BIOINFORMATICS APPLICATIONS NOTE

Vol. 15 no. 4 1999
Pages 339–340

The Biopolymer Markup Language

D. Fenyö

Proteometrics, LLC, 38 W. 38th Street, New York, NY 10018, USA

Received on October 12, 1998; accepted on January 7, 1999

Abstract

Summary: *An XML derived from a data model designed to be a hierarchical representation of an organism has been specified and a browser to use this language has been developed.*
Availability: *The language definition is available in HTML form at http://www.proteometrics.com/BIOML/. The BioML browser is available on request from the author.*
Contact: *bioml@proteometrics.com*

The lack of a standard data communication format has limited the usefulness of biopolymer sequence databases. Each database uses its own data format to return information to the user. In most cases, the data formats are sufficiently cryptic that they hamper access to this information by potential users who are more familiar with the concepts taught in biology than they are with the requirements of relational databases.

Even for computer programmers, the structure of the data output from these databases can be difficult to parse because the formats use the structure of the database as a data model. Lines frequently begin with a descriptor tag that indicates how the data was stored in the database. However, these tags do not indicate when one piece of information ends and another begins: readers are expected to determine these limits using their best judgement, which is often a difficult task.

We have chosen to adapt the concepts of the eXtensible Markup Language (XML, Bray *et al*., 1998) as a general framework for building a logical, easy-to-parse data format for biopolymer sequence annotations. The design requirements for the new language were that it must:

1. be extensible, i.e. it should conform to the XML format;
2. be able to precisely capture molecular biology information (e.g. about a protein or a gene) and provide a true representation of the underlying biology;
3. be easily and unambiguously read by humans and machines;
4. logically connect every element in a clearly expressed statement nesting structure;
5. include data that is not ASCII as a basic data type; and
6. support the simple conversion of existing data.

The Biopolymer Markup Language (BioML) (Fenyö, 1998) is the result of this design process. BioML allows the expression of complex annotation for protein and nucleotide sequence information. BioML was designed to mimic the hierarchical structure of a living organism. XML-type languages are hierarchical by their nature and the general concepts of biological organization make a suitable template for constructing a simple, easy to comprehend XML.

The most common model for an XML is to choose its elements so it can accurately model the structure of a printed document. The now familiar Hypertext Markup Language (HTML) is very similar to an XML of this type, although HTML itself does not strictly follow XML rules. The less familiar Chemical Markup Language (CML) (Murray-Rust *et al*., 1998) and Bioinformatic Sequence Markup Language (BSML) (Spitzner, 1997) are also based on document models. These languages were all designed to layout text and images on screens or paper and are well suited to this task.

A simple example illustrating the way in which BioML can be used to mimic a biological structure is illustrated in the following highly simplified code fragment that expresses the idea of a gene:

```
<bioml>
  <organism>
    <chromosome>
      <gene>
      </gene>
    </chromosome>
  </organism>
</bioml>
```

The elements are arranged so that objects contained within the physical boundaries of a larger object are enclosed by the element tags representing that larger object. The appropriate annotation for a particular object is inserted between the start and end tags for a particular object. For example, the DNA sequence for the gene would be inserted between the <gene> and </gene> tags (verbally these tags can be read as ‘start gene’ and ‘end gene’). By a judicious choice of the set of elements used to create the grammar, most common situations requiring markup and annotation in biopolymer research can be expressed clearly in the context of the underlying physical structures that are being described. The indentations in this example and the other shown below are strictly for the purposes of clarity: BioML handles white space in the same manner as other XMLs.

Very complicated structures can be written down unambiguously using this type of scheme. For example, a ribosome could be described by the following code fragment.

```
<bioml>
  <organism>
    <species>Homo sapiens</species>
    <cell>
      <organelle type="RER">
        <particle type="ribosome">
          <protein id="1">
          </protein>
          <protein id="2">
          </protein>
          ...
          <rna id="1">
          </rna>
          <rna id="2">
          </rna>
          ...
        </particle>
      </organelle>
    </cell>
  </organism>
</bioml>
```

All of the appropriate annotation, such as peptide and RNA sequences, post-translational modifications, general notes and literature references can be inserted between the appropriate start and end tags in this structure.

BioML also contains a subset of elements that are used to express ideas useful for annotation, but which are not necessarily fundamental physical properties of the object in question. These annotation elements allow the specification of crosslinking (such as disulphide bonds), domain structures in oligonucleotides and oligopeptides, or specific residue modifications.

The language also contains a number of general purpose elements that are necessary to perform hyper-linking to other documents, such as those for describing literature references and for linking to HTML files held on any accessible server. There is also a general purpose 'note' element that can be used to insert parsed text information about the enclosing object, a 'form' element for creating custom information input forms, and a 'comment' element that can be used to describe the code itself.

BioML parts with the standard definition of an XML in that it allows the inclusion of non-text information. This information can be any binary data that can be represented by a string of bytes. A string of 256 bytes of binary data would be represented by

```
<binary format="X" length="256">...</binary>
```

where the ellipsis represents the non-character data. In a real example, the 'X' in the format attribute would be replaced by a data format that a browser could decide to represent or ignore (e.g. GIF or GZIP).

BioML also contains a simple mechanism for accepting information in other data formats. The formatted data can be wrapped in a 'data' element if it is more convenient to maintain it in that format rather than translating it into BioML. For example, three dimensional protein structure information in the standard Protein Data Bank (PDB) format would be wrapped by the following tags:

```
<data format="PDB" length="xxx">...</data>.
```

The length attribute is necessary here as well: it prevents confusing a parsing program with unexpected characters contained within the 'data' element ('xxx' represents the number of bytes between the start and end data tags).

An experimental browser for BioML documents has been written in C++ (Microsoft Visual C++, version 5) for Windows 95, 98 and NT 4.0 platforms. It uses the hierarchical organization of the language to construct an index tree that allows the user to navigate through the document quickly. The logical structure of a BioML document allows it to contain considerably more information than a typical HTML document, therefore implementing some type of indexed user interface may be a necessary feature of any BioML browser. A simple editor and code viewer are included with the browser. CGI scripts that translate SWISSPROT and PIR format data entries into BioML have been written. The data from these sources can be used to provide 'bare-bones' templates for the further annotation of a selected sequence.

Acknowledgements

The author would like to thank Brian T. Chait and Ronald C. Beavis for their suggestions regarding both BioML and this manuscript.

References

Bray,T., Paoli,J. and Sperberg-McQueen,C.M. (1998) Extensible Markup Lanuage (XML) 1.0. Available at http://www.w3.org/TR/1998/REC-xml-19980210.html.

Fenyö,D. (1998) The Biopolymer Markup Language. Available at http://www.proteometrics.com/BIOML/index.html.

Murray-Rust,P., Rzepa,H.S. and Leach,C. (1998) CML – Chemical Markup Language. Available at http://www.ch.ic.ac.uk/cml/.

Spitzner,J.H. (1997) Bioinformatic Sequence Markup Language (BSML). Available at http://visualgenomics.com/sbir/rfc.htm.