

Chapman & Hall/CRC
Mathematical and Computational Biology Series

ALGORITHMS FOR NEXT-GENERATION SEQUENCING

Wing-Kin Sung



CRC Press

Taylor & Francis Group

A CHAPMAN & HALL BOOK

ALGORITHMS FOR NEXT-GENERATION SEQUENCING

Advances in sequencing technology have allowed scientists to study the human genome in greater depth and on a larger scale than ever before – as many as hundreds of millions of short reads in the course of a few days. But what are the best ways to deal with this flood of data?

Algorithms for Next-Generation Sequencing is an invaluable tool for students and researchers in bioinformatics and computational biology, biologists seeking to process and manage the data generated by next-generation sequencing, and as a textbook or a self-study resource. In addition to offering an in-depth description of the algorithms for processing sequencing data, it also presents useful examples illustrating how the algorithms work.

Features

- One of the first books published on this key topic
- Written by a leading practitioner
- Focuses on algorithms
- Covers technologies used in next-generation sequencing
- Includes a wide range of case studies and applications

Wing-Kin Sung is a professor in the Department of Computer Science of the National University of Singapore and a senior group leader in the Genome Institute of Singapore. He has over 20 years of experience in algorithm and bioinformatics research.



CRC Press

Taylor & Francis Group
an informa business

www.crcpress.com

6000 Broken Sound Parkway, NW
Suite 300, Boca Raton, FL 33487
711 Third Avenue
New York, NY 10017
2 Park Square, Milton Park
Abingdon, Oxon OX14 4RN, UK

K16105

ISBN: 978-1-4665-6550-0

90000



9 781466 565500

www.crcpress.com

ALGORITHMS FOR NEXT-GENERATION SEQUENCING



Chapman & Hall/CRC Mathematical and Computational Biology Series

ALGORITHMS FOR NEXT-GENERATION SEQUENCING

Wing-Kin Sung



CRC Press

Taylor & Francis Group

Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business

A CHAPMAN & HALL BOOK

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2017 by Taylor & Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed at CPI on sustainably sourced paper
Version Date: 20170421

International Standard Book Number-13: 978-1-4665-6550-0 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (<http://www.copyright.com/>) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Visit the Taylor & Francis Web site at
<http://www.taylorandfrancis.com>

and the CRC Press Web site at
<http://www.crcpress.com>

ALGORITHMS FOR NEXT-GENERATION SEQUENCING

Preface

Next-generation sequencing (NGS) is a recently developed technology enabling us to generate hundreds of billions of DNA bases from the samples. We can use NGS to reconstruct the genome, understand genomic variations, recover transcriptomes, and identify the transcription factor binding sites or the epigenetic marks.

The NGS technology radically changes how we collect genomic data from the samples. Instead of studying a particular gene or a particular genomic region, NGS technologies enable us to perform genome-wide study unbiasedly. Although more raw data can be obtained from sequencing machines, we face computational challenges in analyzing such a big dataset. Hence, it is important to develop efficient and accurate computational methods to analyze or process such datasets. This book is intended to give an in-depth introduction to such algorithmic techniques.

The primary audiences of this book include advanced undergraduate students and graduate students who are from mathematics or computer science departments. We assume that readers have some training in college-level biology, statistics, discrete mathematics and algorithms.

This book was developed partly from the teaching material for the course on Combinatorial Methods in Bioinformatics, which I taught at the National University of Singapore, Singapore. The chapters in this book are classified based on the application domains of the NGS technologies. In each chapter, a brief introduction to the technology is first given. Then, different methods or algorithms for analyzing such NGS datasets are described. To illustrate each algorithm, detailed examples are given. At the end of each chapter, exercises are given to help readers understand the topics.

Chapter 1 introduces the next-generation sequencing technologies. We cover the three generations of sequencing, starting from Sanger sequencing (first generation). Then, we cover second-generation sequencing, which includes Illumina Solexa sequencing. Finally, we describe the third-generation sequencing technologies which include PacBio sequencing and nanopore sequencing.

Chapter 2 introduces a few NGS file formats, which facilitate downstream analysis and data transfer. They include fasta, fastq, SAM, BAM, BED, VCF and WIG formats. Fasta and fastq are file formats for describing the raw sequencing reads generated by the sequencers. SAM and BAM are file formats

for describing the alignments of the NGS reads on the reference genome. BED, VCF and WIG formats are annotation formats.

To develop methods for processing NGS data, we need efficient algorithms and data structures. Chapter 3 is devoted to briefly describing these techniques.

Chapter 4 studies read mappers. Read mappers align the NGS reads on the reference genome. The input is a set of raw reads in fasta or fastq files. The read mapper will align each raw read on the reference genome, that is, identify the region in the reference genome which is highly similar to the read. Then, the read mapper will output all these alignments in a SAM or BAM file. This is the basic step for many NGS applications. (It is the first step for the methods in Chapters 6–9.)

Chapter 5 studies the de novo assembly problem. Given a set of raw reads extracted from whole genome sequencing of some sample genome, de novo assembly aims to stitch the raw reads together to reconstruct the genome. It enables us to reconstruct novel genomes like plants and bacteria. De novo assembly involves a few steps: error correction, contig assembly (de Bruijn graph approach or base-by-base extension approach), scaffolding and gap filling. This chapter describes techniques developed for these steps.

Chapter 6 discusses the problem of identifying single nucleotide variations (SNVs) and small insertions/deletions (indels) in an individual genome. The genome of every individual is highly similar to the reference human genome. However, each genome is still different from the reference genome. On average, there is 1 single nucleotide variation in every 3000 bases and 1 small indel in every 1000 bases. To discover these variations, we can first perform whole genome sequencing or exome sequencing of the individual genome to obtain a set of raw reads. After aligning the raw reads on the reference genome, we use SNV callers and indel callers to call SNVs and small indels. This chapter is devoted to discussing techniques used in SNV callers and indel callers.

Apart from SNVs and small indels, copy number variations (CNVs) and structural variations (SVs) are the other types of variations that appear in our genome. CNVs and SVs are not as frequent as SNVs and indels. Moreover, they are more prone to change the phenotype. Hence, it is important to understand them. Chapter 7 is devoted to studying techniques used in CNV callers and SV callers.

All above technologies are related to genome sequencing. We can also sequence RNA. This technology is known as RNA-seq. Chapter 8 studies methods for analyzing RNA-seq. By applying computational methods on RNA-seq, we can recover the transcriptome. More precisely, RNA-seq enables us to identify exons and split junctions. Then, we can predict the isoforms of the genes. We can also determine the expression of each transcript and each gene.

By combining Chromatin immunoprecipitation and next-generation sequencing, we can sequence genome regions that are bound by some transcription factors or with epigenetic marks. Such technology is known as ChIP-seq. The computational methods that identify those binding sites are known

as ChIP-seq peak callers. Chapter 9 is devoted to discussing computational methods for such purpose.

As stated earlier, NGS data is huge; and the NGS data files are usually big. It is difficult to store and transfer NGS files. One solution is to compress the NGS data files. Nowadays, a number of compression methods have been developed and some of the compression formats are used frequently in the literatures like BAM, bigBed and bigWig. Chapter 10 aims to describe these compression techniques. We also describe techniques that enable us to randomly access the compressed NGS data files.

Supplementary material can be found at

http://www.comp.nus.edu.sg/~ksung/algo_in_ngs/.

I would like to thank my PhD supervisors Tak-Wah Lam and Hing-Fung Ting and my collaborators Francis Y. L. Chin, Kwok Pui Choi, Edwin Cheung, Axel Hillmer, Wing Kai Hon, Jansson Jesper, Ming-Yang Kao, Caroline Lee, Nikki Lee, Hon Wai Leong, Alexander Lezhava, John Luk, See-Kiong Ng, Franco P. Preparata, Yijun Ruan, Kunihiko Sadakane, Chialin Wei, Limsoon Wong, Siu-Ming Yiu, and Louxin Zhang. My knowledge of NGS and bioinformatics was enriched through numerous discussions with them. I would like to thank Ramesh Rajaby, Kunihiko Sadakane, Chandana Tennakoon, Hugo Willy, and Han Xu for helping to proofread some of the chapters. I would also like to thank my parents Kang Fai Sung and Siu King Wong, my three brothers Wing Hong Sung, Wing Keung Sung, and Wing Fu Sung, my wife Lily Or, and my three kids Kelly, Kathleen and Kayden for their support.

Finally, if you have any suggestions for improvement or if you identify any errors in the book, please send an email to me at ksung@comp.nus.edu.sg. I thank you in advance for your helpful comments in improving the book.

Wing-Kin Sung

Contents

Preface	xi
1 Introduction	1
1.1 DNA, RNA, protein and cells	1
1.2 Sequencing technologies	3
1.3 First-generation sequencing	4
1.4 Second-generation sequencing	6
1.4.1 Template preparation	6
1.4.2 Base calling	7
1.4.3 Polymerase-mediated methods based on reversible terminator nucleotides	7
1.4.4 Polymerase-mediated methods based on unmodified nucleotides	10
1.4.5 Ligase-mediated method	11
1.5 Third-generation sequencing	12
1.5.1 Single-molecule real-time sequencing	12
1.5.2 Nanopore sequencing method	13
1.5.3 Direct imaging of DNA using electron microscopy . .	15
1.6 Comparison of the three generations of sequencing	16
1.7 Applications of sequencing	17
1.8 Summary and further reading	19
1.9 Exercises	19
2 NGS file formats	21
2.1 Introduction	21
2.2 Raw data files: fasta and fastq	22
2.3 Alignment files: SAM and BAM	24
2.3.1 FLAG	26
2.3.2 CIGAR string	26
2.4 Bed format	27
2.5 Variant Call Format (VCF)	29
2.6 Format for representing density data	31
2.7 Exercises	33

3	Related algorithms and data structures	35
3.1	Introduction	35
3.2	Recursion and dynamic programming	35
3.2.1	Key searching problem	36
3.2.2	Edit-distance problem	37
3.3	Parameter estimation	38
3.3.1	Maximum likelihood	39
3.3.2	Unobserved variable and EM algorithm	40
3.4	Hash data structures	43
3.4.1	Maintain an associative array by simple hashing	43
3.4.2	Maintain a set using a Bloom filter	45
3.4.3	Maintain a multiset using a counting Bloom filter	46
3.4.4	Estimating the similarity of two sets using minHash	47
3.5	Full-text index	49
3.5.1	Suffix trie and suffix tree	49
3.5.2	Suffix array	50
3.5.3	FM-index	51
3.5.3.1	Inverting the BWT B to the original text T	53
3.5.3.2	Simulate a suffix array using the FM-index	54
3.5.3.3	Pattern matching	55
3.5.4	Simulate a suffix trie using the FM-index	55
3.5.5	Bi-directional BWT	56
3.6	Data compression techniques	58
3.6.1	Data compression and entropy	58
3.6.2	Unary, gamma, and delta coding	59
3.6.3	Golomb code	60
3.6.4	Huffman coding	60
3.6.5	Arithmetic code	62
3.6.6	Order- k Markov Chain	64
3.6.7	Run-length encoding	65
3.7	Exercises	66
4	NGS read mapping	69
4.1	Introduction	69
4.2	Overview of the read mapping problem	70
4.2.1	Mapping reads with no quality score	70
4.2.2	Mapping reads with a quality score	71
4.2.3	Brute-force solution	72
4.2.4	Mapping quality	74
4.2.5	Challenges	75
4.3	Align reads allowing a small number of mismatches	76
4.3.1	Mismatch seed hashing approach	77
4.3.2	Read hashing with a spaced seed	78
4.3.3	Reference hashing approach	82
4.3.4	Suffix trie-based approaches	84

4.3.4.1	Estimating the lower bound of the number of mismatches	87
4.3.4.2	Divide and conquer with the enhanced pigeon-hole principle	89
4.3.4.3	Aligning a set of reads together	92
4.3.4.4	Speed up utilizing the quality score	94
4.4	Aligning reads allowing a small number of mismatches and indels	97
4.4.1	q -mer approach	97
4.4.2	Computing alignment using a suffix trie	99
4.4.2.1	Computing the edit distance using a suffix trie	100
4.4.2.2	Local alignment using a suffix trie	103
4.5	Aligning reads in general	105
4.5.1	Seed-and-extension approach	107
4.5.1.1	BWA-SW	108
4.5.1.2	Bowtie 2	109
4.5.1.3	BatAlign	110
4.5.1.4	Cushaw2	111
4.5.1.5	BWA-MEM	112
4.5.1.6	LAST	113
4.5.2	Filter-based approach	114
4.6	Paired-end alignment	116
4.7	Further reading	117
4.8	Exercises	118
5	Genome assembly	123
5.1	Introduction	123
5.2	Whole genome shotgun sequencing	124
5.2.1	Whole genome sequencing	124
5.2.2	Mate-pair sequencing	126
5.3	De novo genome assembly for short reads	126
5.3.1	Read error correction	128
5.3.1.1	Spectral alignment problem (SAP)	129
5.3.1.2	k -mer counting	133
5.3.2	Base-by-base extension approach	138
5.3.3	De Bruijn graph approach	141
5.3.3.1	De Bruijn assembler (no sequencing error)	143
5.3.3.2	De Bruijn assembler (with sequencing errors)	144
5.3.3.3	How to select k	146
5.3.3.4	Additional issues of the de Bruijn graph approach	147
5.3.4	Scaffolding	150
5.3.5	Gap filling	153
5.4	Genome assembly for long reads	154

5.4.1	Assemble long reads assuming long reads have a low sequencing error rate	155
5.4.2	Hybrid approach	157
5.4.2.1	Use mate-pair reads and long reads to improve the assembly from short reads	160
5.4.2.2	Use short reads to correct errors in long reads	160
5.4.3	Long read approach	161
5.4.3.1	MinHash for all-versus-all pairwise alignment	162
5.4.3.2	Computing consensus using Falcon Sense	163
5.4.3.3	Quiver consensus algorithm	165
5.5	How to evaluate the goodness of an assembly	168
5.6	Discussion and further reading	168
5.7	Exercises	170
6	Single nucleotide variation (SNV) calling	175
6.1	Introduction	175
6.1.1	What are SNVs and small indels?	175
6.1.2	Somatic and germline mutations	178
6.2	Determine variations by resequencing	178
6.2.1	Exome/targeted sequencing	179
6.2.2	Detection of somatic and germline variations	180
6.3	Single locus SNV calling	180
6.3.1	Identifying SNVs by counting alleles	181
6.3.2	Identify SNVs by binomial distribution	182
6.3.3	Identify SNVs by Poisson-binomial distribution	184
6.3.4	Identifying SNVs by the Bayesian approach	185
6.4	Single locus somatic SNV calling	187
6.4.1	Identify somatic SNVs by the Fisher exact test	187
6.4.2	Identify somatic SNVs by verifying that the SNVs appear in the tumor only	188
6.4.2.1	Identify SNVs in the tumor sample by posterior odds ratio	188
6.4.2.2	Verify if an SNV is somatic by the posterior odds ratio	191
6.5	General pipeline for calling SNVs	192
6.6	Local realignment	193
6.7	Duplicate read marking	195
6.8	Base quality score recalibration	195
6.9	Rule-based filtering	198
6.10	Computational methods to identify small indels	199
6.10.1	Split-read approach	199
6.10.2	Span distribution-based clustering approach	200
6.10.3	Local assembly approach	203
6.11	Correctness of existing SNV and indel callers	204
6.12	Further reading	205

6.13 Exercises	206
7 Structural variation calling	209
7.1 Introduction	209
7.2 Formation of SVs	211
7.3 Clinical effects of structural variations	214
7.4 Methods for determining structural variations	215
7.5 CNV calling	217
7.5.1 Computing the raw read count	218
7.5.2 Normalize the read counts	219
7.5.3 Segmentation	219
7.6 SV calling pipeline	222
7.6.1 Insert size estimation	222
7.7 Classifying the paired-end read alignments	223
7.8 Identifying candidate SVs from paired-end reads	226
7.8.1 Clustering approach	227
7.8.1.1 Clique-finding approach	228
7.8.1.2 Confidence interval overlapping approach	229
7.8.1.3 Set cover approach	233
7.8.1.4 Performance of the clustering approach	236
7.8.2 Split-mapping approach	236
7.8.3 Assembly approach	237
7.8.4 Hybrid approach	238
7.9 Verify the SVs	239
7.10 Further reading	242
7.11 Exercises	242
8 RNA-seq	245
8.1 Introduction	245
8.2 High-throughput methods to study the transcriptome	247
8.3 Application of RNA-seq	248
8.4 Computational Problems of RNA-seq	250
8.5 RNA-seq read mapping	250
8.5.1 Features used in RNA-seq read mapping	250
8.5.1.1 Transcript model	250
8.5.1.2 Splice junction signals	252
8.5.2 Exon-first approach	253
8.5.3 Seed-and-extend approach	256
8.6 Construction of isoforms	260
8.7 Estimating expression level of each transcript	261
8.7.1 Estimating transcript abundances when every read maps to exactly one transcript	261
8.7.2 Estimating transcript abundances when a read maps to multiple isoforms	264
8.7.3 Estimating gene abundance	266

8.8	Summary and further reading	268
8.9	Exercises	268
9	Peak calling methods	271
9.1	Introduction	271
9.2	Techniques that generate density-based datasets	271
9.2.1	Protein DNA interaction	271
9.2.2	Epigenetics of our genome	273
9.2.3	Open chromatin	274
9.3	Peak calling methods	274
9.3.1	Model fragment length	276
9.3.2	Modeling noise using a control library	279
9.3.3	Noise in the sample library	280
9.3.4	Determination if a peak is significant	281
9.3.5	Unannotated high copy number regions	283
9.3.6	Constructing a signal profile by Kernel methods	284
9.4	Sequencing depth of the ChIP-seq libraries	285
9.5	Further reading	286
9.6	Exercises	287
10	Data compression techniques used in NGS files	289
10.1	Introduction	289
10.2	Strategies for compressing fasta/fastq files	290
10.3	Techniques to compress identifiers	290
10.4	Techniques to compress DNA bases	291
10.4.1	Statistical-based approach	291
10.4.2	BWT-based approach	292
10.4.3	Reference-based approach	295
10.4.4	Assembly-based approach	297
10.5	Quality score compression methods	299
10.5.1	Lossless compression	300
10.5.2	Lossy compression	301
10.6	Compression of other NGS data	302
10.7	Exercises	304
	References	307
	Index	339