*Proceedings of the 3rd International Workshop*
*on Distributed Statistical Computing (DSC 2003)*
*March 20–22, Vienna, Austria    ISSN 1609-395X*
*Kurt Hornik, Friedrich Leisch & Achim Zeileis (eds.)*
*http://www.ci.tuwien.ac.at/Conferences/DSC-2003/*

# Database and R Interfacing for Annotated Microarray Data

Michael Mader    Werner Mewes

**Abstract**

A framework for expression data repositories (MAGE-ML compatiblity) emerged during the last years. Focus is now shifting towards the integration of expression databases into genomic and other bioinformatic database system as well as with statistical analysis platforms. We present an implementation of such a repository system at MIPS (`http://mips.gsf.de/`). An important requirement for this integration is a flexible and fast interface between the DBMS and the statistical environment. This interface must handle not only expression values but also a broad variety of annotation. Therefore, we have implemented the interface employing template programming with the Oracle Template Library (OTL).

## 1 Introduction

Expression experiments are today's most heterogeneous data sources in functional bioinformatics. This data complexity is not only due to varying data quality and size but is also an effect of multiple data formats, data origins (i.e. lab techniques and equipments), and levels of annotation. Systems for the storage, handling, and analysis/interpretation of such data are still under heavy development and standards just recently started to emerge (e.g. MAGE-ML standardization by the OMG in 2002).

The existing implementations fall into three major groups:

1. Public and dataset-centric databases like *GEO* (NCBI) and *ArrayExpress* (EBI) with rudimentary mining functions and user permission management

2. Local management systems without public access but similar functionality as (1.). Sometimes open-source (*SMD* Sherlock et al. (2001), *TM4* Saeed et al. (2003)) and with more advanced retrieval systems. Many commerical systems fall into this category.

3. Public repositories with extensive user permission management, comprehensive annotation, and mining functionality. Sometimes with database-level connections to local nodes (e.g. in high throughput labs and remote analysis centers). Examples are *iChip* (DKFZ) and the *MIPS Expression repository*.

Since the MIPS expression analysis group is focussing on the functional analysis of expression data in inter-experimental and inter-species contexts, we have set up a repository of group 3. This system – the MIPS Expression data repository (http://mips.gsf.de/proj/mouseExpress/ME.html) – consists of six major components.

1. DBMS
2. upload system (including MAGE-ML and many vendor-specific parsers)
3. retrieval module
4. annotation system and database connectivity
5. database exchange with local instances and iChip instances
6. R-based web statistics interface (RSPerl, RSPython, . . . )

The components 3 to 6 are still under development with rapidly improving functionality.

From the above description, it becomes clear that the surplus gain from repositories is mainly due to three features:

- Comprehensive Annotation
- Mining (across datasets, species, and technologies; employing the annotation)
- Integration with statistical analysis environment (no file handling issues)

We further focus on the integration of microarray repositories with statistics environments.

## 2 Requirements and problem description

The requirements for an interface between the DBMS of the repository and the stastistical environment are mainly defined by the following key restrictions:

- WWW context (i.e. response time critical)
- Large data objects (and, hence, data flow) are – if compared to business systems – large. Data dimensions of 40000*100 + $\geq$ 10% annotation are standard
- Heterogeneous annotation data

The annotation data have to be integrated from various sources including purely relational data (e.g. MIAME annotation in MIPS Expression data repository) and Entity-Attribute-Value systems (e.g. "element"/gene annotation and sample parameters in MIPS Expression data repository) as well as external links (e.g. further gene annotation in MIPS Expression data repository). Their data types range from binary and numeric to dendrogram notes (FunCat/GO) and controlled vocabulary

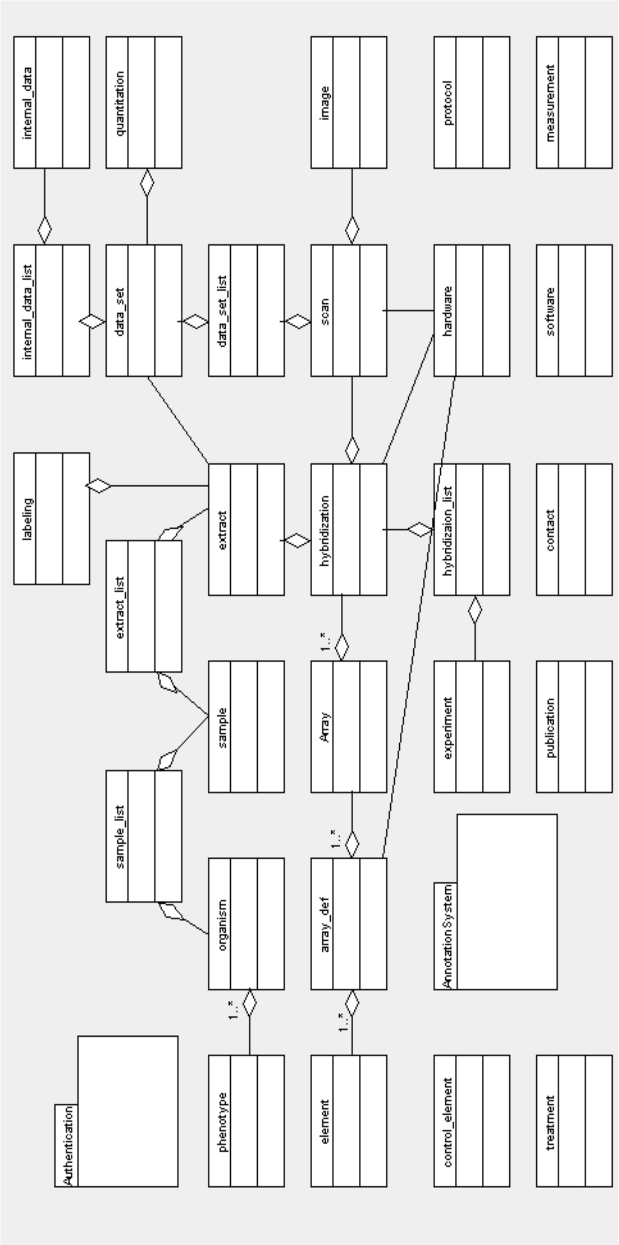
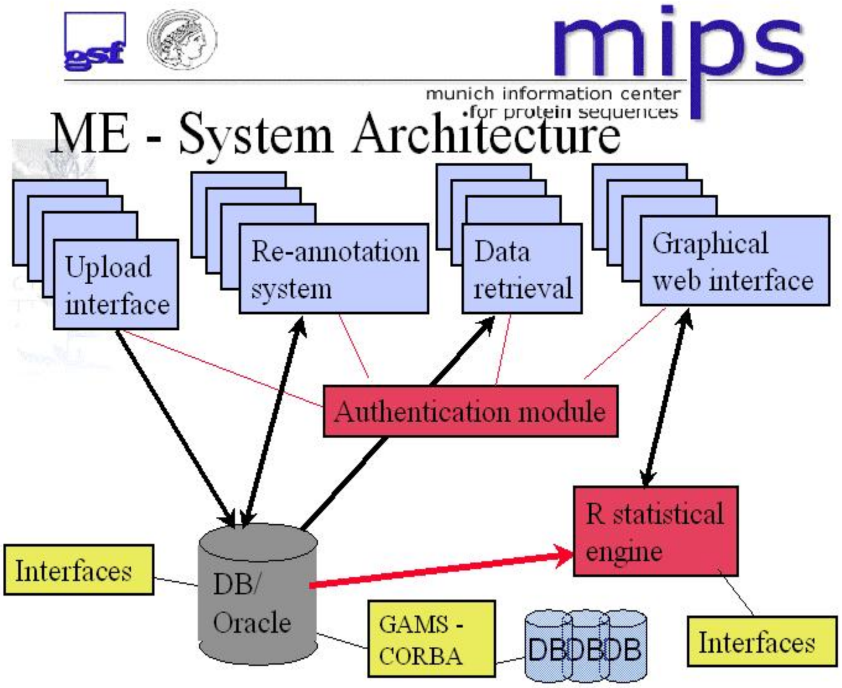


Figure 1: Simplified schema of microarray repositories (MIPS Expression data repository, total 45 tables)



mips

munich information center
for protein sequences

ME - System Architecture

Upload interface

Re-annotation system

Data retrieval

Graphical web interface

Authentication module

R statistical engine

Interfaces

DB/ Oracle

GAMS - CORBA

DB DB DB

Interfaces

Figure 2: Structure of microarray repositories (MIPS Expression data repository)

| task | level 1 | level 2 | level 3 |
|---|---|---|---|
| input (data retrieval) | $\leq 30\%$ | $\leq 10\%$ | $\leq 5\%$ |
| normalization & preprocessing | $\leq 25\%$ | $\leq 30\%$ | $\leq 10\%$ |
| (gene-wise) testing | $\leq 30\%$ | $\leq 40\%$ | $\leq 15\%$ |
| classification & clustering | | $\leq 10\%$ | $\leq 60\%$ |
| graphics & output | $\leq 15\%$ | $\leq 10\%$ | $\leq 10\%$ |

Table 1: Contributions to total computing time for different tasks and user groups, rough estimates from *ME – MIPS Expression data repository* users, `ROracle`-based retrieval

to absolute freetext. The amount of annotation available per "element"/gene ranges from essentially none to over ten twenty parameters with possibly several values.

For a detailed analysis of data transfer volume and response time issues, we shortly investigate the usage of modern microarray repositories or integrated analysis platforms. Basically we can group common microarray analysis pipelines in three groups reflecting the questions asked by the user and his/her level of statistical education:

1. Simple ones by biologists and medical staff without statistical background (level 1 users)

2. Advanced pipelines by lab researchers with statistical background (level 2 users)

3. Sophisticated queues by bioinformaticians and statisticians (level 3 users)

Pipelines from level 1 users are very frequent and normally only involve data retrieval, preprocessing and normalization, gene-wise testing on differential expression, and some simple data visualization (scatterplot matrices, MA-plots, color-coding, . . . ).

Level 2 users tend to include a variety of different algorithms for each of the basic steps and frequently use some basic unsupervised clustering technique.

Sophisticated users put much effort in planning reasonable and comprehensive pipelines either focussing on algorithm comparison, advanced clustering and classification algorithms or functional analysis.

Table 1 clearly shows that data retrieval from small local database servers can have considerable impact on total computing time. This is especially true if you consider the average usage in everydays lab routine (i.e. retrieving data, normalization, some graphical output like scatterplot matrices, gene-wise testing on differential expression). In this case data retrieval may consume up to 25% of the response time (in a time-critical environment).

## 3 Implementation and results

In the description above `R`'s standard interfaces `RDBI` and `ROracle` have been used. If you take a closer look at retrieval process, the high impact can be traced down to:

1. Long *prepare* times for repeated SQL statements;

2. Data/annotation handling in R (binds, type transitions, ...).

The *prepare* time issue partially due to database design (enforced by MAGE compatibility). However, it is also one of the main restrictions in ROracle-based data retrieval – SQL bind variables are not yet implemented.

For the data-type handling issue we do not see a straight forward way to solution within R (remember, data are large and annotation includes heterogeneous types and extends).

Therefore, we decided to implement a microarray-specific database interface based on the capabilities of the OTL (Oracle Template Library) of Kuchin (2002).

The performance increase we observe ( 50% less in response time) is mainly due to:

- OTL's binding is much faster as `ROracle` *prepare*
- STL vector reorganization is faster (and in this case less memory consumptive) as data reorganization is R

The second statement may not hold in general but is valid if reorganization involves different datatypes like with expression data annotation.

An alternative implementation using Oracle's RefCursor from within OTL is slightly more flexible. So far – due to the RefCursors – performance reaches better levels only on big dedicated database servers. On typically small local database servers (advanced workstation hardware) its performance is comparable to `ROracle` (memory is critical).

# 4 Discussion

The standaridzation of MAGE-ML by the OMG in 2002 and the MIAME defined a framework for expression data storage and analysis systems. After a short inspection of the current status and usage of these systems it becomes clear that there is a high demand for integration of storage and analysis, especially when it comes to incorporation of annotation and analyses across datasets and species borders.

Therefore, we have presented an alternative way of interface programming between Oracle (and other DBMS such as DB2) and R suitable for fast data transfer and flexible handling of additional information (annotation in our case). It is based on the OTL (Oracle Template Library) by Kuchin (2002), a high-level interface programming library based on template programming with full functionality (starting with version 4). The recent implementation is specialised for microarray and proteomics data retrieval.

A generalized version for combined retrieval of data matrices from purely relational and annotation from Entity-Attribute-Value (EAV) systems from Oracle, DB2 or via ODBC is in work and will be released as open-source.

## 4.1 Future developments

Our current work in improving this system heads in two directions:

1. Incorporation of data mining capabilities (creation of customized datasets).
2. A subsystem for bidirectional transfer of analysis results between R and the repository.

The mining capabilities will reduce data transfer volume and avoid problems with very large data objects within R. The second development – exchange of results between R and the database – will be a side-effect of the improvement of our pipeline system (*FunDaMiner*, public release planned for May 2003).

## Acknowledgements

We acknowlege comments by Jan Budczies and Sabine Tornow.

This work is supported by a grant from the BMBF ("MouseExpress", FKZ 01KW9928) to Werner Mewes.

## References

S. Kuchin. Oracle, ODBC and DB2/CLI template library, version 4.0.30. http://otl.sourceforge.net/home.htm, 2002.

A. I. Saeed, V. Sharov, J. White, J. Li, W. Liang, N. Bhagabati, J. Braisted, M. Klapa, T. Currier, M. Thiagarajan, A. Sturn, M. Snuffin, A. Rezantsev, D. Popov, A. Ryltsov, E. Kostukovich, I. Borisovsky, Z. Liu, A. Vinsavich, V. Thrush, and J. Quackenbush. TM4: A free, open-source system for microarray data management and analysis. *BioTechniques*, 34:374–378, 2003.

G. Sherlock, T. Hernandez-Boussard, A. Kasarkis, G. Binkley, J. C. Matese, S. S. Dwight, M. Kaloper, S. Weng, H. Jin, C. A. Ball, M. B. Eisen, P. T. Spellman, P. A. Brown, D. Botstein, and J. M. Cherry. The Stanford microarray database. *Nucleic Acids Research*, 29(1):152–155, 2001.

### Corresponding author

Michael Mader
GSF Research Center
Institute for Bioinformatics
Ingolstädter Landstr. 1
D-85764 Neuherberg
Germany