Importing MAGE-ML format microarray data into BioConductor

Steffen Durinck1, *, Joke Allemeersch1, Vincent J. Carey2, Yves Moreau1,3 and Bart De Moor1

1Department of Electronical Engineering, ESAT-SCD, K.U. Leuven, Kasteelpark Arenberg 10, 3001, Leuven-Heverlee, Belgium, 2Channing Laboratory, Brigham and Women’s Hospital, 75 Francis Street, Boston, MA 02115, USA and 3On leave at Center for Biological Sequence Analysis, BioCentrum, Technical University of Denmark, Kemitorvet, Building 208, 2800, Lyngby, Denmark

Received on March 1, 2004; revised on May 17, 2004; accepted on June 29, 2004
Advance Access publication July 9, 2004

Abstract

The microarray gene expression markup language (MAGE-ML) is a widely used XML (eXtensible Markup Language) standard for describing and exchanging information about microarray experiments. It can describe microarray designs, microarray experiment designs, gene expression data and data analysis results. We describe RMAGEML, a new Bioconductor package that provides a link between cDNA microarray data stored in MAGE-ML format and the Bioconductor framework for preprocessing, visualization and analysis of microarray experiments.

Contact: joke.allemeersch@esat.kuleuven.ac.be

Introduction

Microarray data is generated in many different formats and often lacks standardized annotation and documentation. The MIAME (minimum information about a microarray experiment) guidelines define a set of fields for describing and annotating microarray data (Brazma et al., 2001). Major journals, such as Nature, Cell and The Lancet have adopted these guidelines and have made submission of microarray expression data, coupled with MIAME-compliant documentation, compulsory for publication.

The MIAME recommendations define an information set, but do not address the formatting and exchangeability of microarray data. Formatting and exchangeability issues have been confronted in the microarray gene expression markup language (MAGE-ML), which is an XML (eXtensible Markup Language) standard for serializing data structured according to the MAGE object model (MAGE-OM) (Spellman et al., 2002). This data model incorporates the MIAME information and addresses many aspects of microarray experiment documentation and data encoding that are not covered by MIAME.

Bioconductor is an open source project that provides a framework for the statistical analysis of genomic data in R (Ihaka and Gentleman, 1996; Gentleman and Carey, 2003; Dudoit and Yang, 2003; Irizarry et al., 2003). Our Bioconductor package RMAGEML extracts information from MAGE-ML documents for cDNA microarray experiments and maps this information to Bioconductor R objects for the analysis of cDNA microarrays. The RMAGEML package implements a three-tiered architecture, transforming MAGE-ML to R objects via middleware that interfaces R to a Java software toolkit created expressly for working with MAGE-ML documents. The current version of RMAGEML transforms documents and data for cDNA experiments; work is in progress for a wider variety of microarray platforms.

Description

MAGE-ML is derived from the MAGE-OM model, which is an object model specification standardized by the Object Management Group (OMG) and maintained by the MGED Society. MAGE-OM consists of several classes of data structures. Examples of classes include BioSequence, ArrayDesign, Array, Protocol and AuditAndSecurity. The MGED society supplements the MAGE-OM with Software ToolKits (MAGEstk) in Java and Perl that specify data structure classes for programming within MAGE-OM, and serialization and deserialization between MAGE-OM structures and MAGE-ML.

The Bioconductor RMAGEML package is written in R and Java, and makes use of the Java-MAGEstk (http://mged.sourceforge.net/software/MAGEstk.php) application programming interface via the SJava interface (http://www.omegahat.org). SJava facilitates creation and manipulation of Java classes and methods in R as references.

* To whom correspondence should be addressed.
The overall architecture of RMAGEML minimizes the involvement of R with details of XML processing and maximizes the use of MGED-sponsored software for coupling data structure and serialization processes to the MAGE-OM and MAGE-ML specifications. The RMAGEML package currently allows import to two BioConductor data structures: `limma::RGlist` and `marrayClasses::marrayRaw`.

Information from different MAGE-ML packages is needed to create these objects. The `DesignElement` package contains a mapping of `Features` (which are the actual locations on the array) to `Reporters` (what is present at those locations). This package also provides a mapping from `Reporters` to their corresponding `BioSequence` references. These `BioSequences` are characterized by their name and database entries in the `BioSequence` package, which are mapped to the genes and `maGnames` slots of `limma` and `marray`, respectively. The `ArrayDesign` package contains information on the layout of the array. From this package, we can derive the position of each `Feature` on the array in terms of `Zone` (block) and row and column within each `Zone`. This layout information is mapped to the `genes` slot of `limma` and a `maLayout` object of `marray`. The `BioAssay` package describes the different steps in the microarray experiment. The `BioMaterial` package describes the sample source and how it is labeled, this is mapped to a `maTargets` object of `marray` and the `targets` slot of `limma`. Finally, the `BioAssayData` package describes the feature references that were assayed and the available `QuantitationTypes`. It also contains the `BioDataCube`, which is a three-dimensional (3D) matrix that stores the actual intensity data. This 3D matrix is usually stored as slices of 2D matrices in `ExternalData` files, which, in the most commonly used ordering, contain the `DesignElementDimension` in the rows and `QuantitationTypes` in the columns. RMAGEML is capable of handling this data structure and maps it to the foreground and background intensity slots of BioConductor objects.

**USAGE**

The RMAGEML package has a manual and vignette describing its use. As an example, you can go to ArrayExpress (http://www.ebi.ac.uk/arrayexpress/) and query the database for the experiment with accession E-MEXP-5. Download the E-MEXP-5 MAGE-ML file, together with the MAGE-ML document describing the array used (accession A-MEXP-7). Place these files in a common directory `dir`, serving as the current working directory of R, with the RMAGEML package and all dependencies loaded. You can now import the E-MEXP-5 data to `limma` using the call `importMAGEML(directory=dir, package="limma")`. An experiment of 18 hybridizations and 5184 spots takes 39 s to import on a 1.9 GHz system with 256 MB RAM.

**DISCUSSION**

One of the first databases to make microarray data available in MAGE-ML format is ArrayExpress at the European Bioinformatics Institute (EBI) (Brazma et al., 2003). As MAGE-ML will become the standard format to exchange microarray data, development of tools that enable working with this format are in high demand. RMAGEML will evolve such that it can exploit the huge amount of information contained in MAGE-ML format data and make possible to store analysis results back as MAGE-ML.

**ACKNOWLEDGEMENTS**

We thank Helen Parkinson for useful discussions on MAGE-ML and Kjell Petersen for help with the Java-MAGEstk API. The research is supported by Research Council KUL: GOA Mefisto 666, GOA-AmbioMics, IDO (IOTA Oncology, Genetic networks); Flemish Government: FWO, Phd/postdoc grants, projects G.0115.01, G.0413.03, G.0388.03, G.0229.03, research communities (ICCoS, ANMMM, MLDM); IWT: PhD Grants, GBOU-SQUAD, GBOU-ANA, GBOU-McKnow, STW/W-Genprom; Belgian Federal Government: DTWC IUAP V-22; EU, FP5, CAGE, ERNSI, Marie-Curie QLRI-1999-50595; USA NIH grant 1R33 HG002708.

**REFERENCES**


