



CisML: an XML-based format for sequence motif detection software

Peter M. Haverty¹ and Zhiping Weng^{1,2,*}

¹Bioinformatics Program and ²Biomedical Engineering Department, Boston University, Boston, MA 02215, USA

Received on December 31, 2003; revised on February 18, 2004; accepted on February 20, 2004
Advance Access publication March 4, 2004

ABSTRACT

Summary: CisML is an XML-based format for sequence motif detection software. This proposed standard is applicable to many types of sequence motif detection programs. It is intended to facilitate the integration of data and the comparison of results from different software packages, and to simplify the development of downstream tools. XSL stylesheets are provided for easy generation of text, html and graphical reports from CisML-formatted data.

Availability: <http://zlab.bu.edu/CisML/>

Contact: zhiping@bu.edu

Supplementary information: Example CisML-formatted data and XSL stylesheets for report generation are available along with the sample output.

INTRODUCTION

Extensible Markup Language (XML) has all the features of HyperText Markup Language (HTML) for easy communication via the World Wide Web. In addition, XML provides the means of defining highly structured semantics so that computer programs can navigate an XML document effortlessly and extract relevant pieces of information. XML-based data descriptions are rapidly becoming standards in many scientific areas (Achard *et al.*, 2001). In biology, XML has been used for common data types such as biological sequences (Fenyó, 1999), microarrays (Spellman *et al.*, 2002) and networks (Kurata *et al.*, 2003). The fields of mathematics (<http://www.w3.org/Math>) and chemistry (Murray-Rust *et al.*, 1995) have also benefited from their own XML standards. Such standards can facilitate the exchange of scientific data by providing precise and consistent rules for their content and format. The rules eliminate the tedious and error-prone task of writing text parsing programs to reformat data for use with other software packages.

Here, we present an XML output and exchange data format for sequence motif recognition programs, called CisML. Many computational tools exist for the detection of functional sequence motifs. These range from tools for detecting

transcription factor binding sites or cis-elements (Stormo, 2000) to protein motif detection programs (Mulder and Apweiler, 2002). A common output format would simplify testing and benchmarking these tools, the creation of pipelines for analyzing their results, the exchange of data between laboratories and combining results from multiple tools to take advantage of their different features. Although CisML was originally conceived for use with tools that analyze a promoter sequence with position specific scoring matrices (PSSMs), it is general and extensible enough to work with all sequence motif detection tools.

CisML DESCRIPTION

CisML describes the detected sequence motifs by separating the results into three main components: information about the pattern that describes functional motifs, the scanned sequence and the matched motif. These correspond to three CisML elements, with *scanned-sequence* being a child element of *pattern* and *matched-element* being a child element of *scanned-sequence*. The most basic element is *matched-element*, which has attributes to describe the location of the matched motifs in a sequence and the score or *P*-value of the match. The *scanned-sequence* element has attributes to describe the identity of the sequence as well as the score or *P*-value with which the sequence as a whole matches a pattern. A single sequence may have multiple matched motifs, so a *scanned-sequence* element can contain multiple *matched-element* elements. A pattern may be used to search multiple sequences, so a *pattern* element can contain multiple *scanned-sequence* elements. The *pattern* element has attributes to describe the identity of the pattern as well as the quality of the match between the pattern and the sequence group it was used to scan. In some cases, a group of patterns is used to scan a sequence in order to detect clusters of motifs (Frith *et al.*, 2003). For these programs, a *multi-pattern-scan* element can be used to group a series of *pattern* elements. The outermost or root element, *cis-element-search*, contains a group of *pattern* or *multi-pattern-scan* elements and the *parameters* and *program-name* elements. Elements within the *parameters* element describe the parameters used with the program that

*To whom correspondence should be addressed.



Fig. 1. A graphical depiction of sequence elements generated from hypothetical CisML formatted data using a simple XSL stylesheet. The locations of pattern matches detected for two sequences (in black) are depicted by short lines (in gray). The accessions and names for four motif patterns are listed in the columns on the left and right, respectively, along with the identities of the two sequences. Application of a simple XSL stylesheet (included in Supplementary materials) renders such images as XML-based SVG images, which lend themselves directly to including clickable elements. Such clickable elements could be used to provide additional information about each pattern match, for example.

generated the CisML formatted data and the names of any input files that were used. This parameter information is intended to be sufficient for another user to reproduce exactly the results described by the CisML file.

The *parameters*, *multi-pattern-scan*, *pattern*, *scanned-sequence* and *matched-element* elements can also contain elements defined by the user under a separate namespace. This extensibility allows for the inclusion of program-specific parameters, pattern descriptions, etc. Sequence and pattern annotation data may become redundant if patterns and sequences are used multiple times in the running of a program. In the future, as the description languages of sequences, and potentially those of patterns, become more standardized, each *scanned-sequence* or *pattern* element would contain links to specific data in external files rather than repetitive annotation. Currently, further annotation can be made available through the use of the database (*db*) and/or the Life Sciences ID (*lsid*) (<http://www.i3c.org/wgr/ta/resources/lsid/docs/>) attributes of these elements.

We have described the specifications for CisML using the World Wide Web Consortium (W3C) Schema format. As support for W3C Schema is not yet universal, we also provide the specifications in the easier to understand, but less detailed, Document Type Definition (DTD) format. These specifications allow humans and computer programs to judge reliably whether or not a CisML document has been written correctly without resorting to writing complicated parsing programs. The descriptions of CisML in both W3C and DTD formats,

as well as simple explanations of their meaning, are available at our Website <http://zlab.bu.edu/CisML/>

One advantage of an XML-based format is the simplicity of generating different representations of the data with the XML Stylesheet Language (XSL). XSL stylesheets can be used with a number of open source XSL processing tools to render XML data into text, HTML, graphical or PDF reports. These tools include LibXSLT (<http://xmlsoft.org/XSLT/>), SAXON (<http://saxon.sourceforge.net>) and FOP (<http://xml.apache.org/fop>). This re-rendering of data can be done without re-running the program that generated the data and without writing complicated parsers. We provide five example CisML stylesheets to generate text or HTML reports from a pattern- or sequence-centric perspective (Supplemental data) or to generate a graphical PDF file documenting the locations of patterns in a given sequence (Fig. 1 and Supplemental data). Figure 1 was made using an XSL stylesheet that creates scalable vector graphic (SVG) images that are embedded in a PDF document using FOP. SVG images can be created to associate clickable Web links with their graphic features, leading to the possibility of generating interactive Web documents in a straightforward fashion for any CisML formatted data.

To aid in the use of CisML, we provide programs to convert MatInspector (Quandt *et al.*, 1995) and tfscan (Rice *et al.*, 2000) outputs to CisML. As such conversions cannot provide all the data for a complete CisML document and are subject to changes in the input formats, the best way to use CisML is to add functions to motif search programs to output

CisML directly. A list of programs currently producing CisML documents is available at our Website.

ACKNOWLEDGEMENTS

We thank Yutao Fu and Zhenjun Hu for thoughtful discussions. This work has been supported in part by NSF grants DBI-0078194, MRI DBI-0116574 and IGERT-9870710, and NIH grants P20GM66401 and R01HG03110.

REFERENCES

- Achard,F., Vaysseix,G. and Barillot,E. (2001) XML, bioinformatics and data integration. *Bioinformatics*, **17**, 115–125.
- Fenyo,D. (1999) The Biopolymer Markup Language. *Bioinformatics*, **15**, 339–340.
- Frith,M.C., Li,M.C. and Weng,Z. (2003) Cluster-Buster: finding dense clusters of motifs in DNA sequences. *Nucleic Acids Res.*, **31**, 3666–3668.
- Kurata,H., Matoba,N. and Shimizu,N. (2003) CADLIVE for constructing a large-scale biochemical network based on a simulation-directed notation and its application to yeast cell cycle. *Nucleic Acids Res.*, **31**, 4071–4084.
- Mulder,N.J. and Apweiler,R. (2002) Tools and resources for identifying protein families, domains and motifs. *Genome Biol.*, **3**, REVIEWS2001.
- Murray-Rust,P., Leach,C. and Rzepa,H.S. (1995) Chemical Markup Language. *Abstr. Pap. Am. Chem. Soc.*, **210**, 40-COMP Part 1.
- Quandt,K., Frech,K., Karas,H., Wingender,E. and Werner,T. (1995) MatInd and MatInspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucleic Acids Res.*, **23**, 4878–4884.
- Rice,P., Longden,I. and Bleasby,A. (2000) EMBOSS: the European Molecular Biology Open Software Suite. *Trends Genet.*, **16**, 276–277.
- Spellman,P.T., Miller,M., Stewart,J., Troup,C., Sarkans,U., Chervitz,S. Bernhart,D., Sherlock,G., Ball,C., Lepage,M. *et al.* (2002) Design and implementation of microarray gene expression markup language (MAGE-ML). *Genome Biol.*, **3**, RESEARCH0046.
- Stormo,G.D. (2000) DNA binding sites: representation and discovery. *Bioinformatics*, **16**, 16–23.