PROTEIN COMPUTER SIMULATION
蛋白质计算机模拟

周麟祥 帅建伟 编著

厦门大学出版社
XIAMEN UNIVERSITY PRESS

PROTEIN COMPUTER SIMULATION
蛋白质计算机模拟

周麟祥 帅建伟 编著

厦门大学出版社
XIAMEN UNIVERSITY PRESS

图书在版编目(CIP)数据

蛋白质计算机模拟＝Protein Computer Simulation/周麟祥，帅建伟编著．—厦门：厦门大学出版社，2008.2
ISBN 978-7-5615-2967-6

Ⅰ.蛋… Ⅱ.①周…②帅… Ⅲ.蛋白质-计算机模拟-英文 Ⅳ.Q51-39

中国版本图书馆 CIP 数据核字(2008)第 019155 号

厦门大学出版社出版发行
(地址：厦门大学 邮编：361005)
http://www.xmupress.com
xmup @ public.xm.fj.cn
厦门昕嘉莹印刷有限公司印刷
(地址：厦门市前埔东路 555 号 邮编：361009)
2008 年 2 月第 1 版 2008 年 2 月第 1 次印刷
开本：787×960 1/16 印张：21.25 插页：2
字数：370 千字
定价：45.00 元(附光盘壹张)
如有印装质量问题请与承印厂调换

Abstract

This textbook covers a course of computer simulation of proteins for science major students. It consists of two knowledge blocks: Molecular Dynamics (MD) and Full Electron Structure Calculation of Proteins. We shall also discuss two main subjects: Knock-off Proteins and Protein-Ligand Interactions. However we shall not discuss the folding dynamics of proteins. These are discussed in " Cluster - Linux - Parallel Calculation System".

The Full Electron Structure Calculation of Proteins is one of the new features of this textbook.

We assume the readers have the basic knowledge of computer languages, proteins and physics. Still, information on Unix, TCL and Phthon are included briefly in this textbook as a refreshing exercise.

The keystone of this textbook is to elaborate the computer simulation of proteins. Therefore, merely reading this textbook is not enough. The students must practice all the problem examples on computers while studying. Otherwise, the students will not understand the topics discussed in this textbook.

PREFACE

21st century will be the century of life science. Life material and life phenomena will be the prominent research objectives for scientists. Biology will not be only a science based on observations and experiments, but also a science based on Quantum Theory. As Walter Gilbert said: "The new paradigm, now emerging, is that all the genes will be known (in the sense of being resident in databases available electronically), and that the starting point of a biological investigation will be theoretical. An individual scientist will begin with a theoretical conjecture, only then turn to experiments to follow up or test that hypothesis."

After completing the Human Genome Project, the main research trend now turns to proteins. In studying proteins, besides using the traditional experimental tools and theoretical methods, large scale computer simulations have indispensably become the third approach.

Computer simulation allows researchers the ability to observe the processes of protein changes in as short as 1 fs and structural changes versus temperature or versus time. Also, new protein structures can be easily obtained by using the Knock-Out Method. Additionally, directions for further biological experiments can be readily identified, saving significant amount of time, especially for difficult experiments. We can say that computer simulation has become an important tool in the study of the functioning of proteins and their structures. It is also an excellent supplementary tool for biological experiments.

Scientific researches have already touched the realm of highly sophisticated systems. There have been different physics at different times. Nano-material and proteins are the two main research directions for physicists in the present time. But, in order to study these two materials numerically, we have to attain the level of 100 ~ 1000 Tflops (10^{12}/s) computer simulation speed. For example, for nano-material:

- To simulate two nano-particles (diameter 5 nm or less), consisting about 1000 atoms, 50 Tflops is needed.
- To simulate the interaction between two nano-particle groups, 250 Tflops is needed.

- To simulate a complete nano-structure, 1000 Tflops is needed.
- For protein macro-molecule:
- To simulate 1 nanosecond, 50 Tflops is needed.
- To simulate the action of enzyme, 250 Tflops is needed.
- To simulate the fold of protein, 1000 Tflops is needed.

To reach such computer simulation speed level, we typically use two kinds of high performance computers: supercomputers and cluster computers. Supercomputers are too expensive. Now most researchers use cluster computers. This textbook will use the cluster system to perform simulations as well:

Cluster + Linux + Parallel Calculation

A protein can complete its fold in micro-seconds (10^{-6} second). But to simulate one nano-second of protein activities, current computers may take up to one whole day. For this reason, we may need to use up to 10000 computers to complete just a folding simulation. To search a medicine that cures the variola illness, we may potentially have to test 35 million medicine molecules. This means we shall need to use about 2 million computers running together to form a virtual supercomputer capable of 1100 Tflops. This is not very practical. For such a scale of simulation, as the next best choice, researchers are trying to use Grid Computing techniques. It is basically distributed computing, using internet to connect many computers into a huge "virtual supercomputer".

In this textbook, we shall only discuss two basic questions of protein simulation: Molecular Dynamics (MD) of Protein and Full Electronic Structure (FES) of Protein. We shall not discuss the basic knowledge of protein and Linux. Students who already have the knowledge can directly skip to Chapter 6 for study. Also, we shall not discuss the research topics of protein either. We believe students will be able to pick up their research topics of protein simulation after they have studied the two simulations in this textbook.

The goal of this textbook is to show students how to run simulation programs in detail without introducing the corresponding theories. So, we shall only simply list some necessary knowledge in this textbook. Student can refer to other related books in this field.

For molecular dynamics of proteins, we shall refer to NAMD program as an example (http://www.ks.uiuc.edu/Research/namd/).

For full electronic structure of protein, we shall use the ODA

program as example (Ladik, J. et at., Chem. Phys., 108:203(1986) and Jiang, Y. Ye, Y.J., and Chen, R. S., Biophys. Chem., 59:95 -105(1996)).

This textbook is a course for science major students who want to study computer protein simulation. Its manuscripts have been lectured for six years to graduate students in the Department of Physics, Fudan University.

The authors are very grateful to the Physics Department, Surface Physics Laboratory of Fudan University; the Physics Department of Xiamen University; and the Xiamen University Press for their kind support in publishing this book. Enthusiastic guidance and encouragement by Professors Xun Wang, Ling Ye, Ruibao Tao, and Xinyi Zhang at Fudan University as well as Professor Qian-er Zhang at Xiamen University are sincerely appreciated.

We profoundly thank the members of Biology Computer Lab in Xiamen University for many help and works in the publishing process.

Finally, the authors would like to point out that the work of biological computer simulation was suggested and supported by Professor Ling Ye early in 2000.

Linxiang Zhou at Fudan University campus
Jianwei Shuai at Xiamen University campus

CONTENTS

PART I PRIOR KNOWLEDGE

To study protein computer simulation, the most basic knowledge is **Unix** operating system. First of all, you need to study Unix, otherwise, you would be not able to do any computer simulation. And then, you need to study **TCL** and **Python** language. The molecular dynamics program NAMD, VMD and NWChem program need these two languages as their input file.

In this textbook we suppose that you have the basic knowledge of computer language C/C++/Java, especially, you clearly know the five elements as a computer language, so we only **list** TCL and Python's main knowledge elements, not give out their detail contents.

For protein and physics knowledge, we also **list** their basic contents only.

Therefore, if you know all knowledge of Part I, you can directly read Part II.

Unix Operating System

To carry out the computer calculation, one must first select an operating system as a platform, which is a piece of software. Unix is a common operating system, which is adopt by all the large scale computer calculation and simulation. Unix is like a "tool kit". It contains many small programs and tools that can be used together to solve complex problems.

Unix possesses several unique features, which are different from other operating systems, namely it may have multi-user, multi-tasking, and is very stable. A Unix operating system is very rarely crashed. Especially, Unix has Shell Script file, which is a strong tool for writing programs. We call it as second development of program. Also, Unix has X Windows providing a graphical interface.

If you run Internet web from PC machine instead of X-window in workstation, you should use **NetTerm** program, **Putty** program, **SSH** program, **Xwin**32 program, **Xserver** program or Windows/start/run. And if you have trouble to install Linux system, you can download the virtualization software called **VMware** workstation (www.vmware.com) to install Linux on Windows.

Chapter 1 EDITING AND MANAGING FILE

1.1 LOGIN AND LOGOUT

1.1.1 Login and Logout

(1) Tool: **NetTerm** or **Xwin32.**

(2) Login: use the telnet program to type the command:% **telnet IP** and answer: login and password or type the command ssh as % ssh IP. The ssh is a new excellent command for Linux.

(3) Logout: **logout** for telnet program (for other programs, often use: quit, exit and bye).

Please note that (1) *We use C-shell only. Its prompt is %.* (2) *There are some Berkley commands called the remote operations, but they are only used on the Unix web.*

Ex1: `% rlogin IP` (remote login)
Ex2: `% rcp-r mylinux: /home/aaa /home/myfile` (remote copy)
Ex3: `% rsh` (remote shell command)
Ex4: `% rsh name_remote_machine unix_command`
Ex5: `% rsh abc ls /home/aaa`

1.1.2 File and Command

1. Command format

In general, a complex Unix command consists of three parts:

% command -option filename or path

Ex: `% ls -l aaa`

Where aaa is a file and % is prompt in C shell. We call "ls", "-l" and "aaa" as **the parameters of the command line.**