Book Review


Alexis B. Carter

Departments of Pathology and Laboratory Medicine, Emory University, 1364 Clifton Road NE, Room F149A, Atlanta, GA 30322

E-mail: *Alexis B. Carter- abcart2@emory.edu
*Corresponding author

Received: 06 October 11 Accepted: 07 October 11 Published: 29 October 11

This article may be cited as:
Available FREE in open access from: http://www.jpathinformatics.org/text.asp?2011/2/1/49/86831

Copyright: © 2011  Carter AB.  This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

As subspecialty board certification in clinical informatics has finally become a reality, Jules Berman’s book Methods in Medical Informatics could not be more timely. This well-written and informative text combines Dr. Berman’s expertise in programming with his vast knowledge of publicly available data sets and everyday healthcare programming needs to result in a book which should, in the opinion of this reviewer, become a staple in health informatics education programs as well as a standard addition to the personal libraries of informaticists.

The book’s title does not do justice to the wealth of information contained therein. While Python, Perl, and Ruby are certainly important components of the text as described below, the contents also house a huge amount of valuable information on publicly available data sets and how they can be accessed and used for medical discovery. Through parallel examples in Python, Perl, and Ruby, the reader is taken through sets of structured exercises, each of which includes a description of the problem or task, a human readable explanation of the script algorithm, script examples and analysis of the expected results. While this book was not intended for the novice programmer, the organization and structure of its content and explanation of the process behind the code easily facilitate a reader’s ability to use the examples on his or her own computer with a minimum of background in each of the languages provided.

During the process of reviewing this book, the reviewer (who had never previously used any of these programming languages and who is a relative novice to code-writing) used the author’s instructions to install the Python compiler. After some additional background reading in Python and its code construct from another source, several of the exercises in the book were tested using Python as the language of choice. As expected in any book containing programming code, versions of the programming languages used when writing a book are sometimes not the most recent stable version available for download by the reader at a later date. This appears to be more of an issue with open source programming...
languages because they are updated more frequently. The stable versions of Python available for download at the time of review were 2.7.2 and 3.2.2 (Perl is on version 5.14.2; Ruby is on version 1.9.2). The version of Python used in the book is 2.5 (Perl 5.8; Ruby 1.8). An example of a difference between these two versions of Python is a slightly change to the syntax for print statements, which led to some initial stalls in getting the examples to run correctly. However, this was a minor setback which was easily overcome and did not detract from rest of the book.

The book is split into four major parts and is supplemented by an epilogue and appendices. Part I on fundamental methods and algorithms covers basic computing functions such as string and image manipulation, including hash creation and text indexing. Part II on medical data resources walks the reader through a vast array of useful and publicly available data resources for research and discovery. These include but are not limited to the National Library of Medicine’s Medical Subject Headings (MeSH), the Cancer Surveillance Epidemiology and End Results (SEER) Program, Online Mendelian Inheritance in Man (OMIM), PubMed, United States census files, Centers for Disease Control and Prevention (CDC) data sets and others. The reader is also introduced into an author-developed taxonomy of neoplasms. In Part III on primary tasks (i.e. fundamental scripts) for medical informatics, basic concept-indexing, scrubbing of patient identifiers from text reports, web page construction and common gateway interfaces are demonstrated as well as image annotation and use of extensible markup language (XML) and resource description framework (RDF) files. Part IV on medical discovery uses case studies to illustrate how these powerful programming techniques can enable researchers to discover information from publicly available data sets. Examples include extracting emphysema rates from CDC data, cancer epidemiological data from the SEER database and others.

The epilogue offers sage advice on how to successfully get involved in programming and informatics as a career, and the appendices have complete instructions on how to acquire all of the programming applications used in the book as well as publicly available data sets. Additional information on other publicly available files, data sets and utilities that were not covered in the examples are also included.

In conclusion, this book is for anyone who wants to learn more about medical informatics. The book contains beautifully simple examples of how to use publicly available programming languages to get a job done, and in addition opens the door to a host of incredibly useful data sets to many who may not have been aware of their existence. Both medical professionals with only peripheral knowledge of programming and nonmedical information technology professionals will find this book useful, and its structure is ideal for biomedical informatics classrooms and clinical informatics fellowships alike.