
"This book has already made me more confident in confronting the large amounts of data that face me in day-to-day research."

—Ronald Jenner, The Natural History Museum, London, UK

"Honestly, I've made more progress on my dissertation in the last week than in the last six months. I've been feeling very lucky to have read this book."

—Katie Mach, graduate student

"Incredibly useful book. I couldn't get through it fast enough. I was literally using what I learned the day after I read it."

—Dan Barshis, postdoc

"There's a better way to do what you're doing, and this book empowers you to do so. It is essential to any biologist, but also to any scientist or computer user wanting to do their work more efficiently."

—Julie Stewart, graduate student

Practical Computing for Biologists shows you how to use general computing tools to work more effectively. It pulls together in one place a broad range of powerful and flexible tools that are applicable to ecologists, molecular biologists, physiologists, and anyone who has struggled with large or complex data sets. Going beyond the subjects taught in most programming and bioinformatics courses, the book covers:

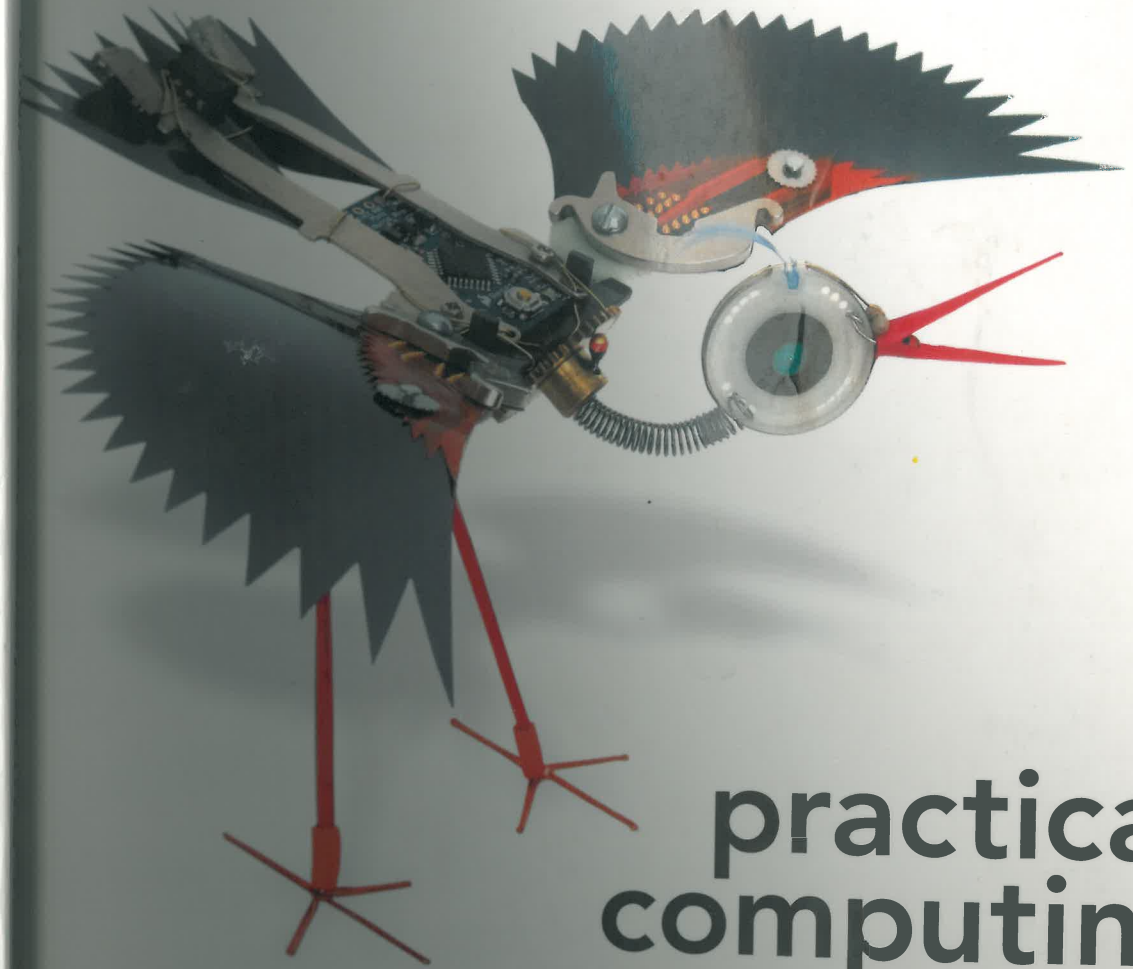
- Reformatting data with regular expressions
 - The Unix command line
 - Combining and automating analyses
 - Python programming and debugging
 - Creating and editing graphics
 - Databases
 - Performing analyses on remote computers
-

ISBN 978-0-87893-391-4



9

www.sinauer.com



practical computing for biologists

HADDOCK • DUNN

CONTENTS IN BRIEF

Before You Begin 1

PART I: Text Files 7

- Chapter 1 *Getting Set Up* 9
- Chapter 2 *Regular Expressions: Powerful Search and Replace* 17
- Chapter 3 *Exploring the Flexibility of Regular Expressions* 31

PART II: The Shell 45

- Chapter 4 *Command-line Operations: The Shell* 47
- Chapter 5 *Handling Text in the Shell* 67
- Chapter 6 *Scripting with the Shell* 83

PART III: Programming 103

- Chapter 7 *Components of Programming* 105
- Chapter 8 *Beginning Python Programming* 125
- Chapter 9 *Decisions and Loops* 141
- Chapter 10 *Reading and Writing Files* 173
- Chapter 11 *Merging Files* 201
- Chapter 12 *Modules and Libraries* 215
- Chapter 13 *Debugging Strategies* 231

PART IV: Combining Methods 243

- Chapter 14 *Selecting and Combining Tools* 245
- Chapter 15 *Relational Databases* 255
- Chapter 16 *Advanced Shell and Pipelines* 299

PART V: Graphics 321

- Chapter 17 *Graphical Concepts* 323
- Chapter 18 *Working with Vector Art* 345
- Chapter 19 *Working with Pixel Images* 363

PART VI: Advanced Topics 381

- Chapter 20 *Working on Remote Computers* 383
- Chapter 21 *Installing Software* 411
- Chapter 22 *Electronics: Interacting with the Physical World* 425

Appendices 449

CONTENTS

Acknowledgments xviii

BEFORE YOU BEGIN 1

- Introduction 1
- Why this book? 1
- Why biologists? 2
- Is this about using a particular computer or program? 3

To readers who will use this book on their own 4

To teachers using this book 4

Beyond this book 5

How to use this book 5

PART I Text Files 7

Chapter 1

GETTING SET UP 9

- An introduction to text manipulation 9
 - What are text files? 10
 - The organization of data within a text file 11
- Text editors 12
 - Installing TextWrangler 12
 - Optimizing text appearance within a text editor 13
 - Line endings 13
- The example files 14
 - Installing the example files 14
 - Exploring the example files 15
- Summary 15

Chapter 2

REGULAR EXPRESSIONS: POWERFUL SEARCH AND REPLACE 17

- A widespread language for search and replace 17
- Understanding the components of this new toolbox 18
 - Setting up the text editor 18
 - Your first wildcard: `\w` for letters and digits 20
 - Capturing text with `()` 21
 - Quantifiers: Matching one or more entities using `+` 23
 - Escaping punctuation characters with `\` 25
 - More special search terms: `\s \t \r . \d` 26
- Example: Reformatting molecular data files 28
- Comments about generating regular expressions 29
- Summary 29

Chapter 3

EXPLORING THE FLEXIBILITY OF REGULAR EXPRESSIONS 31

Character sets: Making your own wildcards 31

Defining custom character sets with [] 31

Applying custom character sets 32

Negations: Defining custom character sets with [^] 33

PART II The Shell 45

Chapter 4

COMMAND-LINE OPERATIONS: THE SHELL 47

Getting started: Don't fear the command line 47

Starting the shell and getting oriented 48

Starting the shell 48

A command-line view of the filesystem 50

The path 51

Navigating your computer from the shell 52

Listing files with `ls` and figuring out where you are with `pwd` 52

How to move around with `cd` 54

Signifying the home directory with `~` 56

Adding and removing directories with `mkdir` and `rmdir` 56

Copying files 57

Moving files 58

Command line shortcuts 59

Up arrow 59

Tab 60

Modifying command behavior with arguments 61

Viewing file contents with `less` 62

Boundaries: `^` beginnings and endings\$ 35

Adding more precision to quantifiers 36

Another quantifier: `*` for zero or more 36

Modifying greediness with `?` 36

Controlling the number of matches with `{}` 37

Putting it all together 38

Generating the replacement query 40

Constructing robust searches 41

Summary 42

Moving forward 42

Viewing help files at the command line with `man` 63

The command line finally makes your life easier 64

Wildcards in path descriptions 64

Copying and moving multiple files 65

Ending your terminal session 65

Summary 66

Recommended reading 66

Chapter 5

HANDLING TEXT IN THE SHELL 67

Editing text files at the command line with `nano` 67

Controlling the flow of data in the shell 69

Redirecting output to a file with `>` 69

Displaying and joining files with `cat` 70

Regular expressions at the command line with `grep` 72

Working with a larger dataset 72

Extracting particular rows from a file 73

Redirecting output from one program to another with pipe `|` 75

Searching across multiple files with `grep` 76

Refining the behavior of `grep` 77

Retrieving Web content using `curl` 78

Other shell commands 81

Summary 81

Chapter 6

SCRIPTING WITH THE SHELL 83

Combining commands 83

The search path 84

How the command line finds its commands 84

Creating your workspace, the scripts folder 85

Editing your `.bash_profile` settings file 86

PART III Programming 103

Chapter 7

COMPONENTS OF PROGRAMMING 105

What is a program? 105

Goals of the next few chapters 105

Practical programming 106

Variables 108

The anatomy of a variable 108

Basic variable types 108

Variables as containers for other variables 110

Arrays and lists 110

Converting between types 111

Variables in action 112

Mathematical operators 112

Comparative and logical operators 113

Functions 115

Flow control 115

Decisions with the `if` statement 115

Checking your new `$PATH` 88

Turning a text file into software 88

Control how the text is interpreted with `#!` 89

Making the text file executable by adjusting the permissions 90

Generating scripts automatically 91

Copying files in bulk 92

Flexible file renaming 95

Automating `curl` to retrieve literature references 97

General approaches to `curl` scripting 99

Aliases 99

Summary 101

Moving forward 101

Looping with `for` and `while` 116

Using lists and dictionaries 118

Lists 118

Dictionaries 119

Other data types 119

Input and output 120

User interaction 120

Files 120

Libraries and modules 122

Comment statements 122

Objects 122

Summary 124

Chapter 8

BEGINNING PYTHON PROGRAMMING 125

Why Python 125

Writing a program 126

Getting a program to run 126

Constructing the <code>dnacalc.py</code> program	127
Simple <code>print</code> statements	128
The <code>len()</code> function	130
Converting between variable types with <code>str()</code> , <code>int()</code> , and <code>float()</code>	131
The built-in string function <code>.count()</code>	132
Math operations on integers and floating point numbers	132
Adding comments with <code>#</code>	134
Controlling string formatting with the <code>%</code> operator	135
Getting input from the user	137
Gathering user input with <code>raw_input()</code>	137
Sanitizing variables with <code>.replace()</code> and <code>.upper()</code>	137
Reflecting on your program	140
Summary	140

Chapter 9 DECISIONS AND LOOPS 141

The Python interactive prompt	141
Getting Python help	144
Adding more calculations to <code>dnacalc.py</code>	144
Conditional statements with <code>if</code>	145
Designating code blocks using indentation	145
Logical operators	146
The <code>if</code> statement	147
The <code>else:</code> statement	147
Introducing <code>for</code> loops	149
A brief mention of lists	150
Writing the <code>for</code> loop in <code>proteincalc.py</code>	151
Generating dictionaries	151
Other dictionary functions	157
Applying your looping skills	158

Lists revisited	159
Indexing lists	159
Unpacking more than one value from a list	161
The <code>range()</code> function to define a list	161
A comparison of lists and strings	163
Converting between lists and strings	164
Adding elements to lists	165
Removing elements from lists	166
Checking the contents of lists	166
Sorting lists	167
Identifying unique elements in lists and strings	167
List comprehension	168
Summary	171
Moving forward	172

Chapter 10 READING AND WRITING FILES 173

Surveying the goal	173
Reading lines from a file	175
Considerations before reading a data file	175
Opening and reading a text file	177
Removing line endings with <code>.strip()</code>	178
Skipping the header line	179
Parsing data from lines	180
Splitting a line into data fields	180
Selecting elements from a list	181
Writing to files	182
Recapping basic file reading and writing	184
Parsing values with regular expressions	184
Importing the <code>re</code> module	185
Using regular expressions with the <code>re</code> module	185

Summary of using <code>re.search()</code> and <code>re.sub()</code>	187
Creating custom Python functions with <code>def</code>	188
Packaging data in a new format	192
Examining markup language	192
Preserving information during conversion	194
Converting to KML format	194
KML file format	194
Generating the KML text	195
Summary	198
Moving forward	199

Chapter 11 MERGING FILES 201

Reading from more than one file	201
Getting user input with <code>sys.argv</code>	202
Converting arguments to a file list	204
Providing feedback with <code>sys.stderr.write()</code>	205
Looping through the file list	206
Printing the output and generating a header line	208
Avoiding hardcoded software	209
Other applications of file reading	211
Summary	213
Moving forward	213

Chapter 12 MODULES AND LIBRARIES 215

Importing modules	216
More built-in modules from the standard library	218

The <code>urllib</code> module	218
The <code>os</code> module	218
The <code>math</code> module	219
The <code>random</code> module	219
The <code>time</code> module	221
Third-party modules	222
NumPy	223
Biopython	225
Other third-party modules	226
Making your own modules	227
Going further with Python	228
Summary	229
Moving forward	229

Chapter 13 DEBUGGING STRATEGIES 231

Learning by debugging	231
General strategies	232
Build upon working elements	232
Think about your assumptions	233
Specific debugging techniques	234
Isolate the problem	234
Write verbose software	235
Error messages and their meanings	237
Common Python errors	237
Shell errors	238
Making your program more efficient	238
Optimization	238
Try and except to handle errors	239
When you're really stuck	240
Summary	241
Moving forward	241

PART IV Combining Methods 243

Chapter 14

SELECTING AND COMBINING TOOLS 245

Your toolkit 245

Categories of data processing tasks 246

Getting digital data 246

Reformatting text files 249

Python scripts 251

General considerations 252

Summary 254

Moving forward 254

Chapter 15

RELATIONAL DATABASES 255

Spreadsheets and data organization 255

Data management systems 257

Anatomy of a database 259

Installing MySQL 260

Getting started with MySQL and SQL 262

Connecting to the MySQL server at the command line 262

Creating a database and tables 264

Adding rows of data to tables and displaying table contents 269

Interacting with MySQL from Python 271

Parsing the input text 271

Formulating SQL from the data 273

Executing SQL commands from Python 275

Bulk-importing text files into a table 279

Creating the `ctd` table 280

Importing data files with the `LOAD DATA` command 281

Exporting and importing databases as SQL files 283

Exploring data with SQL 283

Summarizing tables with `SELECT` and `COUNT` 283

Collating data with `GROUP BY` 285

Mathematical operations in SQL 286

Refining selections by row with `WHERE` 286

Modifying rows with `UPDATE` 289

Selecting data across tables 290

Generating output using Python 291

Looking ahead 294

Database users and security 294

Creating a root password 294

Adding a new MySQL user 295

Summary 296

Moving forward 297

Recommended reading 297

Chapter 16

ADVANCED SHELL AND PIPELINES 299

Additional useful shell commands 299

Extract lines with `head` and `tail` 299

Extract columns with `cut` 300

Sorting lines with `sort` 301

Isolating unique lines with `uniq` 302

Combining advanced shell functions 303

Approximate searches with `agrep` 306

Additional `grep` tips 307

Remember aliases? 308

Functions 309

Functions with user input 313

A dictionary function 313

Translating characters 313

Looping through all arguments passed to a function 314

Removing file extensions 315

Finding files 316

Revisiting piped commands 317

Repeating operations with loops 317

Wrappers 318

PART V Graphics 321

Chapter 17

GRAPHICAL CONCEPTS 323

Introduction 323

General image types 324

Vector versus pixel 324

Deciding when to use vector art, pixel art, or both 325

Image resolution and dimensions 326

Image resizing and the DPI misconception 328

Image colors 330

Color models and color space 330

Converting between color models 332

Color gamut and color profiles 333

Color choices 334

Summarizing the decision-making process 335

Layers 337

General considerations for presenting data 337

Eliminate visual clutter 337

Use transparency for overlapping data 338

Make effective use of space 338

Consistency 340

Maintaining data integrity 341

Why you should avoid PowerPoint 342

Summary 342

Moving forward 342

Recommended reading 343

Thoughts on pipelines 319

Summary 320

Recommended reading 320

Chapter 18

WORKING WITH VECTOR ART 345

Vector art mechanics 345

File formats 345

Generating vector art 346

Exporting images from another program 346

Drawing new images 347

Tracing photographs 347

Anatomy of vector art 348

Bézier curves 348

Stroke and fill 349

Working with vector art editors 349

Selecting and manipulating entire objects 350

Selecting and manipulating parts of an object 351

Creating Bézier curves with the pen tool 351

Modifying Bézier curves 352

The Join function 353

Stroke and fill 353

Layers 354

Illustrator tips 355

Inkscape tips 357

A typical workflow 358

Creating regularly arranged objects 359

Best practices for composing vector objects 361

Summary 361

Moving forward 362

Chapter 19 WORKING WITH PIXEL IMAGES 363

- Image compression 363
 - General principles 363
 - Implications for image workflows 364
- Pixel image file formats 364
 - Transparency 366
- Pixel art editors 366
- Working with pixel images 366
 - Masks and nondestructive editing 366
 - Levels adjustment 367
 - Grayscale images 368
 - Antialiasing 368
 - Layers 369
 - Colors in GIMP 369

PART VI Advanced Topics 381

Chapter 20 WORKING ON REMOTE COMPUTERS 383

- Connecting to a remote computer 383
 - Clients and servers 383
 - Typical scenarios for remote access 384
 - Finding computers: IP addresses, hostnames, and DNSs 385
 - Security 387
- Secure command-line connections with ssh 387
 - The ssh command 388
 - Troubleshooting ssh 388
 - Working on the remote machine 389
- Transferring files between computers 390
 - File archiving and compression 390
 - File transfer with sftp 391

- Photoshop shortcuts 370
- Command-line tools for image processing 370
 - The sips program 371
 - ImageMagick: convert and mogrify 371
 - ExifTool 372
- Image creation and analysis tools 372
 - ImageJ 372
 - MATLAB 374
 - R 374
 - Animations 375
- Photography 375
 - Aperture and exposure time 375
 - Color balance 378
 - Automatic versus manual operation 378
- Summary 379
- Moving forward 379

- Copying files with scp 392
- Other file transfer programs using SFTP 393
- Other file sharing protocols 393
- Full GUI control of a remote computer with VNC 393
- Troubleshooting remote connections 394
 - Getting local with a Virtual Private Network (VPN) 394
 - Mapping network connections with traceroute 395
 - Configuring the `[backspace]` key 395
- Controlling how programs run 396
 - Terminating a process 397
 - Starting jobs in the background with `&` 397
 - Checking job status with `ps` and `top` 397
 - Suspending jobs and sending them to the background 399
 - Stopping processes with `kill` 400

- Keeping jobs alive with `nohup` 402
- Changing program priority with `renice` 403
- High-performance computing 403
 - Parallel programs 404
 - Job management tools on clusters 405
- Setting up a server 405
 - Configuring the ssh server 406
 - Finding your addresses 407
 - Connecting to your own computer with ssh 408
- Summary 409

Chapter 21 INSTALLING SOFTWARE 411

- Overview 411
- Interpreted and compiled programs 412
- Approaches to installing software 414
 - `Readme.txt` and `Install.txt` 414
 - Installing programs from precompiled binaries 414
 - Automated installation tools 414
- Installing command-line programs from source code 415
 - Getting your computer ready 416
 - Unarchiving the source code 416
- Compiling and installing binaries 417
 - Variation 1: Off-the-shelf Makefile 418
 - Variation 2: Generating a Makefile with `./configure` 419
 - Installing Python modules 420
- Troubleshooting 421
 - What to do when software won't compile or installations don't work 421
- Summary 423
- Moving forward 423

Chapter 22 ELECTRONICS: INTERACTING WITH THE PHYSICAL WORLD 425

- Custom electronics in biology 425
 - Typical scenarios for custom electronics in biology 425
 - Simple circuits with complex micro-controllers 426
- Basic electronics 428
 - Electricity 428
 - Basic components 429
- Encoding information with electric signals 430
 - Analog encoding 430
 - Digitally encoded signals 431
- Building circuits 433
 - Schematics 433
 - Breadboards 433
 - Translating a schematic to a breadboard 434
- Serial communication in practice 435
 - Baud rate and other settings 436
 - Null modem 436
 - Software for serial communication 437
 - Serial comms through Python 437
- Arduino microcontroller boards in practice 438
 - Where to start 438
 - Building circuits with Arduino 439
 - Programming Arduino 440
- Other options for data acquisition 443
- Common sources of confusion 445
 - Measuring voltage 445
 - Current flow versus electron flow 445
 - Pull-up and pull-down resistors 445
- Summary 446
- Moving forward 446
- Recommended reading 447

Appendices 449

Appendix 1 WORKING WITH OTHER OPERATING SYSTEMS 451

- Microsoft Windows 451
 - Should I work in Windows or install Linux?* 451
 - Text editors for text editing and regular expressions in Chapters 1–3* 452
 - Cygwin for emulating Unix shell operations in Chapters 4–6* 453
 - Python on Windows for Chapters 8–12* 455
 - Working with MySQL on Windows for Chapter 15* 457
 - Working with vector and pixel art in Windows for Chapters 17–19* 457
- Linux 458
 - Installing Linux* 458
 - Text editing and regular expressions with jEdit for Chapters 1–3* 463
 - Using the Linux shell for shell operations in Chapters 4–6* 464
 - Python on Linux for Chapters 8–12* 465
 - Working with MySQL for Chapter 15* 465
 - Working with vector and pixel art in Linux for Chapters 17–19* 466

Appendix 2 REGULAR EXPRESSION SEARCH TERMS 467

Appendix 3 SHELL COMMANDS 471

Appendix 4 PYTHON QUICK REFERENCE 479

- Conventions for this appendix 479
- Format, syntax, and punctuation in Python 479
- The command-line interpreter 480
- Command summary 480
 - Variable types and statistics* 480
- Strings 480
- Gathering user input 481
- Building strings 482
- Comparisons and logical operators 482
- Math operators 483
- Decisions 484
- Loops 484
- Searching with regular expressions 485
 - Regex to find matching subsets in a string* 485
 - Regex to substitute into a string* 485
- Working with lists 486
- List comprehension 488
- Dictionaries 488
- Creating functions 489
- Working with files 490
- Using modules and functions 491
- Miscellaneous Python operations 493
 - Presenting warnings and feedback* 493
 - Catching errors* 493
 - Shell operations within Python* 493
 - Reference and getting help* 493

Appendix 5 TEMPLATE PROGRAMS 495

- Python 2.7 or earlier 496
- Python 3 496
- Perl 496
- bash shell 497
- C 497
- C++ 498
- Java 498
- JavaScript 499
- PHP 500
- Ruby 500
- MATLAB 501
- R 501
- Arduino 502

Appendix 6 BINARY, HEX, AND ASCII 503

- Alternate base systems 503
- Hexadecimal 505
- ASCII and Unicode characters 505
- Images and color 506
- Decimal, hex, binary, and ASCII values 507

Appendix 7 SQL COMMANDS 511

INDEX 515