

Contents

<i>Preface</i>	vii
1. Simple String Search	1
1.1 Storing a String in an Array	1
1.2 Brute-Force String Search	2
1.3 Encoding Strings into Integers	4
1.4 Sorting k -mer Integers and a Binary Search	7
1.5 Binary Search for the Boundaries of Blocks	8
2. Sorting	11
2.1 Insertion Sort	11
2.2 Merge Sort	12
2.3 Worst-Case Time Complexity of Algorithms	16
2.4 Heap Sort	17
2.5 Randomized Quick Sort	22
2.6 Improving the Performance of Quick Sort	27
2.7 Ternary Split Quick Sort	32
2.8 Radix Sort	33
3. Lookup Tables	39
3.1 Direct-Address Tables	39
3.2 Hash Tables	41
3.3 Table Size	44
3.4 Using the Frequencies of k -mers	45
3.5 Techniques for Reducing Table Size	46

4. Suffix Arrays	51
4.1 Suffix Trees	52
4.2 Suffix Arrays	54
4.3 Binary Search of Suffix Arrays for Queries	56
4.4 Using the Longest Common Prefix Information to Accelerate the Search	58
4.5 Computing the Longest Common Prefixes	62
4.5.1 Application to Occurrence Frequencies of k -mers . . .	65
4.5.2 Application to the Longest Common Factors	67
4.6 Suffix Array Construction – Doubling	68
4.7 Larsson-Sadakane Algorithm	69
4.8 Linear-Time Suffix Array Construction	74
4.9 A Note on Practical Performance	81
5. Space-Efficient String Search	83
5.1 Rabin-Karp Algorithm	83
5.2 Accelerating the Brute-Force String Search	86
5.3 Knuth-Morris-Pratt Algorithm	88
5.4 Bad Character Heuristics	94
6. Approximate String Search	99
6.1 Edit Operations and Alignments	101
6.2 Edit Graphs and Dynamic Programming	103
6.3 Needleman-Wunsch Algorithm	105
6.4 Smith-Waterman Algorithm for Computing Local Alignments	108
6.5 Overlap Alignments	111
6.6 Alignment of cDNA Sequences with Genomes and Affine Gap Penalties	114
6.7 Gotoh’s Algorithm for Affine Gap Penalties	116
6.8 Hirschberg’s Space Reduction Technique	120
7. Seeded Alignments	125
7.1 Sensitivity and Specificity	126
7.2 Computing Sensitivity and Specificity	129
7.3 Multiple Hits of Seeds	131
7.4 Gapped Seeds	134
7.5 Chaining and Comparative Genomics	135

7.6	Design of Highly Specific Oligomers	141
7.7	Seeds for Computing Mismatch Tolerance	145
7.7.1	Naive Algorithm	145
7.7.2	BYP Method	146
7.8	Partially Matching Seeds	147
7.9	Overlapping, Partially Matching Seeds	151
8.	Whole Genome Shotgun Sequencing	155
8.1	Sanger Method	156
8.1.1	Improvements to the Sequencing Method	159
8.2	Cloning Genomic DNA Fragments	160
8.3	Basecalling	163
8.4	Overview of Shotgun Sequencing	165
8.5	Lander-Waterman Statistics	170
8.6	Double-Stranded Assembly	172
8.7	Overlap-Layout-Consensus	175
8.7.1	Overlap	175
8.7.2	Layout	177
8.7.3	Consensus	180
8.8	Practical Whole Genome Shotgun Assembly	182
8.8.1	Vector Masking	183
8.8.2	Quality Trimming	186
8.8.3	Contamination Removal	187
8.8.4	Overlap and Layout	188
8.8.4.1	Seed and Extend	188
8.8.4.2	Seeding	190
8.8.4.3	Greedy Merging Approach	191
8.8.4.4	Longer Repeat Sequence	192
8.8.4.5	Iterative Improvements	193
8.8.4.6	Accelerating Overlap Detection	194
8.8.4.7	Repeat Sequence Detection	196
8.8.4.8	Error Correction	199
8.8.4.9	Repeat Separation	199
8.8.4.10	Maximal Read Heuristics	201
8.8.4.11	Paired Pair Heuristics	202
8.8.4.12	Parallelization	203
8.8.4.13	Eliminating Chimeric Reads	204
8.8.5	Scaffolding	205

8.8.5.1 How Many Mate Pairs Are Needed for Scaffolding?	207
8.8.5.2 Iterative Improvements	208
8.8.6 Consensus	210
8.9 Quality Assessment	211
8.10 Past and Future	213
<i>Software Availability</i>	221
<i>Bibliography</i>	223
<i>Index</i>	233