cluML: a markup language for clustering and cluster validity assessment of microarray data

Nadia Bolshakova and Pádraig Cunningham
Department of Computer Science, Trinity College Dublin, Ireland

Abstract: cluML, a free, open, XML-based format, is a new markup language for microarray data clustering and cluster validity assessment. This format has been designed to address some of the limitations observed in traditional formats, such as inability to store multiple clustering (including biclustering) and validation results within a dataset. The approach described performs an effective tool to support biomedical knowledge representation in gene expression data analysis. Even though cluML was developed for DNA microarray analysis applications, it may be effectively used for the representation of clustering and validation of other biomedical and physical data with no limitations.

Keywords: XML, clustering, biclustering, cluster validation, gene expression, microarray analysis

Availability: http://www.cs.tcd.ie/Nadia.Bolshakova/cluML.html

Contact: Nadia.Bolshakova@cs.tcd.ie

Introduction

The eXtensible Markup Language (XML) has become a standard for structuring documents in bioinformatics (Achard et al 2001). The Word Wide Consortium (W3C) has supervised the specifications of XML (http://www.w3.com/XML) since its inception in 1996. In biology, XML has been used for description of different types of data, e.g. microarrays (Spellman et al, 2002), biological sequences (Fenyo, 1999) and networks (Kurata et al, 2003). Systems biology (Hucka et al, 2003) and health science (Wang et al, 2003) have also benefited from their own XML formats. One of the most

important challenges in this XML explosion would be to identify novel and useful patters from large document collections. Although a number of data mining and artificial intelligence techniques have been successfully applied to different knowledge discovery domains, new solution will be required to approach this problem. Today a bioinformatics information system typically deals with large data sets reaching a total volume of over gigabytes. Gene expression technologies allow measuring the expression of thousands of genes simultaneously under multiple experimental conditions (Schena et al 1995). Microarray experimental results are affected by a multitude of variables that affect their outputs.

Three independent efforts started in 2000 to develop data storage formats for microarray data. Rosetta Biosoftware submitted Gene Expression Markup Language, GEML (http://www.rosettabio.com/tech/geml/default.htm, MAGE-ML predecessor), European Bioinformatics Institute (http://www.ebi.ac.uk) proposed MGED's Microarray Markup Language (MAML) and NetGenics (http://www.netgenics.com) made a Corba-based proposal. These three submitters decided to work together on a joint revised submitting that has become the basis for the Microarray and Gene Expression Data (MAGE, http://www.mged.org) UML model and DTD. The groups submitted XML based proposal to the Object Management Group (OMG) and the MAGE-ML was approved that year. The MIAME (Minimum Information About Microarray Experiment) standard format was developed by Microarray Gene Expression Data (MGED) Group (Brazma et al 2001) to guide the development of microarray databases and data management software. For instance, ArrayExpress is a public repository for microarray based gene expression data. It implements the MIAME annotation standard, XML based data exchange format MAGE-ML (Microarray Gene Expression Markup Language) (Brazma et al 2003).

An important step in the analysis of DNA microarray data is the detection of samples and/or genes with similar expression patterns. A number of clustering algorithms were developed and implemented for gene expression data. The prediction of the correct number of clusters in a data set is a critical problem in unsupervised classification. Various cluster validity indices have been proposed to measure the quality of clustering results (Bolshakova and Azuaje 2003a).

Research so far has not provided XML-based formats for DNA microarray clustering and cluster validity assessment. Therefore, there is the need to design such a format, which integrates clustering, and validation results for predicting the optimal number of clusters in gene expression data analyses.

The authors propose to adapt XML to documents, which contain multiple clustering (including biclustering) and validation data for interchange between databases and other sources of data.

Format Overview

cluML is an XML-based format for representing a dataset, associated clustering results and their validation. While the format was developed specifically for application within the Machaon microarray expression data clustering environment (Bolshakova and Azuaje 2003b), it is not limited to microarray-related applications. Nevertheless this paper is mostly focused on microarray data structure. In this context a dataset consists of a number of named samples/conditions (objects) represented by the values of expression of some set of genes (features). Thus, in more broad terms the format represents a single dataset consisting of named objects. Each object is represented by an element containing a number of child nodes representing named features. Each feature has a numerical value.

For example:

```
<objects type="microarray-samples">
    . . .
    <object name="sample_31">
        <feature name="U22376" value="408" />
        <feature name="X59417" value="1784" />
        . . .
    </object>
    <object name="sample_32">
        <feature name="U22376" value="1047" />
        <feature name="X59417" value="1214" />
        . . .
```

```
        </object>
        . . .
</objects>
```

The only object type specified by the format at the moment is “microarray-samples”, while other values of type attribute may be used for different applications.

The format supports representing of multiple partitionings of both objects and features, as well as multiple sets of biclusters (Cheng. and Church, 2000) attached to the dataset. Each partitioning element contains a set of corresponding (bi)clusters. And each (bi)cluster by itself contains a set of references to either objects, or features, or - in the case of a bicluster - both, for example:

```
<partitioning name="K-means clustering results" method="K-means">
        <object-clusters>
                <cluster name="0">
                        <object name="sample_1" />
                        <object name="sample_4" />
                </cluster>
                <cluster name="1">
                        . . .
                </cluster>
        </object-clusters>
. . .
```

or

```
<partitioning name="biclustering">
<biclusters>
        <cluster name="1">
                <object name="sample_31" />
                <object name="sample_33" />
                <feature name="U05259" />
```

```
            <feature name="M92287" />
        </cluster>
</biclusters>
. . .
```

At the same time, each partitioning may contain a set of parameters used by a particular clustering algorithm implementation, which has produced it. For example:

```
<cluster-parameters>
        <parameter name="Metrics" value="Euclidean" />
        <parameter name="K" value="2" />
        <parameter name="Transformation" value="No transformation" />
        <parameter name="Initialization" value="First K elements" />
</cluster-parameters>
```

The current version of the format does not specify any particular conventions for those parameters names and values.

Validation results associated with each partitioning are also reflected in the format. The results of each validation contain a method identity, a set of validation parameters (if any) and the results themselves, for example:

```
<validation method="DB">
        <validation-parameters>
            <parameter name="Metrics" value="Euclidean" />
            <parameter name="Intercluster Distance" value="Complete linkage" />
            <parameter name="Intracluster Distance" value="Complete diameter" />
            <parameter name="Transformation" value="No transformation" />
        </validation-parameters>
        <validation-results>
            <result name="Davies-Bouldin Index" value="1.4849593243324117" />
        </validation-results>
</validation>
```

Both parameters and results are just name-value pairs. For many algorithms the validation result contains a single value, but such algorithms as Silhouettes may produce multiple values. Each partitioning may have a number of results associated with it.

cluML naturally supports overlapping, as references to the same set of objects and/or features may be used in multiple clusters or biclusters. Fuzzy clusters are not supported by the present version of the software. However, the non-zero values of membership function may be attached to each reference to object and/or feature within each cluster as additional attribute. That is considered as one of possible extensions for the next version of the format.

To summarise the format contains:

- a dataset;
- multiple partitionings of its objects and/or features;
- multiple sets of biclusters;
- multiple validation results for each partitioning;
- names of clustering and validation methods used;
- parameters for each clustering and validation method used.

More formal definition of the format is provided in the form of XML Schema (http://www.cs.tcd.ie/Nadia.Bolshakova/cluML.html).

cluML is an XML-based format for representing a dataset, associated clustering results and their validation. This format has been designed to address some of the limitations observed in traditional formats, such as inability to store multiple clustering and validation results within a dataset. Even though cluML was developed for DNA microarray analysis applications, it may be effectively used for the representation of clustering and validation of other biomedical and physical data with no limitations. The approach described performs an effective tool to support biomedical knowledge representation in gene expression data analysis.

The Machaon CVE (cluster validation tool for gene expression data) (Bolshakova and Azuaje 2003b) has been updated to support this XML-based format. Machaon CVE

(Clustering and Validation Environment) system aims to partition samples or genes into groups characterised by similar expression patterns, and to evaluate the quality of the clusters obtained. The program may be downloaded from the cluML website (http://www.cs.tcd.ie/Nadia.Bolshakova/cluML.html).

Acknowledgements

This material is based upon works supported by the Science Foundation Ireland under Grant No. S.F.I.-02IN.1I111.

References

Achard F., Vaysseix G. and Barillort E. 2001. XML, bioinformatics and data integration. *Bioinformatics*, 17:115-125.

Brazma A., Hingamp P., Quackenbush J., Sherlock G., Spellman P., Stoeckert C., Aach J., Ansorge W., Ball C.A., Causton H.C., Gaasterland T., Glenisson P., Holstege F.C., Kim I.F., Markowitz V., Matese J.C., Parkinson H., Robinson A., Sarkans U., Schulze-Kremer S., Stewart J., Taylor R., Vilo J. and Vingron M. 2001. Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nature Genetic*, 29:365-71.

Brazma A., Parkinson H., Sarkans U., Shojatalab M., Vilo J., Abeygunawardena N., Holloway E., Kapushesky M., Kemmeren P., Lara G.G., Oezcimen A., Rocca-Serra P. and Sansone S.A. 2003. ArrayExpress - a public repository for microarray gene expression data at the EBI. *Nucleic Acids Research*, 1:68-71.

Bolshakova N. and Azuaje F. 2003a. Cluster validation techniques for genome expression data. *Signal Processing*, 83:825-833.

Bolshakova N. and Azuaje F. 2003b. Machaon CVE: cluster validation for gene expression data. *Bioinformatics*, 19:2494-5.

Cheng Y. and Church G. 2000. Biclustering of expression data. *Proc. International Conf. on Intelligent Systems in Molecular Biology*, 93-103.

Fenyo D. 1999. The biopolymer markup language. *Bioinformatics*, 15: 339-340.

Hucka M., Finney A., Sauro H.M., Bolouri H., Doyle J.C., Kitano H., Arkin A.P., Bornstein B.J., Bray D., Cornish-Bowden A., Cuellar A.A., Dronov S., Gilles E.D.,

Ginkel M., Gor V., Goryanin I.I., Hedley W.J., Hodgman T.C., Hofmeyr J.H., Hunter P.J., Juty N.S., Kasberger J.L., Kremling A., Kummer U., Le Novere N., Loew L.M., Lucio D., Mendes P., Minch E., Mjolsness E.D., Nakayama Y., Nelson M.R., Nielsen P.F., Sakurada T., Schaff J.C., Shapiro B.E., Shimizu T.S., Spence H.D., Stelling J., Takahashi K., Tomita M., Wagner J. and Wang J. 2003. The systems biology markup language (SBML): a medium for representation and exchange of biochemical network models. *Bioinformatics*, 19: 524-531.

Kurata H., Matoba N. and Shimizu N. 2003. CADLIIVE for constructing a large-scale biomedical network based on a simulation-detected notation and its application to yeast cell cycle. *Nucleic Acids Research*, 31:4071-4084.

Schena M., Shalon D., Davis R.W. and Brown P.O. 1995. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270:467-470.

Spellman P.T., 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 B.J., Robinson A., Bassett D., Stoeckert Jr C.J. and Brazma A. 2002. Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biology*, 3:0046.1-0046.9.

Wang H., Azuaje F., Jung B. and Black N. 2003. A markup language for electrocardiogram data acquisition and analysis (ecgML), *BMC Med Inform Decis Mak*. 3:4.