Research

Design and implementation of microarray gene expression markup language (MAGE-ML)

Paul T Spellman[1], Michael Miller[2], Jason Stewart[3], Charles Troup[4], Ugis Sarkans[5], Steve Chervitz[6], Derek Bernhart[6], Gavin Sherlock[7], Catherine Ball[7], Marc Lepage[8], Marcin Swiatek[9], WL Marks[10], Jason Goncalves[10], Scott Markel[11], Daniel Iordan[10], Mohammadreza Shojatalab[5], Angel Pizarro[12], Joe White[13], Robert Hubley[14], Eric Deutsch[14], Martin Senger[5], Bruce J Aronow[15], Alan Robinson[5], Doug Bassett[2], Christian J Stoeckert Jr[12] and Alvis Brazma[5]

Addresses: [1]Department of Cell and Molecular Biology, University of California at Berkeley, Berkeley, CA 94720-3206, USA. [2]Rosetta Biosoftware, 113th Ave NE, Kirkland, WA 98034, USA. [3]Open Informatics, Arizona St SE, Albuquerque, NM 87108, USA. [4]Bioscience Research - Agilent Technologies, Deer Creek Rd, Palo Alto, CA 94304, USA. [5]European Bioinformatics Institute, EMBL Hinxton Outstation, Cambridge CB10 1SD, UK. [6]Affymetrix, Inc., Vallejo St, Emeryville, CA 94608, USA. [7]Department of Genetics, Stanford University School of Medicine, Stanford, CA 94305-5120, USA. [8]Molecular Mining Corporation, Rideau St, Kingston, ON K7K 2Z8, Canada. [9]Imaging Research Inc., Glenridge Ave, St. Catharines, ON L2S 3A1, Canada. [10]Iobion Informatics LLC, North Torrey Pines Road, La Jolla, CA 92037, USA. [11]LION bioscience Inc., Executive Drive, San Diego, CA 92121, USA. [12]Center for Bioinformatics, University of Pennsylvania, Guardian Drive, Philadelphia, PA 19104, USA. [13]The Institute for Genomic Research, Medical Center Drive, Rockville, MD 20850, USA. [14]Computational Biology, Institute for Systems Biology, North 34th St, Seattle, WA 98103-8904, USA. [15]CHRF, Burnet Ave, University of Cincinnati College of Medicine, Cincinnati, OH 45229, USA.

Correspondence: Paul T Spellman. E-mail: spellman@fruitfly.org

Published: 23 August 2002

Genome ***Biology*** 2002, **3(9)**:research0046.1–0046.9

The electronic version of this article is the complete one and can be found online at http://genomebiology.com/2002/3/9/research/0046

Received: 19 March 2002
Revised: 13 July 2002
Accepted: 18 July 2002

(Print ISSN 1465-6906; Online ISSN 1465-6914)

Abstract

Background: Meaningful exchange of microarray data is currently difficult because it is rare that published data provide sufficient information depth or are even in the same format from one publication to another. Only when data can be easily exchanged will the entire biological community be able to derive the full benefit from such microarray studies.

Results: To this end we have developed three key ingredients towards standardizing the storage and exchange of microarray data. First, we have created a minimal information for the annotation of a microarray experiment (MIAME)-compliant conceptualization of microarray experiments modeled using the unified modeling language (UML) named MAGE-OM (microarray gene expression object model). Second, we have translated MAGE-OM into an XML-based data format, MAGE-ML, to facilitate the exchange of data. Third, some of us are now using MAGE (or its progenitors) in data production settings. Finally, we have developed a freely available software tool kit (MAGE-STK) that eases the integration of MAGE-ML into end users' systems.

Conclusions: MAGE will help microarray data producers and users to exchange information by providing a common platform for data exchange, and MAGE-STK will make the adoption of MAGE easier.

Background

Among the many challenges that microarrays present to both bioinformaticists and biologists, data communication is one of the most significant. This stems from the fact that, unlike biological sequences, microarray data requires data structures that are both multidimensional and varied, and no natural or standard ways to move results between research groups yet exist. This applies to both the underlying gene-expression data and the descriptive biological annotations that provide context for the gene-expression measurements. Hundreds of papers have now been published, but no more than a handful have presented data in the same format, and none has provided adequate contextual information to allow reproduction of experiments.

For the past two years, MGED (the microarray gene expression data group) has been wrestling with standards-based microarray data exchange. Recently MGED has published a specification describing MIAME, the minimal information for the annotation of a microarray experiment [1]. MIAME is based on the consensus of hundreds of participants in the MGED conferences (for more information see [2]) and specifies which data and contextual information should be supplied when a microarray gene-expression dataset is published. Some journals (for example, *Nature*) have begun to endorse or encourage MIAME compliance for papers describing the results of microarray experiments. It is expected (indeed hoped) that in the future, journals and funding agencies will require MIAME-compliant data for continued funding and publishing, respectively.

It is not enough to specify that certain data and accessory information be provided. It is essential, if MIAME is to be useful, that a standard transmission format exists for the data. Many of the authors of this paper have previously responded to this need by developing XML-based data-communication syntaxes for microarray experiments (GEML, gene expression markup language [3] from Rosetta Inpharmatics, and MAML, microarray markup language from MGED).

XML (extensible markup language) is a set of rules whereby new vocabularies may themselves be defined. In some respects it is similar to HTML, in that tags are used to encode information, but in HTML the information is related to the formatting of a document, using a predefined set of tags. In XML, the tags do not indicate how a document should be formatted, but instead provide semantic context to the content of the document. XML vocabularies define their own tags, and thus use XML to hold information in a way such that that information can be understood. Because of this, and the wide support that XML has received since its release as a W3C recommendation in 1998 [4] both GEML and MAML chose XML for encoding microarray data. Usually an XML document is not a stand-alone document, but will refer to another document, called the document type definition, or DTD. The DTD contains a set of rules, or 'declarations', that specify which tags can be used, and what they contain. It is the DTD that we specify in MAGE-ML. XML documents created to use MAGE-ML will refer to this DTD. In response to a request for proposals from the OMG (object management group [5]) for methods of communicating gene-expression data, the designers of GEML and MAML submitted their vocabularies as potential solutions to this problem.

Results

Since the MAML and GEML proposals were submitted to the OMG, members of many groups including MGED, Rosetta, Agilent, and Affymetrix have worked together to design a common data structure for communicating microarray-based gene-expression data that is flexible and robust. This paper describes the results from this effort, which are collectively referred to as MAGE (microarray gene expression), specifically: MAGE-OM, an object model; MAGE-ML, the XML representation of MAGE-OM; and MAGE-STK, a software toolkit that is composed of a suite of open-source [6] software developed to facilitate adoption of MAGE. Through participation in the OMG, MAGE provides a stable specification that many in the community have been waiting for before adopting a data-exchange format for their systems. The full MAGE-OM specification can be found at [7]. General information on using MAGE and detailed information on the STK is available from [8].

The object model: MAGE-OM

MAGE-OM is a data-centric model that contains 132 classes grouped into 17 packages containing, in total, 123 attributes and 223 associations between classes. Classes in our model represent distinct things or events, and each class may have attributes as well as associations to other classes. A list of the MAGE-OM packages and the classes contained within each one are listed in Table 1. The packages are used to organize classes that share a common purpose, for example the Array package contains classes that describe individual arrays, including detailed information on relevant manufacturing processes. The key components of MAGE-OM reflect many of the core requirements of MIAME, specifically: experiment goals and design (Experiment package); biological materials used and description of their creation (BioMaterial package); array design and purpose (ArrayDesign, BioSequence packages); array manufacture (Array package); hybridization, wash, and scan information (BioAssay package); gene-expression data (BioAssayData package).

Other utility packages support requirements shared by the above components, specifically information on people and organizations, protocols used, simple annotations, free-text descriptions, and the ability to specify links to predefined ontologies such as those provided by MGED [2].

Table 1

Packages and classes

Logical
Extendable
Describable
Identifiable
NameValueType

AuditAndSecurity
Audit
Contact
Organization
Person
Security
SecurityGroup

Description
Description
OntologyEntry
ExternalReference
DatabaseEntry
Database

BioSequence
BioSequence
SeqFeature
SeqFeatureLocation
SequencePosition
CompositePosition
ReporterPosition

ArrayDesign
ArrayDesign
CompositeGroup
ReporterGroup
DesignElementGroup
FeatureGroup
PhysicalArrayDesign
ZoneGroup
ZoneLayout
Zone

DesignElement
DesignElement
Feature
Reporter
CompositeSequence
Position
FeatureLocation
CompositeCompositeMap
ReporterCompositeMap
FeatureReporterMap
CompositePosition
ReporterPosition
FeatureInformation
MismatchInformation

Array
ArrayManufacture
Array
ArrayGroup
Fiducial
ArrayManufactureDeviation
FeatureDefect
ZoneDefect
PositionDelta
ManufactureLIMS
ManufactureLIMSBioMaterial

BioMaterial
BioMaterial
BioSource
BioSample
LabeledExtract
Treatment
BioMaterialMeasurement
CompoundMeasurement
Compound

BioAssay
BioAssay
PhysicalBioAssay
BioAssayTreatment
ImageAcquisition
BioAssayCreation
Hybridization
Image
Channel
FeatureExtraction
MeasuredBioAssay
DerivedBioAssay

BioAssayData
BioAssayData
Transformation
DerivedBioAssayData
MeasuredBioAssayData
BioAssayDimension
QuantitationTypeDimension
DesignElementDimension
FeatureDimension
ReporterDimension
CompositeSequenceDimension
BioDataCube
BioDataTuples
BioAssayDatum
QuantitationTypeMapping
QuantitationTypeMap
DesignElementMapping
DesignElementMap
BioAssayMapping
BioAssayMap

Experiment
Experiment
ExperimentDesign
ExperimentFactor
FactorValue

HigherLevelAnalysis
BioAssayDataCluster
Node
NodeContents
NodeValue

Measurement
Measurement
Unit
TemperatureUnit
MassUnit
VolumeUnit
DistanceUnit
TimeUnit
QuantityUnit
ConcentrationUnit

Protocol
Protocol
Hardware
Software
Parameterizable
Parameter
ProtocolApplication
HardwareApplication
SoftwareApplication
ParameterizableApplication
ParameterValue

QuantitationType
QuantitationType
StandardQuantitationType
SpecializedQuantitationType
PresentAbsent
MeasuredSignal
DerivedSignal
Ratio
ConfidenceIndicator
Pvalue
Error
ExpectedValue

Bold indicates packages and italics indicates abstract classes.

Mapping of biological experiments to MAGE-OM

MAGE-OM provides a structure for the logical flow of experiments using the six requirements listed above. While the MAGE model is not a laboratory information management system (LIMS), laboratory information does have a critical role in understanding microarray data, and much of this information is accounted for in our model (for example, protocols and sources for clones used in manufacturing

microarrays). A conceptualized model of the biological workflow modified from the one in the OMG submission document is presented in Figure 1. The workflow is based on two parallel processes - the manufacture of microarrays, and the generation of biological samples - that come together in a hybridization. A detailed breakdown of the model and its relation to the workflow are presented below.

Descriptions and protocols

During the design of MAGE we recognized that despite all efforts it would be impossible to create a model that would be completely expressive for all microarray experiments. Thus, to allow extensible experiment documentation, several classes that allow additional information to be associated with other entities in the model were included. Two of these classes are Annotation and Description; all model components can be annotated by association with an instance of the Annotation class, and many can be described. Annotations are simple mechanisms for storing local parameters, using name/value/type (NVT) associations. Descriptions are a method of providing extra information allowing both NVTs and free text: for example, MAGE-OM allows users to describe protocols (as well as experimental materials, see below) using free-text descriptions. From a database and computational perspective, structured annotations are preferable because they facilitate searches and automated data processing (see below).

There are three other key components of the Description package: Databases, DatabaseEntries, and OntologyEntries. Databases are references to external sources of information such as GenBank or the *Saccharomyces* Genome Database, whereas DatabaseEntries refer to individual records in those databases. OntologyEntries are a mechanism by which groups can use commonly defined terms to aid communication. In designing MAGE it was recognized that it would be impossible to specify fully all the many possible parameters and their allowable values, but it was important that a flexible and extensible mechanism existed to support this possibility. Thus, the model contains OntologyEntries, whose roles are named to fit their purpose; for example, BioSequences has an OntologyEntry named 'species', which should be used to refer to an entry in the NCBI taxonomy database.

The MAGE model provides a rich mechanism for describing protocols and their use. Protocols can use equipment

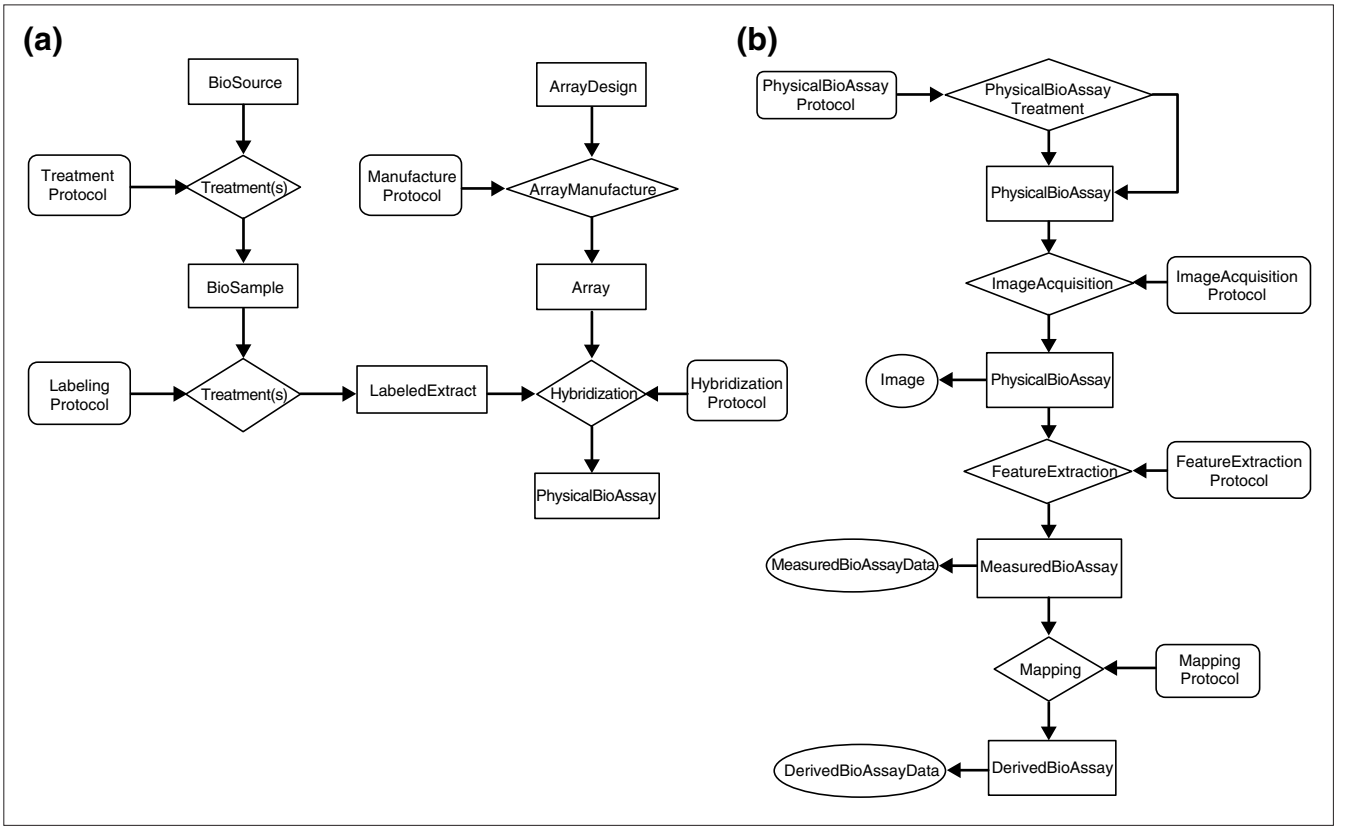

Figure 1
A workflow diagram showing the various steps of microarray analysis. Rectangles represent physical things, diamonds represent events, ovals represent data, and rounded rectangles represent methods. **(a)** Workflows that lead up to a hybridized microarray. Two convergent paths (microarray production and sample preparation) are joined by the hybridization event to create a PhysicalBioAssay. **(b)** BioAssay manipulations.

(hardware) and software, as well as have a list of parameters whose values can change between individual uses of the protocol. The use of a protocol, termed a ProtocolApplication, involves specifying the values for each of the protocol parameters, as well as setting the protocol parameters for any hardware or software used. To encode a PCR reaction using this method the protocol could describe the thermal cycling conditions and the make/model of the instrument, while the protocol application might give the sequences of the primers and the serial number of the thermal cycler. Another example would be to define the feature-extraction protocol whereby data were extracted from a scanned microarray image, which would have a reference to a piece of software. The software might have parameters for the layout of the features (spacing and position) and the background calculation method.

Array information

Three MAGE packages in the object model, ArrayDesign, Array, and DesignElement, contain information regarding the design, manufacture and contents of microarrays. The DesignElement package is arguably the most complex of the three, allowing users to specify information about the biological materials deposited on an array. The ArrayDesign package stores the intended pattern of individual array elements, while the Array package records information on the actual events that produced arrays. ArrayDesigns allow the user to specify the protocol used, a relevant contact, the grids structure, and which groups of DesignElements are present in the design.

There are three classes of DesignElements: Features, Reporters and CompositeSequences. Features represent a unique address on the array, specified either using Cartesian or logical coordinates (zone/sector, row, column). It is important to note that in MAGE, Features do not possess substantial biological information; only Reporters and CompositeSequences have associations to BioSequenes. Reporters are the first level of DesignElement abstraction, and correspond to one or more features. A Reporter models a physical sequence, and thus if exactly the same biological sequence is spotted on an array twice, as two Features, both of these Features are represented by the same Reporter. However, two expressed sequence tag (EST) clones mapping to the same UniGene cluster are represented by two distinct Reporters, as would be two distinct oligonucleotides which map to the same open reading frame.

CompositeSequences are the highest level of abstraction, allowing multiple Reporters or multiple CompositeSequences to be combined so that data can be mapped to DNA sequences that are longer than the lengths of individual Reporters. Thus, the two previously mentioned ESTs that map to the same UniGene cluster can be grouped in a CompositeSequence that represents that UniGene cluster. It is important to note that the CompositeSequence with which a Reporter may be associated is dynamic, as gene predictions in an organism become more refined. In addition, the biological annotation associated with a CompositeSequence is also dynamic, as our knowledge of the functions of genes, and the processes in which they participate, changes over time. A CompositeSequence should not be merely thought of as a 'gene'. There are many different types of sequence features in a genome, such as centromeres, telomeres, intergenic regions, RNA genes and protein-coding genes. Even a 'gene' itself has several components, such as a promoter, and introns and exons, as well as potential signals in the 3´-untranslated region. A CompositeSequence can theoretically be used to represent any of these, or even a chromosome itself.

The Array package stores information on arrays created on the basis of an ArrayDesign. This includes the manufacturing protocols, contacts, and details of the exact materials used for each Feature. It is important to note that the materials are actually BioMaterials (described below), which allows them to be very detailed; examples include PCR reactions based on different identifiable cDNA clones. Position changes and other feature defects can be recorded for each array. To accommodate some manufacturing processes, arrays may be part of an ArrayGroup (several distinct arrays that are synthesized and distributed as a single entity). The ArrayGroup also records the locations of signifying marks (fiducials) and details about the array's substrate.

Preparation of experimental materials, hybridizations, and scans

Experimental samples from the laboratory are termed BioMaterials in the MAGE-OM, with key subclasses being BioSource, BioSample, and LabeledExtract. The BioSource is used to designate the innate (or starting) properties of a sample, such as genotype, age, species, and disease state. Each of these properties is set by the associations to OntologyEntries. A BioMaterial is derived from one or more BioMaterials through Treatment events, which also specify the protocol and amount(s) of the BioMaterial(s) used. The treatment provides an association to the action OntologyEntry. Examples of common terms in this ontology might be: add, centrifuge, and incubate. Each treatment also has an optional actionMeasurement that would allow relevant values to be stored. Together, action and actionMeasurement might be used to specify "incubate 10 minutes". BioMaterials can be designated as being of a certain type (an OntologyEntry) such as the MIAME concept of an Extract, so that the role in the laboratory can be specified. The final BioMaterial is the LabeledExtract, which is also created by a treatment event from other BioMaterials. The LabeledExtract has an association to labels (compounds that will be used to measure the abundance of components of the LabeledExtract).

In the workflow above, an Array and one or more BioMaterials (for two-color microarrays this would be two LabeledExtracts) are combined to create a PhysicalBioAssay. The

BioEvent model used by MAGE-OM allows for a BioEvent to be applied to a PhysicalBioAssay, to create a new PhysicalBioAssay, for example, a washing event can be applied to the above PhysicalBioAssay to create a new PhysicalBioAssay. One specialized BioAssayTreatment is the ImageAcquisition, the product of which has associations to one or more Images. A MeasuredBioAssay is generated by the FeatureExtraction event operating on a PhysicalBioAssay. The FeatureExtraction is Software guided and the MeasuredBioAssay has an association with the numeric data that are produced (that is, the tab-delimited data output by the scanner software). MeasuredBioAssays can be processed or combined by a Transformation producing a DerivedBioAssay. Again, the DerivedBioAssay is not numeric data, its data are obtained through an association to DerivedBioAssayData.

Data model and storage

The data model is perhaps the most complex aspect of MAGE-OM. We view the data as a three-dimensional matrix (or cube) of values whose axes are labeled by DesignElements (the 'genes'), BioAssays ('experimental samples'), and QuantitationTypes (parameters from the scanning software). Figure 2a shows data for four DesignElements, three QuantitationTypes and a single BioAssay, which is similar to the table of data generally presented by a FeatureExtraction event. The QuantitationTypes shown include Channel 1 Foreground and Background, Channel 2 Foreground and Background, and the ratio between the background subtracted intensities of Channel 2 over Channel 1.

QuantitationTypes are the types of data that are communicated in microarray experiments and can be either Standard or Specialized types. StandardQuantitationTypes were designed to allow third parties to understand the structure and meaning of the gene-expression data. Four of the StandardQuantitationTypes are MeasuredSignal (that is, intensity), DerivedSignal (background subtracted intensity), Ratio (the result of dividing one signal by another), and PresentAbsent (enumerated evaluation of presence). There are three more StandardQuantitationTypes: PValue, Error (standard deviation), and ExpectedValue, which are ConfidenceIndicators, meaning that their values modify or describe another QuantitationType. For example, if a gene's relative expression between two BioMaterials is known by another experimental method to be unchanged, then the ExpectedValue of the Ratio should be 1.00, while the Ratio itself might be 1.08. This ability allows control values to be included in the gene-expression data so others can evaluate the accuracy and reliability of the measurements provided. Each StandardQuantitationType has an optional association to Channel and a designator indicating whether the value refers to the background or foreground. The SpecializedQuantitationTypes are user defined; possible examples include the number of pixels in each feature, or the pixel-by-pixel correlation between two channels within a feature.

The matrix of values is represented in the model as either a cube of values (BioAssayDataCube), or as value coordinate tuples (BioDataTuples). Both of these data sources share BioAssayDimensions that identify all of the DesignElements, QuantitationTypes, or BioAssays used. Each of the dimensions of the cube specifies the order of the values along one axis (Figure 2b).

Experiments

MAGE-OM's highest-level object is the Experiment, a collection of results for one or more BioAssay(s), the intent of which is to communicate the biological properties tested. Experiments have an ExperimentDesign, which records the following key MIAME requirements: replicate; quality and data-processing information (in free-text form); OntologyEntries describing the type of experiment; and a set of ExperimentalFactors (the parameters of the experiment). For example, a simple gene-expression time course could vary cell density and glucose concentration over time; these would be the ExperimentalFactors. Each ExperimentalFactor also has an association to FactorValues, which hold the values for each of the factors for each BioAssay in the Experiment. This will facilitate queries of gene-expression repositories where a user might wish to find all experiments that varied the dosage of exposure to a given drug. The Experiment also has a set of Providers who are the relevant Contacts for the Experiment (the experimenters themselves).

Data analysis

The Experiment also has associations to analyses (HigherLevelAnalyses) that go beyond the recorded data. Typical examples might include clustered results (from hierarchical or k-means clustering) or results from self-organizing maps. More complex HigherLevelAnalyses might include predictions of gene function, experiment class (for example, tumor classification), or associated regulatory sequences (transcription factor binding sites).

At present the model only contains a limited framework supporting node- and tree-based clusters. Each clustering has an association to the data from which the results were generated as well as one or more nodes. Each node can contain other nodes to create a tree or it can contain one or more dimensions, allowing the node to identify which of the matrix indices are clustered together (for example, a node might contain a DesignElementDimension with two genes).

MAGE-ML

A few simple rules were used to translate MAGE-OM into the DTD named MAGE-ML. First, each class in the object model is represented as an element with an attribute list matching the attributes of the class. Second, for each association of that class, a daughter element having the role's name with 'assn' appended to the end is created. Further, if the association is by reference 'ref' is appended and if the cardinality of the association is greater than one 'list' is

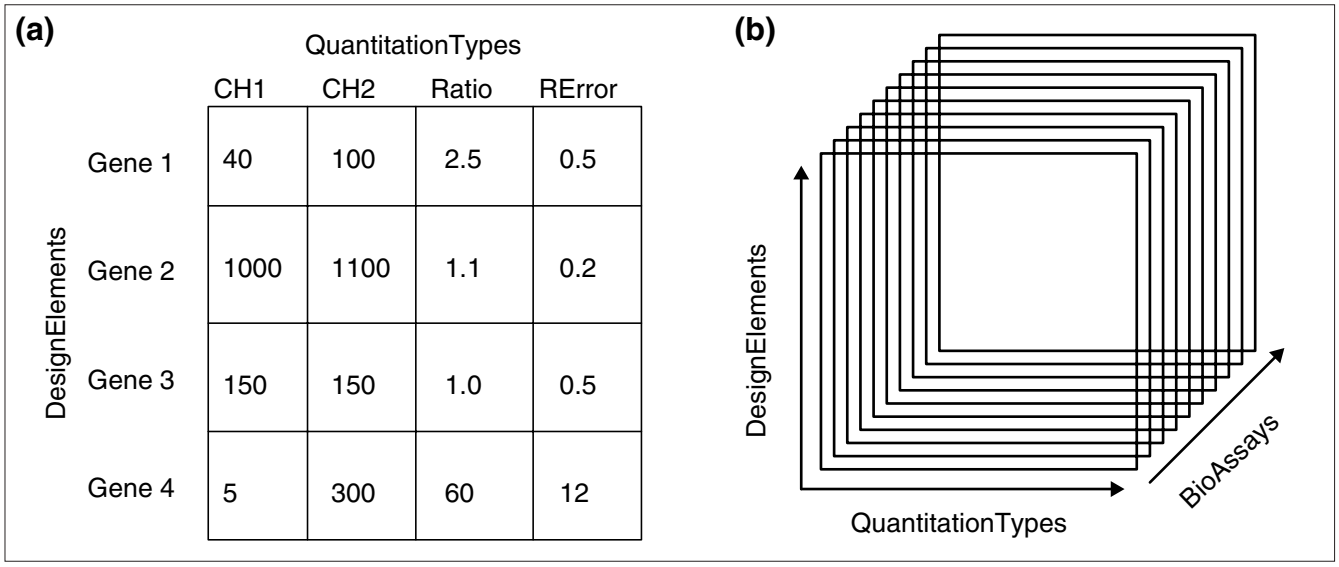

Figure 2
The BioDataCube. BioDataCubes are composed of a matrix of values. **(a)** A two-dimensional slice of a BioDataCube for a single Bioassay. Each combination of DesignElements and QuantitationTypes is allowed a value. **(b)** The cube of values is a set of slices, in this view one slice for each BioAssay.

appended. For example, in MAGE-OM, Person has an association, 'affiliation', by reference to an Organization. In the XML this would appear as:

```
<Person identifier="Person1" name="John Doe">
    <Affiliation_assnref>
        <Organization_ref identifier="ABC Inc." />
    </Affiliation_assnref>
</Person>
```

We have adopted this strict naming scheme so that MAGE-ML is predictable and so that future additions and extensions to MAGE-ML will be compatible. Several special elements are also created: one for the model (MAGE-ML); one for each of the packages in the model; and a list element that each identifiable class within a package receives and that serves to organize each instances of that class. The packages ensure that MAGE-ML is modular, so that, for example, related ArrayDesigns, DesignElements, and BioSequences can be stored in different documents.

Production uses of MAGE

There is substantial evidence that MAGE is a useful specification. First, Rosetta Biosoftware has used GEML (a MAGE-ML predecessor) as an internal data-communication standard for over two years, primarily as an intermediate step in their data pipeline. The Rosetta pipeline principally included data associated with array design, printing, scanning and feature extraction. Their XML format handles many technology types, including two-channel ratio data and single-channel intensity data. At a minimum, GEML has shown that an XML format can work very well for a disconnected high-throughput internal pipeline. It showed that the large amount of data in gene-expression experiments can be effectively encoded in an XML format and also showed that a single experiment can be spread through several documents and/or data stores and linked using common identifiers.

More recently, the European Bioinformatics Institute (EBI) has begun using MAGE as the basis for the ArrayExpress gene-expression database [9]. ArrayExpress is a relational implementation of MAGE-OM that can import, and in the near future, export MAGE-ML documents. This project has proved that there is more value to MAGE-OM than just an easier-to-read MAGE-ML specification; database schema, various software pieces for data loading and access, as well as templates for web pages were automatically generated from MAGE-OM. Although this effort is just beginning, a number of submissions have been successfully entered into the database.

Affymetrix has also implemented MAGE-compliant software, particularly an exporter application programmers interface (API) in a software component called the 'Expression Data Access Component Exporter' or 'EDAC-Exporter' that will export Affymetrix experiment result data files to MAGE-ML-compliant XML documents. Affymetrix is further providing the EBI's ArrayExpress with MAGE-ML files with ArrayDesign, DesignElement, and BioSequence information to facilitate submissions of experimental data

from Affymetrix GeneChips and queries of this data within ArrayExpress.

Finally the National Cancer Institute (NCI) has over 150 experiments, including oligo, spotted array, CGH (comparative genome hybridization), and SAGE (serial analysis of gene expression) experiments in their database (which is based on MAGE). Currently, the NCI database can generate MAGE-ML documents for all submitted experiments.

The software toolkit : MAGE-STK

We do not expect that individuals will access and use MAGE-ML directly. Instead, we have developed a suite of software tools based on MAGE-OM that is collectively called the MAGE-STK. These tools define an API to MAGE-OM. The suite currently supports three implementations: MAGE-Perl, MAGE-Java, and MAGE-C++. Each of the APIs use similarly named methods and classes to provide access to their objects. The goal of MAGE-STK is to provide an intermediate object layer that can then be used to export data to MAGE-ML, to store data in a persistent data store such as a relational database, or as input to software-analysis tools.

Currently, each API provides a MAGE-ML reader and a MAGE-ML writer: the reader packages load a set of MAGE-ML documents into the API, creating a collection of objects; the writer packages create a MAGE-ML document from a collection of API objects.

Because there is no public adaptation of MAGE-OM to a relational database schema, the APIs must be mapped to and from a site's local model (that is, their database schema) before use. Once this is done, each group will be able to share (and receive) data from other MAGE users in an unambiguous and predictable format - MAGE-ML. Projects are currently underway which map MAGE-OM to a relational database schema (see, for example, ArrayExpress [10]), and as soon as such a public database schema exists, the APIs will provide a default set of mappings to that schema.

As addressed in MIAME, one of the current shortcomings of public microarray data is the lack of sufficient contextual information. One of the next goals of the MAGE-STK project is to develop tools for annotating MAGE data. The overriding goal is to create tools that make it less burdensome on the experimenter to annotate his or her experiments. MIAME currently specifies a large amount of annotation as required for the 'minimal' amount. MIAME will live or die by whether it is reasonable to provide this level of annotation, not whether it is unreasonable not to. Thus tools that allow a user rapidly and accurately to annotate microarrays, or groups of microarrays, or even related experiments, are of paramount importance.

All MAGE software is open source, available under the Massachusetts Institute of Technology (MIT) license which allows unrestricted use of the MAGE-STK for any purpose, academic or commercial. All MAGE software and documentation can be found at [11].

Discussion

Future of MAGE

MAGE-ML has been designed to be flexible so that it can be used in a wide variety of technical settings (spotted two-color cDNA arrays, Affymetrix arrays, and so on). These extend well beyond gene-expression experiments, for without modification, data from all DNA microarray experiments and technologies that we know of can be stored. Examples of some of the uses and technologies supported include: one- and two-color spotted arrays (cDNAs, PCR products, or oligonucleotides) on glass or nylon, *in situ* synthesized oligonucleotides (Affymetrix/photolithography, ink-jet), RNA abundance (gene expression, polysome profiling), and DNA abundance (array-CGH, chromatin immunoprecipitation, genotyping). We believe that with, at most, modest changes, MAGE-ML can support experiments that use arrays made of proteins, cells, or tissues.

One of the main weaknesses of MAGE-OM is the limited support it provides for data analysis. Our goal is to enhance the current capabilities in the next version substantially by providing explicit support for different clustering and analysis methods. Another development initiative is to support conclusions based on the microarray data, for example the ability to indicate sequence motifs or biological effects (such as mortality) associated with individual (or sets of) nodes in a cluster.

MAGE-ML will adapt to the changes that occur because the developers are actively committed to maintaining useful standards. We appreciate all comments about MAGE-ML. MAGE-ML has a moderated mailing list at mged-mage@lists.sourceforge.net, which is available as a mechanism for announcements, instructive material, and general discussion.

Interaction of MAGE with other projects

The success of this project largely depends on the development, availability, and use of ontological terms not defined in the MAGE project. We expect some of these to be designed by MGED (those covering the general properties of microarrays, and biological sciences) but some will certainly be developed outside the MGED umbrella. We are actively interested in this problem and look forward to interacting with these development efforts. For information on the MGED ontologies please see [12].

One of the prime motivations for developing MAGE-ML was to encourage the analysis of microarray/gene-expression data by individuals other than the original authors. The first key step in this process is the development of databases

(public and private) to warehouse the experimental results. Such databases will then allow MIAME-compliant data to be retrieved as MAGE-ML documents. Many of us are actively involved in developing ArrayExpress - a repository of gene-expression data hosted by the EBI, while others represent companies and academics developing laboratory or institutional databases (Affymetrix, Iobion, and Rosetta, University of Pennsylvania, GeneX). We believe that the efforts presented here will also allow software vendors to streamline the analysis of gene-expression data, thus removing the current communication impasse that is so common.

Acknowledgements

We would like to thank all the members of MGED for advice and encouragement, Agilent, Rosetta, Molecular Mining, Imaging Research, Iobion, and TIGR for developer support, Affymetrix and Iobion for financial contributions. C.S. was funded by grant DE-FG02-00ER62893 from the DOE. P.T.S. was supported by an NSF biocomputing post-doctoral fellowship.

References

1. Brazma A, Hingamp P, Quackenbush J, Sherlock G, Spellman P, Stoeckert C, Aach J, Ansorge W, Ball CA, Causton HC, *et al.*: **Minimum information about a microarray experiment (MIAME) - toward standards for microarray data.** *Nat Genet* 2001, **29:**365-371.
2. **MGED - Microarray Gene Expression Data Society** [http://www.mged.org]
3. **Gene Expression Markup Language (GEML)** [http://www.rosettabio.com/products/conductor/geml/default.htm]
4. **Extensible Markup Language (XML) 1.0 (Second Edition)** [http://www.w3c.org/TR/2000/REC-xml-20001006]
5. **OMG - Object management group** [http://www.omg.org]
6. **Open Source Initiative** [http://www.opensource.org]
7. **OMG TC Work in progress** [http://www.omg.org/techprocess/meetings/schedule/Gene_Expression_RFP.html]
8. **MicroArray and Gene Expression - MAGE** [http://www.mged.org/Workgroups/MAGE/mage.html]
9. Brazma A, Sarkans U, Robinson A, Vilo J, Vingron M, Hoheisel J, Fellenberg K: **Microarray data representation, annotation and storage.** In *Advances in Biochemical Engineering/Biotechnology*. Edited by Hoheisel J. Heidelberg: Springer; 2002, 77:113-139.
10. **ArrayExpress** [http://www.ebi.ac.uk/arrayexpress/]
11. **MGED Software – MAGEstk: The MAGE Software Toolkit** [http://www.mged.org/Workgroups/MAGE/magestk.html]
12. **Microarray Gene Expression Data (MGED) Society Ontology Working Group (OWG)** [http://www.cbil.upenn.edu/Ontology/MGED_ontology.html]