changing of test statistics was traced with the number of reporter genes in expression profiles (N). 1,000 bootstrapping re-samplings were carried out. The blue lines represent the bootstrapping medians and 90% CIs of size-weighted average AUC when expression profiles were separately derived from individual datasets. The red lines represent the corresponding test statistics derived from the combination of two datasets. The combined dataset increased median AUC by 0.022 when N was 100.

The 2-year survival of 14 Stanford and 9 Ontario patients were correctly classified, giving an overall 76.7% accuracy ($p = 0.005$, 90% CI [62-92%]). ROC curves were separately built with SEP scores of each validating dataset. The size-weighted average AUC of two curves was 0.82. When the same patients were used to validate the 100-gene profile derived from the Michigan dataset, the size-weighted average accuracy and AUC is 66.7% and 0.77. (See Appendix F.2 for complete lists of reporter genes.)

Because of insufficient patient follow-up, some patients in the validating datasets could not be classified into either prognosis group, which reduced the statistical power of validation. Kaplan-Meier survival curve, a technique frequently used in clinical research to deal with incomplete medical record, was generated with all 40 patients in the original datasets. These patients were categorized into two prognosis groups according to their SEP scores and survival curves were built based on the classification and patient follow-up data. Fig. 10A and 9B respectively showed the survival curves created with the combined dataset and Michigan dataset as the training data. The two curves in Fig. 10A were separated with a p-value equal to 0.029, while those in Fig. 10B were insignificantly separated ($p = 0.12$).

The analyses of lung cancer datasets further verified the superiority of multi-dataset expression profiling. However, as a prognostic index of lung cancer, SEP was not as differential as it was for breast cancer. An obvious reason of this difference was the smaller sample sizes of the lung datasets. The profiling quality might also be influenced by how patients were classified. Supported by the existing recurrence model, classification of breast cancer in this study might more ‘intrinsically’ reflect the difference of patients at genomic level. Nevertheless, when the profile derived from the combined dataset was validated by two other datasets, the results were satisfying.

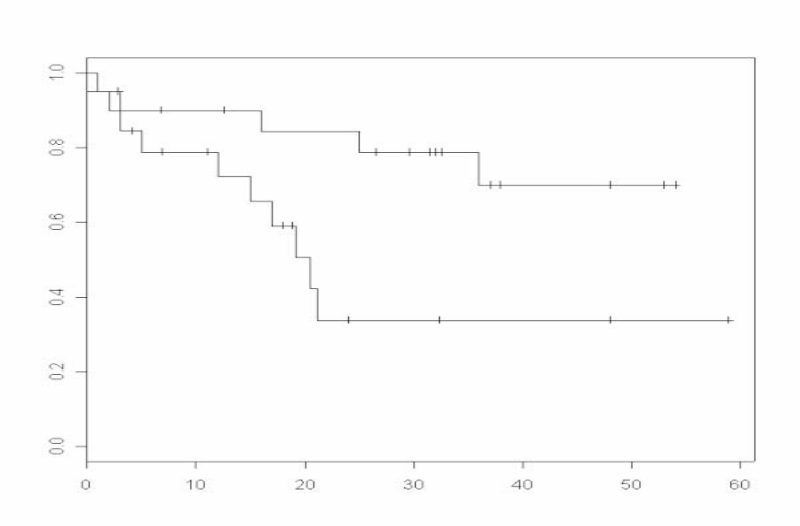



Figure 10A Expression Profile Identified from the Combination of Harvard and Michigan Datasets

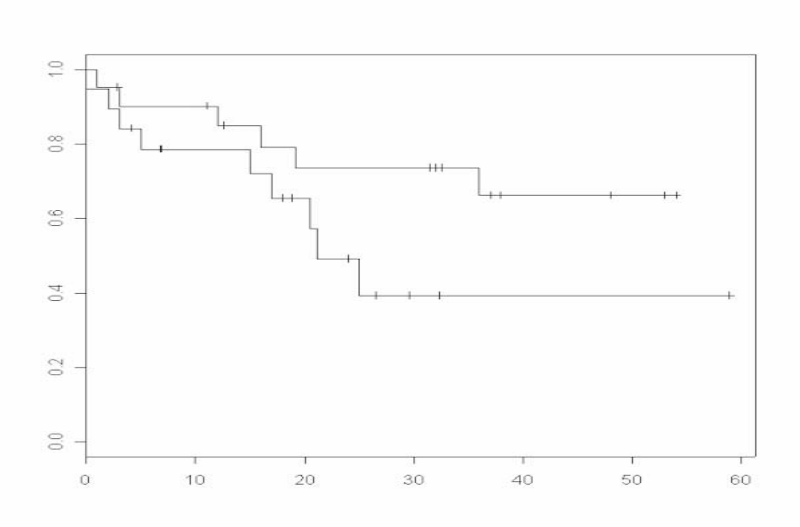



Figure 10B Expression Profile Identified from Michigan Lung Dataset Only

## Figure 10 Comparison of Expression Profiling Strategies Using Survival Curves

Expression profiles based on 1,000 re-samplings of lung cancer datasets were used to calculate SEP scores of all 40 patients in Stanford and Ontario testing lung datasets. Lung cancer patients were classified into prognosis groups with cutoff of SEP equal to 0. A Kaplan-Meier curve was built with each groups according to actual patient follow-up. The separation of two curves indicates the quality of an expression profile. (10A) Curves were generated from the expression profile identified from the combined dataset of Harvard and Michigan datasets. The separation of two curves is significant with p-value equal to 0.029. (10B) Curves were generated from the expression profile identified from Michigan lung cancer dataset only. The separation of the two curves is marginally significant with p-value equal to 0.12.

### 4.1.3 Sensitivity vs. Specificity of Reporter Gene Selection

Microarray data are featured by the large number of variables (genes) and much smaller number of observations (patients). When a hypothesis test is repeatedly applied to all the genes for their differential expression, some tests will get significant p-values just because of the random distribution of data. Consequently, identified reporter gene list will include false positives. Shortening the reporter list will reduce its sensitivity while shortening it will reduce its specificity. Therefore, the balance between specificity and sensitivity of reporter selection should be considered during expression profiling.

The necessity of achieving high sensitivity of reporter selection was questioned by previous results. As showed in Fig. 5 and 6, adding more reporters to profiles had little influence on bootstrapping results once the curves reached a plateau. Furthermore, the low consistence between reporter gene lists of two breast cancer datasets suggested that two disparate sets of reporters could perform very similarly on outcome prediction. This suggestion was advocated by a biological interpretation. Because of the regulatory interaction between genes, the expression of some genes is highly correlated with each other. If two genes are ideally co-expressed, they can replace each other in an expression profile. Including both of them in a profile has no impact on the profile except introducing redundancy. Therefore, it is not necessary to incorporate every true positive reporter to make a reliable profile. Conversely, low specificity of reporter selection can have considerably negative influence. Not only false positives will introduce extra variance into expression profiles, but also they will provide misleading information about functions of genes and their relationship to prognosis.

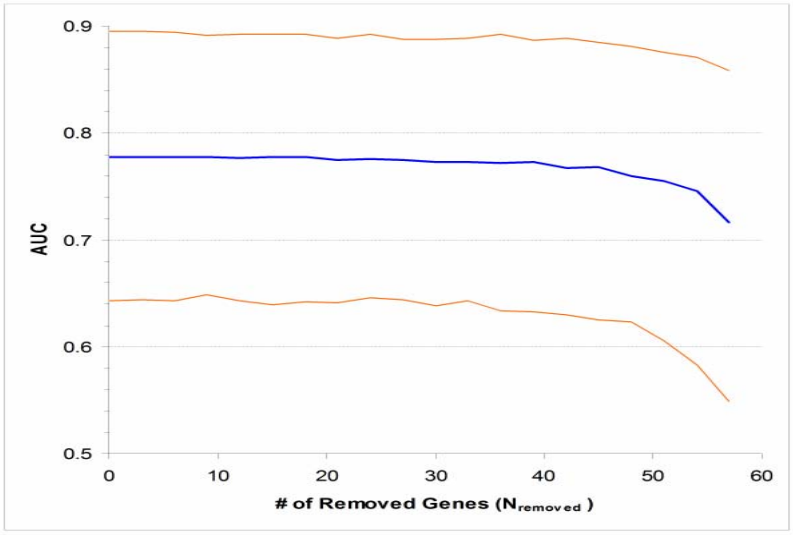



Figure 21A Consequence of Decreasing Selection Sensitivity via Reduction Procedure

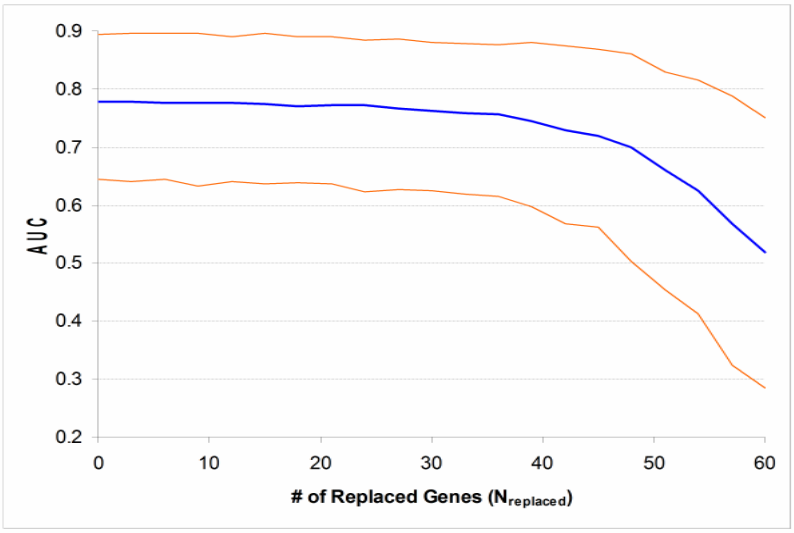



Figure 11B Consequence of Decreasing Selection Specificity via Replacement Procedure

## Figure 11 Changing of Expression Profile Quality with Sensitivity and Specificity

**Stepwise simulation procedures were carried out to trace how the decreasing of reporter selection sensitivity and specificity would change the quality of expression profiles. Results were collected from 1,000 re-sampling analysis of the combined dataset of two breast cancer datasets. The top-ranked 60 genes selected by each re-sampling were used for the following stepwise procedures. (11A) At each step, three reporter genes were randomly selected and removed, which artificially decreased the sensitivity of expression profile. The remaining reporters were used to calculate the SEP scores of testing patients, and the resultant scores were used to build ROC curves. The bootstrapping median and 90% CI of ROC curve area (AUC) are summarized in the figure. (11B) After the reduction procedure at each step, three replacing genes were randomly selected from all genes and added into the reporter list, which kept the length of list unchanged but artificially decreased its sensitivity. The replacing genes inherited the weight (z statistic of RST test) of the replaced genes. The revised list was used to calculate the SEP scores of testing patients, and the resultant scores were used to build ROC curves. The bootstrapping median and 90% CI of ROC curve area (AUC) are summarized in the figure.**

Two simulation strategies were designed to evaluate how the change of reporter selection sensitivity and specificity would change the quality of expression profiles. Both strategies were applied to the combination of two breast cancer dataset and their first step was to run another 1,000 bootstrapping re-samplings on this dataset. RST was performed on each gene and genes were ranked according to their Z statistics. 60 top-ranked genes were selected from each re-sampling as reporters.

The next step of reduction strategy gradually decreased the sensitivity of reporter list using a stepwise process. Three genes were randomly selected and removed from the list at each step, followed by re-calculating SEP of testing patients with the remaining genes. Fig. 11A presents the relationship between AUC of SEP scores and the sensitivity of reporter list. Generally, the median and 90% CI of AUC decreased when more genes were removed. However, the descending was slow most of the time. They were almost unchanged until about one-third genes were removed; decreased slightly after the removal of another one-third genes; and dropped down more obviously afterwards. The profiles of last three reporters had a median AUC equal to 0.716. These results suggested that the loss of sensitivity could be tolerated by expression profiles to an extensive level.

The replacement strategy, on the other hand, simulated the consequence of decreasing reporter selection specificity by substituting reporter genes with false positives. It had the same procedure as the reduction strategy except that in replacement strategy, the removed reporter genes were replaced by genes randomly selected from the whole datasets, keeping the size of expression profiles unchanged. The replacing genes inherited the weight of the replaced genes, so they could be considered as artificially introduced noises. The stepwise process continued until all 60 genes were replaced. Fig. 11B presented the relationship

between AUC of SEP scores and the specificity of reporter lists. Similar to Fig. 11A, all curves were gradually decreasing when more false positives were added. The median AUC was steady at the beginning; dropped by about 0.02 when half reporters were replaced; and fell rapidly afterwards. Furthermore, the range of 90% CI tended to be wider when more reporters were replaced. The median AUC was about 0.5 when the profiles contained only false positives.

Comparison of Fig. 11A and 10B indicated that at the same level of selection sensitivity, drop of specificity might have considerable negative impact. This conclusion advocated the superiority of multi-dataset expression profiling, because the combined dataset selected reporters more specifically than the individual ones (Fig. 7). In the current study, the sensitivity and specificity of reporter selection were arbitrarily balanced according to Fig. 5A and other results.

## 4.2 MAMA Project

### 4.2.1 Project Requirements and Use Cases

The general purpose of the MAMA project was to provide researchers with a data-mining platform for discovering gene expression profiles about cancer. To serve this purpose, three core components of MAMA system were developed: the centralized storage of microarray data in a relational database, the access to the database via a server program, and the data manipulation/analysis functions packed in a client-side program.

The requirement document of the MAMA project was given Appendix G. The schema design of the MAMA database had two considerations. On one side, the database was required to be MAGE-compliant, which accomplished the standardization of microarray data, but also increased the complexity of database schema. On the other side, the database was expected to enable quick access to its data by reducing the complexity of queries. The dilemma of these two requirements was solved by storing frequently requested data in denormalization tables. More details about MAMA database denormalization will be discussed in ‘Database Schema’ section. The requirements to the server program were straightforward. It should be able to connect to the database and handle concurrent requests from various clients. Requirements of the MAMA system related to data analysis were put on the client program. Besides regular data manipulation operations, this program was expected to maintain the data objects in a tree-like structure and provide the open-source APIs for developers to plug in data analysis methods. Unlike commercial microarray software, commonly available functions like clustering and graphics were not implemented. The client program was also required to have a Graphic User Interface (GUI), which should render data objects and system status in a consistent style.

Limited by the scale of the MAMA project, non-functional requirements was balanced to satisfy its essential utilities. The functionality and extensibility of the client program were given the priority. The targeted users of MAMA system are those researchers who are familiar with the basics of microarray analysis and want to apply more sophisticated or user-specific statistical methods. As a result, the MAMA client was required to be functionally extensible. Ease of use was also a major concern. User guides and FAQ were needed to provide end users sufficient guidance. By reading the documentation, an experience microarray researcher was expected to grasp the basic operations of the MAMA system within an hour. On several occasions, however, user-friendliness was traded to give end users more control over their data and analysis results. For example, end users have to make more effort to maintain data on their local disk since all data analysis functions were assigned to the client program. Security, reusability, and portability of MAMA were not explicitly required, although the client program was coded with Java, which made it portable to all computers that installed the Java Virtual Machine (JVM). Finally, performance was considered as an unstable fact of MAMA system. The complexity and the amount of data involved would decide the performance of operations. Furthermore, the performance of plug-in methods would be the responsibility of developers who added them into the application. A major bottleneck of performance was the operations performed on complete datasets, such as reading/writing them from/to files or retrieving them from database. The execution of these operations should be preserved as much as possible. Frequently used metadata about microarray experiments should be extracted and separately saved in advance, so it would not be necessary to dynamically summarize the metadata from all the source data when they were required.

**Table 9:**
**Example: 'Create Workspace' Use Case**

| **Use Case Name** | **Create Workspace** | |
|---|---|---|
| **Author** | **Zhe Zhang** | |
| **Date** | **2004/11/16** | |
| **Objective** | Create a new, empty workspace and open it. If there already is a workspace opened, close it | |
| **Actor** | User, System | |
| **Level** | Primary | |
| **Trigger** | User decides to create a new workspace | |
| **Included Use Case** | <<save workspace>> | |
| **Extended Use Case** | | |
| **Frequency** | Intermediate | |
| **Pre-condition** | • Client program is running | |
| **Post-condition** | • A new, empty workspace is created and opened in the client program | |
| **Main Flow** | **Actor Action** | **System Action** |
| | 1. User clicks 'Workspace' menu, then clicks 'New' menu item | |
| | | 2. System shows a dialog box asking for the name of the new workspace |
| | 3. User specifies the directory of the workspace and names it, then clicks 'Create' | |
| | | 4. System creates the new workspace object and opens it in the client program |
| | 5. User Clicks 'OK' | |
| | | 6. System terminates process |
| **Sub flows** | **Steps** | **Blanching Action** |
| | 4. There already is a currently opening workspace in the client program | 1. System prompts for what to do:<br>• Save<br>• Not save<br>• Cancel<br>2. User selects one<br>3. System responds to user's selection<br>• Save it, open the new one<br>• Not Save it, open the new one<br>• Abort creating, keep the old one<br>INCLUDE <<save workspace>> |
| | 3. User clicks 'Cancel' | System aborts process |
| **Exceptions** | **Conditions** | **Actions** |
| | 4. Redundant workspace name | System prompts for what to do:<br>• Overwrite (will replace the old one)<br>• Change name (will repeat step 3)<br>• Cancel (will abort process) |

Use cases of the MAMA client program were broken into multiple levels, from general categories to step-by-step description, following the design patterns of UML (Unified Modeling Language) [123, 124]. The major packages of use cases were diagramed in Fig. 12A. It showed that most use cases were about management, presentation, or operation of data objects. Use cases of 'Manage Workspace' package and their relationship were summarized in Fig. 12B, and the flowchart of an example use case, 'Create Workspace', was given in Fig. 12C as an UML activity diagram. Furthermore, the details about 'Create Workspace' use case were described in Table 9. (See Appendix H for more use cases.)

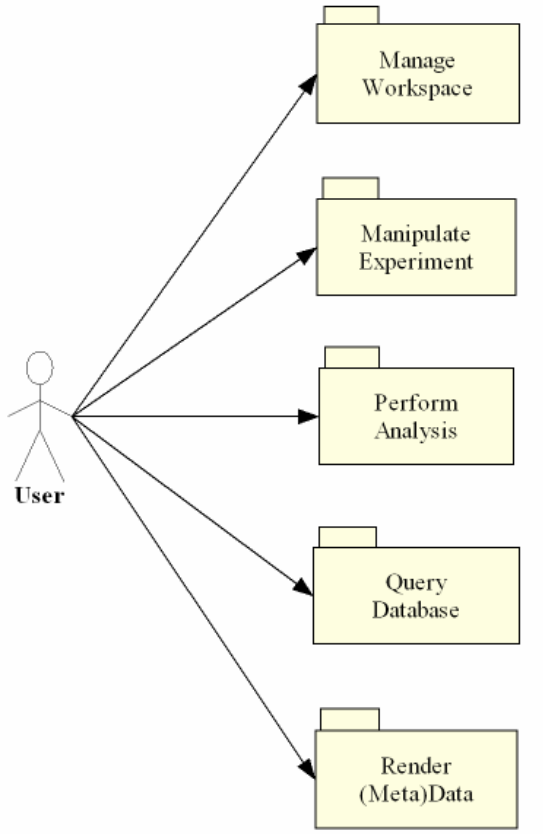



Figure 12A All Use Case Packages

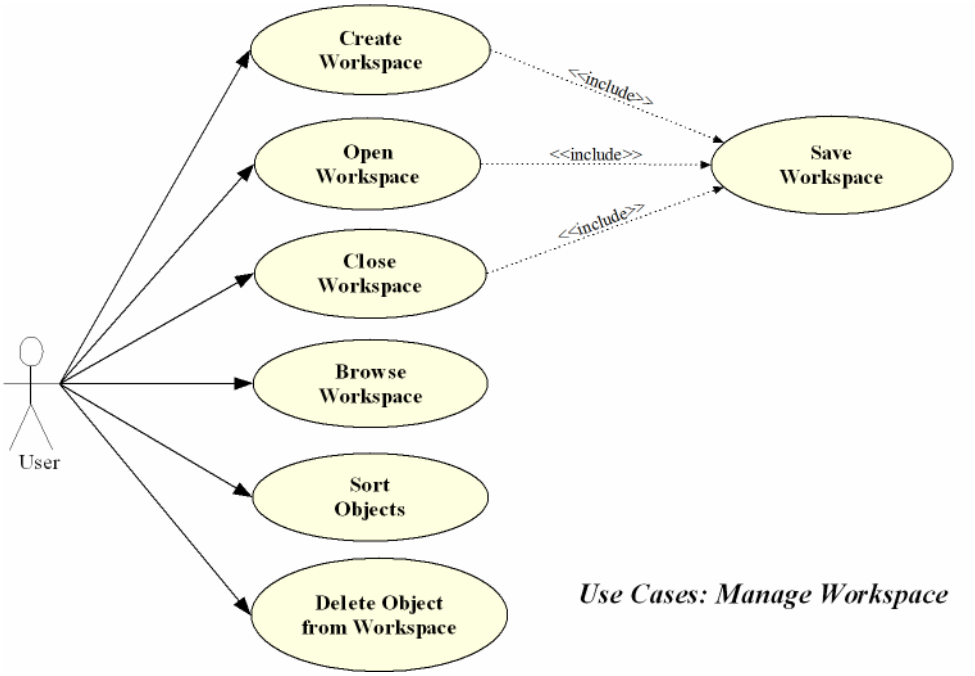



Figure 12B All Use Cases of 'Manage Workspace' Package

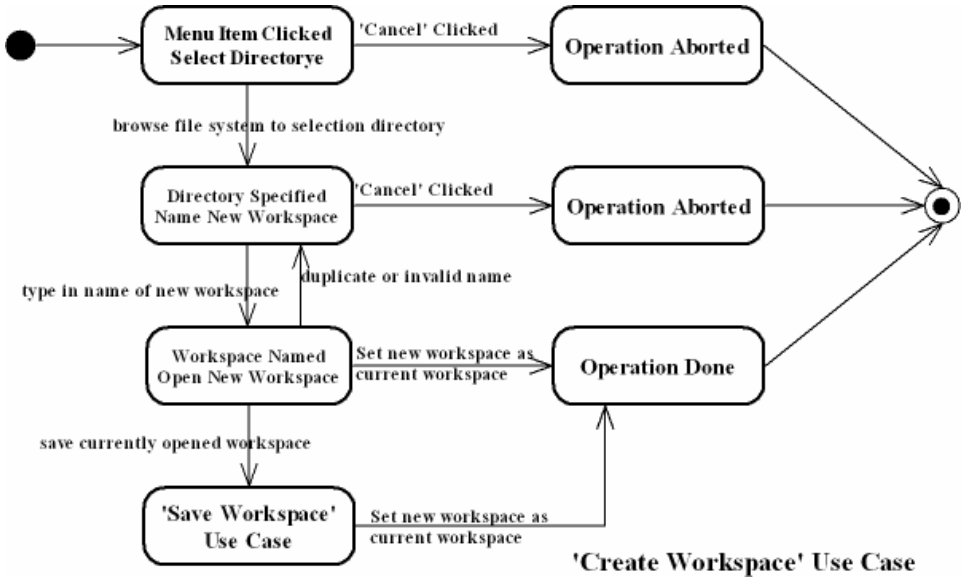



Figure 12C Event Flow Diagram of 'Create Workspace' Use Case

## Figure 12 Examples: Use Cases

The functions of MAMA system were described as a number of use cases. All use cases were categorized into several packages (12A), and use cases might be related to each other (12B). The events happened during each use case could be represented in flow diagrams (12C).

### 4.2.2 Design of MAMA System

Besides satisfying the requirements mentioned above, the basic objectives of designing MAMA include:

- Minimize the requirement on hardware and maintenance resources.
- Encapsulate software components.
- Follow existing standards related to microarray as much as possible.
- Control the scale of this project at a feasible level.

#### 4.2.2.1 Software Development Environment

The following software and hardware were used for the development of MAMA system. This list can also be considered as the recommended system requirements of installing and running MAMA database or programs.

Software:

- Server computer
    - Operation system: Solaris 9 (Sun Microsystems Inc.).
    - Database management system: Oracle Release 9.0.1 (Oracle Corporation).
    - Database access API: Java JDBC Package (Sun Microsystems Inc.).
    - Web server and servlet engine: Apache Tomcat Version 4.1 (Apache Software Foundation).
    - Server program API: J2EE Servlet Specification 2.3 (Sun Microsystems Inc.).
- Personal computer
    - Operation system: Microsoft Windows XP Professional Version 2002 (Microsoft Corporation).

- Programming language: Java2 SDK, Standard Edition Version 1.4.2 (Sun Microsystems Inc.).
- Source code and project management: Eclipse Platform Version 3.0.1 (Eclipse contributors and others).
- Statistical functions: Common-Math Library Release 1.0, Jakarta Commons Project (The Apache Software Foundation).
- Java/XML mapping: Castor XML Version 0.9.6 (Exolab Group, Intalio Inc., and Contributors).
- UML diagrams: SmartDraw Version 7.01 (Hemera Technologies Inc.).
- Local database management: Microsoft Access 2000 (Microsoft Corporation.) --- The local installation of the MAMA database was used just to simplify the coding/testing efforts during the developmental stages. To access this database, a local version of the server program and its Tomcat container were also installed.

Hardware:

- Server computer
  - Model: Sun Fire 280R (Sun Microsystems Inc.).
  - CPU: 2 X 1200 MHz UltraSparc-III+.
  - Memory: 2.0 GB of RAM.
  - Disk space: 10 GB assigned to the MAMA database.
- Personal computer
  - Model: Inspiron 5100 Notebook (Dell Inc.).
  - CPU: Pentium(R) 4 CPU 2.66GHz.

- Memory: 1.0 GB of RAM.
- Disk space: 15GB reserved for the development of MAMA project.

#### 4.2.2.2 System Architecture

The system architecture of MAMA system was straightforward as shown in Fig. 13. The database maintained the permanent storage of microarray datasets collected and loaded by its administrators or curators. Any end users would be able to query this database. The server program is a Java Servlet managed by an Apache Tomcat container. This server can simultaneously handle multiple requests from different clients and interact with the database to query the data. The data retrieved from the MAMA database is wrapped into Java serializable objects by the server before they are sent to the clients. Therefore, all database operations are encapsulated in the server program and the client program does not need to directly interact with the database. The server and client programs communicate with each other using a pre-defined protocol, which codes the status of requests or responses. The complete protocol is listed in Appendix I. As long as the protocol is unchanged, the server or client program can be separately updated without notifying the other. Furthermore, other developers can add their own programs to communicate with the MAMA server or client via this protocol. The MAMA client program was also developed as a stand-alone application for microarray analysis. It can be run on the local computer without connecting to the server. Microarray datasets are imported into the client program directly from text files, annotated with standard vocabularies, and saved on local disk in pre-defined formats for users to load later. The client program saves all local files as XML documents except those containing expression data matrixes. Because of the large amount and simple structure of those matrixes,

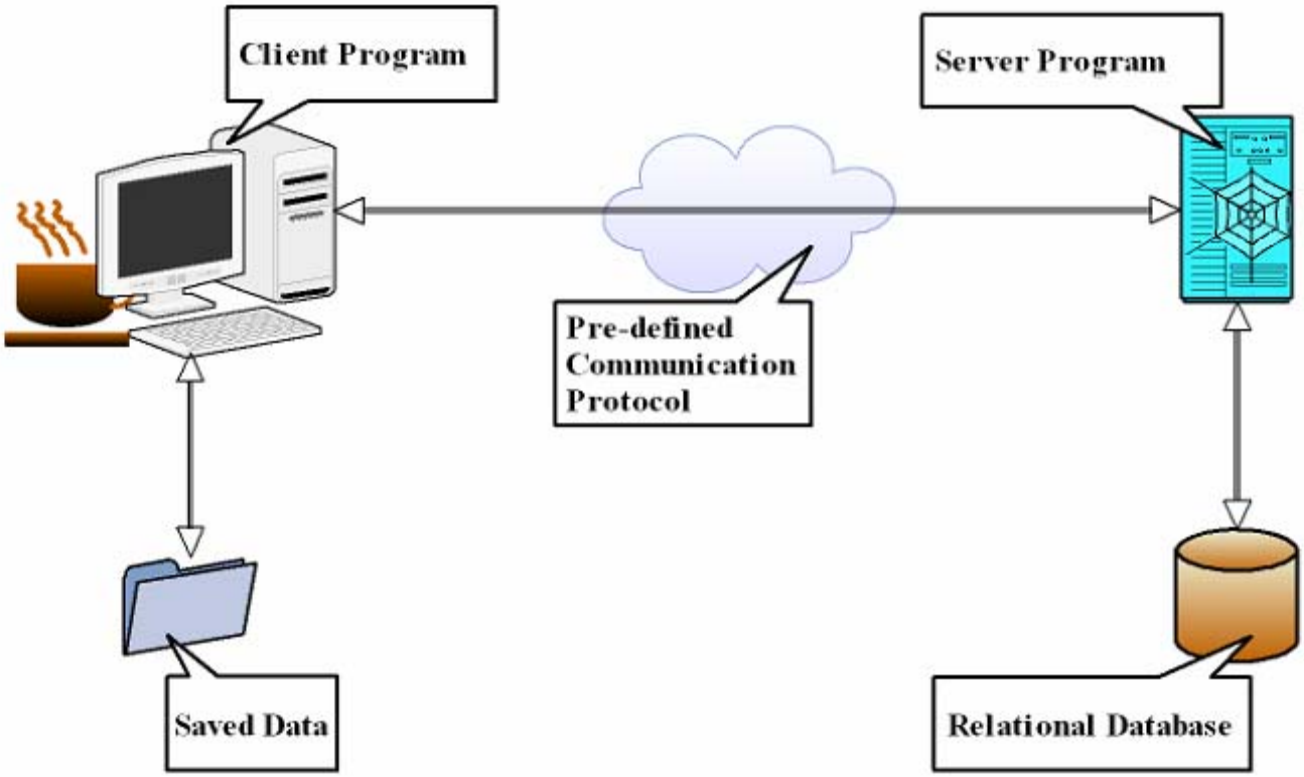



**Figure 13 System Architecture**

The MAMA system has a client-server architecture. The client and server programs communicate over the web using a pre-defined a protocol. These two programs are independent of each other, so the updating of one side has no effect on the other side. The server program is a Java Servlet that can concurrently handle requests from multiple clients. Client requests are translated into SQL queries that will be sent to an Oracle relational database at the backend. The database maintains the permanent storage of collected microarray datasets. All database operations are encapsulated in the server-side, which will wrap query results into Java serializable objects before sending them back to the clients. The client can work as a stand-alone application, bit it needs to communicate with the server to fetch data from the database.

they are saved as tab-delimited text files with sample identifiers as column names and gene identifiers as row names. XML documents and java objects are mapped to each other using functions provided by Castor XML API. The mapping rules are specified in file 'mapping.MAMA', which is wrapped together with the client program.

The system architecture of MAMA suggested a 'light server' and a 'heavy client'. It simplified the development and maintenance of the server. Consequently, end users should rely on the computational power of their local computers for time-consuming operations. They also need to manage their data and analysis results on their local disk, which might be preferred by sophisticated users. Another critical feature of this architecture was the encapsulation of functions, which makes the MAMA system more modifiable and extensible.

#### 4.2.2.3 Database Schema

The schema of the MAMA database had two levels. The first level copied the schema of ArrayExpress [87], a public microarray database developed by EBI (European Bioinformatics Institute). This schema was directly derived from MAGE-OM, which defines concepts related to most aspects of microarray experiments. MAGE-OM was designed as a complex data model using object-oriented mode. Consequently, the database schema of ArrayExpress would make the SQL queries intricate and inefficient. Many queries need to join a number of large tables and sometimes, the response time would be practically unacceptable. For example, retrieving the identifiers of sequences in an array design involved the join of at least six database tables, most of which might contain millions of records. To improve the performance of the MAMA database, a second level of database schema was added, which included a group of denormalization tables. The purpose of these

tables was to redundantly store frequently requested data. Because of the introduction of redundancy, maintaining database integrity would be a heavy burden and error-prone when data were inserted, updated, or deleted. The MAMA database, however, would barely be harmed by this drawback of denormalization since it functions similar to a data warehouse. Insertions and deletions will be rare operations in this database, and updating will be even rarer because the existing data are the results of previous microarray experiments. Therefore, the two levels of the MAMA database serve different purposes. MAGE-OM tables permanently would store microarray datasets in a standard schema and allow other MAGE-compliant systems to reuse them, while the denormalization tables benefits the data analysis applications by improving the accessibility of the database. To further reduce the response time of queries, most columns of denormalization tables are indexed.

Database tables used by the current version of MAMA and their relationship are presented in Appendix J. Fig. 14 demonstrates a group of tables involved in the key entities of microarray analysis. The table located on up-left corner was a denormalization table storing the gene expression measurements. Each row of this table corresponds to a processed measurement, which is 2-dimensionally labeled with biomaterial (sample) and design element (sequence) identifiers. According to this diagram, each experiment can have only one array design, so the original experiment using multiple array designs should be split in advance. This diagram also contains a special table: ‘T_DATA_SUMMARY’. This table stores some common statistics about the expression level of a specific gene in a specific experiment. Since ‘experiment’ is the basic unit of data analysis in MAMA, these statistics are frequently queried metadata that should be conveniently acquired.

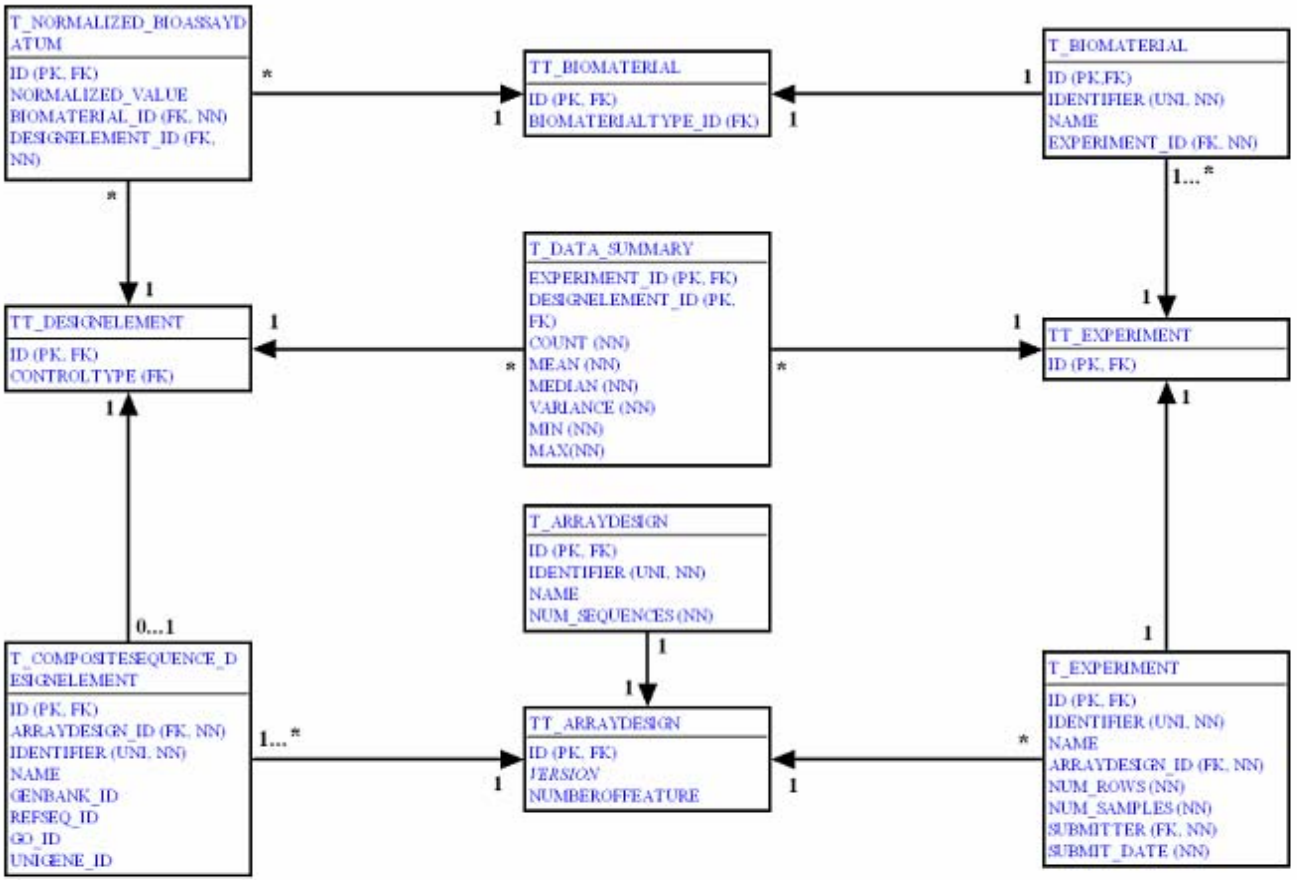



**Figure 14 A Fraction of Database Schema**

This small segment of the MAMA database schema includes several core tables. The name of ArrayExpress tables starts with 'TT_', while the name of denormalization tables starts with 'T_'. Gene expression measurements will be re-processed by pre-defined criteria before they are stored in table 'T_PROCESSED_DATUM'. It represents the expression of each gene (design element) in each sample (biomaterial) with a single, summary data point. 'T_DATA_SUMMARY' holds the frequently queried descriptive statistics of expression measurement for each experiment. Table 'T_BIOMATERIAL' makes a shortcut mapping between samples and experiments, and includes redundant fields for fast access. The complete database schema is given in Appendix J.

#### 4.2.2.4 Data Flow

The MAMA system provides two types of data storage: a remote database for publicly available long-term storage and a local file system for temporary, user-specific and analysis-oriented storage. Fig. 15 demonstrates how data would be transferred between various locations of the MAMA system, which is directed by a group of software packages. The following sequentially described the data flow steps in a typical data analysis procedure:

1. Source microarray dataset is imported in the format of tab-delimited text. The MAMA client program provides a GUI wizard to guide users through the process. The wizard prompts for the description and metadata of the dataset. The data submitter can choose to send the dataset to one of the two following locations:
   1A. The dataset is sent to the MAMA database for long-term storage through the server program. After receiving the submitted dataset, the server program pre-processes its expression data before loading it into the database. The guidelines of data pre-processing are given in Appendix K. Loading datasets into database is a complicated and time-consuming operation and only the administrators or curators of the database have the authority to execute it. Ordinary users need to contact these people if they want their data to be stored in the database.
   1B. The dataset is parsed into an 'Experiment' object and added into the currently opened workspace of the client program. The new 'Experiment' object is saved as files in the local file system. This operation is simpler and faster comparing to 1A. It may be preferred by users who do not want to expose their data to the public.
2. If a needed dataset already exists in the database, the client program fetches it from the database by sending a request to the MAMA server. Retrieved dataset are parsed

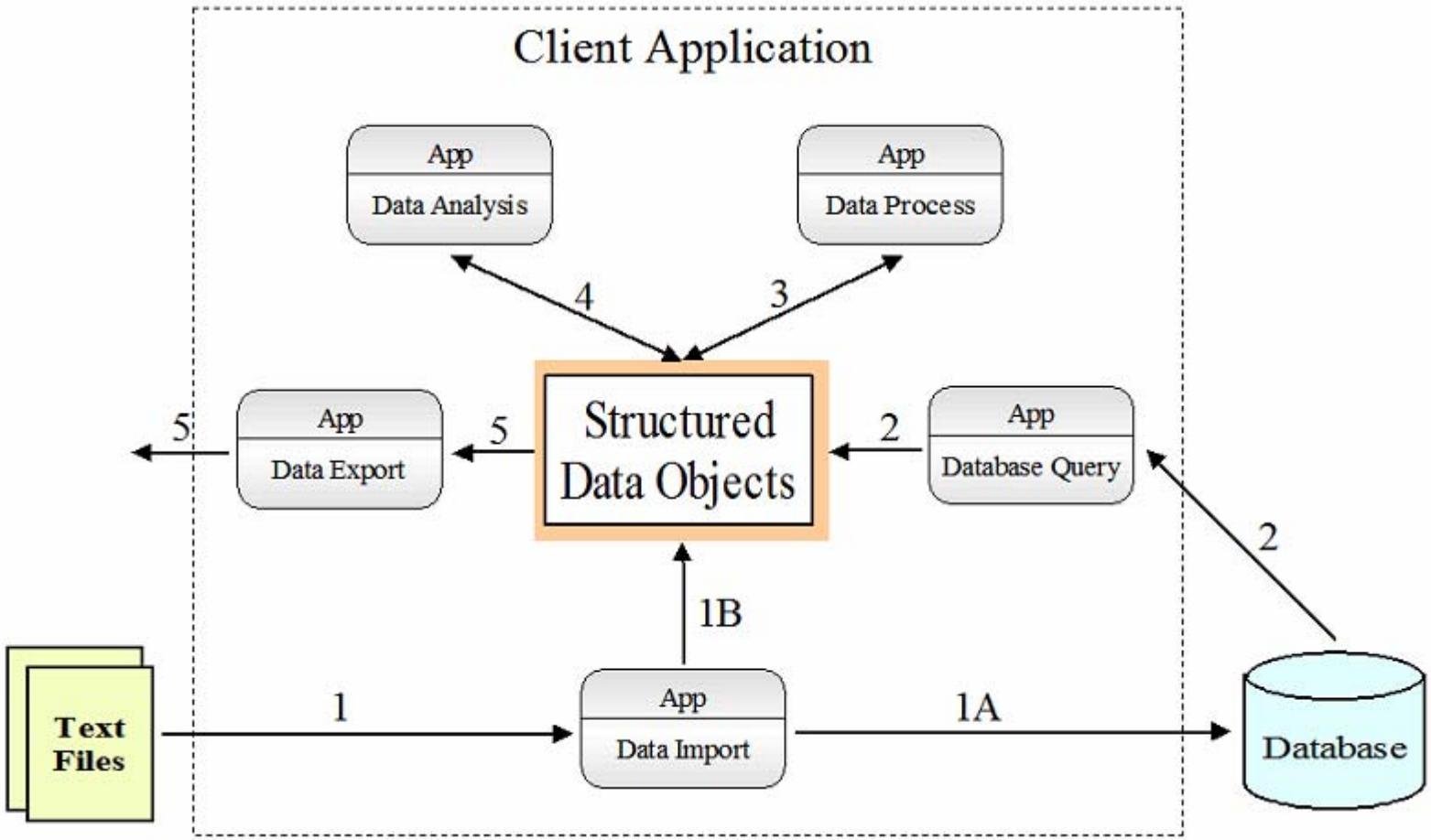



**Figure 15 The Data Flow within MAMA System**

The imported data can be stored remotely in database or locally as files, but data process and analysis functions only use local data. (1) Dataset was imported into the MAMA system from tab-delimited text files via data importing Wizard. User chose to send the dataset to the database (1A) or save it locally (1B). (2) The client program fetched an existing dataset from the database by sending a request to the server program. (3) User customized the dataset or created virtual experiment for the following data analysis. (4) Data analysis was executed on the selected dataset(s). (5) Analysis results or other data objects were exported from the MAMA system.

into an ‘Experiment’ object, added into the current workspace, and saved locally. Instead of fetching complete datasets, users can also query the database about the existing data, such as the sequences in an array design or the samples in a dataset.

3. Before an analysis is performed on experiments in the current workspace, users might need to customize the contents or re-process the expression data of those experiments. For example, if users were only interested in analyzing genes whose expression varied radically in samples, a filtering operation would be carried out to remove genes having low-variance.
4. The client program defined an ‘Analysis’ object and executed it using selected dataset(s). After the analysis was finished, its results are saved in local file system too.
5. Saved data objects, such as analysis results, can be exported from the client program as text files.

#### 4.2.2.5 Software Architecture

Source codes of both the client and server programs were written with Java2 SDK Standard Edition and managed within Eclipse platform. The fundamental software design principle in the MAMA project was to achieve the encapsulation of functions. Fig. 16A demonstrates the architecture of the client program, which mostly follows the MVC (Model-View-Controller) software design pattern. The ‘Model’ package defines Java classes corresponding to various data objects, such as nucleotide sequences and biological samples. At runtime, the client program maintains all data objects in a tree-like structure within a ‘Workspace’. The data objects kept by the client program are presented on the user interface through the ‘View’ package, which defines various graphic components, such as dialog

boxes and menus. Graphical components are used to render data objects (lists, tables, and so on), or listen to user actions (buttons, menu items, and so on) initiated by keyboard input or mouse click. User actions are passed to and handled accordingly by processes implemented in the 'Controller' package. The handling of actions might cause a change of system status. Consequently, the controllers notify the 'Model' and/or 'View' to update their contents. The controllers have the key position in this architecture. When necessary, they execute the methods implemented in the 'Data Analysis' package to carry out statistical analyses, or interact with the 'Communication' package to establish connection with the server program.

The server program has a simpler structure with four layers: servlet, listener, handler and database facade. The whole server program is designed as a Java servlet, which is activated the first time it is requested by a client program and keeps running until it is explicitly shut down. The activated MAMA servlet creates a listener that listens to requests sent by clients at a network port. The listener responds to each request by generating a handler, so concurrent requests are handled simultaneously. Each handler interacts with a single client by exchanging data objects. Since the current version of MAMA server only handles database query requests, the incoming data objects are passed to the database facade layer. This layer parses the requests into database queries and sends the queries to the MAMA database.

The details about client-server communication in MAMA system are illustrated in Fig. 16B. The following gave the step-by-step description of this procedure:

1. When a 'Controller' needs to access the MAMA database, it defines a Java 'Query' object and specifies the query attributes in this object. The controller sends this 'Query' to a 'Communication Facade' and specifies the action, such as 'insert' or 'select'.

2. The ‘Communication Facade’ wraps the ‘Query’ and the action into a ‘Request’ object and assigns a key to the query. Each facade maintains a ‘Requester’, which might contain multiple ‘Request’ objects. Once all ‘Request’ objects are added into the ‘Requester’, the facade passes the ‘Requester’ to a ‘Communicator’, which handles the network communication with the server.
3. The ‘Communicator’ contacts the ‘Servlet’ in order to set up a network connection. If the ‘Servlet’ has not been activated, it is loaded and creates a ‘Listener’ to listen for requests from the clients.
4. The request for connection from the ‘Communicator’ is caught by the ‘Listener’. The ‘Listener’ generates a thread as a ‘Handler’ to handle the incoming request.
5. A network socket is established between the ‘Communicator’ and the ‘Handler’ for exchanging data. The ‘Requester’ is sent to the server through this socket. The ‘Communicator’ waits for response from the server.
6. The ‘Handler’ passes the received ‘Requester’ to a ‘Database Facade’ and waits for results.
7. The incoming ‘Requester’ is unwrapped by the ‘Database Facade’ and the ‘Request’ objects inside it are collected. Each ‘Request’ is parsed into an SQL query based on its attributes. The query is sent to the database.
8. The database returns query results to the ‘Database Facade’.
9. The ‘Database Facade’ uses the returned results to generate a ‘Response’ object and assigns it a key equal to the key of the corresponding ‘Request’ object. A response code and a text message are also added to each ‘Response’ to indicate the consequence of executing the request. The interpretation of response codes are given

in Appendix I. After all ‘Response’ objects are collected, the ‘Database Facade’ wraps them into a ‘Responser’ and a status code is added to the ‘Responser’ to indicate the overall consequence of executing the requests. The interpretation of status codes are also given in Appendix I. Finally, the ‘Database Facade’ passes the ‘Transfer’ to the ‘Handler’.

10. The ‘Handler’ sends the ‘Responser’ including all the responses to the ‘Communicator’, and terminates itself.
11. The ‘Communicator’ reads data returned from the server, closed the network socket, and sends the ‘Responser’ object to the ‘Communication Facade’.
12. The ‘Communication Facade’ retrieves the status code from the ‘Responser’ and reports it on the user interface if there is any error. The ‘Responser’ is then unwrapped to create a list including all the ‘Response’ objects. The response code and text message of ‘Response’ objects are reported on the user interface if there is any error. The ‘Communication Facade’ informs the ‘Controller’ that the query results are ready. Finally, the ‘Controller’ retrieves each ‘Response’ by providing its key and uses it to create a ‘Query Result’ object corresponding to the ‘Query’ object.

The client-server communication architecture presented above gives an example of how various software modules were encapsulated in the MAMA system. The same principle was applied to other operations implemented in the client program.

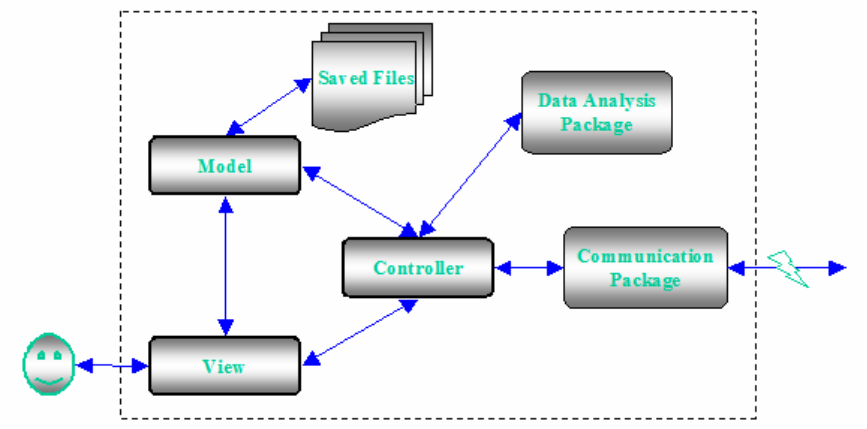



Figure 16A Software Architecture of the MAMA Client Program

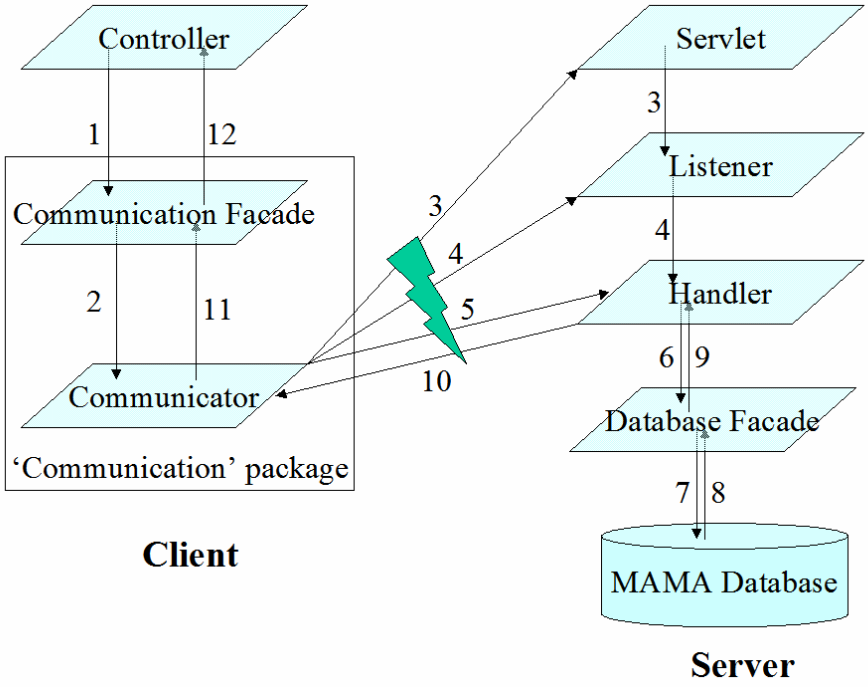



Figure 16B Layered Software Structure of Client-server Communication in MAMA System

**Figure 16 Software Architecture**

The primary principal of software design in MAMA project was the encapsulation of functions. Therefore, a software package was designed for each basic function. (16A) The software architecture of the client program followed the MVC (Model-View-Controller) design pattern. The 'Model' package defined Java classes corresponding to various data objects, the 'View' package renders data objects on user interface, and the 'Controller' package handles user actions. The 'Controller' was the key component in this architecture. It can interact with the 'Data Analysis' packages to execute data analysis methods or with the 'Communication' package to establish connection with the server program. (16B) The client-server communication had multiple layers. Each of these layers encapsulated certain functions. For example, the 'Database Facade' layer is in charge of translating user requests to database queries and wrapping query results into data objects.

#### 4.2.2.6 Graphical User Interface

The MAMA client program has a graphical user interface (GUI). The main window includes a split panel with two panes and a group of pull-down menus (Fig. 17A). The left pane renders the hierarchical structure of data objects in two trees. The upper tree is called 'Database Snapshot', representing the status of MAMA database at the time when the snapshot was taken. The summary about the status of the database has three parts: existing array designs, experiments, and sample traits. The sample traits are a set of controlled vocabularies used to describe samples. The lower tree renders the contents of the currently opened workspace. Each workspace contains three types of objects: 'Query', 'Experiment', and 'Analysis', in any numbers. The details of a selected object are presented in the right pane, usually by table(s). For example, the right pane of Fig. 17A lists all samples of experiment 'E-MICH-01' with available clinical data of patients.

Dialog boxes are extensively used for users to specify inputs or parameters of operations. Fig. 17B shows a dialog designed for inserting a new sample trait (ontology entry) into the database. In this dialog, an ontology entry called 'Age' is submitted by this dialog. According to the inputs, this entry is defined by MGED Ontology, which also assigned it an accession number and a URI link to it. Furthermore, 'Age' has 'year' as its unit. When an operation has too many inputs or parameters to be included in one dialog box, it is split into several steps sequentially organized by a wizard. Fig. 17C shows one of four steps of a wizard used to import a microarray dataset from text files. The name and location of the imported files are specified with this dialog. As shown in Fig. 17B and 16C, the design of GUI components has consistent style.

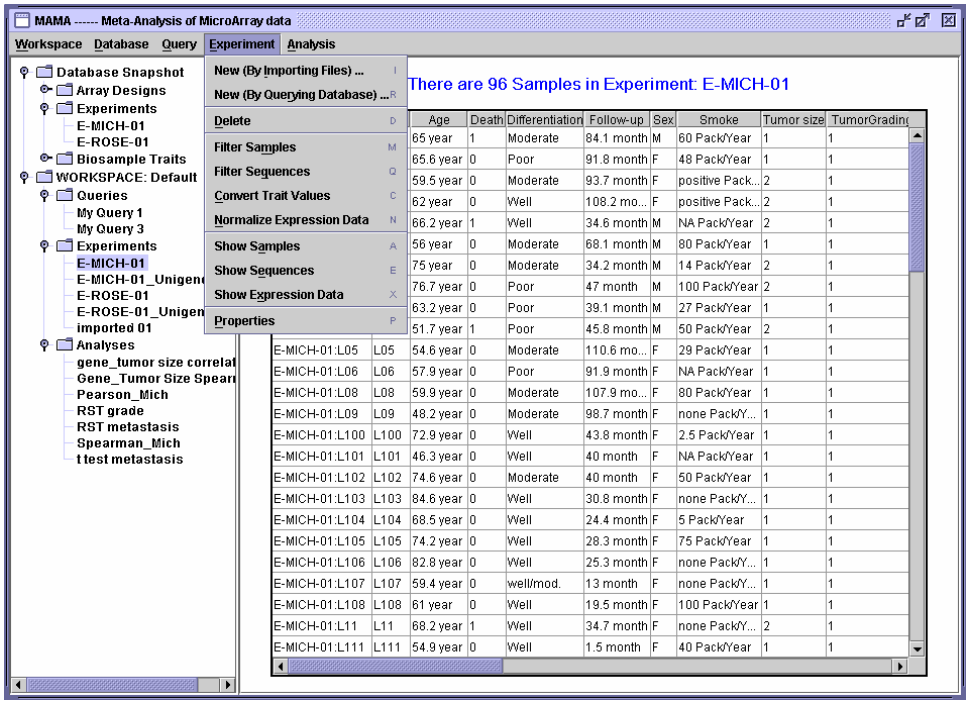



Figure 17A Main Window of the MAMA Client Program

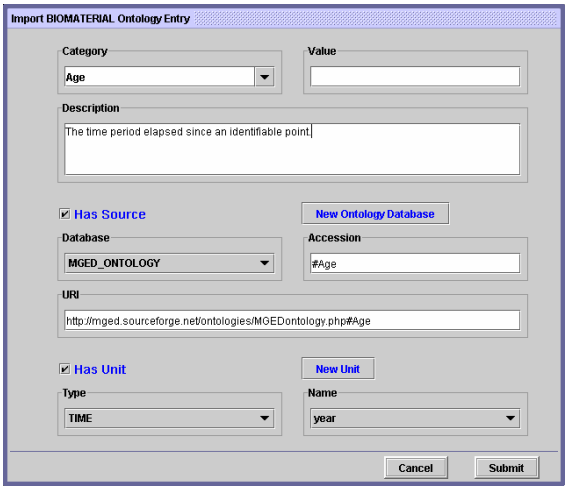



Figure 17B User Interface Example: Import New Entry of Biomaterial Ontology

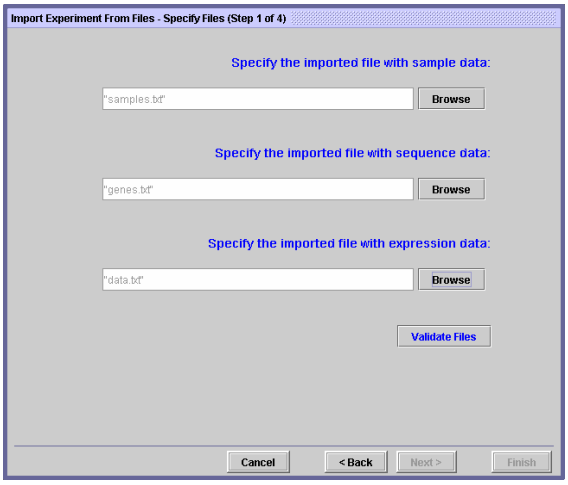



Figure 17C User Interface Example: Import Microarray Dataset from Text Files

## Figure 17 Graphic User Interface of the Client Program

Graphic user interface was coded with Java Swing API. Dialog boxes were extensively used in a consistent style for users to specify the inputs of an operation. (17A) The main window of the client program has two panes. The left pane allows users to browser the data objects in the current workspace, and the right pane renders the details of a selected data objects, such as all samples in a virtual experiment. (17B) This dialog defines an entry of biomaterial ontology to be submitted into the database. (17C) This dialog is part of the wizard that imports a new experiment into MAMA system. Users can specify the imported files of samples, genes, and expression measurements within this box.

### 4.2.3 Data Annotation

The results of microarray analysis are meaningful only when the sequences and samples in a dataset are properly annotated. Different microarray studies have been using various naming systems or terminologies to annotate their datasets. Since the MAMA system was design to allow for simultaneously analyzing multiple datasets, it is critical to consistently annotate different datasets with standard and/or common systems. Consequently, the importing and the processing of microarray datasets often involve the mapping of annotation systems.

The annotation of biological samples is more complicate. Researchers often use different terms to refer to the same concept, such as ‘tumor size’ and ‘tumor diameter’. Therefore, sample traits should be described with ontology or controlled vocabularies in the MAMA system to achieve cross-dataset analysis. The mapping between sample annotations was more difficult and error-prone. Data submitters should fully understand the definition of standardized terms before using them to describe samples. Currently, the controlled vocabularies used in MAMA have four sources as listed in Table 10A. MGED Ontology defines concepts and terms closely related to microarray experiments, such as ‘ArrayDesign’ and ‘SurfaceType’. Although MGED Ontology made a major effort on the description of biological samples (‘BioMaterial’), it only covered some general concepts, such as ‘Age’ and ‘Sex’. While the terms provided by MGED Ontology had the priority, NCI Thesaurus and Metathesaurus were used as its supplements for cancer-specific traits and trait values, such as ‘Angioinvasion’ and ‘Metastasis’. Comparatively, NCI Thesaurus was smaller and more stable and Metathesaurus was larger and more frequently updated. A number of ontology categories have been imported into the current version of MAMA. A list of these categories

and their definition can be obtained by taking a database snapshot and viewed by opening the 'Sample Traits' folder in the left pane of Fig. 17A. Users are allowed to extend the pool of controlled vocabularies by submitting new ontology databases or entries to MAMA system.

The sequences of microarray datasets are annotated with the identifiers assigned by various sequence databases. A large number of sequence databases about nucleotide acids (genes), gene products (proteins), and gene functions had been used to provide systematic annotations. Table 10B lists the sequences databases already registered in the MAMA database. Microarray sequences labeled with the identifiers of these databases is acceptable. The list can be further extended if necessary. In order to simplify the related operations, the client program recognizes and processes sequence annotations provided by four systems: GenBank, RefSeq, Unigene, and GO (Gene Ontology), which are highlighted in Fig. 11B. For example, to search a sequence in the database or the current workspace, only sequence identifiers provided by these systems can be used to specify the searched sequence. When a new 'Experiment' object was created in the client program by downloading from the database or directly importing from text files, other annotation types are not be accepted. Therefore, before a source dataset is loaded into the MAMA system, the annotation of its sequences should be mapped to identifiers of those four systems.

**Table 10:**
**Data Annotation Resources**

Table 10A Biological Sample Annotation Resources

| Name | Provider | URI |
|---|---|---|
| MGED Ontology | MGED | http://mged.sourceforge.net/ontologies/MGEDontology.php |
| NCBI Taxonomy | NCBI | http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html |
| NCI Metathesaurus | NCI | http://ncimeta.nci.nih.gov/indexMetaphrase.html |
| NCI Thesaurus | NCI | http://nciterms.nci.nih.gov/NCIBrowser/Dictionary.do |

Table 10B Nucleotide Sequence Annotation Resources

| Name | Provider | URI |
|---|---|---|
| Affymetrix Probe Set | Affymetrix | http://www.affymetrix.com |
| Blocks Database | FHCRC | http://blocks.fhcrc.org |
| Enzyme Nomenclature | IUBMB | http://www.chem.qmw.ac.uk/iubmb/enzyme |
| EMBL | EBI | http://www.ebi.ac.uk/embl |
| Ensembl | ENSEMBL | http://www.ensembl.org |
| Entrez Gene | NCBI | http://www.ncbi.nih.gov/entrez/query.fcgi?db=gene |
| Entrez Protein | NCBI | http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Protein |
| **GenBank** | **NCBI** | **http://www.ncbi.nlm.nih.gov/Genbank/** |
| **GO** | **GO Consortium** | **http://www.geneontology.org/index.shtml** |
| GPCRDB | GPCR | http://www.gpcr.org/7tm |
| InterPro | EBI | http://www.ebi.ac.uk/interpro |
| KEGG | Kanehisa Lab. | http://www.genome.jp/kegg |
| LocusLink | NCBI | http://www.ncbi.nlm.nih.gov/LocusLink |
| MGD | Jackson Lab. | http://www.informatics.jax.org/ |
| NetAffx | Affymetrix | http://www.affymetrix.com/ |
| OMIM | NCBI | http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=OMIM |
| Pfam | SANGER | http://www.sanger.ac.uk/Software/Pfam |
| Protein Kinase Classification | PKR | http://pkr.sdsc.edu/html/pk_classification/pk_catalytic/pk_hanks_class.html |
| **RefSeq** | **NCBI** | **http://www.ncbi.nlm.nih.gov/RefSeq** |
| SCOP | UC, Berkeley | http://scop.berkeley.edu |
| SPTR Database | MRC | http://www.hgmp.mrc.ac.uk/Bioinformatics/Databases/sptr-help.html |
| EGAD | TIGR | http://www.tigr.org/tdb/egad/egad.shtml |
| **Unigene** | **NCBI** | **http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=unigene** |

### 4.2.4 Working with Data Objects

Since data objects are structured hierarchically by the client program, all user data can be saved on local disk as a single XML document for the ease of maintenance. However, reading, writing, and parsing such XML documents are very inefficient because of the large size of microarray datasets. A tradeoff was made to improve the performance of data

reading/writing at the cost of more software development and data maintenance efforts. It was decided that user data would be spilt into multiple files having proper size. The relationship between saved data objects is implied by the location and name of the files. Any file writing operation retains the consistency of data in related files. The client program uses Castor XML package to accomplish the mapping between Java data objects and XML elements. The mapping information is provided in an XML document named 'mapping.MAMA', which was wrapped into the client program. (See Appendix L for details about XML mapping.)

#### 4.2.4.1 Workspace

'Workspace' is the top-level data object and has a unique name (identifier). Users can create multiple Workspaces on local disk, but each running client program will open only one Workspace at any time. All data manipulation and analysis operations must be carried out within an opened Workspace. The 'Default' Workspace is opened automatically when the client program is activated. Contents of each Workspace are saved in an XML document and a directory of files, both named after its identifier. The XML document stores the metadata about a Workspace and the data objects inside it, while the contents of the data objects are stored in the directory as individual files. To simplify the maintenance of data, users are recommended to create a new Workspace for each data analysis project.

The following illustrated an XML document of the metadata about 'Default' Workspace:

```
<?xml version="1.0" encoding="UTF-8" ?>
<workspace name="Default" created="2005-09-30T19:19:28.603-04:00" last-
modified="2005-12-01T18:49:52.447-05:00">
    <query identifier="BCL seq" subtype="Sequence" created="2005-10-
    25T01:12:58.450-04:00" last-run="2005-10-25T01:13:00.132-04:00">
      <inner-join-limit operator="LIKE" value="bcl" field="NAME" />
```

```
    <outer-join-limit operator="=" value="E-MICH-01" field="Experiment
    Identifier" />
    <outer-join-limit operator="=" value="E-ROSE-01" field="Experiment
    Identifier" />
  </query>
  <query identifier="My Query 1" subtype="Sample" created="2005-07-
  21T16:56:28.877-04:00" last-run="2005-07-22T00:05:48.450-04:00">
    <inner-join-limit operator=">=" value="50" field="Age" />
    <inner-join-limit operator="=" value="Cancer" field="DiseaseState" />
    <outer-join-limit operator="=" value="Breast" field="Organism Part" />
    <outer-join-limit operator="=" value="Lung" field="Organism Part" />
  </query>
  ......
  <experiment identifier="E-MICH-01" name="Michigan Lung Cancer -
  Adenocarcinoma" description="Complete dataset of Michgan Lung cancer
  study." created="2005-09-21T23:20:32.988-04:00" last-modified="2005-
  09-21T23:20:32.988-04:00" num-samples="96" num-sequences="7129">
    <source-experiments identifier="E-MICH-01" name="" description="" num-
    samples="0" num-sequences="0" />
  </experiment>
  <experiment identifier="E-ROSE-01" name="Profiling of breast cancer
  recurrence, Rosetta Inpharmatics" description="Complete dataset of
  Rosetta Breast cancer research" created="2005-09-21T23:03:08.255-
  04:00" last-modified="2005-09-21T23:03:08.255-04:00" num-
  samples="117" num-sequences="24481">
    <source-experiments identifier="E-ROSE-01" name="" description="" num-
    samples="0" num-sequences="0" />
  </experiment>
  ......
</workspace>
```

#### 4.2.4.2 Query

'Query' is one of three data types contained by Workspaces. The client program creates a new Query by specifying its type and limits, such as 'select all tissue samples collected from lung tumors'. The execution of Queries involves all three components of MAMA system. The client sends a Query to the server program, which parses it into an SQL query to the database. After results are returned from the database, the server wraps them into a 'Query Result' object and sends it to the client. Each pair of Query and Query Result objects is separately saved as two files within the directory of their Workspace.

The current version of MAMA supports the query of three data types: microarray experiments, biological samples, and nucleotide sequences. Fig. 18A shows a dialog box used to define a samples Query. Its upper half specifies the experiment, organism part, and material type of the required samples. The lower half, on the other hand, allows users to put up to three limits on any field of the samples. Therefore, the Query defined by Fig. 18A is interpreted as: 'select all samples obtained from the breast or lung tissue of the donators who had been diagnosed with cancer before 60 years old'. The Query is named 'Test Query' and its results are given in Fig. 18B, which shows that totally 59 samples in two experiments were selected from the database.

#### 4.2.4.3 Experiment

'Experiment' is the essential data object of the MAMA system. It contains the microarray dataset, and is the unit of query and data analysis results (Fig. 18B). It was arbitrarily decided that each Experiment included one and only one microarray dataset. The contents of this dataset is N sequences, M samples, and a 2-dimensional matrix of expression measurements whose size should be N x M. Therefore, each Experiment would be saved in Workspace as four files: metadata, sequences and their annotations, samples and their traits, and the expression data matrix.

Each microarray dataset stored in the MAMA database usually corresponded to an actual microarray study. In the client program, however, Experiment objects are not always equivalent to studies carried out in laboratories. They were often generated for a specific analysis by filtering, splitting, or combining the data of existing datasets. Therefore, they are sometime 'Virtual Experiments'. As shown in Fig. 17A, a new Virtual Experiment can be

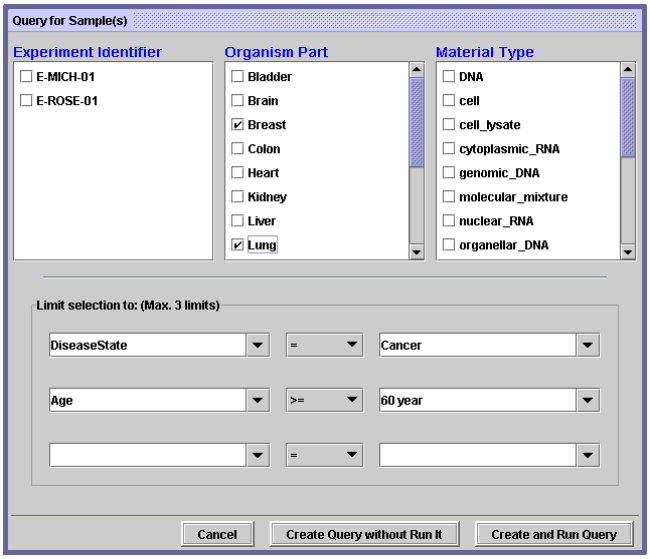



Figure 18A Specification of a New Database Query

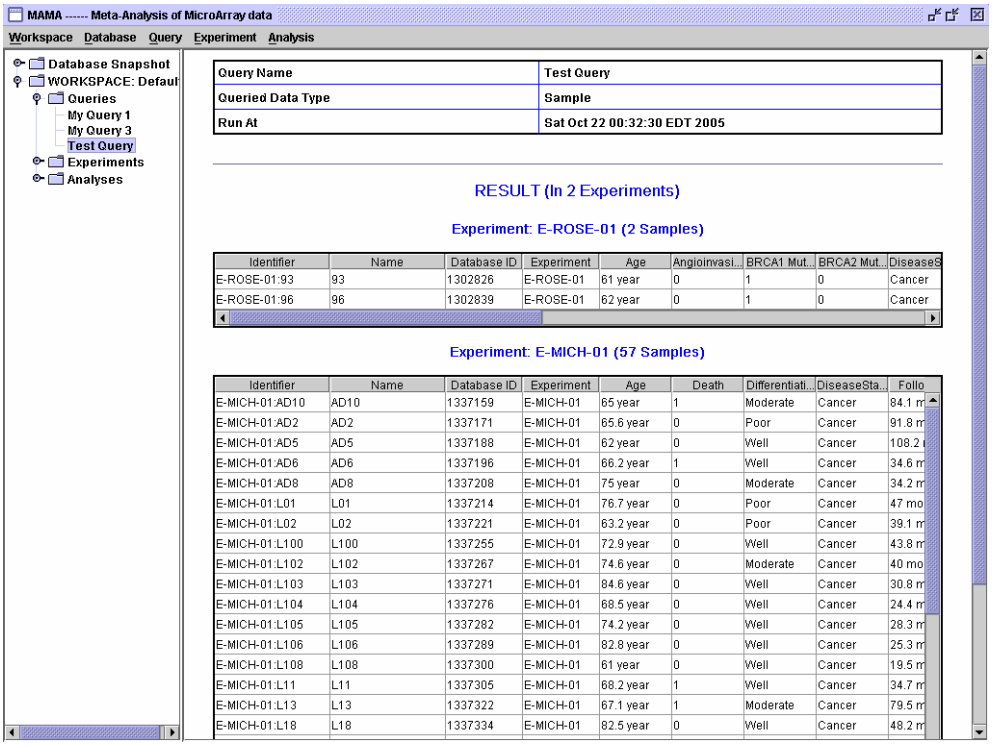



Figure 18B Results of a Database Query

## Figure 18 Demo: Database Query

**The user defines a 'Query' object by specifying parameters in a dialog box and executed it to retrieve results. (18A) The inputs specify a query for biological samples. This query asks for all samples that were taken from lung or breast tissue of cancer patients older than 60 at diagnosis. (18B) The execution of query defined in Fig. 18A finds qualified samples in two experiments (2 in E-ROSE-01 and 57 in E-MICH-01). The query results are saved and can be showed to users later in the right pane of the main window.**

created by directly importing data from text files or querying the database. The former option allows users to use the client program without exposing their source data to the public, which would be required by those researchers who had not published their data.

After an Experiment is created, it can be further customized to meet the requirements of subsequent data analysis. The current client program implements four types of data customization: sample filtering, sequence filtering, sample trait conversion, and normalization of expression measurements. Samples and sequences can be filtered with their annotations or values of summary statistics. For example, Fig. 19A specifies that all samples obtained from patients who were more than 40 years old will be removed from the Experiment, and Fig. 19B states that only sequences whose expression have a variance within the top 25% of all sequence will be kept. Conversion of trait values is used to change the scale or unit of a sample trait. Such modification is necessary when different studies measured a sample trait differently. For example, tumor size is represented as diameter (millimeter or centimeter) or TNM grading system (T1-T4) in different studies. Furthermore, discretizing continuous variables to categorical or interval variables was required by statistical methods like chi-square test. Fig. 19C shows a dialog box that changes the unit of patient follow-up data from ‘month’ to ‘year’ by dividing the original values by 12.0. Finally, the expression data of a Virtual Experiment is normalized by user specification. Since microarray data normalization had a large number of variations, the current version of MAMA was unable to implement enough normalization methods to meet the request of users. Instead, it allowed users to plug in their own methods using an API provided in the source codes. The process of method plug-in is discussed in the last section of this chapter.

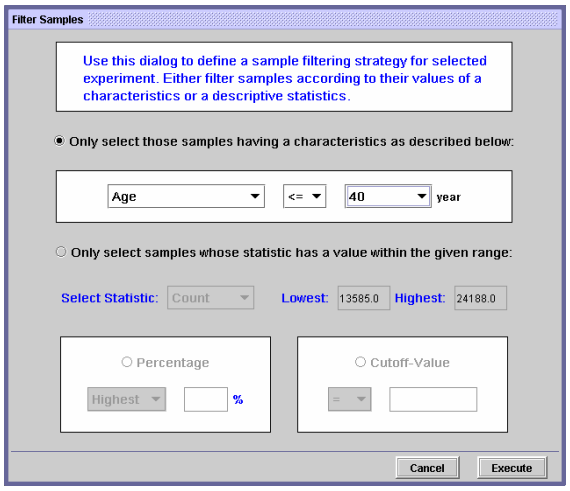



Figure 19A Example of Experiment Customization: Filter Samples

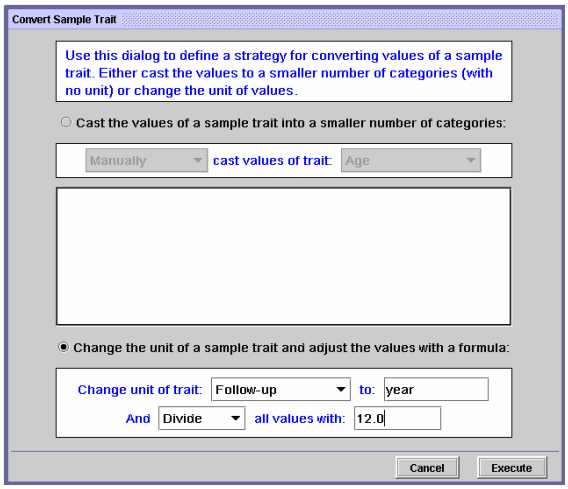



Figure 19C Example of Experiment Customization: Convert the Values of a Sample Trait

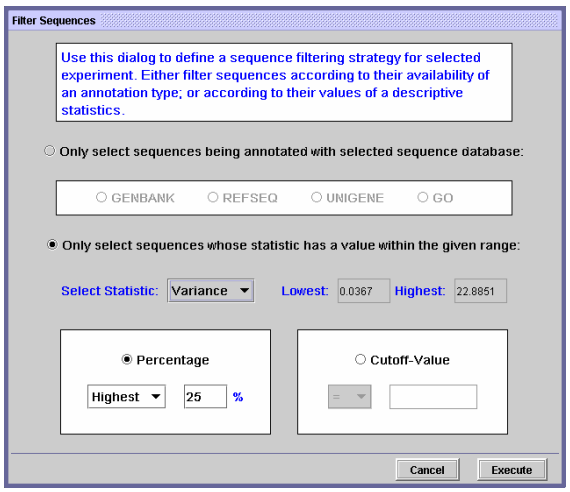



**Figure 49B Example of Experiment Customization: Filter Sequences**

## Figure 19 Demo: Microarray Experiment Customization

The client program is able to customize the contents of microarray experiment in various ways in order to perform certain data analysis procedure. (19A) Samples of an experiment can be filtered according to their values of a trait or a descriptive statistic. This dialog specifies that all samples taken from individuals older than 40 will be removed. (19B) Sequences of an experiment can be filtered according to their availability of an annotation type or a descriptive statistic. This dialog specifies that only sequences whose variance of expression measurements are in the top-25% will be kept. (19C) Values of a sample trait can be converted to a different type of variable. Such conversion is necessary when the original variable cannot be used by a specific statistical method. Users can choose to cast the original variable automatically or manually, or change its unit. In this dialog, it is specified that the unit of patient follow-up will be changed to 'year' by dividing each original values by 12.0.

#### 4.2.4.4 Analysis

'Analysis' objects define the data analysis operations executed by the client program. The attributes of an Analysis include name of the statistical method, Experiments to be analyzed, and how to report the results. The execution of an Analysis generates an 'Analysis Result' object, which is saved in a separate file. If multiple Experiments were included in an Analysis, the results obtained from individual experiments can be used for meta-analysis. The results of meta-analysis are put into the Analysis Result object.

The creation of an Analysis object had at least two steps. The first step is to specify the statistical method used for the analysis. As shown in Fig. 20A, the Rank Sum Test was selected to evaluate gene-trait correlation by measuring the differential expression of gene(s) between two sample groups. The second step was to specify the inputs of the Analysis with a dialog box (Fig. 20B). This dialog has three parts. The upper left box is used for specifying the variables to be analyzed. In Fig. 20B, the tested variable was the expression of all genes and the treatment variable was the survival outcome of patients. The upper right box is used for selecting the Experiments to be analyzed. All Experiments including the specified variables are listed. The bottom box specifies how to report the analysis results. If multiple Experiments are selected in this box, meta-analysis is possible. Results of data analysis are exported as text files if users want to process them using other programs such as Microsoft Excel.

### 4.2.5 Implementation of Analysis Methods

Because a large number of statistical methods have been applied to microarray analysis, the limited resources of developing MAMA system made it infeasible to satisfy users with

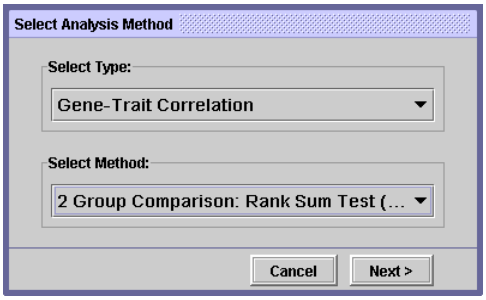



Figure 20A Selection of Data Analysis Method

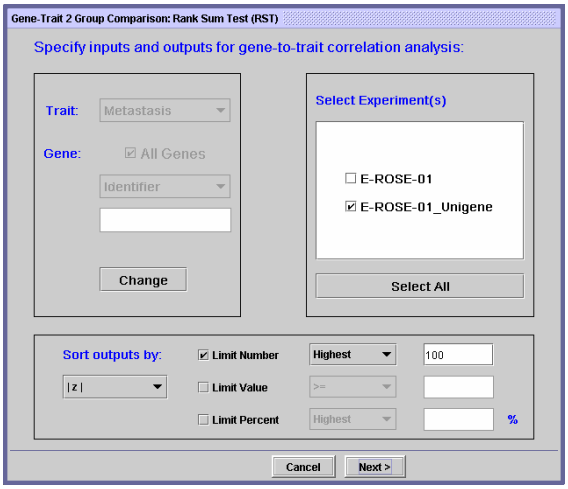
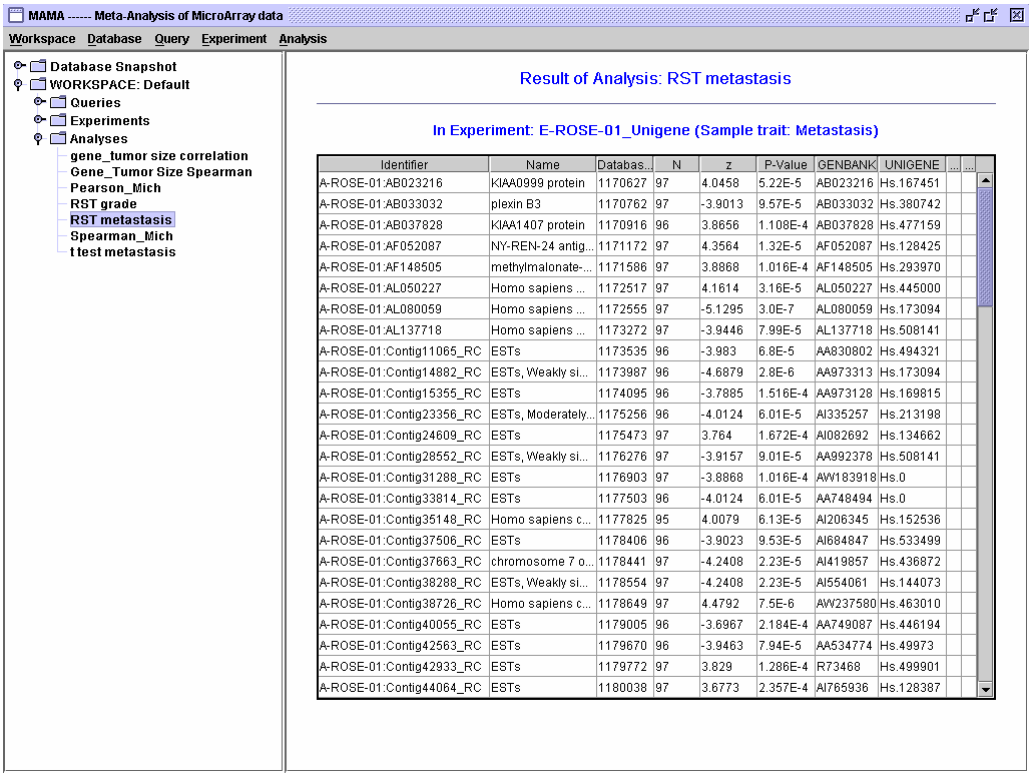



Figure 20B Specification of a New Data Analysis Operation




| Identifier | Name | Databas... | N | z | P-Value | GENBANK | UNIGENE |
|---|---|---|---|---|---|---|---|
| A-ROSE-01:AB023216 | KIAA0999 protein | 1170627 | 97 | 4.0458 | 5.22E-5 | AB023216 | Hs.167451 |
| A-ROSE-01:AB033032 | plexin B3 | 1170762 | 97 | -3.9013 | 9.57E-5 | AB033032 | Hs.380742 |
| A-ROSE-01:AB037828 | KIAA1407 protein | 1170916 | 96 | 3.8656 | 1.108E-4 | AB037828 | Hs.477159 |
| A-ROSE-01:AF052087 | NY-REN-24 antig... | 1171172 | 97 | 4.3564 | 1.32E-5 | AF052087 | Hs.128425 |
| A-ROSE-01:AF148505 | methylmalonate-... | 1171586 | 97 | 3.8868 | 1.016E-4 | AF148505 | Hs.293970 |
| A-ROSE-01:AL050227 | Homo sapiens ... | 1172517 | 97 | 4.1614 | 3.16E-5 | AL050227 | Hs.445000 |
| A-ROSE-01:AL080059 | Homo sapiens ... | 1172555 | 97 | -5.1295 | 3.0E-7 | AL080059 | Hs.173094 |
| A-ROSE-01:AL137718 | Homo sapiens ... | 1173272 | 97 | -3.9446 | 7.99E-5 | AL137718 | Hs.508141 |
| A-ROSE-01:Contig11065_RC | ESTs | 1173535 | 96 | -3.983 | 6.8E-5 | AA830802 | Hs.494321 |
| A-ROSE-01:Contig14882_RC | ESTs, Weakly si... | 1173987 | 96 | -4.6879 | 2.8E-6 | AA973313 | Hs.173094 |
| A-ROSE-01:Contig15355_RC | ESTs | 1174095 | 96 | -3.7885 | 1.516E-4 | AA973128 | Hs.169815 |
| A-ROSE-01:Contig23356_RC | ESTs, Moderately... | 1175256 | 96 | -4.0124 | 6.01E-5 | AI335257 | Hs.213198 |
| A-ROSE-01:Contig24609_RC | ESTs | 1175473 | 97 | 3.764 | 1.672E-4 | AI082692 | Hs.134662 |
| A-ROSE-01:Contig28552_RC | ESTs, Weakly si... | 1176276 | 97 | -3.9157 | 9.01E-5 | AA992378 | Hs.508141 |
| A-ROSE-01:Contig31288_RC | ESTs | 1176903 | 97 | -3.8868 | 1.016E-4 | AW183918 | Hs.0 |
| A-ROSE-01:Contig33814_RC | ESTs | 1177503 | 96 | -4.0124 | 6.01E-5 | AA748494 | Hs.0 |
| A-ROSE-01:Contig35148_RC | Homo sapiens c... | 1177825 | 95 | 4.0079 | 6.13E-5 | AI206345 | Hs.152536 |
| A-ROSE-01:Contig37506_RC | ESTs | 1178406 | 96 | -3.9023 | 9.53E-5 | AI684847 | Hs.533499 |
| A-ROSE-01:Contig37663_RC | chromosome 7 o... | 1178441 | 97 | -4.2408 | 2.23E-5 | AI419857 | Hs.436872 |
| A-ROSE-01:Contig38288_RC | ESTs, Weakly si... | 1178554 | 97 | -4.2408 | 2.23E-5 | AI554061 | Hs.144073 |
| A-ROSE-01:Contig38726_RC | Homo sapiens c... | 1178649 | 97 | 4.4792 | 7.5E-6 | AW237580 | Hs.463010 |
| A-ROSE-01:Contig40055_RC | ESTs | 1179005 | 96 | -3.6967 | 2.184E-4 | AA749087 | Hs.446194 |
| A-ROSE-01:Contig42563_RC | ESTs | 1179670 | 96 | -3.9463 | 7.94E-5 | AA534774 | Hs.49973 |
| A-ROSE-01:Contig42933_RC | ESTs | 1179772 | 97 | 3.829 | 1.286E-4 | R73468 | Hs.499901 |
| A-ROSE-01:Contig44064_RC | ESTs | 1180038 | 97 | 3.6773 | 2.357E-4 | AI765936 | Hs.128387 |

Figure 20C Results of a Data Analysis Operation

## Figure 20 Demo: Data Analysis Operation

Data analysis operations are initiated through a wizard: (20A) User selects the method category and the statistical method used for a new 'Analysis' object. (20B) User specifies the input variables of an analysis procedure and how to report the results. This dialog states that the correlation between cancer metastasis and all genes in experiment 'E-ROSE-01_Unigene' will be calculated with Rank Sum Test, and 100 genes having the highest magnitude of z statistic will be reported in the results. (20C) The results of an analysis are saved and can be showed to user later in the right pane of the main window.

enough methods. Consequently, a method plug-in mechanism is provided by MAMA to allow users to extend the client program by adding their own analysis methods. Therefore, the MAMA system is mostly considered as a platform of data-mining instead of a data analysis package.

Statistical methods are classified into several categories, such as gene-trait correlation, and MAMA provides an API for each method category. The inputs and outputs of methods in the same category have the same data types, which were specified in a Java abstract class. For example, the API of gene-trait correlation methods stated that any method in this category should be a hypothesis test involving a sample trait and the expression of a gene. The inputs of this test were two equal-length arrays of double values and the outputs included a test statistic and corresponding p-value. Once the API of a method category was available, plug-in of a new method into this category had two steps. First, users implement the API by extending the abstract class. In the sub-class, users specify the name of the new method and the symbol of test statistic, and implement the algorithm of the method. The second step of method plug-in is to register the new method. All registration information should be written to an XML document called ‘plugins.MAMA’, which is wrapped into the client program. To register a method, users need to add an ‘analysis-method’ XML element under its category. This element itself should include two required elements: ‘name’, which is the method name readable to users, and ‘class’, which is the name of the Java class implementing this method. Before the creation of a new Analysis object, this registration file is loaded and parsed into a list of categories and methods for users to select (Fig. 20A).

The following is a segment of the method registration file, which illustrates the registration information of two methods used to evaluate gene-trait correlation:

```
<analysis-type name="Gene-Trait Correlation">
    <analysis-method name="2 Group Comparison: Student's T" class="GeneToTraitStudentT">
    </analysis-method>
    <analysis-method name="2 Group Comparison: Rank Sum Test (RST)" class="GeneToTraitRST">
    </analysis-method>
    …
</analysis-type>
```

Meta-analysis methods can be implemented and registered using a similar plug-in mechanism. The current version of MAMA provides two categories of meta-analysis method: combined test and measurement of effect size (Appendix E).

# CHAPTER V
# CONCLUSION

Complicated diseases, such as Alzheimer's, cardiovascular diseases, and most types of cancer, are currently considered incurable because of the lack of systematic perspective about the molecular-level perturbations in individual patients. Development of high-throughput biological technologies presents a new opportunity to overcome these diseases. The enormous amount of data generated by these technologies is changing the face of biomedical studies, which involves statistical analysis and information process more and more. At the same time, researchers are being challenged by the requirement of translating technologies into clinical medicine. Properly designed data mining strategies are critical for recognizing causative information, which will help discovering new drug targets or making more reliable clinical decisions, from raw high-throughput data.

Microarray is relatively more mature and less expensive compared to other high-throughput technologies. Therefore, it has been commonly applied to the identification of gene expression patterns in specific types or subtypes of diseases. A variety of microarray datasets have been publicly available, which makes it possible to integrate multiple datasets for more powerful statistical analysis.

The current study was performed to solve some issues involved in the practical application of microarray data to cancer. It included two related projects. The data analysis

project justified the advantages of integrating microarray datasets and the necessity of developing a computer system like MAMA. The MAMA project, in return, will drastically accelerate the process of similar microarray analyses.

## 5.1 Data Analysis

The data analysis phase of this study was focused on the confirmation of two hypotheses. The first hypothesis was intended to validate the value of microarray technology in clinics by suggesting that microarray data provide extra clinical information besides commonly used indexes. The truthfulness of this hypothesis is the basis of all similar studies because its denial means that inclusion of microarray data in clinical decision will not make disease prognosis more accurate. Surprisingly, it has been overlooked by most microarray studies about cancer prognosis. Using two datasets about breast cancer and four datasets about lung cancer, this study attempted to validate this hypothesis by cross-validation of independent datasets and logistic regression models. The results implied that (1) when indexes were applied separately, gene expression profiles were superior classifiers of cancer patients than currently used prognostic indexes; (2) when all indexes were applied jointly, inclusion of expression profiles improved the overall accuracy of classification; and (3) to achieve optimal classification, expression profiles should be applied in corporation with other indexes. These implications altogether solidly confirmed the clinical value of microarray data.

At the beginning of this study, it was proposed to perform expression profiling across multiple microarray datasets. The hypothetical advantage of this strategy is that larger overall sample size will increase the generality of resultant expression profiles. At the same time, it may be criticized of ignoring the extensive diversity of independently generated microarray

datasets. Our study demonstrated that as long as assumptions were properly made and source data were consistently annotated and processed, expression profiles derived from multiple datasets would have better quality than those obtained from individual datasets. In specific patient subpopulations, genes highly correlated to clinical indexes are more likely to have observed significant correlation to cancer outcome than other genes, but they will lose their status in other subpopulations having different clinical background. This is one of the reasons why expression profiles obtained from independent studies are mostly inconsistent. On the other hand, genes not influenced by the confounder would win over the long haul as long as they had certain level of consistent correlation to outcome in general population. CDC20 and BECN1 are the examples of such genes (Table 8).

Expression profiles composed of those genes will be more reproducible, and more precisely differentiate patients into prognosis groups. The validation of this hypothesis is critical. Due to the high expense and complexity, microarray studies often do not have enough samples to obtain significant statistical results about cancer features. Therefore, reusing existing data by integration analysis will allow researchers to extract information or draw conclusions that can not be reached by analyzing individual datasets. The 60 reporter genes and their weight derived from the combined dataset made an optimal expression profile of breast cancer recurrence achievable using the given datasets. The value of this profile is worthy of some further investigation.

The lung cancer datasets were also used to ensure the advantage of data integration. The source data were re-processed differently because of the disparity between breast and lung cancers. Since lung cancer subtypes are highly dissimilar in terms of tissue type, survival rate, and so on, only adenocarcinoma patients were used. Patients were categorized into two

prognosis groups according to their 2-year survival outcome because recurrence outcome of most patients was not available and about 60% invasive lung cancer patients did not survive more than two years after diagnosis. As shown in Fig. 9 and 10, results from lung datasets also strongly advocated integration analysis.

In addition to the confirmation of those two hypotheses, the data analysis project of the current study also made the following conclusions:

The artificial variable SEP (Score for Expression Profile) was designed and successfully fitted into data analysis procedures. The utilization of SEP was the key of statistical analyses in this study. As a numeric and continuous variable, SEP was suitable for many analytical methods. For example, the density distribution of SEP scores in Fig. 1A and subsequent chi-square tests provided strong evidence about the general confounding effect of clinical indexes on gene-outcome correlation. Because of these confounders, the observed significant correlation of a gene to disease outcome could be the result of high correlation between the gene and a clinical index. Genes taking advantage of confounders will have higher chance to be selected as a reporter. This is why many reporter genes can be linked to one or more clinical indexes. Sampling criteria varies among studies. In the case of breast cancer datasets analyzed in this study, all patients in Rosetta dataset were lymph node-positive while Stanford dataset included both node-positive and -negative patients. Hence, controlling the confounding effect of clinical indexes will improve the generality of expression profiles.

A partial correlation procedure was the first strategy used in this study to control the confounders of expression profiling. The procedure calculated a residual value for each gene expression measurement and replaces the original measurements with the residuals in the following steps.  Although theoretically all confounders could be controlled by recursively

calculating residuals, extra variance will be introduced into results since the residuals were estimated from the sample data. Consequently, the resultant expression profiles would have higher false positive rate and lower quality. For example, when the 127-gene profile obtained from the partial correlation analysis of Rosetta dataset was cross-validated with Stanford dataset, it did not perform better than the profile generated by regular correlation analysis.

The major data analysis procedure of this study combined training/testing validation and bootstrap re-sampling. This procedure was used to avoid overfitting in results and make unbiased comparison of expression profiling strategies. However, it should be noted that this procedure did not take full advantage of the source data because all expression profiles were generated from the training data, which contained just about two-third of the complete datasets. For example, when the breast cancer 60-gene profile obtained from a complete dataset was cross-validated with the other dataset, the overall accuracy of patient classification using SEP was 73.61%, higher than the bootstrapping median (70.59%) obtained from within-dataset validation.

Results of this study advocated a breast cancer recurrence model suggesting that the progression of secondary tumors had two growth patterns. Each of these patterns corresponded to a post-diagnosis recurrence peak. Instead of arbitrarily categorizing patients according to their 5-year prognosis as what most breast cancer studies prefer, this study adopted this 2-peak model and classified patients into two groups corresponding to the peaks. Considering breast cancer as a cell growth abnormality, this classification had more biological grounds. Comparison of Fig. 6B and 8 shows that at N=100, the median AUC obtained from the lung data was 0.1 lower than that from the breast data, a relatively significant dropping. This difference might be caused by smaller sample size of two training

lung datasets, but more possibly because of the lack of a well-defined biological model to support the 2-year survival classification of patients. In addition, recurrence is a better output variable of expression profiling than survival because the factors influencing survival are more diverse.

The stepwise procedures were applied to trace the consequence of adjusting the sensitivity and specificity of reporter gene selection. The necessity of high reporter selection sensitivity was questioned by various results. First, as in Figure 5 and 6, medians of test statistics gradually reached a plateau with the raising of N. When N was greater than 60, increasing its value had very little influence on the quality of expression profiles. Secondly, cross-validation results demonstrated that two mostly different reporter gene list had similar performance on testing patients (Table 6). Thus, both lists must miss some true positive reporters since each of them was valid classifiers of testing patients and included some true positives. Finally, the result of the reduction process (Fig. 11A) showed that the loss of sensitivity could be tolerated to an extensive level without significantly reducing the quality of expression profiles. On the other hand, high selection specificity was proved to be critical. The relationship between higher reporter selection specificity and better expression profiles was suggested by results presented in Fig. 7. The combined dataset selected reporter genes more consistently, implying that the expression profiles obtained from it included less false positives. Furthermore, the replacement process (Fig. 11B) caused more dramatic subsequence than the reduction process. Comparison of Fig.10A and 10B concluded that with the same sensitivity, decreasing specificity of reporter selection would quickly decrease the quality of expression profiles. Therefore, an effective expression profile should include false positives as few as possible, but do not have to take in all or most of true positives.

Nevertheless, high sensitivity should still be preferred when no substantial tradeoff is required. The more true positives an expression profile includes, the more reliable and robust it will be. In this study, the optimal tradeoff was arbitrarily decided based on the observed trend as the value of N was increased. More investigation on this topic is expected in the future.

The data integration strategy used in the current study was straightforward and easy to perform. It assumed that patients of independent studies were sampled from the same population and their expression data had similar distribution and range after proper data re-processing. A major criticism of this strategy might be the information leaking due to the filtering of genes before combining two datasets having different array design. A large portion of genes in the source data were not included in the combined dataset because they were not in both datasets. However, both breast and lung combined datasets still included about 5,000 Unigene clusters. According to previous conclusion about reporter selection sensitivity, the quality of resultant expression profiles was merely influenced by the filtering process. If the data integration involves more than two microarray datasets, inconsistent array design will make the current strategy less feasible. For example, there were only about 1,000 Unigene clusters included by all four lung datasets used in this study. An alternative strategy of integration analysis is meta-analysis, the analysis of results obtained from individual studies. With meta-analysis, each gene will get a summary statistic no matter its presence in multiple datasets.

Microarray is an evolving technology. A pre-requisite of its clinical application is the standardization of platform, protocol, data analysis, and so on, which will make large-scale clinical tests doable and provide a common reference for sample categorization. Otherwise,

datasets generated by independent studies are not directly comparable. For example, when SEP was calculated with the same reporter genes and their weight in this study, scores of different patient cohorts usually did not have ranges analogous to each other. The standardization of microarray relies on the knowledge learned from the existing data. As more and more microarray datasets about cancer or other diseases are published, there is increasing interest on comparing and summarizing multiple datasets to discover general expression patterns, which help the design of standard array template. By successfully verifying and realizing the advantages of multi-dataset expression profiling, the current study will accelerate the standardization of microarray.

## 5.2 MAMA Project

Although its name highlighted meta-analysis, the MAMA system is more of a data-mining platform than a meta-analysis toolbox. Particularly, it provided users with a centralized storage of microarray datasets, a data annotation and management tool, a data-mining environment for simultaneously investigating multiple datasets. Therefore, any researcher interested in the expression profiling of tumor tissues may take advantage of it. MAMA is also an open-source project. Applications of MAMA include, but are not limited to:

- Store and share microarray datasets about cancer.
- Correlate the expression of genes to cancer features, such as recurrence or ER status of patients.
- Identify or confirm co-expression of genes across multiple datasets to help the building of genetic pathways.

- Generate gene expression patterns from one or multiple datasets, and validate these patterns with data from independent sources.
- Implement and test novel methods or procedures of microarray data analysis.
- Help researchers to discover clinical indexes or molecular markers of cancer.

A noticeable feature of the MAMA system is the simplicity of the server program. Due to the limited human and computer resources, a heavy duty server, which would handle data analysis operations for all users, was avoided to minimize the development and maintenance efforts. Therefore, MAMA does not provide a web-like interface for users to interact with the system through a web browser. Instead, all data manipulation and analysis functions were implemented in the client program, which need to be downloaded and installed by users themselves. Consequently, users have to take more responsibility on the execution of operations. For example, they need to ensure that their local computer meets the hardware requirements of complicate data analysis procedures. On the other side, this system architecture improved the extensibility of the MAMA system. Modification and addition of data analysis functions are limited at the client-side. Since all source codes are freely available, users are able to customize the functions of MAMA system without setting up their own database and server.

MAMA was developed as a highly flexible system for both of data manipulation and analysis. It is assumed that independent datasets should have similar subjects and definition of variables when they are integrated by meta-analysis or other statistical techniques. For example, two studies respectively examining prognosis of breast and lung cancer usually cannot be integrated because their subjects are too dissimilar. In practice, since each study has its own purpose and experimental design, datasets used by meta-analysis usually need to

be re-processed first. The data manipulation functions provided by the MAMA client allow users to filter sample patients or sequence and convert variables before specific analyses. Consequently, user-defined 'virtual' studies, which have a different objective from the original studies, can be carried out to discover new information from existing data. Furthermore, by establishing a method plug-in mechanism, MAMA allows users to implement and apply their own methods of expression data normalization.

Flexibility is critical for the usability of MAMA. Cancer is a complex disease involving many aspects. To identify an optimal gene expression pattern, researchers often want to conveniently try and compare different strategies (e.g. 5-year vs. 3-year prognosis) or methods (e.g. parametric vs. non-parametric test) of gene expression profiling. The MAMA system fulfilled this requirement by its high flexibility, which could be error-prone at the same time. If users do not thoroughly understand the data or methods of their analysis, variables could be incorrectly defined, methods could be misused or mistakenly implemented, and analysis results could be inaccurately interpreted. Therefore, the targeted users of MAMA are those already familiar with the characteristics and statistical methods of microarray analysis.

An important lesson learned from this study is the complexity of realizing the medical application of microarray technology. Although most biomedical researchers would agree that high-throughput technologies will have extensive application in clinical medicine, no substantial breakthrough has been made so far. A possible reason is that the current knowledge about cancer and other complex diseases is still not enough for researchers to fully take advantage of these technologies. Besides, datasets generated with these technologies usually have low quality and small sample size, probably the reason why results

of microarray studies usually have not been taken seriously by most medical practitioners. Microarray-based diagnosis requires the standardization of technology and the data analysis procedures. While microarray technologies will keep developing in near future, suitable and practicable data analysis procedure are essential now. Similar to the data analysis procedure used in clinical trials, samples selected from various subpopulations should be pooled together to draw more solid and general conclusions. This study presented such a procedure during the data analysis phase and the MAMA system will help other researchers to develop more.

Although microarray provides gene expression measurements at a genomic level, its value should not be exaggerated. The comprehensive description of biological systems should cover information at different levels, including sequence, mRNA, protein, metabolite, and so on. The integration of data at multiple levels will provide a better understanding about investigated subjects. For example, the results of this study demonstrated that expression pattern and other clinical indexes jointly accomplished the best prognostic model of breast cancer. While systems biology is recently becoming one of the most active topics of biomedical research, its success highly relies on the development of data integration techniques. This study shared some commonness with systems biology researches in terms of data integration. For example, data objects should be formally and consistently annotated. Therefore, the vision and process of developing the MAMA project are partly applicable to similar projects of systems biology.

While the current version of MAMA system has met its basic requirements, it is still prototypic. Future upgrades under consideration are:

- Data collection: More datasets will be loaded into MAMA database as a continuous effort.
- Data presentation: New data presentation functions will be added for users to navigate data contents more conveniently. Examples of such functions are rendering data distribution in diagrams and sorting or filtering analysis results in tables.
- Method categories: According to feedbacks from users, new categories of data analysis methods will be implemented and corresponding API for method plug-in will be provided.
- Prediction models: The current version of MAMA did not support the functions for generating prediction models, an important application of microarray data. Realizing this feature involves a major upgrade of MAMA source codes. New data objects need to be defined and new data analysis functions, such as testing a model with datasets, need to be implemented.

Biomedical informatics is a new but promising field. Its prospects are highly dependent upon the insight and vision of researchers. Presenting some fresh ideas to the research community, this study strongly supported the application of microarray on cancer clinics by its data analysis results and data mining platform.

# Appendix A:
# Demo of Data Analysis Steps Using a Pseudo-dataset

This demo uses a simple and artificial microarray dataset of 10 breast cancer patients to demonstrate some data analysis steps utilized in this study. Table 1 and 2 separately list the clinical data of all patients and the expression measurements of all sequences in the source dataset. The original study annotated the sequences using accessions of NCBI RefSeq database and has processes microarray images to generate a 2-dimensional matrix of gene expression data. The expression profiling procedure used in this study would start with the categorization of sample patients into prognosis groups.

**Table 1 Clinical Information of Sample Patients**

| Sample ID | Recurrence | Follow-up (year) | ER Status | Tumor Size | Grade | Age (year) |
|---|---|---|---|---|---|---|
| **p_1** | 1 | 2.53 | 1 | 2 | 2 | 43 |
| **p_2** | 0 | 6.44 | 1 | 2 | 1 | 44 |
| **p_3** | 1 | 1.66 | 0 | 2 | 3 | 41 |
| **p_4** | 1 | 1.3 | 1 | 2 | 3 | 41 |
| **p_5** | 0 | 11.98 | 0 | 2 | 3 | 48 |
| **p_6** | 1 | 1.16 | 1 | 1 | 2 | 49 |
| **p_7** | 0 | 10.14 | 0 | 2 | 1 | 46 |
| **p_8** | 0 | 8.8 | 0 | 2 | 3 | 48 |
| **p_9** | 0 | 1.29 | 1 | 1 | 3 | 48 |
| **p_10** | 1 | 6.64 | 1 | 1 | 2 | 38 |

**Table 2 Expression Measurements in Source Dataset**

| Sequence | p_1 | p_2 | p_3 | p_4 | p_5 | p_6 | p_7 | p_8 | p_9 | p_10 |
|---|---|---|---|---|---|---|---|---|---|---|
| NM_003000 | 0.03 | -0.09 | 0.08 | 0.08 | -0.01 | 0.09 | 0.1 | 0 | -0.02 | -0.08 |
| NM_003001 | 1.11 | -0.12 | 1.28 | 0.2 | 1.11 | -0.04 | 1.07 | -0.13 | 1.02 | -0.14 |
| NM_003002 | -0.26 | -0.17 | -0.35 | -0.26 | -0.29 | 0.03 | -0.06 | 0.12 | -0.27 | -0.13 |
| NM_003003 | -0.7 | -0.04 | -0.73 | -0.05 | -0.71 | -0.03 | -0.63 | -0.01 | -0.69 | 0.03 |
| NM_003004 | 0.82 | 0.25 | 0.6 | -0.08 | 0.78 | 0.16 | 0.63 | -0.23 | 0.65 | 0.07 |
| NM_003005 | -0.89 | 0.01 | -0.81 | 0.05 | -0.8 | 0.01 | -0.6 | -0.02 | -0.77 | 0.13 |
| NM_003006 | -0.78 | -0.01 | -0.71 | 0.13 | -0.86 | -0.09 | -0.7 | -0.01 | -0.76 | -0.03 |
| NM_003007 | -1.23 | 0.25 | -1.09 | 0.23 | -1.23 | 0.05 | -1.1 | 0.04 | -1.16 | 0.45 |
| NM_003033 | -1.29 | 0.25 | -1.11 | 0.26 | -1.34 | -0.01 | -1.16 | 0.08 | -1.19 | 0.44 |
| NM_198139 | 1.09 | -0.15 | 1.06 | -0.18 | 1.21 | 0 | 1.16 | -0.07 | 0.95 | -0.34 |

## A.1 Categorization of Sample Patients

Breast cancer patients are categorized into prognosis groups based on their follow-up data. Patients who had observed recurrence within three years after diagnosis are classified into poor prognosis group (p_1, p_3, p_4, and p_6). Patients who were followed up for at least three year and had no observed recurrence are classified into good prognosis group (p_2, p_5, p_7, and p_8). The follow-up of p_9 was too short and the recurrence of p_10 happened too late. These two patients cannot be put into either group and will be excluded from all the following steps.

## A.2 Mapping Sequences to Unigene Clusters

All sequences are mapped to Unigene clusters. File containing the mapping information between RefSeq and Unigene is available at NCBI website. Both of sequence NM_003007 and NM_198139 are mapped to cluster Hs.1968 (SEMG1), so the expression measurements

of these two sequences are averaged for each patient to get rid of redundancy. Sequence NM_003303 cannot be mapped to any Unigene cluster, so it is removed from the dataset. Table 3 gives the gene expression data of entire dataset after this step.

**Table 3 Contents of Dataset after Sample and Sequence Filtering**

| Sequence | Gene Name | p_1 | p_2 | p_3 | p_4 | p_5 | *p_6* | p_7 |
|---|---|---|---|---|---|---|---|---|
| Hs.1968 | SEMG1 | -0.07 | 0.05 | -0.02 | 0.03 | -0.01 | 0.02 | 0.03 |
| Hs.356270 | SDHD | -0.26 | -0.17 | -0.35 | -0.26 | -0.29 | 0.03 | -0.06 |
| Hs.444472 | SDHC | 1.11 | -0.12 | 1.28 | 0.2 | 1.11 | -0.04 | 1.07 |
| Hs.464184 | SEC14L1 | -0.7 | -0.04 | -0.73 | -0.05 | -0.71 | -0.03 | -0.63 |
| Hs.465924 | SDHB | 0.03 | -0.09 | 0.08 | 0.08 | -0.01 | 0.09 | 0.1 |
| Hs.506670 | SELPLG | -0.78 | -0.01 | -0.71 | 0.13 | -0.86 | -0.09 | -0.7 |
| Hs.546296 | SECTM1 | 0.82 | 0.25 | 0.6 | -0.08 | 0.78 | 0.16 | 0.63 |
| Hs.73800 | SELP | -0.89 | 0.01 | -0.81 | 0.05 | -0.8 | 0.01 | -0.6 |

## A.3 Pre-processing of Expression Measurements

Details about pre-processing expression data are given in 'Specification for Curation of Expression Measurements'. In this demo, it is assumed that all expression measurements are in good quality and have been $\log_{10}$-transformed. Therefore, all measurements are directly normalized. The first normalization step is to make the median of gene expression in each patient equal to 0.0 and the standard deviation (SD) equal to 1.0. For example, the median and SD of gene expression in patient p_1 are respectively -0.17 and 0.74, so each expression measurement of p_1 is subtracted by -0.17 and then divided by 0.74. Afterward, expression measurements of each gene are also normalized with the same process. The resultant normalized data matrix is given in Table 4.

**Table 4 Normalized Expression Measurements**

| Sequence | Gene Name | p_1 | p_2 | p_3 | p_4 | p_5 | p_6 | p_7 |
|---|---|---|---|---|---|---|---|---|
| Hs.1968 | SEMG1 | 0.14 | 2.34 | 0.61 | -0.83 | 0.45 | -0.18 | -0.14 |
| Hs.356270 | SDHD | 0.03 | -0.97 | -0.07 | -1.96 | -0.03 | 0.35 | 0.09 |
| Hs.444472 | SDHC | 0.23 | -1.62 | 0.44 | -0.22 | 0.23 | -1.6 | 0.22 |
| Hs.464184 | SEC14L1 | -0.14 | 1.62 | -0.18 | 0.14 | -0.25 | 0.3 | -0.84 |
| Hs.465924 | SDHB | 0.09 | -1.85 | 0.34 | 0.13 | -0.09 | 1.84 | -0.12 |
| Hs.506670 | SELPLG | -0.09 | 1.28 | 0.09 | 2.02 | -0.28 | -0.81 | -0.44 |
| Hs.546296 | SECTM1 | 0.11 | 0.67 | -0.07 | -1.4 | 0.07 | 0.47 | -0.11 |
| Hs.73800 | SELP | -0.99 | 1.38 | -0.74 | 1 | -0.82 | 0.74 | -0.88 |

## A.4 Re-sampling of Patients

Bootstrap strategy repeatedly re-samples patients to generate training and testing subgroups. The following steps will be applied to one of such bootstrap re-samplings. The results obtained from all re-samplings will be summarized to give unbiased estimation of test statistics. It is assumed that patient p_1, p_2, p_3, p_4, p_5, and p_7 are assigned to the training subgroup, leaving p_6 and p_8 in the testing subgroup.

## A.5 Correlation Analysis

The Pearson correlation coefficient (r) of each gene to recurrence outcome is calculated with data of all training patients. For example, r of sequence Hs.1968 (SEMG1) is calculated with {1, 0, 1, 1, 0, 0} and {0.14, 2.34, 0.61, -0.83, 0.45, -0.14}, and the result equals to -0.47. Resultant correlation coefficients of all genes are given in Table 5. Genes are also ranked according to the magnitude of their r, from the highest to the lowest. Coefficient r can be transformed to t statistic using formula: $t = r * ((n - 2) / (1 - r^2))^{1/2}$, where n is sample size.

**Table 5 Results of statistical tests on gene-recurrence correlation**

| Sequence | Gene Name | Pearson Correlation | | Partial Correlation | | Rank Sum Test | |
|---|---|---|---|---|---|---|---|
| | | r | rank | r | rank | r | rank |
| Hs.1968 | SEMG1 | -0.47 | 7 | -0.51 | 3 | 0.65 | 5.5 |
| Hs.356270 | SDHD | -0.24 | 8 | -0.04 | 8 | 0.65 | 5.5 |
| Hs.444472 | SDHC | -0.72 | 3 | 0.73 | 2 | -0.87 | 3 |
| Hs.464184 | SEC14L1 | -0.51 | 6 | -0.48 | 5 | -0.65 | 5.5 |
| Hs.465924 | SDHB | 0.95 | 1 | 0.8 | 1 | -1.96 | 1 |
| Hs.506670 | SELPLG | 0.64 | 4 | 0.05 | 7 | -1.09 | 2 |
| Hs.546296 | SECTM1 | -0.59 | 5 | -0.49 | 4 | 0.65 | 5.5 |
| Hs.73800 | SELP | 0.92 | 2 | -0.38 | 6 | 0.22 | 8 |

## A.6 Partial Correlation Analysis

ER status is the controlled variable in this demo. The following description uses sequence Hs.1968 as an example to demonstrate the process of controlling ER status from expression data.

1. Training patients are classified based on their ER status. ER-positive group includes patient p_1, p_2, and p_4 and ER-negative group includes patient p_3, p_5, and p_7.
2. Average expression level of Hs.1968 in ER-positive and –negative patients is separately calculated. The values are considered as conditional expected expression (E) of gene Hs.1968 in all patients.
    - $\text{Mean}^{+}$ {0.14, 2.34, -0.83} = 0.55
    - $\text{Mean}^{-}$ {0.61, 0.45, -0.14} = 0.31

3. The residuals are calculated by subtracting expression measurements in Table 4 with corresponding E values. In the case of Hs.1968, the residuals of all eight patients are:
    - {-0.41, 1.79, 0.30, -1.38, 0.14, -0.73, -0.45, -1.03}

After all expression measurements are transformed to residuals, the partial correlation coefficient (r') between each gene and the recurrence outcome is calculated with the residuals of training patients using the same formula of Pearson correlation. Table 5 gives the r' of each gene and the corresponding rank. The r and r' values of some genes, such as Hs.444472, are dramatically different.

## A.7 Wilcoxon Rank Sum Test (RST)

RST is performed on training data of each gene to calculate a Z statistic. The following description uses sequence Hs.1968 as an example to demonstrate the process of RST.

1. Training patients are put into two groups of opposite recurrence outcome. Group 1 includes patient p_1, p_3 and p_4, and group 2 includes patient p_2, p_5, and p_7. Size of both groups is three.
2. Expression measurements of gene Hs.1968 in all training patients are transformed to ranks. So, given data points {0.14, 2.34, 0.61, -0.83, 0.45, -0.14}, corresponding ranks will be {4, 1, 2, 6, 3, 5}. If there are equal data points, their ranks will be averaged.
3. Parameter $W_1$ is calculated as the summation of ranks assigned to group 1:
    - $W_1 = \Sigma\ ranks_{group1} = rank_{p\_1} + rank_{p\_3} + rank_{p\_4} = 4 + 2 + 6 = 12$
4. Parameter $U_1$ is calculated with $W_1$ and the size of group1:
    - $U_1 = W_1 - N_1\ (N_1 + 1) / 2 = 12 - 3\ (3 + 1) / 2 = 6$

5. Mean is calculated as:
    - Mean = $(N_1 + N_2) / 2 = (3 + 3) / 2 = 4.5$
6. Variance is calculated as:
    - Variance = $N_1 N_2 (N_1 + N_2 + 1) / 12 = 3 * 3 * (3 + 3 + 1) / 12 = 5.25$
7. Z statistic is calculated as:
    - $Z = (U – Mean) / Variance^{1/2} = (6 – 4.5) / 5.25^{1/2} = 0.655$

Table 5 also gives the Z statistic of each gene and the corresponding ranks. The ranks of genes having equal Z statistics are averaged.

## A.8 Calculation of SEP Score

In this demo, reporter genes are selected based on RST results. The number of reporters (N) is arbitrarily set to two. Therefore, top-ranked sequence Hs.465924 and Hs.506670 are selected as reporters and their weights are respectively -1.96 and -1.09. The SEP score of each patient is calculated with the following steps, and the intermediate results and final SEP scores are given in Table 6.

1. The expected expression level (E) of each reporter gene is calculated by averaging the expression measurement of each reporter in all training patients. In the case of Hs.465924, its value of E is calculated as:
    - $Mean_{Hs.465924}$ {0.09, -1.85, 0.34, 0.13, -0.09, -0.12} = -0.25
2. The difference between the observed and the expected expression levels of each reporter gene is calculated and then weighted by the RST Z statistic of the gene.
3. The resultant values obtained from the last step are summed up to generate a SEP score for each patient.

The SEP scores of patients can be applied to other statistical analysis to evaluate expression profiles or the strategy used to generate them. For example, if cutoff of SEP is set as 0, both testing patient p_6 and p_8 will be classified into poor prognosis group. According to actual observation of recurrence outcome, p_6 is correctly classified, but p_8 is not, giving a classification accuracy of 50%.

**Table 6 Step-by-step calculation of SEP scores**

| Patient | Hs.465924 (E = -0.25, W = -1.96) | | | Hs.506670 (E = 0.43, W = -1.09) | | | SEP |
|---|---|---|---|---|---|---|---|
| | X | X–E | W(X–E) | X | X–E | W(X–E) | |
| **p_1** | 0.09 | 0.34 | **-0.67** | -0.09 | -0.52 | **0.57** | ***-0.1*** |
| **p_2** | -1.85 | -1.6 | **3.14** | 1.28 | 0.85 | **-0.93** | ***2.21*** |
| **p_3** | 0.34 | 0.59 | **-1.16** | 0.09 | -0.34 | **0.37** | ***-0.79*** |
| **p_4** | 0.13 | 0.38 | **-0.74** | 2.02 | 1.59 | **-1.73** | ***-2.47*** |
| **p_5** | -0.09 | 0.16 | **-0.31** | -0.28 | -0.71 | **0.77** | ***0.46*** |
| ***p_6*** | 1.84 | 2.09 | **-4.1** | -0.81 | 0.38 | **-0.41** | ***-4.51*** |
| **p_7** | -0.12 | 0.13 | **-0.25** | -0.44 | -0.01 | **0.01** | ***-0.24*** |
| ***p_8*** | -0.21 | 0.04 | **-0.08** | 1.18 | 0.75 | **-0.82** | ***-0.9*** |

# Appendix B:
# Sample Class Diagrams of MAGE-OM

MAGE-OM (MicroArray Gene Expression – Object Model) is a complex data model developed by MGED (Microarray Gene Expression Data) Society to facilitate the sharing of microarray data between data systems. Because MAGE-OM is described with UML (Unified Modeling Language), definitions and relationships of entities within this model can be graphically represented with class diagrams. Fig. 1 gives the class packages and root classes of MAGE-OM. Classes have an object-oriented structure and most of them inherit class 'Extendable'. Fig. 2 is a UML class diagram including major classes of 'BioMaterial' package. The 'OntologyEntry' class defines the standard vocabularies that will be used to describe characteristics of biological samples.

References:


1. Source of figures: http://www.ebi.ac.uk/arrayexpress/Schema/MAGE/MAGE.htm
2. MAGE Web Home: http://www.mged.org/Workgroups/MAGE/mage.html
3. More MAGE: http://www.mged.org/Workgroups/MAGE/MAGEdescription2.pdf

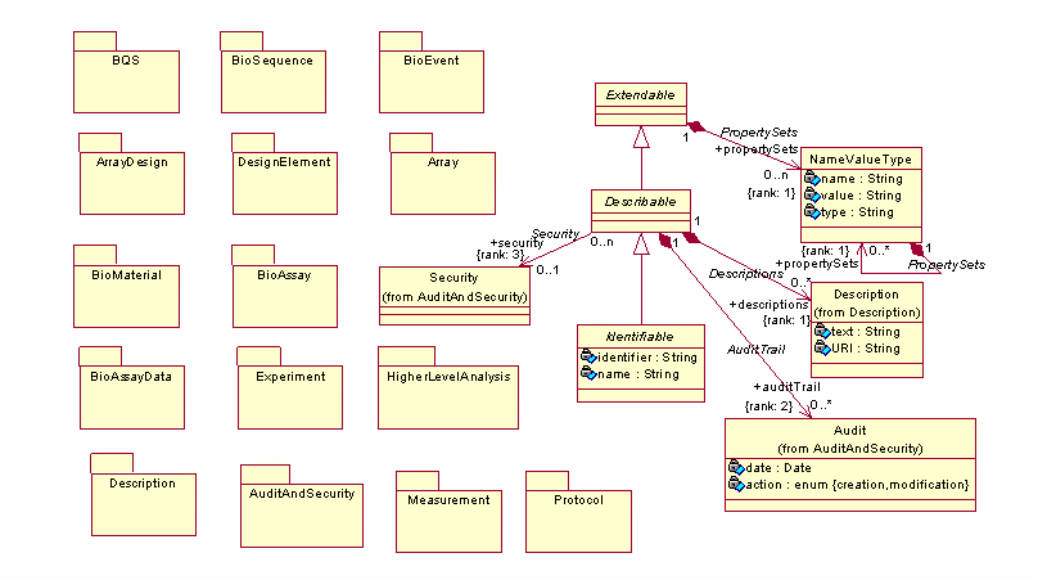



**Figure 1 Class Packages and Root Classes of MAGE-OM**

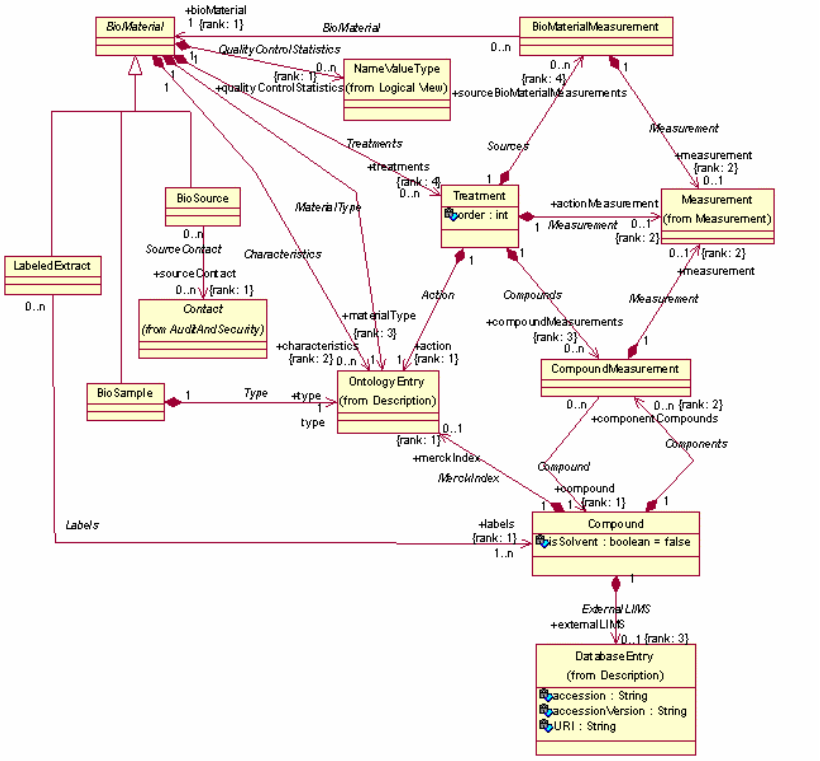



**Figure 2 Class Diagram of 'BioMaterial' Package**

# Appendix C:
# Architecture of Tomcat/Servlet Server

Java servlets are small server-side programs that response to connection of clients. They extend the functionality of web servers with improved performance and security. The execution of servlets needs Java Virtual Machine (JVM) and a service called ‘servlet engine’. The servlet engine loads a servlet the first time it is required by client, and keeps it activated to handle concurrent requests. The servlet keeps activated until it is explicitly unloaded or the servlet engine is stopped. Apache Tomcat is a container that provides a servlet-supporting environment. The Tomcat server includes a servlet engine, which incorporates servlets into a web server to make their services available to the clients. Fig. 1 demonstrates the general architecture of a Tomcat/Servlet server.

References:


1. Servlet Web Home: http://java.sun.com/products/servlet/
2. Servlet API: http://java.sun.com/products/servlet/2.2/javadoc/
3. Tomcat Web Home: http://tomcat.apache.org/

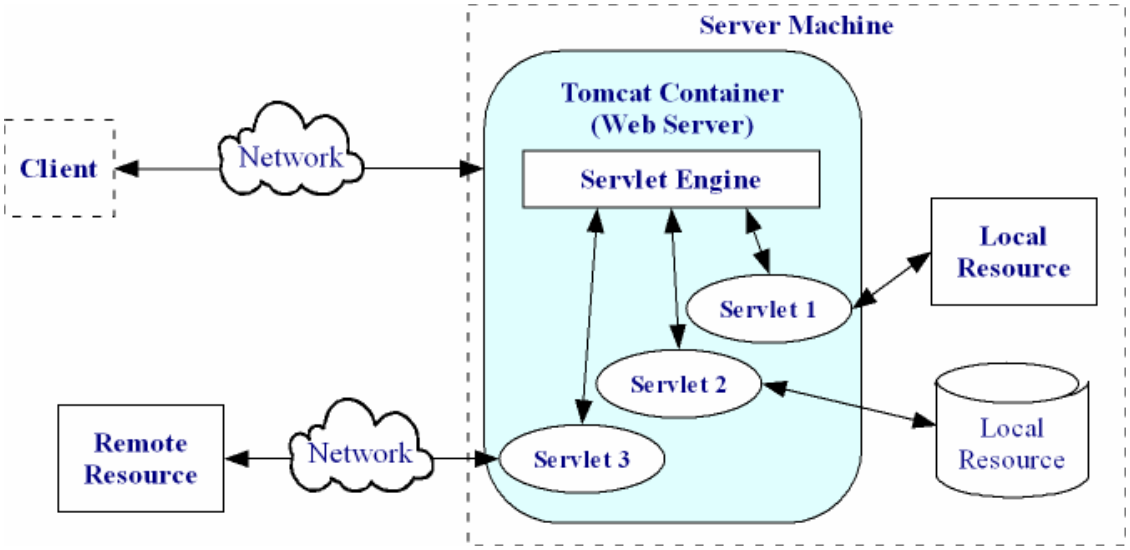



**Figure 1 Architecture of Tomcat/Servlet Server**

# Appendix D:
# Architecture of MVC Design Pattern

MVC (Model-View-Controller) software architecture is commonly used in applications having GUI. It breaks the functions of an application into three parts. The ‘Model’ maintains the contents of data objects and is independent of the visual representation of data. The ‘View’ has two major tasks. It determines how the data should be rendered on screen and responds to user actions. The ‘Controller’ accepts the user actions, handles the events, and consequently generates results. These three elements interact with each other to keep themselves updated. As showed in Fig. 1, a user action is received by View and passed to Controller, which will change Model and/or View after the action is handled. Model is independent of both View and Controller, and View is independent of Controller. Therefore, different types of functions are encapsulated, and the code updates in one element will not influence other elements as long as their interfaces keep unchanged.

References:


1. Source of figure: http://java.sun.com/blueprints/guidelines/designing_enterprise_applications_2e/app-arch/app-arch2.html#1106102
2. Inderjeet Singh, B.S., Mark Johnson, and the Enterprise Team, *Designing Enterprise Applications with the J2EETM Platform, 11.1.1 Model-View-Controller Architecture*. Second ed. 2002.

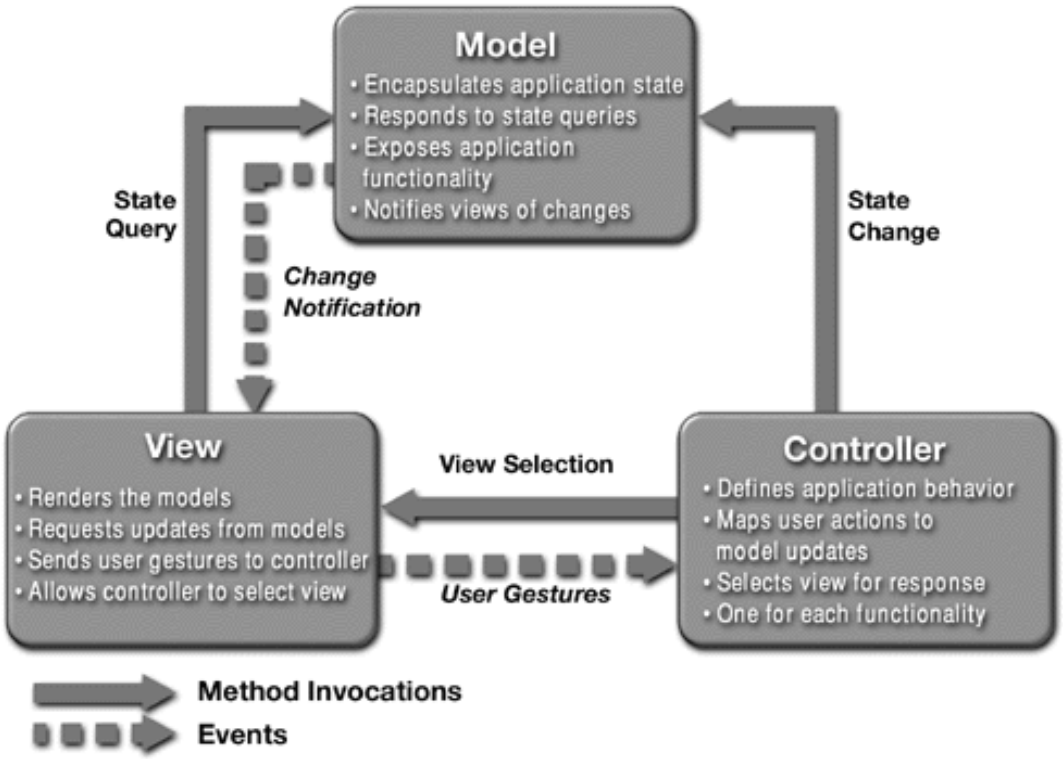



**Figure 1 Model-View-Controller Architecture**

# Appendix E:
# Meta-analysis Methods

Meta-analysis collects the results of individual studies to draw integrated conclusion and is often referred to as 'analysis of analyses'. Successful meta-analysis is able to reuse research resource to obtain information that cannot be made available by individual studies. There are two common types of meta-analysis methods: combined test and measures of effect size.

## E.1 Combined Test

Combined test is a procedure that summarizes the results from independent tests of the same hypothesis. It can be considered as a hypothesis test performed on the p-values or test statistics of individual tests.

Fisher combined test is one of the best known meta-analysis method. It uses the p-values of individual tests to calculate a $\chi^2$ statistic:

$$\chi^2 = -2 \Sigma \log_e p$$

The $\chi^2$ statistic obtained from this formula following chi-square distribution with degrees of freedom equal to 2N, where N is the number of tests combined.

Winer combined test uses the t statistics and df (degrees of freedom) of individual tests to calculate a Z statistic:

$$Z = (\Sigma t) / (\Sigma (df / (df - 2)))^{1/2}$$

In this formula, df / (df – 2) is the variance of a t distribution. When each df is no less than 10, the resultant Z statistic can be transformed to a p-value based standard normal distribution.

Stouffer combined test, on the other hand, uses z statistics of individual tests to calculate a summarized Z statistic:

$$Z = \Sigma z / N^{1/2}$$

In this formula, N is the number of tests combined. The resultant Z statistic follows standard normal distribution. When all results of individual tests are obtained from large samples, Winer and Stouffer combined tests will have nearly the same results.

The results of different combined tests are mostly consistent with each other although each method has its strengths and limitations. Before a combined test, it is often necessary to transform various statistics, such as t and z, into a common statistic or one-tailed p-value.

## E.2 Measures of Effect Size

Combined tests provide the statistical significance of the results, but do not any give insight into the strength of the relationship, which can be achieved by measures of effect size. The phrase ‘effect size’ means ‘the degree to which the null hypothesis is false’ while the null hypothesis states that the effect size is zero.

There are types of effect sizes commonly dealt with by meta-analysis methods:

1. differences of two groups in their means, as d statistic calculated by Student’s t test.
    - $d = |Mean_1 – Mean_2| / \sigma$
2. the degree of association between two variables, as Pearson correlation coefficient (r).

The average of these effect sizes obtained from individual studies is calculated by meta-analysis methods for further steps:

$$d_{average} = \Sigma d / N$$

and

$$r_{average} = \Sigma r / N$$

## E.3 Transformation of Test Statistics

For the purpose of meta-analysis, it is often necessary to transform test statistics to each other.

1. Transform to r:
    - t: $r = (t^2 / (t^2 + df))^{1/2}$
    - F: $r = (F / F + df(error))^{1/2}$
    - $\chi^2$: $r = (\chi^2 / n)^{1/2}$
    - d: $r = d / (d2 + 4)^{1/2}$
2. Transform to d:
    - t: $d = 2t / df^{1/2}$
    - F: $d = 2F^{1/2}/df(error)^{1/2}$
    - r: $d = 2r / (1 – r^2)^{1/2}$

# Appendix F:
# Complete Lists of Reporter genes

This appendix presents the complete reporter gene lists derived from breast and lung cancer microarray datasets using bootstrap procedure. In the tables, 'ID' is the Unigene accession of a reporter while 'Name' is its Unigene symbol. 'Count' represents how many times a reporter was ranked within top-100 by bootstrapping re-samplings. Finally, 'Weight' is the Z statistic of a reporter gene obtained from Wilcoxon Rank Sum Test (RST) applied on the data of all patients in the dataset.

## F.1 Reporter Lists of Breast Cancer

Table 1-3 separately give the 60-gene reporter list derived from two independent breast cancer datasets and their combination. These lists represent gene expression profiles corresponding to 3-year recurrence outcome of breast cancer and the list in Table 3 is recommended by this study. The counts of reporters are based on 10,000 bootstrapping re-samplings.

## F.2 Reporter Lists of Lung Cancer

Table 4-6 separately give the 60-gene reporter list derived from two independent breast cancer datasets and their combination. These lists represent gene expression profiles corresponding to 2-year survival outcome of lung cancer and the list in Table 6 is recommended by this study. The counts of reporters are based on 1,000 bootstrapping re-samplings.

**Table 1 The Reporter List Derived from Rosetta Breast Dataset**

| Sequence | Name | Count | Weight |
|---|---|---|---|
| Hs.435861 | SCUBE2 | 9991 | 6.57 |
| Hs.27860 | PTGER3 | 9463 | 5.51 |
| Hs.148767 | RQCD1 | 9432 | -5.4 |
| Hs.352962 | HRB | 9248 | -5.09 |
| Hs.171834 | PCTK1 | 8768 | -5.12 |
| Hs.25001 | YWHAG | 8743 | -5.05 |
| Hs.9589 | UBQLN1 | 8723 | -5.15 |
| Hs.433512 | ACTR3 | 8537 | -4.74 |
| Hs.20013 | P29 | 8213 | 4.66 |
| Hs.181461 | ARIH1 | 8119 | -4.7 |
| Hs.84113 | CDKN3 | 8114 | -4.64 |
| Hs.1578 | BIRC5 | 8079 | -4.55 |
| Hs.439200 | KIAA0090 | 8012 | 4.7 |
| Hs.2006 | GSTM3 | 7915 | 4.55 |
| Hs.351680 |  | 7855 | -4.72 |
| Hs.30743 | PRAME | 7646 | -4.68 |
| Hs.429 | ATP5G3 | 7590 | -4.52 |
| Hs.437546 | SMARCE1 | 7577 | 4.62 |
| Hs.283532 | BM039 | 7396 | -4.35 |
| Hs.421337 | XTP1 | 7038 | -4.25 |
| Hs.82285 | GART | 6862 | -4.39 |
| Hs.276466 | FLJ21062 | 6852 | 4.29 |
| Hs.178761 | PSMD14 | 6549 | -4.2 |
| Hs.287472 | BUB1 | 6547 | -4.2 |
| Hs.35096 | KAISO-L1 | 6383 | 4.14 |
| Hs.424966 | PIR | 6144 | -4.12 |
| Hs.79353 | TFDP1 | 6035 | -4.11 |
| Hs.155204 | ZNF174 | 5950 | 4.12 |
| Hs.25913 | PEX12 | 5771 | 4.07 |
| Hs.35120 | RFC4 | 5679 | -4 |
| Hs.184161 | EXT1 | 5652 | -4 |
| Hs.173162 | NOC4 | 5647 | -4 |

| | | | |
|---|---|---|---|
| Hs.2025 | TGFB3 | 5644 | 4.02 |
| Hs.23255 | NUP155 | 5636 | -4 |
| Hs.190389 | KIAA0266 | 5492 | 4 |
| Hs.128425 | NY-REN-24 | 5463 | 4 |
| Hs.153752 | CDC25B | 5249 | -3.92 |
| Hs.49932 | C21orf45 | 5246 | -3.92 |
| Hs.350966 | PTTG1 | 5204 | -3.85 |
| Hs.443793 | MIR | 5159 | 3.92 |
| Hs.53447 | KNSL8 | 5108 | 3.92 |
| Hs.77448 | ALDH4A1 | 5074 | 3.85 |
| Hs.78885 | BTD | 4962 | 3.92 |
| Hs.173034 | AMPH | 4940 | 3.85 |
| Hs.433951 | GPX4 | 4831 | 3.8 |
| Hs.7888 | | 4811 | 3.8 |
| Hs.388921 | PSMD2 | 4717 | -3.8 |
| Hs.348501 | PCM1 | 4652 | 3.74 |
| Hs.284153 | FANCA | 4650 | -3.8 |
| Hs.163091 | HIP14L | 4647 | -3.8 |
| Hs.407912 | COL4A2 | 4646 | -3.74 |
| Hs.77515 | ITPR3 | 4548 | -3.74 |
| Hs.81934 | ACADSB | 4538 | 3.8 |
| Hs.83383 | PRDX4 | 4515 | -3.7 |
| Hs.109706 | HN1 | 4382 | -3.7 |
| Hs.436187 | TRIP13 | 4343 | -3.74 |
| Hs.308045 | BRRN1 | 4308 | -3.7 |
| Hs.110457 | WHSC1 | 4303 | -3.66 |
| Hs.153357 | PLOD3 | 4281 | -3.66 |
| Hs.410784 | LRP8 | 4211 | -3.7 |

**Table 2 The Reporter List derived from Stanford Breast Dataset**

| Sequence | Name | Count | Weight |
|---|---|---|---|
| Hs.411509 | GSTP1 | 9866 | -5.17 |
| Hs.1657 | ESR1 | 9297 | 4.33 |
| Hs.329989 | PLK | 9268 | -4.25 |
| Hs.85137 | CCNA2 | 8925 | -4.09 |
| Hs.211589 | PPEF1 | 8822 | 4.28 |
| Hs.149156 | GLDC | 8755 | -4.14 |
| Hs.79748 | SLC3A2 | 8719 | -4.11 |
| Hs.94865 | TEAD4 | 8599 | -4.07 |
| Hs.150684 | XPO6 | 8465 | -4.04 |
| Hs.77329 | PTDSS1 | 8305 | -3.92 |
| Hs.287472 | BUB1 | 8049 | -4 |
| Hs.301011 | KIAA0876 | 8019 | 3.8 |
| Hs.79241 | BCL2 | 8003 | 3.85 |
| Hs.433984 | SLC4A2 | 7632 | -3.74 |
| Hs.12853 |  | 7585 | 3.7 |
| Hs.434367 | TXNRD1 | 7402 | -3.62 |
| Hs.82109 | SDC1 | 6652 | -3.47 |
| Hs.416854 | RERG | 6635 | 3.42 |
| Hs.178695 | MAPK13 | 6614 | -3.4 |
| Hs.267659 | VAV3 | 6421 | 3.42 |
| Hs.5372 | CLDN4 | 6149 | -3.36 |
| Hs.408219 | BCL7B | 6099 | -3.28 |
| Hs.435249 | KIAA1025 | 5763 | 3.24 |
| Hs.102471 | C6orf56 | 5627 | -3.22 |
| Hs.169946 | GATA3 | 5530 | 3.21 |
| Hs.78619 | GGH | 5439 | -3.19 |
| Hs.3416 | ADFP | 5113 | -3.1 |
| Hs.30901 | SLC39A3 | 5067 | -3.07 |
| Hs.406491 | TLE1 | 5043 | -3.08 |
| Hs.188011 | MS4A7 | 4994 | 3.08 |
| Hs.405774 | CTRL | 4918 | -3.05 |
| Hs.166071 | CDK5 | 4698 | -3 |

| | | | |
|---|---|---|---|
| Hs.225952 | PTPRT | 4690 | 2.98 |
| Hs.83114 | CRYZ | 4581 | 2.97 |
| Hs.69771 | BF | 4561 | 3 |
| Hs.278526 | RNTRE | 4555 | -2.97 |
| Hs.406458 | GPI | 4524 | -2.98 |
| Hs.159637 | VARS2 | 4432 | -2.95 |
| Hs.368149 | CCT7 | 4333 | -2.94 |
| Hs.90911 | SLC16A5 | 4281 | -2.94 |
| Hs.91728 | PMSCL1 | 4109 | -2.89 |
| Hs.155287 | KIAA0010 | 4019 | -2.82 |
| Hs.387906 | ABI-2 | 3993 | 2.82 |
| Hs.5719 | CNAP1 | 3984 | -2.86 |
| Hs.150444 | KIAA0373 | 3979 | 2.83 |
| Hs.82963 | GNRH1 | 3971 | 2.85 |
| Hs.437459 | MYO1E | 3965 | -2.85 |
| Hs.179718 | MYBL2 | 3913 | -2.85 |
| Hs.83583 | ARPC2 | 3862 | -2.83 |
| Hs.410784 | LRP8 | 3858 | -2.85 |
| Hs.460184 | MCM4 | 3857 | -2.8 |
| Hs.24395 | CXCL14 | 3836 | 2.82 |
| Hs.183800 | RANGAP1 | 3829 | -2.8 |
| Hs.12820 | USP39 | 3826 | -2.79 |
| Hs.430725 | RHOIP3 | 3812 | -2.79 |
| Hs.369358 | SRPK1 | 3810 | -2.82 |
| Hs.250712 | CACNB3 | 3770 | -2.77 |
| Hs.362805 | MEIS2 | 3754 | -2.79 |
| Hs.260555 | C14orf45 | 3695 | 2.77 |
| Hs.432750 | HPN | 3685 | 2.77 |

**Table 3 The Reporter List Derived from the Combination of Two Breast Datasets**

| Sequence | Name | Count | Weight |
|---|---|---|---|
| Hs.171834 | PCTK1 | 9862 | -4.94 |
| Hs.435861 | SCUBE2 | 9732 | 5.18 |
| Hs.287472 | BUB1 | 9681 | -4.56 |
| Hs.1657 | ESR1 | 9647 | 5.23 |
| Hs.35096 | KAISO-L1 | 9559 | 5.14 |
| Hs.25001 | YWHAG | 9426 | -4.94 |
| Hs.436187 | TRIP13 | 9360 | -4.48 |
| Hs.173162 | NOC4 | 8970 | -4.42 |
| Hs.85137 | CCNA2 | 8673 | -4.17 |
| Hs.169946 | GATA3 | 8651 | 4.93 |
| Hs.1578 | BIRC5 | 8595 | -4.6 |
| Hs.82906 | CDC20 | 8466 | -4.42 |
| Hs.410784 | LRP8 | 8428 | -4.76 |
| Hs.267659 | VAV3 | 8415 | 5.01 |
| Hs.78619 | GGH | 8407 | -4.29 |
| Hs.308045 | BRRN1 | 8299 | -4.5 |
| Hs.163091 | HIP14L | 8176 | -4.43 |
| Hs.12272 | BECN1 | 8158 | 4.92 |
| Hs.3416 | ADFP | 8043 | -4.45 |
| Hs.77329 | PTDSS1 | 8035 | -4.36 |
| Hs.83383 | PRDX4 | 7839 | -4.36 |
| Hs.79353 | TFDP1 | 7821 | -4.32 |
| Hs.301011 | KIAA0876 | 7750 | 4.94 |
| Hs.153752 | CDC25B | 7690 | -4.47 |
| Hs.9589 | UBQLN1 | 7634 | -3.94 |
| Hs.27860 | PTGER3 | 7542 | 4.79 |
| Hs.2006 | GSTM3 | 7365 | 4.62 |
| Hs.49932 | C21orf45 | 7178 | -4.69 |
| Hs.421337 | XTP1 | 7061 | -3.9 |
| Hs.79241 | BCL2 | 7047 | 4.59 |
| Hs.12109 | CIAO1 | 6994 | -4.07 |
| Hs.5719 | CNAP1 | 6931 | -4.1 |

| | | | |
|---|---|---|---|
| Hs.434367 | TXNRD1 | 6828 | -4.21 |
| Hs.78771 | PGK1 | 6698 | -4.14 |
| Hs.111554 | ARL7 | 6626 | -4.41 |
| Hs.77515 | ITPR3 | 6519 | -4.24 |
| Hs.81934 | ACADSB | 6417 | 4.68 |
| Hs.374378 | CKS1B | 6363 | -4.07 |
| Hs.406491 | TLE1 | 6356 | -4.3 |
| Hs.24395 | CXCL14 | 6352 | 4.37 |
| Hs.109706 | HN1 | 6339 | -4.34 |
| Hs.435326 | BAF53A | 6336 | -3.91 |
| Hs.153357 | PLOD3 | 6083 | -3.98 |
| Hs.350966 | PTTG1 | 6024 | -3.96 |
| Hs.433512 | ACTR3 | 5987 | -4.13 |
| Hs.413636 | C7orf14 | 5963 | -4.26 |
| Hs.278526 | RNTRE | 5925 | -4.16 |
| Hs.348501 | PCM1 | 5901 | 4.79 |
| Hs.311054 | ITGBL1 | 5724 | 4.38 |
| Hs.184161 | EXT1 | 5655 | -4.17 |
| Hs.171955 | TROAP | 5653 | -4.16 |
| Hs.188011 | MS4A7 | 5600 | 4.74 |
| Hs.20830 | KIFC1 | 5402 | -3.95 |
| Hs.424966 | PIR | 5357 | -3.96 |
| Hs.35962 | | 5350 | -3.99 |
| Hs.409065 | FEN1 | 5326 | -4.02 |
| Hs.226390 | RRM2 | 5259 | -3.99 |
| Hs.82285 | GART | 5256 | -3.88 |
| Hs.184601 | SLC7A5 | 5216 | -4.14 |
| Hs.69771 | BF | 5168 | 4.08 |

**Table 4 The Reporter List Derived from Harvard Lung Dataset**

| Sequence | Name | Count | Weight |
|---|---|---|---|
| Hs.412707 | HPRT1 | 1000 | -4.57 |
| Hs.447492 | PGAM1 | 949 | -3.66 |
| Hs.411312 | ITGA2B | 889 | 3.52 |
| Hs.41270 | PLOD2 | 841 | -3.3 |
| Hs.91747 | PFN2 | 827 | -3.24 |
| Hs.408093 | TCF2 | 825 | 3.28 |
| Hs.79037 | HSPD1 | 816 | -3.24 |
| Hs.119000 | ACTN1 | 797 | -3.12 |
| Hs.84136 | PITX1 | 761 | -3.07 |
| Hs.153647 | MATN2 | 745 | 3.03 |
| Hs.195825 | RBPMS | 725 | 2.95 |
| Hs.172589 | PWP1 | 693 | -2.88 |
| Hs.155048 | LU | 673 | 2.91 |
| Hs.381072 | PPIF | 662 | -2.88 |
| Hs.51 | PIGA | 634 | 2.77 |
| Hs.75318 | TUBA1 | 633 | -2.79 |
| Hs.293885 | GARS | 630 | -2.74 |
| Hs.409965 | PNN | 620 | -2.71 |
| Hs.436181 | HOXB7 | 595 | -2.69 |
| Hs.89901 | PDE4A | 593 | 2.7 |
| Hs.77917 | UCHL3 | 584 | -2.64 |
| Hs.75823 | AF1Q | 581 | -2.64 |
| Hs.154672 | MTHFD2 | 578 | -2.63 |
| Hs.2006 | GSTM3 | 573 | 2.65 |
| Hs.79347 | KIAA0211 | 569 | 2.65 |
| Hs.155206 | STK25 | 531 | -2.63 |
| Hs.512601 | HUMCYT2A | 529 | -2.57 |
| Hs.446579 | HSPCA | 528 | -2.56 |
| Hs.58414 | FLNC | 527 | -2.56 |
| Hs.282260 | RPE | 526 | -2.57 |
| Hs.77613 | ATR | 521 | -2.53 |
| Hs.59889 | HMGCS2 | 507 | 2.48 |

| | | | |
|---|---|---|---|
| Hs.409065 | FEN1 | 504 | -2.41 |
| Hs.79110 | NCL | 498 | -2.51 |
| Hs.75160 | PFKM | 480 | -2.46 |
| Hs.290432 | HOXB2 | 473 | -2.43 |
| Hs.78771 | PGK1 | 460 | -2.43 |
| Hs.512587 | MST1 | 442 | 2.43 |
| Hs.150358 | DPYSL3 | 441 | -2.34 |
| Hs.10842 | RAN | 440 | -2.4 |
| Hs.420563 | NDUFS1 | 437 | -2.41 |
| Hs.73769 | FOLR1 | 429 | 2.41 |
| Hs.512711 | TPI1 | 428 | -2.32 |
| Hs.46319 | SHBG | 428 | 2.32 |
| Hs.67928 | ELF3 | 418 | 2.3 |
| Hs.155079 | PPP2R5A | 415 | 2.28 |
| Hs.433941 | SEPW1 | 414 | 2.34 |
| Hs.245540 | ARL4 | 407 | -2.25 |
| Hs.437475 | STAT6 | 403 | 2.33 |
| Hs.360033 | KIAA0186 | 402 | -2.28 |
| Hs.79081 | PPP1CC | 398 | -2.3 |
| Hs.6906 | RALA | 386 | -2.3 |
| Hs.640 | CALCR | 380 | -2.2 |
| Hs.278311 | PLXNB1 | 375 | 2.22 |
| Hs.83583 | ARPC2 | 372 | -2.29 |
| Hs.111903 | FCGRT | 369 | 2.22 |
| Hs.226755 | YWHAH | 361 | -2.22 |
| Hs.1420 | FGFR3 | 360 | 2.26 |
| Hs.181973 | CYP2A13 | 359 | 2.22 |
| Hs.118127 | ACTC | 356 | -2.16 |

**Table 5 The Reporter List Derived from Michigan Lung Dataset**

| Sequence | Name | Count | Weight |
|---|---|---|---|
| Hs.352962 | HRB | 946 | -3.85 |
| Hs.75968 | TMSB4X | 930 | 3.74 |
| Hs.75514 | NP | 917 | -3.66 |
| Hs.99029 | CEBPB | 912 | -3.62 |
| Hs.511822 | CRK | 908 | -3.7 |
| Hs.119192 | H2AFZ | 900 | -3.47 |
| Hs.576 | FUCA1 | 881 | 3.42 |
| Hs.73800 | SELP | 853 | 3.42 |
| Hs.231975 | CREM | 850 | -3.47 |
| Hs.517814 | CCR2 | 829 | 3.27 |
| Hs.77961 | HLA-B | 822 | 3.21 |
| Hs.156324 | PRKACB | 803 | 3.24 |
| Hs.304682 | CST3 | 796 | 3.21 |
| Hs.408615 | P2RX5 | 783 | 3.15 |
| Hs.362807 | IL7R | 721 | 3.08 |
| Hs.2375 | EMR1 | 711 | 2.98 |
| Hs.17287 | KCNJ15 | 708 | 3.03 |
| Hs.75671 | STX1A | 690 | -3.03 |
| Hs.381072 | PPIF | 662 | -2.89 |
| Hs.433416 | NME2 | 653 | -2.92 |
| Hs.169824 | KLRB1 | 649 | 2.82 |
| Hs.83795 | IRF2 | 642 | 2.89 |
| Hs.433888 | RAB11B | 634 | 2.89 |
| Hs.1765 | LCK | 612 | 2.77 |
| Hs.173381 | DPYSL2 | 580 | 2.76 |
| Hs.79993 | PEX7 | 575 | -2.74 |
| Hs.436949 | CD6 | 557 | 2.65 |
| Hs.162757 | LRP1 | 531 | 2.69 |
| Hs.414480 | DBP | 530 | 2.61 |
| Hs.386748 | MS4A2 | 527 | 2.63 |
| Hs.512640 | PRKCSH | 526 | 2.64 |
| Hs.394609 | SORT1 | 506 | 2.69 |

| | | | |
|---|---|---|---|
| Hs.75932 | NAPA | 496 | 2.61 |
| Hs.119651 | GPC3 | 494 | 2.5 |
| Hs.285091 | C18orf1 | 482 | 2.57 |
| Hs.142912 | FZD2 | 475 | 2.56 |
| Hs.278426 | PDAP1 | 473 | -2.54 |
| Hs.435342 | SLU7 | 468 | 2.54 |
| Hs.1578 | BIRC5 | 466 | -2.54 |
| Hs.150580 | SUI1 | 455 | -2.54 |
| Hs.169476 | GAPD | 446 | -2.54 |
| Hs.434367 | TXNRD1 | 443 | -2.51 |
| Hs.95327 | CD3D | 434 | 2.43 |
| Hs.73793 | VEGF | 428 | -2.47 |
| Hs.524835 | UBC | 415 | -2.43 |
| Hs.12013 | ABCE1 | 411 | -2.39 |
| Hs.73172 | GFI1 | 406 | 2.4 |
| Hs.159494 | BTK | 405 | 2.37 |
| Hs.409934 | HLA-DQB1 | 397 | 2.37 |
| Hs.172609 | NUCB1 | 393 | 2.36 |
| Hs.439911 | TERT | 392 | -2.37 |
| Hs.57718 | CHRNA2 | 382 | 2.28 |
| Hs.246381 | CD68 | 378 | 2.28 |
| Hs.150930 | XRCC4 | 377 | -2.36 |
| Hs.433319 | CTF1 | 374 | 2.32 |
| Hs.417361 | UGP2 | 369 | -2.4 |
| Hs.91390 | PARG | 369 | -2.32 |
| Hs.169849 | MYBPC1 | 357 | 2.29 |
| Hs.68877 | CYBA | 353 | 2.24 |
| Hs.388617 | RORA | 353 | 2.24 |

**Table 6 The Reporter List Derived from Combination of Two Lung Datasets**

| Sequence | Name | Count | Weight |
|---|---|---|---|
| Hs.381072 | PPIF | 965 | -4.33 |
| Hs.412707 | HPRT1 | 911 | -3.97 |
| Hs.75514 | NP | 904 | -4.08 |
| Hs.409065 | FEN1 | 882 | -3.84 |
| Hs.433416 | NME2 | 873 | -3.87 |
| Hs.79037 | HSPD1 | 873 | -3.85 |
| Hs.119192 | H2AFZ | 868 | -3.81 |
| Hs.41270 | PLOD2 | 846 | -3.82 |
| Hs.433888 | RAB11B | 840 | 3.91 |
| Hs.10842 | RAN | 822 | -3.76 |
| Hs.576 | FUCA1 | 817 | 3.7 |
| Hs.55279 | SERPINB5 | 804 | -3.77 |
| Hs.155048 | LU | 764 | 3.72 |
| Hs.75318 | TUBA1 | 750 | -3.69 |
| Hs.304682 | CST3 | 735 | 3.56 |
| Hs.195825 | RBPMS | 712 | 3.58 |
| Hs.78771 | PGK1 | 711 | -3.63 |
| Hs.447492 | PGAM1 | 696 | -3.56 |
| Hs.352962 | HRB | 693 | -3.59 |
| Hs.172589 | PWP1 | 668 | -3.46 |
| Hs.94367 | TITF1 | 654 | 3.5 |
| Hs.433941 | SEPW1 | 639 | 3.4 |
| Hs.463110 | ANXA8 | 631 | -3.46 |
| Hs.75671 | STX1A | 625 | -3.43 |
| Hs.6906 | RALA | 623 | -3.47 |
| Hs.58414 | FLNC | 619 | -3.48 |
| Hs.435342 | SLU7 | 616 | 3.44 |
| Hs.1578 | BIRC5 | 583 | -3.38 |
| Hs.154672 | MTHFD2 | 583 | -3.34 |
| Hs.153884 | APACD | 581 | -3.39 |
| Hs.414480 | DBP | 579 | 3.33 |
| Hs.3281 | NPTX2 | 572 | -3.42 |

| | | | |
|---|---|---|---|
| Hs.2006 | GSTM3 | 562 | 3.4 |
| Hs.437475 | STAT6 | 562 | 3.4 |
| Hs.78563 | UBE2G1 | 552 | -3.33 |
| Hs.149957 | RPS6KA1 | 552 | 3.31 |
| Hs.436657 | CLU | 526 | 3.26 |
| Hs.79347 | KIAA0211 | 518 | 3.31 |
| Hs.1420 | FGFR3 | 516 | 3.28 |
| Hs.32393 | DARS | 516 | -3.29 |
| Hs.394609 | SORT1 | 501 | 3.24 |
| Hs.154846 | PIK4CB | 494 | 3.25 |
| Hs.111903 | FCGRT | 487 | 3.21 |
| Hs.404814 | VDAC1 | 486 | -3.28 |
| Hs.348500 | VIPR1 | 486 | 3.35 |
| Hs.446429 | PTGDS | 483 | 3.26 |
| Hs.405958 | CDC6 | 473 | -3.35 |
| Hs.82432 | KIAA0089 | 456 | 3.18 |
| Hs.119000 | ACTN1 | 455 | -3.18 |
| Hs.356342 | RPL27A | 443 | -3.22 |
| Hs.352119 | GGT1 | 441 | 3.18 |
| Hs.77917 | UCHL3 | 436 | -3.17 |
| Hs.83795 | IRF2 | 433 | 3.11 |
| Hs.1594 | CENPA | 429 | -3.19 |
| Hs.75932 | NAPA | 428 | 3.1 |
| Hs.191990 | PRKCBP1 | 422 | -3.1 |
| Hs.282260 | RPE | 419 | -3.09 |
| Hs.79993 | PEX7 | 416 | -3.14 |
| Hs.169824 | KLRB1 | 410 | 3.25 |
| Hs.81892 | KIAA0101 | 407 | -3.09 |

# Appendix G:
# Requirements of MAMA Project

## G.1 General Description

Microarray technology is a powerful tool for the research of complex diseases like cancers. Many microarray datasets have been generated from cancer tissues in order to identify gene expression profiles corresponding to various features of tumors. However, due to the high expense of microarray experiment, the sample sizes of individual microarray studies (usually around 100) are rather small comparing to the number of genes (up to 30 thousands) under investigation. The generality of profiles identified from single datasets are then questionable. One solution to this issue is to increase the sample size and power of statistical analyses by integrating information from multiple datasets. Results from previous studies already showed that despite of various inter-study variations, independent microarray datasets did share significant consistence if proper assumptions and data processing were made.

It is not technically straightforward to achieve analysis involving multiple microarray datasets. Datasets from independent resources are processed and formatted differently. Each study has its own experiment design, so the cancer tissues may be sampled from disparate populations. Even when two studies use identical samples, inter-dataset variations still could be significant because microarray experiment is a complicate process during which factors including experimenter, protocol, instrument, and array quality may introduce systematic bias into the final measurements. Furthermore, statistical methods used for microarray analysis are becoming more and more complex and sophisticated, and increasing sample size

will aggravate the computational burden of these methods. For biologists and statisticians who want to focus their work on the high-level data analysis, dealing with these issues is distracting and time-consuming.

The general purpose of MAMA project is to provide researchers a data mining platform that will support the precise gene expression profiling of cancer tissues using microarray data from independent resources. Users of MAMA will be able to access a centralized repository of microarray datasets about cancer and investigate them simultaneously, so they can identify gene expression profiles of certain features of cancer patients, such tumor stage or patient survival. The data repository will be a relational database located on a server machine. All datasets will be processed and formatted with same criteria before they are loaded into the database. Besides gene expression measurements, the database will also store related information about microarray experiments, such as author contact, experiment design and clinical scenario of patients. This database is accessible to users through a server program, which can handle concurrent requests from multiple clients. The client program developed by MAMA project will be data analysis application, which implements statistical methods suitable for microarray analysis. Other components of MAMA system include user interface, web server, data processing package, and so on.

## G.2 Definitions

Workspace --- It is the root of data object in MAMA client program within which end users perform operations such as manipulating, retrieving, and analyze data. Each client program opens and operates on one and only one workspace at a time. Workspaces can be

stored on local disk as XML documents. Workspace maintains the information about data, results, and procedures in a hierarchical tree structure.

Virtual experiment --- Virtual experiment is a key feature of MAMA system. Each virtual experiment is built with genes and samples originated from one or more original studies. Building virtual experiments will allow researchers to analyze data with purposes different from those of the original studies. For example, with a microarray dataset originally used to profile normal and cancer lung tissues, users can select only the data from the cancer tissues to build a virtual experiment and use this experiment to profile subtypes of lung cancer. In MAMA system, virtual experiments are the units to which analysis methods are applied.

Meta-analysis --- Meta-analysis is the technique used to draw summary conclusions from the results of multiple independent studies. Therefore, the inputs of meta-analysis are the results of individual studies rather than the source data of those studies. Implementing meta-analysis algorithms is a major, but not the only, purpose of developing MAMA system. For example, meta-analysis can be used to evaluate the consistence of several expression profiles or to combine test statistics obtained from individual tests.

Expression profile --- Expression profiles are identified from one or more microarray datasets using certain statistical methods and can be used to classify sample tissues. An expression profile includes a group of reporter genes and their relative weights.

Metadata of microarray experiment --- In MAMA system, metadata refer to all information related to microarray experiments except the gene expression measurements, including attributes of experiments, samples, and genes. Users are commended to investigate the metadata of a dataset before loading the complete dataset into the client program because of the large size of the expression measurements matrix.

MIAME, MAGE, and MGED ontology --- The MGED (Microarray Gene Expression Data) society (www.mged.org) has developed three standards for description of microarray data: MIAME, MAGE, and MGED ontology. MIAME (Minimum Information About a Microarray Experiment) describes the information needed to unambiguously interpret results of a microarray experiment and potentially reproduce it. Providers of microarray data can satisfy MAMA requirements by following up a checklist. The database of this system will be designed as MIAME-compatible. MAGE (MicroArray and Gene Expression) is a standard defining the entities related to microarray experiments and their relationship. MAGE is composed of an object model represented with UML (MAGE-OM) and a markup language developed as an XML-DTD (MAGE-ML). MAGE-OM captures the information specified by MIAME. The MGED ontology provides a standard specification of microarray-related vocabularies and their relationship. MAMA database adopts MAGE-OM for database schema and MGED ontology for description of samples and other data objects.

## G.3 Functional Requirements

Database:

1. A relational database schema
2. MAGE-OM and MIAME compliant
3. Use controlled vocabularies, such as MGED ontology, to describe samples, experiment design, and other data objects.
4. Support storage of microarray datasets generated on both of oligonucleotide and cDNA platform.
5. Have genes annotated by accessions of common sequence databases.

6. Separately store frequently requested data, such as metadata about microarray experiments, for quick access.
7. Accept data submission through the web.

Server Program:

1. Access to the database.
2. Process microarray datasets by following a standard guideline before loading them into the database.
3. Execute build-in database queries for experiments, samples, sequences and expression measurements.
4. Implement a pre-defined communication protocol between the server and the client programs.
5. Handle concurrent requests from multiple clients.

Client Program:

1. Load/retrieve data to/from the database by communicating with the server program.
2. Import microarray datasets from the database or text files
3. Maintain data objects in a tree-like structure.
4. Save data on local disk as XML documents and map between java objects and XML elements.
5. Manipulate data objects with operations such as filtering samples and normalizing expression measurements.

6. Have a graphic user interface and render data objects with GUI components such as lists and tables.
7. Implement common statistical processes of microarray analysis, such as calculation of descriptive statistics and gene-feature correlation.
8. Provide APIs for plug-in of user-specific methods.
9. Support meta-analysis of microarray data.

User Interface:

1. Render data objects and their relationship in a way consistent to the data model.
2. Allow for browsing of data objects in a folder-like structure.
3. Provide Wizards for multi-step operations.
4. Group related operations into menus.

## G. 4 Non-functional Requirements

Performance --- The performance of MAMA system is mostly dependent upon the network connection and complexity of operations. While immediate response is unachievable, users should be informed that the operations is on its way. On the other hand, a quick response is required for operations such as showing the description of an experiment. Due to the large size of microarray datasets, the downloading of complete datasets is usually time-consuming. Therefore, users will be recommended to execute such operations only when they are necessary. The performance of plug-in methods is the responsibility of corresponding developers.

Security --- Security is a major concern of MAMA system since the database and the client-side program is open to the public. However, the database needs to be protected from the unexpected data. Therefore, only authorized users or administrators are allowed to load or update database data.

Availability --- The client-side package will be downloaded for free and run on any computer installing Java Virtual Machine. Database and server-side program will be available as long as the server machine is running. Updating and maintenance will be scheduled occasionally. There is no backup of server and database services.

Usability --- User guides should be provided for users to learn the concepts and methods of MAMA system. Operations need user guide include, but not limited to: dataset importing, method plug-in, defining database query or analysis, and so on. The user interface should look familiar to experienced users of GUI software. Guided by instructions, these users should be able to execute most operations within an hour. The analysis methods implemented in the standard release should partly satisfy basic data analysis needs of users.

Modifiability and extensibility --- The client program of MAMA system should be highly modifiable and extensible. Major software packages, such as user interface and data analysis packages, should be encapsulated, so the updating of one package will not affect the others. As an open-source application, all source codes of MAMA system will be publicly available. Furthermore, a plug-in mechanism should be provided for other developers to add their own methods. The data model and database schema of MAMA are less flexible. They should be cautiously designed at the beginning.

Portability --- The client program should be runnable on any computer having Java Virtual Machine installed. Stored data will be exported in a standard format, such as tab-

delimited text or XML. Besides Oracle, it is not required that MAMA database can be installed to other database administration system, such as MySQL.

Maintainability --- Due to the limited human resource, the system will have a simple architecture to keep the burden of its maintenance minimal. Source codes will be updated and tested periodically. The major maintenance issue is the updating and inspection of microarray datasets.

Reusability --- No a concern for MAMA project.

## G. 5 System Requirements for Development

Software:

1. Administration of relational database (Oracle)
2. High-level programming language (Java, with Java Virtual Machine)
3. Web server and Servlet engine (Apache Tomcat)
4. Source code and project management (Eclipse)
5. XML parsing (Castor)
6. UML diagram (SmartDraw)
7. Ontology (MGED Ontology, NCI Thesaurus, … )

Hardware:

1. Enough disk space on server machine to store at least 100 microarray datasets in regular size.
2. I/O bandwidth of server machine to handle 10 or more concurrent requests.

3. Stable network connection between server and client.
4. High speed network connection of client machine (1 mbps or faster).
5. Enough internal memory (512 MB or more) of client machine to handle operations that need to load the complete expression data matrix into the memory.

# Appendix H:
# User Cases of the MAMA Client Program

The following is a list of major use cases of the MAMA client program:

- Manage Workspace: use cases operating on workspaces, the root of tree-like structure of MAMA client data model.
    - Create Workspace: create a new workspace and related files on local disk and open it as the current workspace
    - Open Workspace: open an existing workspace as the current workspace, update the data object tree on user interface
    - Delete Workspace: delete an existing workspace and related files from local disk.
    - Save Workspace: save the currently opened workspace and its contents to local disk.
    - Sort Objects: sort the data objects by specified order on user interface.
    - Delete Object from Workspace: delete a specified data object from the currently opened workspace.
- Query Database: use cases querying the database to retrieve data.
    - Query Experiments: query for microarray experiments.
    - Query Samples: query for biological samples by their features.
    - Query Sequences: query for nucleotide sequences by their names or accessions.

    - Query Database Metadata: take a snapshot about the current status of MAMA database, retrieving information such as the number of currently stored datasets.
    - Export Query Results: export query results to external files in standard format.
- Manipulate Experiment: use cases creating or customizing the microarray experiments.
    - Submit Experiment to Database: submit a new microarray dataset to the database by running a wizard or importing XML documents.
    - Create Virtual Experiment: create a experiment in the current workspace by querying database or direct submitting.
    - User-specific Data Processing: customize the contents of a virtual experiment by filtering samples or sequences, discretizing sample features, or normalizing expression measurements.
- Data Analysis: use cases related to the analysis of microarray data.
    - Create Analysis: create and run a microarray analysis by specifying statistical method to use and the inputs of the analysis.
    - Create Meta-analysis: create and run a meta-analysis based on the outputs of individual analyses, and specify the meta-analysis methods.
    - Plug in Method: Plug in a user-specific statistical method.
    - Export Results: export the results of an analysis to external files in standard format.
- Render Data: use cases specifying the functions of user interface.

- Initiate Operations with Menus: browse menus and select menu item to initiate an operation.
- Run Wizard: run a GUI wizard for multi-step operations.
- Summarize Database Status: summarize the major contents of database, such as existing array designs or experiments.
- Browse Structured Data Objects: browse the data objects in a folder-like structure.
- List Details of Data Objects: render the details of a data objects, such as the samples, sequences, or expression measurements of an experiment.

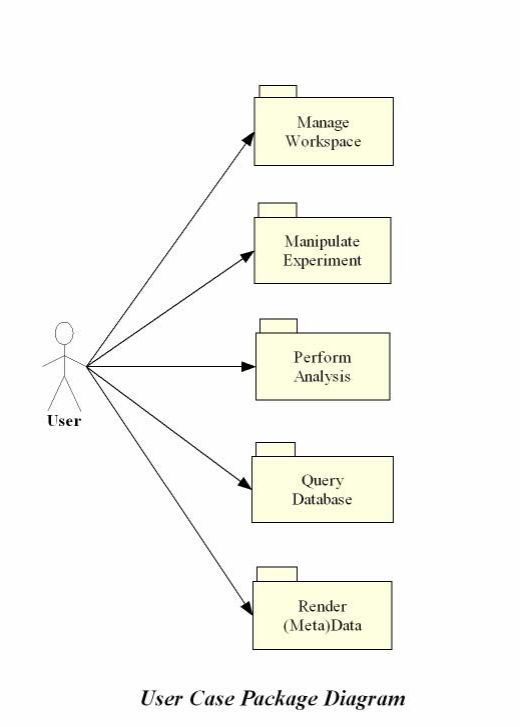



**Figure 1 Use Cases Packages**

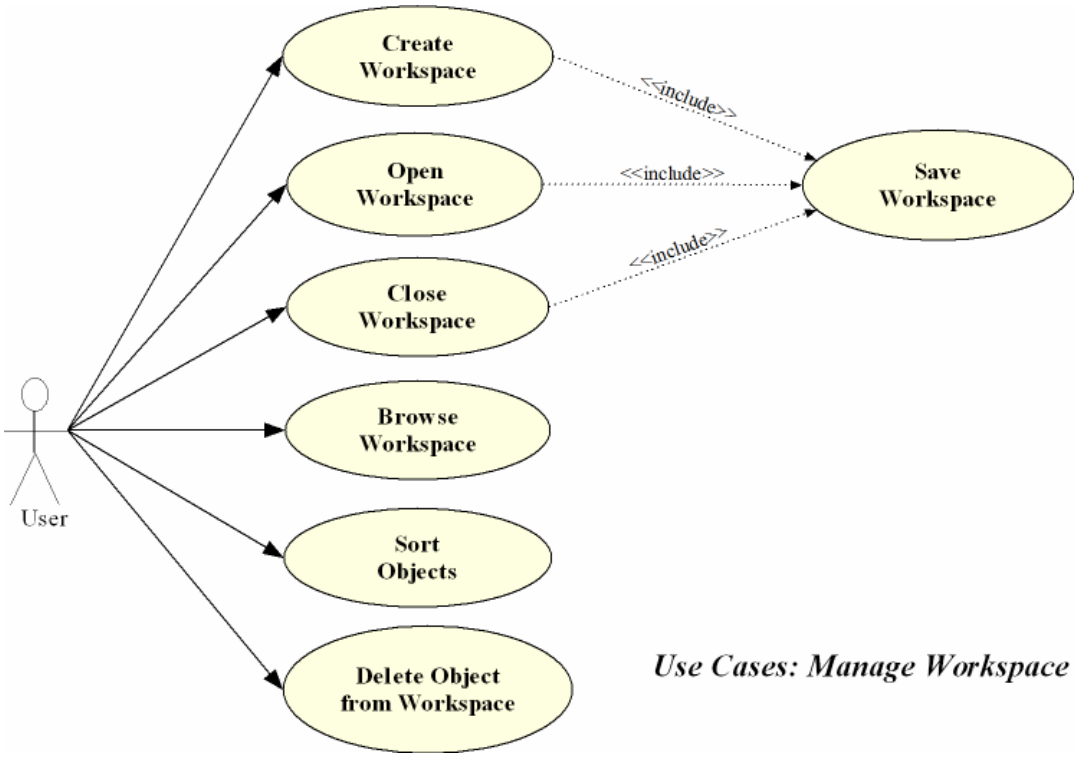



**Figure 2 Use Cases about Workspace**

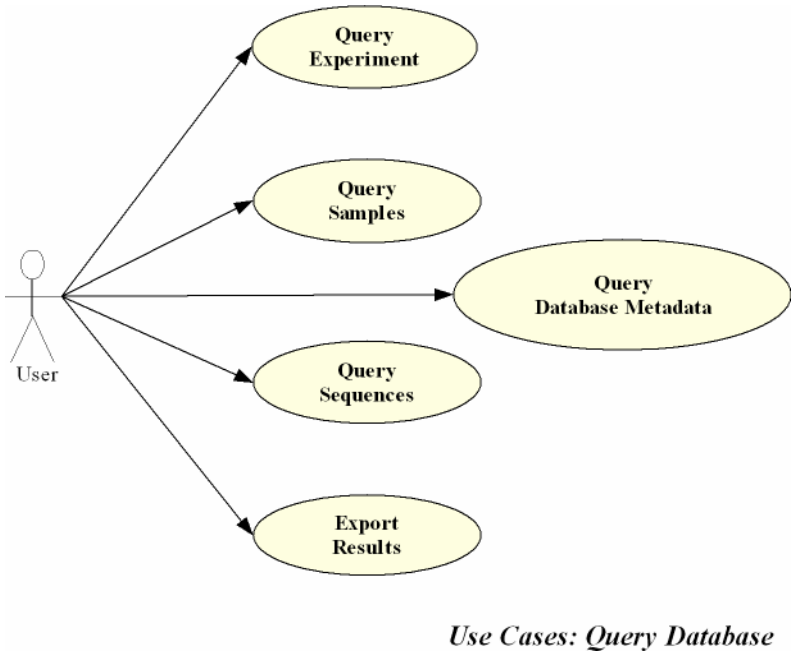



**Figure 3 Use Cases about Database Query**

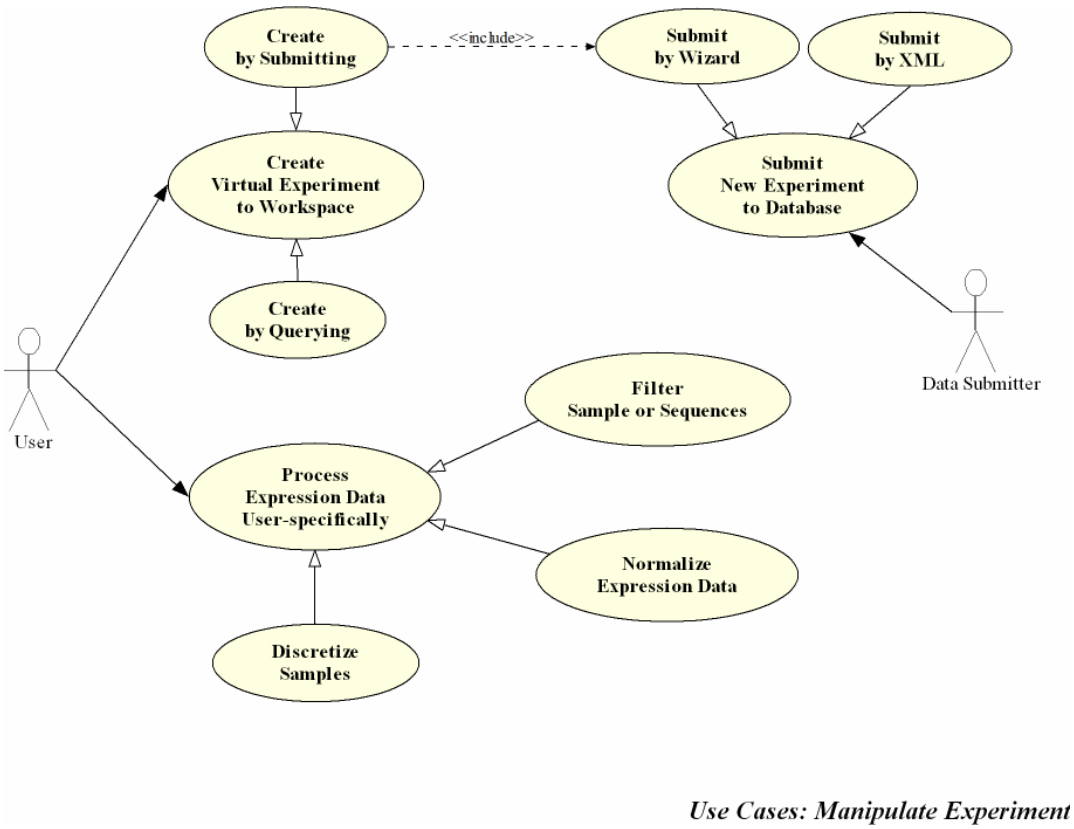



**Figure 4 Use Cases about Microarray Experiment**

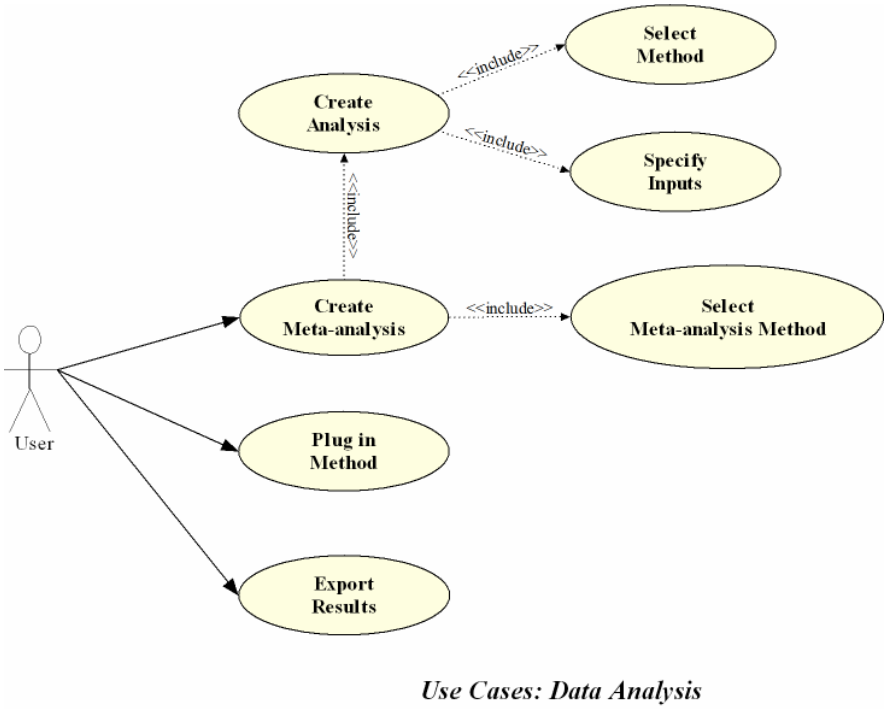



**Figure 5 Use Cases about Data Analysis**

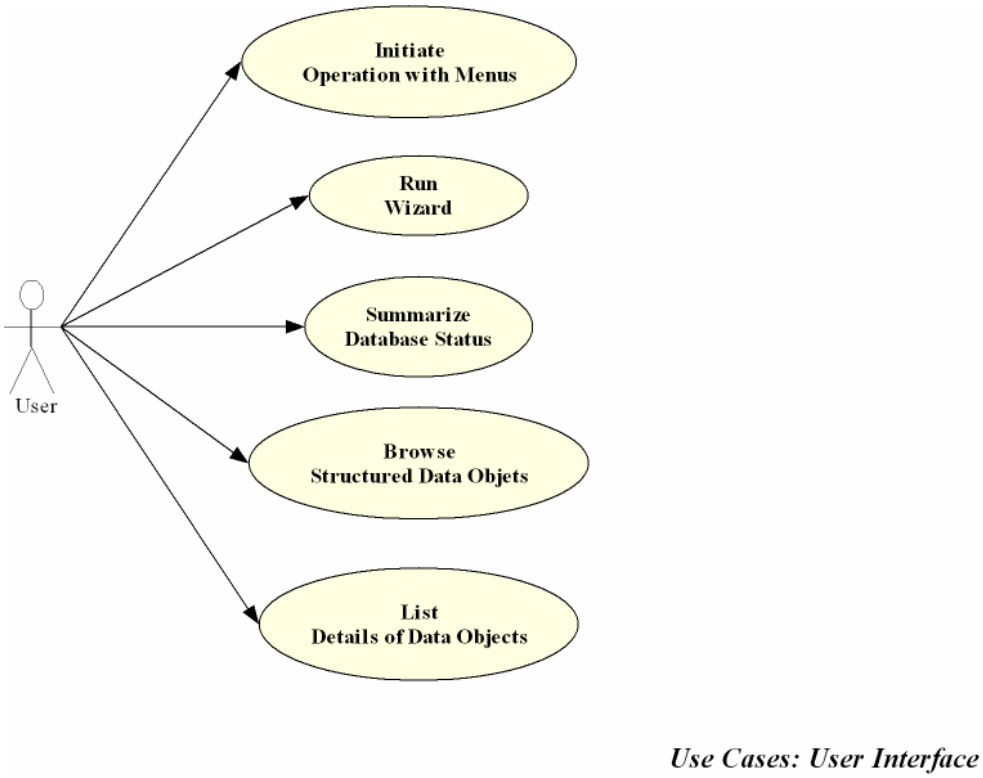



**Figure 6 Use Cases about User Interface**

The following tables give the detailed description about the use cases of Workspace package:

| **Use Case Name** | **Create Workspace** | |
|---|---|---|
| **Author** | **Zhe Zhang** | |
| **Date** | **2004/11/16** | |
| **Objective** | Create a new, empty workspace and open it. If there already is a workspace opened, close it | |
| **Actor** | User, System | |
| **Level** | Primary | |
| **Trigger** | User decides to create a new workspace | |
| **Included Use Case** | <<save workspace>> | |
| **Extended Use Case** | | |
| **Frequency** | Intermediate | |
| **Pre-condition** | • Client program is running | |
| **Post-condition** | • A new, empty workspace is created and opened in the client program | |
| **Main Flow** | **Actor Action** | **System Action** |
| | 1. User clicks 'Workspace' menu, then clicks 'New' menu item | |
| | | 2. System shows a dialog box asking for the name of the new workspace |
| | 3. User specifies the directory of the workspace and names it, then clicks 'Create' | |
| | | 4. System creates the new workspace object and opens it in the client program |
| | 5. User Clicks 'OK' | |
| | | 6. System terminates process |
| **Sub flows** | **Steps** | **Blanching Action** |
| | 4. There already is a currently opening workspace in the client program | 1. System prompts for what to do:<br>• Save<br>• Not save<br>• Cancel<br>2. User selects one<br>3. System responds to user's selection<br>• Save it, open the new one<br>• Not Save it, open the new one<br>• Abort creating, keep the old one<br>INCLUDE <<save workspace>> |
| | 3. User clicks 'Cancel' | System aborts process |
| **Exceptions** | **Conditions** | **Actions** |
| | 4. Redundant workspace name | System prompts for what to do:<br>• Overwrite (will replace the old one)<br>• Change name (will repeat step 3)<br>• Cancel (will abort process) |

| **Use Case Name** | **Open Workspace** | |
|---|---|---|
| **Author** | **Zhe Zhang** | |
| **Date** | **2004/11/16** | |
| **Objective** | Open a stored workspace from disk and make it the active workspace in the client program. Close the currently opening workspace. | |
| **Actor** | User, System | |
| **Level** | Primary | |
| **Trigger** | User decides to open a stored workspace. | |
| **Included Use Case** | | |
| **Extended Use Case** | | |
| **Frequency** | Intermediate | |
| **Pre-condition** | • Client program is running<br>• There is at least one workspace stored in the disk | |
| **Post-condition** | • A stored workspace is resumed in the client program | |
| **Main Flow** | **Actor Action** | **System Action** |
| | 1. User clicks 'Workspace' menu, then clicks 'Open' menu item | |
| | | 2. System shows a dialog box for user to select a workspace XML file from the disk |
| | 3. User browses the directories on the disk and clicks the file to be opened, then clicks 'open' | |
| | | 4. System validates the XML file with DTD, parses it to workspace object, and opens it in the client program |
| **Sub flows** | **Steps** | **Blanching Action** |
| | 4. There already is a opening workspace in the client program | 1. System prompts for what to do:<br>• Save<br>• Not save<br>• Cancel<br>2. User selects one<br>3. System responds to user's selection<br>• Save it, open the stored one<br>• Not Save it, open the stored one<br>• Abort opening, keep the old one<br>INCLUDE <<save workspace>> |
| | 3. User clicks 'Cancel' | System aborts process |
| **Exceptions** | **Conditions** | **Actions** |
| | 4. XML file is invalid | System shows error message and aborts process |

| **Use Case Name** | **Close Workspace** | |
|---|---|---|
| **Author** | **Zhe Zhang** | |
| **Date** | **2004/11/16** | |
| **Objective** | Close the workspace currently opened in the client program. | |
| **Actor** | User, System | |
| **Level** | Primary | |
| **Trigger** | User decides to close the opened workspace. | |
| **Included Use Case** | | |
| **Extended Use Case** | | |
| **Frequency** | intermediate | |
| **Pre-condition** | • Client program is running<br>• There is an opened workspace in the client program | |
| **Post-condition** | • There is no opened workspace in the client program | |
| **Main Flow** | **Actor Action** | **System Action** |
| | 1. User clicks 'Workspace' menu, then clicks 'Close' menu item | |
| | | 2. System asks user whether to save the workspace |
| | 3. User makes the selection about saving | |
| | | 4. System removes the workspace from the client program |
| **Sub flows** | **Steps** | **Blanching Action** |
| | 3a. User selects to save the workspace to disk | System transforms the workspace to XML file and saves it to the disk |
| | 3b. User selects not to save the workspace to disk | System closes workspace without saving it |
| | 3c. User clicks 'Cancel' | System aborts closing |
| **Exceptions** | **Conditions** | **Actions** |
| | There is no currently opened workspace | None |

| **Use Case Name** | **Save Workspace** | |
|---|---|---|
| **Author** | **Zhe Zhang** | |
| **Date** | **2004/11/16** | |
| **Objective** | Save the opened workspace to disk as an XML file | |
| **Actor** | User, System | |
| **Level** | Included | |
| **Trigger** | User needs to backup the opened workspace, or other procedures, such as close workspace, initiate it | |
| **Included Use Case** | None | |
| **Extended Use Case** | None | |
| **Frequency** | high | |
| **Pre-condition** | • Client program is running<br>• There is an opened workspace in the client program | |
| **Post-condition** | • A up-to-date version of the workspace is saved to disk | |
| **Main Flow** | **Actor Action** | **System Action** |
| | 1. User sends 'Save workspace' command | |
| | | 2. System transforms the workspace object to an XML file and saves it to disk |
| **Sub flows** | **Steps** | **Blanching Action** |
| | 2a. The XML file of this workspace already exists in the disk | Overwrite the old one |
| | 2b. There is no XML file of this workspace exists | Just write the XML file to the disk |
| | | |
| **Exceptions** | **Conditions** | **Actions** |
| | Transform to XML error | System aborts saving |

| **Use Case Name** | **Delete an Object from Workspace** | |
|---|---|---|
| **Author** | **Zhe Zhang** | |
| **Date** | **2004/11/16** | |
| **Objective** | Remove a query, experiment, analysis, or profile object from workspace. | |
| **Actor** | User, System | |
| **Level** | Included | |
| **Trigger** | An object in the workspace is not needed any more. | |
| **Included Use Case** | | |
| **Extended Use Case** | <<Browse Workspace>> | |
| **Frequency** | High | |
| **Pre-condition** | • Client program is running<br>• There is a opened workspace in the client program<br>• There at least exists one sub-class object of the opened workspace | |
| **Post-condition** | • An object is removed from the workspace | |
| **Main Flow** | **Actor Action** | **System Action** |
| | 1. User clicks to highlight an object, then right-clicks | |
| | | 2. System shows right-click menu |
| | 3. User clicks ‘delete’ | |
| | | 4. System asks for confirmation of deletion |
| | 5. User clicks ‘delete’ | |
| | | 6. System removes the object |
| **Sub flows** | **Steps** | **Blanching Action** |
| | 5a. User clicks ‘delete’ | System removes the object |
| | 5b. User clicks ‘cancel’ | System aborts deletion |
| **Exceptions** | **Conditions** | **Actions** |
| | Objects is used by other objects | System shows error message and aborts deletion |

| **Use Case Name** | **Sort Objects** | |
|---|---|---|
| **Author** | **Zhe Zhang** | |
| **Date** | **2004/11/17** | |
| **Objective** | Sort a field in the table of objects currently showed in the central panel. | |
| **Actor** | User, System | |
| **Level** | Primary | |
| **Trigger** | User decides to sort the object according to one of its attributes. | |
| **Included Use Case** | | |
| **Extended Use Case** | | |
| **Frequency** | High | |
| **Pre-condition** | • Client program is running<br>• There is an opened workspace<br>• There is list of objects showed in the central panel as a table | |
| **Post-condition** | • Search result is showed in the central panel | |
| **Main Flow** | **Actor Action** | **System Action** |
| | 1. User right-clicks a field name in the table | |
| | | 2. System shows a right-click menu |
| | 3. User clicks ‘sort’ in the menu | |
| | | 4. System sorts the rows in the table and refreshes table to show updated order of objects |
| **Sub flows** | **Steps** | **Blanching Action** |
| | 3a. Sort ascending | System sorts objects in ascending |
| | 3b. Sort descending | System sorts objects in descending |
| **Exceptions** | **Conditions** | **Actions** |

# Appendix I:

# Client-server Communication Protocol of the MAMA System

| Code | Definition |
|---|---|
| 1000 | Continue --- waiting for user to send more requests. |
| | |
| 2000 | User request has been successfully handled; the type of request is unspecified. |
| 2010 | 'None' request is handled, an 'echo' response is sent back to the client. |
| 2100 | 'Select' request has been handled successfully without specified knowledge about returned data. |
| 2110 | No entry is returned by database as the result of query specified in the 'select' request. |
| 2120 | A single entry is returned by database as the result of query specified in the 'select' request. |
| 2130 | Multiple and same type of entries are returned by database as the result of query specified in the 'select' request. |
| 2140 | Multiple and different type of entries are returned by database as the result of query specified in the 'select' request. |
| 2200 | 'Insert' request has been handled successfully, with unspecified consequence. |
| 2210 | Successfully 'insert' a single entry into a single database table. |
| 2220 | Successfully 'insert' multiple entries into a single database table. |
| 2230 | Successfully 'insert' one or multiple entries into multiple database tables. |
| 2300 | 'Delete' request has been handled successfully, with unspecified consequence. |
| 2400 | 'Update' request has been handled successfully, with unspecified consequence. |
| 2500 | 'Save' request has been handled successfully, with unspecified consequence. |
| 2510 | Successfully save data in the request into a single file. |
| 2520 | Successfully save data in the request into multiple files. |
| 2530 | Successfully save data in the request into one or multiple files within a newly created directory. |
| 2600 | 'Load' request has been handled successfully without specified knowledge about loaded data. |
| 2610 | No data are loaded and returned. |
| 2620 | Data in a single file is loaded and returned. |
| 2630 | Data from multiple file is loaded and returned. |
| 2640 | Data are loaded from a specified directory with one or multiple files. |
| | |

| | |
|---|---|
| 4000 | There exists error in the request, type of error and request is unspecified. |
| 4010 | Null request error: the received request is null. |
| 4020 | Null action error: the ‘action’ field of the request is null. |
| 4030 | Unknown action error: the ‘action’ field specified in the request cannot be recognized. |
| 4040 | Obsolete version error: received request has newer version than the one implemented on the server. |
| 4100 | ‘Select’ request includes client error due to unspecified reason. |
| 4110 | Null query error: query is not specified in the request. |
| 4120 | Unknown data type error: the query specified in the request cannot be interpreted by the server program. |
| 4130 | SQL error: error happens when database executes SQL script . |
| 4200 | ‘Insert’ request includes client error due to unspecified reason. |
| 4210 | Null request error: no data are given in the request to ‘insert’. |
| 4220 | Unknown data type error: server doesn’t know how to handle the type of data to be inserted. |
| 4230 | SQL error: error happens when database executes SQL script. |
| 4231 | Batch execution error: error happens when database executes a batch of SQL scripts. |
| 4240 | Duplicate entry error: value of ‘non-duplicate’ field of inserted entry already exists in the database. |
| 424X | X: an Integer indicates the field going wrong, sequentially number fields from 1. |
| 4250 | Missing data error: value of ‘non-null’ field is not given in the inserted entry. |
| 425X | X: an Integer indicates the field going wrong, sequentially number fields from 1. |
| 4260 | Unexpected data content error: the data to be submitted include unexpected contents. |
| 426X | X: an Integer indicates the field going wrong, sequentially number fields from 1. |
| 4500 | ‘Insert’ request includes client error due to unspecified reason. |
| 4510 | File not found error: cannot find XML mapping file in specified location. |
| 4520 | I/O error: I/O error happened during file read or write. |
| 4530 | Mapping error: error in the mapping between XML and data objects. |
| 4540 | Marshal error: error when parsing data object to XML document. |
| 4550 | Validation error: XML file failed validation. |
| | |
| 5000 | Error on the server side or in the response. |
| 5010 | Null response error: the expected response is null object. |
| 5020 | Null action error: the ‘action’ field of the response is null. |

| | |
|---|---|
| 5030 | Unknown action error: the ‘action’ field specified in the response cannot be recognized. |
| 5040 | Obsolete version error: received response has newer version than the one implemented on the client. |
| 5050 | Missed response error: expected response of a specified request not received from server. |
| 5050.X | X: identifier of the specified request. |
| 5100 | Database connection error: unspecified error happens during establishing or maintaining a connection to database |
| 5110 | Driver not found error: Fail to load database driver. |
| 5120 | Database access error |
| 5200 | Program error: Bug detected in server program. |
| | |
| 200 | Requests handled and response returned. |
| | |
| 400 | Error in the received ‘REQUESTER’ with unknown reason. |
| 410 | Unexpected data type error: the object received from client is null or not a ‘REQUESTER’. |
| 420 | No request error: no request is given. |
| 430 | User name specified in the request does not exist. |
| 440 | Password specified in the request is wrong. |
| 450 | Obsolete version error: received ‘REQUESTER’ has newer version than the one implemented on the server. |
| 460 | Client-side communication error |
| 461 | Communication timeout: client did not send requests to server within specified time scale. |
| | |
| 500 | Error happened when on the server is handling the ‘REQUESTER’ or in the ‘RESPONSER’. |
| 510 | Unexpected data type error: the object received from server is null or not a ‘RESPONSER’. |
| 520 | No response error: no response is given in the ‘RESPONSER’. |
| 550 | Obsolete version error: received ‘RESPONSER’ has newer version than the one implemented on the client. |
| 560 | Server-side communication error |
| 561 | Communication timeout: server did not send responses within specified time. Server may still finish the operation. |

# Appendix J:
# Database Schema of MAMA project

The complete MAMA database includes tables derived from MAGE object model and denormalization tables. The overall schema is too complex to be showed in one diagram and most MAGE tables are not involved in the current version of MAMA system. This appendix only demonstrates tables having data loaded into them. The general schema is broken down to smaller diagrams for the convenience of illustration. These schema diagrams are given in the ‘Database Schema’ folder within the data disc attached to this article. MAGE-derived tables have names started with ‘TT_’ are and denormalization tables have names started with ‘T_’. In the current version of MAMA system, some database tables are only used for permanent storage of source data and not accessible to the queries. These tables are drawn in black color. Tables whose data will be queried by the current version of MAMA client program are drawn in blue color.

# Appendix K:
# Specification for Pre-processing of Expression Data

The general purpose of this specification is to define a data processing guideline, which will be applied to expression data matrix of all microarray datasets before loading them into the MAMA database. It is expected that by following this guideline, all gene expression datasets permanently stored in this system will have same format, as well as similar median (or mean), scale, and distribution. These processed datasets could be considered as the basis of many user-specific operations, such as retrieving data and building virtual experiments.

MIAME specified three level of data processing: image, image quantitative output, and expression data matrix. However, images and the immediate quantitative output of images are often missing in the published experimental results. Instead, authors tend to provide their data after performing some data curating steps. For example, some authors just make ratio data available for their 2-color cDNA arrays. Furthermore, different authors treat their raw data differently according to the purpose of their studies. While there still have no widely-adopted standards about the processing of microarray data yet, we try to define a 'common sense' guideline for the curating of all microarray datasets. This guideline will utilize relatively common and straightforward processing strategies so it could be generally accepted. For example, it will prefer linear normalization rather than the non-linear ones. In later development stages, we may provide user-specific processing options by implementing more sophisticated steps, but all datasets permanently stored in the database will always be treated following this guideline. This guideline will be implemented as a data curating program. A data curator runs this program by choosing parameters according to related

metadata and description, and make sure that the guideline is fulfilled before loading data to database. We will keep tuned to the progression of MGED data transformation and normalization working group. Once standards are recommended by this group, we will adopt them soon.

The data curating steps proposed in this guideline can be classified into four categories, which are:

- Filtering, e.g. removal of low quality or non-positive data points;
- Transformation, e.g. ratio and/or log transformation;
- Within array normalization, e.g. density-dependent normalization; and
- Between array normalization, e.g. scale normalization

In practice, it is not always doable, or necessary, to apply all these steps on a specific dataset. For example, if authors have carried out ratio transformation and only publish the resulting ratio data, density-dependent normalization will be unfeasible and ratio transformation can be skipped during the curating. Some authors may publish their data in multiple levels, which is a more plausible activity according to MIAME requirements. In such instances, the 'rawest' level of data will be taken, as many common processing steps as possible can be applied to each dataset.

Currently, we only consider one-color oligonucleotide/cDNA data and two-color cDNA data in this guideline. Different processing steps will be applied to these two different types of data. In later stages, more types of expression data will be added.

## K.1 Guidelines for Pre-processing 2-color cDNA Datasets

The processing of 2-color data starts from the intensity measurements obtained from two channels if they are available. 'Rawer' data such as readings on each pixel will not be processed because they are highly dependent on the scanning equipment and image analysis software used by authors. It is assumed that the original authors had performed necessary background correction and spatial adjustment before they published their 2-color or ratio data. Up to seven steps could be applied to a 2-color cDNA dataset. Some of them are not always necessary, such as log-transformation, if they have been done the original authors. The normalization of the expression measurements is limited to each single array. Users perform cross-array normalization within the client program.

Fig.1 illustrates the following steps using an activity diagram.

1. Filtering of low quality measurements. This step will only be performed when the original authors provided a single-value index, such as a flag, to indicate the quality of measurements.
2. Filtering of non-positive values. All non-positive values should be removed before log-transformation no matter the original authors provided the measurements as intensity or ratio data. Low variance and other types of filtering will not be performed. Instead, user-specific filtering was enabled in the client program. The descriptive statistics of each gene are calculated in advanced and saved in a separate database table for the convenience of advance filtering. This step will be performed on all datasets.
3. Ratio transformation. If intensity measurements of the two colors are provided separately, their ratio is calculated. The gene expression intensity in the

samples will be divided by the gene expression intensity in the reference, no matter which color the samples are labeled with.

4. Log transformation. All ratio data will be $\log_2$-transformed. If the original authors only provided log-transformed ratio data with base=n, all data points will be adjusted by multiplying a constant $\log_2 n$. This step will be performed on all datasets.
5. Intensity-dependent linear normalization. When intensity data is available, this step is carried out to correct the dye imbalance caused by different spot intensity. Linear normalization is preferred other than non-linear ones because of its simplicity and generality. After this step, data are transformed to corresponding residual values of a linear model.
6. Global median normalization of samples. All measurements of each sample are subtracted with their median value, so they will be centered at zero. This step will be performed on all datasets.
7. Scale normalization of samples. All expression measurements of each sample are divided by a scaling factor, which indicates the variance of data in each array. For simplicity, the standard deviation of measurements is used as the scaling factor.

## K.2 Guidelines for Pre-processing 1-color cDNA Datasets

1. Filtering low quality data points. Quality filtering will only be performed when authors provide a single-value index, such as a flag, to indicate the quality of data points.

2. Global median normalization. All data points within an array are subtracted with their median value, so they will be centered at zero. This step will be performed on all datasets.
3. Log transformation. All expression data will be $\log_2$-transformed. If data provider already log-transformed data with base=n, their log ratio data will be adjusted by multiplying a constant $\log_2 n$. If provider have not log-transformed the source data, the transformation will follow: 1) If value X is greater than 1.0, transform it to $\log_2 X$; else if X is between 1 and -1, transform it to 0; else if X is less than -1, transform it to $-\log_2|X|$. This step will be performed on all datasets.
4. Scale normalization. Each data point is divided by a scaling factor, which indicates the variance of data in each array. For simplicity, the standard deviation is used as the scaling factor.

References:


1. Quackenbush, J. (2002). "Microarray data normalization and transformation." Nat Genet **32 Suppl**: 496-501.
2. Tseng, G. C., M. K. Oh, et al. (2001). "Issues in cDNA microarray analysis: quality filtering, channel normalization, models of variations and assessment of gene effects." Nucleic Acids Res **29**(12): 2549-57.
3. Park, T., S. G. Yi, et al. (2003). "Evaluation of normalization methods for microarray data." BMC Bioinformatics **4**(1): 33.
4. Kroll, T. C. and S. Wolfl (2002). "Ranking: a closer look on globalisation methods for normalisation of gene expression arrays." Nucleic Acids Res **30**(11): e50.
5. Smyth, G. K. and T. Speed (2003). "Normalization of cDNA microarray data." Methods **31**(4): 265-73.

6. Yang, Y. H., S. Dudoit, et al. (2002). "Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation." Nucleic Acids Res **30**(4): e15.

7. http://www.mged.org/Workgroups/MIAME/miame_checklist.html

8. http://genome-www5.stanford.edu/mged/normalization.html

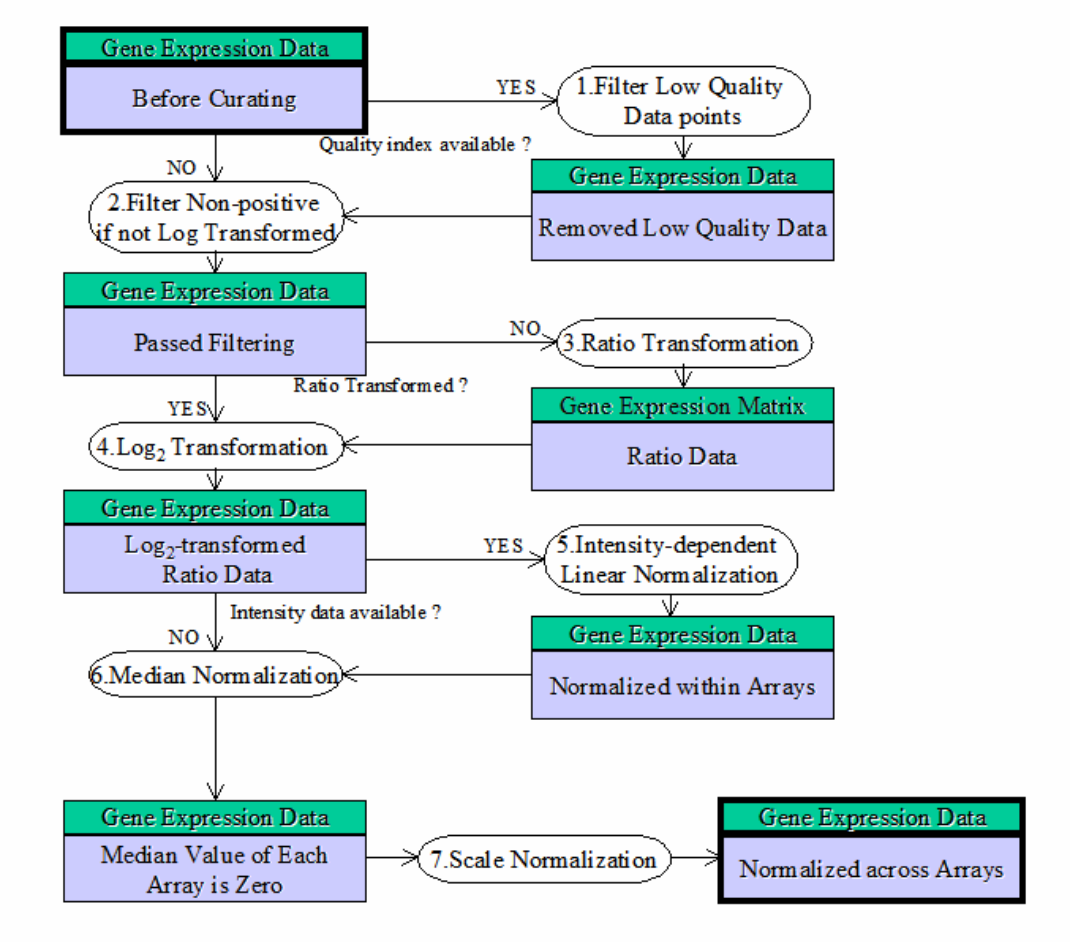



**Figure 1 Pre-processing of 2-Channel cDNA Microarray Datasets**

# Appendix L:
# Mapping between XML Elements and Java Data Objects

The Java/XML data binding is accomplished in the MAMA client program using the XML package developed by Castor project. The Castor XML achieved mapping between XML elements and Java data objects by reading a file including mapping information. The file is called ‘mapping.MAMA’ in MAMA system and can be downloaded together with the client program. In the mapping file, Java classes and their attributes are related to XML elements.

The following is a segment of the mapping file that maps Java ‘QueryResult’ class and its attributes to XML elements. The complete mapping file is given in the ‘XML Mapping’ folder within the data disc attached to this article.

```
<!-- CLASS: QueryResult -->
<class name="edu.upenn.bmif.mama.model.QueryResult" auto-
complete="false">
     <field name="runTime">
          <bind-xml node="attribute"/>
     </field>
     <field name="query"
     type="edu.upenn.bmif.mama.model.Query"
     get-method="getQuery" set-method="setQuery" >
          <bind-xml name="query" node="element" />
     </field>
     <field name="resultUnits"
     type="edu.upenn.bmif.mama.model.QueryResultUnit"
     collection="collection"
     get-method="getResultUnits" set-method="addResultUnit" >
          <bind-xml name="result-unit" node="element" />
     </field>
</class>
```

# Appendix M:
# Glossary

## Correlation and Partial Correlation

Statistical correlation is a measure about the relationship between two variables. Strength of correlation can be numerically represented by a number of correlation coefficients, among which the best known one is Pearson's product-moment correlation coefficient (r). The value of r ranges from -1.0 to 1.0. Two variables having perfect negative or positive correlation will correspondingly have an r of -1.0 or 1.0 while a value of 0.0 represents a totally unrelated relationship.

When a third variable intervenes the correlation between two variable, it can be controlled out by partial correlation analysis. Partial correlation is commonly used for controlling only one variable because the process itself will introduce extra bias into results. However, it can be iteratively used to control more variables too if sample is large enough. According to the type of intervening effect, partial correlation coefficient may be equal to or larger/smaller than corresponding correlation coefficient.

## Database Denormalization

Database denormalization is a technique used to speed up database access by introducing certain level of redundant data storage. A normalized database schema often stores logically related data in separated tables. Queries that draw data by joining several table could be slow. Denormalization improves the performance of database by tradeoff some costs. Besides data redundancy, it requires extra efforts of database designer to maintain data integrity.

Denormalization design are more error-prone in practice. However, it fits to databases whose stored data are rarely updated or deleted, such as data warehouses.

## Java

Java is an object-oriented programming language developed by Sun Microsystems. Sun's standard edition of Java is freely available, and developers are using it for various types of software from web applications to desktop programs. With a purely object-oriented architecture, Java codes are highly reusable and extensible. The running of Java programs is independent of operation systems, but only relies on installation of Java Virtual Machine (JVM). The official Java website is java.sun.com.

## Meta-Analysis

Meta-analysis is a type of statistical methods that combines the results from multiple independent studies dealing with the same research question. Therefore, it is often referred to as 'analysis of analyses'. By integrating findings of individual studies about the same hypothesis, meta-analysis methods conclude a summary overall result of hypothesis testing. Properly designed meta-analysis will take full advantage of the research efforts having been made by discovering information that cannot be obtained from individual studies. Despite of its advantages, meta-analysis also receives criticisms. For example, one of its weakness is that it has no control on the quality of individual studies, and badly design studies may cause biased results even if the meta-analysis method is faultless. Steps of a typical meta-analysis include: problem formulation, data collection and evaluation, analysis and interpretation, etc.

## Microarray

Microarray is a high-throughput technology that allows researchers to measure the expression level of genes at a genomic level. Physically, it is collection of tiny DNA spots attached to a solid surface such as glass and silicon chip to form a 2-dimensional array of probes. In a microarray experiment, these spots are hybridized to the DNA in a given cell extraction and the expression level of genes in the extraction is measured by the amount of hybridized DNA. Two types of nucleotide sequences are commonly used as probes: cDNA and oligonucleotide.

## MVC (Model-View-Controller) Architecture

MVC is a software design pattern often used for graphic user interface of applications. Its basic idea is to divide and encapsulate codes of an application into three major components: model (data model), view (user interface), and controller (business logic), so the modification of one component will have minimal influence on the others. These packages interact with each other by passing inputs/outputs without worrying about the implementation details in other packages. MVC is often used in web applications within which code modularization is preferred. Although MVC has various flavors, the typical control flow works as the following:

1. ‘View’ renders ‘model’ on user interface;
2. User interacts with ‘view’ to initiate an action;
3. The action is passed to ‘controller’;
4. The action is handled by ‘controller’, which may access ‘model’ to get data input;
5. ‘Controller’ informs ‘view’ and/or ‘model’ for proper updates at the end of the action;

6. User interface waits for next action.

## Open Source Software

Officially, open source and free software are two similar but different concepts, separately defined by Open Source Initiative (www.opensource.org) and Free Software Foundation (www.fsf.org). Developers of open source software make their source codes freely available for other developers to modify or extend under an open source license. Although the definition open source software involves complicate legal issues, it generally means free-of-charge software to ordinary users.

## Parametric vs. Non-Parametric Statistical Tests

Choice of parametric or non-parametric methods is a common decision for statistical tests. Parametric methods have relatively stricter assumptions on analyzed data, including:

1. normal distribution
2. homogeneous variances between data groups
3. continuous measures with equal intervals

Non-parametric methods do not require above assumptions, so they are computationally easier and quicker but statistically less powerful. Most parametric methods have their equivalent non-parametric ones. For example, the most common parametric and non-parametric methods for two group comparison are respectively Student's t test and Wilcoxon rank sum test.

## Servlet and Tomcat

Java servlet is a type of web application implementing the Java Servlets API. A servlet accepts HTTP requests from its clients and correspondingly responds with dynamically generated web pages. Dynamic building of web pages is necessary when contents of pages are based on user inputs, retrieved from database, or frequently updated (e.g. weather report). Servlets interact with web server via servlet container, which maps a URL to each particular servlet. Tomcat is such a servlet container developed by Apache Software Foundation. It implements the standard servlet specification from Sun Microsystems. Tomcat also includes its own HTTP server internally, so it also works as an independent web server.

## Statistical Power

In statistical hypothesis testing, 'power' of a test means the probability of rejecting the null hypothesis $H_0$ when the alternative hypothesis $H_a$ is true. Larger sample size usually leads to higher power. Quality of experimental data and used analysis methods also have their influence. Methods that need stricter assumptions usually have higher statistical power.

## TNM Classification

TNM is the most widely used staging system of malignant tumors. It is developed and maintained by International Union Against Cancer and has become a standard in clinical practice. The three letters stand for Tumor, Node, and Metastasis. This system classifies cancer patients into categories according to size of tumor, number of infected lymph nodes, and presence of distant metastasis, so proper treatment decisions can be made based on their classification. Definition of categories varies among cancer types.

## UML (Unified Modeling Language)

UML defines the standards of data modeling and documentation language for design of object-oriented software. It is used to formally specify, visualize, and document the structure and functions of software under development. Standardized diagrams play a key role in UML design. Commonly used UML diagram include use case diagram (general functions), class diagram (system structure), activity diagram (general system workflows), sequence diagram(interaction between classes and detailed workflows), and so on.

## XML (eXtensible Markup Language)

A markup language labels and format the contents of plain text to make them machine-readable. XML is a general-purposed markup language recommended by W3C consortium. Its primary purpose is to facilitate automatic data sharing across different software systems. The basic units of XML documents are called elements, which have a hierarchical structure. XML elements can be defined with XML schema or DTD (Data Type Definition). Format of XML documents must follow a few rules, so computer programs can recognize their elements and parse their contents. Many programming languages including Java provide standard libraries for parsing and writing XML documents. XML documents are nothing but labeled plain text files, which makes them independent of platforms and unaffected by changes in software. How to deal with these documents are program-specific. Despite of many advantages of XML format, its verbose structure is not very friendly for human reading and may substantially reduce program performance.

# REFERENCES


1. Greene, F.L., American Joint Committee on Cancer., and American Cancer Society., *AJCC cancer staging manual*. 6th ed. 2002, New York: Springer-Verlag. xiv, 421 p.

2. Thor, A., *A revised staging system for breast cancer.* Breast J, 2004. **10 Suppl 1**: p. S15-8.

3. Greene, F.L., *Cancer staging, prognostic factors, and our surgical challenges.* Am Surg, 2005. **71**(8): p. 615-20.

4. Michor, F., et al., *Can chromosomal instability initiate tumorigenesis?* Semin Cancer Biol, 2005. **15**(1): p. 43-9.

5. Sieber, O., K. Heinimann, and I. Tomlinson, *Genomic stability and tumorigenesis.* Semin Cancer Biol, 2005. **15**(1): p. 61-6.

6. Bonassi, S., et al., *Chromosomal aberrations and risk of cancer in humans: an epidemiologic perspective.* Cytogenet Genome Res, 2004. **104**(1-4): p. 376-82.

7. Demant, P., *The genetic factors in cancer development and their implications for cancer prevention and detection.* Radiat Res, 2005. **164**(4 Pt 2): p. 462-6.

8. Vogelstein, B. and K.W. Kinzler, *Cancer genes and the pathways they control.* Nat Med, 2004. **10**(8): p. 789-99.

9. Osborne, C., P. Wilson, and D. Tripathy, *Oncogenes and tumor suppressor genes in breast cancer: potential diagnostic and therapeutic applications.* Oncologist, 2004. **9**(4): p. 361-77.

10. Gomez-Lazaro, M., F.J. Fernandez-Gomez, and J. Jordan, *p53: twenty five years understanding the mechanism of genome protection.* J Physiol Biochem, 2004. **60**(4): p. 287-307.

11. Menard, S., et al., *Role of HER2/neu in tumor progression and therapy.* Cell Mol Life Sci, 2004. **61**(23): p. 2965-78.

12. Lleonart, M.E., et al., *Tumor heterogeneity: morphological, molecular and clinical implications.* Histol Histopathol, 2000. **15**(3): p. 881-98.

13. Sasaki, K. and S. Kawauchi, *Molecular cytogenetic analysis of solid tumors.* J Orthop Sci, 2003. **8**(3): p. 457-9.

14. Wolf, M., et al., *High-resolution analysis of gene copy number alterations in human prostate cancer using CGH on cDNA microarrays: impact of copy number on gene expression.* Neoplasia, 2004. **6**(3): p. 240-7.

15. Balogh, K., et al., *Genetic screening methods for the detection of mutations responsible for multiple endocrine neoplasia type 1.* Mol Genet Metab, 2004. **83**(1-2): p. 74-81.

16. Taylor, C.F. and G.R. Taylor, *Current and emerging techniques for diagnostic mutation detection: an overview of methods for mutation detection.* Methods Mol Med, 2004. **92**: p. 9-44.

17. Larsen, L.A., et al., *Recent developments in high-throughput mutation screening.* Pharmacogenomics, 2001. **2**(4): p. 387-99.

18. Velculescu, V.E., et al., *Serial analysis of gene expression.* Science, 1995. **270**(5235): p. 484-7.

19. Porter, D. and K. Polyak, *Cancer target discovery using SAGE.* Expert Opin Ther Targets, 2003. **7**(6): p. 759-69.

20. Watson, A., et al., *Technology for microarray analysis of gene expression.* Curr Opin Biotechnol, 1998. **9**(6): p. 609-14.

21. Lockhart, D.J. and E.A. Winzeler, *Genomics, gene expression and DNA arrays.* Nature, 2000. **405**(6788): p. 827-36.

22. Ramsay, G., *DNA chips: state-of-the art.* Nat Biotechnol, 1998. **16**(1): p. 40-4.

23. Mischel, P.S., T.F. Cloughesy, and S.F. Nelson, *DNA-microarray analysis of brain cancer: molecular classification for therapy.* Nat Rev Neurosci, 2004. **5**(10): p. 782-92.

24. Savage, K.J. and R.D. Gascoyne, *Molecular signatures of lymphoma.* Int J Hematol, 2004. **80**(5): p. 401-9.

25. Bullinger, L. and P.J. Valk, *Gene expression profiling in acute myeloid leukemia.* J Clin Oncol, 2005. **23**(26): p. 6296-305.

26. Greer, B.T. and J. Khan, *Diagnostic classification of cancer using DNA microarrays and artificial intelligence.* Ann N Y Acad Sci, 2004. **1020**: p. 49-66.

27. Yan, W. and S.S. Chen, *Mass spectrometry-based quantitative proteomic profiling.* Brief Funct Genomic Proteomic, 2005. **4**(1): p. 27-38.

28. Shi, Y., et al., *The role of liquid chromatography in proteomics.* J Chromatogr A, 2004. **1053**(1-2): p. 27-36.

29. Righetti, P.G., et al., *Critical survey of quantitative proteomics in two-dimensional electrophoretic approaches.* J Chromatogr A, 2004. **1051**(1-2): p. 3-17.

30. Alaiya, A., M. Al-Mohanna, and S. Linder, *Clinical cancer proteomics: promises and pitfalls.* J Proteome Res, 2005. **4**(4): p. 1213-22.

31. Zhou, M., T.P. Conrads, and T.D. Veenstra, *Proteomics approaches to biomarker detection.* Brief Funct Genomic Proteomic, 2005. **4**(1): p. 69-75.

32. Robert, J., et al., *Predicting drug response based on gene expression.* Crit Rev Oncol Hematol, 2004. **51**(3): p. 205-27.

33. Bertucci, F., et al., *Gene expression profiling of cancer by use of DNA arrays: how far from the clinic?* Lancet Oncol, 2001. **2**(11): p. 674-82.

34. Glanzer, J.G. and J.H. Eberwine, *Expression profiling of small cellular samples in cancer: less is more.* Br J Cancer, 2004. **90**(6): p. 1111-4.

35. Bucca, G., et al., *Gene expression profiling of human cancers.* Ann N Y Acad Sci, 2004. **1028**: p. 28-37.

36. Svrakic, N.M., et al., *Statistical approach to DNA chip analysis.* Recent Prog Horm Res, 2003. **58**: p. 75-93.

37. Cuperlovic-Culf, M., N. Belacel, and R.J. Ouellette, *Determination of tumour marker genes from gene expression data.* Drug Discov Today, 2005. **10**(6): p. 429-37.

38. Armstrong, N.J. and M.A. van de Wiel, *Microarray data analysis: from hypotheses to conclusions using gene expression data.* Cell Oncol, 2004. **26**(5-6): p. 279-90.

39. Yagi, T., et al., *Identification of a gene expression signature associated with pediatric AML prognosis.* Blood, 2003. **102**(5): p. 1849-56.

40. Tsai, C.A., Y.J. Chen, and J.J. Chen, *Testing for differentially expressed genes with microarray data.* Nucleic Acids Res, 2003. **31**(9): p. e52.

41. Pavlidis, P., *Using ANOVA for gene selection from microarray studies of the nervous system.* Methods, 2003. **31**(4): p. 282-9.

42. Yang, D., et al., *Applications of Bayesian statistical methods in microarray data analysis.* Am J Pharmacogenomics, 2004. **4**(1): p. 53-62.

43. Wang, A. and E.A. Gehan, *Gene selection for microarray data analysis using principal component analysis.* Stat Med, 2005. **24**(13): p. 2069-87.

44. Shaffer, J.P., *Multiple Hypothesis-Testing.* Annual Review of Psychology, 1995. **46**: p. 561-584.

45. Broberg, P., *A comparative review of estimates of the proportion unchanged genes and the false discovery rate.* BMC Bioinformatics, 2005. **6**: p. 199.

46. Pounds, S. and C. Cheng, *Improving false discovery rate estimation.* Bioinformatics, 2004. **20**(11): p. 1737-45.

47. Merikangas, K.R. and N. Risch, *Genomic priorities and public health.* Science, 2003. **302**(5645): p. 599-601.

48. Kudoh, K., et al., *Monitoring the expression profiles of doxorubicin-induced and doxorubicin-resistant cancer cells by cDNA microarray.* Cancer Res, 2000. **60**(15): p. 4161-6.

49. Sorlie, T., et al., *Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications.* Proc Natl Acad Sci U S A, 2001. **98**(19): p. 10869-74.

50. Pusztai, L., et al., *Molecular profiles of invasive mucinous and ductal carcinomas of the breast: a molecular case study.* Cancer Genet Cytogenet, 2003. **141**(2): p. 148-53.

51. West, M., et al., *Predicting the clinical status of human breast cancer by using gene expression profiles.* Proc Natl Acad Sci U S A, 2001. **98**(20): p. 11462-7.

52. Gruvberger, S., et al., *Estrogen receptor status in breast cancer is associated with remarkably distinct gene expression patterns.* Cancer Res, 2001. **61**(16): p. 5979-84.

53. Mimori, K., et al., *Identification of molecular markers for metastasis-related genes in primary breast cancer cells.* Clin Exp Metastasis, 2005. **22**(1): p. 59-67.

54. Mackay, A., et al., *cDNA microarray analysis of genes associated with ERBB2 (HER2/neu) overexpression in human mammary luminal epithelial cells.* Oncogene, 2003. **22**(17): p. 2680-8.

55. Maxwell, P.J., et al., *Identification of 5-fluorouracil-inducible target genes using cDNA microarray profiling.* Cancer Res, 2003. **63**(15): p. 4602-6.

56. Weldon, C.B., et al., *Identification of mitogen-activated protein kinase kinase as a chemoresistant pathway in MCF-7 cells by using gene expression microarray.* Surgery, 2002. **132**(2): p. 293-301.

57. Cianfrocca, M. and L.J. Goldstein, *Prognostic and predictive factors in early-stage breast cancer.* Oncologist, 2004. **9**(6): p. 606-16.

58. Coradini, D. and M.G. Daidone, *Biomolecular prognostic factors in breast cancer.* Curr Opin Obstet Gynecol, 2004. **16**(1): p. 49-55.

59. *Tamoxifen for early breast cancer: an overview of the randomised trials. Early Breast Cancer Trialists' Collaborative Group.* Lancet, 1998. **351**(9114): p. 1451-67.

60. van 't Veer, L.J., et al., *Gene expression profiling predicts clinical outcome of breast cancer.* Nature, 2002. **415**(6871): p. 530-6.

61. Gruvberger, S.K., et al., *Expression profiling to predict outcome in breast cancer: the influence of sample selection.* Breast Cancer Res, 2003. **5**(1): p. 23-6.

62. Gieseg, M.A., et al., *The influence of tumor size and environment on gene expression in commonly used human tumor lines.* BMC Cancer, 2004. **4**: p. 35.

63. O'Donnell A, J., et al., *Estrogen receptor-{alpha} mediates gene expression changes and growth response in ovarian cancer cells exposed to estrogen.* Endocr Relat Cancer, 2005. **12**(4): p. 851-66.

64. Wolf, F.M., *Meta-analysis : quantitative methods for research synthesis*. Sage university papers series. Quantitative applications in the social sciences ; no. 07-059. 1986, Beverly Hills: Sage Publications. 65 p.

65. Schulze, R., *Meta-analysis : a comparison of approaches*. 2004, Toronto: Hogrefe & Huber. xi, 242 p.

66. Choi, J.K., et al., *Integrative analysis of multiple gene expression profiles applied to liver cancer study.* FEBS Lett, 2004. **565**(1-3): p. 93-100.

67. Xu, L., et al., *Robust prostate cancer marker genes emerge from direct integration of inter-study microarray data.* Bioinformatics, 2005. **21**(20): p. 3905-11.

68. Ghosh, D., et al., *Statistical issues and methods for meta-analysis of microarray data: a case study in prostate cancer.* Funct Integr Genomics, 2003. **3**(4): p. 180-8.

69. Rhodes, D.R., et al., *Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression.* Proc Natl Acad Sci U S A, 2004. **101**(25): p. 9309-14.

70. Feinstein, A.R., *Principles of medical statistics*. 2002, Boca Raton, FL: Chapman & Hall/CRC. 701 p.

71. Retsky, M.W., et al., *Computer simulation of a breast cancer metastasis model.* Breast Cancer Res Treat, 1997. **45**(2): p. 193-202.

72. Demicheli, R., et al., *Proposal for a new model of breast cancer metastatic development.* Ann Oncol, 1997. **8**(11): p. 1075-80.

73. Ball, C.A., et al., *Standards for microarray data.* Science, 2002. **298**(5593): p. 539.

74. Brazma, A., et al., *Minimum information about a microarray experiment (MIAME)-toward standards for microarray data.* Nat Genet, 2001. **29**(4): p. 365-71.

75. Spellman, P.T., et al., *Design and implementation of microarray gene expression markup language (MAGE-ML).* Genome Biol, 2002. **3**(9): p. RESEARCH0046.

76. Stoeckert, C.J., Jr., H.C. Causton, and C.A. Ball, *Microarray databases: standards and ontologies.* Nat Genet, 2002. **32 Suppl**: p. 469-73.

77. Wheeler, D.L., et al., *Database resources of the National Center for Biotechnology Information.* Nucleic Acids Res, 2005. **33**(Database issue): p. D39-45.

78. Benson, D.A., et al., *GenBank.* Nucleic Acids Res, 2005. **33**(Database issue): p. D34-8.

79. Hubbard, T., et al., *Ensembl 2005.* Nucleic Acids Res, 2005. **33**(Database issue): p. D447-53.

80. Pontius, J.U., L. Wagner, and G.D. Schuler, *UniGene: a unified view of the transcriptome.*, in *The NCBI Handbook.* 2003: Bethesda (MD): National Center for Biotechnology Information.

81. Schuler, G.D., *Pieces of the puzzle: expressed sequence tags and the catalog of human genes.* J Mol Med, 1997. **75**(10): p. 694-8.

82. Manduchi, E., et al., *RAD and the RAD Study-Annotator: an approach to collection, organization and exchange of all relevant information for high-throughput gene expression studies.* Bioinformatics, 2004. **20**(4): p. 452-9.

83. Ball, C.A., et al., *The Stanford Microarray Database accommodates additional microarray platforms and data formats.* Nucleic Acids Res, 2005. **33**(Database issue): p. D580-2.

84. Gollub, J., et al., *The Stanford Microarray Database: data access and quality assessment tools.* Nucleic Acids Res, 2003. **31**(1): p. 94-6.

85. Sherlock, G., et al., *The Stanford Microarray Database.* Nucleic Acids Res, 2001. **29**(1): p. 152-5.

86. Barrett, T., et al., *NCBI GEO: mining millions of expression profiles--database and tools.* Nucleic Acids Res, 2005. **33**(Database issue): p. D562-6.

87. Parkinson, H., et al., *ArrayExpress--a public repository for microarray gene expression data at the EBI.* Nucleic Acids Res, 2005. **33**(Database issue): p. D553-5.

88. Sarkans, U., et al., *The ArrayExpress gene expression database: a software engineering and implementation perspective.* Bioinformatics, 2005. **21**(8): p. 1495-501.

89. Rhodes, D.R., et al., *ONCOMINE: a cancer microarray database and integrated data-mining platform.* Neoplasia, 2004. **6**(1): p. 1-6.

90. Rhodes, D.R. and A.M. Chinnaiyan, *Integrative analysis of the cancer transcriptome.* Nat Genet, 2005. **37 Suppl**: p. S31-7.

91. Rhodes, D.R., et al., *Mining for regulatory programs in the cancer transcriptome.* Nat Genet, 2005. **37**(6): p. 579-83.

92. Rhodes, D.R. and A.M. Chinnaiyan, *Bioinformatics strategies for translating genome-wide expression analyses into clinically useful cancer markers.* Ann N Y Acad Sci, 2004. **1020**: p. 32-40.

93. Sasik, R., E. Calvo, and J. Corbeil, *Statistical analysis of high-density oligonucleotide arrays: a multiplicative noise model.* Bioinformatics, 2002. **18**(12): p. 1633-40.

94. Schadt, E.E., et al., *Analyzing high-density oligonucleotide gene expression array data.* J Cell Biochem, 2000. **80**(2): p. 192-202.

95. Visco, A.G. and L. Yuan, *Differential gene expression in pubococcygeus muscle from patients with pelvic organ prolapse.* Am J Obstet Gynecol, 2003. **189**(1): p. 102-12.

96. Patel, S. and J. Lyons-Weiler, *caGEDA: a web application for the integrated analysis of global gene expression patterns in cancer.* Appl Bioinformatics, 2004. **3**(1): p. 49-62.

97. Argraves, G.L., et al., *ArrayQuest: a web resource for the analysis of DNA microarray data.* BMC Bioinformatics, 2005. **6**(1): p. 287.

98. Sherlock, G. and C.A. Ball, *Storage and retrieval of microarray data and open source microarray database software.* Mol Biotechnol, 2005. **30**(3): p. 239-51.

99. Heyer, L.J., et al., *MAGIC Tool: integrated microarray data analysis.* Bioinformatics, 2005. **21**(9): p. 2114-5.

100. Dudoit, S., R.C. Gentleman, and J. Quackenbush, *Open source software for the analysis of microarray data.* Biotechniques, 2003. **Suppl**: p. 45-51.

101. Gentleman, R.C., et al., *Bioconductor: open software development for computational biology and bioinformatics.* Genome Biol, 2004. **5**(10): p. R80.

102. Carey, V.J., et al., *Network structures and algorithms in Bioconductor.* Bioinformatics, 2005. **21**(1): p. 135-6.

103. Durinck, S., et al., *Importing MAGE-ML format microarray data into BioConductor.* Bioinformatics, 2004. **20**(18): p. 3641-2.

104. Wigle, D.A., et al., *Molecular profiling of non-small cell lung cancer and correlation with disease-free survival.* Cancer Res, 2002. **62**(11): p. 3005-8.

105. Garber, M.E., et al., *Diversity of gene expression in adenocarcinoma of the lung.* Proc Natl Acad Sci U S A, 2001. **98**(24): p. 13784-9.

106. Bhattacharjee, A., et al., *Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses.* Proc Natl Acad Sci U S A, 2001. **98**(24): p. 13790-5.

107. Gordon, G.J., et al., *A prognostic test for adenocarcinoma of the lung from gene expression profiling data.* Cancer Epidemiol Biomarkers Prev, 2003. **12**(9): p. 905-10.

108. Sokal, R.R. and F.J. Rohlf, *Biometry : the principles and practice of statistics in biological research.* 3rd ed. 1995, New York: W.H. Freeman. xix, 887 p.

109. Ashburner, M., et al., *Gene ontology: tool for the unification of biology. The Gene Ontology Consortium.* Nat Genet, 2000. **25**(1): p. 25-9.

110. Maglott, D., et al., *Entrez Gene: gene-centered information at NCBI.* Nucleic Acids Res, 2005. **33**(Database issue): p. D54-8.

111. Kersey, P.J., et al., *The International Protein Index: an integrated database for proteomics experiments.* Proteomics, 2004. **4**(7): p. 1985-8.

112. Shapiro, S.S., M.B. Wilk, and H.J. Chen, *A Comparative Study of Various Tests for Normality.* Journal of the American Statistical Association, 1968. **63**(324): p. 1343-&.

113. Gimenez-Abian, J.F., et al., *A topoisomerase II-dependent checkpoint in G2-phase plant cells can be bypassed by ectopic expression of mitotic cyclin B2.* Cell Cycle, 2002. **1**(3): p. 187-92.

114. Cahill, D.P., et al., *Characterization of MAD2B and other mitotic spindle checkpoint genes.* Genomics, 1999. **58**(2): p. 181-7.

115. Wu, C.W., C.W. Chi, and T.S. Huang, *Elevated level of spindle checkprotein MAD2 correlates with cellular mitotic arrest, but not with aneuploidy and clinicopathological characteristics in gastric cancer.* World J Gastroenterol, 2004. **10**(22): p. 3240-4.

116. Pangilinan, F., et al., *Mammalian BUB1 protein kinases: map positions and in vivo expression.* Genomics, 1997. **46**(3): p. 379-88.

117. Martin-Lluesma, S., V.M. Stucke, and E.A. Nigg, *Role of Hec1 in spindle checkpoint signaling and kinetochore recruitment of Mad1/Mad2.* Science, 2002. **297**(5590): p. 2267-70.

118. Wang, X., et al., *Increased levels of forkhead box M1B transcription factor in transgenic mouse hepatocytes prevent age-related proliferation defects in regenerating liver.* Proc Natl Acad Sci U S A, 2001. **98**(20): p. 11468-73.

119. Jiang, W., et al., *PRC1: a human mitotic spindle-associated CDK substrate protein required for cytokinesis.* Mol Cell, 1998. **2**(6): p. 877-85.

120. Daidone, M.G., et al., *Clinical studies of Bcl-2 and treatment benefit in breast cancer patients.* Endocr Relat Cancer, 1999. **6**(1): p. 61-8.

121. Cheung, K.L., C.R. Graves, and J.F. Robertson, *Tumour marker measurements in the diagnosis and monitoring of breast cancer.* Cancer Treat Rev, 2000. **26**(2): p. 91-102.

122. Ries LAG, et al., *SEER Cancer Statistics Review, 1975-2001*. 2004, National Cancer Institute. Bethesda, MD.

123. Booch, G., J. Rumbaugh, and I. Jacobson, *The unified modeling language user guide*. 2nd ed. 2005, Upper Saddle River, NJ: Addison-Wesley. xviii, 475 p.

124. Rumbaugh, J., I. Jacobson, and G. Booch, *The unified modeling language reference manual*. 2nd ed. The Addison-Wesley object technology series. 2005, Boston: Addison-Wesley. xx, 721 p.