

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

江苏工业学院图书馆
藏书章



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

作者简介

周麟祥

物理学教授。1939年12月生，原籍浙江嘉兴。1960年毕业于厦门大学物理系。

1993—1997年任美国林肯大学客座教授。曾在厦门大学任教。目前在复旦大学物理系主要从事蛋白质分子动力学和全电子结构的计算机模拟研究工作。

帅建伟

物理学教授。1968年6月生，原籍四川。1995年毕业于厦门大学物理系。1996—2007年在香港、日本、德国、美国任职。2007年回国前，在美国加州大学Irvine分校工作。现在厦门大学物理系任闽江学者特聘教授，从事生物系统计算机模拟的研究工作。

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 al. , 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

PREFACE	1
PART I PRIOR KNOWLEDGE	1
Chapter 1 EDITING AND MANAGING FILE	2
1.1 LOGIN AND LOGOUT	2
1.1.1 Login and Logout	2
1.1.2 File and Command	2
1.1.3 Set Up	3
1.2 EDITOR vi	4
1.2.1 Login and Logout	4
1.2.2 Two Statuses	4
1.2.3 Command	4
1.3 MANAGING FILE	5
1.3.1 Login/Logout	5
1.3.2 Create/Delete	5
1.3.3 List/Read	5
1.3.4 Editing File	6
1.3.5 Grep/Find	7
1.3.6 Check	8
1.3.7 Encode	9
1.3.8 Crypt	9
1.3.9 Tar	9
1.3.10 Compress	10
1.3.11 Symbols	10
1.3.12 Echo	11
1.3.13 Awk '{...}' (Aho, Weinberger and Kernighan)	11
1.3.14 Print	12
1.3.15 Calculator bc	12
1.3.16 Format Converting	12
Chapter 2 ENVIRONMENT	14
2.1 KERNEL AND SHELL	14

2. 1. 1	Three Layers: User-shell-kernel	14
2. 1. 2	Four Kinds of Shells	14
2. 1. 3	Shell Tree and Top-level Directories	14
2. 2	FILE PERMISSION	15
2. 2. 1	Permission	15
2. 2. 2	Change Mode	16
2. 2. 3	Default Permission in the . cshrc File	17
2. 3	CUSTOMIZING ENVIRONMENT	17
2. 3. 1	User's Environment Files	17
2. 3. 2	How to Edit Environment Files	17
2. 3. 3	Set Up in . cshrc File	18
2. 4	INTERNET IN UNIX SYSTEM	20
2. 4. 1	Internet Program in Unix	20
2. 4. 2	Email	21
2. 4. 3	FTP	22
2. 4. 4	Telnet	23
2. 5	MANAGING SYSTEM	24
2. 6	CUSTOMIZING NET SERVER	26
2. 6. 1	Three Commands	26
2. 6. 2	Seven Files	27
Chapter 3 SHELL SCRIPT		29
3. 1	WHAT IS SCRIPT	29
3. 2	SHELL SCRIPT LANGUAGE	30
3. 2. 1	Shell Variable	30
3. 2. 2	Operator	32
3. 2. 3	Control	33
3. 2. 4	Function	35
3. 2. 5	I/O	35
3. 2. 6	Access to Database or Another Shell Script	40
Chapter 4 COMPILING, DEBUG AND RUNNING		43
4. 1	COMPILING A SIMPLE SOURCE CODE	43
4. 1. 1	Compiling Process	43
4. 1. 2	Compiling Command and Its Options	43
4. 2	MAKEFILE AND COMMAND:MAKE	44
4. 2. 1	The Static Library and Dynamic Library	44

4.2.2	How to Create Static Library	44
4.2.3	Makefile	45
4.2.4	Command “make”	49
4.3	DEBUG	50
4.3.1	Debug Program	50
4.3.2	Method	50
4.4	RUNNING	52
4.5	PROCESS OF RUNNING	57
Chapter 5 TOOL COMMAND LANGUAGE		59
5.1	VARIABLE	60
5.1.1	Definition of Variable	60
5.1.2	Expression Using Keyword “EXPR” and “INCR”	60
5.1.3	List	61
5.1.4	Array	62
5.1.5	String	62
5.2	OPERATOR	63
5.3	CONTROL	64
5.3.1	If	64
5.3.2	Switch	65
5.3.3	For	65
5.3.4	While	65
5.3.5	Foreach	65
5.4	PROCEDURES	66
5.5	I/O	67
5.6	EVAL AND CATCH (omit)	68
Chapter 6 PYTHON SCRIPT LANGUAGE		69
6.1	VARIABLE	69
6.1.1	Numeric Variable	69
6.1.2	String	70
6.1.3	List	71
6.1.4	Class	72
6.2	OPERATOR	77
6.3	CONTROL	77
6.3.1	While	78
6.3.2	For	78

6.3.3	If	79
6.3.4	the range () Function	80
6.4	FUNCTION	81
6.5	I/O	81
6.5.1	Standard I/O	81
6.5.2	File I/O	83
6.6	EXCEPTION	83

Chapter 7 AMINO ACID AND PROTEIN 85

7.1	AMINO ACID	85
7.2	PROTEIN	89
7.2.1	Polypeptide Backbone	89
7.2.2	Peptide Plan	90
7.2.3	The Structure of Protein *	90
7.2.4	Protein Folding and Ligand-Protein Interaction	91

Chapter 8 PHYSICAL BASIC KNOWLEDGE 93

8.1	GENERAL PHYSICS	93
8.2	PHYSICS STEP	93
8.2.1	Degree of Freedom	94
8.2.2	Parameters of H and L	94
8.2.3	Motion Equation	94
8.2.4	Four Constraints	96
8.2.5	The Basic Scale of Construction World	99
8.3	MATH STEP	101

PART II PROTEIN MOLECULAR DYNAMICS 103

Chapter 9 MD PRINCIPLE 104

9.1	WHAT IS MD	104
9.2	CHARMM POTENTIAL	105
9.3	VERLET ALGORITHM OF NEWTON LAW	106
9.4	PARALLEL CALCULATION	108
9.5	ANNEALING TECHNIQUE AND ENVIRONMENTAL EFFECT	109
9.6	PHYSICAL QUANTITY	109
9.6.1	Structure	110

9. 6. 2	Statistical quantity	110
Chapter 10	NAMD PROGRAM	115
10. 1	GENERAL/METHOD TO OPERATE A PROGRAM	115
10. 2	NAMD PROGRAM	116
10. 3	INPUT FILES OF CHARMM VERSION	117
10. 3. 1	Topology File	117
10. 3. 2	Parameter File	122
10. 3. 3	How to Build a Topology File	125
10. 3. 4	Building Input Files	139
10. 4	INPUT FILES OF X-PLOR VERSION	164
10. 4. 1	CNS Topology and Parameter Files	165
10. 4. 2	How to Run the Program xplor64	165
10. 4. 3	The Water Environment Program Solvate in X-PLOR	170
10. 5	RUNNING NAMD PROGRAM	171
10. 5. 1	Running Command	171
10. 5. 2	Two Important Errors when Running MD	173
10. 6	OUTPUT FILES OF NAMD	173
10. 6. 1	Log File	174
10. 6. 2	DCD Output File	180
10. 6. 3	The Program catdcd and trio in Utilities Library	187
10. 6. 4	VMD Program	190
Chapter 11	EXAMPLE; DYNAMICAL TRANSITION*	194
11. 1	BUILDING INPUT FILES	194
11. 1. 1	Building PDB File and PSF File	195
11. 1. 2	Write Configuration File	201
11. 2	APPROACH OF OUTPUT FILES	202
11. 2. 1	The Mean Square Proton Displacement $\langle u^2 \rangle$ vs Temperature	202
11. 2. 2	The Fractal Dimensions of the Free Energy Landscape	207
11. 2. 3	The Density Map of Mean Square Displacements	211

Chapter 12	KNOCK OUT OF PROTEIN	216
12.1	KNOCK OUT METHOD	216
12.2	THE PROCESS OF KNOCKING OUT	216

PART III FULL ELECTRON STRUCTURE

CALCULATION OF PROTEIN	227
------------------------------	-----

Chapter 13 ELECRON STRUCTURE THEORY OF MACROMOLECULE

13.1	MOLECULE'S SCHRÖDINGER EQUATION	228
13.1.1	Schrödinger Equation of a Whole Molecule	228
13.1.2	Born-Oppenheimer Approximation	228
13.2	DENSITY FUNCTIONAL THEORY (DFT)	229
13.2.1	Hohenberg-Kohn Theorem	229
13.2.2	Kohn-Sham Equation	230
13.2.3	Solution on Kohn-Sham equation	231
13.2.4	Density Function, Operator and Matrix	234
13.3	MOLECULAR ORBIT THEORY	236
13.3.1	Molecular Orbit Theory	236
13.3.2	Hartree-Fock Equation	237
13.3.3	Hartree-Fock-Roothan Equation	238
13.4	SOME KNOWLEDGE ON MOLECULE	239
13.4.1	Double-electron Integral	239
13.4.2	Density Matrix, Mullikon Population and Charge Distribution	240
13.4.3	Molecular Orbital	242
13.4.4	Configuration Interaction	243
13.4.5	The Electron Distribution in a Molecule and σ , π and δ Bond	243
13.4.6	Hybridised Orbital	244
13.4.7	Hydrogen Bond	245
13.4.8	Frontier Molecular Orbital Theory (FMO)	245
13.4.9	Hand Molecule	246
13.4.10	Effect of Water	246
13.5	SEMIEMPIRICAL METHOD	246
13.5.1	Semiempirical Method	246
13.5.2	HMO and EHMO	246

Chapter 14	HF SOLUTION OF PROTEIN	248
14.1	OVERLAPPING DIMER APPROXIMATION (ODA) METHOD	248
14.1.1	Break down a Protein Sequence as Dimer	249
14.1.2	Dimer's Surrounding Environment and Water Environment	251
14.1.3	Calculation of Dimer's H and S Matrix	253
14.1.4	From whole H and S Matrix to $U_n(x)$ Function	253
14.2	ENFC	254
Chapter 15	ELECTRON STRUCTURE PROGRAM	256
15.1	ODA PROGRAM*	256
15.2	PROGRAMS AND PREPARING WORK	256
15.3	RUNNING ODA PROGRAM STEP BY STEP	261
Chapter 16	EXAMPLE OF ELECTRON STRUCTURE	270
Chapter 17	PROTEIN-LIGAND INTERACTION	279
17.1	A NEW WAY FOR PROTEIN-LIGAND INTERACTION	279
17.2	WAVE FUNCTIONS INTERACTION	280
17.2.1	Perturbation Theory of Quantum Mechanics	280
17.2.2	The Perturbation Theory on Combination of Two Molecules	280
17.2.3	Interaction between Two Wave Functions	282
17.3	FREE ENERGY CALCULATION ^[28~31]	284
17.3.1	General Theory of Free Energy	285
17.3.2	Adapted Biasing Force (ABF) Method	286
17.3.3	Running ABF in NAMD Program	289
17.3.4	Other Methods for Calculating Free Energy	298
17.4	EXAMPLE; FKBP12-FK506*	299
17.4.1	Studying Flowchart	299
17.4.2	MD Simulation	299
17.4.3	Frontier Orbital Calculation	301
17.4.4	Activity Atom Interaction on Perturbation Theory	306
17.4.5	Free Energy Calculation to Determine Protein-Ligand Binding Strength	308

17. 4. 6 NMR Experiments Testing Conclusion	314
Answer for Some Programs	318
INDEX	320
REFERENCES	324

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, **Xwin32** 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.