

Explorations In Searching Compressed Nucleic Acid And Protein Sequence Databases And Their Cooperatively-Compressed Indices

Paul Gardner-Stephen

Bachelor of Science,

Flinders University of South Australia

A Thesis Submitted for the Degree of Doctor of Philosophy

Flinders University

School of Informatics and Engineering

Adelaide, South Australia

2008

(Submitted 20 December 2007)

Acknowledgements

Before I acknowledge those who have helped me make it through my candidature, I wish to thank the unbroken chain of people who have believed in me, encouraged me, nurtured and supported me, and without whom I would not have made it to the starting line: Mrs Jan Stephen, my mother, who first kindled my interest in science, and Mr. Keith Gardner, my father; Mr. Peter Brune, a primary school teacher, who believed that I would be someone great one day (perhaps I will get there yet); Dr. Allen Hodson, a high school teacher who taught me more lessons about life and statistics than I suspect he realises; Dr. Todd Rockoff, a university lecturer who kindled my interest in hardware design; Mr. Michael Renz of Germany, who taught me my first lessons about the commercial world; Mr. Murray Rogers, who took a scruffy young university student and made a respectable (but still scruffy) Systems Administrator; and Prof. Greg Knowles, who had enough faith in me to take me on as his student.

For their support throughout my candidature, I wish to thank again: Mr. Murray Rogers for graciously releasing my time so that I could study; and Prof. Greg Knowles for his support and investment as my supervisor. In addition to these people I wish to thank: Prof. Janet Verbyla as head of the School of Informatics & Engineering, and my colleagues (academic staff, general staff and fellow candidates) for their support and friendship during my candidature, and also my friends who have not abandoned me, even though “as busy as a Gardner-Stephen” has become a proverb among them.

A special mention must go to Ms. Fran Banytis of the Staff Development & Training Unit, for the way she has invested much of her self into my candidature. Her encouragement and support throughout my candidature has been as valuable as it has been unexpected.

Finally I would like to thank Dr Dione Gardner-Stephen, my beautiful wife who not only loved me enough to trade in her perfectly good maiden name for one that no-one can spell, but believed in me and supported me through the thick and thin of candidature, not only with the insight and understanding of another doctoral candidate, but in the way that only a loving wife can. I love you much more than words can express.

Through the actions of all these people I see the guiding hand of a God of love, and who has been gracious enough, not merely to bring me to this place, but to invite me to know Him personally. That God is Jesus Christ of Nazareth, and to Him I say “thank you” for the contributions that each of you have made in my life.

Abstract

Nucleic acid and protein databases such as GenBank are growing at a rate that perhaps eclipses even Moore's Law of increase in computational power¹. This poses a problem for the biological sciences, which have become increasingly dependant on searching and manipulating these databases. It was once reasonably practical to perform exhaustive searches of these databases, for example using the algorithm described by Smith and Waterman, however it has been many years since this was the case. This has led to the development of a series of search algorithms, such as FASTA, BLAST and BLAT, that are each successively faster, but at similarly successive costs in terms of thoroughness.

Attempts have been made to remedy this problem by devising search algorithms that are both fast and thorough. An example is CAFE, which seeks to construct a search system with a sub-linear relationship between search time and database size, and argues that this property must be present for any search system to be successful in the long term.

This dissertation explores this notion by seeking to construct a search system that takes advantage of the growing redundancy in databases such as GenBank in order to reduce both the search time and the space required to store the databases and their indices, while preserving or increasing the thoroughness of the search.

The result is the creation and implementation of new genomic sequence search and alignment, database compression, and index compression algorithms and systems that make

¹More accurately, Moore's Law predicts that the capacity for transistors on an integrated circuit will double approximately every two years. In practice, due to the efforts of computer architects this has translated into a roughly corresponding increase in computation throughput.

progress toward resolving the problem of reducing search speed and space requirements while improving sensitivity. However, success is tempered by the need for databases with adequate local redundancy, and the computational cost of these algorithms when servicing un-batched queries.

"I Paul Gardner-Stephen, certify that this thesis does not incorporate without acknowledgment any material previously submitted for a degree or diploma in any university; and that to the best of my knowledge and belief it does not contain any material previously published or written by another person except where due reference is made in the text."

Candidate:

Paul Gardner-Stephen

Contents

| | |
|---|-----------|
| Acknowledgements | ii |
| Abstract | iv |
| I Introductory | 1 |
| 1 Introduction | 2 |
| 1.1 Motivation | 2 |
| 1.2 Statement And Scope Of Thesis | 4 |
| 1.2.1 Introduction | 4 |
| 1.2.2 Revealing Recurrences Through Data Compression | 5 |
| 1.2.3 Using Recurrences To Compress Indices | 7 |
| 1.2.4 Using Recurrences To Improve Search Sensitivity | 8 |
| 1.2.5 Opportunity And Research Questions | 11 |
| 1.2.6 Summary And Statement Of Thesis | 12 |
| 1.3 Assumptions | 13 |
| 1.4 Contributions | 14 |
| 1.5 Structure Of Thesis | 15 |

| | | |
|----------|---|-----------|
| 2 | Background | 18 |
| 2.1 | Sequence Alignment | 18 |
| 2.1.1 | Sequence Search And Alignment | 18 |
| 2.1.1.1 | Bioinformatics And Sequence Similarity Searching | 18 |
| 2.1.1.2 | Exact Sub-String Alignment | 20 |
| 2.1.1.3 | Non-Exact Sub-String (Non-Gapped) Alignment | 21 |
| 2.1.1.4 | Non-Exact Gapped Alignment | 23 |
| 2.1.1.5 | Dynamic Programming | 24 |
| 2.1.1.6 | Proteomic Similarity | 30 |
| 2.1.1.7 | Biological And Statistical Significance Of Alignments | 31 |
| 2.1.2 | Comparing The Performance Of Sequence Search And Alignment Algorithms | 33 |
| 2.1.2.1 | Human Judgement | 33 |
| 2.1.2.2 | Benchmarks | 34 |
| 2.1.2.3 | Sensitivity Metrics | 35 |
| 2.2 | Accelerating Sequence Searching | 36 |
| 2.2.1 | Heuristic Algorithms | 37 |
| 2.2.1.1 | A Brief Comparison Of Selected Heuristic Algorithms | 37 |
| 2.2.1.2 | FASTA | 37 |
| 2.2.1.3 | BLAST | 38 |
| 2.2.1.4 | BLAT | 40 |
| 2.2.1.5 | FLASH | 41 |

| | | |
|----------|---|-----------|
| 2.2.1.6 | CAFE | 41 |
| 2.2.1.7 | Acceptance Of Heuristic Algorithms | 43 |
| 2.2.1.8 | Ignorance Of Users To Specific Heuristic Trade-Offs | 44 |
| 2.2.2 | Clustering (Parallel Computing) | 45 |
| 2.2.3 | Indexing | 46 |
| 2.2.4 | Summary | 47 |
| 2.3 | Compression | 48 |
| 2.3.1 | Introduction | 48 |
| 2.3.2 | Entropy Coding Methods | 48 |
| 2.3.3 | Dictionary Methods | 51 |
| 2.3.4 | Statistical Modelling Methods | 56 |
| 2.3.5 | Performance Of Compression Algorithms | 58 |
| 2.3.6 | Burrows-Wheeler Transform | 61 |
| 2.3.7 | Synchronisation | 64 |
| 2.3.8 | DNA Compression | 65 |
| 2.4 | Constructing Compact Indices | 66 |
| 2.4.1 | Compressing Index Postings | 68 |
| 2.4.2 | Efficient Index Construction | 71 |
| 2.4.3 | Document Reordering And Filtering | 72 |
| 2.4.4 | A Compelling Opportunity: Cooperative Compression | 72 |
| 3 | Materials And Methods | 74 |
| 3.1 | Selection Of Databases | 75 |

| | | |
|---------|--|----|
| 3.1.1 | Nucleic Acid | 75 |
| 3.1.2 | Protein | 77 |
| 3.2 | Query Selection | 77 |
| 3.3 | Speed And Sensitivity | 79 |
| 3.4 | Peer Group Of Algorithms | 81 |
| 3.4.1 | Smith-Waterman (SSEARCH 3.4t25) | 81 |
| 3.4.2 | BLAST (NCBI-BLAST 2.2.6) | 83 |
| 3.4.3 | BLAT | 84 |
| 3.4.4 | Academic Version Of PatternHunter | 84 |
| 3.4.5 | FASTA | 85 |
| 3.4.6 | CAFE | 85 |
| 3.4.7 | Algorithms Introduced In This Dissertation | 86 |
| 3.5 | Batching Environment | 86 |
| 3.5.1 | Overview | 86 |
| 3.5.2 | Directory Structure | 87 |
| 3.5.2.1 | Top Level Directories | 87 |
| 3.5.3 | Generation Of Standard Queries | 89 |
| 3.5.4 | Execution Of Batches | 90 |
| 3.5.4.1 | Executing A Batch | 90 |
| 3.5.4.2 | Summarisation Of Search Results | 91 |
| 3.5.5 | Comparison Of Batched Search Results | 91 |
| 3.6 | Benchmark Results | 96 |

| | | |
|-------|------------------------------------|----|
| 3.6.1 | Database And Index Sizes | 96 |
| 3.6.2 | Search Speed | 96 |
| 3.6.3 | Search Sensitivity | 97 |

II Cooperative Compression Of Redundant Proteomic Databases 109

| | | |
|----------|--|------------|
| 4 | DASH: Search & Alignment For Cooperatively Compressed Databases And Indices | 110 |
| 4.1 | The DASH Algorithm | 116 |
| 4.1.1 | Stage 1: Searching For Non-Gapped Alignments | 117 |
| 4.1.1.1 | Addressing Selectivity | 118 |
| 4.1.1.2 | Stop Words | 123 |
| 4.1.1.3 | Limiting Alignment Numbers In Flight | 124 |
| 4.1.1.4 | Suppression Of Repeated Discovery Of Long Alignments | 125 |
| 4.1.1.5 | Locally Adaptive Query Striding | 125 |
| 4.1.1.6 | Combined Effect Of Alignment Candidate Reduction Measures | 127 |
| 4.1.2 | Stage 2: Optimal Assembly Of HSPs | 129 |
| 4.1.3 | Stage 3: Alignment Finishing Using Adaptive Banded Dynamic Programming | 132 |
| 4.1.4 | On Search Time Complexity | 137 |
| 4.2 | Search Parameters | 138 |
| 4.2.1 | Tunable Parameters | 138 |

| | | |
|----------|---|------------|
| 4.2.1.1 | Tunable Alignment Properties | 138 |
| 4.2.1.2 | Tunable Query Striding Parameters | 139 |
| 4.2.1.3 | Tunable HSP Properties | 139 |
| 4.2.1.4 | Tunable DP And HSP Assembly Parameters | 140 |
| 4.2.1.5 | Canonical Parameter Sets | 142 |
| 4.3 | DASH Search Program (dash) | 144 |
| 4.3.1 | Scoring, Statistics And Output Format | 144 |
| 4.4 | Results | 149 |
| 4.4.1 | Illustrated Example Of Alignment Assembly | 149 |
| 4.4.2 | Example Of Superior Alignment Assembly | 151 |
| 4.5 | Summary | 153 |
| 5 | FOLDDDB: First Steps In Cooperatively Compressed Databases And Indices | 154 |
| 5.1 | FOLDDDB Index Structure And Algorithm | 156 |
| 5.1.1 | FOLDDDB Index Structure | 156 |
| 5.1.1.1 | Text-Partitioned Structure | 156 |
| 5.1.1.2 | Partition Layout | 158 |
| 5.1.1.3 | Compression Of Inverted Lists | 158 |
| 5.1.2 | Excluding Stop k -mers | 160 |
| 5.1.3 | Record Folding As A Prototype Of Cooperatively Compressed In- dexing | 162 |
| 5.1.4 | Construction Of Folded Database Index | 164 |
| 5.2 | Searching Folded Databases With DASH | 168 |

| | | |
|-------|---|-----|
| 5.3 | Method | 168 |
| 5.4 | Results And Discussion | 171 |
| 5.4.1 | Effect Of Sequence Folding | 171 |
| 5.4.2 | Effect Of Query Length On Search Time | 175 |
| 5.5 | Conclusions | 181 |

III Cooperative Compression Of Less Redundant Nucleic Acid Databases 188

6 NP3: Compressing Sorted Nucleic Acid Databases 189

| | | |
|---------|--|-----|
| 6.1 | Design Considerations | 192 |
| 6.1.1 | Compression And Decompression Speed | 193 |
| 6.1.2 | Opaque Block Compression Unsuitable | 193 |
| 6.1.2.1 | The Lack Of Explicit Recurrence Records | 194 |
| 6.1.2.2 | The Boundaries Of Clusters Of Similar Database Records May Not Be Known | 194 |
| 6.1.2.3 | The Requirement For Random Access To Database Records | 195 |
| 6.1.2.4 | The Poor Performance Of General Purpose Compression Algorithms On DNA | 195 |
| 6.1.3 | Existing DNA Compression Schemes Unsuitable | 195 |
| 6.1.4 | DNA Specific LZ77 Compression Suitable | 196 |
| 6.1.4.1 | Explicit Recurrence Records | 196 |
| 6.1.4.2 | Boundaries Of Clusters Need Not Be Known | 196 |

| | | |
|---------|---|-----|
| 6.1.4.3 | Provision Of Fast Random Access To Database Records | 197 |
| 6.1.5 | Encoding Recurrence Records. | 197 |
| 6.2 | The NP3 Algorithm | 199 |
| 6.2.1 | Administrative Information | 199 |
| 6.2.2 | Compression Of Sequence Descriptions | 201 |
| 6.2.3 | Discovery Of Recurrences | 204 |
| 6.2.3.1 | Recurrence Search Algorithms | 204 |
| 6.2.3.2 | Discovery Of Recurrences | 205 |
| 6.2.4 | Generation Of Possible Record Encodings | 207 |
| 6.2.4.1 | Ad Hoc Code Table | 207 |
| 6.2.4.2 | Recently Referenced Address Table | 210 |
| 6.2.4.3 | Selection Of Codes During Compression | 213 |
| 6.2.5 | Computation Of Optimal Code Streams | 213 |
| 6.2.5.1 | Tabulation Of Coding Options | 215 |
| 6.2.5.2 | Calculation Of Optimal Path | 215 |
| 6.2.5.3 | Effect Of Extension Code (Code J) | 218 |
| 6.2.5.4 | Effect Of RRAT | 218 |
| 6.2.5.5 | Computational Cost | 219 |
| 6.2.5.6 | Summary | 219 |
| 6.2.6 | Segmentation Of Long Records | 220 |
| 6.2.7 | Database Partitioning | 221 |
| 6.2.7.1 | Parallel Compression Of NP3 Files | 221 |

| | | |
|----------|---|------------|
| 6.2.7.2 | Ease Of Updating And Appending To NP3 Files | 222 |
| 6.2.8 | Decompression | 222 |
| 6.2.8.1 | Sequential Record Access | 222 |
| 6.2.8.2 | Random Record Access | 222 |
| 6.3 | Results | 223 |
| 6.3.1 | Compression Of The Human UniGene (Nucleic Acid) Database . . | 223 |
| 6.3.2 | Compression Speed | 225 |
| 6.3.3 | Decompression Speed | 225 |
| 6.3.3.1 | Global Random Decompression | 226 |
| 6.3.3.2 | Local Random Decompression | 227 |
| 6.3.3.3 | Sequential Decompression | 227 |
| 6.3.4 | Compression Of De Facto Corpus | 228 |
| 6.4 | Conclusions | 229 |
| 6.5 | Future Directions | 230 |
| 7 | NIX: Producing Compact Cooperatively Compressed Indices Of Biological Sequence Databases | 232 |
| 7.1 | The NIX Indexing Algorithm | 233 |
| 7.1.1 | Omission Of Redundant Postings | 233 |
| 7.1.2 | Reconstruction Of Omitted Index Postings | 237 |
| 7.1.3 | Re-Use And Minimisation Of HSP Discovery Effort | 238 |
| 7.2 | NIX Index Format | 240 |
| 7.2.1 | Why Pointers To k -mer Indices Were Not Compressed | 241 |

| | | |
|---------|--|-----|
| 7.2.2 | Compressing The Inverted Lists | 241 |
| 7.2.2.1 | Creating A Fast Interpolative Coder | 243 |
| 7.3 | Modifications To NP3 | 246 |
| 7.3.1 | Optimisation One: Preferring Inter-Record References | 246 |
| 7.3.2 | Optimisation Two: Per-Posting Rebate | 247 |
| 7.3.3 | Optimisation Three: Maximising Inter-Record Reference Target Coverage | 247 |
| 7.4 | Searching NP3/NIX Ensembles With DASH | 248 |
| 7.5 | Comparison of NP3 and GeNML | 248 |
| 7.6 | Presentation Of Duplicated Results | 249 |
| 7.7 | Method | 251 |
| 7.8 | Results | 252 |
| 7.8.1 | Improved Search Sensitivity | 252 |
| 7.8.2 | Reduced Index Sizes | 262 |
| 7.8.3 | Increased Search Time | 265 |
| 7.8.3.1 | NP3 And NIX Decompression Costs | 265 |
| 7.8.3.2 | Time Spent Performing Dynamic Programming, Discov- ering HSPs, And Translating HSPs | 270 |
| 7.8.4 | Cooperative Compression Of A Less Redundant Database | 272 |
| 7.8.5 | Comparison Of GeNML And NP3 | 273 |
| 7.8.6 | Compressed Database And Index Sizes | 273 |
| 7.8.7 | Effect Of Query Length On Search Time | 275 |
| 7.9 | Discussion | 278 |

| | | |
|-----------|---|------------|
| 7.9.1 | Analysis Of Performance With Disk Based Index | 278 |
| 7.9.1.1 | Analysis Of DASH With FOLDDDB And NP3/NIX | 278 |
| 7.9.1.2 | The Beneficial Effect Of Partitioned Data | 279 |
| 7.9.1.3 | Comparison Of Batched DASH Versus NCBI-BLAST | 280 |
| 7.10 | Conclusions | 280 |
| 7.11 | Future Directions | 282 |
| 7.11.1 | Sorting Databases | 282 |
| 7.11.2 | Improving Search Efficiency | 282 |
| 7.11.3 | Avoiding NP3 Decompression Time | 282 |
| 7.11.4 | Presenting Relationships Among Search Results | 283 |
| 7.11.5 | Improving Compression Performance By Using Dissimilar Regres- sors | 283 |
| IV | Summary Of Results And Conclusions | 298 |
| 8 | Conclusions | 299 |
| 8.1 | Conclusions | 301 |
| V | Appendix | 303 |
| A | Invocation Commands For Search Algorithms | 304 |

List of Figures

| | | |
|------|--|----|
| 2.1 | Example Of Dynamic Programming Evaluation. | 26 |
| 2.2 | Compression of “wooloomooloo” using SEQUITUR. | 55 |
| 2.3 | Example DMC Initial Model. | 58 |
| 2.4 | Example Cloning Of States In A DMC Model. | 58 |
| 3.1 | Calculation Of PatternHunter Variant Metric. | 80 |
| 3.2 | Example Of Incorrect Result From SeqAln. | 82 |
| 3.3 | Example Complete Batching Environment Directory Structure. | 88 |
| 3.4 | Example Batching Environment Description File. | 89 |
| 3.5 | Example Batching Environment Template File. | 89 |
| 3.6 | Use Of pickquery Program To Obtain Standard Nucleic Acid Queries. | 90 |
| 3.7 | Sample Use Of runbatch Program. | 90 |
| 3.8 | Example Of The Terse Alignment Format. | 91 |
| 3.9 | Example Invocation Of runbatch With Custom Filter. | 91 |
| 3.10 | Example Command Sequence To Execute And Summarise The Results Of Several Batches. | 92 |
| 3.11 | Example Batching Environment Directory Structure After Running Batch. | 93 |

| | | |
|------|--|-----|
| 3.12 | Example Batching Environment Data Directory. | 94 |
| 3.13 | Example Command Sequence To Selectively Compare Several Batches. | 96 |
| 3.14 | PatternHunter Variant Scores (See Section 3.3) Of Algorithms For Nucleic Acid Queries (Against The Human UniGene (Nucleic Acid) Database). | 98 |
| 3.15 | PatternHunter Variant Scores (See Section 3.3) Of Algorithms For Nucleic Acid Queries (Against The Human Genome database). | 99 |
| 3.16 | PatternHunter Variant Scores (See Section 3.3) Of Algorithms For Protein Queries (Against The GenPept (Protein) Database). | 100 |
| 4.1 | PatternHunter Variant Scores (See Section 3.3) Of Various Algorithms (Nucleic Acid) (Against The Human UniGene (Nucleic Acid) Database). | 112 |
| 4.2 | PatternHunter Variant Scores (See Section 3.3) Of Various Algorithms (Protein) (Against The GenPept (Protein) Database). | 113 |
| 4.3 | The Three Stages Of The DASH Sequence Alignment Algorithm. | 116 |
| 4.4 | Table Look Up For Un-Gapped Alignment And Score. | 121 |
| 4.5 | Example Of Optimising Striding, $S_{max} = 4$ | 127 |
| 4.6 | Hypothetical Complex HSP Assembly Situation. | 131 |
| 4.7 | Hypothetical Complex HSP Assembly Situation. | 132 |
| 4.8 | Simple Example Of Adaptive Band Placement During Dynamic Programming. | 134 |
| 4.9 | Example Of DASH Adaptive Banded Dynamic Programming. | 135 |
| 4.10 | Alignment Resulting From Figure 4.9 (final score = +11). | 136 |
| 4.11 | Pseudo Code For The DASH Search Algorithm: Overview. | 144 |
| 4.12 | Pseudo Code For The DASH Search Algorithm: HSP discovery. | 145 |
| 4.13 | Pseudo Code For The DASH Search Algorithm: HSP Assembly. | 146 |

| | | |
|------|---|-----|
| 4.14 | Pseudo Code For The DASH Search Algorithm: Dynamic Programming Ends Of Alignments. | 147 |
| 4.15 | Example DASH Output. | 148 |
| 4.16 | Example Of Alignment Between Two Very Similar Sequences. | 150 |
| 4.17 | DASH Alignment of S3317510 Versus S3290308. | 151 |
| 4.18 | BLAST Alignment of S3317510 Versus S3290308. | 152 |
| 5.1 | Fast Ad Hoc Index Posting Compression Algorithm. | 160 |
| 5.2 | Example Of Alignment Unfolding for a Folded Record. | 163 |
| 5.3 | Pseudo Code For Index Construction Process. | 165 |
| 5.4 | Pseudo Code For Index Construction Process. | 167 |
| 5.5 | Pseudo Code For Index Construction Process. | 169 |
| 5.6 | Pseudo Code For Index Construction Process. | 170 |
| 5.7 | PatternHunter Variant Scores (See Section 3.3) For Nucleic Acid Queries (Using The Human UniGene (Nucleic Acid) Database). | 172 |
| 5.8 | PatternHunter Variant Scores (See Section 3.3) For Protein Queries (Using The GenPept (Protein) Database). | 173 |
| 5.9 | Several Alignments From Search Results Due To Cooperative Compression. . | 176 |
| 5.10 | Search Time Versus Query Length For BLAST Searching The UniGene Nu- cleotide Database. | 177 |
| 5.11 | Search Time Versus Query Length For DASH (Mode 2) Searching The Uni- Gene Nucleotide Database. | 178 |
| 5.12 | Search Time Versus Query Length For BLAST Searching The UniGene Pro- tein Database. | 179 |

| | | |
|------|--|-----|
| 5.13 | Search Time Versus Query Length For DASH (Mode 2) Searching The Uni- Gene Protein Database. | 180 |
| 6.1 | NP3 Flow Chart. | 200 |
| 6.2 | Example Of Three Records Each Containing A Common, i.e., Recurrent, Region. | 207 |
| 6.3 | Recently Referenced Address Table (RRAT) Management. | 210 |
| 6.4 | Comparison Of Different RRAT Advancement Strategies. | 212 |
| 6.5 | Coding Options For Three Successive Offsets In An Example Sequence. . . . | 214 |
| 7.1 | Forward And Reverse Indexing Of Chains Of Recurrences. | 236 |
| 7.2 | HSP Translation Scenarios. | 239 |
| 7.3 | Example Interpolative Coding. | 244 |
| 7.4 | Pattern Hunter Variant Scores For Nucleic Acid Queries (Using The Human UniGene (Nucleic Acid) Database). | 263 |
| 7.5 | How Finding Extra HSPs Can Reduce Dynamic Programming Time. | 271 |
| 7.6 | DASH+NP3/NIX Search Time Versus Query Length | 277 |

List of Tables

| | | |
|------|--|----|
| 1.1 | The Multiple Text Alignment Of Eight Translations Of Nehemiah 3:14a. | 5 |
| 1.2 | Repeated Phrases In The Eight translations of Nehemiah 3:14a. | 6 |
| 1.3 | Non-Redundant Index Of Nehemiah 3:14a. | 9 |
| 1.4 | Consensus Region Between NIV And Other Translations Of Nehemiah 3:14a. | 11 |
| 2.1 | IUPAC-IUB Codes (Joint Commission on Biochemical Nomenclature 1983) And Their 4 bit Representations As Used In This Dissertation. | 20 |
| 2.2 | Non-Gapped Alignment Of CGACT And CGTGT. | 23 |
| 2.3 | Gapped Alignment Of CGACT And CGAAGCT. | 24 |
| 2.4 | Evaluated Dynamic Programming Space For CGACT And CGAAGCT. | 29 |
| 2.5 | Example Of Local Sequence Alignment. | 29 |
| 2.6 | Example Of Global Sequence Alignment. | 30 |
| 2.7 | Example Of Proteomic Sequence Alignment Using A Substitution Matrix. . . . | 30 |
| 2.8 | Inferred Relative Speed Of Various Heuristic Sequence Similarity Search Algorithms. | 44 |
| 2.9 | Example Of LZ77 Coding For “banana”. | 52 |
| 2.10 | Example Of LZ78 Coding For “banana”. | 52 |
| 2.11 | Table Of Compression Ratio Results For The Canterbury Corpus. | 59 |

| | | |
|------|--|-----|
| 2.12 | Relative Decompression Times For The Canterbury Corpus. | 60 |
| 2.13 | Burrows-Wheeler Transform Step 1 | 62 |
| 2.14 | Burrows-Wheeler Transform Step 2 | 62 |
| 2.15 | Burrows-Wheeler Transform Step 3 | 63 |
| 2.16 | Burrows-Wheeler Transform Step 4 | 63 |
| 2.17 | Compression Performance Of Various DNA Compression Algorithms. | 67 |
| 3.1 | Per Chromosome And Total Size Statistics Of The April 2003 Draft Of The Human Genome. | 76 |
| 3.2 | List of Statistical Summary Files Produced By Batch Environment. | 95 |
| 3.3 | Human UniGene (Nucleotide) Database And Index Sizes For Surveyed Algorithms. | 101 |
| 3.4 | Human Genome Nucleotide Database And Index Sizes For Surveyed Algorithms. | 102 |
| 3.5 | Protein Database And Index Sizes For Surveyed Algorithms. | 102 |
| 3.6 | Comparison Of Search Speed For Various Algorithms Against The Human UniGene (Nucleic Acid) Database. | 103 |
| 3.7 | Comparison Of Search Speed For Various Algorithms Against The Human Genome Database. | 104 |
| 3.8 | Comparison Of Protein Search Speed For Various Algorithms (Against The GenPept (Protein) Database). | 105 |
| 3.9 | Nucleotide Sensitivity Of Various Algorithms (UniGene Nucleotide Database). | 106 |
| 3.10 | Sensitivity Of Various Algorithms (Human Genome Nucleotide Database) | 107 |
| 3.11 | Sensitivity Of Various Algorithms (GenPept Protein Databases). | 108 |

| | | |
|-----|---|-----|
| 4.1 | PatternHunter Variant Scores: 100% Required Versus 50% Required. | 112 |
| 4.2 | IUPAC-IUB Codes And Their 4-bit Representations. | 122 |
| 4.3 | Differential Scoring Against Wild Card Bases. | 123 |
| 4.4 | Effect Of Index Posting Evaluation Reduction Strategies. | 128 |
| 4.5 | DASH Canonical Parameter Sets For Nucleic Acid Searching: M2. | 142 |
| 4.6 | DASH Canonical Parameter Sets For Nucleic Acid Searching: M4. | 142 |
| 4.7 | DASH Canonical Parameter Sets For Protein Searching: M2. | 143 |
| 4.8 | DASH Canonical Parameter Sets For Protein Searching: M4. | 143 |
| 5.1 | Human UniGene (Nucleic Acid) Database And Index Sizes In Megabytes (MB) And Bits Per Base (B/B). | 182 |
| 5.2 | GenPept Protein Database And Index Sizes In Megabytes (MB) And Bits Per Acid (B/A). | 183 |
| 5.3 | Comparison Of Nucleotide Search Speed (Using The Human UniGene (Nu- cleic Acid) Database). | 184 |
| 5.4 | Comparison Of Protein Search Speed (Using The GenPept (Protein) Database). | 185 |
| 5.5 | Nucleotide Sensitivity Scores (PatternHunter Variant) Versus The Results Of The Smith-Waterman Algorithm (Using The Human UniGene (Nucleic Acid) Database). | 186 |
| 5.6 | Protein Sensitivity Scores (PatternHunter Variant) Versus The Results Of The Smith-Waterman Algorithm (Using The GenPept (Protein) Database). | 187 |
| 6.1 | NP3 Binary Encoding Scheme For Nucleotide Sequence Data. | 209 |
| 6.2 | Codes Corresponding To The Coding Costs In Table 6.3. | 216 |
| 6.3 | Matrix Of Coding Costs (In Bytes). | 216 |

| | | |
|------|---|-----|
| 6.4 | Matrix Of Cumulative Coding Costs. | 217 |
| 6.5 | The Optimum Code For The Record In Figure 6.5. | 218 |
| 6.6 | Example Of Sequence Segmentation Tags In Record Descriptions. | 221 |
| 6.7 | Comparison Of NP3 File Size With Other Formats For Human UniGene (Nucleic Acid) Database. | 224 |
| 6.8 | NP3 Decompression Performance When Using Inter-Record References When Equally Cheap. | 226 |
| 6.9 | NP3 Decompression Performance When Using Inter-Record References Only If Cheaper. | 226 |
| 6.10 | Nucleotide Decompression Speed (No Descriptions) Of GZIP, NP3, And The GeNML Implementation Of Chapter 7. | 228 |
| 7.1 | Sensitivity Results For Each Of The Ten Configurations Of DASH Mode 2 + NP3/NIX, And Posting Frequency Exclusion Threshold = $1.5 \times$ Random Expectation. | 253 |
| 7.2 | Sensitivity Results For Each Of The Ten Configurations Of DASH Mode 2 + NP3/NIX, And Posting Frequency Exclusion Threshold = $2.5 \times$ Random Expectation. | 254 |
| 7.3 | Sensitivity Results For Each Of The Ten Configurations Of DASH Mode 2 + NP3/NIX, And Posting Frequency Exclusion Threshold = $5.0 \times$ Random Expectation. | 255 |
| 7.4 | Sensitivity Results For Each Of The Ten Configurations Of DASH Mode 2 + NP3/NIX, And Posting Frequency Exclusion Threshold = $10 \times$ Random Expectation. | 256 |

| | | |
|------|---|-----|
| 7.5 | Sensitivity Results For Each Of The Ten Configurations Of DASH Mode 4 + NP3/NIX, And Posting Frequency Exclusion Threshold = $1.5 \times$ Random Expectation. | 257 |
| 7.6 | Sensitivity Results For Each Of The Ten Configurations Of DASH Mode 4 + NP3/NIX, And Posting Frequency Exclusion Threshold = $2.5 \times$ Random Expectation. | 258 |
| 7.7 | Sensitivity Results For Each Of The Ten Configurations Of DASH Mode 4 + NP3/NIX, And Posting Frequency Exclusion Threshold = $5.0 \times$ Random Expectation. | 259 |
| 7.8 | Sensitivity Results For Each Of The Ten Configurations Of DASH Mode 4 + NP3/NIX, And Posting Frequency Exclusion Threshold = $10 \times$ Random Expectation. | 260 |
| 7.9 | Nucleic Acid Database And Index Sizes In Megabytes (MB) And Bits Per Base (B/B) (NIX E-value = 1.5). | 266 |
| 7.10 | Nucleic Acid Database And Index Sizes In Megabytes (MB) And Bits Per Base (B/B) (NIX E-value = 2.5). | 267 |
| 7.11 | Nucleic Acid Database And Index Sizes In Megabytes (MB) And Bits Per Base (B/B) (NIX E-value = 5.0). | 268 |
| 7.12 | Nucleic Acid Database And Index Sizes In Megabytes (MB) And Bits Per Base (B/B) (NIX E-value = 10.0). | 269 |
| 7.13 | Relative Size Of Most Compact Index Versus Negative Control. | 270 |
| 7.14 | Comparison Of Nucleic Acid Search Speed (Using The Human UniGene (Nucleic Acid) Database), DASH M2, E=1.5. | 286 |
| 7.15 | Comparison Of Nucleic Acid Search Speed (Using The Human UniGene (Nucleic Acid) Database), DASH M2, E=2.5. | 287 |

| | | |
|------|--|-----|
| 7.16 | Comparison Of Nucleic Acid Search Speed (Using The Human UniGene (Nucleic Acid) Database), DASH M2, E=5. | 288 |
| 7.17 | Comparison Of Nucleic Acid Search Speed (Using The Human UniGene (Nucleic Acid) Database), DASH M2, E=10. | 289 |
| 7.18 | Comparison Of Nucleic Acid Search Speed (Using The Human UniGene (Nucleic Acid) Database), DASH M4, E=1.5. | 290 |
| 7.19 | Comparison Of Nucleic Acid Search Speed (Using The Human UniGene (Nucleic Acid) Database), DASH M4, E=2.5. | 291 |
| 7.20 | Comparison Of Nucleic Acid Search Speed (Using The Human UniGene (Nucleic Acid) Database), DASH M4, E=5. | 292 |
| 7.21 | Comparison Of Nucleic Acid Search Speed (Using The Human UniGene (Nucleic Acid) Database), DASH M4, E=10. | 293 |
| 7.22 | Break Down Of DASH+NP3/NIX Search Time. | 294 |
| 7.23 | Human Genome Nucleic Acid Database And Index Sizes For Surveyed Algorithms. | 295 |
| 7.24 | Nucleotide Decompression Speed (No Descriptions) Of GZIP, NP3, And The GeNML Implementation Of Chapter 7. | 296 |
| 7.25 | Size of DNA Databases Compressed Using GeNML, NP3, NP3(GeNML) And NIX. | 297 |
| A.1 | SSEARCH 3.4t25 (Smith-Waterman) Configuration. | 304 |
| A.2 | BLAT Configuration. | 305 |
| A.3 | NCBI-BLAST 2.2.6 Default Configuration. | 305 |
| A.4 | NCBI-BLAST 2.2.6 No Filtering Configuration. | 305 |
| A.5 | NCBI-BLAST 2.2.6 Report Everything Configuration. | 305 |

| | | |
|------|--|-----|
| A.6 | PatternHunter Configuration. | 306 |
| A.7 | FASTA Configuration. | 306 |
| A.8 | CAFE Configuration. | 306 |
| A.9 | DASH+FOLDDDB M2 Configuration. | 306 |
| A.10 | DASH+FOLDDDB M4 Configuration. | 307 |
| A.11 | DASH + NP3/NIX Configuration 1: No Cooperative Compression (Negative Control). | 307 |
| A.12 | DASH + NP3/NIX Configuration 2: Forward Indexing. | 307 |
| A.13 | DASH + NP3/NIX Configuration 3: Forward Indexing, Prefer Inter-Record References. | 308 |
| A.14 | DASH + NP3/NIX Configuration 4: Forward Indexing, Prefer Inter-Record References, Rebate Estimated Savings Of Omitted Postings. | 308 |
| A.15 | DASH + NP3/NIX Configuration 5: Forward Indexing, Prefer Inter-Record References, Rebate Estimated Savings Of Omitted Postings, Do Not Exclude Stop k -mers. | 308 |
| A.16 | DASH + NP3/NIX Configuration 6: Reverse Indexing. | 308 |
| A.17 | DASH + NP3/NIX Configuration 7: Reverse Indexing, Prefer Inter-Record References. | 309 |
| A.18 | DASH + NP3/NIX Configuration 8: Reverse Indexing, Prefer Inter-Record References, Rebate Estimated Savings Of Omitted Postings. | 309 |
| A.19 | DASH + NP3/NIX Configuration 9: Reverse Indexing, Prefer Inter-Record References, Rebate Estimated Savings Of Omitted Postings, Maximise Distinct Source Material. | 309 |

A.20 DASH + NP3/NIX Configuration 10: Reverse Indexing, Prefer Inter-Record References, Rebate Estimated Savings Of Omitted Postings, Maximise Distinct Source Material, Do Not Exclude Stop k -mers. 310