The Biophysicist

Computational Cell Physiology: With Examples in Python by Stephen M. Baylor

Jungmin Han

Computational Medicine Section, Laboratory of Biological Modeling, National Institute of Diabetes and Digestive and Kidney Diseases, National Institutes of Health, Bethesda, MD 20892, USA

Computational Cell Physiology: With Examples in Python by Stephen M. Baylor. 2020. Independently published. 498 pp. ISBN 978-1661705145

Computational Cell Physiology: With Examples in Python introduces key cellular mechanisms that cover a wide range of cell physiology topics, particularly associated with excitable cells such as nerve, muscle, and endocrine cells. The book is written by Dr. Stephen M. Baylor, Professor Emeritus of Physiology at the University of Pennsylvania, whose expertise lies in excitation-contraction coupling in vertebrate skeletal muscle and the role of the calcium ion as an intracellular chemical messenger. What makes this book stand out from the rest is that it uses a relatively easy-to-learn programming language, Python, to describe and computationally simulate the mechanisms in question from a quantitative perspective. By doing so, the author exemplifies how a quantitative approach can be of service to studying the underlying cellular mechanisms. Prospective readers, especially those with some programming skills who want to pursue research in cell biology, will find this book extremely valuable. More importantly, as the author points out, the book is a concrete resource for nurturing one's quantitative thinking habits.

The book explains the principles of diffusion, electrophysiology, and mass action and how these components come into play in describing biophysical mechanisms underlying various cellular processes, such as the generation of action potentials and muscle contraction. Chapters 1 and 2 cover the general background of cell physiology. Needless to say, it is not feasible to include all the aspects of the field in just 2 chapters. However, the chapters discuss topics that are pertinent to the contents in later chapters. More background materials are elaborated at the beginnings of later chapters, which helps the reader grasp each chapter's main subject. In some chapters, the author describes experimental techniques used to study the electrical properties of cells, such as the patch-clamp and voltage-clamp techniques.

Many readers, especially those with limited exposure to experimental settings, will appreciate these descriptions. The cellular mechanisms elaborated in the later chapters are well supported by figures that show experimental recordings and schematics of the relevant cellular units.

© 2021 Biophysical Society.

http://prime-pdf-watermark.prime-prod.pubfactory.com/ | 2025-01-15

Chapters 3 and 4 are entirely dedicated to introducing Python to the readers. Chapter 3 provides simple Python scripts that generate graphs of 1-dimensional and 2-dimensional diffusion profiles. The author clarifies that the purpose of theses scripts was not to intimidate the readers but to illustrate the elegance of Python scripts in executing calculations and generating graphs. Each script is well commented and followed by extensive line-by-line explanations, *Program Notes*, that inform the readers about the role and components of each line. Likewise, the Python scripts presented in the succeeding chapters have detailed, line-by-line descriptions to facilitate understanding the code and quantitative aspects of the cellular mechanisms. Such documentation is tremendously useful for the reader to explore the code setup and simulate cellular processes in different parameter regimes. Chapter 4 includes a summary of the essential elements of Python, including the types of variables, statements, and syntaxes, along with some shortcuts.

Although the chapter provides relevant information about the programming language, to follow the examples and exercises discussed in the later chapters, it may be useful to read a few introductory books on Python to gain some background knowledge.

Having a GitHub repository with Binder or Google Colaboratory with all the examples and exercises in the book as Jupyter notebooks would be great supplementary material. To follow the examples, the readers need to install Python and necessary libraries on their working machine, which may not be allowed if they do not have administrator privileges. One of the advantages of having a GitHub repository or Google Colaboratory is that the readers can access and execute the example notebooks interactively without having to install the required software and Python libraries on their local machines. It also provides a solution to dealing with different versions of Python libraries, because the notebooks can be saved with the user-defined working environment and its dependencies. Moreover, the readers can build a user community in which they can share their programming experiences and help each other better understand the content of the book. Putting all the materials together and setting up a web page could be an exciting summer project for an undergraduate student.

Computational Cell Physiology: With Examples in Python has a balanced combination of content from biology, cell physiology, mathematics, and programming that synergistically support each other. One of the strengths of this book is that it has substantial documentation on the Python script examples, which provides all the necessary programming information to understand the code, while also explaining how the cellular mechanism in question is incorporated. A suitable audience for this book would be undergraduate and graduate students studying cell physiology who want to develop their programming and quantitative skills.

http://prime-pdf-watermark.prime-prod.pubfactory.com/ | 2025-01-15