Title
Applications in High-throughput Sequencing Technologies: From Compression Algorithms and Data Warehousing to Understanding Gene Regulation and Diseases at Scale

Permalink
https://escholarship.org/uc/item/7h88026w

Author
Rigor, Paul

Publication Date
2015

Peer reviewed|Thesis/dissertation
Applications in High-throughput Sequencing Technologies: From Compression Algorithms and Data Warehousing to Understanding Gene Regulation and Diseases at Scale

DISSERTATION

submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in Computer Science

by

Paul Macaraeg Rigor

Dissertation Committee:
Professor Pierre Baldi, Chair
Associate Professor Xiaohui Xie
Professor Paolo Sassone-Corsi

2015
DEDICATION

To my parents, Corazon Rigor and Gil Gutierrez; brothers, James Rigor and Nikko Rigor-Gutierrez; and grandparents, Josefina and Gervacio Rigor, and Cadidia and Sultan Dimatunday Macatanong.
# TABLE OF CONTENTS

**LIST OF FIGURES** vii

**LIST OF TABLES** viii

**ACKNOWLEDGMENTS** ix

**CURRICULUM VITAE** x

**ABSTRACT OF THE DISSERTATION** xiv

1 Data structures and compression algorithms for HTS technologies 1

1.1 Introduction ............................................. 1

1.2 Methods .................................................. 3

1.2.1 General Approach ........................................ 3

1.2.2 Specific Encoding Strategies ............................... 5

1.3 Results and Discussion ..................................... 10

1.3.1 Data Extraction and Statistics ............................ 11

1.3.2 Encoding of Location Information .......................... 13

1.3.3 Encoding of Mismatch Information .......................... 16

1.3.4 Final Encoding ........................................... 17

1.3.5 Implementation ........................................... 17

1.4 Conclusions ............................................... 19

1.5 Figures .................................................. 24

1.5.1 Figure 1.1 - Dataset 2 Nucleotide Substitutions ............ 24

1.6 Tables .................................................. 24

1.6.1 Table 1.1 - Example of Elias Gamma (EG) Coding .......... 24

1.6.2 Table 1.2 - Example of Monotone Value (MOV) Coding ........ 25

1.6.3 Table 1.3 - Statistics of Three High-Throughput Data Sets .... 26

1.6.4 Table 1.4 - Compression Algorithm Results on Three HTS Data Sets 26

1.6.5 Table 1.5 - Comparison of Compression Results ................ 27

1.6.6 Table 1.6 - Comparison of Compression and Decompression Timing ... 28

1.6.7 Table 1.7 - Sequence encoding using Huffman Trees ........... 29
2 MotifMap: Novel method for TFBS prediction

2.1 Introduction .......................................................... 30
2.2 Methods ........................................................................ 33
  2.2.1 Known motifs and motif-matching z-score .................. 33
  2.2.2 Phylogenetic tree and sequence alignments ............... 34
  2.2.3 Bayesian branch length score ................................. 34
  2.2.4 A specific BBLS proposal ........................................ 37
  2.2.5 Estimating the false discovery rate ........................... 39
2.3 Results .......................................................................... 39
  2.3.1 Ranking motifs according to their BBLS ................. 40
  2.3.2 Properties of identified sites ..................................... 43
  2.3.3 Comparison with PhastCons elements ...................... 46
  2.3.4 Motif sites overlapping SNPs .................................... 47
2.4 Web server and interface ................................................ 48
2.5 Discussion ..................................................................... 49

3 MotifMap: Updated Modular Redesign for High-Performance Computing 52

3.1 Introduction .............................................................. 52
3.2 Methods ........................................................................ 54
  3.2.1 Normalized log-odd score (NLOD) ......................... 54
  3.2.2 Bayesian branch length score (BBLS) ....................... 55
  3.2.3 False discovery rate (FDR) ........................................ 57
  3.2.4 Sequence alignments and modular design ................ 57
  3.2.5 Redundancy Filter .................................................. 59
3.3 Results .......................................................................... 59
  3.3.1 New MotifMaps ..................................................... 59
  3.3.2 Evaluation of new methods using experimental data .... 60
  3.3.3 Localization analysis: binding site location properties .. 61
  3.3.4 MotifMap system, web server, and data integration ... 62
3.4 Discussion ..................................................................... 63
  3.4.1 Mouse P53 apoptotic pathway ............................... 65
  3.4.2 Mouse Gli targets .................................................. 66
  3.4.3 Further integration and challenges .......................... 67
3.5 Conclusion ..................................................................... 69
3.6 Figures .......................................................................... 70
  3.6.1 Figure 3.1 - Explanation of methods ....................... 70
  3.6.2 Figure 3.2 - Distribution of distance to closest gene for binding sites .... 71
  3.6.3 Figure 3.3 a & b - Distribution regulatory elements by conservation .... 71
  3.6.4 Figure 3.4 - Known apoptotic targets of P53 ............ 72
  3.6.5 Figure 3.5 a & b - Targets of Gli ............................. 72
3.7 Tables ........................................................................... 73
  3.7.1 Table 3.1 - Multiple alignment information. .............. 73
  3.7.2 Table 3.2 - Non-redundant transcription factor binding sites. 74
  3.7.3 Table 3.3 - Prototype vs Updated MotifMap pipelines. .. 74
4 Application of MotifMap: Sirtuin circadian transcriptome

4.1 Introduction

4.2 Results
  4.2.1 Partitioning of the circadian transcriptome by SIRT6 and SIRT1
  4.2.2 SIRT6 controls their chromatin recruitment
  4.2.3 SIRT6 controls SREBP-1-dependent circadian transcription
  4.2.4 Differential circadian metabolic phenotypes

4.3 Discussion

4.4 Experimental Procedures
  4.4.1 Animal housing and experimental design
  4.4.2 DNA microarray analysis
  4.4.3 Biological pathway analysis
  4.4.4 Gene expression analysis
  4.4.5 Chromatin Immunoprecipitation (ChIP)
  4.4.6 Western blot analysis
  4.4.7 Plasmids and luciferase reporter assays
  4.4.8 Mass spectrometry (MS) for metabolomics analysis
  4.4.9 Statistical analysis and MotifMap

4.5 Figures and Tables
  4.5.1 Figure 4.1
  4.5.2 Figure 4.2
  4.5.3 Figure 4.3
  4.5.4 Figure 4.4
  4.5.5 Figure 4.5
  4.5.6 Figure 4.6
  4.5.7 Figure 4.7
  4.5.8 Supplemental Figure 4.8
  4.5.9 Supplemental Figure 4.9
  4.5.10 Supplemental Figure 4.10
  4.5.11 Supplemental Figure 4.11
  4.5.12 Supplemental Figure 4.12
  4.5.13 Supplemental Figure 4.13
  4.5.14 Supplemental Figure 4.14
  4.5.15 Supplemental Figure 4.15
  4.5.16 Supplemental Table 4.16
  4.5.17 Supplemental Table 4.17
  4.5.18 Supplemental Table 4.18

Bibliography
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Dataset 2 Nucleotide Substitutions</td>
<td>24</td>
</tr>
<tr>
<td>2.1</td>
<td>ROC curves for different methods for predicted TFBS</td>
<td>41</td>
</tr>
<tr>
<td>2.2</td>
<td>Estimated false positive rate as a function of BBLS</td>
<td>45</td>
</tr>
<tr>
<td>2.3</td>
<td>Distribution of the motif sites relative to the TSSs of genes</td>
<td>47</td>
</tr>
<tr>
<td>3.1</td>
<td>Explanation of methods</td>
<td>70</td>
</tr>
<tr>
<td>3.2</td>
<td>Distribution of distance to closest gene for human binding sites</td>
<td>71</td>
</tr>
<tr>
<td>3.3</td>
<td>Distribution regulatory elements by conservation</td>
<td>72</td>
</tr>
<tr>
<td>3.4</td>
<td>Known apoptotic targets of P53</td>
<td>73</td>
</tr>
<tr>
<td>3.5</td>
<td>Targets of Gli</td>
<td>74</td>
</tr>
<tr>
<td>4.1</td>
<td>Main Figure 4.1</td>
<td>94</td>
</tr>
<tr>
<td>4.2</td>
<td>Main Figure 4.2</td>
<td>95</td>
</tr>
<tr>
<td>4.3</td>
<td>Main Figure 4.3</td>
<td>96</td>
</tr>
<tr>
<td>4.4</td>
<td>Main Figure 4.4</td>
<td>97</td>
</tr>
<tr>
<td>4.5</td>
<td>Main Figure 4.5</td>
<td>98</td>
</tr>
<tr>
<td>4.6</td>
<td>Main Figure 4.6</td>
<td>99</td>
</tr>
<tr>
<td>4.7</td>
<td>Main Figure 4.7</td>
<td>100</td>
</tr>
<tr>
<td>4.8</td>
<td>Supplemental Figure 4.8</td>
<td>101</td>
</tr>
<tr>
<td>4.9</td>
<td>Supplemental Figure 4.9</td>
<td>101</td>
</tr>
<tr>
<td>4.10</td>
<td>Supplemental Figure 4.10</td>
<td>102</td>
</tr>
<tr>
<td>4.11</td>
<td>Supplemental Figure 4.11</td>
<td>103</td>
</tr>
<tr>
<td>4.12</td>
<td>Supplemental Figure 4.12</td>
<td>104</td>
</tr>
<tr>
<td>4.13</td>
<td>Supplemental Figure 4.13</td>
<td>105</td>
</tr>
<tr>
<td>4.14</td>
<td>Supplemental Figure 4.14</td>
<td>106</td>
</tr>
<tr>
<td>4.15</td>
<td>Supplemental Figure 4.15</td>
<td>107</td>
</tr>
<tr>
<td>4.16</td>
<td>Supplemental Figure 4.16</td>
<td>108</td>
</tr>
<tr>
<td>4.17</td>
<td>Supplemental Figure 4.17</td>
<td>109</td>
</tr>
<tr>
<td>4.18</td>
<td>Supplemental Figure 4.18</td>
<td>110</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>Example of Elias Gamma (EG) Coding</td>
<td>24</td>
</tr>
<tr>
<td>1.2</td>
<td>Example of Monotone Value (MOV) Coding</td>
<td>25</td>
</tr>
<tr>
<td>1.3</td>
<td>Statistics of Three HTS Data Sets</td>
<td>26</td>
</tr>
<tr>
<td>1.4</td>
<td>Compression Results on Three HTS Data Sets</td>
<td>27</td>
</tr>
<tr>
<td>1.5</td>
<td>Comparison of Compression Results</td>
<td>28</td>
</tr>
<tr>
<td>1.6</td>
<td>Comparison of Compression and Decompression Timing</td>
<td>29</td>
</tr>
<tr>
<td>1.7</td>
<td>Sequence encoding using Huffman Trees</td>
<td>29</td>
</tr>
<tr>
<td>2.1</td>
<td>Comparison of the area under ROC curve</td>
<td>42</td>
</tr>
<tr>
<td>2.2</td>
<td>Top 50 motifs with FDR &lt; 0.1</td>
<td>51</td>
</tr>
<tr>
<td>3.1</td>
<td>Multiple alignment information</td>
<td>75</td>
</tr>
<tr>
<td>3.2</td>
<td>Non-redundant transcription factor binding sites</td>
<td>75</td>
</tr>
<tr>
<td>3.3</td>
<td>Performance comparison of the prototype and updated MotifMap pipelines</td>
<td>75</td>
</tr>
<tr>
<td>3.4</td>
<td>Performance of the mouse MotifMap</td>
<td>76</td>
</tr>
<tr>
<td>3.5</td>
<td>Mouse P53 apoptotic pathway</td>
<td>76</td>
</tr>
</tbody>
</table>
ACKNOWLEDGMENTS

I would like to acknowledge all of our past and present collaborators. Without their data, none of this work would have been possible: Paolo Sassone-Corsi Lab, Harmut Hudel Lab, Susan Sandmeyer Lab, and Dr. Sender and his colleagues at CHOC.

In particular I would like to acknowledge Michael Zeller, Vishal Patel, Kenny Daily, Selma Masri, Scott Christley, Gabe Ozorowski, Christophe Magnan, Ramzi Nasr, Arlo Randall, Matthew Kayala, Pietro di Lena, Ken Nagata, Chloe Azencott, Yu Liu, Nick Ceglia, Peter Sadowski, and last but certainly not least, Joni Carrasco.

I would also like to thank those entities whose funding made this work possible:

3. Lister Hill National Center for Biomedical Communication’s (LHNCBC) for the National Library of Medicine (NLM) Rotation for Medical Informatics Trainees. Bethesda, MD. 2011
4. UCI Ignition Award for High-throughput Virtual Drug Screening Pipeline. 2010

This work has been made possible due to support by a grant from the Hyundai Foundation to Dr. Leonard Sender; NIH (AG043745), Merieux Research Grant (53923) and Sirtris Pharmaceuticals (SP-48984) grants to Paolo Sassone-Corsi; and grants NIH LM01, NIH NLM T15 LM07, and NSF IIS-0513376 to Pierre Baldi.

We acknowledge also the support of the CHOC, the UCI Institute for Genomics and Bioinformatics, and the UCI Genomics High-Throughput Facility. Additional support of our computational infrastructure has been provided by Jordan Hayes, Yuzo Kanomata, Hans Wunsch and the rest of the ICS Helpdesk.

Professors Rick H. Lathrop, Ruslan Afasizhev, and Charless Fowlkes are acknowledged for providing guidance while serving on my advancement committee.

Professor Xiaohui Xie for providing exceptional training for the BIT program coursework as well as guidance with the MotifMap project.

I thank Professor Pierre Baldi for his mentorship and guidance during my graduate career and for his valuable advice on all aspects of my thesis, as well as, sharing his great guitar skills.

In addition, the inclusion of our previous articles discussed in this document was made possible by the generous policies of Cell Press, Oxford University Press, and BioMed Central Ltd.
CURRICULUM VITAE

Paul Macaraeg Rigor

EDUCATION

Doctor of Philosophy in Computer Science 2015
University of California, Irvine Irvine, California

Bachelor of Science in Neuroscience 2004
University of California, Los Angeles Los Angeles, California

Bachelor of Science in Cognitive Science 2004
University of California, Los Angeles Los Angeles, California

RESEARCH EXPERIENCE

Graduate Research Assistant 2008–2014
University of California, Irvine Irvine, California

Visiting Pre-doctoral Fellow (NIH/NLM) 2011–2011
Lister Hill National Center for Biomedical Communications (NLM) Bethesda, Maryland

TEACHING EXPERIENCE

Adjunct Faculty 2014
Chapman University Orange, California

Teaching Assistant 2013
University of California, Irvine Irvine, California

AWARDS

Bioinformatics Training Grant 2008-2012
NIH Grant 5T15LM007743

Travel Award 2011
Web Publishing of Scientific Data and Services Training Course

Virtual Drug Discovery Pipeline 2010
UCI Ignition Award

Best Paper: MSR “Scale” Challenge 2007
International Workshop on Mining Software Repositories
WORK EXPERIENCE

Consultant 2015
DataOmics
Los Angeles, California

Data Scientist 2014–2015
OspreyData
San Juan Capistrano, California

Software Developer, Contractor 2014
Envoy
Irvine, California

University of California, Irvine
Irvine, California

Software Engineer 2005–2006
University of California, Irvine
Irvine, California

Bioinformatics Systems Administrator 2004–2005
University of California, Los Angeles
Los Angeles, California
LEADERSHIP

Founder, President, and Advisor
UCI OpenJam Music Community

REFEREED JOURNAL PUBLICATIONS

Partitioning Circadian Transcription by SIRT6 Leads to Segregated Control of Cellular Metabolism

A Genomic Analysis Pipeline and Its Application to Pediatric Cancers
M. Zeller, C. N. Magnan, V. R. Patel, P. Rigor, L. Sender, and P. Baldi. IEEE TCBB Accepted

Retrotransposon profiling of RNA polymerase III initiation sites
Qi, Xiaojie; Daily, Kenneth; Nguyen, Kim; Wang, Haoyi; Mayhew, David; Rigor, Paul; Forouzan, Sholeh; Johnston, Mark; Mitra, Rob David; Baldi, Pierre and Sandmeyer, Suzanne. Genome Research

MotifMap: integrative genome-wide maps of regulatory motif sites for model species

Data structures and compression algorithms for high-throughput sequencing technologies
Kenny Daily, Paul Rigor, Scott Christley, Xiaohui Xie and Pierre Baldi. BMC Bioinformatics October 14, 2010

MotifMap: a human genome-wide map of candidate regulatory motif sites.

Sourcerer: Mining and Searching Internet-Scale Software Repositories. Data Mining and Knowledge Discovery
E. Linstead, S. Bajracharya, T. Ngo, P. Rigor, C. Lopes, P. Baldi. Data Mining and Knowledge Discovery

Mining Internet-Scale Software Repositories
Erik Linstead, Paul Rigor, Sushil Bajracharya, Cristina Lopes, Pierre Baldi. NIPS 2007 December 2013

Mining Eclipse Developer Contributions via Author-Topic Models 2007

CONFERENCE POSTERS/PUBLICATIONS

DockingDB: A cyber-infrastructure for computer-aided drug discovery and design based on ChemDB 2011
Paul Rigor and Pierre Baldi. ACS Conference, CINF Scholarship Track March 2011

MotifMap: Genome-Wide Map of Regulatory Binding Sites 2011
Kenneth M. Daily, Paul Rigor, Vishal Patel, Xiaohui Xie, Pierre Baldi. HTS to P4 Medicine Conference, UCI February 2011

Data Structures and Compression Algorithms for High-Throughput Sequencing Technologies 2009
Kenneth M. Daily, Paul Rigor, Scott Christley, Xiaohui Xie, Pierre Baldi. ISMB/ECCB July 2009


SOFTWARE

MotifMap http://motifmap.ics.uci.edu/
Comprehensive, multi-species genome-wide map of regulatory binding sites.

Sourcerer http://sourcerer.ics.uci.edu/
A Search Engine for Open Source Code.

ChemDB http://cdb.ics.uci.edu/
A Search Engine for Chemicals.
ABSTRACT OF THE DISSERTATION

Applications in High-throughput Sequencing Technologies: From Compression Algorithms and Data Warehousing to Understanding Gene Regulation and Diseases at Scale

By

Paul Macaraeg Rigor

Doctor of Philosophy in Computer Science

University of California, Irvine, 2015

Professor Pierre Baldi, Chair

High-throughput sequencing (HTS) technologies play important roles in the life sciences by allowing the rapid parallel sequencing of very large numbers of relatively short nucleotide sequences, in applications ranging from genome sequencing and resequencing to digital microarrays and ChIP-Seq experiments. As experiments scale up, HTS technologies create new bioinformatics challenges for the storage, distribution, and downstream analyses of HTS data. In this thesis, we address the challenges and opportunities in the storage and downstream analyses of data derived from HTS technologies.

To address the growing amount of HTS data, we develop data structures and compression algorithms for HTS data storage. A processing stage maps short sequences to a reference genome or a large table of sequences. Then the integers representing the short sequence absolute or relative addresses, their length, and the substitutions they may contain are compressed and stored using various entropy coding algorithms, including both old and new fixed codes (e.g. Golomb, Elias Gamma, MOV) and variable codes (e.g. Huffman). The general methodology is illustrated and applied to several HTS data sets. Results show that the information contained in HTS files can be compressed by a factor of 10 or more, depending on the statistical properties of the data sets and various other choices and constraints. Our
algorithms fair well against general purpose compression programs such as gzip, bzip2 and 7zip; timing results show that our algorithms are consistently faster than the best general purpose compression programs.

Achieving a comprehensive map of all the regulatory elements encoded in the human genome is a fundamental challenge of biomedical research. So far, only a small fraction of the regulatory elements have been characterized, and there is great interest in applying computational techniques to systematically discover these elements. Such efforts, however, have been significantly hindered by the overwhelming size of non-coding DNA regions and the statistical variability and complex spatial organizations of mammalian regulatory elements. The MotifMap system uses databases of transcription factor binding motifs, refined genome alignments, and a comparative genomic statistical approach to provide comprehensive maps of candidate regulatory elements encoded in the genomes of model species.

MotifMap and its integration with other data provide a foundation for analyzing gene regulation on a genome-wide scale, and for automatically generating regulatory pathways and hypotheses. The power of this approach is demonstrated and discussed using the P53 apoptotic pathway and the Gli hedgehog pathways as examples.

Further application of MotifMap is underlined by its integration into CircadiOomics (developed in the Baldi lab), which aims to decode transcriptional machinery that are under circadian control. Here we utilize MotifMap in understanding and delineating the roles of a subset of the sirtuin family of deacetylases in regulating circadian rhythms. Circadian rhythms are intimately linked to cellular metabolism. Specifically, the NAD+-dependent deacetylase SIRT1, the founding member of the sirtuin family, contributes to clock function. Whereas SIRT1 exhibits diversity in deacetylation targets and subcellular localization, SIRT6 is the only constitutively chromatin-associated sirtuin and is prominently present at transcriptionally active genomic loci. Comparison of the hepatic circadian transcriptomes reveals that SIRT6 and SIRT1 separately control transcriptional specificity and therefore de-
fine distinctly partitioned classes of circadian genes. SIRT6 interacts with CLOCK:BMAL1 and, differently from SIRT1, governs their chromatin recruitment to circadian gene promoters. Moreover, SIRT6 controls circadian chromatin recruitment of SREBP-1, resulting in the cyclic regulation of genes implicated in fatty acid and cholesterol metabolism. This mechanism parallels a phenotypic disruption in fatty acid metabolism in SIRT6 null mice as revealed by circadian metabolome analyses. Thus, genomic partitioning by two independent sirtuins contributes to differential control of circadian metabolism.
Chapter 1

Data structures and compression algorithms for HTS technologies

1.1 Introduction

Over the past four decades, sequencing technologies have been one of the major driving forces in the life sciences producing, for instance, the full genome sequences of thousands of viruses and bacteria, and dozens of eukaryotic organisms, from yeast to man [76]. This trend is being accentuated by modern high-throughput sequencing (HTS) technologies: several human genomes were recently produced [109, 183, 19, 179] and a project to sequence 1,000 human genomes in the next few years is under way [86]. Different HTS technologies are competing to be able to sequence an individual human genome for less than $1,000 within a few years [164] and reaching the point where human genome sequencing will be a commodity. Furthermore, not only are HTS technologies useful for sequencing and resequencing genomes, but they are also instrumental to accurately identify and measure mRNA and other nucleotide sequences in new important high-throughput applications such as digital expression microar-
rays, ChIP-Seq [122] and SNP genotyping. In all cases, the amount of data produced by HTS technologies as experiments scale up creates significant bioinformatics challenges to understand, store and share data. To address some of these challenges, we develop data structures and compression algorithms for the efficient management and storage of HTS data.

Several different HTS technologies have been conceived and developed to differing degrees of maturity. They can be classified into four broad classes: amplification followed by mass spectrometry, *in vitro* cloning, *in vivo* cloning, and single molecule [62]. There are currently three commercially advanced HTS systems: SOLiD (Applied Biosystems), Solexa (Illumina), 454 (Roche), all based on the *in vitro* cloning approach. Each system depends on a sheared DNA sample which is diluted onto some type of matrix, clonally amplified, and then transformed via repetitive enzymatic cycles into a series of four distinct fluorescent signals (each representing a different base) monitored at each cycle by a CCD camera.

The series of fluorescent signals at each position is converted into a DNA sequence and a quality score for each position. A typical run (or lane) can generate tens of millions of sequence reads, and with a set of experiments that includes biological replicates, control and treatment samples, etc., then the total number of reads can reach into the billions. Important variations between technologies exist, for instance in terms of the length and quality of the sequences. However, all existing systems rely on the parallel sequencing of many short sequences and produce outputs of very long lists of relatively short sequences. Thus the fundamental problem we wish to address is the storage and compression of such lists.

Because of the variations that may exist between different technologies and different constraints associated with different deployment scenarios, our goal is not to provide a single solution, but to describe general methods by which customized solutions can be developed. Thus after presenting the basic idea, we review several relevant representations and compression algorithms. The approach is illustrated on several HTS data sets.
1.2 Methods

1.2.1 General Approach

In the standard text format, a file of \( N \) short sequences of average length \( l \) requires \( N(l + 1) \) bytes (or \( 8N(l + 1) \) bits) to store, using one ASCII byte per character, and including a character (carriage return) to separate two consecutive sequences. The ranges of \( N \) and \( l \) can vary depending on the experiment and the technology, but to fix the ideas one can imagine current values of \( N \) in the \( 10^5 - 10^9 \) range and of \( l \) in the \( 10^1 - 10^3 \) range, with most typical values in the 20-100 range. Additional information regarding, for instance, the quality of the sequences can be included in the output files.

To store and compress this information, we imagine first that the short sequences can be mapped to a reference genome. This is the typical situation for resequencing experiments, including large-scale sequencing of diploid human genomes using mapping software such as Illumina’s ELAND, MAQ [111], ZOOM [116], or Bowtie [107]. In this case, each short sequence \( s_i \) can be represented by its address \( a_i \) in the reference genome. If the length of each sequence is not fixed and known in advance or stored in the header of the file, the length of the sequence \( l_i \) must also be included. If the match is not exact, variations from the genomic sequence must also be included by recording their address and type. For simplicity, we will assume only substitutions. Thus something like \“(1500, 25, 3C)\)” could be used to record a short sequence whose starting point matches position 1500 in the reference genome, with a length of 25 nucleotides, and a substitution by a C in position 3. Relative addresses, rather than absolute addresses, can be used not only to record variations within sequences, but also for the address of the sequences themselves. The same sequence could be encoded by \“(100,25,3C)\)” to indicate that it is found 100 nucleotide downstream of the previously occurring sequence, provided the file has been preprocessed to reorder the sequences linearly along the genome. With relative addresses, the dynamic range of...
the integers to be encoded may be considerably smaller than with absolute addresses. If a sequence can be mapped equally well to multiple locations on the genome, any one of them can be chosen to represent the sequence. Finally, specific experiments could come with additional information. For instance, in SNP mapping experiments, the locations and types of variations could be constrained and leveraged to increase compression. It is worth noting that for this approach, the availability of a reference genome is not as restrictive as it may seem; minimally, a reference of DNA sequences to which experimental sequences can be reasonably mapped is needed. For simplicity in this work, we focus on the case of interest where a reference genome is available.

The idea of compressing DNA sequences is not new. Compression algorithms such as Biocompress-2 [58], CTW [126], OffLine [9], DNACompress [22] and DNA compressor [121] consider the task of directly compressing large sequence strings. While others consider alternative compression tasks such as COIL [184] which compresses a large database of unrelated sequences or DNAzip [23] which compresses variations to a reference genome. Our goal is different as we want to compress a large number of very short sequences while using a large reference sequence. Whether or not it is advantageous over the standard text format, however, depends on the details of the implementation and, more often than not, the data being compressed. A successful implementation of the basic idea depends crucially on careful consideration of the encoding scheme. In particular, the choice of the function converting integers to binary strings, has a great effect on the resulting compression. For our application, these integers are the absolute or relative addresses and lengths of sequence reads. It is essential to understand that simply converting integers to their binary value (e.g. converting “25” to “11001”) does not work since one does not know where one integer ends and the next starts. No symbol other than 0 or 1 is available to separate consecutive integers. Furthermore, such a simple encoding scheme does not take into account any entropy considerations. Likewise, a general purpose compression scheme for text data, such as Lempel-Ziv (gzip [1]), is likely to be far from optimal for HTS data. Thus we are interested in binary
encoding schemes for sequences of integers that can be parsed automatically and, consistent with information theory, are entropy efficient, in the sense that fewer bits are used to encode more frequent events.

A simple back-of-the-envelope calculation, however, can show why the situation is hopeful. Suppose the information associated with the integer $j$ representing an address can be stored in about $2 \log j$ bits (all logarithms are taken to the base 2). This corresponds to a penalty factor of two over the plain binary encoding and can be achieved with the coding methods described in the next section. Then the equality $8(l + 1) = 2 \log j$ shows that some degree of compression is achieved as long $j$ is less than $2^{4(l+1)}$. Even with $l$ as small as 20, this yields $2^{84}$ which is much larger than the length of any genome. Assuming that the length of each sequence must also be stored, this may require at most a fixed number of bits $C$. If the lengths are between 20 and 36, for instance, they can be described with 4 bits. From the relation $8(l + 1) = 4 + 2 \log j$ we find again that compression is achievable as long as $j$ is less than $2^{4l+2}$, which is again easily achieved in the current environment. Furthermore, with for instance $l = 24$ these relations show that a 20-fold or so compression rate should be achievable with reasonable values of $j$. A similar calculation can be made including information about the number of substitutions, their locations, and types for each sequence.

### 1.2.2 Specific Encoding Strategies

To begin with, we illustrate these issues here by considering how the integer addresses $a_i$, relative or absolute, can be encoded into a binary string. From Shannon’s entropy coding theory [128, 25], optimal encoding of these integers from a compression standpoint depends on their distribution in order to assign shorter binary codes to more probable symbols (integers). For simplicity, we distinguish two broad classes of codes: fixed codes, such as Golomb [56] and Elias codes [41] and their more recent variants [13], and variable codes, such as Huff-
man codes [72]. In a fixed code, the integer $i$ is always encoded in the same way, whereas in a variable code the encoding changes.

**Fixed Codes: Golomb and Golomb-Rice Codes**

Both Golomb codes and Elias codes encode an integer $j$ by catenating two bit strings: a preamble $p(j)$, that encodes $j$’s scale, and a mantissa. Golomb codes were specifically developed to encode stationary coin flips with $p \neq 0.5$. Thus they are known to be optimal and asymptotically approach the Shannon limit if the data is generated by random coin flips or, equivalently, if the distribution over the integers is geometric, although they can be used for any other distribution. The more skewed the probability $p$ is (towards 0 or 1) the greater the level of compression that can be achieved.

Golomb codes have one integer parameter $m$. Given $m$, any positive integer $j$ can be written using its quotient and remainder modulo $m$ as $j = \lfloor j/m \rfloor + (j \mod m)$. To encode $j$, the Golomb code with parameter $m$ encodes the quotient and remainder by using:

- $\lfloor j/m \rfloor$ 1-bits for the quotient;
- followed by a 0, as a delimiter (unary encoding of $\lfloor j/m \rfloor$);
- followed by the phased-in binary code for $j \mod m$ for the remainder (described below).

The encoding of integers $0, \ldots, m - 1$ normally requires $B = \lceil \log m \rceil$ bits. If $m$ is not a power of two, then one can sometimes use $B - 1$ bits. More specifically, in the “phased-in” approach:

- if $i < 2^B - m$, then encode $i$ in binary, using $(B - 1)$ bits;
- if $i \geq 2^B - m$, then encode $i$ by $i + 2^B - m$ in binary, using $B$ bits.
For instance, for $m = 5$, $i = 2$ is encoded as “10” using 2 ($= B - 1$) bits, and $i = 4$ is encode as “111” using 3 ($= B$) bits. Thus the encoding of $j$ requires in total $\lceil j/m \rceil + 1 + \lfloor \log m \rfloor$ or $\lceil j/m \rceil + 1 + \lceil \log m \rceil$ bits and the codeword for the integer $j + m$ has one more bit than the codeword for the integer $j$. Unless otherwise specified, all logarithms are taken to base 2. We use also “$\lfloor \log m \rfloor$” to denote “$\lfloor \log m \rfloor$ or $\lceil \log m \rceil$”.

Finally, Golomb-Rice codes are a particularly convenient sub-family of Golomb codes, when $m = 2^k$. To encode $j$, we concatenate $\lfloor j/2^k \rfloor$ 1-bits, one 0-bit, and the $k$ least significant bits of $j$. The length of the encoding of $j$ is thus $\lfloor j/2^k \rfloor + k + 1$. The decoding of Golomb-Rice codes is particularly simple. First, read and count the number of 1 bits until the first 0 bit is found. The number of 1 bits is the quotient $q = \lfloor j/m \rfloor$. Then, read the next $\log m$ bits to get the binary representation of the remainder $r = j \mod m$. The decoded value equals $j = (q \times m) + r$.

**Elias Codes**

In the Elias Gamma coding scheme, the preamble $p(m)$ is a string of zeroes of length $\lfloor \log j \rfloor$, and the mantissa $m(j)$ is the binary encoding of $j$. More precisely, to encode the scale and value of $j$:

- write $\lfloor \log j \rfloor$ 0-bits;
- followed by the binary value of $j$ beginning with its most significant 1-bit.

The length of the encoding of $j$ is $2 \lfloor \log j \rfloor + 1$ (Table 1.1). The decoding is obvious: first read $n$ 0-bits until the first 1-bit is encountered, then read $n$ more bits to get the binary representation of $j$. 


Applying the relationship

\[- \log P(j) \approx 2[\log j] + 1 \quad (1.1)\]

to the integer probabilities, shows that Elias Gamma encoding asymptotically approaches the Shannon limit for \( P(j) \approx Cj^{-2} \). This is a power law relationship with exponent -2 and \( C \) is a normalizing constant. Note that for both Golomb and Elias Gamma codes, several different consecutive integers can be encoded into a bit vector with the same length, hence the relationships \(- \log P(j) \approx \text{length}(j)\) is only approximate with respect to geometric or power-law distributions over the integers.

**Monotone Value Coding (MOV Coding)**

More recently, new families of efficient fixed codes for integers have been developed \([133, 132, 13, 66]\), for instance in the case of increasing or quasi increasing sequences of integers, by encoding only the deltas of the preambles. Here we introduce a modification of the codes described above, presented with the Elias Gamma codes, for messages consisting of monotone sequences of integers, such as consecutive absolute addresses of sequence reads. When the value of the integers being encoded increases monotonically, additional lossless compression can be obtained by encoding only the scale increases and their location (Table 1.2).

More precisely, if a sequence of increasing addresses is given by \((j_1, j_2, \ldots, j_K)\) with \(j_1 < j_2 \ldots < j_K\):

- encode \(j_1\) using Elias Gamma encoding;

- for \(i = 2, \ldots, K\):
  
  - write \([\log j_i] - [\log j_{i-1}]\) 0-bits;
followed by the binary value of $j_i$ beginning with its most significant 1-bit.

The MOV-encoded vector of addresses can be decoded by a simple algorithm:

- set $k = 1$;
- decode each integer in succession by repeating the following steps:
  - increment $k$ by the number of 0-bits in the input stream before reaching the first 1-bit;
  - counting this first 1-bit as the first digit of the integer, read the remaining $k - 1$ bits of the integer from the input stream.

Another variation called Monotone Length Coding (MOL Coding) can be used for quasi-monotone sequence tolerating occasional deviations from a monotone behavior [13]. Another scheme that may be useful for encoding integers but cannot be described for conciseness reasons is the Binary Interpolation [133] scheme, together with several variants.

**Variable Codes**

In genomic applications, in general the integers may not have a well defined distribution, in which case it is always possible to use a general entropy encoding scheme, such as Huffman coding [72, 128, 25] which essentially builds a prefix code by using a binary hierarchical clustering algorithm starting from the events (integers) with the lowest probability. While Huffman coding achieves compression close to the entropy limit, the price to pay over fixed coding schemes such as Golomb and Elias Gamma, or the more recent codes mentioned above, is the storage of the Huffman table which can be quite large in some applications. However this is a fixed cost with respect to the database size, and therefore whether this cost is acceptable or not depends on the specific application. Small gains in compression over Huffman coding may be obtained using arithmetic coding [151, 188], but at a non-trivial
price in the complexity of computations. For more information about integer encodings, refer to references [89] and [187].

**Byte Arithmetic**

Direct implementations of the decoding algorithms process the compressed representations bit-by-bit; however, it is possible to implement faster decoders, which decode the compressed data byte-by-byte. These faster decoders work by looking up information from pre-computed tables. These tables are indexed by: (1) all possible bytes $B$ (ranging from 0 to 255); and (2) a bit-index $i$ (ranging from 0 to 7) which marks the position of the decoder within the byte. These tables may store quantities such as the binary value of byte $B$ starting from bit $i$, the number of bits turned on in byte $B$ starting from bit $i$, and the unary value of byte $B$ starting from bit $i$. The exact quantities stored depend on the details of a particular decoder implementation.

In practice, byte arithmetic considerably increases decoding speed, sometimes approaching as much as an eight-fold improvement over the corresponding bit-by-bit implementation. The exact value of the speedup depends on several factors including the characteristic of the data, the exact compression scheme, and the hardware used.

### 1.3 Results and Discussion

For conciseness, we present a subset of representative results focusing on Elias, MOV and Variable codes. Golomb code results are in general comparable to Elias Gamma results, typically with a slight decrease in performance. Furthermore, Golomb’s code require tuning one additional parameter ($m$) and thus are not reported here.
For each data set, we transformed the data into a uniform flat file format, separating the location information from the mismatch information for each read, then performed encoding on the location and mismatch information separately. The following sections describe the different encoding strategies used for location and mismatch data and their corresponding results.

### 1.3.1 Data Extraction and Statistics

We selected three data sets representative of typical short read sequence data derived from different experimental settings aimed at addressing different biological questions, from genome sequencing to transcription factor binding site mapping. Each of the data sets correspond to a different combination of genome coverage, repetitiveness and locational specificity in the genome, so that our encoding results provide insights into how different strategies can be applied and tailored to different data. Table 1.3 gives some basic statistics for each data set, including the sizes for the original standard text format for the sequence reads, the uniform flat file format as described above, and the Bowtie alignment output. The uniform flat file format sizes are further split into the sizes of the location data and mismatch data. The Bowtie alignment output contains additional information beyond the minimal location and mismatch data required to reconstitute the reads, so Table 1.3 provides compression sizes of that additional information with a set generic compression tools. Those sizes will be used later for comparison of compression results for Bowtie output.

**Dataset 1**

The first data set is obtained from the laboratory of Dr. Suzanne Sandmeyer at University of California Irvine and comes from an experiment aimed at mapping retrotransposon Ty3 insertion sites in the yeast genome. It consists of 6,439,584 sequence reads, all of length 19
bp. By the nature of the underlying experiments, the sequences in this data set are highly clustered, often with a high degree of repetition. The reads have at most two substitutions. The numbers of sequences with 0, 1, and 2 substitutions are given by 3,468,077 (54%), 895,997 (14%), and 2,075,510 (32%) respectively.

Dataset 2

The second data set comes from a chromatin immunoprecipitation assay (ChIP-Seq) used to map the in vivo binding site locations of the neuron-restrictive silencer factor (NRSF) in humans [83]. It consists of 1,697,990 sequence reads, all of length 25 bp and mapped to the most recent human genome sequence (hg18). The reads have at most two substitutions. The numbers of sequences with 0, 1, and 2 substitutions are given by 1,297,153 (76%), 302,939 (18%), and 97,899 (6%) respectively. Figure 1 shows the number of mismatches found at each position along the read, as well as the types of substitutions. The number of mismatches increases towards the end of the read, as expected with the Solexa sequencing technology where the error rate increases further along the length of the read. Our interest is encoding the read sequence as is without attempting to differentiate between true SNPs and sequencing errors, but the distribution clearly shows that the majority of these mismatches are observed at the end of the read which can be used to advantage when encoding the variations.

Dataset 3

The third data set corresponds to a full diploid human genome sequencing experiment for an Asian individual [179]. This is a very large data set with enough reads to provide 36-fold average coverage, and we utilized the existing mapping of the reads provided by the YH database [110] to the human reference genome. For illustrative purposes, we report only
the results corresponding to the reads associated with chromosome 22. For chromosome 22, there are 31,118,531 reads that vary in length from 30 to 40 bp for a total of 1,108,701,700 bp of sequence data. The numbers of sequences with 0, 1, and 2 substitutions are given by 19,126,772 (61%), 6,166,549 (20%), and 5,825,210 (19%) respectively.

1.3.2 Encoding of Location Information

The location information for a mapped read consists of a chromosome identifier, a position along that chromosome, the strand, the length of the read, and the number of mismatches it contains. In the flat file format, each read is specified on a single line with the values separated by a comma. One technique is to encode each of the attributes individually. For this standalone technique, we compute the frequency of occurrence of each of the attributes, order them, and then use EG encoding on their ordered index. An alternative method is to combine all of the attributes together. The \((C,S,M)\) lookup method takes the attributes combined together as tuples, for example \((\text{chromosome}, \text{strand}, \text{number of mismatch})\), then computes the frequency of a subset of these. The combination method of REG Indexed is described in detail below.

Table 1.4 gives the comprehensive set of compression algorithm results. For the data sets where all of the reads have the same length (1 and 2), we omit the length of the read and assume it is specified in a header structure for the data. The top part of Table 4 shows the standalone and combined techniques for encoding the location information, while the bottom part of Table 1.4 is for encoding the mismatch information described in the next section. For the standalone methods, multiple encoding techniques are tried for the start location while only EG encoding is used on the other attributes. The best standalone compression attainable is also shown; this should be compared to the best compression attainable by the either of the combined methods to determine which encoding method to use for encoding.
the start location. This in combination with the mismatch encoding gives the best total compression for the dataset. Throughout the table, the best compression for a specific technique is shown in italics.

All of the results are based upon theoretical calculations without doing the actual encoding. In the Implementation section, we describe our GenCompress software package which implements some of the encoding methods. The methods implemented in GenCompress are marked with a † in Table 4, and the best compression achieved by GenCompress is also shown. In the following, we describe in more detail the various methods used for encoding the location information.

1. Elias Gamma (EG) Absolute: We assume that the reads cannot be reordered in any way and thus must be processed exactly as specified. The chromosome, absolute start coordinate, strand integer values, and for Dataset 3 the read length were encoded using Elias Gamma codes.

2. Elias Gamma (REG) Relative: We assume that the reads can be ordered in any way in order to achieve better compression results. We group all of the reads for each chromosome together. Within each chromosome, the reads are sorted by increasing position number; therefore the relative distance between adjacent reads is encoded rather than their absolute positions. These relative addresses correspond in general to significantly smaller integer values than the absolute addresses, especially for long chromosomes or reads with high-coverage. The chromosome, strand integer values, number of mismatches, and the read length (only for Dataset 3), were encoded using Elias Gamma codes.

3. Relative Elias Gamma Indexed (REG Indexed): We again assume that the reads can be ordered in any way in order to achieve better compression results. We group all of the reads for each chromosome together, then group reads for each strand together within
a chromosome, and further group them for the number of mismatches they contain. Within each bin of (chromosome, strand, number of mismatches), we then encode the relative distances as stated above. Because the reads are grouped by chromosome, strand, and number of mismatches, there is no need to encode that information for each read. Instead, those values are stored along with a count of the number of reads for that group in a header structure. Thus, except for the additional read length information for Dataset 3, only the relative distances between the reads are encoded using Elias Gamma.

4. Monotone Value (MOV): Like the EG Relative encoding above, we reorder the reads for Monotone Value encoding according to chromosome and position. However, we use MOV codes for the absolute locations as the positions are now in increasing order.

5. Huffman: We can use the start positions (or relative starts, denoted RHuffman) to compute a Huffman tree which we use to encode. The resulting size encoding with this method also includes storing the Huffman tree, which is needed for decoding.

Table 1.4 shows that significant compression of the location information is achieved for all three data sets. The REG Indexed encoding was best for all data sets. Slightly better compression can be obtained on Dataset 1 by only using unique positions in the chromosome, and encoding the count of the number of occurrences, (unique relative Elias Gamma encoding), which might be expected from the highly clustered and repetitive nature of the data (data not shown). While the Relative Huffman coding (RHuffman) performs the best compression on the actual start position integers, the other columns of the location must also be stored, making this method worse than the REG Indexed method. On Dataset 1, the size of the alphabet being encoded (the possible relative start distances) is small, due to the extreme sparseness of the data. Huffman coding is known to be less efficient on this type of data. As the size of the alphabet increases, Huffman coding performs better relative to the run length encoding methods (Dataset 2 has the largest alphabet to use for building the Huff-
man tree, and has the largest gain in performance relative to the other methods). The high compression of almost 25 fold of data for the human genome sequencing project (Dataset 3) is very encouraging as these are some of the largest data sets being generated. This data set corresponds to much of the data being generated by the 1000 genomes project. The ChIP-Seq data (Dataset 2) has the lowest compression fold of the three data sets, yet even so Table 1.4 shows that the total encoding of the data achieves over an order of magnitude reduction in size, significantly better than gzip.

### 1.3.3 Encoding of Mismatch Information

The mismatch information for a mapped read consists of the positions of one or two mismatches located somewhere along the read and the nucleotide value (A, C, G, T) for the mismatch. The flat file format for mismatch information has one read per line; the line is blank if no mismatches otherwise it contains a comma separated list of mismatches (e.g. 19A,22C for two mismatches).

The sequence mismatches can be encoded in multiple ways. One possibility is to encode the position of the mismatch directly from the start of the read. However, the mismatches have a clear tendency to occur towards the end of the read as illustrated in Figure 1, so measuring the mismatch from the end of the read can reduce the number of bits to encode the position (i.e. a mismatch at position 24 of a read of length 25 would be measured as 1). For both strategies, the position is encoded using Elias Gamma codes.

The nucleotide substituted at a particular position must also be encoded; the straightforward naïve approach is to map them to integers (A to 1, C to 2, etc.). This can be optimized by ordering the nucleotides by their frequency of occurrence, so the most frequent substitution maps to the lowest integer. In both cases, the value is encoded using Elias Gamma codes.
We also tried another strategy where the position and nucleotide substitution are combined together into a single value. The combination are ordered by frequency and then encoded using Elias Gamma codes. This technique may possibly be effective if there is a large number of duplicate mismatches across all the reads.

We investigated all of these strategies and the results can be seen in Table 1.4. For each data set, a different method ended up having the best compression.

1.3.4 Final Encoding

With the location and mismatch information combined together, we have a final encoding for a representative set of short read sequence data produced by HTS technologies. Using encoding techniques that consider the inherent structure of sequence data consistently perform well for all the data sets we tested. For our test data sets, it is interesting to note the size ratios for encoding the location information versus the mismatches. For data set 1, the mismatches clearly dominate the total compression size while the location information is very small due to the clustered nature of the reads on the genome. On the other hand, data set 2 has much fewer mismatches and the read locations are sparsely distributed across the genome, so the location information dominates the total compression size. Data set 3 is balanced between the two because it has full coverage of the genome. For data sets 1 and 3 which have a large number of mismatches, using dbSNP data as reference variation data may offer the opportunity for further compression.

1.3.5 Implementation

We have implemented a subset of the encoding techniques describe in this article into a software package called GenCompress. The implementation is primarily based on the Rel-
ative Elias Gamma Indexed (REG Indexed) encoding for the location information and the Combined encoding for the mismatch information because they offer the all around compression. Currently, the quality scores are not yet encoded, and the decoded data recapitulates only the location and mismatch information. GenCompress performs two efficient passes on each dataset obtaining statistics on the mismatches during the first pass prior to actual encoding on the second pass. A more advanced data structure is currently being developed to avoid these two passes. However, this implementation allows for stream processing with minimal memory requirements. GenCompress is currently only compatible with output of the bowtie short-read aligner. A future implementation of a decoder will use the bowtie libraries to obtain the actual genomic sequences in the decoded output, and a framework exists for supporting multiple aligner suites.

Table 1.5 provides a comparison of the compression sizes and ratios for the three data sets for the best theoretical algorithm, our GenCompress implementation and the general purpose programs of gzip, bzip2 and 7zip. Data sizes for the raw sequence, uniform flat file format, and the output file from the bowtie alignment program are repeated from Table 1.5; and the compression ratios for each method is shown with respect to the different data formats. The bowtie alignment program produces additional data beyond just location and mismatch information, so that data is separated and compressed with 7zip. For GenCompress and the theoretical algorithm, the compression ratio of ”Bowtie” is just for the location and mismatch information, while the ”Bowtie+” ratio also includes the additional 7zip compressed data. For the gzip, bizp2 and 7zip, the ”Bowtie” results are for the full bowtie alignment output file.

Table 1.5 indicates the results are somewhat mixed. For data set 1, the general purpose programs get a better compression than GenCompress for the raw sequence and the uniform flat file format. The number of mismatches dominates data set 2, so novel methods to more efficiently encode those mismatches would be useful. However, GenCompress is equivalent
and up to 3x better than the general purpose programs for data sets 2 and 3. Interestingly in all cases, GenCompress does the best compression for the bowtie alignment data which is significant because that data ends up being larger than the original raw sequence reads by 2-3x.

Table 1.6 compares the timing results for compression and decompression of the data sets for GenCompress and the general purpose programs. GenCompress uses REG Indexed and combined mismatches encoding mismatches which corresponds to the compression results in Table 1.5, and the general purpose programs were invoked using default settings. We performed the timing of compression and decompression ten times on each dataset per method, and Table 1.6 provides the average timing for those ten runs. The hardware we use is a Dell T7550 with two quad core Intel Xeon E5450, 24GB of RAM and four 146GB SAS drives configured as RAID 1. The timing results show that both GenCompress and gzip are consistently faster than bzip2 and 7zip.

### 1.4 Conclusions

We have presented a set of data structures and compression algorithms for high-throughput sequencing data. We have transformed the nucleotide sequences into location and mismatch information through a mapping procedure to a reference genome, then applied fixed codes to encode that location and mismatch information in an efficient manner. We note that the mapping procedure does not need to be precise and find the correct position in the genome. We require only that a position is found because we use it only for the purpose of compressing and later decompressing the sequence. In fact, any arbitrary genome sequence can be used for mapping the reads, but it is likely that the genome which most closely matches the organism for the read data will provide the best performance. The methodology we have proposed is general, and we have illustrated its effectiveness on a representative set of HTS
data. Results show that some of the information in the HTS data can be compressed by a factor of 10 or more. The proposed algorithms are comparable or slightly better than the best general compression algorithms such as bzip2 and 7zip, but those programs require a much greater processing time compared to our algorithms.

The term “post-genomic era” has become somewhat fashionable, it is clear that the genomic era is far from over, and may in fact be only at an early stage of development. With the advent of HTS technology and increasingly new experimental protocols for using the technology, the sequence databases are only expected to continue rising in size. While the local storage of this data is not especially burdensome with inexpensive hard drives of greater than 1 TB available, sharing and transferring the data is time-consuming because network speeds are an order of magnitude slower than disk. Take for example the Yan Huang genome [179] we used for Dataset 3; the total amount of read data generated by the project is almost 120 GB. Uncompressed, a typical university network can upload this data in 120 hours at roughly 1 GB per hour; that may actually take a couple of weeks of real work time by a researcher handling the upload process, and that doesn’t even consider restarts that may occur due to network outages or that the network has to be shared with many other users. Extrapolating the compression results we obtained for chromosome 22 would reduce the total read data down to 5 GB, reasonable to upload in a single day. The situation becomes worse for downloading as numerous researchers may attempt to download the data, quickly saturating the server network bandwidth; advanced compression techniques such as we have introduced would allow more researchers to obtain the data in a timely fashion.

It is not likely that exactly one encoding strategy will be optimal for all types of HTS data. Different experimental conditions are going to generate various data distributions whereby one encoding strategy can be more effective than another. For encoding the location information, we have shown that two different strategies are effective. Furthermore in this article, we have only been able to consider Solexa data. It would be worthwhile to investigate both
454 data with its wide range of read lengths and SOLiD with its color space representation as the sequencing error distributions may be different for these technologies, thus affecting the mismatch locations and the strategy used for encoding them. It isn’t necessary that a single strategy be picked; our software computes the respective compression metrics for all of the strategies, so they can be compared and the best one chosen automatically. We have focused exclusively on the nucleotide sequence data of the short reads from HTS; however, there is additional data that is also present including read identifiers and quality scores. A complete solution would require that this information also be encoded so that it can be recovered later. The quality scores are needed as they are used by assembly programs for determining the statistical significance of the final assembled sequence, and also by programs that call SNPs and other structural variations. Read identifiers are often just sequential numbers with little meaning, but they do become needed for cross referencing when mate paired reads are sequenced. However in both cases, the encoding techniques described in this paper can be applied to significantly reduce the size of this data.

The techniques we have described assume that the reads have already been mapped to a reference genome before they are encoded, and in our analysis we have only discussed encoding mapped reads. However in all sequencing experiments, there are reads which do not map to the reference genome. In many cases, these reads are contaminants such as bacteria and might not be relevant for the particular experiment, but in other cases those reads may be important if it is a de novo sequencing project or the reference genome is unfinished or has poor coverage. The simple solution is to use a generic compression program like gzip for those unmapped reads. Another possibility is to use multiple reference genomes. If those reads map successfully to other genomes, then equivalent levels of compression can be expected with just a small amount of header overhead required to reference the genome used.

It is also common for reads to map to multiple places on the reference genome. Our current
implementation just takes the first best match, so additional optimization might be worthwhile to investigate. For example, if a read maps to one location which is very far from other reads, but has another mapping which is close to other reads, it would be advantageous to take the latter mapping because its relative starting position should be a smaller number. Optimizing all of the reads in this fashion is possibly a very difficult problem though because minimizing the distances between reads corresponds to doing clustering, and many optimal clustering algorithms are known to be NP-complete. However, approximate clustering which can be done with an online algorithm may offer some advantage. There is other data beside the read starting location which can be considered including the chromosome, strand and number of mismatches. The REG Indexed strategy performed consistently well, so taking alternative mappings for reads might create denser data bins which can be effectively encoded with fewer bits. Future research needs to consider whether the extra computational expense of performing these optimizations are a worthy compromise to the compression gains.

One might consider that because there is a lot of repetitive sequence within genomes, that an optimal encoding strategy like Huffman codes directly on the sequence itself would work very well. We investigated this possibility by encoding k-mers for data sets 1 and 2, the results can be seen in Table 7. The results seem to bear this out. Data set 1 which is a mapping experiment of retrotransposons would be expected to have considerable repetitive sequence, and Huffman coding does a very good job of compressing this data set. It does better than our algorithms and the general purpose compression programs. On the other hand, Huffman coding gets worse compression than our algorithms for data set 2.

Similar encoding ideas have been applied by us to a related but different problem, the storage of entire genomes [23]. Results indicate that almost 1000-fold compression can be obtained for the human genome by encoding only the variations of the genome against a reference genome and a reference SNP database. These techniques might be further improved if we consider that most SNPs in the human genome are biallelic (exist in only one of two forms),
are clustered together into haplotypes, and shared among many individuals. Instead of storing the variation for each read, a SNP map might be utilized which summarizes the variation across all of the reads or subsets of reads. Implementation requires careful consideration of the difference between true SNPs and sequencing errors, and whether an existing database like dbSNP is appropriate to use as reference (as it will not include mismatches due to sequencing errors) or a custom constructed database would work better.
1.5 Figures

1.5.1 Figure 1.1 - Dataset 2 Nucleotide Substitutions

![Figure 1.1: Distribution of nucleotide substitutions at each read position in Dataset 2. The shading of the bar indicates which nucleotide was present in the read.](image)

1.6 Tables

1.6.1 Table 1.1 - Example of Elias Gamma (EG) Coding.

<table>
<thead>
<tr>
<th>Number</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2-3</td>
<td>01x</td>
</tr>
<tr>
<td>4-7</td>
<td>001xx</td>
</tr>
<tr>
<td>8-15</td>
<td>0001xxx</td>
</tr>
<tr>
<td>16-31</td>
<td>00001xxxx</td>
</tr>
</tbody>
</table>

Table 1.1: Each integer $j$ is encoded by concatenating $\lfloor \log j \rfloor$ 0’s with the binary value of $j$. 
1.6.2 Table 1.2 - Example of Monotone Value (MOV) Coding.

The principle is illustrated using the vector of addresses (1, 2, 3, 9, 14, 26, 29). Each integer $j$ is converted to a binary representation of length $\lfloor \log j \rfloor$ which begins with a 1-bit. 0-bits are used between two consecutive integers only when the length (scale) increases. The number of 0-bits is equal to the increase in the length. The final encoding of the vector is $1 \ 0 \ 11 \ 00 \ 1001 \ 1110 \ 0 \ 11010 \ 11101$.

<table>
<thead>
<tr>
<th>Number</th>
<th>Encoding</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>1001</td>
</tr>
<tr>
<td>14</td>
<td>1110</td>
</tr>
<tr>
<td>26</td>
<td>11010</td>
</tr>
<tr>
<td>29</td>
<td>11101</td>
</tr>
</tbody>
</table>

Table 1.2: Example of Monotone Value (MOV) Coding
### 1.6.3 Table 1.3 - Statistics of Three High-Throughput Data Sets.

<table>
<thead>
<tr>
<th></th>
<th>Dataset 1</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reads (×10^6)</td>
<td>6.4</td>
<td>1.7</td>
<td>31</td>
</tr>
<tr>
<td>Read length</td>
<td>19</td>
<td>25</td>
<td>23-44</td>
</tr>
<tr>
<td>Coverage</td>
<td>Very sparse</td>
<td>Sparse</td>
<td>Full</td>
</tr>
<tr>
<td>File sizes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Raw Sequence</td>
<td>1,030,333,440</td>
<td>353,181,920</td>
<td>8,869,613,392</td>
</tr>
<tr>
<td>Uniform</td>
<td>912,352,288</td>
<td>252,540,968</td>
<td>4,946,059,912</td>
</tr>
<tr>
<td>Location</td>
<td>743,517,128</td>
<td>226,557,032</td>
<td>4,232,120,216</td>
</tr>
<tr>
<td>Mismatches</td>
<td>168,835,160</td>
<td>25,983,936</td>
<td>713,939,696</td>
</tr>
<tr>
<td>Bowtie</td>
<td>3,145,664,248</td>
<td>902,954,872</td>
<td>19,475,952,512</td>
</tr>
<tr>
<td>Bowtie Extra Fields</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gzip</td>
<td>50,382,904</td>
<td>106,576,328</td>
<td>839,247,848</td>
</tr>
<tr>
<td>7zip</td>
<td>36,306,064</td>
<td>93,238,688</td>
<td>778,347,264</td>
</tr>
</tbody>
</table>

Table 1.3: Statistics of Three HTS Data Sets

### 1.6.4 Table 1.4 - Compression Algorithm Results on Three HTS Data Sets.

<table>
<thead>
<tr>
<th></th>
<th>Dataset 1</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>gzip</td>
<td>50,382,904</td>
<td>106,576,328</td>
<td>839,247,848</td>
</tr>
<tr>
<td>7zip</td>
<td>36,306,064</td>
<td>93,238,688</td>
<td>778,347,264</td>
</tr>
</tbody>
</table>

Table 1.4: Compression Algorithm Results on Three HTS Data Sets.
<table>
<thead>
<tr>
<th></th>
<th>Dataset 1</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Standalone Methods</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Read Length</td>
<td>6,439,584</td>
<td>1,697,990</td>
<td>59,267,219</td>
</tr>
<tr>
<td>Chromosome</td>
<td>31,576,860</td>
<td>9,997,062</td>
<td>31,118,531</td>
</tr>
<tr>
<td>Strand</td>
<td>6,439,584</td>
<td>1,697,990</td>
<td>31,118,531</td>
</tr>
<tr>
<td># Mismatches</td>
<td>12,382,598</td>
<td>2,499,664</td>
<td>55,624,291</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>50,399,042</strong></td>
<td><strong>14,194,716</strong></td>
<td><strong>117,861,353</strong></td>
</tr>
<tr>
<td><strong>Start Location</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MOV†</td>
<td>121,565,953</td>
<td>44,200,254</td>
<td>787,554,494</td>
</tr>
<tr>
<td>EG†</td>
<td>236,691,716</td>
<td>86,701,276</td>
<td>1,543,990,407</td>
</tr>
<tr>
<td>REG†</td>
<td>10,745,562</td>
<td>26,180,752</td>
<td>76,430,489</td>
</tr>
<tr>
<td>Huffman</td>
<td>91,019,189</td>
<td>82,444,521</td>
<td>1,324,964,740</td>
</tr>
<tr>
<td>RHuffman</td>
<td>10,311,095</td>
<td>19,066,500</td>
<td>65,905,674</td>
</tr>
<tr>
<td><strong>Best Standalone</strong></td>
<td><strong>60,710,137</strong></td>
<td><strong>33,261,216</strong></td>
<td><strong>183,767,027</strong></td>
</tr>
<tr>
<td><strong>Combined Methods</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(C,S,M) Lookup</td>
<td>64,424,309</td>
<td>33,809,380</td>
<td>158,272,463</td>
</tr>
<tr>
<td>REG Indexed†</td>
<td>12,133,110</td>
<td>32,342,080</td>
<td>144,975,985</td>
</tr>
<tr>
<td><strong>Mismatches</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Nucleotide</td>
<td>13,917,023</td>
<td>1,307,870</td>
<td>53,441,350</td>
</tr>
<tr>
<td>From Start</td>
<td>30,028,807</td>
<td>4,177,576</td>
<td>159,433,004</td>
</tr>
<tr>
<td>From End</td>
<td>32,671,455</td>
<td>2,333,372</td>
<td>153,865,294</td>
</tr>
<tr>
<td>Total Start</td>
<td>43,945,830</td>
<td>5,485,446</td>
<td>212,874,354</td>
</tr>
<tr>
<td>Total End</td>
<td>46,588,478</td>
<td>3,641,242</td>
<td>207,306,644</td>
</tr>
<tr>
<td>Combined†</td>
<td>44,033,309</td>
<td>3,757,400</td>
<td>186,298,126</td>
</tr>
<tr>
<td><strong>Best Compression</strong></td>
<td><strong>56,078,940</strong></td>
<td><strong>35,983,322</strong></td>
<td><strong>390,541,330</strong></td>
</tr>
<tr>
<td>GenCompress</td>
<td><strong>56,166,419</strong></td>
<td><strong>36,099,480</strong></td>
<td><strong>390,541,330</strong></td>
</tr>
</tbody>
</table>

Table 1.4: Compression Algorithm Results on Three High-Throughput Data Sets.

1.6.5 Table 1.5 - Comparison of Compression Results.
<table>
<thead>
<tr>
<th>Dataset 1</th>
<th>Dataset 2</th>
<th>Dataset 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Sequence</td>
<td>1,030,333,440</td>
<td>353,181,920</td>
</tr>
<tr>
<td>Uniform</td>
<td>912,352,288</td>
<td>252,540,968</td>
</tr>
<tr>
<td>Bowtie</td>
<td>3,145,664,248</td>
<td>902,954,872</td>
</tr>
<tr>
<td>Bowtie Extra Fields (7zip)</td>
<td>36,306,064</td>
<td>93,238,688</td>
</tr>
<tr>
<td>Best Compression</td>
<td>56,078,940</td>
<td>35,983,322</td>
</tr>
<tr>
<td>Raw Sequence</td>
<td>18</td>
<td>10</td>
</tr>
<tr>
<td>Uniform</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>Bowtie</td>
<td>56</td>
<td>25</td>
</tr>
<tr>
<td>Bowtie+</td>
<td>34</td>
<td>7</td>
</tr>
<tr>
<td>GenCompress</td>
<td>56,166,419</td>
<td>36,099,480</td>
</tr>
<tr>
<td>Raw Sequence</td>
<td>18</td>
<td>9</td>
</tr>
<tr>
<td>Uniform</td>
<td>16</td>
<td>7</td>
</tr>
<tr>
<td>Bowtie</td>
<td>56</td>
<td>25</td>
</tr>
<tr>
<td>Bowtie+</td>
<td>34</td>
<td>7</td>
</tr>
<tr>
<td>gzip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Raw Sequence</td>
<td>41,378,624</td>
<td>95,688,992</td>
</tr>
<tr>
<td>Uniform</td>
<td>42,918,256</td>
<td>54,762,528</td>
</tr>
<tr>
<td>Bowtie</td>
<td>459,640,264</td>
<td>236,156,432</td>
</tr>
<tr>
<td>Bowtie+</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>bzip2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Raw Sequence</td>
<td>42,233,336</td>
<td>94,030,320</td>
</tr>
<tr>
<td>Uniform</td>
<td>36,400,576</td>
<td>54,656,000</td>
</tr>
<tr>
<td>Bowtie</td>
<td>250,373,616</td>
<td>171,835,792</td>
</tr>
<tr>
<td>Bowtie+</td>
<td>13</td>
<td>5</td>
</tr>
<tr>
<td>7zip</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Raw Sequence</td>
<td>30,651,664</td>
<td>83,319,584</td>
</tr>
<tr>
<td>Uniform</td>
<td>27,852,952</td>
<td>34,482,312</td>
</tr>
<tr>
<td>Bowtie</td>
<td>247,481,992</td>
<td>183,522,960</td>
</tr>
<tr>
<td>Bowtie+</td>
<td>13</td>
<td>5</td>
</tr>
</tbody>
</table>

Table 1.5: Comparison of Compression Results

1.6.6 Table 1.6 - Comparison of Compression and Decompression Timing.
### Table 1.6: Comparison of Compression and Decompression Timing

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Compression (sec)</th>
<th>Decompression (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>GenCompress</td>
<td>gzip</td>
</tr>
<tr>
<td>Dataset 1</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>Dataset 2</td>
<td>5</td>
<td>13</td>
</tr>
<tr>
<td>Dataset 3</td>
<td>111</td>
<td>70</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>15</td>
<td>13</td>
</tr>
</tbody>
</table>

1.6.7 **Table 1.7 - Sequence encoding using Huffman Trees.**

The data is preprocessed by counting the frequencies of $k$-mers, and this is used to build a Huffman tree. The tree is used to encode the data, and the number of bits needed to store the data as well as the tree are given.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$k$</th>
<th>Sequence bits</th>
<th>Tree bits</th>
<th>Total bits</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>31,674,558</td>
<td>40</td>
<td>31,674,598</td>
</tr>
<tr>
<td>2</td>
<td>28,340,409</td>
<td>324</td>
<td>28,340,733</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>27,708,166</td>
<td>1,951</td>
<td>27,710,117</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>22,565,417</td>
<td>10,471</td>
<td>22,575,888</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>19,126,288</td>
<td>53,178</td>
<td>19,179,466</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>21,056,658</td>
<td>256,303</td>
<td>21,312,961</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>94,680,841</td>
<td>52</td>
<td>94,680,893</td>
</tr>
<tr>
<td>2</td>
<td>81,954,644</td>
<td>549</td>
<td>81,955,193</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>81,038,827</td>
<td>4,303</td>
<td>81,043,130</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>80,554,549</td>
<td>27,458</td>
<td>80,582,007</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>83,570,470</td>
<td>148,206</td>
<td>83,718,676</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>79,977,714</td>
<td>622,784</td>
<td>80,600,498</td>
<td></td>
</tr>
</tbody>
</table>

**Table 1.7: Sequence encoding using Huffman Trees**
Chapter 2

MotifMap: Novel method for TFBS prediction

2.1 Introduction

Among the 3 billion bases of the haploid human genome, only a small portion (<2%) corresponds to protein-coding regions. A central challenge of biology is to map and understand the role of the remaining 98% non-coding regions of the human genome. It is commonly believed that many of these non-coding regions are involved in gene regulation, but their specific roles and organization, including which regulatory motifs are contained in which regions, are still poorly known.

Mapping the locations of regulatory motifs across the human genome is challenging because these motifs are typically short, contain fuzzy sequence patterns, and are hidden in the vast background of non-coding sequences. Hence, the key computational challenge is to detect the locations of the motifs without introducing too many false positives.
Comparative genomics provides a powerful tool for detecting regulatory elements in the genome. This is because functional elements often evolve at a much slower rate than neutral sequences, and therefore they often stand out from the surrounding sequences by virtue of their greater levels of conservation. Previous work has demonstrated the power of comparative genomics for discovering novel regulatory motifs in human (Elemento and Tavazoie, 2005 [38]; Ettwiller et al., 2005 [44]; Xie et al., 2005 [196]). However, whether comparative genomics can provide sufficient power for detecting individual motif sites (not just overall motif patterns) in the human genome has not been fully addressed. In particular, a global map of motif sites for all known regulatory motifs in the human genome has not been attempted.

Recent availability of a dozen placental mammalian genomes significantly boosts our power for detecting motif sites in the human genome (Miller et al., 2007 [131]). These genomes are closely related to each other, and thus likely share a basic cassette of regulatory motifs. On the other hand, these genomes have been carefully chosen to represent distinct branches of the mammalian evolutionary tree. As such, they are ideal for separating regulatory sequences from neutral ones (Margulies et al., 2005 [123]).

When using multiple species for motif site discovery, one must take into consideration the phylogenetic relationship between the species. This is important for distinguishing truly conserved sites from spurious ones due to species proximity. A number of computational algorithms have been proposed (Li and Wong, 2005 [112]; Moses et al., 2004 [135]; Siddharthan and van Nimwegen, 2007 [167]). Most of these methods use a probabilistic framework by modeling the evolutionary process of a motif site explicitly and performing statistical inference over the phylogenetic tree. Although these methods have had some success, mostly in yeast, several factors limit their applicability and effectiveness. First, it is not completely clear how to model the evolution of regulatory motif sites. All the previous methods assume that the nucleotides at different positions of a motif site evolve independently. This is clearly an over-simplification. For instance, an insertion or deletion event at a single position can
completely abolish a motif site, and consequently relax evolutionary constrains at all other positions. Recent work has demonstrated the importance of considering such inter-position dependencies in modeling motif site evolution (Lusk and Eisen, 2008 [119]). Second, most of the previous methods assume that motif sites are conserved throughout the evolution of all the species being compared. In reality, it is often the case that a motif site is conserved and shared in only a subset of the species or lineages. Third, these methods are highly sensitive to the quality of the multiple sequence alignments and to missing sequences. This could be problematic for the mammalian genomes used here, which are repeat-rich and littered with sequencing gaps.

Recently, an alternative method has been proposed for motif site discovery using multiple genomes (Stark et al., 2007 [172]). The method works by first identifying the set of species in which the motif occurs, calculating the total branch length score (BLS) of the subtree covering these species and then using BLS to quantify the conservation level of a motif site. The scoring scheme has been successfully applied for motif site discovery in both flies (Stark et al., 2007 [172]) and mammals (Xie et al., 2007 [191]). This method does not rely on sequence alignments to fit an evolutionary model and, by construction, automatically focuses only on the relevant subset of species that may share a given element. As such, it is not sensitive to the limitations outlined above.

Although useful in practice, the method based on BLS leaves a lot of room for improvement. First, unlike some of the other methods, BLS lacks a solid theoretical foundation. Thus, it is unclear under which circumstances the method will be more effective or more prone to errors. Second, it is often difficult to strictly classify whether a sequence corresponds to motif site or not. It is more desirable to take the uncertainty of motif site matching into consideration. Third, a more principled approach is needed for determining which set of ancestral sequences contains a given motif.

Here we propose a new scoring scheme, the Bayesian branch length score (BBLs), to ad-
dress these issues. Using BBLS and the genomes of 18 mammals, we are able to derive a genome-wide map of over 380 known regulatory motifs and assess its accuracy. Browsing and visualization of these elements and the corresponding map is achieved through the MotifMap web server.

2.2 Methods

2.2.1 Known motifs and motif-matching z-score

Motifs were extracted from two major transcription factor binding sites databases: Transfac (Wingender et al., 1996 [186]) and JASPAR (Sandelin et al., 2004 [159]). We used only motifs associated with mammalian cells. In total, we curated 560 motifs, represented in the form of position-specific frequency matrices. We used a log-odds score $y$ to quantify how well a sequence element $x$ matches a motif, defined by $y(x|\theta) = \log[P(x|\theta)/P(x|\theta_0)]$, where $\theta$ is the frequency matrix of the motif and $\theta_0$ is the background frequency of the four nucleotides across the entire genome. We further normalized the score to be between 0 and 1, $S(x, \theta) = (y(x|\theta) - y_{\text{min}})/(y_{\text{max}} - y_{\text{min}})$, where $y_{\text{min}}$ and $y_{\text{max}}$ are the minimum and maximum log-odds scores the motif can possibly achieve. Thus $S(x, \theta)$ denotes the motif-matching score for sequence $x$ and motif $\theta$.

To determine the threshold score for calling a match, we randomly sampled 10 million locations in nonrepeat regions of the human genome and calculated motif matching scores at these random locations. For each motif, we calculated the mean ($\mu$) and variance ($\sigma^2$) of the motif-matching scores at these locations. Based on $\mu$ and $\sigma$, we converted each motif-matching score into a z-score, $z(x, \theta) = (S(x, \theta) - \mu)/\sigma$. We used a z-score threshold of 4.27 (corresponding to a threshold $S_{\text{th}}(\theta) = \mu + 4.27\sigma$ on $S$) for calling a site a match, corresponding to a nominal P-value of 1e-6 for finding a motif purely by chance under a
normal distribution model.

2.2.2 Phylogenetic tree and sequence alignments

Genomes and sequence alignments of 18 mammals used in this study were downloaded from the USCS genome browser (http://genome.ucsc.edu/) (Miller et al., 2007 [131]). The phylogenetic tree connecting the species (Supplementary Material) was constructed using the 4-fold degeneracy of the third codon position of coding DNAs (Miller et al., 2007 [131]). The tree has a total branch length of 3 mutations per site. We extracted orthologous sequences from the whole genome alignment. When searching for motif sites in other species, we extended the alignment at both ends by 15 bp to account for potential misalignments.

2.2.3 Bayesian branch length score

Denote the phylogenetic tree of the $n$ species being compared by $T$ and the nodes in the tree by $i=1,\ldots,N$, where $N=2n-1$. Without any loss of generality, we assume that the first $n$ nodes are leaf nodes and the $N^{th}$ node is the root of the tree. Suppose we are provided with the orthologous sequences of a putative motif site in the genomes of the $n$ species. Denote the set of orthologous sequences by $V$ (visible), and the set of corresponding ancestral sequences associated with the nonleaf nodes of the tree by $H$ (hidden).

We assume that evolution along each edge of the tree is either neutral or constrained (i.e. under selective pressure to preserve a motif site). We use a binary variable $\sigma_i$ to denote whether the edge leading to node $i$ is evolutionary constrained ($\sigma_i = 1$) or not, when traversing the tree from the root to the leaves. For a given assignment vector $\sigma$ with some nonzero entries, the log-odds score of observing the set $V$ under $\sigma$ against the neutral model ($\sigma = 0$) can be computed as
where the summation is over all ancestral nodes, the sequences of which are not directly observable. It is difficult to know a priori which edges are evolutionarily constrained. One strategy to deal with this uncertainty is to take a Bayesian approach and integrate over both alternatives

\[ L(V) = \log \sum_{H} \sum_{\sigma} P(V, H|\sigma) P(\sigma) \log \sum_{H} P(V, H|\sigma = 0) \]

where \( P(\sigma) \) is a prior distribution over \( \sigma \).

To calculate \( L(V) \) one must explicitly model the evolution of the motif sequences over all the branches of the tree. Most previous attempts have taken a simplified approach to this problem by assuming independent evolution at different positions of the motif sequence. This is clearly an oversimplification. Here we use a different method to derive an approximation to \( L(V) \) that avoids direct modeling of motif site evolution.

Using Jensen’s inequality, we note that the \( L(V) \) is lower-bounded by

\[ L(V) \geq \sum_{H} Q(\sigma|V) \left[ \log \sum_{\sigma} P(V, H|\sigma) P(\sigma) - \log P(V, H|\sigma = 0) \right] \]

where \( Q(\sigma|V) = \exp P(H|V \sigma = 0) \) is the posterior distribution of \( H \) under the neutral model. Because of the tree structure, the joint distribution \( P(V, H) \) can be factorized as a product and the log-odds score becomes

\[ L(V) \geq \sum_{i=1}^{N-1} \sum_{x_i, x_{i+1} \in H} Q(x_i, x_{i+1}) \left[ \log \sum_{\sigma} P(\sigma|x_i, x_{i+1}, \sigma) - \log P(x_i, x_{i+1}, \sigma = 0) \right] \]
where $x_i$ denotes the sequence at node $i$, and $\pi(i)$ represents the parent of node $i$. $Q(x_i, x_{\pi(i)})$ is the posterior distribution of the sequences at two neighboring nodes of the tree, conditioned on the set $V$ under the neutral model. Assuming a non-informative prior on $P(\sigma_i)$ applying Jensen’s inequality again, we have

$$L(V) \geq \sum_{i=1}^{N-1} \sum_{x_i, x_{\pi(i)} \in H} Q(x_i, x_{\pi(i)}) \left[ R(\sigma_i = 1|x_i, x_{\pi(i)}) \log \frac{P(x_i|x_{\pi(i)}, \sigma_i = 1)}{P(x_i|x_{\pi(i)}, \sigma_i = 0)} - \log 2 \right]$$

where $R(\sigma_i = 1)$ is the posterior probability for edge $i$ to be constrained.

If the evolution of a motif site at a given edge is truly constrained, we expect the corresponding log likelihood ratio term in $L(V)$, $\log(P(x_i|x_{\pi(i)}, \sigma = 1)/P(x_i|x_{\pi(i)}, \sigma = 0))$, to be proportional to the length of the edge (Supplementary Material[193]). Under this assumption, the lower bound on $L(V)$ can be approximated by the sum of the length of all the edges, with edge $i$ weighted by the probability $R(\sigma_i = 1)$:

$$L \geq k * \text{BLS} + C, \quad \text{where} \quad \text{BLS} = \sum_{i=1}^{N-1} R(\sigma_i = 1) l_i$$

and $l_i$ is the length of the edge leading to node $i$. $k$ and $C$ are a constants.

The BLS can be viewed as a special case of the above approximation, in which $R(\sigma_i = 1) = 1$ for all leaf nodes whose sequences contain a motif site and for the ancestors contained in the subtree connecting these leaf nodes. In other words, the state variables $\sigma_V = (\sigma_1, \sigma_2, \ldots, \sigma_n)$ of the leaf nodes are now deterministic ($\sigma_i = 1$ if leaf node $i$ contains a motif site), and the $\text{BLS}(\sigma_V)$ is calculated by

$$\text{BLS}(\sigma_1, \sigma_2, \ldots, \sigma_n) = \sum_{i=1}^{N-1} \sigma_i(\sigma_V) l_i$$
Here $\sigma_i(\sigma_V) = 1$ if the subtree $T_i$ of node $i$ contains a leaf node with the state variable being 1, and in addition the complement $T_i^c$ of the subtree also contains a leaf node with the state variable being 1. Note that $T_i$ consists of node $i$ and all of its descendents, whereas $T_i^c$ is comprised of all other nodes not included in $T_i$.

### 2.2.4 A specific BBLS proposal

We consider a direct extension of the BLS mentioned above. Suppose there is uncertainty in determining whether a leaf node contains a motif or not, and the uncertainty is described by the probabilities $p_i = P(\sigma_i = 1)$ for all $i = 1, \ldots, n$. Given the probabilities for all the leaf nodes $p_v = (p_1, p_2, \ldots, p_n)$, a straightforward extension of the standard BLS is to sum over the uncertainties

$$BBLS(p_V) = \sum_{\sigma_V} P(\sigma_V)BBLS(\sigma_V) = \sum_{i=1}^{N-1} \left[ \sum_{\sigma_V} P(\sigma_V)\sigma_i(\sigma_V) \right] l_i$$

where $P(\sigma_V) = P(\sigma_1)P(\sigma_2)\ldots P(\sigma_n)$ and the sum is over all combinations of possible states for the $n$ leaf nodes. In the context of the general BBLS framework discussed above, this specific proposal corresponds to taking $R(\sigma_i) = \Sigma\sigma_V P(\sigma_V)\sigma_i(\sigma_V)$.

BBLS($\rho V$) involves the summation of $2^n$ terms. Therefore, in general, it is infeasible to calculate BBLS directly using the above equation when $n$ is large. However, in the Supplementary Methods[193], we prove that BBLS($\rho V$) can be calculated in time that scales linearly with $n$. Specifically, it can be calculated efficiently using the following formula

$$BBLS(p_V) = \sum_{i=n+1}^{N} P(\sigma_{c1}(i) = 1)P(\sigma_{c2}(i) = 1)P(\sigma_{T_i} = 0) \left[ l_{c1}(i) + l_{c2}(i) \right]$$
where $c^1(i)$ and $c^2(i)$ denote the two child nodes of node $i$. $P(\sigma = 1)$ is the probability that $T_i$ contains at least one leaf node with the state variable being 1, and $P(\sigma_{T_i} = 0)$ is the probability that $T_i$ contains no leaf nodes with the state variable being 1. Both $P(\sigma = 1)$ and $P(\sigma_{T_i} = 0)$ can be calculated recursively, bottom-up from the leaf nodes to the root for $P(\sigma = 1)$, and top-down from the root to the leaf nodes for $P(\sigma_{T_i} = 0)$.

$$
P(\sigma = 1) = 1 - [1 - P(c^1(i) = 1)] [1 - P(c^2(i) = 1)]
$$

$$
P(\sigma_{T_i} = 0) = P(\sigma_{T_i} = 0) [1 - P(c(i) = 1)]
$$

where $s(i)$ denotes the sister node of node $i$. The variable $l^*_i$ is the effective branch length associated with node $i$. It too can be calculated recursively bottom-up from the leaf nodes to the root according to

$$
l^*_i = l_i + \frac{P(c^1(i) = 1) l^*_i c^1(i) + P(c^2(i) = 1) l^*_i c^2(i)}{P(\sigma = 1)}
$$

with the initialization $l^*_i = l_i$ for leaf node $i$.

The above method does not depend on how $P(\sigma = 1)$ is assigned. For motifs modeled by a positional weight matrix, $P(\sigma = 1)$ is assigned for each leaf node $i$ according to the motif-matching score of the sequence at the corresponding node

$$
P(\sigma = 1) = \max \left\{ \frac{(S(x_i, \theta) - S_{th}(\theta))}{(S_{\max}(\theta) - S_{th}(\theta))}, 0 \right\}
$$

That is we assign a nonzero probability only to the nodes with motif-matching score above the threshold $S_{th}$. The probability itself is chosen to be linearly proportional to the motif-matching score.
2.2.5 Estimating the false discovery rate

For each known motif, we generated 10 control motifs by randomly shuffling the columns of the position-specific frequency matrix associated with the known motif, while keeping the frequency of the four nucleotides in each column unchanged. Because the mutation rate of the CG-dinucleotide is typically much higher than the rate of the other 15 dinucleotides, the CG-content of the motif was kept unchanged (i.e. by tying together neighboring columns with a high CG-dinucleotide frequency) during the shuffling. We then applied the same motif-site discovery algorithm to these control motifs. The false discovery rate (FDR) is estimated to be the median number of sites discovered for the control motifs divided by the number of sites discovered for the known motif.

2.3 Results

We have developed a computational pipeline to search for the sites of 560 known motifs in the non-coding and non-repeat regions of the human genome. Once a putative site is detected in homo, we then determine whether the site also occurs in the orthologous regions of other mammalian genomes (Supplementary Fig. 1[193]). The pipeline returns the species within which the motif occurs and corresponding motif-matching log-odd scores, determined by the position-specific frequency matrix of the motif. We initially retained those sites with motif-matching z-score >4.27 and with matching sites in at least four nonprimate species. For each of these identified sites, we then summarize its conservation level in other species using both the total BLS and the BBLS.

The initial list of candidate motif sites includes 3.9 million sites (corresponding to 1.9 non-overlapping unique sites) throughout the human genome. Because these sites are identified purely by computational methods, it is essential to find ways to rank these sites and estimate
the accuracy of these predictions. Next we seek to address these questions.

2.3.1 Ranking motifs according to their BBLS

Each of the identified motif sites is associated with two conservation scores: BLS and BBLS. We tested which of the two scores can better distinguish bona fide sites from spurious ones. For this purpose, we used the CTCF motif as a benchmark. The CTCF motif is a good testing benchmark because so far it is the only motif whose locations have been experimentally mapped in multiple tissues (T cells and fibroblasts) and with multiple methods, including both ChIP-on-chip (Kim et al., 2007[99]) and ChIP-seq methods (Barski et al., 2007[15]). Altogether, the previous experimental efforts have identified a total number of 26,114 CTCF sites in the human genome.

Our initial list of candidate motif sites includes 25,098 CTCF sites, among which 9,761 (39%) overlapped with experimentally identified sites. Using these 9,761 sites as our positives, we examined how true positive and false positive rates for CTCF site prediction change when different threshold scores are chosen for BLS or BBLS. The ROC curves for these two different scoring schemes are shown in Figure 2.1A. Although both of the scoring methods clearly have predictive power at separating true CTCF sites from spurious ones, there are considerable differences in predictive accuracy among them. In particular, the method based on BBLS significantly outperforms BLS. The area under the curve (AUC) of the ROC curve for the BBLS method is 0.81, considerably better than the AUC for BLS (0.75).

As a comparison, we also tested the performance of BBLS against two other commonly used methods for ranking candidate motif sites: PhastCons conservation score (Siepel et al., 2005[169]) and MONKEY (Moses et al., 2004[135]). The PhastCons score is calculated using a phylogenetic hidden Markov model (HMM). It provides a measure of how an individual nucleotide is conserved without referencing the underlying motif model. In contrast to
Figure 2.1: ROC curves for different methods for predicting the sites of CTCF (A), NRSE (B) and P53 (C). PhastCons: PhastCons conservation score; BLS: branch length score; BBLBS: Bayesian branch length score; MONKEY: conservation P-value calculated by MONKEY (Moses et al., 2004); and UCSC TFBS: predicted sites from the UCSC genome browser.

PhastCons, MONKEY specifically models the evolution of a motif site by taking into account the weight matrix model associated with the motif although, similarly to PhastCons, it also assumes that each individual position of the motif evolves independently. We calculated true positive and false positive rates for the CTCF site prediction by choosing different PhastCons
and MONKEY conservation score thresholds, and plotted their ROC curves (Fig. 2.1A). The AUC of the ROC curves for PhastCons and MONKEY is 0.56 and 0.69, respectively, both of which are considerably <0.81 for BBLS (Table 2.1).

<table>
<thead>
<tr>
<th>Factor</th>
<th>CTCF</th>
<th>NRSE</th>
<th>P53</th>
<th>MYC</th>
<th>STAT1</th>
<th>NFkappaB</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLS</td>
<td>0.747</td>
<td>0.756</td>
<td>0.659</td>
<td>0.634</td>
<td>0.554</td>
<td>0.708</td>
</tr>
<tr>
<td>BBLS</td>
<td>0.814</td>
<td>0.941</td>
<td>0.861</td>
<td>0.683</td>
<td>0.606</td>
<td>0.722</td>
</tr>
<tr>
<td>MONKEY</td>
<td>0.693</td>
<td>0.658</td>
<td>0.566</td>
<td>0.540</td>
<td>0.545</td>
<td>0.558</td>
</tr>
<tr>
<td>UCSC TFBS</td>
<td>–</td>
<td>0.681</td>
<td>0.596</td>
<td>0.587</td>
<td>0.529</td>
<td>0.712</td>
</tr>
<tr>
<td>PhastCons</td>
<td>0.557</td>
<td>0.533</td>
<td>0.481</td>
<td>0.548</td>
<td>0.494</td>
<td>0.651</td>
</tr>
</tbody>
</table>

Table 2.1: Comparison of the area under ROC curve (AUC) for five different methods for predicting motif sites of six transcription factors. Best results in bold.

In addition to CTCF, we also compared the performance of the different methods for predicting the sites of five other motifs NRSE (neuron-restrictive silencer element, Johnson et al., 2007[84]), P53 (Wei et al., 2006[181]), STAT1 (Robertson et al., 2007[153]), MYC (Zeller et al., 2006[199]) and NFkappaB (Lim et al., 2005[115]). The binding sites of these five motifs have recently been mapped in human cells using the high-throughput techniques ChIP-seq or ChIP-pet (chromatin immunoprecipitation coupled with paired end ditag sequencing). Overall, the experiment work has identified 2,274 NRSE sites, 542 P53 sites, 41,515 STAT1 sites, 4,296 MYC sites and 488 NFkappaB sites in human cells. We tested the performance of the four methods discussed above for predicting these experimentally identified sites and plotted their ROC curves in Figure 2.1B and C and Supplementary Figs S3S8[193]. In addition to the four methods mentioned above, we also tested how the predicted TFBS sites available from the UCSC genome browser (Karolchik et al., 2003[90]) overlap with the experimentally identified sites for the five additional motifs (CTCF predictions are not available from the UCSC TFBS). TFBS uses the sum of the motif-matching scores in different species to score a motif site, without taking into account the phylogenetic relationship between the species.
Note that in evaluating the performance of these different methods, we have defined true positive sites as those that are supported by the chip data and that match to a given positional weight matrix. They represent 37% of the 26,114 CTCF sites, 45% of the 2,274 NRSE sites, 21% of the 542 P53 sites, 24% of the 4,296 MYC sites, 3% of the 41,515 STAT1 sites and 12% of the 488 NFkappaB sites that are identified by the chip experiments. If all experimentally identified sites are used as true positives, the false negative rates corresponding to each method will need to be uniformly scaled down by these factors.

BBLS consistently outperforms all other four approaches mentioned above for predicting the sites of the five motifs (Table 2.1). The AUC of the ROC curves for BBLS is 0.94, 0.86, 0.68, 0.61 and 0.72 for NRSE, P53, MYC, STAT1 and NFkappaB, respectively, all of which are considerably better than the second best method (BLS for NRSE, P53 and STAT1 and TFBS for NFkappaB). The PhastCons conservation score consistently ranks lower than the four other methods (except for NFkappaB), reinforcing the importance of considering the motif model in measuring cross-species conservation.

Because of the BBLS’s better performance, we retained it for further analyses in the following.

2.3.2 Properties of identified sites

As the motif sites are predicted purely by computational methods, we have followed additional lines of evidence to support the functionality of these sites.

First, we examined how many of the sites are expected to occur purely by chance. For this purpose, for each motif, we generated a set of 10 corresponding control motifs (see Methods section), and identified their sites using the same computational pipeline described above. In total, we obtained 2.6 million sites for these control motifs, based on which we estimate that about one-third of the 3.9 million predicted sites likely correspond to true functional
elements. This is encouraging given the high chance of random matches for short motifs in the human genome, and the heterogeneity of the quality of the curated motifs.

By increasing the threshold cutoffs of the BBLS and motif matching z-score, we can further improve the accuracy of our predictions. Consider, for example, the CTCF motif. The method identified a total number of 25,098 sites, 45% of which were estimated to be false positives (based on control motifs). However, by increasing the BBLS threshold, the prediction accuracy can be improved significantly (Fig. 3). In fact, using a combination of stringent BBLS and motif z-scores, we were able to accurately predict the sites for 155 motifs with FDRs <10%, leading to a total number of 204,421 (corresponding to 122,277 non-overlapping) highly accurate predictions (Table 2.2, Supplementary Table S1[193]). By relaxing the FDR criterion to 0.5, the sites for an additional 225 motifs can be reliably predicted, corresponding to a total number of 1.5 million (787,517 non-overlapping) sites (Supplementary Table S2[193]). For the remaining 180 motifs, it seems that we still lack the power to pinpoint their locations with high precision. This could result from several causes, including their small size, the incorrect characterization of their position weight matrices or simply because they are lineage-specific and not shared by most of the mammals.

The number of sites identified for each motif is highly uneven (Table 2.2, Supplementary Fig. S1[193]). A few motifs have an especially high number of instances in the genome. For instance, the top four most highly frequent motifs (SF-1, RFX1, CTCF and SP1) each occurred over 10,000 times in the genome, while by contrast the median number of sites among the motifs is only 384. The SF-1 motif contains an 8-mer sequence pattern (TRACCTTG) recognized by many nuclear hormone receptors. Its large number of occurrences (15,492) in the genome may suggest the widespread role of nuclear receptors in gene regulation. The RFX1 motif is similar to the X-box motif that has been extensively studied in nonverte-
brates, such as yeast and nematode. In Caenorhabditis elegans, several hundreds X-box sites appear upstream of genes involved in the development of sensory cilia (Efimenko et al., 2005), and play an important role in cilia genesis. In mammals, the RFX1 elements are less well studied. Their high level of occurrence in the mammalian genomes is not expected, and suggests that RFX1 might be involved in roles other than cilia genesis in the mammalian gene regulatory system. The third most frequent motif is recognized by the CTCF protein, which is involved in insulator activity, and plays an important role in demarcating distinct regions of the genome into functionally distinct domains (Kim et al., 2007[99]; Xie et al., 2007[191]).

![Figure 2.2: Estimated false positive rate as a function of BBLIS. Blue represents the number of predicted sites for the CTCF motif. Gray represents the number of predicted sites for the CTCF control motif. Red curve plots the rate of true positives as a function of BBLIS.](image)

Second, we examined the distribution of the predicted motif sites in the genome relative to the location of the genes. For this analysis, we focused on the 122,277 sites corresponding to the 155 highly specific motifs discovered above. For each of these sites, we identified its nearest gene and the distance between the motif site and the transcriptional start site (TSS)
of the gene. We found a significant enrichment of motif sites in the regions around the TSS (Fig. 2.2). In fact, as much as 32% of the sites are located within 2 Kb of a TSS. This number corresponds to a 10-fold enrichment over what is expected by chance (for random sites, only 3% are expected). The enrichment near the TSS is of course concordant with a possible involvement in the regulation of the corresponding genes.

Third, we examined the overlap between the predicted motif sites and the experimentally identified ones. Again we used the CTCF motif as a test case. Altogether, the previous experimental efforts identified a total number of 26,114 CTCF sites in the human genome. In our computational predictions, we identified 12,295 conserved sites with FDR <0.1. Of these sites, 7,321 (60%) are also identified by the experimental methods. In contrast, the control motif of CTCF only discovered 1,130 sites, out of which only 42 overlapped the experimental identified sites. This demonstrates the high specificity of the computational predictions.

Taken together, these analyses provide strong evidence supporting the functionality of the predicted sites. The predictions have relatively low false positive rates, and as such provide a reliable set of sites for future experimental validations.

2.3.3 Comparison with PhastCons elements

Previous comparative studies have discovered that a significant portion of the human genome evolves at a much slower rate than that of neutral sequences. For instance, the PhastCons program has identified over 2 million conserved sequence elements in the genome, with average size of about 150 bp (Miller et al., 2007[131]; Siepel and Haussler, 2004[169]). The PhastCons elements also show enrichment in regions near gene TSSs, although less significantly than the enrichment shown by the predicted motif sites (Fig. 2.3). We checked the overlap between the predicted motif sites and the PhastCons elements, and found that most
of the predicted sites (72%) are located inside these PhastCons elements. However, a significant portion (28%) of the motif sites is not detected by the previous method. Most likely, this is because these sites work mostly alone and as such are located in regions without other functional elements. The PhastCons program lacks sufficient power for detecting such short sequence elements. It is worth noting that the FDRs for the predicted sites located outside the PhastCons elements are typically comparable to those located inside the PhastCons elements (Supplementary Fig. S10[193]). Thus our method based on matching conserved motif sites provides a complementary approach to the commonly used PhastCons program for detecting functional elements in the genome.

![Figure 2.3: Distribution of the motif sites relative to the TSSs of genes. Also shown are the distributions of random loci (solid) and the conserved PhastCons elements (excluding coding exons) (dashed) relative to TSS.](image)

### 2.3.4 Motif sites overlapping SNPs

Recent progress in genome-wide association studies have identified many genetic variations (mostly SNPs) associated with complex phenotypes. One interesting observation emerging from these studies is that most of the discovered SNPs occur outside of protein-coding regions.
and, in most case, are not associated with any known functions. There is a great deal of interest in figuring out the potential functions of these SNPs.

We checked the overlap between known SNPs and the predicted motif sites. Of the 12 million SNPs deposited in the dbSNP database (version 126), 89,032 SNPs overlap with at least one of the 1.9 million non-overlapping initial candidate motif sites, corresponding to a density of 2.99 SNPs/Kbp. Of the 4 million SNPs (release 22) genotyped in three human populations by the HAPMAP consortium (Frazer et al., 2007[52]), 42,548 overlap with the initial candidate motif sites. If we focus on the high-confidence list of 787,517 million motif sites discovered with FDR <50%, we find 28,294 dbSNPs (density: 2.65 SNPs/Kbp) and 13,535 HAPMAP SNPs overlapping these sites. If we focus on the high-confidence list of 122,277 motif sites discovered with FDR <10%, we find 4,293 dbSNPs (density: 2.59 SNPs/Kbp) and 1,864 HAPMAP SNPs overlapping these sites. The decrease in SNP density for the three sets of predicted sites likely reflects the stronger purifying selection acting on sites associated with higher prediction confidence. The list of these SNPs and their corresponding motifs (see Supplementary Website) provide a concrete and testable hypothesis regarding the potential functional role of these SNPs. An interesting follow-up study would be to investigate the correlation between the genotype of these SNPs and the variation on the gene expression of their corresponding target genes. The list of the SNPs may also be useful when selecting SNPs for genotyping in disease gene mapping studies or for testing SNPs involved in recent positive selection (Sabeti et al., 2007[156]; Wang et al., 2006[178]).

2.4 Web server and interface

We have constructed a database and web server for the predicted motif sites, and created a user-friendly web interface for retrieving, analyzing and visualizing these data (Supplementary Fig. S9[193]).
The web interface allows users to filter motif sites using different threshold scores and conservation criteria, including BLS and BBLS, as well as FDR. For a given motif, users can retrieve the genome-wide locations of the motif, and load them into the USCS genome browser for visualization.

2.5 Discussion

We have created an initial map of candidate regulatory motif sites across the human genome. The map currently contains 3.9 million sites, corresponding to 560 motifs. We have demonstrated that the method is especially effective for 155 motifs, for which the predicted sites have an estimated FDR <0.1.

While here we have focused on the human map, it is clear that the same methods give immediately similar maps for all 17 species. In particular, the mouse and rat maps may also be of general interest and will be made available in the near future through the same web interface.

Because the transcription factors binding to the motifs used in this study are known, it is possible to construct a regulatory network for each genome by connecting these transcription factors and their target genes (estimated from the presence of motif sequences near the corresponding TSS). This could provide an alternative strategy for regulatory network construction and, in future work, it would be interesting to see how the network structures compare with those derived from other methods.

Our prediction methods depend heavily on comparative genomics to boost the signal-to-noise ratio of the motif signals. It has been noticed that many regulatory sites in human are lineage-specific and do not appear to be conserved in other species (King, 2007[100]). For these motif sites, methods other than sequence comparison are required. One potential
direction could be to search for a local clustering of motif sites rather than an individual site, and to develop methods for detecting regulatory modules.

The computational analysis of the motif sites presented here is, of course, only a first step towards building a comprehensive map of regulatory elements contained in the human genome. With the identification of additional motifs and better methods for mapping motif sites, the regulatory motif map will be further refined. We intend to provide an active and regularly updated central server, and make it useful for biologists interested in gene regulation in humans, as well other mammals.
<table>
<thead>
<tr>
<th>Name</th>
<th>Number of sites</th>
<th>FDR</th>
<th>Motif score</th>
<th>BBLs</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>SF-1</td>
<td>15861</td>
<td>0.10</td>
<td>0.982</td>
<td>0.933</td>
<td>M00727</td>
</tr>
<tr>
<td>RXF1</td>
<td>15319</td>
<td>0.10</td>
<td>0.767</td>
<td>0.500</td>
<td>LM1&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>CTCF</td>
<td>12195</td>
<td>0.09</td>
<td>0.741</td>
<td>0.667</td>
<td>LM2&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>Sp1</td>
<td>10507</td>
<td>0.09</td>
<td>0.912</td>
<td>0.500</td>
<td>M00931</td>
</tr>
<tr>
<td>Artt</td>
<td>7434</td>
<td>0.10</td>
<td>0.874</td>
<td>1.433</td>
<td>MA0004&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>AP-1</td>
<td>6513</td>
<td>0.10</td>
<td>0.906</td>
<td>1.167</td>
<td>M00925</td>
</tr>
<tr>
<td>Lhx3</td>
<td>5438</td>
<td>0.10</td>
<td>0.894</td>
<td>0.933</td>
<td>M00510</td>
</tr>
<tr>
<td>USF1</td>
<td>5081</td>
<td>0.10</td>
<td>0.950</td>
<td>1.133</td>
<td>MA0093&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>Oct1</td>
<td>4769</td>
<td>0.10</td>
<td>0.943</td>
<td>1.500</td>
<td>M00342</td>
</tr>
<tr>
<td>Nrf1</td>
<td>4035</td>
<td>0.10</td>
<td>0.900</td>
<td>0.600</td>
<td>M00652</td>
</tr>
<tr>
<td>Myf</td>
<td>3808</td>
<td>0.10</td>
<td>0.858</td>
<td>0.967</td>
<td>MA0055&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>Elk-1</td>
<td>3130</td>
<td>0.10</td>
<td>0.811</td>
<td>0.667</td>
<td>M00225</td>
</tr>
<tr>
<td>ATF3</td>
<td>3106</td>
<td>0.08</td>
<td>0.806</td>
<td>0.500</td>
<td>M00513</td>
</tr>
<tr>
<td>ELK4</td>
<td>2777</td>
<td>0.10</td>
<td>0.903</td>
<td>0.867</td>
<td>MA0076&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>RPS8</td>
<td>2773</td>
<td>0.10</td>
<td>0.907</td>
<td>1.267</td>
<td>M00332</td>
</tr>
<tr>
<td>HSF1</td>
<td>2684</td>
<td>0.10</td>
<td>0.892</td>
<td>0.500</td>
<td>M01023</td>
</tr>
<tr>
<td>MAX</td>
<td>2623</td>
<td>0.10</td>
<td>0.924</td>
<td>1.033</td>
<td>MA0058&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>CREB</td>
<td>2482</td>
<td>0.10</td>
<td>0.825</td>
<td>0.833</td>
<td>M00917</td>
</tr>
<tr>
<td>CRE-BP1</td>
<td>2396</td>
<td>0.10</td>
<td>0.939</td>
<td>0.667</td>
<td>M0041</td>
</tr>
<tr>
<td>LIM4_M2</td>
<td>2176</td>
<td>0.09</td>
<td>0.790</td>
<td>0.500</td>
<td>LIM4&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>c-Myc</td>
<td>2082</td>
<td>0.10</td>
<td>0.838</td>
<td>0.833</td>
<td>M00615</td>
</tr>
<tr>
<td>CREB/ATF</td>
<td>1931</td>
<td>0.10</td>
<td>0.959</td>
<td>0.933</td>
<td>M00981</td>
</tr>
<tr>
<td>NF-Y</td>
<td>1913</td>
<td>0.09</td>
<td>0.804</td>
<td>1.167</td>
<td>M00287</td>
</tr>
<tr>
<td>Stra13</td>
<td>1773</td>
<td>0.10</td>
<td>0.868</td>
<td>0.767</td>
<td>M00985</td>
</tr>
<tr>
<td>UF1H3B</td>
<td>1750</td>
<td>0.10</td>
<td>0.894</td>
<td>0.567</td>
<td>M01068</td>
</tr>
<tr>
<td>ATF</td>
<td>1610</td>
<td>0.10</td>
<td>0.900</td>
<td>0.800</td>
<td>M00338</td>
</tr>
<tr>
<td>POU6F1</td>
<td>1547</td>
<td>0.10</td>
<td>0.870</td>
<td>1.000</td>
<td>M00465</td>
</tr>
<tr>
<td>AP-4</td>
<td>1537</td>
<td>0.10</td>
<td>0.968</td>
<td>1.800</td>
<td>M01176</td>
</tr>
<tr>
<td>GABPA</td>
<td>1426</td>
<td>0.10</td>
<td>0.936</td>
<td>0.933</td>
<td>MA0062&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>NRSF</td>
<td>1360</td>
<td>0.09</td>
<td>0.797</td>
<td>0.500</td>
<td>LIM4&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>ERR</td>
<td>1276</td>
<td>0.08</td>
<td>0.976</td>
<td>0.500</td>
<td>M00511</td>
</tr>
<tr>
<td>Tal-1beta</td>
<td>1215</td>
<td>0.10</td>
<td>0.850</td>
<td>1.133</td>
<td>M00700</td>
</tr>
<tr>
<td>TGIF</td>
<td>1160</td>
<td>0.10</td>
<td>0.956</td>
<td>1.700</td>
<td>M00418</td>
</tr>
<tr>
<td>Stat1</td>
<td>1146</td>
<td>0.10</td>
<td>0.893</td>
<td>0.533</td>
<td>MA0088&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>USF</td>
<td>1034</td>
<td>0.10</td>
<td>0.912</td>
<td>0.933</td>
<td>M00121</td>
</tr>
<tr>
<td>c-Ets-1</td>
<td>941</td>
<td>0.09</td>
<td>0.970</td>
<td>1.467</td>
<td>M00332</td>
</tr>
<tr>
<td>MEF-1</td>
<td>880</td>
<td>0.10</td>
<td>0.850</td>
<td>1.000</td>
<td>M00279</td>
</tr>
<tr>
<td>YY1</td>
<td>859</td>
<td>0.10</td>
<td>0.920</td>
<td>1.133</td>
<td>M01035</td>
</tr>
<tr>
<td>MEF-2</td>
<td>744</td>
<td>0.10</td>
<td>0.767</td>
<td>1.067</td>
<td>M00231</td>
</tr>
<tr>
<td>NF-E2</td>
<td>700</td>
<td>0.10</td>
<td>0.988</td>
<td>0.833</td>
<td>M00337</td>
</tr>
<tr>
<td>TAL1</td>
<td>651</td>
<td>0.10</td>
<td>0.944</td>
<td>1.567</td>
<td>MA0091&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>GFH1B</td>
<td>622</td>
<td>0.10</td>
<td>0.945</td>
<td>0.933</td>
<td>M01038</td>
</tr>
<tr>
<td>HNF-6</td>
<td>535</td>
<td>0.10</td>
<td>0.992</td>
<td>0.867</td>
<td>M00639</td>
</tr>
<tr>
<td>PPARg</td>
<td>523</td>
<td>0.10</td>
<td>0.830</td>
<td>0.933</td>
<td>MA0065&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>MEIS1A</td>
<td>508</td>
<td>0.10</td>
<td>0.895</td>
<td>1.133</td>
<td>M00420</td>
</tr>
<tr>
<td>NrZ1</td>
<td>435</td>
<td>0.10</td>
<td>0.934</td>
<td>1.367</td>
<td>M00221</td>
</tr>
<tr>
<td>HNF4</td>
<td>446</td>
<td>0.09</td>
<td>0.827</td>
<td>1.500</td>
<td>M01031</td>
</tr>
<tr>
<td>HSF</td>
<td>399</td>
<td>0.10</td>
<td>0.978</td>
<td>0.500</td>
<td>M00641</td>
</tr>
<tr>
<td>NR2F1</td>
<td>387</td>
<td>0.10</td>
<td>0.914</td>
<td>1.333</td>
<td>MA0017&lt;sup&gt;a&lt;/sup&gt;</td>
</tr>
<tr>
<td>GCNF</td>
<td>371</td>
<td>0.10</td>
<td>0.891</td>
<td>0.500</td>
<td>M00526</td>
</tr>
</tbody>
</table>

Table 2.2: Top 50 motifs with FDR <0.1 ranked by the number of sites. Most of these motifs are from the TRANSFAC database.<sup>a</sup> From JASPAR database. <sup>b</sup> From Xie et al. 2007
Chapter 3

MotifMap: Updated Modular Redesign for High-Performance Computing

3.1 Introduction

A central challenge of biology is to map and understand gene regulation on a genome-wide scale. For any given genome, only a small fraction of the regulatory elements embedded in the DNA sequence have been characterized, and there is great interest in developing computational methods to systematically map all these elements and understand their relationships. Such computational efforts, however, are significantly hindered by the overwhelming size of non-coding regions and the statistical variability and complex spatial organizations of regulatory elements and interactions, especially in mammalian species.

While many gene-specific, condition-specific, and factor-specific resources for motif binding sites exist [198, 54, 57, 104], it is perhaps surprising that genome-wide systematic catalogs
of binding sites for most species do not. Past efforts have focused primarily on the yeast and fly genomes and with severe restrictions, for instance in terms of data (e.g. ChIP-seq only) or genomic regions (e.g. promoter only). The prototype MotifMap system [194] used an improved comparative genomics approach to provide one of the first genome-wide maps for the human genome and test its accuracy. This system, however, has several limitations including the direct use of coarse genome alignments for searching for binding sites leading to missed and incorrectly scored sites, and the unavailability of maps for other model species. Furthermore, while the available lists of transcription factors are not exhaustive, new information about transcription factors and regulatory interactions is continuously being produced and thus such maps must be periodically updated.

Here we describe improvements to the prototype methods that are used with a new whole-genome alignment and an expanded list of transcription factors to create a new, more comprehensive, map for the human genome. Furthermore, we apply the updated methodology to the genomes of other model organisms for which alignments and estimated phylogenetic trees are available, creating genome-wide maps for the yeast, worm, fly and mouse genomes.

At its core, MotifMap uses data from transcription factor binding motif databases, specifically JASPAR [144] and TRANSFAC [127], to search a reference genome for binding sites and produce three scores at each site. For yeast and fly, we have supplemented the matrices available from JASPAR and TRANSFAC with those available from a number of publications (see Additional file 1 for full list of sources for each species). The first score is the Normalized Log-Odds (NLOD) score derived from the position weight matrix of the corresponding transcription factor. The second score is the Bayesian Branch Length score (BBLS) to measure the degree of evolutionary conservation. Functional elements, such as those playing a regulatory role, often evolve more slowly than neutral sequences and can be detected by their higher level of conservation. MotifMap uses publicly available whole genome alignments and the corresponding phylogenetic trees to leverage the power of comparative genomics in order
to eliminate false positive hits. The third score is the False Discovery Rate (FDR) estimated by using Monte Carlo methods. The three scores at each site are used, in combination with other filters, to generate genome-wide maps.

The quality of the maps is assessed and compared to our previous results [194] as well as other methods [168, 143] in various ways, including comparison to experimental data, such as high-throughput ChIP-seq data. The maps provide a foundation for inferring regulatory networks and can be integrated with a variety of other heterogeneous and autonomous data sources.

3.2 Methods

3.2.1 Normalized log-odd score (NLOD)

Binding sites for each transcription factor are identified by scanning the genome sequence with a position weight matrix. We transform each original weight matrix into a log-odds matrix to account for the background frequency of the nucleotides across the genome. The log-odds score of a sequence is computed as:

\[
\text{LOD}(S) = \sum_{j=1}^{|S|} f(x)
\]

where

\[
f(x) = \begin{cases} 
\log_2(x) & \text{if } x > e_b2^c \\
\frac{x}{2^c \log(2)e} + c & \text{if } x \leq e_b2^c 
\end{cases}
\]
where \( x = \frac{q_{i,j}}{b_i} \), the value \( q_{i,j} \) from the position weight matrix is the probability of observing nucleotide \( i \) (\{A,C,G,T\}) at position \( j \) in a sequence \( S \) of length \( |S| \), and \( b_i \) is the probability of observing nucleotide \( i \) in the entire genome. Since the log function decreases rapidly around zero, frequencies near zero will be converted to large negative numbers in the scoring matrix. Traditionally, pseudocounts are added to the frequency matrices to avoid this, often decided heuristically for each matrix. An alternative approach lower bounds the values of each scoring matrix directly by replacing the log function around zero (when \( \frac{q_{i,j}}{b_i} \) is low) with a linear one that is continuous and smooth. In this paper, we use \( c = -3 \).

The motif matching score is scaled to fall between 0 and 1 to yield the normalized log-odds score:

\[
\text{NLOD}(x) = \frac{\text{LOD}(x) - y_{\min}}{y_{\max} - y_{\min}}
\]

where \( y_{\max} \) and \( y_{\min} \) are the maximum and minimum LOD scores that the matrix can achieve by using the most likely or least likely nucleotide at each position. A z-score is also derived from the NLOD score by estimating the mean and variance of the score of random sequences across the genome. For mammalian species, we use a z-score threshold of 4.27, corresponding to a \( p \)-value of 0.00001, to find a list of initial candidate sites across the reference genome. For yeast, fly, and worm, we use a lower threshold corresponding to a z-score between 2.57 and 3.72, or a \( p \)-value between 0.005 and 0.0001. Finally, we restrict the total number of binding sites by ordering the sites for each motif individually by their z-score, and keeping sites with a z-score at least as high as the \( k^{th} \) site. For our purposes, \( k = 100,000 \), as was done in the prototype version.

### 3.2.2 Bayesian branch length score (BBLS)

Many previous methods have shown that evolutionary conservation can be used to identify transcription factor binding sites [45, 39, 190]. An innovative aspect of the MotifMap system
is how the degree of evolutionary conservation is assessed using the Bayesian Branch Length Score (BBLS) [194], which itself is an improvement over a previous score, the Branch Length Score (BLS) [171, 192]. More precisely, given a multiple alignment of $N$ species and their evolutionary tree, a transcription factor motif, and the genome coordinates of a candidate binding site, let $\sigma_i = 0$ or 1 denote the presence or absence of the motif at the aligned location in the corresponding species $i$. The BLS is simply the total length of the branches associated with the most recent common ancestor of all the species for which $\sigma_i$ is set to 1. However, in reality $\sigma_i$ is not a binary variable but rather comes with a probability $p_i$ measuring the degree of confidence in whether the corresponding motif is present or not in species $i$ at the corresponding location. Given a set of $N$ aligned species, the BBLS takes into account this uncertainty by computing the expected BLS in the form:

$$BBLS = E(BLS) = \sum_{\sigma_1, \ldots, \sigma_N} P(\sigma_1, \ldots, \sigma_N)BLS(\sigma_1, \ldots \sigma_N)$$

where

$$r_i = \begin{cases} p_i & \text{if } \sigma_i = 1 \\ 1 - p_i & \text{if } \sigma_i = 0 \end{cases}$$

The values of $p_i$ for the leaves of the tree are derived using the NLOD score described above. If the corresponding $z$-score is too low, $p_i$ is set to 0. An efficient dynamic programming approach, avoiding the addition of an exponential number of terms (Equation 3.1), has been
derived [194], and a corresponding software implementation is available (see below).

### 3.2.3 False discovery rate (FDR)

For every motif weight matrix, we generate control matrices by randomly shuffling the columns of the motif weight matrix. The shuffling is repeated up to 10,000 times so as to produce up to 10 control matrices. The shuffled matrices must be sufficiently different from the original one to be used as control matrices. In practice, we use a cutoff of 0.35 on the similarity measure computed by first taking the average correlation between columns over pairs of windows of length 8 in the original and permuted motif, then taking the maximum of these correlations over all pairs of windows, and then normalizing by the length of the motif. Only binding matrices are retained that: (1) are at least eight nucleotides long; and (2) can produce at least three sufficiently distant shuffled versions for the Monte Carlo FDR procedure. In addition, for mammalian species, each shuffled matrix is restricted to have the same CG-dinucleotide frequency as the original matrix. The same motif searching procedure is used with each control matrix. The false discovery rate is computed as the median number of sites found using the shuffled matrices divided by the number of sites found for the real matrix at a particular (NLOD,BBLS) score combination or higher.

### 3.2.4 Sequence alignments and modular design

The prototype version of MotifMap searched the low-resolution multiple alignment files obtained from the UCSC Genome Browser [149] directly. As a result, possible alignments of a motif could be missed in other species, for example in poorly aligned regions with many gaps. To address this problem, the overall methodology used to search for aligned transcription factor binding sites has been considerably improved (Figure 3.1).
The new approach searches instead the reference genome directly and uses the low-resolution alignments only as a seed to identify regions in other species aligning to the motif in the reference species. An expanded sequence including 15 base pairs on each side of each binding site in the reference species is used to identify aligned regions in the other genomes. This expanded sequence helps compensate for the low-resolution nature of the whole genome alignments [20]. Furthermore, instead of using the aligned regions directly, which may be too short or contain many gaps, we find all the alignment blocks overlapping the expanded sequence. Due to the nature of the algorithm used to build the multiple alignments, the sequences in different aligning blocks for any single species may be very far apart from each other on the chromosome, or even on completely separate chromosomes. As a result, we only merge blocks that are within 30 base pairs of each other. This operation yields a set of blocks of aligning regions; each block contains sequences from other species aligned to the binding site. For each species, we find the motif sequence with the highest normalized log-odds score across all blocks. Finally, the scores corresponding to the selected sequence from each species are used for BBLS scoring.

In practice, requiring a minimum number of species to be aligned to the reference sequence at each binding site improves performance. The default requirement, used for instance in the case of the yeast map, is set to at least one other species (i.e. BBLS > 0). For the human map, in the public version of MotifMap, binding sites are required to be conserved across at least four non-primate species. This also enables a fair comparison to the prototype version that used the same requirement.

Because the new modular design of MotifMap is not dependent on searching the UCSC coarse multiple alignment files directly, it enables one to also use other alignments if necessary, such as high resolution alignments of the upstream regions of known homologous or orthologous genes, even when these are not in the UCSC format (e.g. the MAF format produced by the multiz alignment software), or to focus the search on any subset of the genome. To
avoid bias from binding sites that occur in regions that are conserved for being part of a translated portion of a gene and are not necessarily under positive selection because of their importance for regulatory control, we exclude exonic regions of the genome from the default public version of MotifMap. Likewise, we exclude repetitive regions.

### 3.2.5 Redundancy Filter

A transcription factor is often annotated with multiple binding matrices in JASPAR and TRANSFAC. For example, each matrix may represent a specific isoform of the factor dependent on the biological context (e.g. cell type or experimental condition). However, in order to estimate a total number of unique potential binding sites, a given site can be counted only once for a given transcription factor, even when this factor has multiple binding matrices. For this purpose, we first perform the genome-wide search independently for each matrix, and then group overlapping binding sites. We choose a representative for each transcription factor in that group by picking the site with the highest BBLS score. The final result is a non-overlapping, non-redundant list of binding sites for each transcription factor.

### 3.3 Results

#### 3.3.1 New MotifMaps

Each MotifMap is generated automatically via a pipeline running on a parallel computer cluster. Comprehensive maps for human, mouse, fly, worm, and yeast have been generated and new maps can be produced automatically. Details about the genomes, alignments, and matrices used in each MotifMap can be seen in Table 3.1. The raw data for the total number of binding sites across the genomes ranges from hundreds of thousands for yeast, worm, and
fly to millions for mouse and human. Table 3.2 summarizes the number of transcription factors, matrices, and binding sites for each available species after all filtering steps have been applied. For the human MotifMap, we predict 519,108 binding sites for 570 matrices, nearly a 5-fold increase over the number of sites and matrices in the prototype version, while maintaining a low FDR of 0.1 or less.

3.3.2 Evaluation of new methods using experimental data

We first compare the updated methodology to the prototype version using data on well-studied transcription factors and experimentally-determined binding sites using high-throughput methods, such as ChIP-seq. While ChIP-seq and related methods are not perfect, they still provide the best available experimental approximations to genome-wide maps of binding sites. While the prototype map used 17 species, a larger number of genomes and genome alignments has become available since its publication. Thus, for comparison purposes, we run the new methodology using both the same tree of 17 species used for the first prototype, as well as an expanded tree containing 32 placental mammals.

Specifically, we consider the same set of highly studied transcription factors (Table 3.3), same motifs, same experimental data [82, 180, 152, 200, 114, 98], and multiple alignments as in Xie et al. [194], to compute the area under the Receiver Operating Characteristic (ROC) curves (AUC) using the updated methodology. For all motifs, we see an improvement of the AUC in the range of 0-5%. For P53, CTCF, and NRSE, we observe an increase in the AUC with a decrease in the number of sites found. For NFKB and STAT1, we observe a modest increase in the number of sites along with an increase in the AUC. We also observe further modest improvements for a few of these transcription factors when the number of species in the multiple alignments is increased from 17 to 32 placental mammals (see the UCSC Genome Browser website for details on the species in each alignment).
We also use ChIP-seq data available for 35 mouse transcription factors obtained from the TRANSFAC suite to further assess the performance of the MotifMap pipeline and compare it to other methods. We evaluate the performance of the BBLS scoring scheme to recover known binding sites identified by ChIP-seq against four other scores: BLS [171, 192], NLOD (as described in this work), PhastCons [168], and PhyloP [143]. Each score is individually used to rank the binding sites identified by MotifMap. We calculate the number of true and false positive sites identified in the ChIP-seq data to compute the AUC, as in Xie et. al. [194]. Table 3.4 summarizes the results for the performance of the MotifMap pipeline in recovering the sites identified by the ChIP-seq methods by reporting the results for the 20 top transcription factors with the largest AUC values. For these 20 transcription factors, we see performances comparable to those seen for the human MotifMap: MotifMap achieves the best AUC result in 16 of them, while relatively small differences (3% or less) are seen for the remaining five, providing further evidence of the overall quality of the MotifMap system and its ability to generalize and identify binding sites in other species.

3.3.3 Localization analysis: binding site location properties

To further assess the quality of the maps, we examine the distribution of the candidate sites relative to the locations of genes across the genome. Using the high confidence data (FDR ≤ 0.1), we find that the majority of sites are within 1Kbp of the transcription start sites (TSS) of known genes across all species. Figure 3.2 shows a plot of the distribution of distance to the closest gene for each binding site for the human genome. This distribution becomes increasingly peaked as one increases the BBLS threshold filter (Figure 3.4). However, we note that we also find high-confidence sites significantly far from known transcription start sites (further than 100Kbp away). These sites would be missed in a promoter-only analysis of transcription factor binding sites. We see similar distributions for mouse, while for smaller genomes (such as yeast and fly) the binding sites are even closer to the transcription start

61
sites. This is expected, since the genomes of these species are more condensed, including shorter promoter and intragenic regions.

### 3.3.4 MotifMap system, web server, and data integration

The MotifMap “system” consists of three main components: (1) a computational pipeline to perform the genome-wide search; (2) a database to store candidate motif binding sites, the scores associated with them, and the relationships to other features; (3) custom code to interface between the database and a web service; and (4) a Flex web application, to display data to users. All steps in the pipeline for identifying and scoring binding sites are performed in parallel using a high performance computer cluster. Along with the locations and scores for each binding site, we compile and store relationships between the binding sites and other genomic features, such as genes (RefSeq [146] and Ensembl [51]) and Gene Ontology (GO) annotations [10]. Some species (fly and yeast) use specific gene annotation resources instead (FlyBase [34] and SGD [145]). The database is currently being expanded as other MotifMaps and new relationships become available. The binding site data and relationships for all available species are publicly available through the MotifMap web site (http://motifmap.igb.uci.edu).

While the prototype MotifMap version had a simple interface to display data, the new web application has been extensively upgraded with multiple features and functionalities to allow users to explore these genome-wide datasets more easily. User can interactively select a model species and one or more transcription factors, visualize the logos of the corresponding motifs, filter the results by various criteria and thresholds (genome location, NLOD/z-score, BBLS, FDR), and retrieve a corresponding list of binding sites, with the distances to the nearest TSS and the corresponding gene annotations. The results can be downloaded in a variety of standard formats (GFF, BED, CSV) or exported directly for
visualization in the UCSC Genome Browser. Furthermore, for each motif binding site, users can view the local multiple alignment and the phylogenetic tree with the corresponding probability scores for each species, as shown in simplified form at the bottom of Figure 3.1. A Python implementation of an efficient algorithm for computing the Bayesian Branch Length Score can also be downloaded from the MotifMap web site.

MotifMap uses an integrative approach combining, for instance, phylogenetic, genomic, and transcription factor data. The resulting maps themselves can in turn be integrated with many other datasets (see Discussion). Two kinds of data that are fully integrated into the MotifMap database and available to the user are GO annotations and SNPs. For instance, for a given GO annotation and the corresponding set of genes, users can retrieve all the nearby candidate binding sites. Likewise, SNPs falling within or near a transcription factor binding site have the potential for influencing the regulation of the corresponding gene [35]. Thus it is useful to be able to list which SNPs in a GWAS (Genome Wide Association Study) or other genotyping study fall within or nearby transcription factor binding sites. Analyses of GWAS data focused primarily on coding regions run the risk of missing important SNPs affecting regulatory regions. The relationship between SNPs and binding sites has been integrated into the MotifMap web application as an additional analysis tool called SNPer, which allows the retrieval of motif binding sites that overlap with SNP sites. The HapMap3 [75] and dbSNP [165] datasets are currently available for use with the mouse and human MotifMap. Users can download the MotifMap results for further integration with specific GWAS or other studies.

3.4 Discussion

The MotifMap approach has allowed us to derive state-of-the-art genome-wide maps of candidate regulatory elements for some of the main model organisms, in particular for mouse
and human. For the worm, the map produced is considerably more primitive because only six transcription factor binding matrices are available in TRANSFAC and JASPAR. However, the availability of the map for this limited set of transcription factors may still be of some use and all the maps will be updated as more binding matrices become available.

Each binding site predicted by MotifMap corresponds in fact to a regulatory hypothesis, thus a single MotifMap can generate from thousands to millions of hypotheses. These hypotheses can be tested and refined in the laboratory, either individually in the case of very specific interactions which can be tested with great precision, or on a larger but less precise scale using high-throughput methods, such as ChIP-seq. These multiple hypotheses can also be further refined and analysed by computational methods using integrative approaches where regulatory hypotheses are simultaneously combined: (1) with each other in the form of regulatory networks; and (2) with other kinds of data. Regulatory hypotheses can be integrated with each other to identify regulatory networks of transcription factors, including regulatory loops and, for instance, hypothesize that transcription factor A regulates transcription factor B, transcription factor B regulates transcription factor C, and transcription factor C regulates transcription factor A. These networks and loops can be thought of as the core regulatory network of a cell. Regulatory hypotheses can also be integrated with many other kinds of data to refine regulatory inferences, as described in the Results section using GO and SNP data and below with other kinds of data. In particular, MotifMap and GO annotations can be used to infer the common functions of a set of genes targeted by a transcription factor or, conversely, to infer the transcription factor that may regulate a set of genes with common GO annotations.

To illustrate these ideas, here we give a simple demonstration of the power of integrating MotifMap and other data to generate regulatory network hypotheses, above the level of an individual regulatory site. For demonstration purposes, we choose two examples. We reconstruct the P53 apoptotic pathway, since it is an important and well-studied pathway
which allows us to assess the quality of the predictions. We also apply the same general ideas to the Gli family of transcription factors to demonstrate the effectiveness of these methods on a relatively less-studied transcription factor and pathway where important regulatory effects remain to be discovered.

### 3.4.1 Mouse P53 apoptotic pathway

We attempt to reconstruct the P53 direct regulatory interactions in the P53 apoptotic pathway in the mouse using data from MotifMap for putative P53 binding sites across the genome. We first compile a list of over 380 unique gene transcripts from the RefSeq database [146] annotated with the Gene Ontology term “Apoptosis” (GO:0006915). We then retrieve predicted P53 binding sites from MotifMap in the promoter region of these genes to generate a regulatory network of P53’s role in apoptosis. The promoter region of a gene is defined as 15Kbp upstream and 3Kbp downstream, which approximately encompasses the region associated with the first intron, from the transcription start site. To evaluate the network generated from MotifMap data, we compare it to the P53 pathway described in the KEGG database [87], which reports 14 genes directly regulated by P53 in the apoptotic pathway (Figure 3.3). Table 3.5 shows the number of known and potentially novel P53 targets predicted as a function of FDR. At a FDR of 0.05, we predict eight target genes from the list of all apoptotic genes, six of which are annotated in KEGG. Searching the literature reveals that the other two target proteins, DDIT4 and PHLDA3, are also known targets of P53 [42, 95] but not annotated in KEGG. At a FDR of 0.25, we predict a total of 71 targets, including 12 of the 14 targets annotated in KEGG; the only exceptions are FAS and TSAP6 (also called STEAP3). FAS is a predicted direct target, but has a slightly higher FDR (0.28). For TSAP6 we find two P53 sites (1784 bp and 4582 bp upstream) with a strong motif matching score; however these sites are not conserved. A novel predicted target is BID, which is annotated in KEGG as a downstream indirect target in the P53 apoptotic
pathway. If we reduce the length of the upstream promoter regions from 15Kbp down to 5Kbp, the same KEGG targets are recovered with the exception of PIDD and SHSA5. A few targets have P53 binding sites downstream of the TSS, in the first intron, and these would not have been recovered with a search focused on promoter regions only. Thus in short the MotifMap system is capable of robustly recovering most of the direct targets of P53 described in KEGG, as well as providing a ranked list of potential new targets, some of which can be confirmed by a literature search.

3.4.2 Mouse Gli targets

Next, we examine the Gli family of transcription factors. Although Gli is a relatively less studied transcription factor, mutations in Gli genes have been associated with multiple developmental disorders and cancers [125]. We started by compiling a list of Gli targets. The KEGG database lists only two annotated targets of Gli1 (Hhip and Ptc1), as well as an autoregulatory loop of Gli1. Gli1 is annotated as a downstream effector of the Sonic hedgehog pathway. In addition Gli1 is known to regulate the Wnt and Bmp signaling pathways, however the specific members of these families are not well defined. Due to the lack of many annotated targets in KEGG, we used the Transcriptional Regulatory Element Database (TRED, http://rulai.cshl.edu/cgi-bin/TRED/tred.cgi?process=home), which contains an additional four annotated Gli family targets. We find Gli binding sites predicted by MotifMap in the promoter region of the seven annotated targets and also many of the Bmp and Wnt proteins. We observe predicted binding sites in the Shh promoter (14,843bp upstream) as well as in the second intron. In addition, we recover the Gli1 autoregulatory loop [182] and regulation of Gli3 by Gli1 [69] (Figure 3.5a). All binding sites for all targets are recovered at an estimated FDR $\leq 0.25$, within 15Kbp upstream and 3Kbp downstream of each gene. Furthermore, we identify a highly conserved binding site (BBLS > 7, perfectly conserved in 27 out of the 30 species in the alignment) near Ptc1. Nkx2-8 and Nkx2-2, both
of which have been reported as targets of Gli family transcription factors [177, 160], have predicted binding sites within 2Kbp upstream of the transcription start site with similar conservation (Figure 3.5b). At a moderate conservation level (BBLS > 3), we identify an additional 21 novel targets, of which Ptch2, Tgfb2, Foxl1, and Runx1 [177, 30, 120, 166] have been annotated as real targets. We also identify Rab34 as a true Gli target at a lower conservation level (BBLS > 2); this threshold includes approximately 100 novel targets.

### 3.4.3 Further integration and challenges

Regulatory networks do not consist only of transcription factors and their direct regulatory interactions, but can include also protein-protein interactions (PPI). Integrating PPI (physical or genetic) data [96, 173] with protein-DNA interactions from MotifMap can yield a more comprehensive view of molecular mechanisms. Integration of this data also allows for the identification and verification of transcription factors that act in complexes. An observation of adjacent binding sites for the factors involved in a complex would be strengthened by data supporting physical interactions between the factors. Adjacent factors need not to physically interact in order to influence transcription, and MotifMap data can be used to identify modules of transcription factors that commonly occur near co-regulated genes. Likewise, incorporating information about RNA elements involved in gene regulation [63] is also important to derive a more accurate and complete global picture.

As so far described, MotifMap provides a static view of potential transcription-factor/DNA interactions. Since transcription factor regulation of most genes does not occur ubiquitously or constantly across all cells in an organism, DNA microarrays and high-throughput sequencing of transcripts (RNA-seq) provide another important source of information about the cell-specific, tissue-specific, or condition-specific expression of genes. Thus MotifMap can be integrated with gene expression data, such as the Gene Expression Omnibus (GEO).
data [14]. This integration provides additional information about, for instance, the average sign of a particular interaction across many experiments, or about the specific portion of the total potential regulatory network that is activated in a given condition.

An important challenge ahead lies in better understanding the role of epigenetics in the regulation of gene transcription. An interesting source of data for further integration with MotifMap comes from the ENCODE project [24] providing the locations of epigenetic signatures, such as histone tail methylations or acetylations, across the human genome for a large number of cell lines. Combinations of these markers can identify transcription factor binding sites that are specific to a particular cell line; for example, the presence of H3K4Me1 and absence of H3K4Me3 denotes enhancer regions. This integration induces regulatory sub-networks, potentially describing important interactions needed for a particular cell type to function properly.

Another considerable challenge is the role of chromatin and 3D structure in gene regulation. New high-throughput techniques like chromosome conformation capture-on-chip (4C), Hi–C and Chromatin Interaction Analysis using Paired-End Tag sequencing (ChIA-PET) allow the detection of long range or inter-chromosomal interactions of DNA [170, 113, 53]. This provides the ability to detect regulatory elements that may be distal to the gene they regulate linearly, but are brought close together in 3-dimensional space. For instance, a recent study used 4C to investigate the properties and dynamics of the genomic loci that are in contact with glucocorticoid receptor (GR) responsive loci [60]. Incorporating this kind of data into MotifMap could provide further evidence of these distant regulatory interactions and improve our ability to infer regulatory mechanisms and networks.

Many other data, such as scientific literature, or information about diseases and drugs, are also being integrated in house with MotifMap. Each data comes with its own noise and limitations and it is the combination of diverse lines of evidence that has the power to solidify inferences and rank hypotheses in a relevant way. This integration process is not new, of
course, and in essence is at the root of IBM’s Watson system for the game of Jeopardy [49]. This integration process is ongoing and raises computational challenges both in its execution and in what can be served publicly given a limited amount of computational resources.

Finally, another potential computational challenge for systems like MotifMap is the dynamic use of evolutionary trees and comparative genomics. The current version of MotifMap builds a genome-wide map, assessing conservation with a single static tree for each species. But clearly not all regulatory elements are conserved, and even when they are, the optimal tree for assessing their degree of conservation may vary with each transcription factor and each biological question. Thus studying how to dynamically assess conservation, including its weaker forms [161, 101], and how to discover regulatory elements that are poorly conserved remain important questions for further investigations.

### 3.5 Conclusion

The MotifMap system aims to provide comprehensive genome-wide map of regulatory elements for each organism. Since experimental data on gene expression obtained with DNA microarray or high-throughput sequencing methods is inherently biased (to a specific condition, cell type, etc.), a resource that catalogs transcription factor binding sites across the entire genome in an unbiased fashion is valuable. We have created the first such comprehensive maps of candidate regulatory motifs across the yeast, fly, worm, mouse, and human genomes. The updated methodology has improved the detection of experimentally validated motif binding sites and, together with integration with other data, the generation of regulatory networks and hypotheses. Overlaying and integrating information from multiple sources, well beyond transcription factor binding motifs and genomic DNA sequences, is key to building better maps and ultimately to understanding gene regulation on a genome-wide scale.
3.6 Figures

3.6.1 Figure 3.1 - Explanation of methods

Diagram of updated methods. The reference genome is searched to find candidate sites and compute NLOD scores. Using the sequence around each site, overlapping aligned blocks of sequence are extracted from the multiple alignment. Nearby blocks are merged and then the best motif binding site in each species is found. The scores of the best motif sequences for each species are used to compute the BBLS score.

![Diagram showing the explanation of methods](image_url)
3.6.2 Figure 3.2 - Distribution of distance to closest gene for binding sites

Distribution of the distance to the closest gene (Transcription Start Site or TSS) for high confidence human motif binding sites.

![Distribution of distance to closest gene](image)

Figure 3.2: Distribution of the distance to the closest gene (Transcription Start Site or TSS) for high confidence human motif binding sites.

3.6.3 Figure 3.3 a & b - Distribution regulatory elements by conservation

Empirical distribution of distances of human transcription factor binding sites to the closest (≤ 10Kbp) RefSeq gene transcription start site (TSS). The sites are grouped into quartiles according to the BBLS score; each group has one quarter of the total binding sites. The BBLS range for each quartile is given at the top of each plot. As the BBLS conservation score increases, we observe a larger proportion of binding sites close to the TSS of the closest gene.
Figure 3.3: Empirical distribution of distances of human transcription factor binding sites to the closest ($\leq$ 10Kbp) RefSeq gene transcription start site (TSS). The sites are grouped into quartiles according to the BBLS score; each group has one quarter of the total binding sites. The BBLS range for each quartile is given at the top of each plot. As the BBLS conservation score increases, we observe a larger proportion of binding sites close to the TSS of the closest gene.

### 3.6.4 Figure 3.4 - Known apoptotic targets of P53

Known apoptotic genes from the KEGG pathway database and the literature for P53. Genes in light green are annotated in KEGG. Orange dots indicate direct targets recovered by MotifMap. DDIT4 and PHLDA3 are examples of additional direct targets identified by MotifMap with FDR < 0.05 which have been reported in the literature but are not present in KEGG.

### 3.6.5 Figure 3.5 a & b - Targets of Gli

Network showing the known Gli targets in mouse. All direct targets were recovered by MotifMap, including the autoregulation loop of Gli1. Nkx2-2, Nkx2-8 and Ptch2 are examples
Figure 3.4: Known apoptotic genes from the KEGG pathway database and the literature for P53. Genes in light green are annotated in KEGG. Orange dots indicate direct targets recovered by MotifMap. DDIT4 and PHLDA3 are examples of additional direct targets identified by MotifMap with FDR < 0.05 which have been reported in the literature but are not present in KEGG.

of additional direct targets identified by MotifMap with binding site conserved in more than 25 out of the 30 species in the genome alignment.

3.7 Tables

3.7.1 Table 3.1 - Multiple alignment information.

Number of species in multiple alignments for each reference species, and the number of original matrices for each species. Comprehensive maps exist for yeast, worm, fly, mouse, and human (both for a 17-species and 32-species alignment). See the UCSC Genome Browser website for more information on the alignments used and the species they contain. †trees used are truncated to remove the most distant species.
Figure 3.5: (a) Network showing the known Gli targets in mouse. All direct targets were recovered by MotifMap, including the autoregulation loop of Gli1. Nkx2-2, Nkx2-8 and Ptch2 are examples of additional direct targets identified by MotifMap with binding site conserved in more than 25 out of the 30 species in the genome alignment. (b) Motif alignment for a highly conserved Gli1 binding site 1365 bp upstream of the Nkx2-8 transcription start site.

3.7.2 Table 3.2 - Non-redundant transcription factor binding sites.

Number of non-redundant transcription factors and binding sites across the genome. See text for definition of “non-redundant”.

3.7.3 Table 3.3 - Prototype vs Updated MotifMap pipelines.

Area under the ROC curve (AUC) and number of sites found for the prototype MotifMap pipeline versus the updated MotifMap pipeline (performed with the original 17-species alignment and a newer 32-species alignment). The best performing method for each motif is shown.
Species | Build | Alignment | # species in alignment | # of matrices
---|---|---|---|---
Saccharomyces cerevisiae | sacCer2 | multiz7way | 7 | 507
Caenorhabditis elegans | ce6 | multiz6way | 6 | 6
Drosophila melanogaster | dm3 | multiz15way | 12 (flies only)† | 262
Mus musculus | mm9 | multiz30way | 30 | 830
Homo sapiens | hg18 | multiz28way | 17 (placentals only)† | 837
Homo sapiens | hg19 | multiz46way | 32 (placentals only)† | 837

Table 3.1: Multiple alignment information

Species | # Transcription Factors | # Matrices | # Sites | # Sites FDR ≤ 0.1
---|---|---|---|---
Yeast | 161 | 147 | 115,387 | 1,577
Worm | 6 | 6 | 88,895 | 69
Fly | 94 | 66 | 191,655 | 36,091
Mouse | 473 | 575 | 6,617,325 | 740,685
Human (hg18) | 468 | 570 | 2,554,732 | 519,108
Human (hg19) | 468 | 530 | 1,410,309 | 457,198

Table 3.2: Non-redundant transcription factor binding sites

<table>
<thead>
<tr>
<th></th>
<th>NFKB</th>
<th>MYC</th>
<th>P53</th>
<th>STAT1</th>
<th>CTCF</th>
<th>NRSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prototype, 17 species</td>
<td>0.722</td>
<td>0.683</td>
<td>0.861</td>
<td>0.606</td>
<td>0.814</td>
<td>0.941</td>
</tr>
<tr>
<td>Update, 17 species</td>
<td>0.797</td>
<td>0.765</td>
<td>0.902</td>
<td>0.780</td>
<td>0.887</td>
<td>0.950</td>
</tr>
<tr>
<td>Update, 32 species</td>
<td>0.786</td>
<td>0.812</td>
<td>0.896</td>
<td>0.820</td>
<td>0.903</td>
<td>0.951</td>
</tr>
</tbody>
</table>

Table 3.3: Performance comparison of the prototype and updated MotifMap pipelines

3.7.4 Table 3.4 - Performance of the mouse MotifMap

Area under the ROC curve (AUC) for predicting transcription factor binding sites identified by ChIP-seq experiments in mouse. Each column is associated with a different method for scoring and ranking the putative sites identified by MotifMap, from which ROC curves and AUCs are computed. The best performing method for each motif is shown in bold.
<table>
<thead>
<tr>
<th>Name</th>
<th>BBLs</th>
<th>BLS</th>
<th>NLOD</th>
<th>PhastCons</th>
<th>PhyloP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTCF</td>
<td>0.901</td>
<td>0.838</td>
<td>0.893</td>
<td>0.798</td>
<td>0.754</td>
</tr>
<tr>
<td>Myc:Max</td>
<td>0.831</td>
<td>0.826</td>
<td>0.731</td>
<td>0.773</td>
<td>0.690</td>
</tr>
<tr>
<td>Zfp281</td>
<td>0.827</td>
<td>0.820</td>
<td>0.611</td>
<td>0.691</td>
<td>0.679</td>
</tr>
<tr>
<td>LBP9 (Tcfc211)</td>
<td>0.809</td>
<td>0.500</td>
<td>0.754</td>
<td>0.800</td>
<td>0.668</td>
</tr>
<tr>
<td>c-Myc</td>
<td>0.778</td>
<td>0.771</td>
<td>0.734</td>
<td>0.758</td>
<td>0.710</td>
</tr>
<tr>
<td>GLI3</td>
<td>0.772</td>
<td>0.771</td>
<td>0.619</td>
<td>0.806</td>
<td>0.659</td>
</tr>
<tr>
<td>GLI1</td>
<td>0.770</td>
<td>0.728</td>
<td>0.727</td>
<td>0.704</td>
<td>0.689</td>
</tr>
<tr>
<td>E2F</td>
<td>0.760</td>
<td>0.737</td>
<td>0.632</td>
<td>0.737</td>
<td>0.667</td>
</tr>
<tr>
<td>Myc</td>
<td>0.760</td>
<td>0.699</td>
<td>0.540</td>
<td>0.703</td>
<td>0.718</td>
</tr>
<tr>
<td>IPF1</td>
<td>0.757</td>
<td>0.765</td>
<td>0.500</td>
<td>0.696</td>
<td>0.689</td>
</tr>
<tr>
<td>RNF96</td>
<td>0.753</td>
<td>0.749</td>
<td>0.609</td>
<td>0.640</td>
<td>0.642</td>
</tr>
<tr>
<td>GKLF (KLF4)</td>
<td>0.740</td>
<td>0.500</td>
<td>0.500</td>
<td>0.695</td>
<td>0.678</td>
</tr>
<tr>
<td>ERR2 (ESRRB)</td>
<td>0.739</td>
<td>0.500</td>
<td>0.516</td>
<td>0.667</td>
<td>0.608</td>
</tr>
<tr>
<td>Zfx</td>
<td>0.733</td>
<td>0.731</td>
<td>0.660</td>
<td>0.677</td>
<td>0.644</td>
</tr>
<tr>
<td>N-Myc</td>
<td>0.730</td>
<td>0.728</td>
<td>0.620</td>
<td>0.690</td>
<td>0.664</td>
</tr>
<tr>
<td>CNOT3</td>
<td>0.683</td>
<td>0.688</td>
<td>0.568</td>
<td>0.614</td>
<td>0.597</td>
</tr>
<tr>
<td>STAT3</td>
<td>0.677</td>
<td>0.634</td>
<td>0.656</td>
<td>0.655</td>
<td>0.614</td>
</tr>
<tr>
<td>PPAR</td>
<td>0.673</td>
<td>0.664</td>
<td>0.642</td>
<td>0.636</td>
<td>0.615</td>
</tr>
<tr>
<td>Dax1</td>
<td>0.668</td>
<td>0.653</td>
<td>0.598</td>
<td>0.612</td>
<td>0.597</td>
</tr>
<tr>
<td>REX1</td>
<td>0.629</td>
<td>0.627</td>
<td>0.596</td>
<td>0.650</td>
<td>0.661</td>
</tr>
</tbody>
</table>

Table 3.4: Performance of the mouse MotifMap

### 3.7.5 Table 3.5 - Mouse P53 apoptotic pathway

Number of known (annotated in KEGG) and potentially novel P53 direct targets predicted at different FDR thresholds.

<table>
<thead>
<tr>
<th>FDR</th>
<th>0</th>
<th>0.05</th>
<th>0.1</th>
<th>0.15</th>
<th>0.2</th>
<th>0.25</th>
</tr>
</thead>
<tbody>
<tr>
<td>KEGG known</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>7</td>
<td>10</td>
<td>12</td>
</tr>
<tr>
<td>Potentially novel</td>
<td>1</td>
<td>2</td>
<td>16</td>
<td>29</td>
<td>50</td>
<td>73</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
<td>8</td>
<td>22</td>
<td>36</td>
<td>60</td>
<td>85</td>
</tr>
</tbody>
</table>

Table 3.5: Mouse P53 apoptotic pathway
Chapter 4

Application of MotifMap: Sirtuin circadian transcriptome

4.1 Introduction

The circadian clock regulates a host of physiological events required for energy balance (Feng and Lazar, 2012[48]; Sahar and Sassone-Corsi, 2012[158]; Wijnen and Young, 2006[185]). These events provide remarkable plasticity for the organism to adapt to surrounding environmental changes, especially given the dynamic input of cellular metabolism on chromatin modifications (Gut and Verdin, 2013[59]; Katada et al., 2012[91]). A functional link between the circadian clock and cellular metabolism was revealed by reports implicating the SIRT1 deacetylase in clock function (Asher et al., 2008[11]; Chang and Guarente, 2013b[21]; Nakahata et al., 2008[137]). Mammalian sirtuins constitute a family of seven NAD+-dependent deacetylases (SIRT1-7) that vary in potency of enzymatic activity and protein targets (Chang and Guarente, 2013a[21]; Hall et al., 2013[61]; Houtkooper et al., 2012[68]). The subcellular localization of the sirtuins varies from cytoplasm, mitochondria, nucleus and nucleolus...
Of the sirtuins, SIRT6 is unique in its constitutive localization to chromatin (Mostoslavsky et al., 2006[136]; Tennen et al., 2010[175]) and its genome-wide occupancy is prominent at transcriptional start sites (TSS) of active genomic loci, that coincides to serine 5 phosphorylated RNA polymerase II binding sites (Ram et al., 2011[147]). SIRT6 has also been reported to be dynamic in its chromatin binding in response to stimuli such as (TNF), resulting in altering the transcriptional landscape of aging and stress-related genes (Kawahara et al., 2011[94]). SIRT6 deacetylates H3 lysine 9 (H3K9) (Kawahara et al., 2009[94]; Michishita et al., 2008[130]) and H3K56 (Michishita et al., 2009[130]; Toiber et al., 2013[176]; Yang et al., 2009[197]) resulting in modulation of gene expression, telomere maintenance and genomic stability (Tennen and Chua, 2011[175]), and the HDAC activity of SIRT6 has been found to be nucleosome-dependent (Gil et al., 2013[55]). Importantly, SIRT6 is also heavily implicated in metabolic regulation as Sirt6-/- mice die at 2-4 weeks of age due to severe accelerated aging and hypoglycemia as a result of altered rates of glycolysis, glucose uptake and mitochondrial respiration (Mostoslavsky et al., 2006[136]; Xiao et al., 2010[189]; Zhong et al., 2010[201]). SIRT6 also controls the acetylation state of PGC-1 in a GCN5-dependent manner that regulates blood glucose levels (Dominy et al., 2010[33]). Liver-specific Sirt6-/- mice develop fatty liver due to altered expression of genes involved in fatty acid beta oxidation and triglyceride synthesis (Kim et al., 2010[97]).

The circadian transcriptome is thought to comprise at least 10% of all transcripts in a given tissue, though genes can gain rhythmicity depending on a tissue-specific permissive environment (Masri and Sassone-Corsi, 2010[124]). Moreover, the potential for a specific gene to become circadian may be related to changes in the metabolic, nutritional and epigenetic state (Eckel-Mahan et al., 2013[37]). A number of studies have revealed the role of chromatin remodeling in providing permissive genomic organization for circadian transcription (Belden et al., 2011[16]; DiTacchio et al., 2011[31]; Doi et al., 2006[32]; Duong et al., 2011[36];
Etchegaray et al., 2003[43]; Katada and Sassone-Corsi, 2010[92]; Koike et al., 2012[103]; Ripperger and Schibler, 2006[150]). We report that SIRT6 defines the circadian oscillation of a distinct group of hepatic genes, different from the ones under SIRT1 control. This partitioning of the circadian genome is achieved by controlling the recruitment to chromatin of the core circadian activators CLOCK:BMAL1 as well as SREBP-1. The sirtuin-dependent partitioning of circadian transcription leads to differential control of hepatic lipid metabolism related to fatty acid-dependent pathways.

4.2 Results

4.2.1 Partitioning of the circadian transcriptome by SIRT6 and SIRT1

Given the unique ability of SIRT6 to function as an HDAC (Kawahara et al., 2009[94]; Michishita et al., 2008[129]) and transcriptional facilitator at chromatin (Kawahara et al., 2011[93]; Ram et al., 2011[147]), we investigated its role in controlling hepatic circadian gene expression and metabolism. DNA microarrays were used to delineate the control of SIRT6 versus SIRT1 on the circadian genome. To do so, we used mice with liver-specific ablation of either Sirt1 or Sirt6 genes and their corresponding WT littermates. Livers were harvested every four hours over the circadian cycle, representing zeitgeber times (ZT) 0, 4, 8, 12, 16 and 20. Groups of genes were selected based on the following criteria: Group 1) Genes that oscillate in WT (SIRT6) liver and whose oscillation is dampened/disrupted in SIRT6-deficient mice (SIRT6 KO). Group 2) Genes that oscillate in SIRT6 KO but not in their corresponding WT littermates. Group 3) Genes that oscillate in WT (SIRT1) liver and whose oscillation is dampened/disrupted in SIRT1-deficient mice (SIRT1 KO). Group 4) Genes that oscillate in SIRT1 KO but not in their corresponding WT littermates. The
group referred to as BOTH includes genes that oscillate similarly in both WT and KO
groups for either SIRT6 or SIRT1 datasets. Oscillating genes were selected based on a 0.01
p-value cutoff. A comparison of both WT strains can be found in Figure 4.1. Of the SIRT6
transcriptome, 691 genes were identified in the WT Group 1, with 779 genes oscillate more
robustly in the SIRT6 KO Group 2, and 506 genes oscillate similarly in both groups (Figure
4.1A). Using the same criteria for the SIRT1 transcriptome, 703 genes oscillate in the WT
Group 3, with 1126 genes oscillate with greater amplitude in the SIRT1 KO Group 4 and
1091 genes oscillate similarly in both groups (Figure 4.1B). This analysis revealed that of the
1,976 rhythmic genes identified in the SIRT6 transcriptome, the expression profile of 1,470
genes was altered by SIRT6 disruption (74%). In addition, of the 2,920 oscillating genes
identified in the SIRT1 experiment, 1,829 genes were changed by SIRT1 disruption (63%).
Thus, SIRT6, in addition to SIRT1, significantly regulates the expression of clock-controlled
genes (CCGs).

Gene ontology (GO) analysis of genes with altered circadian oscillation in SIRT6 versus
SIRT1 transcriptomes revealed some striking differences (Figure 4.1C; 4.1D; Fig. S4.9).
The most highly represented biological processes are transcription, transcriptional regulation
and nuclear processes, enriched in both WT liver groups and SIRT1 KO livers, but
completely absent from the SIRT6 KO livers. In addition to transcription, enrichment in
mitochondrial and intracellular non-membrane bound organelle (GO term describing ribo-
somes, cytoskeleton and chromosomes) was highly enriched. The SIRT6 KO group shared
little homology with WT or SIRT1 KO groups in significantly selected biological pathways.
GO terms enriched in SIRT6 KO were endoplasmic reticulum, Golgi apparatus, protein local-
ization/catabolism, RNA processing and translation (Fig. 4.1C). GO biological pathway
analysis highlighted unique classes of genes represented exclusively in the SIRT6 KO group,
indicating that disruption of hepatic SIRT6 results in altered circadian biological function.

Next we focused on understanding how these two sirtuins differentially regulate distinct
classes of circadian genes. Importantly, there is little overlap between the groups of SIRT1- and SIRT6-dependent circadian genes (160 common genes) (Fig. 4.2A). These are mostly implicated in cytoplasmic and mitochondrial pathways and are linked to metabolic processes and stress response, as described by GO biological pathway analysis (Fig. 4.2A). Thus, SIRT6 and SIRT1 regulate distinct biological classes of circadian genes. For a detailed view of these genes controlled by SIRT6 or SIRT1, refer to CircadiOmics: http://circadiomics.igb.uci.edu (Patel et al., 2012[140]). Analysis of the circadian phase of gene expression reveals a peak in phase of the genes oscillating in SIRT6 KO mice at ZT 16 and ZT 20, differently from the genes significantly expressed in SIRT1 KO mice peaking at ZT 4 and ZT 8 (Fig. 4.2B).

Circadian expression was confirmed for distinct classes of genes based on their rhythmic profile: 1) Genes whose expression profile is unaltered between WT versus SIRT6 KO and SIRT1 KO. Briefly, these genes are involved in transcription and regulation of rhythmic processes, as the bulk of core clock genes are generally resistant to change in expression (Fig. S4.9). 2) Genes whose circadian expression is similarly regulated by SIRT6 and SIRT1. Examples include Nephronectin (Npnt), encoding an extracellular matrix protein, which oscillates in WT liver and is dampened similarly in both SIRT6 KO and SIRT1 KO mice (Fig. 4.2C). Conversely, circadian expression of Dbp is equally increased in amplitude at ZT 8 in both SIRT6 KO and SIRT1 KO animals (Fig. 4.2D). These genes, while responding in opposite manner to the ablation of either sirtuin, belong to the same class of genes similarly regulated by both sirtuins. 3) Genes whose amplitude in oscillation is more robust when either SIRT6 or SIRT1 is ablated. For example, fatty acid synthase (Fasn), 3-hydroxy-3-methyl-glutaryl-CoA reductase (Hmgcr) and lanosterol synthase (Lss) were uniquely regulated by SIRT6, as the amplitude of circadian oscillation was enhanced in SIRT6 KO (Fig. 4.2E). The circadian profiles of these genes were unaltered in SIRT1 KO, as compared to WT (Fig. S4.10A). Conversely, genes with enhanced circadian amplitude exclusively in SIRT1 KO, including regulator of G-protein signaling 16 (Rgs16), serine dehydratase (Sds) and methylenetetrahydrofolate dehydrogenase 1-like (Mthfd1l) are shown (Fig. 4.2F). The profiles of Rgs16, Sds
and Mthfd11 genes are not altered in amplitude between WT and SIRT6 KO mice (Fig. S4.10B). Also, the expression of Sirt6 and Sirt1 is not altered in the SIRT1 KO and SIRT6 KO livers, respectively (Fig. S4.10C). Thus, control of circadian gene expression by SIRT6 and SIRT1 appears to define unique sub-domains of oscillating CCGs that are involved in distinct biological functions.

4.2.2 SIRT6 controls their chromatin recruitment

Since SIRT6 is reported to localize to actively transcribed genomic loci (Ram et al., 2011[147]), we sought to decipher the molecular mechanism by which SIRT6 controls circadian transcription. Fractionated liver extracts that lack SIRT6 result in a drastic increase in BMAL1 association to chromatin, though the total amount of nucleoplasmic BMAL1 was unaltered (Fig. 4.3A and Fig. S4.11). Also Bmal1 circadian expression is not altered in SIRT6 KO or SIRT1 KO, as compared to WT liver (Fig. S4.10D).

Since BMAL1 association at chromatin is enhanced in the absence of SIRT6, we analyzed promoter-specific recruitment of the circadian machinery. Chromatin immunoprecipitation (ChIP) analysis was performed to understand if recruitment of the circadian machinery was altered in the absence of SIRT6 or SIRT1, which would therefore contribute to altered CCG expression observed in our microarray analysis. Circadian BMAL1 recruitment to the Rgs16 and Mthfd11 promoters is unaltered in the absence of SIRT1 (Fig. 4.3B; 4.3C), despite the increased amplitude in gene expression in SIRT1 KO (Fig. 4.2F). Schematic representation of the promoter as well as selective recruitment of BMAL1 to different putative E-Boxes in the Rgs16 and Mthfd11 promoters is shown in Fig. S4.12, illustrating that BMAL1 recruitment is virtually identical in WT and SIRT1 KO livers. In addition to Rgs16 and Mthfd11, BMAL1 recruitment is also unaltered at Dbp and Per1 promoters in WT versus SIRT1 KO, despite the significant changes in circadian gene expression (Fig. S4.13). In contrast, lack of SIRT6
results in a significant increase in circadian BMAL1 occupancy (ZT 4 and ZT 8) at the Dbp promoter (Fig. 4.3D). A detailed schematic of the Dbp promoter illustrating select sites of circadian BMAL1 recruitment as well as Dbp expression profile are shown (Fig. 4.3D). Also, an increase in Ac-H3K9 across all time points is seen (Fig. 4.3E). Additional data shows altered BMAL1 recruitment to Per1 and Amd1 promoters in the absence of SIRT6 (Fig. S4.14). To further address the effect of SIRT6, we used a Dbp-luciferase reporter and found that ectopic expression of SIRT6 results in dose-dependent dampening of CLOCK:BMAL1-driven transcription (Fig. 4.3F), similar to results with SIRT1 (Bellet et al., 2013[17]). Thus, differently from SIRT1, SIRT6 controls circadian function by operating directly at the transcription level by recruiting the clock machinery to chromatin.

Moreover, we sought to confirm that SIRT6 interacts with the circadian transcription complex. SIRT6 physically interacts with CLOCK and BMAL1, individually or together, as shown by co-IP (Fig. 4.4A; 4.4B). Also, SIRT6 does not interact directly with SIRT1 by co-IP (Fig. 4.4B). Furthermore, when CLOCK and BMAL1 are ectopically expressed with SIRT6 alone or in combination with both SIRT6 and SIRT1, the SIRT6 IP complex only interacts with CLOCK and BMAL1, not SIRT1 (Fig. 4.4C). Furthermore, by fractionating WT mouse liver, we reveal that subcellular localization of SIRT6 is predominantly in the nucleus and constitutively at chromatin at all ZTs, while SIRT1 is nuclear but not chromatin-bound (Fig. 4.4D) (Mostoslavsky et al., 2006 [136]; Tennen et al., 2010[174]). Likewise, co-IP experiments from chromatin fractions of HEK-293 cells confirmed that SIRT1 does not reside at chromatin (Fig. 4.4E). Also, the SIRT6-dependent interaction with CLOCK and BMAL1 is found at chromatin (Fig. 4.4E). In addition, sequential co-IP experiments were performed to pull-down the SIRT6- and SIRT1-dependent clock complexes from the same HEK-293 cell lysates. Primary IP against Flag-SIRT1 revealed an interaction with CLOCK and a secondary IP with HA-SIRT6 also revealed an interaction with CLOCK (Fig. 4.4F), in keeping with evidence showing that these two sirtuins independently interact with the clock machinery. Lastly, SIRT1 has been shown to deacetylate BMAL1 at lysine 537 (Hirayama et
While SIRT1 readily deacetylates BMAL1, SIRT6 is not able to do so (Fig. 4.4G), highlighting different mechanisms of action of these two sirtuins that reside in partitioned subcellular clock complexes.

4.2.3 SIRT6 controls SREBP-1-dependent circadian transcription

Based on circadian gene expression profiles altered in SIRT6 KO liver, a number of genes were found to be SREBP targets such as Fasn, Hmgcr and Lss (Fig. 4.4D). MotifMap (Daily et al., 2011[28]; Xie et al., 2009[195]) was used to determine global transcription factor binding site enrichment in promoters with altered expression profiles when SIRT6 was disrupted. SREBP binding sites are highly enriched (137 sites) compared to serum response factor (SRF), peroxisome proliferator-activated receptor gamma (PPAR), forkhead box (FOXO), or E26 transformation-specific (ETS) family motifs (Fig. 4.5A). Next, genes whose expression is disrupted by loss of SIRT6 were compared to published ChIP-sequencing data (Seo et al., 2009[162]; Seo et al., 2011[163]) to determine the extent of SREBP-1 and SREBP-2 gene targets that overlap with SIRT6. In addition to the genes already mentioned, other SREBP targets appear disrupted in SIRT6 KO, including fatty acid elongase family members (Elov1), low density lipoprotein receptor (Ldlr) and acetoacetyl-CoA synthetase (Aacs) genes (which are also represented in MotifMap SREBP hits). As a role for SIRT1 in SREBP signaling cannot be excluded, we compared SREBP-1 target genes (Seo et al., 2009[162]) to SIRT6- and SIRT1-dependent gene targets (Table S4.16). Interestingly, overlapping genes between SIRT6/SREBP-1 targets were enriched in GO terms for fatty acid and lipid metabolism, while SIRT1/SREBP-1 overlapping targets were enriched in lipid and steroid metabolism (Fig. 4.5A), suggesting a partition in biological function in SIRT1- or SIRT6-specific control of SREBP.

As there are no significant changes in SREBP-1c circadian transcript and protein levels
in SIRT6 KO livers (Fig. 4.5B), we carried out ChIP experiments. Strikingly, SREBP-1 circadian recruitment to the Fasn promoter, a known SREBP-responsive gene (Seo et al., 2009[162]), is significantly increased in the absence of SIRT6, as compared to WT (Fig. 4.5C). The increase is prominent at ZT 4, thereby preceding the peak of Fasn transcription at ZT 16, a scenario in keeping with accumulated evidence, especially in a circadian context (Koike et al., 2012[103]; Rey et al., 2011[148]). A schematic of the Fasn promoter illustrating selective SREBP-1 recruitment to the TSS versus negative control regions as well as recruitment to the Hmgcr and Lss promoters are shown (Fig. 4.5C; Fig. S4.15). Also, an increase in Ac-H3K9 levels is present at the Fasn promoter, across most ZTs (Fig. 4.5C). Based on this evidence, it is expected that SREBP-1c contributes to Fasn circadian gene expression. To confirm this, we used livers from WT and SREBP-1c KO mice (Seo et al., 2009[162]) at ZT 4 and ZT 16 and observed a significant dampening of Fasn circadian expression in SREBP-1c KO livers, while Dbp and Rev-Erb circadian expression remains unaltered (Fig. 4.5D). Thus, SIRT6 appears to define a class of genes whose amplitude in oscillation is directed by SREBP-1c. To functionally explore the effects of SIRT6 on SREBP-1c mediated transcription, we used a luciferase reporter with either a full-length Fasn promoter (containing the previously described binding site of SREBP-1, referred to as Fas Luc -1594/+65) (Bennett et al., 1995[18]) or a mutant that disrupts SREBP-1 binding (Fas Luc -65 MT) (Joseph et al., 2002[85]). Co-expression with SREBP-1c showed robust Fasn promoter activation that is strongly repressed by increasing amounts of SIRT6 (Fig. 4.5E). Importantly, the Fas-luc -65 MT reporter is not sensitive to SIRT6-mediated repression. The effect is specific as SIRT1 is not able to repress SREBP-1c-driven activation of Fasn (Fig. 4.5E). Thus, SIRT6 is implicated in regulating proper SREBP-1c chromatin recruitment, resulting in circadian transcription of its target genes.
4.2.4 Differential circadian metabolic phenotypes

Metabolomics analysis was used to determine in an unbiased manner the physiological consequences of SIRT6 or SIRT1 disruption along the hepatic circadian cycle. Heat maps highlight oscillating metabolites (JTK_cycle < 0.05 p-value cutoff) in WT livers that were disrupted in SIRT6 KO (left panel) and metabolites that oscillated more robustly in SIRT6 KO livers, as compared to WT (right panel) (Fig. 4.6A). In total, 77 metabolites displayed a genotype-dependent effect in the SIRT6 metabolome and 142 metabolites were dependent on circadian rhythmicity (Fig. 4.6A). We also compared the metabolome profile obtained from the SIRT6 KO mice to livers from SIRT1 KO animals (Fig. 4.6B). Heat maps illustrate the metabolomics data for SIRT1, with oscillating metabolites only in WT livers (left panel) and those found to oscillate robustly in SIRT1 KO (right panel). In the SIRT1 metabolome, 42 metabolites displayed a genotype effect, while 199 show a time of day dependent effect. In total, 85 metabolites robustly oscillate exclusively in SIRT6 KO and 57 metabolites display strong rhythmicity in SIRT1 KO livers.

Metabolites were grouped into biological functional categories (peptides, cofactors and vitamins, lipids, nucleotides, amino acids and carbohydrates) to determine where significant changes occurred in the livers from SIRT6 KO and SIRT1 KO mice versus their WT littermates. The most robust change was seen in lipid-related metabolites in SIRT6 KO livers (Fig. 4.6C). These lipids were heavily related to fatty acid metabolism, including circadian disruption of fatty acid synthesis (medium and long chain fatty acids), storage, cellular membrane lipids and signaling. Using SIRT6 microarray data, genes were run through DAVID to identify possible altered gene expression profiles that match in GO biological function with the metabolomics dataset. A strong correlation in GO biological function was found (Fig. 4.6D), comparing the high-throughput metabolome and transcriptome data when SIRT6 is disrupted (for detailed analysis consult CircadiOomics: http://circadiomics.igb.uci.edu (Patel et al., 2012[140])).
A group of lipids that displayed a strikingly enhanced circadian oscillation with a peak at ZT 16 were membrane lysolipids that are related to cellular synthesis or degradation. Also, genes encoding phospholipases related to lipid signaling displayed altered expression profiles with a paralleled change in eicosanoid metabolite rhythms in response to SIRT6 disruption, indicating that signaling/inflammatory events are SIRT6-regulated. As an example, genes of the phospholipase A2 family (Pla2g2a and Pla2g12a) gained circadian oscillation in the absence of SIRT6, which corresponded with circadian up-regulation of downstream 15-HETE levels. In addition to Fasn, fatty acid elongases and fatty acid transporters were significantly altered in response to SIRT6 disruption. Both carnitine and acetylcarnitine, which are important for beta-oxidation of fatty acids in the mitochondria, gain circadian oscillation and peak at ZT16 in the SIRT6 KO livers. While synthesis and breakdown of fatty acid pathways are related to SIRT6, storage of fatty acids into triglycerides was also altered as evidenced by a gain in oscillation of Agpat6 and glycerol-3-phosphate in the SIRT6 KO mice. These metabolite pathways parallel the altered SREBP transcriptional response and indicate that SIRT6 is required for proper circadian regulation of fatty acid synthesis, storage, breakdown and signaling.

4.3 Discussion

Circadian control of metabolism is thought to be critical for organismal homeostasis (Feng and Lazar, 2012; Sahar and Sassone-Corsi, 2012), and the identification of the molecular players implicated in this control is likely to reveal novel pharmacological strategies. Specifically, SIRT6 regulates hepatic circadian transcription consequently linked to downstream modulation of fatty acid metabolism (Fig. 4.7). SIRT6 interacts with core clock proteins and controls circadian chromatin recruitment of BMAL1 to target promoters. Importantly, SIRT6 also controls SREBP-1 recruitment to target promoters, such as Fasn, and helps main-
tain proper cyclic transcription. In fact, circadian metabolomics analyses reveal that SIRT6 controls lipid metabolism, contributing to the regulation of pathways involved in fatty acid synthesis and beta oxidation, triglyceride storage, signaling and cellular membrane lipids.

One conclusion of this study is that two sirtuins, SIRT6 and SIRT1, control distinct subdomains of the circadian genome through different mechanisms. SIRT6 has been reported to reside at transcriptionally active loci (Ram et al., 2011[147]) and its chromatin association is dynamic in response to stimuli so as to activate specific biological classes of genes (Kawahara et al., 2011[93]). It is tempting to speculate that SIRT6 operates as a transcriptional marker, and given its HDAC function, it may have multiple roles in dictating the boundaries of transcription. As supporting evidence of this notion, we show that SIRT6 contributes to chromatin recruitment of both the circadian machinery as well as SREBP-1. There is no evidence that SIRT1 functions in the same manner. Indeed, SIRT1 is not implicated in chromatin recruitment of the clock machinery (Fig. 4.4; Figs. S4.12-S4.13; (Bellet et al., 2013[17])). SIRT1 appears to modulate circadian transcription purely as a deacetylase, by targeting both histone proteins and non-histone proteins such as BMAL1 and PER2 (Asher et al., 2008[11]; Nakahata et al., 2008[137]). Intriguingly, free fatty acids (FFA) are potent endogenous activators of SIRT6 HDAC activity, but not SIRT1 (Feldman et al., 2013[47]). Thus, endogenous fatty acids could play a role in activating or sensitizing SIRT6, a notion that is particularly appealing as our metabolomics data reveal that fatty acids peak in abundance at the beginning of the light phase (after feeding), which also coincides with peaks in BMAL1 and SREBP1 recruitment to chromatin (Fig. 4.7). In keeping with this idea, the beginning of the light phase must therefore provide a permissive chromatin state, as recruitment of SIRT6-dependent transcription factors occurs primarily at ZT 4 and ZT 8, and in the case of SREBP-1, in advance of the peak in gene expression (Figs. 4.3 and 4.5). Indeed, it has been proposed that an activated state of the circadian landscape exists between ZT 4 and ZT 12, when CLOCK:BMAL1 recruitment occurs and this active state is in advance of nascent transcription (Koike et al., 2012[103]). In virtue of its tight chromatin association,
SIRT6 could thereby operate by sensing changing cellular metabolite levels (NAD+ or fatty acids) and translate this information to control circadian transcription. In this respect SIRT6 would be unique among sirtuins, since SIRT1 (Asher et al., 2008[11]; Chang and Guarente, 2013b[21]; Nakahata et al., 2008[137]) and SIRT3 (Peek et al., 2013[142]) appear to be implicated in circadian regulation uniquely through their enzymatic function.

Aside from transcriptional/translational regulation of the clock, enzymatic activity of a number of factors influences circadian rhythms and could also contribute to SIRT6 function. SIRT6 was recently reported to directly regulate SREBP cleavage to its mature protein form, as a result of SIRT6 localization to the promoters of genes such as SREBP cleavage-activating protein (SCAP) and site-1/2 proteases (S1P and S2P) which are involved in SREBP proteolytic cleavage and transport from the endoplasmic reticulum (ER)/Golgi apparatus (Elhanati et al., 2013[40]). In addition to the circadian regulation of the SREBP lipogenic transcriptional program (Le Martelot et al., 2009[108]), enzymatic regulation at the ER has been described whereby a secondary 12-hour rhythm in the unfolded protein response (UPR) pathway activates SREBP signaling and deregulates lipid metabolism (Cretenet et al., 2010[26]). Though we are looking at 24-hour rhythms, these results highlight a possible connection that could further link SIRT6, SREBP and ER-dependent enzymatic pathways that in time may contribute to the transcriptional role of SIRT6 and the clock described here.

Various mouse models have delineated the role of SREBP transcription factors in the lipogenic program (Horton et al., 2003[67]; Seo et al., 2009[162]; Seo et al., 2011[163]). SREBP-1a and SREBP-1c (the form dominantly expressed in liver) activate both genes involved in fatty acid synthesis and the subsequent incorporation into triglycerides for storage and inclusion into cellular membranes. SREBP-2 is primarily implicated in cholesterol biosynthesis. Based on the results obtained by metabolomics analysis, our data indicate a disruption in fatty acid synthesis, breakdown, incorporation into membrane lipids and storage, with little
disruption in cholesterol related pathways. While we do not exclude the role of other factors such as hepatocyte nuclear factor 4 (HNF-4), liver X receptor (LXR), and peroxisome proliferator-activated receptors (PPARs), our results (Figs. 4.5 and 4.6) point to SREBP-1c as a dominant player implicated in SIRT6 circadian regulation of fatty acid metabolism. Intriguingly, Kanfi et al. reported that when transgenic mice over-expressing SIRT6 were challenged with a high fat diet (HFD), these mice were protected from diet-induced obesity due to SIRT6 repression of PPAR-target genes (Kanfi et al., 2010[88]). In this respect, recent results from our laboratory have shown that HFD regimen in mice reprograms the hepatic circadian transcriptome by inducing de novo oscillations of PPAR-dependent genes (Eckel-Mahan et al., 2013[37]). Given the seemingly ubiquitous localization of SIRT6 at transcriptionally active genomic loci (Kawahara et al., 2011[93]; Ram et al., 2011[147]), and its role as a regulator of circadian transcription as well as SREBP signaling, SIRT6 could also be implicated in diet-induced metabolic regulation of SREBP, PPARs or other factors. The remarkable role of SIRT6 in regulating the circadian transcriptome and defining a landscape for biologically relevant genomic loci places this epigenetic regulator in a central position to control the extensive circadian lipid metabolic program in the liver.

4.4 Experimental Procedures

4.4.1 Animal housing and experimental design

Liver-specific SIRT6 KO mice (Kim et al., 2010[97]) and liver-specific SIRT1 KO mice (Nakahata et al., 2008[137]) were previously described. All experiments were performed in accordance to the Institutional animal care and use committee (IACUC) guidelines at UCI. Animals were housed in a 12 hour light/dark paradigm and fed ad libitum.
4.4.2 DNA microarray analysis

Microarray analysis was performed at the UCI Genomics High throughput Facility and deposited in GEO (GSE57830). For detailed methodology, refer to Supp. Experimental Procedures.

4.4.3 Biological pathway analysis

The Database for Annotation, Visualization and Integrated Discovery (DAVID) (Huang da et al., 2009a[70], b[71]) was used to identify GO terms related to biological process and molecular function related to our gene lists of interest.

4.4.4 Gene expression analysis

Equal amounts of total RNA were reverse-transcribed to cDNA using iScript cDNA synthesis kit (Bio-Rad Laboratories, Hercules, CA), according to manufacturers protocol. cDNA was used for quantitative real-time PCR using iQ SYBR green supermix (Bio-Rad Laboratories). Gene expression was normalized to 18S ribosomal RNA. Primers sequences used for gene expression analysis were designed with Primer3 software (Rozen and Skaletsky, 2000[155]) and are listed in Table S4.17.

4.4.5 Chromatin Immunoprecipitation (ChIP)

Liver samples were quickly minced in PBS containing 1mM MgCl2 and protease inhibitor cocktail (Roche Applied Science, Indianapolis, IN). Livers were homogenized and crosslinked with 2mM Disuccinimidyl glutarate (DSG) for 30 minutes at room temperature followed by a secondary crosslink step using 1% formaldehyde for 15 minutes. Crosslinking was quenched
4.4.6 Western blot analysis

Livers were homogenized in RIPA lysis buffer containing protease inhibitor cocktail, NaF and PMSF, sonicated briefly and rocked to lyse cells at 4°C. Transfected cells were harvested in RIPA and similarly rocked to lyse cells. 35-50μg of protein lysate was resolved on 10% SDS-PAGE. Antibodies used for western blots or IP include: SIRT6 and TBP (Abcam), AcBMAL1, SIRT1 and Myc (Millipore), Flag (Sigma-Aldrich), and SREBP-1 (Santa Cruz and kind gift from Dr. Timothy Osborne).

4.4.7 Plasmids and luciferase reporter assays

JEG3 cells were transfected with 50-100ng myc-BMAL1, myc-CLOCK (plasmids previously described (Katada and Sassone-Corsi, 2010[92])), Flag-SREBP1c (generous gift from Dr. Timothy Osborne), or Flag-SIRT6 (kind gift from Dr. Katrin Chua) in 24-well plate format. LacZ was also ectopically expressed with either Dbp-Luc (Kiyohara et al., 2008[102]), Fasn-Luc -1594/+65 or Fasn-Luc -65 MT (Bennett et al., 1995[155]; Joseph et al., 2002[85]). For detailed methodology, refer to Supp. Experimental Procedures.

4.4.8 Mass spectrometry (MS) for metabolomics analysis

Metabolomics analysis was carried out by Metabolon, Inc. (Durham, NC) using published methodology (Evans et al., 2009[46]). For detailed methodology, refer to Supp. Experimental Procedures.
4.4.9 Statistical analysis and MotifMap

For analysis of rhythmic genes and metabolites, the non-parametric test, JTK_CYCLE, was used incorporating a window of 20-28 hours for the determination of circadian periodicity (Hughes et al., 2010[73]), including amplitude and phase analysis. A gene was considered circadian, if at least one of its transcripts passed a p-value cutoff of 0.01, while metabolites were considered circadian at a p-value cutoff of 0.05. MotifMap (Daily et al., 2011[28]; Xie et al., 2009[195]) was used to determine enriched transcription factor binding sites in the microarray data sets, using a false discovery rate (FDR) of 0.1, a BBLS cutoff of 0.25 and searching within 1kb of the TSS.
4.5 Figures and Tables

4.5.1 Figure 4.1

Figure 4.1: Main Figure 4.1
4.5.2 Figure 4.2

Figure 4.2: Main Figure 4.2
4.5.3 Figure 4.3

Figure 4.3: Main Figure 4.3
Figure 4.4: Main Figure 4.4
4.5.5 Figure 4.5

Figure 4.5: Main Figure 4.5
4.5.6 Figure 4.6

Figure 4.6: Main Figure 4.6
4.5.7 Figure 4.7

Figure 4.7: Main Figure 4.7
### Table 1: Gene expression overlap by ZT between SIRT6/SIRT1 WT strains

<table>
<thead>
<tr>
<th>ZT</th>
<th>Number of genes</th>
<th>Percent of genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>24672</td>
<td>92</td>
</tr>
<tr>
<td>04</td>
<td>24820</td>
<td>93</td>
</tr>
<tr>
<td>08</td>
<td>23857</td>
<td>89</td>
</tr>
<tr>
<td>12</td>
<td>25423</td>
<td>95</td>
</tr>
<tr>
<td>16</td>
<td>23759</td>
<td>89</td>
</tr>
<tr>
<td>20</td>
<td>24151</td>
<td>90</td>
</tr>
</tbody>
</table>

### Table 2: Metabolite overlap by ZT between SIRT6/SIRT1 WT strains

<table>
<thead>
<tr>
<th>ZT</th>
<th>Number of metabolites</th>
<th>Percent of metabolites</th>
</tr>
</thead>
<tbody>
<tr>
<td>00</td>
<td>374</td>
<td>99</td>
</tr>
<tr>
<td>04</td>
<td>376</td>
<td>100</td>
</tr>
<tr>
<td>08</td>
<td>366</td>
<td>97</td>
</tr>
<tr>
<td>12</td>
<td>373</td>
<td>99</td>
</tr>
<tr>
<td>16</td>
<td>342</td>
<td>91</td>
</tr>
<tr>
<td>20</td>
<td>360</td>
<td>95</td>
</tr>
</tbody>
</table>

Figure 4.8: Supplemental Figure 4.8

Figure 4.9: Supplemental Figure 4.9
Figure 4.10: Supplemental Figure 4.10
Figure 4.11: Supplemental Figure 4.11
Figure 4.12: Supplemental Figure 4.12

A Mouse Rgs16 promoter

Mouse Mthfd1l promoter

Figure 4.12: Supplemental Figure 4.12
Figure 4.13: Supplemental Figure 4.13

A. Dbp Promoter

B. Per1 Promoter
Figure 4.14: Supplemental Figure 4.14

A

**Amd1 promoter**

- WT liver
- SIRT6 KO liver
- IgG

**Relative BMAL Recruitment**

<table>
<thead>
<tr>
<th>ZT</th>
<th>WT liver</th>
<th>SIRT6 KO liver</th>
<th>IgG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>8</td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>12</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>16</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>20</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>

B

**Per1 promoter**

- WT liver
- SIRT6 KO liver
- IgG

**Relative BMAL Recruitment**

<table>
<thead>
<tr>
<th>ZT</th>
<th>WT liver</th>
<th>SIRT6 KO liver</th>
<th>IgG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>3.0</td>
<td>3.0</td>
<td>3.0</td>
</tr>
<tr>
<td>8</td>
<td>2.5</td>
<td>2.5</td>
<td>2.5</td>
</tr>
<tr>
<td>12</td>
<td>1.0</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>16</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>20</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
</tbody>
</table>
Figure 4.15: Supplemental Figure 4.15
**Supplementary Table 1: Common SREBP1/SIRT6/SIRT1 target genes by Biological Function**

<table>
<thead>
<tr>
<th>GO Term</th>
<th>SIRT6/SREBP1 Overlapping Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lipid biosynthetic process</td>
<td>fasn, elovl1, acacb, fdps, dhrs9, chka, pnpla3, hmgcr, lss, dgat2, elovl6</td>
</tr>
<tr>
<td>Fatty acid metabolic process</td>
<td>fasn, aacs, elovl1, acacb, ppard, elovl6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>GO Term</th>
<th>SIRT1/SREBP1 Overlapping Genes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lipid biosynthetic process</td>
<td>hsd3b6, hsd3b4, fabp5, hmgcr, acaca, insig1, hsd3b5</td>
</tr>
<tr>
<td>Steroid metabolic process</td>
<td>hsd3b6, hsd3b4, hmgcr, serpina5, srebf1, insig1, hsd3b5</td>
</tr>
</tbody>
</table>

Figure 4.16: Supplemental Figure 4.16
### 4.5.17 Supplemental Table 4.17

Supplementary Table 2: Primer sequences for Gene Expression.

<table>
<thead>
<tr>
<th>Primer Name</th>
<th>Primer Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>mNpnt_F</td>
<td>5’-TGACTTGCACTGGAGACAG-3’</td>
</tr>
<tr>
<td>mNpnt_R</td>
<td>5’-CCTAAGGACAGGCACAGGT-3’</td>
</tr>
<tr>
<td>mDbp_F</td>
<td>5’-AATGACCTTTGAACCTGATCCCCGCT-3’</td>
</tr>
<tr>
<td>mDbp_R</td>
<td>5’-GCTCCAGTACCTCTCATCTCTTGT-3’</td>
</tr>
<tr>
<td>mFasn_F</td>
<td>5’-ATCCCTGGAACGAGAACGATCT-3’</td>
</tr>
<tr>
<td>mFasn_R</td>
<td>5’-AGAGACGTGACTCTCGGACTT-3’</td>
</tr>
<tr>
<td>mHmgcr_F</td>
<td>5’-AGCAGACAATGTTGAGGAGTT-3’</td>
</tr>
<tr>
<td>mHmgcr_R</td>
<td>5’-GCGTTTTCTCCAGGATTGTC-3’</td>
</tr>
<tr>
<td>mLss_F</td>
<td>5’-CATGCGGACATACCTACCAAG-3’</td>
</tr>
<tr>
<td>mLss_R</td>
<td>5’-TATGGACCTGGACCTAGCA-3’</td>
</tr>
<tr>
<td>mRgs16_F</td>
<td>5’-GCTCACACATCTTTGACGA-3’</td>
</tr>
<tr>
<td>mRgs16_R</td>
<td>5’-CCACATCGGAACGAACTGTA-3’</td>
</tr>
<tr>
<td>mSds_F</td>
<td>5’-ATCATCGGCATGAGACCT-3’</td>
</tr>
<tr>
<td>mSds_R</td>
<td>5’-CTCGGAGAAATGGGTGTG-3’</td>
</tr>
<tr>
<td>mMthfd1l_F</td>
<td>5’-ACGAACGACCGATTTCGAG-3’</td>
</tr>
<tr>
<td>mMthfd1l_R</td>
<td>5’-CGCTCTCTGCTGTA-3’</td>
</tr>
<tr>
<td>mAmd1_F</td>
<td>5’-CGCATGAAATTCTGACTGCTG-3’</td>
</tr>
<tr>
<td>mAmd1_R</td>
<td>5’-CAGAATCCACTCCACTACGA-3’</td>
</tr>
<tr>
<td>mSrebp1c_F</td>
<td>5’-GGAGCCATGGATGCACATT-3’</td>
</tr>
<tr>
<td>mSrebp1c_R</td>
<td>5’-GGCCCGGGGAAGTCACGT-3’</td>
</tr>
<tr>
<td>mRevErba_F</td>
<td>5’-AGGCTGCTAGTTGTTGTTG-3’</td>
</tr>
<tr>
<td>mRevErba_R</td>
<td>5’-CTCCATCGTCCATCAATC-3’</td>
</tr>
<tr>
<td>mBmal1_F</td>
<td>5’-GCAATGGACACTGACTACCAAGA-3’</td>
</tr>
<tr>
<td>mBmal1_R</td>
<td>5’-TCCTGGACATTGTGACTGAT-3’</td>
</tr>
<tr>
<td>mSirt1_F</td>
<td>5’-CAGACCCCTCAAGGATGTTT-3’</td>
</tr>
<tr>
<td>mSirt1_R</td>
<td>5’-CTTTGGACCTCTGCAACCTG-3’</td>
</tr>
<tr>
<td>mSirt6_F</td>
<td>5’-ATCTTGGACCCACGAGG-3’</td>
</tr>
<tr>
<td>mSirt6_R</td>
<td>5’-GCTTCTCGAAGGTGTTGTC-3’</td>
</tr>
<tr>
<td>mPer1_F</td>
<td>5’-ACCAGCCTGTCATGATGACATA-3’</td>
</tr>
<tr>
<td>mPer1_R</td>
<td>5’-GTGCACAGCACCAGTTCCC-3’</td>
</tr>
</tbody>
</table>

Figure 4.17: Supplemental Figure 4.17
### Supplemental Table 3: Primer sequences for ChIP.

<table>
<thead>
<tr>
<th>Primer Name</th>
<th>Primer Sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>mDbp Intron 1_F</td>
<td>5'-ATGCTCACACGGTGAGACA-3'</td>
</tr>
<tr>
<td>mDbp Intron 1_R</td>
<td>5'-CTGCTCACAGGTGACCTCAT-3'</td>
</tr>
<tr>
<td>mDbp 3'UTR_F</td>
<td>5'-GGCTGGATGTATAGCTAGCA-3'</td>
</tr>
<tr>
<td>mDbp 3'UTR_R</td>
<td>5'-GGCAACGAGAGCAGACAAGA-3'</td>
</tr>
<tr>
<td>mDbp promoter_F</td>
<td>5'-CCACTGGGGAGGCAATGAG-3'</td>
</tr>
<tr>
<td>mDbp promoter_R</td>
<td>5'-CCACTGGGGAGGCAATGAG-3'</td>
</tr>
<tr>
<td>mRgs16-568_F</td>
<td>5'-TGGCCAAATCGCTAGTTTTT-3'</td>
</tr>
<tr>
<td>mRgs16-568_R</td>
<td>5'-GGGTGCATTTCAAGAAAAG-3'</td>
</tr>
<tr>
<td>mRgs16-2523_F</td>
<td>5'-GCAGGGAGGTTGAGAGGT-3'</td>
</tr>
<tr>
<td>mRgs16-2523_R</td>
<td>5'-GTGCTTCATCTCCACAGA-3'</td>
</tr>
<tr>
<td>mRgs16-1341_F</td>
<td>5'-CACCAGCTGTGAGGTTTT-3'</td>
</tr>
<tr>
<td>mRgs16-1341_R</td>
<td>5'-CCAGAGGTTCTTCAAGGAT-3'</td>
</tr>
<tr>
<td>mMthfd1L-714_F</td>
<td>5'-TGGAACCTTTCAAGACAAGG-3'</td>
</tr>
<tr>
<td>mMthfd1L-714_R</td>
<td>5'-AGGAGAAMCTGATCAAGA-3'</td>
</tr>
<tr>
<td>mMthfd1L+1554_F</td>
<td>5'-GGACGACGATCCAGGGT-3'</td>
</tr>
<tr>
<td>mMthfd1L+1554_R</td>
<td>5'-ACACACGGCACTGAGA-3'</td>
</tr>
<tr>
<td>mMthfd1L+1254_F</td>
<td>5'-TCAGTGTTTTTTCAGGCT-3'</td>
</tr>
<tr>
<td>mMthfd1L+1254_R</td>
<td>5'-TGGGGATGGAACTTTGCTCCTT-3'</td>
</tr>
<tr>
<td>mFasn TSS_F</td>
<td>5'-GTGGGAGGAGAGATGAGG-3'</td>
</tr>
<tr>
<td>mFasn TSS_R</td>
<td>5'-CACATGGGCTGACAGCTTG-3'</td>
</tr>
<tr>
<td>mFasn 3'UTR_F</td>
<td>5'-GGGATACCTCAGGAAAC-3'</td>
</tr>
<tr>
<td>mFasn 3'UTR_R</td>
<td>5'-ACAGCCCTCCTCCTATCCT-3'</td>
</tr>
<tr>
<td>mHmegr -388_F</td>
<td>5'-GTAGTCAGGGGGTTTG-3'</td>
</tr>
<tr>
<td>mHmegr -388_R</td>
<td>5'-GTGGATACCCGAACTCAG-3'</td>
</tr>
<tr>
<td>mLss -69_F</td>
<td>5'-CCACCTGGGTAGTGTGCT-3'</td>
</tr>
<tr>
<td>mLss -69_R</td>
<td>5'-TGCTGTTCAGACACCTCC-3'</td>
</tr>
<tr>
<td>mPer1 Ebox_F</td>
<td>5'-ATCCTGATCGATGGCTGACCTGA-3'</td>
</tr>
<tr>
<td>mPer1 Ebox_R</td>
<td>5'-TCTCTTCTGGCATCTGGATGT-3'</td>
</tr>
<tr>
<td>mAmd1+1407_F</td>
<td>5'-GGGCGTGGGTAGAATGTT-3'</td>
</tr>
<tr>
<td>mAmd1+1407_R</td>
<td>5'-GAAACGTCAAGGTTGT-3'</td>
</tr>
</tbody>
</table>

---

Figure 4.18: Supplemental Figure 4.18
Bibliography


Appendix A

DockingDB

A.1 Introduction

Although there are several open-source and commercially available computational tools for virtual drug screening for drug-lead discovery – including but not restricted to Dock6[106], Autodock[134], Autodock Vina[138] and Schrodinger’s Glide[7] – there is still a lack of a more generic, tool agnostic, and scalable framework that is able to leverage the advantages offered by these tools in a high-performance computing (HPC) environment. Further, even with the guidance of in silico methodologies such as the Relaxed Complex Scheme[8] (RCS), the overhead of managing an inherently data and computationally intensive virtual drug hunting process becomes burdensome for most researchers.

Here we have developed DockingDB which is a framework built on top of an HPC pipeline and existing proteomics and chemical informatics tools and databases – SCRATCH[79], COSMOS[3], and ChemDB[78] – to support an iterative virtual drug screening methodology such as the RCS. DockingDB provides an extensible framework for data, process, and analysis application management, and permits deployment onto a cluster of commodity machines
(using grid engines, eg, Sun Grid Engine, PBS, etc) or cloud computing environment (Amazon EC2 and S3, http://aws.amazon.com/). Not only is the DockingDB framework agnostic of the grid scheduler but also of the molecular mechanics and ligand docking tools. Moreover, we have developed several extensions to the pipeline which take advantage of machine learning techniques for post-molecular mechanics simulation, pre-docking, and post-docking analyses. These extensions include clustering and dimensionality reduction techniques such as vector quantization[117] (k-means and k-medoids), spectral clustering[12], and manifold learning[154]. Furthermore, we are re-implementing these machine learning techniques for the GPU to accelerate computation in key components of the pipeline.

The utility of our framework becomes apparent especially in collaborative efforts with experimental structural biologists and organic chemists who might have limited access to HPC resources. Currently, we are utilizing our framework in investigating two biological problems and describe preliminary results: 1) two potential targets involved in cancer metastasis, and 2) a putative heme degradation enzyme critical to the survival of Mycobacterium tuberculosis. Several other docking projects are in progress.

A.2 Background and motivation

Our framework mainly addresses the difficulties involved in structure-based virtual drug screening as opposed to the ligand-based approach. In general, we proceed according to the iterative schema outlined by the Relaxed Complex Scheme (as shown in Figure A.1) for selecting representative structures of a target protein from molecular dynamics simulations against which ligand and fragment libraries are docked. The input to this schema is usually a trajectory from molecular dynamics simulations generated by open-source tools such as NAMD[81], AMBER[27] and Gromacs[64]. Otherwise, a protein structure from PDB can also be used; although this generally is not acceptable since crystallographic snapshots typically
capture non-native conformations but rather conformations amenable to crystallization. The following sections discuss the common problems and what we consider impediments to high-throughput virtual drug screening.

![Diagram of Receptor Ensemble and Ligand Ensemble](image)

Figure A.1: Relaxed Complex Scheme from Amaro, et al 2008.

**A.2.1 Data management problem**

Typically, the size of digital ligand libraries range in the order of thousands and up to the millions of small molecules (Zinc[77], ChemDB, PubChem). From our experience, manipulating even just thousands of files on either local or over network storage becomes excruciatingly cumbersome as the number of molecules approaches the limit of traditional file-systems (NTFS, FAT32, ext3, NFS). The I/O limitations of traditional storage stifles not only ligand preparation but also post-docking analysis in which multiple conformations may be produced. Although some form of directory management may be employed by more technically savvy users, often times such scripts become too esoteric and specific for a particular protein
target and thus not generalizable. Thus, a generic, extensible, and technology agnostic data management is thus required to encapsulate and abstract the subtle nuances across different docking experiments.

A.2.2 Task management problem

Aside from the limitations of traditional storage systems when dealing with large datasets of ligands, most post-molecular mechanics (e.g., VMD[74]) and docking tools (e.g., PyRx[4]) only support running on a single host. Thus, specialty scripts often times need to be developed in order to perform simulations across a cluster. Although these can be sufficient to parallelize either docking or post-molecular dynamics analysis, the output of these programs are still stored on traditional storage systems. Further, aggregation of the results becomes cumbersome. As with storage “tricks”, the parallelization schemes also become esoteric and not generalizable. For a high-throughput virtual screen to be feasible, the parallelization of the computational tools needs to be scheduler agnostic as well as independent of the underlying computational platform – local machine, commodity clusters, or cloud computing.

A.2.3 Application management problem

Typically, system preparation (whether configuring a molecular dynamics simulation and post-processing the trajectory files, or preparing ligands and analyzing post-docking results) entails juggling several standalone applications. For example, preparing conformers for the Relaxed Complex Scheme requires loading the trajectory file on a single-host application such as VMD. The trajectory files usually need to be strided in order to fit enough time steps in memory in order to perform RMSD clustering or QR-factorization (or manual selection techniques). Although some of these applications are multithreaded, they are still limited to one host. Further, filtering of ligands both before and after docking simulations still rely
on traditional file-based storage. And although the available GUIs (e.g., VMD and PyRx) are useful especially for visualization, they require human interaction and therefore impede high-throughput manipulation, preparation, and analysis of targets and ligands.

A.3 DockingDB: Architecture

To address the issues described above, we present the DockingDB architecture shown in Figure A.2. We have abstracted three key components in order to support multiple storage systems, grid schedulers, and applications. These key components include Data Management, Process Management and the Application layers.

A.3.1 Data management

In order to avoid I/O limitations of traditional storage systems, we have developed an abstract data access layer (see Figure A.3) that can support not only multiple flavors of relational database systems but can also leverage distributed file-systems such as Hadoop as well as cloud storage such as Amazon S3. The DockingDB data layer provides an extensible API that can be utilized by other components of the framework, e.g., user applications and task processes.

Our current deployment leverages a MySQL database running on two Niagara2-based Sun-Fire T5120 servers that are attached to a high-end Fibre Channel Storage Area Network (SAN) equipped with a BlueArc Titan filer and Hitachi Data Systems (HDS) disk arrays. These databases serve as centralized storage systems from which ligands are queried and obtained, and onto which docking results are stored for future analysis. Further, this layer provides access and integration to external databases such as ChemDB.
Figure A.2: DockingDB Architecture. The core of the architecture (off-yellow bounding box) consists of the data access layer, process management, and client access layer. The data access layer provides access to underlying storage technologies (lower-left blue boxes). The process management layer provides access to the underlying task scheduler and computational resources (lower-right box). The client access layer provides the programming interface for utilizing the data and task layers. Analysis tools are written utilizing the client API.

A.3.2 Process management

The process layer shown in Figure A.4 provides a scheduler agnostic API which permits deployment of tasks on popular grid schedulers as well as cloud computing providers (e.g., Amazon EC2). The process layer allows programs to be ”dropped in” and deployed in a parallel manner. Our supported tools can be currently deployed on a Sun Grid Engine running on heterogenous clusters of multi-core Intel/AMD hosts.
Figure A.3: DockingDB Data Layer provides a storage technology agnostic API to leverage more modern relational, distributed, and cloud storage technologies..

Figure A.4: DockingDB Process Layer provides an abstraction for managing tasks across clusters of compute nodes or cloud computing infrastructure such as Amazon EC2.
A.3.3 Application layer and extensions

At the highest level, the application layer shown on Figure A.5 provides a client API that leverages the storage systems and tools supported in both the data and process layers. The application layer provides access through direct programmatic integration or through web-services. As such, both standalone and web applications can access supported tools and backend storage.

![Diagram of DockingDB Application Layer](image)

Figure A.5: DockingDB Application Layer provides the "glue code" API through which the underlying storage and compute technologies can be accessed. This layer provides the client API upon which web-based or standalone applications can be developed.

A.3.4 Current deployment

As described by RCS (see Figure A.1), the intuitive approach for structure-based docking involves sampling an ensemble of conformations of the target protein as well as aggregating an ensemble of ligand libraries for docking. These two steps need to be performed in an automated fashion on a high-performance computing environment to make high-throughput virtual screening feasible.

In order to select conformers in a high-throughput fashion, we have implemented a MapReduce[29] application for RMSD-based vector quantization (using k-medoids variant). In order to be agnostic of the underlying molecular mechanics tool, we utilize and extend a trajectory-file agnostic, open-source MD package[2]. Further, we have implemented a GPU-based (CUDA Framework) spectral graph clustering tool which provides over 24x speed up (20 minutes to
50 seconds). At the time of writing, we are engaging in procuring a dedicated GPU-cluster based on the Fermi architecture from NVIDIA[6]. The selected trajectory frames are then cached on a central database.

In the case of ensemble ligand preparation, we have curated several thousands ligands from several databases that include (but not limited to) the NCI diversity set II, Maybridge and DrugBank as well as ZINC. We have also integrated ChemDB (http://cdb.ics.uci.edu) and its fast chemical search[139] capabilities to filter through many public and private databases of chemicals based on chemical similarity[105] and properties as well as vendor metadata[80]. The curation process usually requires a 3D minimization step for which we have integrated several tools including Corina[157], OpenBabel[5], and COSMOS[3]. These tools provide the initial, energy minimized 3D conformation of the ligands.

Once the set conformers of a target protein and a filtered set of ligands have been selected, a set of docking experiments can be deployed using AutoDock Vina, Dock, or a favorite docking tool which contains the desired set of potential energy functions. Configuration for each docking program is also stored in the central database. For example, the bounding box can be restricted to a specified region (e.g., active site) or encompass the entire protein (i.e., blind docking). In our cyber-infrastructure, the docking simulations are run on several hundred multi-core compute nodes which are managed by the Sun Grid Engine. The input (target protein conformer and ligand) and output (list of ligand poses) of each docking simulation are obtained and stored in our centralized relational database, respectively.

In the case of AutoDock Vina and Dock, several meta scores can be parsed in order to provide an initial list of attributes (free energy of binding, van der Waals and electrostatic components, etc) to be used in filtering the results in the post-docking analysis step. In order to fully automate this portion of the pipeline, we take advantage of the centralized database to filter based on any set of these attributes. For example, per protein conformer we can select the top hits (or least favorable hits) with a simple SQL query. Further, in
order to fully assess the ensemble of potential drug-leads, we perform additional filtering that removes redundancies across top hits per protein conformer.

Additionally, we are extending chemical fingerprint features used in chemical similarity search to further prune the sets of ligands. To do so, we incorporate pharmacophore information (electron donor and acceptor pairs, aromatic interaction, etc) between the ligands and target. We are currently designing and testing additional dimensionality reduction techniques (some implemented for the GPU) such as manifold learning[154] at the time of this writing. We expect that the inclusion of the pharmacophore fingerprints along with other machine learning techniques would further elucidate favorable ligand features for a target protein.

A.3.5 Applications: Biological systems

Regardless of the complexity of a cyber-infrastructure and underlying tools, the ultimate test lies in experimental assays. At the time of this writing, we have completed both the
post-molecular dynamics analysis and docking experiments on *Annexin A2* [118] which is a member of the *annexin* calcium-dependent phospholipid-binding protein family that play a role in the regulation of cellular growth and in signal transduction pathways. We are also investigating a related protein called *metastasin* [141] from the S100 family of proteins. These two proteins have been implicated in cancer metastasis.

For the case of *Annexin A2*, we have run docking experiments based on five representative structures (see Figure A.6) from the last 60-ns of a 100-ns molecular dynamics simulation. Our results indicate about 60% non-overlapping sets of top 1000 hits for these five conformers. Indeed, the ensemble approach provided us with five distinct conformations exposing certain core regions were were not initially apparent from the original crystallographic structure alone. We are waiting on the results from our experimental collaborators from FMP Berlin (http://www.fmp-berlin.de/) who are performing medium-throughput NMR spectroscopy on the N-15 labeled *Annexin A2*.

Additionally, we are waiting on the completion of a 100-ns molecular dynamics simulation for a protein involved in the heme degradation in the *Mycobacterium tuberculosis*. This protein is critical for the survival of the organism and therefore plays a potential target for inhibition. Further post-MD and docking analysis should soon be available.

### A.4 Future

We are continually designing and developing several other extensions that are built on top of the DockingDB cyber-infrastructure. The post-docking analysis tools which we are developing should soon be incorporated with the National Biomedical Computation Resource (NBCR, http://www.nbcr.net) through a collaboration with Dr. Rommie Amaro.
A.5 Acknowledgements

This investigation was supported by National Institutes of Health T15LM07443 from the National Library of Medicine through their Biomedical Informatics Training (BIT) program. The DockingDB cyber-infrastructure has been deployed at the data center in the Institute for Genomics and Bioinformatics in the Donald Bren School of Information and Computer Sciences at the University of California in Irvine. The synthesis and amplification of the Annexin A2 and metastasin for the project described above were performed by Gabe Ozorowski and myself, respectively, in collaboration with Dr. Harmut Luecke in the School for Biological Sciences.