
Harvard Brain Tissue Resource Center
National Brain Databank
Neuroscience Gene Expression Repository

Research on Standards and Platforms
Working Technical Report

August 11, 2003

Project Lead
Nitin Sawhney, Ph.D.

Technical Development
Tom Hickerson, Shai Sachs, Dmitry Andreyev

Abstract

The Harvard Brain Tissue Resource Center (or The Brainbank) at the McLean Hospital is one of three federally funded centers for the collection and distribution of human brain specimens for research, and the only designated acquisition center. The Brainbank seeks to establish a publicly accessible repository (The National Brain Databank) to collect and disseminate results of postmortem studies of neurological and psychiatric disorders. The National Brain Databank will primarily provide neuropathology information including gene expression data, which will be accessed and queried using a web-based interface. The project will utilize key microarray metadata standards such as MAIME and MAGE-ML and best practices employed by existing gene expression repositories like NIH's Gene Expression Omnibus (GEO) and ArrayExpress at the European Bioinformatics Institute.

The National Brain Databank initiative requires a long term perspective to develop an appropriate application platform with a scaleable and robust database while incorporating suitable microarray standards and ontologies. In this technical paper, we survey the overall lifecycle of research at the Brainbank with respect to the microarray experiments. We also review the main gene expression repositories and analytic tools as well as the emerging MAIME and MAGE-ML standards being adopted by the research community.

We propose a system architecture that allows integration of existing Affymetrix-based microarray data using the MAGE object model and interfaces, while retaining the data in its raw form. We believe the proposed repository will benefit from an architecture using the *Java J2EE* application framework and the *Oracle 9i* relational database running on a secure and high-performance *Linux*-based server. This architecture enables an open, scaleable and extensible approach towards development and deployment of the repository in conjunction with existing software tools and standards in academic settings. We believe that the basic framework outlined in this technical report should serve as a robust foundation for the evolving gene expression repository at the Brainbank.

Table of Contents

Key Recommendations	3
1 Introduction: Objectives of the National Brain Databank	4
2 Lifecycle of Research at the Brainbank.....	5
2.1 Acquisition and Curation of Brain Tissue Samples	5
2.2 Gene Expression Experiments using Microarrays	5
2.3 Analysis of Expression Data: Software Tools and Data Standards	7
2.4 Current Computing Infrastructure and Databases at the Brainbank.....	8
2.5 Basic Requirements for National Brain Databank.....	9
3 Public Gene Expression Repositories	12
4 Microarray Standards and Ontologies.....	13
4.1 Motivation for Microarray Standards	13
4.2 What is an Ontology?	14
4.3 Understanding the Role of MIAME.....	14
4.4 Understanding MAGE-OM and MAGE-ML.....	15
4.5 Software Support for MAGE	16
4.5.1 Affymetrix GDAC Exporter	16
4.5.2 MGED's MAGE-stk	16
4.5.3 Commercial Software: Rosetta Resolver	16
4.6 Data Formats used by Gene Expression Repositories	16
4.6.1 SOFT Format at GEO	16
4.6.2 MAGE Standards at ArrayExpress	17
4.6.3 GeneXML at GeneX.....	17
4.7 Historical Evolution of MAGE Standards.....	17
4.8 Proposed Use of MIAME/MAGE and Related Technologies.....	18
4.8.1 National Brain Databank Database Structure	18
4.8.2 Importing Experimental Data.....	18
4.8.3 Curating the Brainbank Data	19
4.8.4 Searching the Data	20
4.8.5 Browsing the Data.....	20
4.8.6 Exporting the Data.....	20
5 National Brain Databank: Proposed Model and Approach	21
5.1 Summary of Preliminary Requirements.....	21
5.2 Proposed Application Model and System Architecture	22
5.3 Designing the Application Platform: Adopting Java J2EE	24
5.3.1 What is J2EE?	24
5.3.2 Case Study: PhenoDB Project at Massachusetts General Hospital.....	25
5.3.3 Available Java Tools and Comparison with Other Languages.....	26
5.4 Adopting a UNIX Operating Environment for the National Brain Databank Server	28
5.5 Adopting a Relational Database: Comparison of Database Platforms	29
6 Summary of Ongoing Requirements Analysis	32
7 Conclusions.....	33
References	34

Appendix: Comparison of Databases and Security Issues

Key Recommendations

- ∅ To support the large volume of heterogeneous data generated from microarray experiments at the Brainbank, the system must provide a range of mechanisms for indexing, annotating and linking the datasets with clinical and diagnostic data on the brain tissue samples. Hence, use of standardized approaches such as the MAIME ontology is important along with a robust and scalable database.
- ∅ To ensure standardized submission, export and exchange with other gene expression repositories, the system should support the MAGE-OM object model and the XML-based MAGE-ML data exchange standards. These standards are increasingly being adopted by many databases and software tools.
- ∅ While the MAGE standards are becoming popular, many existing databases and analytic tools are only now beginning to adapt to these standards. Hence, for the foreseeable future the Brainbank must continue to provide gene expression data in their native formats to enable analysis by current software. The system must export data using MAGE while providing access to raw data files stored in the server.
- ∅ To maintain the high standards for archiving and disseminating data to the neuroscience community, the Brainbank must carefully curate data submitted from internal experiments and external investigators. Hence software tools and workflows should be provided to annotate, validate, cross-reference, and map data to the internal representation. These data submission and curation mechanisms should be MAIME compliant and can be adapted from existing software tools.
- ∅ Similar to existing gene expression repositories, the National Brain Databank must provide adequate tools for querying the diagnostic and gene expression data along a number of searchable parameters. This requires that the experimental data be submitted using MAIME compliant processes as well as indexing the raw data and clinical reports to extract relevant keywords and terms for extensive queries.
- ∅ To allow data to be usable it must be referenced to standardized Gene sequences in GenBank and linked to relevant publications in online resources such as PubMed. The system must support mechanisms to cross-link and reference these online sources using a combination of manual and automated methods.
- ∅ Since the brain samples collected and gene expression data generated are based on patient profiles and the online repository is designed to be a publicly accessible resource, data must be selectively disseminated to comply with HIPAA guidelines. Hence, the system should support user authentication mechanisms, a range of user roles and privileges for certain datasets and files, while enforcing adequate security measures in a robust and secure database.
- ∅ Extracting and archiving gene expression data in the online repository requires acquiring data from specialized software like Affymetrix using export tools like GDAC and other utilities for converting content to MAIME and MAGE-ML-based formats. The system must support extensible interfaces and APIs to allow integration with such tools. It is important to use nonproprietary platforms, open standards and methodologies in the design of the system architecture.
- ∅ The deployment architecture for the National Brain Databank must ensure long term scalability, robustness, performance, extensibility and interoperability with other systems and platforms. We propose a system architecture using the *Java J2EE* application framework and the *Oracle 9i* relational database running on a secure and high-performance *Linux*-based server. We believe this architecture provides the most secure and extensible foundation in the long term for deploying a public gene expression repository.

1 Introduction: Objectives of the National Brain Databank

*The Harvard Brain Tissue Resource Center*¹ (or *The Brainbank*) directed by Dr. Francine M. Benes at the McLean Hospital is one of three federally funded centers for the collection and distribution of human brain specimens for research, and the only designated acquisition center. The center's brain tissue provides a critical resource for scientists worldwide to assist in their investigations into the functioning of the nervous system and the evolution of many psychiatric diseases.

The Brainbank seeks to establish a publicly accessible repository (*The National Brain Databank*) to collect and disseminate results of postmortem studies of neurological and psychiatric disorders. For this project, *Akaza Research*² has been contracted to conduct research, design and development of the public gene expression repository for the Brainbank's National Brain Databank. Akaza Research is an informatics consulting firm based in Cambridge, MA that provides its academic and nonprofit clients with open and customized solutions to facilitate public research in the life sciences. The National Brain Databank will primarily provide neuropathology information including gene expression data along with anonymous demographics, which will be accessed and queried using a web-based interface. While general information will be publicly available, authorized researchers will have access to detailed results and export data into relevant standardized formats. As the system evolves, distributed researchers will have the ability to upload their own results using a specified metadata format, pending a process of approval and curation from administrators at the National Brain Databank.

The project will utilize key microarray metadata standards such as *MAIME* and *MAGE-ML*³ and best practices employed by existing gene expression repositories like *NIH's Gene Expression Omnibus (GEO)*⁴ and *ArrayExpress* at the *European Bioinformatics Institute*. Akaza is conducting requirements analysis to identify the core specifications of the system over several phases of software releases that address the near-term needs and long term vision of the National Brain Databank. This research and analysis effort conducted in conjunction with the Brainbank will be distilled into technical papers (such as this one) and formal specifications. Based on feedback from the Brainbank, Akaza will commence on the design and development of the system's first release which will include a project website, implementing the new database schema and data migration from existing *MS Access* and *MS SQL Server* databases at the Brainbank, as well as the deployment of the core *Java J2EE* based web-application framework for the online repository.

A key aspect of the National Brain Databank project includes specification and design of appropriate metadata formats and related import/export mechanisms. In addition, several workflow processes will be implemented to provide administrators with mechanisms for selective authorization of users, data import/depositing and curation/administration of the repository. The Brainbank eventually intends to support the neuroscience research community by expanding the scope of neuropathology information available to include SNP and proteomics data, while providing additional online tools for advanced search and cross-indexing, and supporting the ability to exchange relevant data with other online repositories. As the system is deployed, Akaza will continue to conduct ongoing evaluation, documentation, training and testing with lead users and administrators for iterative refinement of the system to ensure a useful and robust repository for the neuroscience research community.

This working technical paper, based on preliminary requirements gathering and background research, summarizes the key goals of the National Brain Databank, the process of research at the Brainbank, existing gene expression repositories and metadata standards as well as relevant software tools and databases. The paper proposes a high-level implementation approach for the National Brain Databank's online gene expression repository including the conceptual database model and application framework, rationale for adopting Java J2EE, Oracle and Linux as the basis for the system and outlines the ongoing requirements analysis work. Based on review and feedback from Brainbank, key decisions and tradeoffs indicated here will be resolved to finalize the key requirements and specifications towards development of the first system release of the National Brain Databank.

¹ <http://www.brainbank.mclean.org>

² <http://www.akazaresearch.com>

³ <http://www.mged.org/>

⁴ <http://www.ncbi.nlm.nih.gov/geo/>

2 Lifecycle of Research at the Brainbank

The Harvard Brain Tissue Resource Center (the Brainbank) was established at McLean Hospital as a centralized, federally funded resource for the collection and distribution of human brain specimens for research. As a designated "NIH National Resource", the Brainbank provides a vital public service by collecting and disseminating postmortem brain tissue samples to the neuroscience research community (at no charge). These brain tissues are typically related to neurological disorders including Huntington's, Parkinson's and Alzheimer's, psychiatric disorders like schizophrenia or manic-depression (bipolar disorder), as well as normal control specimens which are essential for comparative work. Collectively, these specimens are used for a wide variety of applications, including receptor binding, immunocytochemistry, in situ hybridization, virus detection, polymerase chain reaction (PCR), DNA sequencing, mRNA isolation, and a broad range of neurochemical assays.

2.1 Acquisition and Curation of Brain Tissue Samples

Having been established for over 20 years, the Brainbank has created a strong reputation as a NIH National Resource for brain tissue collection, archiving and dissemination to aid neuroscience research. To maintain this high standard, the Brainbank takes very meticulous care in receiving, documenting, caring for, and collecting background data for its cases. Samples are examined by neuropathologists and extensive case histories and family interviews are performed wherever possible, given privacy and practical limitations.

There are currently over 5800 brains stored in the Brainbank. Previously, brain tissue samples for Huntington's, Parkinson's, and Alzheimer's disease were collected, whereas now the Brainbank additionally collects samples from patients with psychiatric disorders such as schizophrenia or manic-depression as well as normal control tissue. The Brainbank also houses private collections of brain tissue samples for the Tourette Syndrome Association (TSA), which are managed by the organization. Over the years, the Brainbank has compiled a representative brain tissue sample for research called the "McLean 66" cohort⁵ (with samples from about 66-67 brains) includes roughly equal numbers of Schizophrenic, Bipolar (hardest to obtain), and control cases. Gene expression data is now being derived from this set and will be included in the online repository initially.

The Brainbank's website currently provides password-based access to an anonymized catalog of brain tissue samples, with demographic information, diagnosis information, some neuropathological and clinical information, and related images. Investigators can browse and query the database and request additional demographic information as well as the actual samples from the Brainbank. Requests for samples are handled by an independent committee that provides a recommendation to the Brainbank, before it can supply these tissue samples to the investigators. Currently the Brainbank supplies nearly 100 investigators with about 4000 samples every year.

2.2 Gene Expression Experiments using Microarrays

In addition to providing brain tissue samples with the relevant patient demographic information, the Brainbank is currently extracting gene expression levels from thousands of DNA samples of its tissue specimens. Over the last 2 years the Brainbank has expanded its capability to extract gene expression data using newly acquired microarray technologies⁶ primarily from *Affymetrix*, including *GeneChip@* microarrays. Previously all gene expression experiments were contracted out to external labs; however the results were neither consistent nor of high quality. Hence the decision was made to bring this capability in-house.

Affymetrix offers high-density microarrays for human, mouse and rat genomes. These arrays are clustered into sets of GeneChips containing probe pairs for up to 12,000 transcripts. For example, the Human U133 Genome Set of more than 39,000 transcripts is divided over two GeneChips labeled A (composed of known genes) and B (composed of express sequence tag or EST⁷ with unknown function). Affymetrix matching uses 25 base pair (bp) probes⁸ affixed to known regions on a DNA chip, which has between 8,900 and 33,000 probes. The Microarray scanner uses lasers to detect DNA stained with fluorescence, to help analyze binding of complementary

⁵ This previously originated as the "McLean 60" cohort sample, which has since been slightly expanded to include additional brain samples.

⁶ Tutorial on microarrays: <http://www.ncbi.nlm.nih.gov/About/primer/microarrays.html>

⁷ Express sequence tag (EST) is a single-pass sequence from either end of a cDNA clone; approximately 200 to 600 base pairs in length.

⁸ A labeled, single-stranded DNA or RNA molecule of specific base sequence, that is used to detect the complementary base sequence by hybridization. At Affymetrix, probe refers to unlabeled oligonucleotides synthesized on a GeneChip probe array.

cDNA/RNA sequences from the tissue sample. Each chip costs about \$600 in materials (including reagents) to prepare and run, and takes about a week to prepare (as part of a batch).

Researchers at the Brainbank create 5-7 gene expression profiles for each case, corresponding to the various brain regions that need to be studied. A gene expression experiment is rarely repeated for the same tissue sample to create another profile, unless the first one yields poor quality data. For example, over-washing and straining of the tissue samples in preparation for gene expression experiments can yield uniformly white images which are not useful for analysis. Hybridization quality is verified using background calculations in the report files generated, particularly examining the 3'/5' signals at housekeeping genes (values of 2 are considered good).

Before the Brainbank had acquired in-house capacity to conduct microarray experiments, it had provided the RNA solutions for the McLean 60 cohort study to a commercial firm, *Psychiatric Genomics*⁹ in Maryland to allow them to replicate these experiments using their own approach and unique procedures. This data may be provided to the Brainbank in the future, and hence it must be archived as a replicated data set accordingly. Experimental replicates may be assigned the same or different accession number.

Gene expression experiments can generate nearly 72 MB of data for a just a single typical array according to Affymetrix¹⁰, hence the storage and management of such data becomes a crucial task. Each sample hybridized to an Affymetrix GeneChip generates five Absolute Analysis files:

1. **EXP:** The experimental file (in ASCII text) stores laboratory information, for each array including the experiment name, sample, and the type of GeneChip array being used.
2. **DAT:** The data file contains the raw image of the scanned GeneChip array, corresponding to the raw hybridization data. These data are not processed, and no scaling factors or normalization is embedded. (40-70 MB)
3. **CEL:** The cell intensity file assigns X, Y coordinates to each cell on the array and calculates the average intensity of each cell. This file can be used to re-analyze data with different expression algorithm parameters. This file provides a normalized image. (the ASCII/Excel file is around 10-12 MB)
4. **CHP:** The chip file is generated using information in the CEL file to determine the presence or absence of each transcript and its relative expression level. (Binary file around 7 MB for Rats and 14MB for Humans)
5. **RPT:** The report file (in ASCII text) provides quick access to the quality control information for each hybridization, including noise, scale factor, target values, percent present, absent or marginal and average signal values, and housekeeping controls such as Sig(3'/5').

The report file is often examined first after running an experiment to ensure the quality of results and then the image files are used to check for any artifacts. Affymetrix software uses the EXP file together with the DAT file to process the raw data in order to generate data analysis files. The chip file is primarily used for statistical analysis. Although the EXP, DAT, CEL, CEL, CHP and RPT files can only be read using Affymetrix software, the quantitative and qualitative expression values in each CHP file can be exported as text (tab delimited) files. The DAT and CHP image files can be saved in TIFF format and later converted to JPEG for easy viewing. Optionally a mask file can also be generated to provide additional information on the microarray chip quality.

The Affymetrix Absolute Analysis text files contain a row for each transcript represented in the microarray and columns of the raw expression data for that transcript (indicating mRNA expression levels). The Affymetrix platform contains multiple pairs of perfect match and mismatch oligonucleotides¹¹ for each transcript examined. The software uses the pattern and intensity of hybridization to these oligos to calculate a relative expression value for each transcript (referred to as 'Signal' in version 5.0 Microarray Suite and 'Average Difference' in previous

⁹ <http://www.psygenomics.com>

¹⁰ http://www.affymetrix.com/technology/data_analysis/

¹¹ A short length of single-stranded nucleotides; used as probes on GeneChip® arrays.

software versions).¹² The algorithms also determine whether each transcript was detected during the hybridization. This qualitative information is reported as a “Present”, “Absent” or “Marginal”.

Each Affymetrix microarray contains thousands of different oligonucleotide probes. The sequences of these probes are available at the Affymetrix NetAffx¹³ website. It provides background/annotation info on the Affymetrix probes (based on probe ID) and also maps relationships between Affymetrix microarray chip probe IDs with that of repositories like GenBank. Currently, GenBank and dChip do not read Affymetrix IDs. Generating and using these IDs from the NetAffx website is somewhat confusing as they are not always corresponding and the relationships between them can often be many to 1, 1 to many, or many to many.

Affymetrix software includes MicroSuite for cataloging microarray data, the MicroDB database, and Data Mining tools which perform statistical tests and run on MicroDB. The *Affymetrix Analysis Data Model*¹⁴ (AADM) is the relational database schema provided along with a set of Application Programming Interfaces (API) implemented as views to provide access to data stored in Affymetrix-based local gene expression databases. While the raw microarray gene expression data may be stored in an internal database, the results are valuable for the neuroscience researchers if the data is shared along with the relevant experimental metadata and demographic details. Hence it is important to consider standards and ontologies for sharing microarray data among databases and analytic tools used by the research community.

2.3 Analysis of Expression Data: Software Tools and Data Standards

A number of software tools are used for analysis of gene expression data generated by Microarray experiments. In addition to Affymetrix’s own Data Mining Tool (DMT) and a number of proprietary commercial tools, several freely available tools are used within the research community including dChip, BioConductor and GeneCluster.

Affymetrix provides the *Data Mining Tool (DMT) v3.0*¹⁵ to allow filtering and sorting of expression results from microarray experiments, perform cluster and matrix analysis as well as annotate genes (manually or from the NetAffx website). DMT software runs on Windows NT and allows multiple queries to be performed in multiple GeneChip experiments simultaneously. To load data, one must register and select the MicroDB database to query and view the CHP files generated by Affymetrix. These can then be filtered to perform relevant analysis. Despite having been developed for Affymetrix users, the software interface does not appear to be intuitive, and many of these features have now been incorporated in publicly available analysis tools.¹⁶

The *DNA Chip Analyzer (or dChip)*¹⁷ is the most commonly used microarray analysis software, particularly utilized at the Brainbank. It was developed by Dr. Cheng Li (2003) at the Harvard School of Public Health and is freely available from Harvard. dChip requires the CDF chip file and the CEL files for conducting analysis. The software can normalize the data, export expression values, filter genes, and perform hierarchical clustering or compare genes between groups of samples. The authors of dChip encourage researchers to make their gene expression results available publicly for analysis by others:¹⁸

“We encourage researchers who generate Affymetrix data to also put the CEL or DAT files available with the paper. This will enhance the efforts of improving on the low-level analysis of Affymetrix microarray such as feature extraction, normalization and expression indexes, as well as ease the data-sharing and cross-reference among researchers since CEL level files can be pooled to analyze in a more controlled manner.

CEL files have text format and contain summarized probe-level (PM, MM) data of Affymetrix array. dChip software uses the raw CEL files. If CEL files are stored in a central database system (containing the raw CEL files or directory links to CEL files), such a function would be convenient (as implemented in the Whitehead Xchip database): users query the database through web interface for their experiments, and request the raw CEL files to be stored temporarily on a ftp site for downloading.”

¹² <http://www.expressionanalysis.com/documents/tech/Tech%20Note%20-%20Data%20Deliverables.pdf>

¹³ <http://www.affymetrix.com/analysis/index.affx>

¹⁴ <http://www.affymetrix.com/support/developer/>

¹⁵ <http://www.affymetrix.com/products/software/specific/dmt.affx>

¹⁶ Manual on Affymetrix Data Mining Tool compiled by Bob Burke at the Brainbank, Summer 2003.

¹⁷ <http://www.biostat.harvard.edu/complab/dchip/>

¹⁸ <http://www.biostat.harvard.edu/complab/dchip/public%20data.htm>

*BioConductor*¹⁹ is collaborative open source software developed by researchers at the Dana Farber Cancer Institute and the Harvard Medical School/Harvard School of Public Health. It provides a range of tools for statistical and graphical methods for analysis of genomic data and facilitates integration of biological metadata from PubMed and LocusLink. It is based on the “R” statistical programming language. The system handles Affymetrix data by allowing users to provide CEL files, as well as phenotypic and MAIME information through graphical widgets for data entry.

*GeneCluster 2.0*²⁰ is a Java-based software tool developed by Whitehead Institute/MIT Center for Genome Research (WICGR). GeneCluster allows data analysis using supervised classification such as K nearest neighbor, gene selection and permutation tests. GeneCluster supports 2 data formats – the WICGR RES file format (*.res) and the GCT (Gene Cluster Text) file format (*.gct). The main difference between the two file formats is the RES file format contains labels for each gene’s absent (A) versus present (P) calls as generated by Affymetrix’s GeneChip software (which are currently ignored by GeneCluster). Data files for use in GeneCluster can be created automatically by a special tool such as WICGR’s Res File Creation Tool or manually by standard tools such as Microsoft Excel and text editors.

To support data exchange with a range of analytic tools, the online repository for the National Brain Databank must provide the CHIP (for Affymetrix DMT), CDF and most importantly the CEL files in raw form for downloading. In addition, any report and experiment files may also be desired by some researchers to gain confidence in the experiments, while experimental metadata in accordance with MAIME will be useful for analysis as well. All files generated by Affymetrix can be placed in a secure directory within the server and referenced in the sample metadata, such that they can be easily accessed if the online user has appropriate privileges. In the future many analytic tools will begin to support MAIME metadata and microarray data import/export in MAGE-ML formats, such as *GeneSpring*²¹ and *GenePix*²².

2.4 Current Computing Infrastructure and Databases at the Brainbank

The Brainbank currently houses its databases in 2 main servers (Brain Servers 1 and 2) while a third server is being deployed for the National Brain Databank and an additional machine will be provided for development.

Clinical Server (or *Brainserver-1*) hosts the primary brain tissue and clinical data. As it contains the initial unanonymized patient data (Brains DB), it maintains restricted access in compliance with HIPAA guidelines. The server configuration is a HP Proliant ML370 G2 with 1 GHz processor, 256 MB RAM and 37.8 GB storage, RAID5 w/ (6) 9.1 GB removable hard drives. It runs on Windows NT 4.00.1381 with MS SQL Server 7.00.839 and MS Access databases. Clinical demographic and diagnostic data for brain samples are archived on these databases. It also includes brain tissue information and freezer inventory as well as neuropathology reports. This server is isolated from other machines on the network to maintain security of sensitive data.

Public Web Server (or *Brainserver-2*) hosts the publicly accessible website for the Brainbank²³ and the Harvard Image Database v1.00²⁴ which allows restricted access to query the anonymized data on brain tissue samples. The server configuration is a HP Proliant ML370 G3 with 2.4 GHz processor, 1.5 GB RAM and 90.2 GB storage, RAID5 w/ (6) 18.2 GB removable hard drives. It runs on Windows NT 4.00.1381, IIS Server with MS SQL Server 7.00.839 and *Webhunter* v4.0 databases. The Webhunter is a database product developed by ADS Image, Inc.²⁵ which is used for querying and indexing brain tissue images stored in SQL Server (previously in Access). The database (Anonymous Brains) contains anonymized brain tissue and clinical data, which is bulk imported manually using SQL Server scripts from the databases in the clinical server.

National Brain Databank (*Brainserver-3* or *National-DB*) will host the public gene expression repository for the Brainbank. Some data from other Brainbank databases will be imported into the database running on this server. The server configuration is a HP Proliant ML370 G3 with dual 2.4 GHz processors, 1.5 GB RAM and 90.2 GB

¹⁹ <http://www.bioconductor.org/>

²⁰ <http://www-genome.wi.mit.edu/cancer/software/genecluster2/gc2.html>

²¹ <http://www.silicongenetics.com/cgi/SiG.cgi/Products/GeneSpring/index.smf>

²² http://www.axon.com/GN_GenePixSoftware.html

²³ <http://www.brainbank.mclean.org>

²⁴ <http://www.brainbank.mclean.org/BrainDB/default.htm>

²⁵ <http://www.adsdb.com>

storage, RAID5 w/ (6) 18.2 GB removable hard drives. It runs on Windows NT 4.00.1381, Apache 1.3.28 web server with MS SQL Server 7.00.839.

Currently the administrators and users of these systems experience various difficulties with the reliability of the WebHunter image software, data access from MS SQL Server and regular maintenance of the NT operating system. Hence, a more robust infrastructure for the National Brain Databank would seem necessary for long-term usage. For this project, we may consider setting up Linux OS with Apache, Java J2EE and Oracle as an alternative to the current software configuration.

2.5 Basic Requirements for National Brain Databank

The primary audience for the online repository consists of psychiatric neuroscience community investigating neurological disorders such as Parkinson's, Huntington's, and Schizophrenia. While this community has many specialized needs, the National Brain Databank should serve as a national resource and hence a publicly accessible repository in-line with the goals of the NIH. As some data in the Brainbank will be confidential in nature, the approach for sharing data should be carefully considered.

The key priorities for the online repository include the ability to:

- I. Designate selective access of diagnostic and gene expression data from McLean 66 cohort sample.
- II. Query data based on gene expression profiles and other diagnostic parameters.
- III. Download gene expression data in multiple standardized formats for analysis.
- IV. Deposit and curate data from investigators who receive brain tissue from the Brainbank.

Currently, it is envisioned that the National Brain Databank will initially provide data for the McLean 66 cohort dataset along the following levels of access permissions:

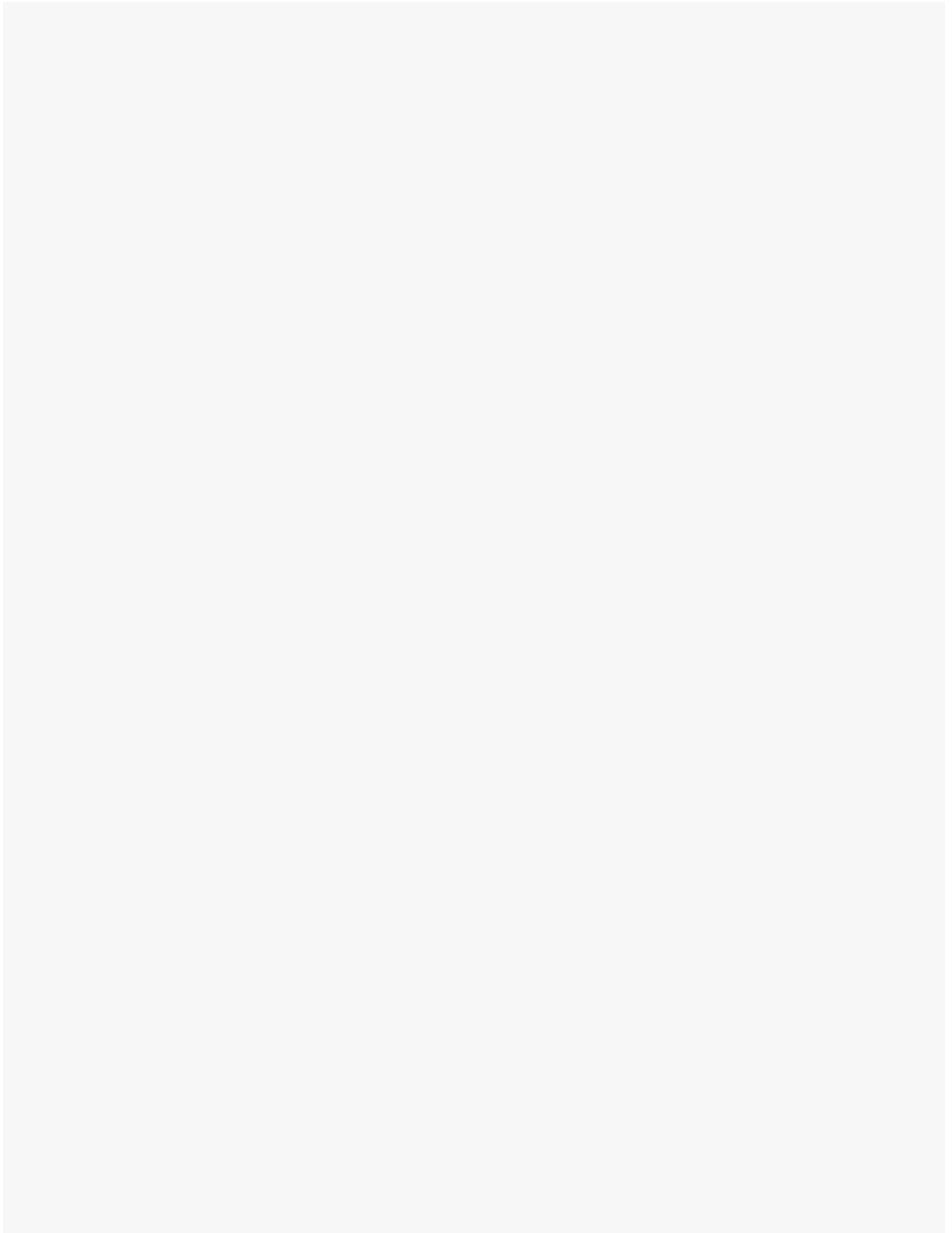
- I. **Gene Expression Profiles:** Access to general information on cases classified by brain region and symptoms, and the related microarray gene expression profiles.
- II. **Demographic:** Anonymized demographic information from patients, provided to selected investigators upon approval of submitted research application. This may include donor's age, gender, postmortem interval, cause of death, use of medication, illnesses etc.
- III. **Diagnostic:** Detailed Neuropathology or toxicology reports can be provided upon further approval.
- IV. **Case Biographies:** The donor's full biographic history is retained internally within the Brainbank and is generally not provided to external investigators in compliance with HIPAA guidelines.

For each case, 5-7 gene expression profiles could be provided corresponding to the main brain regions.

The Brainbank continues to collect data, with a goal to double the cases archived each year. In the future, the repository should be able to provide data on cohorts related to other disorders such as Parkinson's etc.

In regards to intellectual property and data sharing with nonprofit and commercial institutions, the Brainbank, in conjunction with its advisory board, made a decision to allow anyone in the neuroscience community to access their data.²⁶ This policy was set up to allow researchers in commercial firms to employ the data for experiments or drug discovery, hence serving a greater purpose than simply academic research. While there are a few caveats related to this type of data access, and usage, the Brainbank intends to require all investigators who use this data to publicly publish their findings in a timely manner. These findings may lead to improved drug therapies and could assist the overall neuroscience community. A similar policy and standards of practice should be established for the gene expression repository as well.

²⁶ Conversation with Dr. Francine Benes on July 7th, 2003.



Key Issues for Design of Online Gene Expression Repository

- § *Nature of Data and Metadata:* What kinds of data (and formats) must be archived in the repository – gene expression profiles (ASCII?), demographics, case studies (html/PDF), microarray images etc? How should this data be represented, stored and indexed in the database/system? What Metadata standards and formats should be considered? What data will be submitted in the future (like data from *in situ* hybridization experiments) and how should the system be designed to be easily extended for new data types and requirements?
- § *Quality of Data:* Due to the high standards for archiving brain samples and related data at the Brainbank, any online repository must maintain a similar level of accuracy and quality of data. This can be enabled via design of process and workflows for collecting and validating data that is archived in the online repository.
- § *Privacy of Data:* What aspects of the data should be maintained anonymously? How is it currently captured and stored? In an online repository what aspects must be selectively made available to which kinds of user roles? How should this process be monitored and administered?
- § *Submitting Data:* Should the repository provide tools for online submission of data (and related metadata) or should data be sent to Brainbank administrators for validation and curation before it is placed in the repository for public access? For investigators who will need to deposit Gene Expression data, what quality/process/format discrepancies should be considered and how would that be handled by the system?
- § *Curating the Data Repository:* While brain tissue samples and related data are carefully collected and disseminated using a process of approval and validation, to what extent should gene expression data and the related demographic information be disseminated by the repository? How should the process of submission, approval, validation and selective access be handled (both by Brainbank administrators and automated workflow processes)?
- § *Querying Data:* How would investigators likely wish to query data in the repository? Primarily by diagnosis and quality control parameters within specific groups of cases? We need to investigate this further based on current data provided, query interfaces in existing repositories and use cases from interviews of potential users.
- § *Application Platform and Database:* What should be the key criteria for selecting the appropriate database, application programming framework and servers for the online repository? Some emerging criteria include scalability, robustness, interoperability and security. A range of databases and programming platforms must be evaluated to recommend an integrated and extensible approach.

In the following sections we will consider approaches and tradeoffs in addressing these key issues, by examining existing best practices and architectures of existing gene expression repositories, emerging metadata standards and ontologies as well as large scale databases and application frameworks for the proposed National Brain Databank.

3 Public Gene Expression Repositories

Over the past few years there have been many efforts worldwide to catalog gene expression data from microarray experiments with in-house databases as well as public data repositories²⁷ [Kellam2001]; two examples include:

*Gene Expression Omnibus (GEO)*²⁸ is a gene expression and hybridization array data repository, as well as a curated, online resource for gene expression data browsing, query and retrieval [Edgar2002]. Developed by the *National Center for Biotechnology Information (NCBI)* at NIH, it claims to be the first fully public high-throughput gene expression data repository and became operational in July 2000. GEO does not intend to replace specialized gene expression databases with coherent datasets, but complement them as a tertiary data distribution hub. It utilizes three key data entities to facilitate gene expression and genomic hybridization experiments: *platforms* (list of probes), *samples* (molecules being probed) and *series* (organizing samples into meaningful datasets). It allows queries of gene profiles and datasets using various Boolean search criteria of microarray annotations and data values. Extensive indexing and linking of the data is performed to allow cross-referencing with NCBI resources such as *PubMed* and *GenBank*. Submission of new gene expression data is facilitated using an interactive web-based form or depositing of ASCII files in a predefined omnibus format (*SOFT*) via FTP for bulk submissions of large data sets. Approved submissions are given an accession number, which may be quoted in subsequent publications.

*ArrayExpress*²⁹ is a similar public gene expression initiative at the *European Bioinformatics Institute (EBI)* based in the U.K. [Brazma2003]. Its developers actively created and adopted the annotated standard MIAME and the related XML data exchange format MAGE-ML, to store generic annotated microarray data in a structured manner. It runs on an Oracle database and provides extensive online query as well as gene expression analysis tools. Submissions are accepted by the curators as *arrays*, *experiments* and *protocols*, each of which is assigned an accession number. In addition to MAGE-ML it provides a freely available online submission tool, *MIAMExpress*.

The *Stanford Microarray Database (SMD)*³⁰ is one of the first academic gene expression databases used institution-wide [Gollub2003]. SMD stores raw and normalized data from microarray experiments, as well as their corresponding image files. In addition, SMD provides interfaces for data retrieval, analysis and visualization. Data is released to the public at the researcher's discretion or upon publication. It primarily serves as a microarray research database for the Stanford microarray community. Due to its close association with one of the first groups to develop large scale microarrays, it currently contains the greatest volume of data of any academic database. SMD currently runs on an eight-processor Sun E4500 under Solaris 8 using Oracle Server Enterprise Edition version 8.1.7.3. and the Apache web server with Perl and C modules. Researchers recently expanded and mapped the data within SMD to the MAGE OM model to store MAIME Compliant data and support XML-based exchange using MAGE-ML [Hernandez-Boussard2003].

*GeneX*³¹ is an open source database and integrated tool set for managing gene expression data. It is being developed by researchers at the *National Center for Genome Resources (NCGR)* and the *Virginia Bioinformatics Institute* [Mangalam, Stewart, Zhou, *et. al.* 2001]. While the system is similar to other public repositories, it supports multiple species and gene expression technologies. It is also designed to be installed at multiple sites to enable a peer-to-peer federation of databases. It uses its own markup language, *GeneXML* for data exchange, though the developers are shifting to MAGE-ML shortly. It provides a Java-based Curaton module for uploading data and metadata and web-based tools for querying and viewing datasets. The system runs on Linux platforms using the PostgreSQL database, and the code is licensed under GNU LGPL.

While these are generalized data repositories and standards, their implementation and best practices are worth examining for our efforts to develop a public repository for the National Brain Databank. Being fairly well known in the research community, potential users may expect a similar approach or functionality for the National Brain Databank. At some point in the future the Brainbank may also wish to share/exchange data with such repositories; hence adopting the right standards and processes will be important.

²⁷ <http://www.microarray.nl/others.html>

²⁸ <http://www.ncbi.nlm.nih.gov/geo/>

²⁹ <http://www.ebi.ac.uk/arrayexpress/>

³⁰ <http://genome-www.stanford.edu/microarray/>

³¹ <http://genex.sourceforge.net/>

4 Microarray Standards and Ontologies

In the past, the exchange of microarray data among databases and online published data has been difficult due to lack of sufficient information on the microarray experiments provided in the published data and lack of unified standards, procedures and formats [Stoeckert2002]. The *Microarray Gene Expression Data (MGED) Society*³² is an international consortium of biologists, computer scientists, and data analysts (supported by various industry organizations) formed to facilitate the sharing of microarray data generated by functional genomics and proteomics experiments. MGED has defined three main components to support standardized storage and exchange of microarray data. Recently, MGED published the specification describing *MIAME*³³, *the minimal information for the annotation of a microarray experiment*, which specifies the data and contextual information to be supplied for publishing microarray gene expression datasets.

To facilitate the exchange of gene expression data, MGED developed the *Microarray Gene Expression Markup Language*³⁴ (*MAGE-ML*) data format [Spellman2002]. MAGE-ML is based on XML and can describe microarray designs, manufacturing information, experimental setup and execution information, gene expression data and data analysis results. Gene expression data from Affymetrix microarray experiments can be exported into Excel spreadsheets as well as in MAGE-ML.

MAGE-ML is rapidly being adopted as the primary standard for exchange of microarray data among leading microarray manufacturers, bioinformatics software applications and online gene expression repositories. Hence, it seems prudent to better understand and utilize the MAGE-ML standard for the Brainbank's National Brain Databank repository. Finally, the *MAGE Software Toolkit*³⁵ (*MAGE-stk*) is a collection of Open Source packages that implement the MAGE Object Model in various programming languages including Java and Perl. The MAGE-stk would be valuable for implementing microarray data import/export and exchange using MAGE-ML in the Java-based web application framework for the National Brain Databank Project.

4.1 Motivation for Microarray Standards

In the past several years, there has been a widespread and rapid increase in the volume of biological experimental data, due in large part to the proliferation of microarray experiments. This trend creates an immediate problem: experimental data must be stored, archived and retrieved frequently to support the goals of scientific research. This trend also creates a variety of opportunities for new discoveries by re-examining experimental data from a single study, or by comparing data for different studies. For these opportunities to be exploited, it is necessary for investigators from different groups to share a massive quantity of data in a systematic way. Furthermore, the data must be formatted so that it can be transferred smoothly between different computational systems, and so that each system can fully utilize the data. The challenge is to make a vast amount of complex data both "human-readable" and "computer-readable" for a diverse audience of investigators and heterogeneous computing environments.

To address this challenge, a variety of standards has emerged and is being adopted by a number of different organizations in the gene expression research community. MIAME has emerged as a standard for communicating information about microarray experiments. MIAME defines the minimal information required to allow researchers to reproduce microarray experiments and to analyze experimental data.

MIAME by itself does not fully address the challenge of computationally storing, transferring, and analyzing microarray data. If a large quantity of data from a variety of different sources is to be efficiently manipulated by a computer, it must be formatted according to a fixed set of rules that can be easily encoded in software. The MAGE data format attempts to solve this aspect of the challenge. In establishing a flexible collection of data structures, and by expressing this collection in a way that readily accommodates both manipulation in software (MAGE-OM) and transfer between data storage systems (MAGE-ML), MAGE opens the door to computationally manipulating MIAME-compliant information.

³² <http://www.mged.org/>

³³ <http://www.mged.org/Workgroups/MIAME/miame.html>

³⁴ <http://www.mged.org/Workgroups/MAGE/mage.html>

³⁵ <http://mged.sourceforge.net/software/MAGEstk.php>

The role of ontologies in addressing this challenge is also crucial. A frequent problem in communication of experimental data is the ability to express the same concepts in multiple ways. For example, the terms “mouse”, “*Mus musculus*” and “*Mus musculus musculus*” are commonly understood to refer to the same species, but it is very difficult to develop software that treats these phrases as related terms. Consequently, applications like database search on microarray data are difficult to develop unless concepts are expressed using a controlled vocabulary of commonly agreed-upon terms. Ontologies in a number of domains are being developed to address this problem.

4.2 What is an Ontology?

An ontology is a “specification of concepts and relationships between concepts” [Stoeckert2002]. Ontologies are commonly used where a variety of information is routinely passed between users and computers, and where computers need to search or otherwise manipulate the information – for example, by making inferences on that data. The simplest form of an ontology is a controlled vocabulary for some domain such as species taxonomy. More complicated ontologies specify complex data structures – for example, an ontology could specify that the age of a specimen in an experiment requires the unit of time age is measured in (e.g. “day”), the age of the specimen in those units (e.g. 7), as well as the time from which that age is measured (e.g. “post mortem”).

In particular, MIAME is defined as an ontology for microarray experiments. At a finer granularity, the information specified in a MIAME-compliant body of information (such as a MAGE-ML file) will be expected to make intensive use of ontologies, for example to name the species from which biological materials for an experiment are derived, or to describe the protocols by which those materials are treated.

4.3 Understanding the Role of MIAME

MIAME, which stands for *Minimum Information About a Microarray Experiment*, is a standard that defines the information necessary for investigators to “interpret unambiguously and potentially reproduce and verify an array based gene expression monitoring experiment.”³⁶ MIAME information is composed of two pieces: 1) information about an array design, and 2) information about a gene expression experiment design.

Information about an array design includes information about the array as a whole (platform type, surface and coating specification, etc.) as well as information about each array design element (i.e. each spot). Information about each spot on a microarray is consists of three parts:

1. *Feature* – the location on the array.
2. *Reporter* – the nucleotide sequence located at a particular feature.
3. *Composite sequence*³⁷ – a set of reporters which collectively are meant to measure expression of a particular gene.

If an investigator uses a commercially manufactured array, it is possible that the manufacturer will provide a MIAME description of the array, thereby saving the investigator the effort of drafting a description.

An experiment is meant to designate a series of hybridizations that are related in some way. Information about an experiment is consists of four parts:

1. Experimental design – including contact information for author and lab, experiment type, experimental factors, quality control steps taken, etc.
2. Samples used, extract preparation and labeling.
Sample information includes information about the source of the biological material attached to the microchip, including clinical information, which may be hyperlinked.
3. Hybridization procedures and parameters.
4. Measured data and specification of the data processing.

³⁶ http://www.mged.org/Workgroups/MIAME/miame_1.1.html

³⁷ This information applies to groups of spots, rather than a particular spot.

It is important to stress that MIAME is not a data format – it does not specify how data should be presented or organized, nor does it specify data types. Any collection of information about an experiment which contains the information specified by MIAME is said to be MIAME compliant, regardless of whether the information is plain text, tab-delimited text, or a MAGE-ML file.

4.4 Understanding MAGE-OM and MAGE-ML

The *Micro Array Gene Expression Object Model (MAGE-OM)* defines a set of classes which hold information about an experiment; the information contained in the object model is MIAME-compliant.³⁸

A *class* is a concept in computer science which defines a collection of data and operations on that data. An *object model* is a description of a collection of classes and the relationships between those classes. Objects are specific instances of classes that emerge to handle the data and functional operations of an object-oriented application like Java.

MAGE-OM is a data format for use within software applications, which makes it easy to manipulate data about an experiment in a wide variety of software development environments including Java and Perl. MAGE-ML is an XML-based data format for use in exchanging microarray data between different databases and applications. The definition type document (DTD) which defines MAGE-ML is derived automatically from the class definitions of MAGE-OM.³⁹

MAGE-OM is composed of thirteen sets of classes, or *packages*.⁴⁰

1. *Experiment* – describes a single experiment.
2. *Bioassay* – describes a single step within an experiment.
3. *ArrayDesign* – describes the design of a group of microarray chips.
4. *DesignElement* – describes a particular design element on a microarray.
5. *Biomaterial* – describes a type of biological material (such as a labeled feature extract.)
6. *Array* – describes a single microarray, which may differ from the design for that microarray as described in *ArrayDesign*.
7. *BioAssayData* – describes the experimental data.
8. *QuantitationType* – describes the way the experimental data is measured.
9. *HigherLevelAnalysis* – a set of classes providing a way to cluster together *BioAssayData*.
10. *Bioevent* – describes the kinds of events which may happen (such as feature extraction, bioassay creation, etc.)
11. *Protocol* – describes the protocol used to cause a single bioevent to occur.
12. *AuditAndSecurity* – describes the people and organizations involved in the experiment.
13. *Description* – describes other information external to the experiment, such as literature references, references to external ontologies, etc.

For example, an abbreviated diagram representing the *Experiment* package is shown below. It is important to note that the package is composed of a number of classes (indicated by rectangles) which are interrelated in complex ways (indicated by arrows). Furthermore, note that this package relies on classes within the *Bioassay* package (indicated by the gray box marked “Bioassay”).

³⁸ <http://www.mged.org/Workgroups/MAGE/introduction.html>

³⁹ A real-world analogy may be useful. MIAME is analogous to the instructions for an IRS tax return – it defines the information that must be supplied to comply with a particular standard. MAGE-OM and MAGE-ML are analogous to a completed paper and electronic tax form – they meet the requirements of the instructions, but they provide the information in different formats.

⁴⁰ <http://www.mged.org/Workgroups/MAGE/mage-om.html>

4.5 Software Support for MAGE

4.5.1 Affymetrix GDAC Exporter

The *GeneChip Data Access Components (GDAC) Exporter* is a tool which exports information from the Affymetrix system to MAGE-ML format.⁴¹ Using a fixed set of mappings, the GDAC Exporter converts a single set of data – composed of EXP, CEL and CHP files – as well as information about an Affymetrix microarray stored in a CDF file into a single MAGE-ML file.⁴²

While the software is freely available on the Affymetrix web site, it requires the Microarray Suite, which is proprietary Affymetrix analysis software; therefore, it can only be used by users and institutions that have purchased the Affymetrix system.

We expect that the GDAC Exporter will easily facilitate the conversion of existing and forthcoming Brainbank data sets into MAGE-ML files. However, we do not expect that outside investigators using the Brainbank repository will have access to an Affymetrix system, and therefore we reserve it as a convenient tool for internal use only.

4.5.2 MGED's MAGE-stk

The MAGE Software Toolkit⁴³ (MAGE-stk) is a set of code available in Java and Perl. The MAGE-stk allows programmers to easily read in a MAGE-ML file and convert it to a MAGE-OM compliant object. This object can subsequently be manipulated by a computer program for a variety of purposes including search, further statistical analysis, etc. The MAGE-stk also provides utilities which facilitate generating a MAGE-ML file from a MAGE-OM compliant object.

We expect that the code for the Brainbank repository will be written in Java, and that all operations on the experimental information contained in the repository will be performed via the MAGE-stk. Furthermore, we believe that other investigators and software developers will be able to use the MAGE-stk to manipulate data retrieved from the Brainbank repository to suit their own needs.

4.5.3 Commercial Software: Rosetta Resolver

Rosetta Resolver is a gene expression analysis tool. In the initial development of Resolver, Rosetta used an internal file format for gene expression data known as Gene Expression Markup Language (GEML). GEML was submitted by Rosetta to the Object Management Group⁴⁴ as a standard for gene expression data.⁴⁵ A subsequent collaboration with two other submitters resulted in the development of MAGE. The newest version of Rosetta Resolver is MAGE-compliant, allowing users to read in and write out MAGE-ML files as well as GEML files.⁴⁶

4.6 Data Formats used by Gene Expression Repositories

4.6.1 SOFT Format at GEO

The Gene Expression Omnibus⁴⁷ (GEO) database is a repository for data from gene expression experiments. GEO allows submissions in two modes: interactive mode, which is suitable for occasional submissions of small amounts of data; and bulk mode, which is suitable for frequent submissions of large amounts of data [Domrachev and Lash 2002]. For bulk submissions, users upload files in simple omnibus format (SOFT) to the GEO server. SOFT files are ASCII text files which contain information about the investigator and the experiment conducted.

The experimental data is organized around platforms, samples, and series:

- § A platform is a list of probes to be used in an experiment. Information about platforms includes the location (spot) of each probe, as well as the biological content of that spot.

⁴¹ http://www.affymetrix.com/support/developer/exporter/GDACExporter/Pages/GDACExporter_home.affx

⁴² http://www.affymetrix.com/support/developer/exporter/GDACExporter/Pages/GDACExporter_Mapping.affx

⁴³ <http://www.mged.org/Workgroups/MAGE/magestk.html>

⁴⁴ <http://www.omg.org/>

⁴⁵ <http://journals.iranscience.net:800/Default/www.genomicglossaries.com/content/microarrays.asp>

⁴⁶ <http://www.rosettatabio.com/products/resolver/default.htm>

⁴⁷ <http://www.ncbi.nlm.nih.gov/geo/>

- § A sample is a list of molecules which are being probed. A sample must be coupled with the platform used to probe it. Information about samples includes the platform used, and for each combination of probes in the platform and molecules in the sample, relevant abundance values such as raw signal, background signal, etc.
- § A series organizes a variety of platform/sample combinations into a single experiment.

As a development of the National Centers for Biotechnology Information (NCBI), GEO has emerged as a popular tool for publishing experimental data sets. Although the SOFT data format is considerably more simplistic than MAGE, it may be wise to allow both submission to the Brainbank repository via SOFT format, and presentation of experimental data in SOFT format to repository users. However, such a feature should be considered carefully, as it is likely that conversion between SOFT and MAGE-ML files is likely to be a difficult and time-consuming process.

4.6.2 MAGE Standards at ArrayExpress

ArrayExpress is a repository for microarray data whose goal is to “[store] well annotated data in accordance with MGED recommendations.”⁴⁸ In a manner similar to that adopted by GEO, ArrayExpress allows users to submit experimental data interactively, for occasional submissions of small amounts of data, and in bulk, for frequent submissions of large amounts of data:

- § For interactive submissions, users employ a web-based tool, MIAMExpress.⁴⁹ By guiding a user through a series of forms which query for the required information, this tool guarantees that the user’s submission is MIAME-compliant.
- § For bulk submissions, users upload MAGE-ML files, either through a web browser or via FTP. Users are expected to use the MAGE-ML validator to check that the files they wish to upload are in valid MAGE-ML format.

ArrayExpress is an open-source project; therefore, the ArrayExpress web site provides a library of information and software programs which can be used to build a MAGE-based repository.

4.6.3 GeneXML at GeneX

The goal of GeneX is to provide “an Open Source database and integrated tool set that will allow researchers to store and evaluate their gene expression data” [Mangalam, Stewart, Zhou, *et. al.* 2001]. The GeneX repository is oriented around GeneXML, a gene expression data format devised explicitly for the purpose of GeneX. Submissions to GeneX must be sent through the Curation Tool, a Java program which operates on the user’s computer and guarantees that the data submitted is Gene-XML compliant. The development of GeneX preceded the MAGE standard. At the time of writing, GeneX is in the process of adopting the MAGE format.

4.7 Historical Evolution of MAGE Standards

The MAGE standard and MIAME ontology are the result of many efforts to define standards for exchange of gene expression data. Perhaps the first attempt to define such standards was the definition of the SOFT file format for the Gene Expression Omnibus. However, the kinds of experiments that can be expressed in SOFT are very limited. Furthermore, the SOFT file format, while reasonably sensible to human readers, does not readily facilitate computational manipulation. Later attempts include GEML, a format developed for internal use by Rosetta in its Resolver and Conductor products; Gene-XML, a format developed for use in the GeneX database; and MAML, a format developed by the Microarray Gene Expression Data Society (MGED). MGED was founded by “many of the major microarray users and developers including Affymetrix, Stanford University and The European Bioinformatics Institute (EBI).”⁵⁰

The turning point for MIAME and MAGE came in November 2000, when Rosetta, the EBI, and NetGenics submitted separate proposals in response to an Object Management Group (OMG) RFP for a gene expression

⁴⁸ <http://www.ebi.ac.uk/arrayexpress/>

⁴⁹ <http://www.ebi.ac.uk/miamexpress/>

⁵⁰ <http://www.mged.org/Mission/index.html>

data format. Subsequently, the three groups collaborated to submit a single proposal, which came to be adopted as the OMG standard for gene expression data.⁵¹ MAGE was thus established as a standard with a broad base of support. MGED was charged with the task of supporting MAGE via development of software tools like the MAGE-stk, documentation, and the organization of MAGE conferences. Many software products and repositories that support MAGE – including Rosetta Resolver, Affymetrix GDAC Exporter, and the EBI ArrayExpress repository – are developed by the same community which developed the MAGE standard. However, the plans for future adoption of MAGE as a standard for GeneX⁵² and GEO,⁵³ as well as the successful adoption of MAGE by the Stanford Microarray Database,⁵⁴ suggest that the data format has gained substantial acceptance in the gene expression research community.

4.8 Proposed Use of MIAME/MAGE and Related Technologies

In light of the widespread acceptance of MAGE as a standard format for exchanging microarray data, we recommend its adoption by the Brainbank repository. Researchers will be able to submit MAGE-ML files to the repository and, where possible, MAGE-ML files will be provided for user download. However, MAGE-ML will not be the sole format supported by the repository, because many users lack adequate technical support to create or use MAGE-ML files. Therefore, data will also be available as raw text tab-delimited files similar to those available at ArrayExpress,⁵⁵ and possibly in the SOFT format in use at GEO.

The main functions of the repository will be: importing experimental data; facilitating administrative curation of submitted data; searching the data; browsing the data, whether via simple HTML or more complicated graphical interfaces; and exporting the data for user download. We suggest a MAGE-based framework and database structure for supporting these functions.

4.8.1 National Brain Databank Database Structure

In developing a database schema for the National Brain Databank repository, we can consider two main solutions. One solution envisions a bipartite structure for the database. Here one part of the database would hold MAGE-compliant data, while separate tables would encompass remaining data, including clinical data, neurological data etc. The part of the database which holds MAGE-compliant data would mirror the structure of the ArrayExpress database,⁵⁶ since this database is largely MAGE-compliant. The other part would be designed after careful consideration of the structure of the data to be stored and the operations which must be performed on it. Another approach is to develop the database schema to map all data within a MAGE structure, i.e. in addition to the microarray data other clinical data at the Brainbank would be specified within MAGE classes and related tables, as annotations, references or name-value pairs. This approach would provide maximal compliance with MAGE standards such that all data could be easily exchanged using MAGE-ML. Further investigation of the extensibility of MAGE-OM classes to handle non-Microarray data at the Brainbank will be beneficial in selecting an appropriate approach.

4.8.2 Importing Experimental Data

For internal data – that is, data produced at HBTRC on the Affymetrix system – the GDAC exporter will be used to export experimental data files in MAGE-ML format. The MAGE-ML file will then be uploaded to the repository server, either via a web-based form or via FTP. In either case, the file will need to be stored in the repository's database system. For this step the source code for MAGE loader/validator⁵⁷ will be adapted to the Brainbank's needs.

For external data – that is, data produced with HBTRC samples at other sites – several import formats could be eventually provided. Researchers could upload MAGE-ML files, which will be validated and loaded into the repository's database system using a system derived from the MAGE loader/validator. For researchers wishing to

⁵¹ <http://www.omg.org/cgi-bin/apps/doc?formal/03-02-03.pdf>.

⁵² <http://genex.sourceforge.net/>

⁵³ <http://ncbi.nlm.nih.gov/About/primer/microarrays.html>

⁵⁴ <http://genome-www5.stanford.edu/MicroArray/SMD/>

A document describing the transition to MAGE-ML is available at http://www.ebi.ac.uk/arrayexpress/SMD_to_MAGE_Mapping.doc

⁵⁵ These files include contacts, experimental data, literature citations, array design, hybridizations, biosources and experimental protocols.

See, for example, <http://www.ebi.ac.uk/arrayexpress/query/result?queryFor=experiment&eSpecies=Mus+musculus>.

⁵⁶ Scripts available at <http://www.ebi.ac.uk/arrayexpress/Implementation/creation.html>

⁵⁷ <http://www.ebi.ac.uk/arrayexpress/Implementation/index.html>

use another format, at least two other options could be considered. Researchers could upload non-MIAME compliant raw data in SOFT format, in accordance with GEO standards; or upload data in a MIAME-compliant way, using the interactive MIAMExpress tool. However, in the first release of the system we would primarily support external data submission through manual curation of files submitted via FTP and their conversion to MAGE-ML.

Several open questions are raised by this proposal:

1. How easily can the GDAC exporter be used, given the size of the existing Brainbank data set? For example, will it be possible to design a software program that will export the Affymetrix data files to MAGE-ML files in one bulk operation, rather than in many interactive operations?
2. What will be the form of the input to, and output from, the GDAC Exporter? Is the data in the existing data set sufficient to guarantee that the output from the GDAC Exporter will be valid MAGE-ML? Does HBTRC have all the files sufficient to operate the GDAC Exporter – in particular, does HBTRC have CDF files describing each of the Affymetrix chips used in the experiments described in the existing data set?
3. How will data which cannot be stored in MAGE-ML files be incorporated into the system? Such data includes the raw CHP files, related brain images, clinical data, neurological reports, toxicology reports, and other data. Will separate submissions be needed for this data, or will it be possible to design a unified submission process – both interactive and bulk – to incorporate these heterogeneous data sets?
4. How will the MAGE loader/validator be adapted to meet the Brainbank's needs? Preliminary inspection of the loader/validator suggest that some modification will be needed, because the loader/validator is written as an interactive command-line script, but the Brainbank's needs include automated execution from within a web-based script. Unfortunately, the loader/validator source code lacks adequate documentation, so adaptation may present a challenge.
5. How will the MIAMExpress tool be adapted for the Brainbank? The software is written in perl, which may complicate implementation of the repository, since the remainder of the repository will be written in JSP. The MIAMExpress developer community will undoubtedly be a valuable resource in this effort.
6. How will the Brainbank wish to handle external submission of microarray data? Should the repository provide automated tools and standards for direct submission and validation, or should there simply be an FTP site with manual curation and conversion of data as deemed appropriate by Brainbank administrators? Certainly the latter approach would be the easiest path for the first system release, while automatic submission could be supported in the future as usage and contributions to the repository increase.

4.8.3 Curating the Brainbank Data

Once information about an experiment is submitted, it will need to be curated to check for technical errors, add annotations, etc. This function will require a complex application that displays and facilitates the information associated with an experiment. For experiments conducted at HBTRC, this tool will also display and facilitate updates of related data, including raw CHP image files, brain images, clinical data, etc. The MAGE-stk will be used to load information about the experiment from the database into a Java data structure that can be manipulated and stored in the relational database. Issues raised by this aspect of the framework include:

1. Will it be necessary to modify the MAGE-stk to make database queries? The MAGE-stk appears to have some support for querying databases, but significantly more support may be necessary for complex operations needed for curation.
2. Will it be necessary to load all of the information associated with an experiment at once, or will it be possible to load different parts at once? If so, it may be necessary to further modify the MAGE-stk, or to extend it to cache entire MAGE objects?

4.8.4 Searching the Data

The framework for this function depends significantly on the kinds of searches that are desired. If the searches are relatively simple, it may be possible to implement search via direct database queries couched in JSP scripts. Otherwise, it may be necessary to use the MAGE-stk to load certain data sets as Java data structures, computationally analyze the data sets in custom JSP scripts, and return those that meet certain criteria. Further clarification of the search function is necessary before the framework can be more fully defined.

4.8.5 Browsing the Data

In the initial development phase, the data will probably be presented in a simple, HTML-based format, and data sets will only be presented one at a time. For this purpose, the MAGE-stk will be used to load MAGE data from the database, and other code will be employed to display the data in the desired display format. In what ways should data be presented to users for quickly browsing, navigating and examining results in detail?

4.8.6 Exporting the Data

The data⁵⁸ will likely be made available in two or more formats, depending on the format in which the original data was submitted:

- § If the data was submitted as a MAGE-ML file or via MIAMExpress, then the data will be exported for download as a MAGE-ML file, and as a series of raw text tab-delimited files similar to those available at ArrayExpress, and optionally as a SOFT file, if all of the proper GEO accession numbers are available.
- § If the data was submitted as a raw ASCII text file (similar to the SOFT format), then there will not be sufficient information to produce a valid MAGE-ML file. The data will be exported as a series of raw text tab-delimited files similar to those available at ArrayExpress and GEO possibly including annotations or other changes to the original ASCII text file.

To satisfy these requirements, it will be necessary to adapt the MAGE-stk to export data into raw tab-delimited text files. The data will need to be retrieved from the database in a manner similar to that described for other functions. Alternatively, all ASCII text data and image files could be stored in their native formats in the file server and their locations referenced in the MAGE-OM classes as hyperlinked files. This would minimize database storage requirements and provide rapid access to pertinent files without performance lag associated with retrieving data and assembling data from the database tables.

⁵⁸ Here we only discuss information about the microarray experiments; some additional effort will be needed to present the clinical data, neurological reports, and other non-experimental data.

5 National Brain Databank: Proposed Model and Approach

5.1 Summary of Preliminary Requirements

Based on the review of Brainbank's research processes and background research on existing gene expression repositories and emerging data exchange standards, there are several key requirements for the National Brain Databank:

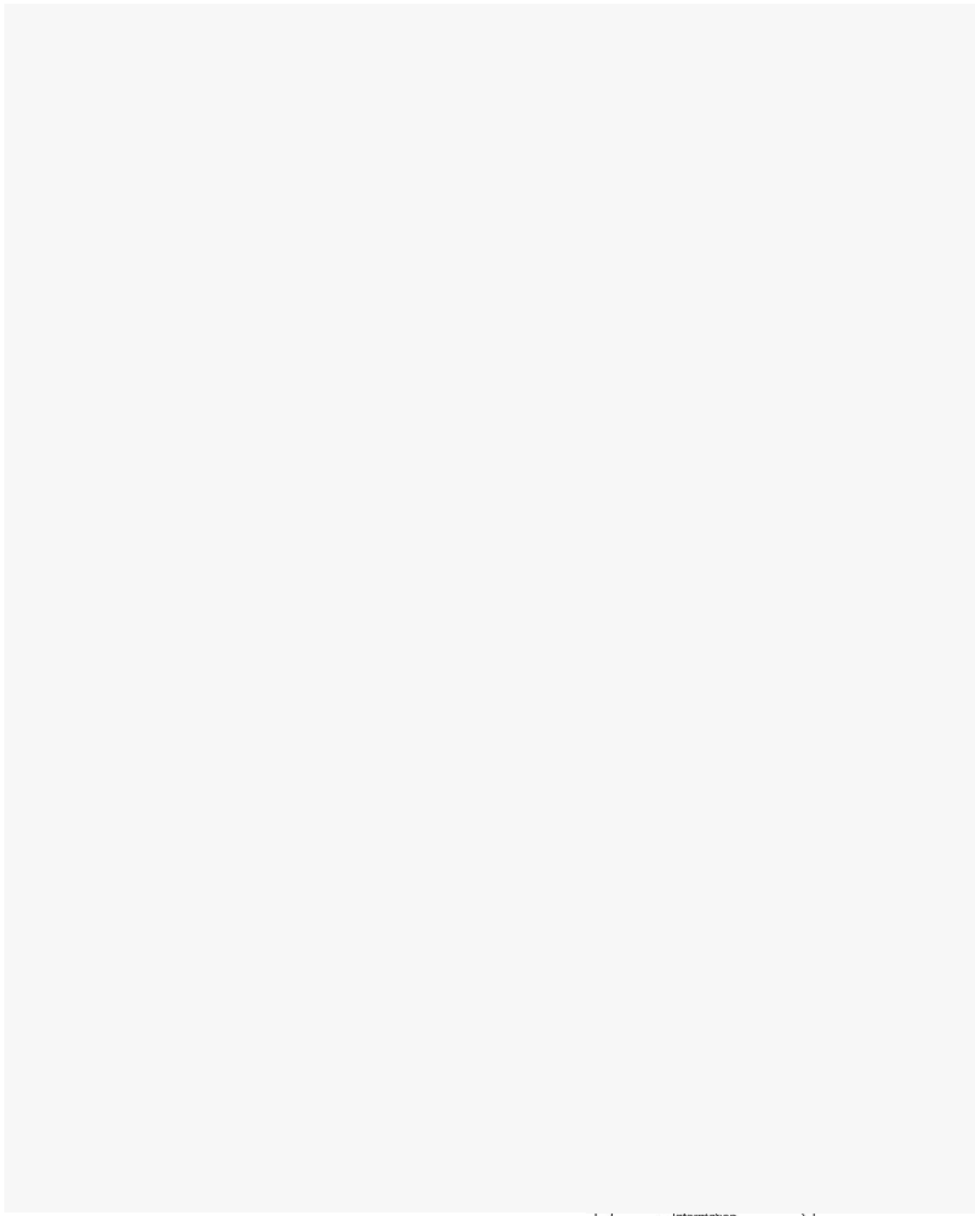
1. *Support large volumes of heterogeneous data:* The data generated from gene expression experiments is represented among a range of file formats including raw text, image and binary files as well as references to diagnostic reports in html or PDF formats. The data generated from each experiment may be up to 72 MB and several replicates for each experiment can be conducted. While metadata on experiments may be captured in the database, the raw files should also be stored in the server and referenced accordingly. However it is not expected that large amount of gene expression data will be submitted from external investigators initially, while much of the data will be internally generated by the Brainbank.
2. *Support Emerging Standards and Ontologies as well as Native formats:* While the emerging MAIME and MAGE-based standards are highly beneficial for representing and exchanging gene expression data, researchers expect to upload and download gene expression data in raw or native data formats as well. The current analytic tools and databases do not yet support MAGE, though some are beginning to incorporate these standards recently. Hence, the repository must allow data import export in both native as well as MAIME/MAGE-based formats.
3. *Tools for Curation and Administering Data:* The Brainbank maintains very high standards in archiving and disseminating data to the neuroscience community, hence any data submitted to the repository must be carefully handled to validate, cross-reference and ensure suitable quality. Appropriate tools for managing the process of curation and metadata submission must be provided to make the system easy to administer. These tools should primarily facilitate internal submission of Affymetrix data at the Brainbank and later be augmented to support external submission via FTP or web-based forms.
4. *Tools for Querying Data and Exporting Datasets:* Similar to existing gene expression repositories, the National Brain Databank must provide adequate tools for querying the diagnostic and gene expression data along a number of searchable parameters. This requires that the experimental data be submitted using MAIME compliant processes as well as indexing the raw data and clinical reports to extract relevant keywords and terms for extensive queries. The system should allow users to filter and generate datasets for export into analytic tools and databases. Hence, use of both MAGE and native formats is necessary.
5. *Linking and Cross-referencing Data:* To allow data to be usable it must be referenced to standardized Gene sequences in GenBank (through NetAffx) and linked to relevant publications in online resources such as PubMed. The system must support mechanisms to cross-link and reference these online sources using a combination of manual and automated methods.
6. *Confidentiality and Security of Data:* Since the brain samples collected and gene expression data generated are based on patient profiles and the online repository is designed to be a publicly accessible resource, data must be selectively disseminated to comply with HIPAA guidelines. Hence, the system should support user authentication mechanisms, a range of user roles and privileges for certain datasets and files, while enforcing adequate security measures in a robust and secure database.
7. *Interoperability with Software Tools and Databases:* Extracting and archiving gene expression data in the online repository requires acquiring data from specialized software like Affymetrix using export tools like GDAC and other utilities for converting content to MAIME and MAGE-ML-based formats. The system must support extensible interfaces and APIs to allow integration with such tools. In the future we may consider integration with analytic software and other online repositories, hence using nonproprietary platforms, open standards and methodologies in the design of the system architecture is critical.

5.2 Proposed Application Model and System Architecture

The requirements for a scalable and extensible online gene expression repository suggests a system architecture that incorporates clinical data, microarray experiment files in various formats and MIAME compliant annotation. The proposed application interface and architecture for the National Brain Databank supports the following:

1. *Managing Cohort Sample Sets:* Brainbank administrators and lead investigators will be able to create “sample cohorts” such as the McLean 66, which can contain individual cases of tissue samples partitioned along various brain regions (5-7). Hence each case may include demographic information, clinical or diagnostic reports and several experimental replicates of the gene expression profiles (for each brain region). The system will provide a curation tool to allow users to create the cohort sets, individual cases and submit relevant metadata and files.
2. *Metadata Submission:* Once individual cases are setup, a MIAME compliant submission mechanism (either desktop or web-based) will be developed to allow all metadata to be provided for the gene expression profiles. An application like MAIMExpress may be adapted for the needs of the Brainbank. Some metadata may be extracted from the MAGE-ML files generated by Affymetrix GDAC Exporter, and the text-based experiment and report files, while other information will need to be manually entered. Clinical and diagnostic reports from neuropathology and toxicology assessments will be referenced and indexed in the database, so that they can be associated, searched and retrieved. Actual brain tissue images will not be placed in the repository initially, though this feature may be implemented in the future.
3. *Submission of Gene Expression Profiles:* Gene Expression Profiles from microarray experiments generated at the Brainbank will be exported to MAGE-ML files using the Affymetrix GDAC Exporter. The XML files will be parsed using the MAGE-stk, a Java API into MAGE-OM objects in the Java-based application. The raw data files will be referenced and stored in the main server, while small-scale microarray images extracted from the DAT files (using *ImageMagik*) will be maintained. The process will require several steps for annotation, validation, cross-referencing and insertion of the sets of related data files for each experimental replicate of the gene expression profile.
4. *Curation of External Data Submitted:* While the primary data submitted in the repository will be internally generated at the Brainbank, in some cases external investigators may submit gene expression profiles for brain tissue samples obtained from the Brainbank. To facilitate external submission, the system will initially maintain a secure FTP site where users can deposit data and related files, which will be curated by Brainbank administrators and submitted to the repository as previously described. However, in the future external investigators would be provided a web-based submission mechanism with appropriate tools for validation and workflow to allow administrators to monitor and curate the process more easily.
5. *Application Framework and Database:* The web-based application will utilize a MAGE-based Object Model to represent all MAIME compliant data internally. The MAGE classes will be extended to support unique aspects of Brainbank’s data while additional classes will be created for application functionality. All data will be archived in a relational database with a table schema that maps to the MAGE OM as well as other user authentication and application information. The database will be interoperable with other tools and utilities, maintain selective security access and regularly archive data in the repository.
6. *Flexible Query and Data Export Mechanisms:* The system will provide intuitive tools for online browsing of data sets and queries along various parameters in the metadata, as well as access to microarray images and diagnostic reports. Data can be exported in MAGE-ML (via MAGE-stk) and the native formats.
7. *Administration:* The system will let administrators manage users by assigning roles and privileges to specific data sets and types of files. Administrators would be able to monitor site usage, enable/disable selective features of the application interface and configure the repository as needed.

We will now consider specific implementation issues in the development of the online repository and database, in particular rationale for selecting appropriate application programming framework (Java J2EE), the backend database (Oracle) and the operating environment (Linux OS). The proposed deployment architecture must ensure long term scalability, robustness, performance, extensibility and interoperability with other systems and platforms.



Information

5.3 Designing the Application Platform: Adopting Java J2EE

In creating any online application, there are many different decisions to be made as far as a technology is concerned. One of the most important choices is the decision to go with a programming language for the application layer itself, which will act as the server's decision maker; it will serve user requests for data to and from the database, and will interpret and display all desired data.

Akaza Research supports the Java programming language and the Java 2 Enterprise Edition (J2EE) framework of services for online applications for a number of reasons:

A. The J2EE framework is robust, and reliable. J2EE is based upon Sun's Java Software Development Kit (SDK), a reliable language for all forms of application development on all forms of processors, from mobile phones to supercomputers. Java's "write once, run anywhere" ability allows the code generated for an application to transfer easily from one operating system to another with minimum change.

B. The J2EE community is a vast source of innovation and support. Java, and J2EE along with it, has been distributed to the open-source community for free. The open-source community has responded by creating a number of tools, programs and frameworks for use with J2EE, and sponsors a number of websites and discussion boards to discuss common problems and popular debugging strategies. Several of these tools will be described later in this document. There are also multiple J2EE servers available on the market, so an application written in Java is not necessarily bound to any one company's servers or technologies.

C. The J2EE set of tools are flexible, yet support an extendible structure. The J2EE collection of technologies allow a developer to create an application in many different ways, but the developer community has been active in supporting organized, extensible frameworks that allow large-scale web applications to be developed in a reasonable amount of time. One of the frameworks that Akaza actively supports is the Model-View-Controller framework (MVC), which divides the required tasks of an application into clean, controlled interfaces that are easy to update and extend. Additionally, Java supports the ability to wrap code inside a Java Archive (JAR) file, which can then be easily copied over to other systems.

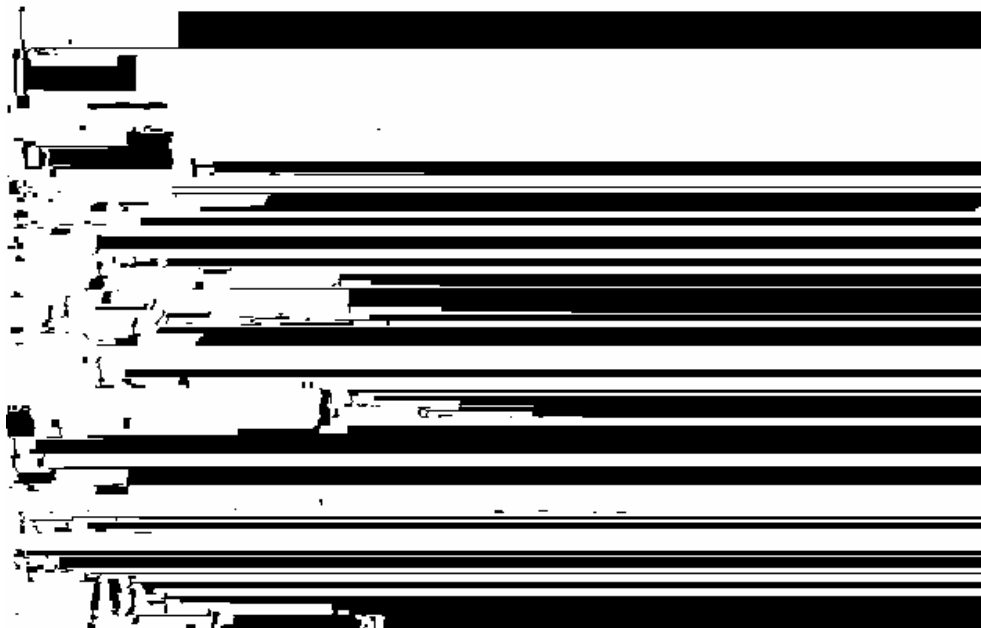
5.3.1 What is J2EE?

Java has been categorized into several sections for ease of use. The Java 2 Standard Edition (J2SE) is the 'standard' set of packages and interfaces which developers use for application development. The Java 2 Micro Edition (J2ME) is the 'micro' edition, for use with microprocessors in cell phones and PDAs. J2EE is the set of packages that are used for web applications; while several packages are not exclusive to usage on the Internet, they are most commonly used there and therefore, are designated under this category.

J2EE includes:

- § Connecting to the database (Java Database Connectivity, or JDBC),
- § The ability to place Java in ordinary HTML pages (JavaServer Pages, or JSP),
- § The ability to generate HTML pages from a Java class (Java Servlets),
- § And objects which hold business logic components (Enterprise JavaBeans, or EJB).

With so many ways to create an application, it is important to subscribe to a framework in building Java classes and objects. The Model-View-Controller (MVC) framework supports an organization style which helps to make applications extensible and easy to understand.



In the MVC, we separate out different functions according to their roles:

- § The Model supports the data model of the application in the form of a database layer, handling database connections and inserting, updating and retrieving relevant data from the database.
- § The View supports the user interface of the application, generating dynamic HTML with JSP and Java Servlets.
- § The Control is the business logic of the application, deciding the workflow of the application, usually taking the form of a controller Java Servlet or an Enterprise JavaBean.

5.3.2 Case Study: PhenoDB Project at Massachusetts General Hospital

In the case of Massachusetts General Hospital, Akaza Research was contracted to create a large phenotypic database for the Tourette Syndrome Consortium, maintaining a large amount of data collected over the years by researchers and clinical investigators. The application was to support multiple sites located all over the world, from the US to the Netherlands to South Africa.

The decided platform was Java together with Oracle on a Sun machine running the Solaris operating system (a Unix variant). In picking Java, Akaza had a great deal of support in creating an application with an available framework, and many open-source tools to choose from. Developers at Akaza went with the collection of Struts tag libraries, a collection of markup tags that gave more power and flexibility to the View portion of MVC. In addition, developers used the MVC framework to create three main modules that could be phased in incrementally; the Reporting module, the Data Collection module, and finally the Administration module.

In creating the Reporting module, developers first had to load legacy data from multiple sources into the new Oracle database schema, and cull the data for patterns of symptoms associated with Tourette, Obsessive-Compulsive Disorders, Attention Deficit Disorders, and so on. The information had to be collected on multiple test subjects and then a specific report format had to be generated with the available data. Researchers would create queries and reports, while administrators of the site would approve the reports for generation, after a review process. Having the guideline of the MVC framework available allowed developers to organize quickly and set this portion of the site up for demonstration in a month's time.

In creating the Data Collection module, the challenge was to generate the paper surveys and reports for Research Assistants to enter the data online, so that fresh data could be instantly updated to the system.

Instruments, sections and questions had to be generated within the database, and associated with certain types of metadata for layout and background information. Data importing took place by uploading Excel spreadsheets containing the pertinent information, and then the instrument versions were ready to be generated online for the researchers' use in entering patient information. In this phase of the project, Java provided objects that are able to retrieve metadata quickly, and the Struts tag libraries were a good resource to help generate forms.

The final module, Administration, focused on retrieving and curating data in the database, allowing for fine-grained control of users, the users' privileges, projects, instruments within projects, and the versions of the instruments that were active in the projects, and so on. After designing the first two modules, much of the database layer was already in place for the Administration interface, and, thanks to the interoperability of Java objects, code was efficiently reused for this part of the project.

In using MVC, the Struts tag libraries, and traditional J2EE programming patterns, the development team at Akaza Research was able to roll out the modules in a matter of months, while remaining flexible enough to adapt to changing requirements. At the writing of this document, the application (PhenoDB) is going through its final stages of testing and deployment on-site.

5.3.3 Available Java Tools and Comparison with Other Languages

We mentioned earlier in this document that there are now many open-source packages available for use in creating a J2EE application based on the Model-View-Controller framework. These tools include the following:

Struts: A development framework that enables developers to quickly assemble applications in a pluggable and extensible manner. Struts consists of both Java Servlets and tag libraries for use within JavaServer Pages, and either of these technologies can be used to support the MVC pattern.

Maverick: Often billed as an alternative to Struts, an MVC framework that allows for multilingual content, also supportive of rapid development and extensibility.

Lucene: A powerful search tool that can be used to implement a search engine for any web-based application.

Velocity: A templating framework that allows a development team to build 'skinnable' applications, whose 'look and feel' can easily be changed.

Ant: An industry-accepted Java build utility that allows you to create sophisticated application and deployment scripts.⁵⁹

ObjectRelationalBridge: An object-relational mapping tool that allows developers to map objects to tables in a database, significantly reducing the amount of data access code that needs to be written by an application team.

Use of Java in Existing Gene Expression Repositories

In looking at gene expression formats and databases, we also see a preference for Java classes to handle the various XML-based formats of information. All of the online databases are starting to embrace the MAGE-ML standard, and there already exists a set of Java classes that mirror the MAGE object model. Additionally, several of the open-source packages built for exchanging gene information (GeneX, OmniGene, MAGEstk) use Java either partially or exclusively.

⁵⁹ Carnell J, Linwood J, Zawadski M, *Professional Struts Applications*, Wrox Press 2003

Comparison with Other Programming Languages

Java is certainly not the only choice in a web application platform, there are many other languages on the market competing for the title of the web application language of choice among developers. In taking a look at other choices, we feel our support for Java is justified. Below are some of the major features of Perl, PHP and Active Server Pages (ASP):

Perl/PHP:

- § Strong developer community, but in the case of Perl, it's not totally focused on web services, so finding help for specific problems using Perl on the web can be limited, depending on what you're looking for.
- § Perl and PHP are more or less tied to Apache, which is not a bad thing, but this locks you into a choice for an operating system (Linux).
- § Perl and PHP are scalable, but certain code rewrites may be required if you switch from Linux to Windows in the case of Perl.
- § The pressure to build a framework is back on the developer's shoulders; Perl and PHP are very hands-off when it comes to embracing a standard to create web sites and web applications.
- § Good connections to the database; but as they say in Perl, 'there's more than one way to do it'; the developer must pick the method/package/module to connect to the database and hope that someone doesn't change the package in a future release. Backwards compatibility is usually supported, but sometimes not.

Microsoft ASP:

- § There is a developer community, but mostly they exist on Microsoft-supported web sites; not as much grassroots participation as with Java.
- § Ties to the Internet Information Server, so again, you are not free to decide your web server/operating system with ASP.
- § While it is scalable, no ASP will ever run on Linux, and so ASP is not portable.
- § Again, no apparent framework available for the developer to plan out and build larger systems.
- § Good connections to the database, but the majority of the ASP literature supports and prefers MS SQL Server over other databases.

Other systems include the following:

Python is a flexible scripting language, but is usually tied to Zope, its own webserver/database variant. It can be used with the Apache web server and mod_python, a not-quite-mature module that is difficult to install and maintain.

Tcl is tied to the Aolserver web server, a fine product in its own right, but if you want to switch web servers or databases, you would have to switch the application language as well. There is questionable support for database connectors for Tcl outside of ones custom developed for the PostgreSQL and Oracle databases.

Ruby, a Japanese variant of Python, still has issues dealing with connection pooling, and currently cannot be considered a 'mature' programming language.

Summary: The use of Java and the J2EE family of services and modules is the obvious choice. With the extensibility of Java, you are still free to:

- § Select your own web server
- § Select your own database
- § Select your own operating system

In short, Java and J2EE are structured to allow for the maximum flexibility and versatility on the Web. The open-source market contains a boundless amount of innovative products and support for the developer of web-based services, to support rapid application development and powerful tools. Using the available resources in the web developers' community and the gene researchers' community, we can create an application for the HBTRC which is robust, comprehensive and scalable.

5.4 Adopting a UNIX Operating Environment for the National Brain Databank Server

Selecting an appropriate operating system for the National Brain Databank is critical not only for technical decisions regarding development of the online repository but also long-term security, performance and robustness of the system. Here we consider the key criteria and compare UNIX and Microsoft NT operating systems in the context of the needs of the National Brain Databank. We recommend UNIX, particularly the Linux variant for a variety of reasons outlined below.

1. Security

UNIX was designed as a multi-user, multi-process system. Therefore, the issue of security was a primary concern in the initial design. Open source nature and closed Internet integration encouraged vendors to be more and more open about security leaks in the system. As a result, security patches are released more quickly, and users are notified more readily of potentially dangerous problems. In contrast, Microsoft has been secretive about security problems. Patches are frequently not released until the large number of users experiences them. Furthermore, Microsoft is a frequent virus target, so security attacks against NT Systems are more likely than attacks against UNIX systems.

2. Performance/Efficiency

The performance of any system is difficult to measure, because the speed of a system can vary tremendously depending on the application executed, the task performed, and a number of other factors. However, tests conducted by RSH Web services show that with UNIX and Windows NT running on 133MHz PC's, UNIX ran 27% faster than Windows NT when reading static HTML content, and with API generated content, UNIX is between 47% and 197% faster. For CGI contents, UNIX is 77% faster than Windows NT⁶⁰.

In addition, UNIX is usually more proficient in the use of its memory, especially when dealing with network services. Because UNIX requires less memory and processor time than Windows NT, UNIX based system has more memory and processor power for other computer functions. NT is tough on server resources and will require a generous allocation of RAM for relative stability to be achieved.

3. Stability

A study in August 2000 of the 100 most popular web sites in the world⁶¹ revealed that UNIX servers are substantially more stable across a wide spectrum of metrics than their Windows counterparts. While NT servers showed a mean downtime of 1.9 percent, UNIX servers showed a mean downtime of 0.5 percent. The mean UNIX failure duration (time to respond after an initial timeout) was 25 minutes, compared with 53 minutes for Windows NT. On average, UNIX servers failed ten times over a 32 day period, while NT servers failed 18 times.⁶² While benchmarks rarely provide a decisive conclusion, the failure of NT servers across a wide variety of tests certainly does not suggest it is reasonably stable.

4. Ease of Use/System Management

UNIX was designed to support remote management; consequently, system administrators can remotely perform management operations from another building or across the world. Because Windows NT is primarily oriented around a graphical user interface, most administrative programs require the administrator's physical presence. This limitation severely hampers usability, and may increase the cost of a system.

While the user interface of UNIX is command-line-oriented, many graphical user interfaces are also available. For example, Red Hat Linux ships with two different interfaces, the X Windows and KDE Desktop environments. Both interfaces have a familiar Windows "look and feel", and can be configured and modified as needed to suit the user's needs.

⁶⁰ http://rshweb.com/support/vservers/UNIX_nt.html

⁶¹ As listed by the German Informationsgemeinschaft zur Feststellung der Verbreitung von Werbeträgern, <http://www.ivw.de>

³ <http://www.heise.de/ct/english/00/08/174/>

5. Interoperability with Software Utilities and Internet Services

Many server applications and programs such as sendmail are written exclusively for use with UNIX and may not work on Windows machines. UNIX has included things such as SMTP (Email), NNTP (News), Telnet, and DNS. All of these protocols and services were somehow forgotten by Windows NT. They can be covered up with third party products and Microsoft's own programs. However, none of these programs and products can compare to UNIX in terms of flexibility and power.

Of particular interest to the Brainbank are the development efforts centered around MIAMExpress, a software tool which may be employed in the Brainbank repository. MIAMExpress is developed in Perl. The support for Perl in UNIX is considerably better than that in Windows. While Perl interpreters and help documentation developed by the Comprehensive Perl Archive Network⁶³ are readily available in UNIX, Windows counterparts are ported by third parties and are typically not easily integrated into the Windows interface. Development and adaptation of this tool would therefore be significantly easier on a UNIX system than on a Windows system.

Interesting note: Since 1997 and until fall 2000, Microsoft used UNIX-based SUN/Solaris OS to run its web-based e-mail service Hotmail. Only with the appearance of Windows 2000 Server, Microsoft finally decided to move Hotmail to its own server software product.

Why Linux?

Linux is the one of the latest flavors of UNIX that inherits all the UNIX functionality and adds a lot more. Its completely open-source nature improves performance, eliminates bugs, and strengthens security. Linux is exceptionally stable and runs on a wide range of hardware. It has low cost of ownership and a variety of built-in tools and application for the development. Its native networking protocols of other OS such as UNIX, Microsoft Windows, IBM, OS/2 and Macintosh make interoperability easy. Finally the graphical user interfaces for Linux available today, decrease the learning curve and make it easier to operate.

There are several variants of Linux, including Red Hat, SuSe, and Debian. Currently both Redhat 8.0 and SuSe are supported by the HP Proliant ML 370 server⁶⁴, which has been setup for the National Brain Databank. Akaza Research will work with David Ennulat at the Brainbank to select the appropriate variant of Linux which best suits the purpose of the repository server, interoperates with the current hardware and is supported by Partners.

5.5 Adopting a Relational Database: Comparison of Database Platforms

To select a database for the National Brain Databank, Akaza conducted background research to compare three database software packages that we consider to be most useful and robust in this setting: MS SQL Server 2000, Oracle 9i, and PostgreSQL. We examined different characteristics such as performance, security, scalability, easy of use, cost of purchase and maintenance, stability, availability, robustness, XML support and JDBC connectivity. Some information was excerpted from official websites of the database vendors, but in addition we examined different experiments and tests published in evaluation studies to assess the stated characteristics from an independent and practical point of view. A detailed description of the analysis is provided in the Appendix.

Among the three databases considered for the National Brain Databank, our assessment is that given the key criteria, using Oracle as the main database provides the best compromise in tradeoffs. Hence, Akaza strongly recommends using Oracle as the database platform for the National Brain Databank. Here we summarize the main rationale for selecting Oracle along several key criteria.

1. Use of the Java J2EE Application Framework

Given that we have selected Java J2EE as the development platform for the Brainbank project, it is important to ensure robust and high performance database connectivity through the JDBC driver. The JDBC driver provided by Oracle tends to provide the best connectivity between the database and Java applications. Lately the relationship between these two software products became as native as the

⁶³ The central body for development of Perl; see www.cpan.org.

⁶⁴ <http://h18004.www1.hp.com/products/servers/software/index.html>

connection between ASP language and SQL Server database. While for SQL Server the JDBC support is not as strong important, primarily because Java has not been a primary programming language for Microsoft products. The current JDBC driver provided by Microsoft has severe limitations on SQL Server performance, and a recent study shows that using this driver the database server was only at about 15-20% CPU utilization. Keeping in mind the volume of data that National Brain Databank database will use, such low utilization would really hurt the performance of the whole system and will slow it down substantially. PostgreSQL has a reasonably good JDBC driver but the database itself performs slower than Oracle and SQL Server; hence Oracle currently provides the best connectivity for Java applications.

2. Security Requirements at the Brainbank

Security is always one of the most important criteria for any software system in a medical research setting, especially given the nature of research and confidential data on brain tissue samples at the Brainbank. Both Oracle and SQL Server database vendors claim that their systems are highly secure. However, in practice security is a highly problematic issue for SQL Server. Since Microsoft released its current version of SQL Server there have been at least two serious attacks on the databases, which managed to access the server and damage data (please see the appendix for a recent case study on the Slammer virus attack on SQL Server⁶⁵). Unlike SQL Server, the Oracle database has not had such serious security problems documented. PostgreSQL tends to have many security problems; being an open source database, features are not readily available in a timely manner and security holes are easy to find. Oracle provided multiple levels of security and many built-in features to handle security issues. Oracle would definitely be a stronger choice if security is a big concern.

3. XML Support

As XML becomes one of the essential tools when working with large data sets, the nature of support provided by the backend database turns out to be a significant issue to consider. Akaza has recommended using the XML data exchange format such as MAGE-ML for gene expression data; which is also used by other public repositories like ArrayExpress. Therefore, XML Support is essential for the Brainbank's database functionality. While both Oracle and MS SQL Server tout their support for XML, in practice Oracle provide more extensive and robust support. PostgreSQL provides a XML implementation called DBDOM however its performance has not been well documented. Even though Microsoft claims "Rich XML support" as a feature of SQL Server 2000, some experts have expressed doubts. "[SQL Server] lags in its ability to store and manipulate XML data," points out Carl Olofson, an analyst with IDC, in Framingham, MA. On the other hand, Oracle XML Support has been steady; it includes JDeveloper that provides a complete, highly extensive XML development solution.

4. Availability and Robustness

The National Brain Databank is meant to be a public repository of data; therefore, it is our assumption that the ongoing availability and accessibility of the database is the important requirement. From our experience, Oracle has shown greatest reliability and availability; projects developed by Akaza Research running on Oracle have never experienced any downtime, however this is not the case with SQL Server. In our prior experience, SQL Server has been down frequently sometimes requiring reboot of the Windows machine. In many cases it takes time to figure out what the problem is before the system can be revived. While the performance of both SQL Server and Oracle are nearly the same, most database evaluation studies show that Oracle tends to have a greater rate of availability over time.

5. Large Database Developers Community

Both Oracle and SQL Server have a very large community of Developers as well as online documentation. Therefore help is easy to find, and almost all questions can be answered through online websites. For PostgreSQL there is some documentation online, however not a very large developer community outside of academic institutions; hence it is hard to find database administrators (DBAs) to

⁶⁵ <http://www.f-secure.com/v-descs/mssqlm.shtml>

assist. For Oracle products, there are a large number of DBAs available to provide maintenance and support, particularly in academic settings where it is frequently used.

6. Ease of Use and Scripting Language

Oracle and PostgreSQL require some effort for the initial installation and configuration on UNIX (PostgreSQL tends to be far more tricky to setup), while SQL Server is generally straight forward to setup on Windows. Up until recently SQL Server's visual and user-friendly interface had been a reason why many companies decided to go with Microsoft database. However, Oracle's recent versions now also provide graphic user interfaces (running in Java) as opposed to the command line prompt. In addition to that Oracle uses PL/SQL language which is more developed and robust than the T/SQL which is used by SQL Server. Oracle features such as arrays, Java methods, Bitmap indexes and Object tables would definitely come handy as the project goes along.

7. Multimedia and Large Amount of Data Support

Support for multimedia (image) and large volume of data is essential at the Brainbank. TIFF or DAT files used to store microarray images can have very large size (10-40 MB). Other gene expression data and future datasets including brain imaging etc pose a challenge for databases. One of the advantages of Oracle versus SQL Server and PostgreSQL is that Oracle is better known in its ability to handle large volume of data, without any substantial decrease in performance. Unlike SQL Server which has previously been used mostly for small and mid-sized data projects, Oracle has usually been selected for large scale data-oriented projects, particularly with large volumes, transactions and multimedia content.

8. Cross-platform Support including Linux

While all three databases can run well on Windows operating systems (OS), only Oracle and PostgreSQL can be run on multiple platforms including many variants of UNIX and Linux. MS SQL Server is intended to be run on Windows only. Given that Akaza has recommended the use of Linux as the primary OS for the National Brain Databank repository, MS SQL Server does not provide a suitable option. On Linux, the high performance of Oracle suggests the most suitable option. However, Oracle databases can also be run on most high-end Windows machines at the Brainbank in the future.

While we have summarized the comparative features of all three databases here and the main rationale for selecting Oracle, we provide detailed review of these databases in the appendix.

6 Summary of Ongoing Requirements Analysis

The next phase of requirements gathering will include the following main tasks leading up to a well defined functional specification of the online repository, based on the preliminary analysis in this technical paper:

1. *Use Cases:* We will identify and conduct interviews with a range of administrators and potential users of the National Brain Databank, including personal at the Brainbank and external investigators. These interviews will provide information that will be organized into 'Use Cases' detailing the nature of interaction for all major tasks supported by the system. The use cases will enable us to derive relevant user roles and privileges for the online repository, and an overall security model.
2. *UML Diagrams and Object Models:* The use cases along with a review of the existing Brainbank database schema and MAGE OM classes will allow us to generate conceptual diagrams using the Unified Modeling Language (UML). This representation will provide an intuitive logical model to understand the relationships between various aspects of microarray data and classes of user interaction. The UML diagrams will enable us to create suitable Object Models, partially adapted from MAGE-OM and MAIME. The Object models will serve as the basis for the main Java application design of the repository.
3. *Database Schema:* The Object Model defined above will be utilized to map the classes to relevant tables in a relational database (using object relational mapping). The database schema developed will also incorporate unique aspects of the Brainbank's research processes and specific application requirements. It will be validated and cross referenced with other tables in the schema and classes in the Object Model to ensure a good correlation between the application and database, and support future extensibility.
4. *Mapping Microarray Data to MAGE standards:* Due to the complexity of the microarray data, once a MAIME compliant object model is developed all relevant data must be mapped to the MAGE standards. Clearly many aspects of the current data and meta-information will not map directly to the MAGE model, hence we will need to develop appropriate extensions to the model and database schema as needed.
5. *Specification of Data Import/Export and Curation Processes:* While the use cases and object models will provide the conceptual approach, additional specification for importing and exporting data for a variety of existing formats and specifications will be documented. This will require working experience with the Microarray software and related tools at the Brainbank. These specifications will be utilized to adapt and develop suitable utilities and application interfaces for the system.
6. *Application User Interface:* Based on the use cases defined and the overall application architecture, we will develop 'screen shots' or mockup interfaces to demonstrate the look and feel of the working application. These interfaces will be iteratively designed with user feedback to ensure standardized usage and intuitive interaction with the online repository.
7. *System Architecture:* We will define a finalized system architecture, which will include selection and integration of application frameworks, databases and web servers. We will specify mechanisms for integrating with appropriate software tools as well as setup of the development and application servers.

These key components will be included in a functional specification document with the corresponding diagrams, schema and models. The functional specs will be provided to the Brainbank for further review and refined based on feedback, before initiating the system development and deployment phases.

7 Conclusions

The Brainbank is embarking on a bold initiative to establish a public gene expression repository for the neuroscience community. The initiative requires a long term perspective to develop an appropriate application platform with a scaleable and robust database and incorporating suitable microarray standards and ontologies. In this working technical paper, we have surveyed the overall lifecycle of research at the Brainbank with respect to the microarray experiments, reviewed the main gene expression repositories and analytic tools as well as the emerging MAIME and MAGE-ML standards being adopted by the research community.

While there are a number of public gene expression repositories, most are either generalized for broad submissions or have a specialized focus of datasets provided. All repositories allow some mechanisms to submit, query and retrieve datasets. While many repositories initially used their own data exchange standards, most are now adopting the MAIME and MAGE standards by extending their own data models and approaches. However, these standards have not been integrated in all databases and analytic tools, hence repositories must continue to support submission and export of data sets in native formats as well as MAGE.

We have proposed a system architecture that allows integration of existing Affymetrix-based microarray data using MAGE object model and interfaces, while retaining the data in its raw form. We believe the proposed repository will benefit from an architecture using the Java J2EE application framework and the Oracle 9i relational database running on a secure and high-performance Linux-based server. This architecture enables an open, scaleable and extensible approach towards development and deployment of the repository in conjunction with existing software tools and standards in academic settings.

The key challenges for developing the online gene expression repository include mapping the complex and diverse data generated in microarray experiments along with Brainbank's unique research data in a coherent and extensible object model and database schema. Developing tools for automatic submission of MAIME compliant data by external investigators and mechanisms for administering and curating data is also a major challenge. Finally, making a complex system secure, accessible and intuitive for regular usage by the neuroscience community is an important and challenging goal for the project. Rigorous requirements analysis and well defined functional specifications with iterative design will allow us to develop a robust and extensible platform underlying the system. Over the course of several software releases, appropriate features will be implemented and evaluated. However the basic framework outlined in this working technical report should serve as a robust foundation for the evolving gene expression repository at the Brainbank.

References

Most publications listed here are available on PubMed

Microarray Data Standards

Stoeckert CJ Jr, Causton HC, Ball CA. **Microarray databases: standards and ontologies.** Nat Genet. 2002 Dec;32 Suppl:469-73. Review.

Spellman PT, Miller M, Stewart J, Troup C, Sarkans U, Chervitz S, Bernhart D, Sherlock G, Ball C, Lepage M, Swiatek M, Marks WL, Goncalves J, Markel S, Iordan D, Shojatalab M, Pizarro A, White J, Hubley R, Deutsch E, Senger M, Aronow BJ, Robinson A, Bassett D, Stoeckert CJ Jr, Brazma A. **Design and implementation of microarray gene expression markup language (MAGE-ML).** Genome Biol. 2002 Aug 23;3(9):RESEARCH0046.

Hernandez-Boussard T, Gollub J, Ball CA, Demeter J, Matese JC, Sherlock G. **Mapping the MAGE-OM to data within the Stanford Microarray Database.** Stanford Technical Report, June 20, 2003.
http://www.mged.org/Workgroups/SMD_to_MAGE_Mapping.doc

Gene Expression Repositories

Kellam P. **Microarray gene expression database: progress towards an international repository of gene expression data.** Genome Biol. 2001;2(5):REPORTS4011. Epub 2001 May 02.

Edgar R, Domrachev M, Lash AE. **Gene Expression Omnibus: NCBI gene expression and hybridization array data repository.** Nucleic Acids Res. 2002 Jan 1;30(1):207-10.

Brazma A, Parkinson H, Sarkans U, Shojatalab M, Vilo J, Abeygunawardena N, Holloway E, Kapushesky M, Kemmeren P, Lara GG, Oezcimen A, Rocca-Serra P, Sansone SA. **ArrayExpress--a public repository for microarray gene expression data at the EBI.** Nucleic Acids Res. 2003 Jan 1;31(1):68-71.

Gollub J, Ball CA, Binkley G, Demeter J, Finkelstein DB, Hebert JM, Hernandez-Boussard T, Jin H, Kaloper M, Matese JC, Schroeder M, Brown PO, Botstein D, Sherlock G. **The Stanford Microarray Database: data access and quality assessment tools.** Nucleic Acids Res 2003 Jan 1;31(1):94-6.

H. Mangalam, J. Stewart, J. Zhou, K. Schlauch, M. Waugh, G. Chen, A. D. Farmer, G. Colello, and J. W. Weller. **GeneX: An Open Source gene expression database and integrated tool set.** IBM Systems Journal 2001; 40(2): 552 – 569.

Gene Expression Analysis

Li, Cheng and Wong, Wing Hung. **DNA-Chip Analyzer (dChip).** In *The analysis of gene expression data: methods and software.* 2003. Edited by G Parmigiani, ES Garrett, R Irizarry and SL Zeger. Springer.

Murphy, D. **Gene expression studies using microarrays: Principles, problems, and prospects.** *Advan Physiol Educ.* 26(4): 256 – 270. 2002.

Schadt, Eric E., Li, Cheng, Su, Cheng, Wong, Wing H. **Analyzing high-density oligonucleotide gene expression array data,** *Journal of Cellular Biochemistry.* 80, 192-202. 2000.

APPENDIX I: Database Comparison of MS SQL Server, Oracle 9i and PostgreSQL

A. Review of MS SQL Server 2000

1. Security

On their website Microsoft claims that “SQL Server 2000 introduces significant new security enhancements” on the level which is “easier to achieve”. New features such as *powerful and flexible role-based security for server, database, and application profiles, integrated tools for security auditing, tracking 18 different security events and additional sub-events, support for sophisticated file and network encryption, including Secure Socket Layer (SSL), Kerberos, and delegation* make SQL Server 2000 one of the highest standards in security. Microsoft received C2 security rating from the National Security Administration (NSA), which is one of the highest ratings of the security standards⁶⁶.

SQL Server’s security model comprises the following 5 components: SQL Server login, Database user, guest user, permissions, Roles. Administrator can configure permissions at server level, database level, object level and even at column level. SQL Server also takes an advantage of Windows authentication. However, even with all new features the software still has some security holes, and could be vulnerable to service attacks. One of the examples of such attack is virus called Slammer (see next section of the appendix). Since the release the number of security holes has been discovered. Even though Microsoft is continuously releasing patches, the issue of 100% security remains in question.

2. Performance

Statistics from Microsoft.com website shows that SQL Server 2000 can handle 709,220 transactions per minute. On the Transaction Processing Council (TPC) TPC-C performance list, SQL Server 2000 holds four of the top five results⁶⁷. However, studies show that the performance of the database depends on few factors such as Web Server, Application Engine, and Database Driver. SQL Server performs perfectly with all-Microsoft software. Yet when it comes to changing any of the above components, its performance slows down and can be even limited. A good example of it is using JDBC driver. On the tests conducted by eWeek magazine⁶⁸, when using JDBC Microsoft SQL 2000 was limited to 200 pages per second for the entire test.

The main reason of such poor performance is the Microsoft JDBC driver that has many problems including memory leaks. Using this driver SQL Server was only at about 15-20% CPU utilization. At the same time SQL Server is easy to tune, i.e. find the best-performing memory configuration in terms of how much memory to assign to the various subsystems used by the database. In average it requires only 50 KB of RAM per connection. Automatic tuning and maintenance features enable administrators to focus on other critical tasks. As pointed out by Carl Olofson, an analyst with IDC, in Framingham, MA, “Microsoft has yet to equal Oracle’s Real Application Clusters technology, which lets businesses run a database across groups of servers for nearly continuous uptime. It also lags in its ability to store and manipulate XML data”⁶⁹.

3. Scalability/Interoperability

Information from Microsoft’s library tests⁷⁰: “On Microsoft Windows 2000 datacenter Server, Microsoft SQL Server 2000 ... scales up to 64 gigabytes of RAM and up to 32 CPUs. This can be used in conjunction with scale-out techniques, such as Distributed Partitioned Views, to handle the largest data sets and transactional loads.”

⁶⁶ <http://www.radium.ncsc.mil/tpep/epl/entries/TTAP-CSC-EPL-00-001.html>

⁶⁷ <http://www.tpc.org>

⁶⁸ http://www.eweek.com/print_article/0,3668,a=23115,00.asp3

⁶⁹ <http://www.nwfusion.com/news/2003/0425microeyes.html>

⁷⁰ <http://www.microsoft.com/sql/evaluation/features/scalable.asp>

However, SQL Server 2000 cannot scale across all platforms. It tied to Microsoft Operating System and Intel hardware. As was pointed above the best performance is achieved using all-Microsoft products. In addition to that “SQL Server can only truly scale on single server systems. This means that SQL Server 2000 can only scale up vertically.”⁷¹

4. Easy of Use, Programming and Configuration.

SQL Server is easy to install, tune, and configure; however, there is a chance of needed reconfiguration during the performance process. It has user-friendly interface and allows you to pose questions in English instead of using multi dimensional expressions. Graphical tools and wizards simplify setup, database design, and performance monitoring. Being a Microsoft product it has easy integration with Microsoft Office applications such as Excel and Access.

At Akaza Research, we have experienced configuration problem with SQL server after it was installed on one of our machines. It is sometimes challenging even to define the problem, not to mention fixing it. The other issue here is portability. SQL code generated by SQL Server built-in utility even sometimes doesn't work on another SQL Server installation.

5. Cost of Purchase/Upgrades/Maintenance

SQL Server 2000 Standard Edition costs \$4,999 US. The price for SQL Server 2000 Enterprise Edition with Management Tools, and Advanced Security features is \$19,999 US according to Microsoft website. However, many academic institutions have licenses that allow them to obtain it at minimal cost.

6. Stability/Availability/Robustness

SQL Server includes couple ways of backing up the system while the database remains online and accessed by users. Microsoft claims ‘High Availability’, but provides no numbers in terms of database availability to the users. As was mentioned in the Security part SQL Server is vulnerable to different malicious attacks which definitely reduce its stability and availability.

7. XML Support

Microsoft claims “Rich XML Support” as one of the most important features of the SQL Server 2000. According Baya Pavliashvili, “It is one of the first relational database engines that offers native XML support.” In contrast, Carl Olofson points out that, “[SQL Server] lags in its ability to store and manipulate XML data.”

Akaza has not developed XML applications using SQL Server; therefore, we cannot judge the level of SQL Server's support of XML and cannot provide advice in that regard.

B. Review of Oracle 9i

1. Security

Security is the built-in feature in Oracle, and it provides multi-level security. Only Oracle can claim 15 security evaluations from independent bodies such as TCSEC, ITSEC, and the Common Criteria⁷². MS SQL Server at the same time has only one. Oracle database was not affected by the latest SQL worms such as Slammer or Code Red. In general Unix Operating System is more secure than Windows.

From our experience at Akaza, Oracle's security feature has performed very well, and we are confident that among all databases we work with Oracle's security features are the most robust and extensible.

⁷¹ http://www.oracle.com/ip/dep/otn/database/oracle9i/db_sql_tp_askms.html

⁷² http://www.oracle.com/ip/index.html?se_home.html

2. Performance

Ability to integrate with third-party products, technology is well supported in the target environment, and its proven ability to support large volumes of transactions⁷³. During the tests conducted by eWeek, Oracle was the fastest database among most popular and widely used ones. According to the tests, “The Oracle and MySQL drivers had the best combination of a complete JDBC feature set and stability”. However, “Oracle9i was the most difficult to tune because it has so many separate memory caches that can be adjusted.” Both Oracle and MySQL have been steady throughout the tests and run the application during all 8 hours.

3. Scalability/Interoperability

Unlike Microsoft SQL Server 2000, Oracle scaled vertically and horizontally, meaning that it can perform well across multiple servers and using different platforms: “Oracle9i Database offers customers the choice of traditional vertical scaling and the option to scale out horizontally with Oracle9i Real Applications clusters. Scale-out architecture delivers major savings based on the ever-increasing power of commodity servers. And because an Oracle9i Real Applications clustered is administered in much the same way a single servers is administration costs are minimized while performance, scalability and availability are maximized.”⁷⁴ Oracle Database supports all known platforms, not only the Windows-based platforms. General minimum requirements are slightly higher than those for SQL Server.

4. Easy of Use, Programming and Configuration

It can be somewhat challenging to configure Oracle for the first time. However, once configuration is done, the database runs smoothly. Also Oracle has many features that can be tuned via start-up parameters; on the other hand it is less friendly. Oracle uses language called PL/SQL which, unlike SQL Server's T/SQL, supports such features as arrays, Java methods, Bitmap indexes, Object tables and third-generation language routines. PL/SQL is a more powerful language than T/SQL, but at the same time harder to learn. Therefore learning curve is bigger. Recently, Oracle has provided graphical user interfaces to manage and configure the databases as well as user roles more easily.

At Akaza, we have used Oracle for a several projects and have acquired a great deal of expertise in many aspects of development, performance tuning and security. The most recent version of Oracle has a graphical user interface, therefore it is now easier to configure the database server, and we believe that with each new release usability will be improved.

5. Cost of Purchase/Upgrades/Maintenance

Oracle is generally designed for high volume production settings and it runs on multiple platforms, hence its license can be relatively expensive. Oracle Standard Edition starts around \$15,000. The Enterprise Edition costs \$40,000. Adding Management Tools, Advanced Security features and Business Intelligence features it goes up to \$96,000. Maintenance and Upgrades can be costly. However, many academic institutions have licenses that allow them to procure Oracle and future upgrades at nominal or no cost.

6. Stability/Availability/Robustness

Oracle website claims that database server has 100% availability time and is available 24/7. Most studies that have been completed prove Oracle's high availability, stating that it is “best when 24 x 7 x 365 operations required”⁷⁵.

Based on our experience at Akaza, there have not been any known cases when Oracle was down. Database availability is always very high.

⁷³ BACS LTD Chooses Oracle9i Technology to Support New Payment-Processing System

<http://www.oracle.com/customers/profiles/PROFILE7938.HTML>

⁷⁴ Oracle Database compared to MS SQL Server: http://www.oracle.com/ip/dep/otn/database/oracle9i/db_sql_tp_askms.html

⁷⁵ <http://formation.espacecourbe.com/choosing/>

7. XML Support

Oracle9i JDeveloper provides a complete, highly productive XML development solution with everything from seamless dynamic XML generation to wizard-driven XML database updates. Key features include:

- § An integrated XML Developer's Kit to parse, transform and validate XML
- § Instant Java, XML and SQL productivity with declarative re-entrant wizards throughout
- § Out of the box support for the latest XML and Web Service Standards
- § XML Schema driven editing built into Oracle9i JDeveloper

At Akaza Research, we used XML along with Oracle for multiple projects and there were no serious compatibility issues. We would strongly recommend using this feature of the software.

C. Review of PostgreSQL

1. Security

Security problems are well documented in PostgreSQL as in other open-source software. Since everyone can have an access to the code, there is a bigger chance of finding security issues and managing attacks on the database server. However, the developers continuously issues patches to known security holes.

2. Performance

Performs very well and steady with the variety of interfaces such as ODBC, JDBC, C/C++, Perl, PHP, Python, Tcl. However, according to Patrick Turmel it performs slowly during large data loads. According to Kelly Computer Resources, PostgreSQL runs in two modes. Normal fsync guarantees that if the OS crashes or loses power in the next few seconds, all your data is safely stored on disk. In this mode, [it is] slower than most commercial databases, partly because few of them do such conservative flushing to disk in their default modes. In no-fsync mode, [it is] usually faster than commercial databases, though in this mode, an OS crash could cause data corruption. "

3. Scalability/Interoperability

Works on every platform but performs better on Unix-based machines.

4. Easy of Use, Programming and Configuration

As for every database system that runs on UNIX machine, PostgreSQL is tricky to get up and running. Among all open-source databases it has the most available functions and procedures.

5. Cost of Purchase/Upgrades/Maintenance

Since PostgreSQL is open source database, it is freely available. No purchase is required. Limited technical support is provided through the mailing list but commercial support is also available. However, its Total Cost of Ownership can be substantial since technical support will require specialized DBAs.

6. Stability/Availability/Robustness

Because of its open-source nature these features are hard to test and there could be substantial variations in them depending on such factors as Operating System, Language or Server Software.

7. XML Support

Commercial databases have for some time now featured various degrees of XML support, while open source databases have lagged behind. There is an XML DOM implementation in PostgreSQL called DBDOM which is distributed under an Apache-style license but its performance has not been tested really well and no known studies are present at this time.

Summary of Key Tradeoffs among Databases

A. MS SQL Server 2000

Advantages:

- 1) Best if runs in all-Microsoft environment (ASP or ASP.NET and Microsoft IIS)
- 2) User-friendly interface, easy to install and to manage.
- 3) Many extra features (though unneeded sometimes)
- 4) Integration with Microsoft office software

Limitations:

- 1) Poor and limited performance with JDBC driver
- 2) Security problems
- 3) XML support is in question
- 4) Tied to Microsoft environment and doesn't scale across multiple machines.
- 5) MS SQL Server language T/SQL is less developed and functioned than PL/SQL

B. Oracle 9i

Advantages:

- 1) Works across multiple platforms and multiple machines
- 2) More robust and developed
- 3) High Availability
- 4) Well established XML Support
- 5) Best for high volumes of data storage, including better multimedia support.

Limitations:

- 1) Could be challenging to install and configure.
- 2) High learning curve.

C. PostgreSQL

Advantages:

- 1) Open source, therefore it is free
- 2) Scales across all platforms.
- 3) Works with the variety of interfaces
- 4) Most developed among all open source databases

Limitations:

- 1) Poor performance on loading large data sets
- 2) Security not as robust as Oracle
- 3) Limited XML support

APPENDIX II: Database Security Case Study

Slammer Virus Attack on MS SQL Server 2000

The Slammer worm (also known as Sapphire) was detected on January 25, 2003; however, there have been reports of the worm being spotted on January 20th. The worm generates massive amounts of network packets, overloading servers and routers and slowing down network traffic. According to many reports⁷⁶, as many as 5 of the 13 internet root nameservers were down because of this during Saturday the 25th. The worm only infects computers running Microsoft SQL 2000 or MSDE 2000. It is not a mass-mailer: it does not send any emails. It only spreads as an in-memory process and never writes itself to the hard drive. The worm uses port 1434 to exploit a buffer overflow in MS SQL server. An infected machine can be cleaned by simply rebooting the machine. However, it will soon be re-infected if the machine is connected to the network without applying relevant patches to MS SQL Server.

While Slammer did not contain a malicious payload, it caused considerable harm simply by overloading networks and taking database servers out of operation⁷⁷. Many individual sites lost connectivity as their access bandwidth was saturated by local copies of the worm and there were several reports of Internet backbone disruption. It is important to realize that if the worm had carried a malicious payload, had attacked a more widespread vulnerability, or had targeted a more popular service, the effects would likely have been far more severe.

The worm infected at least 75,000 hosts, perhaps considerably more, and caused network outages and such unforeseen consequences as canceled airline flights, interference with elections, and ATM failures.

⁷⁶ <http://www.f-secure.com/v-descs/mssqlm.shtml>

⁷⁷ <http://www.cs.berkeley.edu/~nweaver/sapphire/>