

Contents

| | |
|---|-----------|
| 1. Introduction to XML for Bioinformatics | 1 |
| 1.1 Introduction to XML | 2 |
| 1.1.1 XML Defined | 2 |
| 1.1.2 Origins of XML | 4 |
| 1.1.3 The XML Family of Specifications | 5 |
| 1.1.4 Web Services Defined | 6 |
| 1.2 Using XML for Biological Data Exchange | 7 |
| 1.2.1 Case Study: The Distributed Annotation System | 8 |
| 1.2.2 XML Formats for Bioinformatics | 11 |
| 1.3 Evaluating XML Usage in Bioinformatics | 12 |
| 1.3.1 Advantages of XML | 12 |
| 1.3.2 Disadvantages of XML | 13 |
| 1.4 Useful Resources | 14 |
| 1.4.1 Articles | 14 |
| 1.4.2 Web Site and Web Resources | 15 |
| 2. Fundamentals of XML and BSML | 17 |
| 2.1 Getting Started with BSML | 17 |
| 2.1.1 Using Genomic Workspace™ | 20 |
| 2.2 Fundamentals of XML | 22 |
| 2.2.1 Working with Elements | 22 |
| 2.2.2 Working with Attributes | 23 |
| 2.2.3 The XML Prolog | 24 |
| 2.2.4 Comments | 24 |
| 2.2.5 Processing Instructions | 24 |
| 2.2.6 Character Encoding | 25 |
| 2.2.7 CDATA Sections | 26 |
| 2.2.8 Creating Well-Formed XML Documents | 27 |
| 2.2.9 Creating Valid XML Documents | 28 |
| 2.2.10 Working with XML Parsers | 30 |
| 2.3 Fundamentals of XML Namespaces | 31 |
| 2.3.1 Why We Need XML Namespaces | 31 |
| 2.3.2 Declaring and Using XML Namespaces | 33 |
| 2.3.3 Declaring a Default Namespace | 34 |

| | | |
|-----------|---|-----------|
| 2.4 | Fundamentals of BSML | 35 |
| 2.4.1 | BSML File Formats | 36 |
| 2.4.2 | BSML Document Structure | 36 |
| 2.4.3 | Representing Sequences | 38 |
| 2.4.4 | Representing Sequence Features | 39 |
| 2.4.5 | Retrieving Live BSML Data via XEMBL | 45 |
| 2.5 | Useful Resources | 47 |
| 3. | DTDs for Bioinformatics | 49 |
| 3.1 | Introduction to DTDs | 49 |
| 3.1.1 | A Bird's-Eye View: Protein DTD | 50 |
| 3.1.2 | Validating XML Documents | 52 |
| 3.2 | Document Type Declarations | 55 |
| 3.3 | Declaring Elements | 57 |
| 3.3.1 | EMPTY | 57 |
| 3.3.2 | ANY | 58 |
| 3.3.3 | #PCDATA | 58 |
| 3.3.4 | Child Elements | 59 |
| 3.3.5 | Mixed Content | 60 |
| 3.4 | Declaring Attributes | 61 |
| 3.4.1 | Attribute Types | 62 |
| 3.4.2 | Attribute Behaviors | 65 |
| 3.5 | Working with Entities | 66 |
| 3.5.1 | General Entities | 66 |
| 3.5.2 | Parameter Entities | 69 |
| 3.5.3 | Entity Summary | 70 |
| 3.5.4 | Conditional DTD Sections | 70 |
| 3.6 | Case Study: NCBI TinySeq | 72 |
| 3.6.1 | NCBI and XML | 72 |
| 3.6.2 | The TinySeq DTD | 73 |
| 4. | XML Schemas for Bioinformatics | 81 |
| 4.1 | Introduction to XML Schemas | 81 |
| 4.1.1 | XML Schemas for Bioinformatics | 82 |
| 4.2 | Essential Concepts: Representing Protein Data | 82 |
| 4.2.1 | The <schema> element | 84 |
| 4.2.2 | Schema Documentation | 86 |
| 4.2.3 | Simple Types vs. Complex Types | 86 |
| 4.2.4 | Global Elements vs. Local Elements | 86 |
| 4.2.5 | Creating Instance Documents | 87 |
| 4.2.6 | Validating Instance Documents | 88 |
| 4.3 | Working with Simple Types | 89 |
| 4.3.1 | Built-in Schema Types | 89 |
| 4.3.2 | Working with Facets | 91 |
| 4.4 | Working with Complex Types | 94 |
| 4.4.1 | Introduction to Complex Types | 94 |
| 4.4.2 | Declaring Empty Element Types | 96 |

| | |
|--|------------|
| 7. Parsing DAS Data with SAX | 175 |
| 7.1 Introduction to SAX | 175 |
| 7.1.1 A First Example | 175 |
| 7.1.2 The XMLReader Interface | 179 |
| 7.1.3 The ContentHandler Interface | 182 |
| 7.1.4 Extending the DefaultHandler | 184 |
| 7.1.5 Using InputSource Objects | 186 |
| 7.2 Validating XML Documents | 188 |
| 7.2.1 Checking for Well-Formedness | 188 |
| 7.2.2 Validating XML Documents: Overview | 190 |
| 7.2.3 Activating the SAX Validation Feature | 191 |
| 7.2.4 The ErrorHandler Interface | 191 |
| 7.2.5 Validating against XML Schemas | 196 |
| 7.3 Elements, Attributes, and Namespaces | 197 |
| 7.3.1 Working with Elements and Namespaces | 197 |
| 7.3.2 Working with Attributes | 202 |
| 7.4 Building Custom Data Structures with SAX | 204 |
| 7.4.1 Parsing DAS Feature Data | 204 |
| 7.4.2 Integrating with BioJava | 208 |
| 8. Parsing DAS Data with JDOM | 215 |
| 8.1 JDOM Basics | 215 |
| 8.1.1 JDOM Package Overview | 215 |
| 8.1.2 Parsing XML Documents with JDOM | 216 |
| 8.2 Parsing DAS Documents with JDOM | 221 |
| 8.2.1 Introduction to the JDOM Element API | 221 |
| 8.2.2 Traversing DAS Documents | 224 |
| 8.2.3 Parsing DAS <i>dsm</i> Documents | 229 |
| 8.3 Creating DAS Documents with JDOM | 233 |
| 8.3.1 Creating New Documents | 233 |
| 8.3.2 Creating New Elements | 234 |
| 8.3.3 A Complete Example | 235 |
| 8.4 Building the JDAS Library | 238 |
| 8.4.1 Using JDAS | 238 |
| 8.4.2 The JDAS Source Code | 243 |
| 9. Web Services for Bioinformatics | 247 |
| 9.1 Introduction to Web Services | 247 |
| 9.1.1 Web Services Defined | 247 |
| 9.1.2 Architectural Options | 250 |
| 9.2 Case Study: Introduction to the NCI caBIO Project | 251 |
| 9.2.1 Background: Connecting to caBIO via the Java RMI Interface | 253 |
| 9.3 Introduction to REST-Based Web Services | 257 |
| 9.3.1 Introduction to REST | 257 |
| 9.3.2 Connecting to the caBIO REST Interface | 258 |
| 9.3.3 Example Application: Command Line caBIO Browser | 262 |

| | |
|--|------------|
| 9.4 Introduction to SOAP | 267 |
| 9.4.1 SOAP Overview | 268 |
| 9.4.2 Constructing SOAP Messages | 270 |
| 9.4.3 Transporting SOAP via HTTP | 273 |
| 9.5 Introduction to Apache Axis | 275 |
| 9.5.1 Building a Web Service with Axis | 276 |
| 9.5.2 Connecting to caBIO with Axis | 281 |
| Appendix | 283 |
| 1 Nucleotide Base Codes | 283 |
| 2 Amino Acid Codes | 283 |
| Bibliography | 285 |
| Index | 291 |

XML represents a new field of scientific inquiry, devoted to answering questions about computational resources to answer those questions. A key goal of bioinformatics is to build database systems and software platforms capable of storing and analyzing large volumes of biological data. To that end, hundreds of biological databases are now available and provide a wide variety of biological data.

Given this diverse set of biological data, the exponential growth of biological data sets, and the desire to share data for open scientific exchange, the bioinformatics community is continually exploring new options for data representation, storage, and exchange. In the past few years, many in the bioinformatics community have turned to XML to address the pressing needs associated with biological data. XML, or Extensible Markup Language, is a technical specification originally created for data representation and exchange over the Internet. XML is an open standard, officially specified by the World Wide Web Consortium (W3C), and deliberately designed to be operating system and programming language independent.

XML is extensible to many application domains and has been successfully used to represent multiple types of data, including e-commerce transactions, search engine results, scalable vector graphics, and even voice recognition and voice synthesis. Since its introduction, XML has also been successfully used to represent a growing set of biological data, including nucleotide sequences, genomic annotations, protein-protein interactions, and signal transduction pathways. XML also forms the backbone of biological data exchange, enabling researchers to aggregate data from multiple heterogeneous data sources.

The goal of this book is to present the fundamentals of XML, and to demonstrate the ways in which XML is being usefully applied in the field of bioinformatics. This chapter presents the first step in this goal, and therefore focuses on three main questions:

- What exactly is XML?
- How is XML currently being used in bioinformatics?
- What are the pros and cons of using XML in bioinformatics?

To explore these topics, we examine the origins of XML, compare XML with HTML, and provide a snapshot of the XML family of specifications. We also take a bird's-eye view of our first chapter in bioinformatics and explore the Distributed Annotation System (DAS).