Book Review

XML for Bioinformatics
Ethan Cerami
Berlin: Springer-Verlag;
ISBN: 0-387-23028-9; 304pp.; 2005; £42.50

Not until I received this book to review, did I realize that I had another book on my shelf, ‘Web Services Essentials’, by the same author. This was a good omen. The Essentials book was where I first cut my teeth on notions such as XML-RPC, SOAP, WSDLs, etc. His new book, ‘XML for Bioinformatics’, does a fine job of extending his Essentials book to a specific scientific domain—bioinformatics. In the Preface of his new book, he says that his goal is to ‘introduce XML to a bioinformatics audience’. I would make the argument that the converse is equally true (bioinformatics to an XML audience), which is particularly appealing to those of us who are professional software developers, the ‘informatics’ people, and want to forge collaborations with ‘bio’ researchers.

The book’s chapters cover the basics, which are listed here in abbreviated forms: (1) Introduction, (2) Fundamentals—XML and BSML (Bioinformatics Sequence Markup Language), (3) DTDs, (4) XML Schemas, (5) Perl, (6) DAS (Distributed Annotation System), (7) SAX, (8) JDOM, (9) Web Services. I was very pleased to see an accompanying website for the book, xmlbio.org, which any self-respecting book that involves large examples of code (and XML text) should have. Unfortunately, some of the information and URL references mentioned in the book are no longer valid, e.g. the free Genomic Workspace™ software tool, which is prevalent in Chapter 2, for viewing BSML documents seems to be no longer free. But this is where having the companion website will be valuable—to post errata and provide updated links. Better yet might be to have an accompanying wiki so that lots of people could contribute ideas, but I digress.

The book is very Java-centric and, in fact, provides several Java code listings. Being Java-centric is no big surprise since much, OK, most of the XML and Web Services world uses Java. He does, however, dedicate one chapter to a non-Java approach to XML for bioinformatics, namely, using the Perl scripting language and the BioPerl project. I realize I may be a part of the minority, but my lab prefers using the Python scripting language and therefore was a bit disappointed that there was no mention of Python-based tools for XML and bioinformatics.

I thought it especially useful to have an entire chapter dedicated to the DAS, an important project and one that lends itself nicely to promoting XML. In addition to this large project, the book also mentions the caBIO project, from the National Cancer Institute, in the final chapter on Web Services. I was surprised to see no mention, however, of the large XML-based project from the prominent Protein Data Bank. Alas, it must be challenging to write a book on this subject, given the shifting sands of standards, projects and tools for XML for bioinformatics. I heartily commend Cerami for doing so and will keep this book handy for reference and for handing to students.

Randy Heiland
Associate Director, Scientific Data Analysis Lab Pervasive Technology Labs at Indiana University
Submitted: 18th October 2005;
Received (in revised form): 19th October 2005