Domino Tiling, Gene Recognition, and Mice

by

Lior Samuel Pachter

B.S. in Mathematics
California Institute of Technology (1994)

Submitted to the Department of Mathematics
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 1999

Author . . . . . .
Department of Mathematics
May 4, 1999

Certified by . . . . . .
Bonnie A. Berger
Associate Professor of Applied Mathematics
Thesis Supervisor

Accepted by . . . . . .
Michael Sipser
Chairman, Applied Mathematics Committee

Accepted by . . . . . .
Richard Melrose
Chairman, Department Committee on Graduate Students

Domino Tiling, Gene Recognition, and Mice

by

Lior Samuel Pachter

Submitted to the Department of Mathematics
on May 17, 1999, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy

Abstract

The first part of this thesis outlines the details of a computational program to identify genes and their coding regions in human DNA. Our main result is a new algorithm for identifying genes based on comparisons between orthologous human and mouse genes. Using our new technique we are able to improve on the current best gene recognition results. Testing on a collection of 117 genes for which we have human and mouse orthologs, we find that we predict 84% of the coding exons in genes correctly on both ends. Our nucleotide sensitivity and specificity is 95% and 98% respectively.

Most importantly, our algorithms are applicable to large scale annotation problems. The methods are completely scalable. We are able to take into account multiple or incomplete genes in a genomic region, splice sites without the usual GT/AG consensus, as well as genes on either strand. In addition to our algorithmic results, we also detail a number of computational studies relevant to the biological phenomena associated with splicing. We discuss the implications of directionality in splice site detection, statistical characteristics of splice sites and exons, as well as how to apply this information to the gene recognition problem.

The second part of the thesis is devoted to combinatorial problems that originate from domino tiling questions. Our main results are upper and lower bounds for forcing numbers of matchings on square grids, as well as the first combinatorial proof that the number of domino tilings of a $2n \times 2n$ square grid is of the form $2^n(2k+1)^2$. Our approach to both problems is concrete and combinatorial, relying on the same set of tools and techniques. We also discuss a number of new problems and conjectures.

Thesis Supervisor: Bonnie A. Berger
Title: Associate Professor of Applied Mathematics

For my grandfather, Jacob (Yankale) Pesate.

Acknowledgments

I thank my mom and dad first, not that "thanks" can express my gratitude for what they have contributed to me. I hope that the readers of this thesis can know what I mean.

The work described herein is the result of, and epitome of, collaborative research. I am deeply indebted and grateful to all who worked with me on the various aspects of what has become a thesis.

The guidance and thoughts of my advisor, professor Bonnie Berger, are evident throughout my work. I thank her for her help and for the unwavering support I received during my studies. I am grateful to professor Daniel Kleitman for suggesting the use of dictionaries in gene recognition, and for the many contributions he made during the ensuing developments. The key idea of applying comparative genomics principles to gene recognition was suggested by professor Eric Lander, who subsequently contributed many excellent ideas in critical moments. Above all, I thank Serafim Batzoglou who has been my colleague and friend for the last two years. His contributions and efforts directly enabled the completion of much of the gene recognition project described in this thesis. My thanks also go to Val Spitkovsky, who created and developed the dictionary described in Chapter 5. Eric Banks, Bill Wallis, Theodore Tonchev and John Dunagan, all contributed, both by coding and thinking, in countless ways to the research I did. Thank you all.

Of course, I cannot omit my thanks to those who turned my love affair with mathematics into a marriage. I thank my father for showing me what mathematics is, my friend and mentor Nitu Kitchloo for helping me realize that I can be a mathematician, professor Daniel Kleitman for showing me how to be one, and my students throughout the years for giving me a reason to continue being one. Special thanks go to my mother who taught me about those things in life that are much more important than mathematics. There have been many others who have inspired me and taught me, and of course along with those mentioned above, they all deserve much more than a thank you note in my thesis.

The influence and help of all of my friends is evident throughout this thesis. Fortunately, I have too many friends to list. Still, I cannot resist the temptation to specially thank Dave Amundsen (I *did* beat him at pigskin), Dave Finberg (the computer and combinatorics stud), Jing Li (thanks for the angst), Nitu Kitchloo (brother), Tal Malkin (vodka madame?), Mats Nigam (thanks Linda!), Boris Schlittgen (see Jing Li) and Glenn Tesler (who TeXed most of the difficult stuff in this thesis).

Thanks to Beth Hardesty for ungrouching me many times, often during critical moments of my graduate student career. Finally, I'd like to thank my love Son Preminger for all her help, understanding and fanchuking.

Contents

List of Tables

List of Figures

Notation / Acronyms

Symbol	Description	Page
bp	base pair	
CDS	coding sequence, abbreviation used in GENBANK	138
$\|S\|$	the length of a string of open and closed brackets	56
F_n	the nth Fibonacci number	56
MMG	global alignment algorithm	
$MMGG$	global alignment algorithm with gap penalties	
$PARTIALALIGN$	alignment algorithm for pieces of sequences	
$GLOBALALIGN$	the nested alignments algorithm	
$MMGGE$	the alignment score for an exon pair	
m_a	score for a match in an alignment algorithm	
m_s	score for a mismatch in an alignment algorithm	
g	score for a gap in an alignment algorithm	
$PAM(a,b)$	the PAM matrix score for a pair of codons a,b	
$CODON_h(a)$	the log odds ratio for codon a in human sequence	
$CODON_m(a)$	the log odds ratio for codon a in mouse sequence	
Sn	Sensitivity	
Sp	Specificity	
AC	Aprroximate Correlation	
$G \subseteq H$	G is an induced subgraph of H	
$V(G)$	vertex set of a graph G	
$E(G)$	edge set of a graph G	
K_n	the complete graph on n vertices	
$K_{n,m}$	the complete bipartite graph	
P_n	the path on n vertices	
R_n	$R_n = P_{2n} \oplus P_{2n}$ ($2n \times 2n$ rectangular grid)	
C_n	the cycle on n vertices	
T_n	$T_n = C_{2n} \oplus C_{2n}$ (torus)	
Q_n	the n dimensional hypercube ($K_2 \oplus \cdots \oplus K_2$ n times)	
$N(n,m)$	number of tilings of an $n \times m$ rectangular grid	113
$\overline{N}(n,m)$	same as $N(n,m)$ with one border square removed	119
$\varphi(M)$	forcing number of a matching M	105
$c(M)$	maximum number of disjoint, alternating cycles in M	106
$\# R$	number of domino tilings of a region R	114
$\#_2 R$	parity of the number of domino tilings of a region R	114

Nucleic Acid Codes (IUPAC)

Code	Bases	Mnemonic
A	A	A-denine
C	C	C-ytosine
G	G	G-uanine
T (or U)	T	T-hymine (or U-racil)
R	A or G	pu-R-ine
Y	C or T	p-Y-rimidine
S	G or C	S-trong (3 H-bonds)
W	A or T	W-eak (2 H-bonds)
K	G or T	K-eto
M	A or C	a-M-ino
B	C or G or T	not A
D	A or G or T	not C
H	A or C or T	not G
V	A or C or G	not T or U
N (or X)	any base	a-N-y (or unknown)
	gap	

Part I

Gene Recognition

Overview

This first part of this thesis outlines the results of an investigation undertaken to identify and annotate genes in human DNA. Even though most of the work described was initiated with this goal in mind, many of the results obtained are of independent biological interest.

We begin in Chapter 1 by reviewing the relevant biology. Much of the discussion is simplified so that the introduction is accessible to persons not familiar with biology (specifically, mathematicians). Additional information may be found in books by Lodish *et al.*, Watson *et al.* [58, 89], or Lander & Waterman [56]. Readers not familiar with biology jargon might find the glossary by Rieger *et al.* [73] to be a useful reference. We then proceed to outline the issues and problems addressed in this thesis. We discuss both the overall goals of gene annotation, as well as the particular aspects of various subproblems such as splice site recognition.

Chapter 2 contains a survey of previous related work.

In Chapter 3 we discuss distinguishing features of introns and exons that we discovered computationally. We survey some well known characteristics (*e.g.* length distributions), and also present some new results of our own. In particular, we describe a new statistical technique for distinguishing introns and exons based on frame preference, as well as novel methods for splice site prediction. Our splice site techniques are applied in subsequent chapters for the purpose of exon prediction.

Chapter 4 presents the framework of our gene recognition program. We describe our use of dynamic programming as a basic framework within which to do gene recognition. The various statistical tests and computational techniques we describe in the subsequent chapters are integrated within the dynamic programming framework to find "optimal" solutions to the various problems we address.

The "dictionary approach" described in Chapter 5 is an efficient way to score exons in a dynamic programming. It is based on finding matches of an input sequence to sequences in a database.

Chapter 6 describes an intriguing new approach to gene recognition based on an analogy of the Rosetta stone deciphering idea. Instead of using computational techniques to predict biological signals in one organism alone, information from another organism is used simultaneously with the first to enhance the signals. The motivation behind this approach is the observed fact that mouse genomic sequence exhibits high similarity to human genomic sequence in coding exons, but this similarity is less apparent in introns and other noncoding regions. We begin by describing a new alignment procedure designed for aligning large genomic regions from the human and mouse. The alignment algorithm represents a breakthrough over previous approaches in speed and accuracy. For example, we show how to align entire 400kB BACs in minutes. We then proceed to show how a good alignment can be used to find coding exons in genes by looking simultaneously at human and mouse aligning fragments. The approach once again uses dynamic programming, as well as many of the same ingredients used in the dictionary approach in Chapter 5; however, the combination of signals in the human and mouse allows for much more accurate predictions.

Chapter 1

Biology Background and Goals

1.1 The Genetic Dogma

The field of biology has been rapidly changing during the past century, largely due to remarkable discoveries in molecular biology. Perhaps more important than the many significant contributions that have been made, is the collective understanding that there is a framework underlying all of biology. The foundation of this framework is made of genes that form the blueprints for a massively parallel, self regulating, dynamical system. This system is incredibly complex, but despite this fact biologists have started to understand it in considerable detail, and one of the principles that have been discovered is that of the **genetic dogma**. This has been an understanding of how the system operates on a large scale. This chapter is intended to introduce the reader to some of the biology and terminology that is used throughout this thesis. Readers with only a mathematical background will find it useful to refer to more general texts such as Lodish *et al.* [58], or introductory chapters on biology (written for mathematicians) such as Chapter 1 in Lander & Waterman [56].

For the purposes of our discussion, we will define a **gene** (see Figure 1-1) to be a single, contiguous region of genomic DNA that encodes for one protein (it will be convenient to also consider the flanking regions that contain promoter signals, *etc.* as also being part of the gene). There are four different nucleotides that make up a sequence of DNA. These are **Adenine (A)**, **Cytosine (C)**, **Guanine (G)** and **Thymine (T)**. For our purposes, we will think of DNA as being a string on an alphabet of size 4 (A,C,G,T). DNA physically exists in the form of a double helix (containing two **strands**, and a gene can be a subsequence occurring on either strand. When a gene is expressed, it is first copied in a process known as **transcription**. This forms a product known as **RNA**, which is a working template from which a protein is produced in a process known as **translation**. Before translation, the RNA undergoes a **splicing** operation [79] conducted by certain enzymes, which typically delete most of it, leaving certain blocks of the original strand of RNA intact. These blocks are called **exons** and the parts that are removed are called **introns**. The result of this pruning is the "mature" RNA (**mRNA**), which is used during translation to make the protein. The protein consists of a sequence of amino acids linked together. During

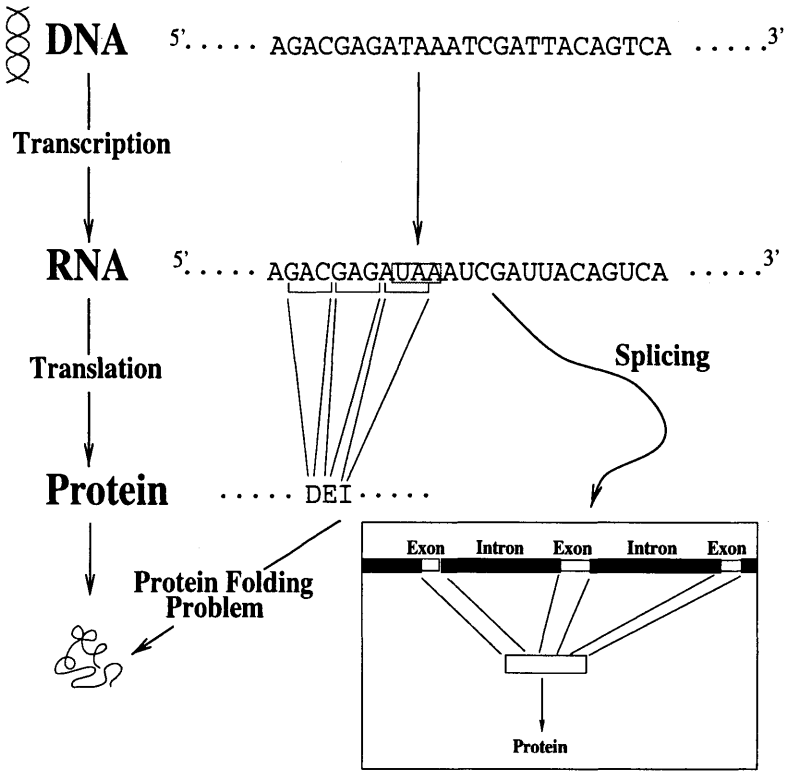


Figure 1-1: A schematic view of the transcription-translation process. During translation the T nucleotide becomes a U (Uracil). In this example, the boxed UAA triplet is not a codon and therefore does not end translation. Rather, the in-frame codons are "...GAC GAG AUA...". These are translated into "...D E I..." (D=Aspartic Acid, E=Glutamic acid, I=Isoleucine). Splicing occurs before translation. The translated amino acid sequence is folded into a protein.

translation, each amino acid is produced by a triplet of consecutive nucleotides, known as a **codon**, according to a known map that is called the **genetic code** (see Table A.1). This defines the **coding frame** of the gene.

The gene actually has a "start" translation signal (ATG) and a stop translation sequence (TAA, TAG or TGA) both within exons; the sequence within exons between these forms the coding part of the sequence which contains all the information used to make the protein. The rest of the gene consists of introns, initial and final "non-coding exons" (these are exons that are glued together with the coding exons, but that are not used for making protein), as well as flanking regions containing biological signals of various sorts. Notice that a gene has directionality; it can appear either on the **forward** direction, or in opposite strand in which case it is traversed in the opposite direction, hence **reverse complement**. The directionality of a gene is always annotated as $5' \rightarrow 3'$, that is, the gene is traversed from the $5'$ to the $3'$ direction. The splice sites on the $5'$ end of an intron are known as **donor splice sites**. On the $3'$ end they are known as **acceptor splice sites**.

1.2 The Splicing Cycle

The mechanism by which introns are spliced from human genes is not completely understood at this time. Nevertheless, a large number of pieces of the puzzle have been discovered, and while they cannot all be put in place at this time, enough is known to present a general picture of the process.

The components responsible for executing various stages of the splicing process are called **spliceosomes**. These are for the most part RNA-Protein complexes. We discuss some of the more important ones in the following sections, although we emphasize that it is widely believed at this time that not all the spliceosomes have been discovered. The splicing cycle is summarized in Figure 1-2[1] in pictorial format.

In order for introns to be properly spliced a number of conditions must be met: There have to be functional splice junction sequences in the pre-mRNA. These are alluded to below, and discussed in more detail in Chapter 3. Secondly, activity of at least three **small nuclear ribonucleoprotein particles** (abbreviated snRNPs and pronounced "snirps", these are spliceosomes) is required. The critical snRNPs are called U1, U2, U5, U4, and U6 (see Figure 1-3 [1]). Finally, the presence of ATP is necessary.

Figure 1-2 shows how an intron is excised in a sequence of steps. U1 attaches at the donor splice site by "recognizing" a consensus sequence around the dinucleotide **GT**. The recognition is accomplished, in part, by the complementarity of the RNA in the snRNP to the sequence. The precursor mRNA is cleaved at the $5'$ site and a lariat (loop) structure is formed between the **G** at the $5'$ site and an **A** further downstream in the intron. This **A** is part of a small subsequence known as the **branchpoint** which is recognized by the U2 snRNP. Finally, the $3'$ exon junction is cleaved and the

[1]From: MOLECULAR CELL BIOLOGY by Lodish *et al.* (c) 1986, 1990, 1995 by Scientific American Books Inc. Used with permission by W. H. Freeman and Company.

exons are ligated together.

The discussion above omits a number of critical, although contested issues in splicing biology. One of the important issues, is the role of mRNA secondary structure in the spliceosome interactions with the sequence. Evidence in this regard ranges from specific experiments affirming the role of secondary structure (*e.g.* Coleman and Roesser [19]), to counterarguments based on purely theoretical evidence such as extremely long introns (which would suggest that local structure may not play a large role). The exact role of secondary structure in splicing remains to be determined. Also, we have ignored some rare and different splicing interactions, where the GT may not be present in the splicing consensus, or other spliceosomes are involved (Sharp & Burge [80]).

1.3 Biological Signals and Patterns in DNA

The extent and variability of consensus sequences associated with biologically relevant signals largely determine the applicability of the signals for exon prediction. We briefly review the important biology associated with commonly used biological signals, and the consensus sequences associated with them.

Promoters

A **Promoter** is a DNA sequence that directs RNA polymerase to bind and initiate specific transcription of genes. Although promoters should, in principle, significantly aid in the distinction of genes (by indicating their exact beginning), the complex nature of the sequences, and their variability, makes the identification of promoters and unsolved problem. In eukaryotes, there is usually a conserved AT-rich region TATA (known as the Goldberg-Hogness or TATA box). Promoters are not analyzed in this thesis, in part because their computational recognition is very difficult given the current biology that is known about them. Nevertheless, we acknowledge that future work should, and will, include promoter recognition.

Kozak Consensus

The translation process described in the first section is executed by a **ribosome** which begins at an **initiator codon**. The initiator codon is usually ATG (methionine), and is surrounded by a relatively weak consensus known as the **Kozak consensus** [53]. The Kozak consensus is the sequence CCRCC**ATG**G. ATG is the preferred initiation codon (and appears in *all* of our learning and test genes), there are exceptions to this "rule". In humans, the codons ATA and ATT also appear as initiation codons and in mice there is also ATC. Because of the rarity of these occurrences, we have not allowed for the possibility of such initiation codons in this thesis, although a careful study of them and their consensus sequences is clearly necessary in future work.

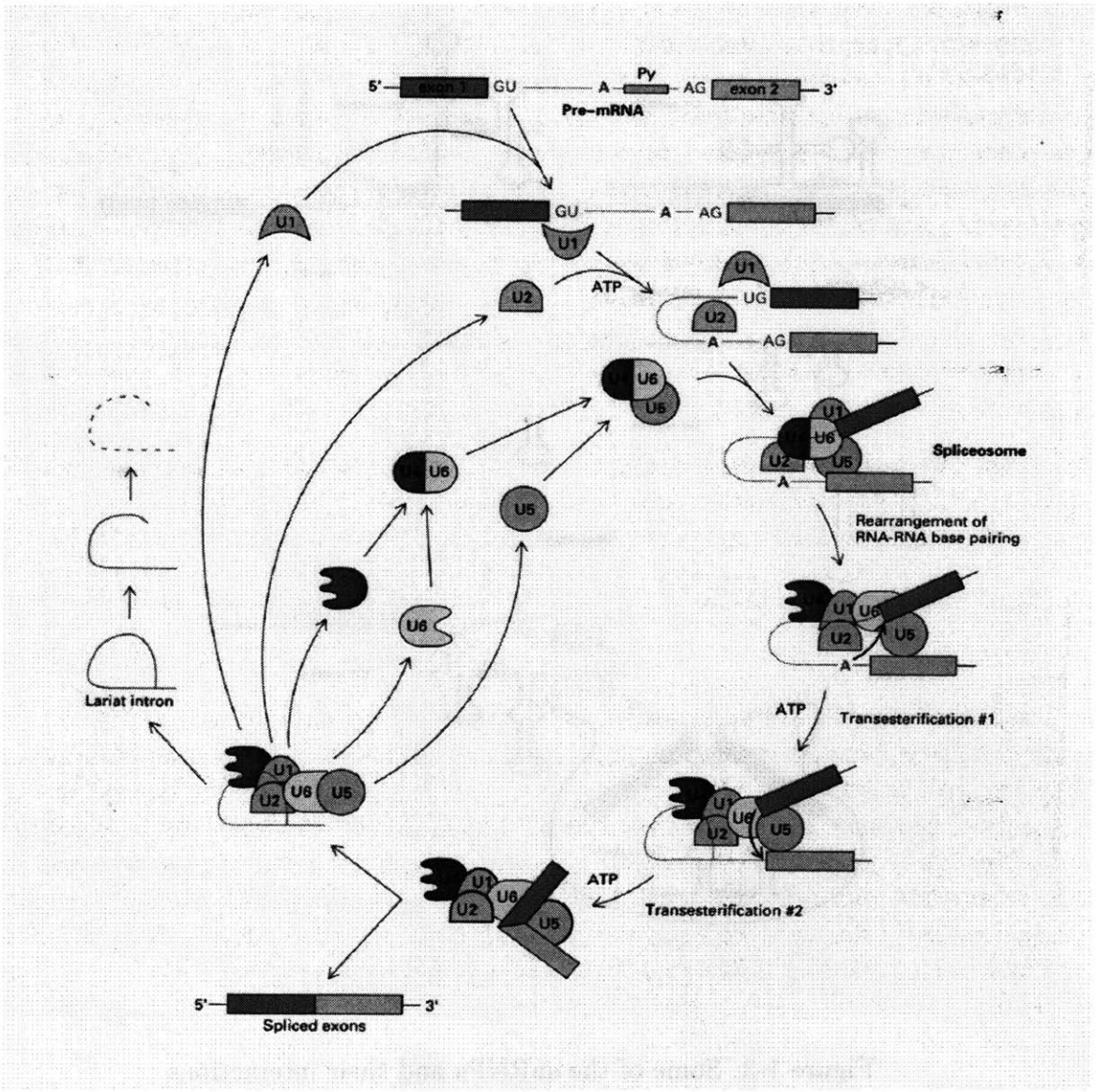


Figure 1-2: The Splicing Cycle

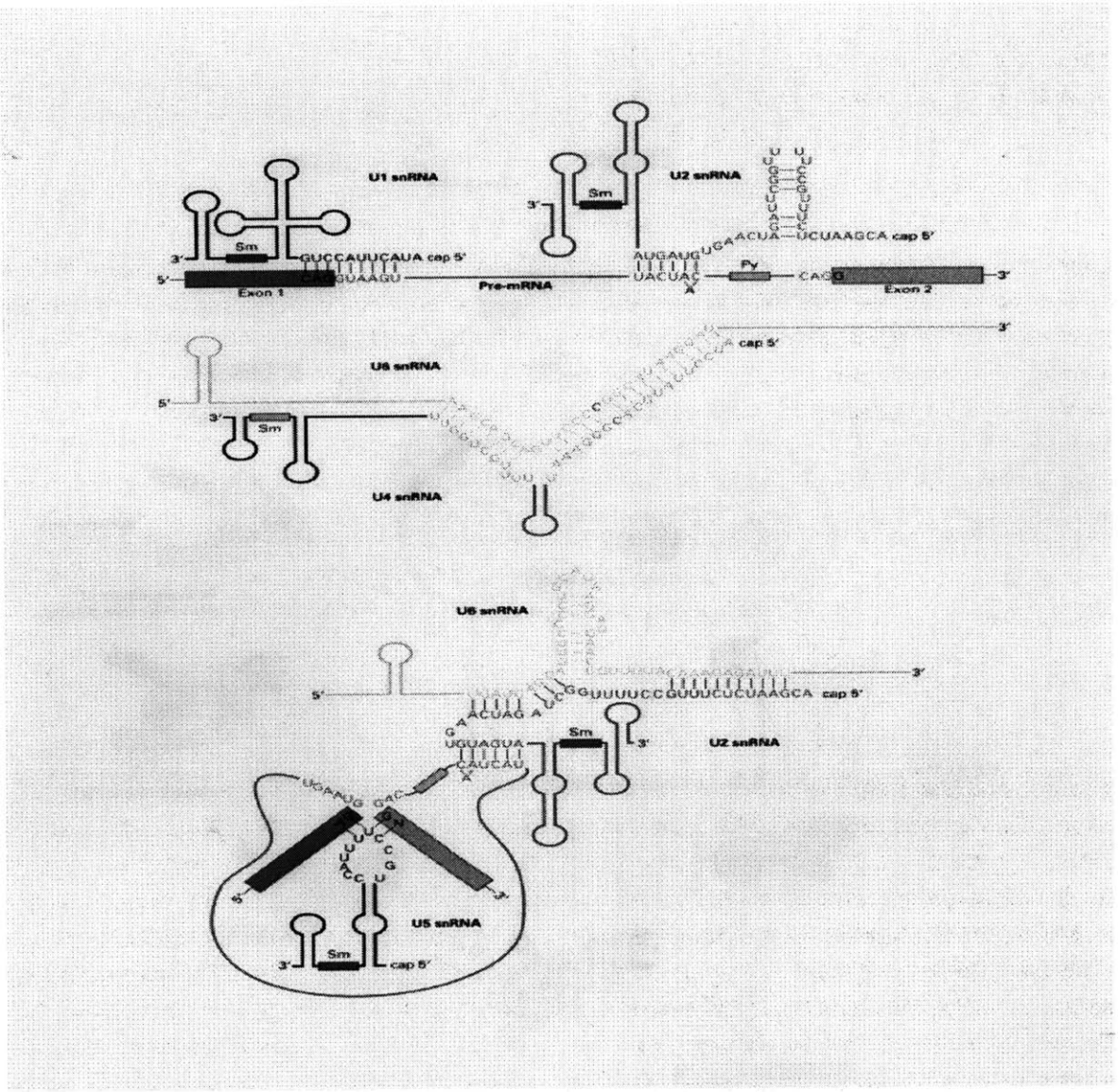


Figure 1-3: Some of the snRNPs and their interactions
The figure shows how the U4 and U6 snRNPs interact with each other, as well as the complex interaction between U2, U5 and U6. U1 is complementary to the consensus sequence CAGGTAAGT at the donor splice site in introns. Notice that the triplet CAG at the beginning of the donor splice site consensus is in the exon.

Signal Peptide

Also known as the **leader sequence** or **signal sequence**, this is a region of DNA following the initiation codon that initiates and mediates translocation of membrane and secretory proteins across the cell membrane or endoplasmic reticulum. The region is translated at the beginning of protein synthesis into a polypeptide that is recognized by protein-RNA complexes (and later cleaved). The region is characterized by a 7–15 long amino acid chain of hydrophobic residues. The cleavage site consists of a more polar C-terminal region.

Splice Sites

Of the many biological signals involved in splicing, the splice sites themselves are the best studied, and many results have been obtained regarding specific consensus patterns, as well as biologically relevant features in the neighborhoods of the splice sites. The biologically relevant characteristics of splice sites vary greatly between species, in what follows we discuss the specific case of humans:

Donor splice sites are characterized by a strong consensus of G**GT**RAG. About half the splice sites obey this consensus. The interaction of the U1 snRNP with this splice site is complex, and many results have been obtained about how and why specific sequences deviate from the consensus. Even though the 9 nucleotides adjacent to the GT seem to be the most important in determining an intron's propensity for splicing, the region adjacent to the splice site in the intron (up to 20 basepairs) seems to also play an important role in splice site selection. In particular, as outlined by McCullough and Berget [63], G triplets play an important role in splicing in certain introns. Surely there are many more such biologically important phenomena.

The acceptor splice site exhibits a much smaller consensus than the donor splice site. Indeed, C**AG** is the most common ending, with the nucleotide after the AG also having some significance. The region immediately preceding the acceptor splice site is known to enhance splicing when it is rich in **pyrimidines** (the nucleotides C or T). The pyrimidine rich region is usually of length about 20, and is known as the **pyrimidine tract**.

Branch Points

The branch point or **branch site** is the site at which the 5′ end of the intron becomes covalently attached near the 3′ end of the intron during splicing. The branch point is usually somewhere between 20-40 basepairs to the left of the acceptor splice site, and often appears right before the pyrimidine tract. The branch point has a strong consensus in yeast, conforming to the specific sequence TACTA**AC**. In humans, the consensus is much weaker, usually YNYUR**AY**, although of the many variants CTG**AC** is common.

Poly A Signal

The **poly A signal** appears after the stop codon of a gene and signifies the site for the initiation of **polyadenylation**. Polyadenylation is an mRNA processing event in eukaryotes characterized by the addition of 50 to 250 adenosine residues to the 3′ end of the mRNA (known as the **poly(A)** tail). The poly A signal consists of a pattern of four to six bases of DNA, for example AATAAA. This consensus pattern (and other consensus sequences) at the end of genes can be used for gene identification, however their identification and application is not explored in this thesis.

Repeats

Repeats are repetitive sequences of DNA that occur throughout eukaryotic genomes. They form approximately 30% of the DNA. Their importance derives from the fact that they are usually not found in coding exons, and therefore their recognition and annotation is of key importance in gene identification. The origin and role of repeats in human DNA is only partially understood, and is the source of much current research (see Smit [81]).

Repetitive DNA sequences can be classified into four main groups:

1. Repeated Genes.
2. Interspersed repetitive sequences.
3. Tandem highly repetitive sequences.
4. Inverted repeat or foldback sequences.

The interspersed repeats fall into two subcategories: short period interspersed repeats (called **SINE**s), and long interspersed repeats (called **LINE**s). The SINEs are usually about 300 bp long sequences, are repeated inside longer DNA segments of a few kilobases, and show high variation (*e.g.* Alu repeats). The LINEs, on the other hand, are long repeats, often more than a few kilobases, that are more homogeneous than their SINE counterparts. Examples include the L1 repeat sequences. Tandem repeats are short sequences of repeated DNA (such as CACACACACA ...). These may occur in coding exons, and are also known as low complexity repeats.

The wide variation in types of repeats, as well as the differences in homogeneity between the different classes, makes them very difficult to identify. Indeed, this is an area of ongoing research, and highly specialized packages such as `RepeatMasker` [97] have been developed for this purpose. In this thesis, we used `RepeatMasker` to mask repeats, although we also investigated the applications of the dictionary for repeat masking (discussed in Chapter 5).

1.4 What do you do with 100KB of human genomic DNA?

Recent advances in DNA sequencing technology have led to rapid progress in the Human Genome Project. Within a few years, the entire human genome will be sequenced. The rapid accumulation of data has opened up new possibilities for biologists, while at the same time unprecedented computational challenges have emerged due to the mass of data. The questions of what to do with all the new information, how to store it, retrieve it, and analyze it, have only begun to be tackled by researchers (for an excellent discussion about these issues see Lander [55]). These problems are distinguished from classical problems in biology, in that their solution requires an understanding not only of biology, but also of mathematics and computer science. Of the many problems, it is clear that the following tasks are of importance:

- Finding genes in large regions of DNA.
- Identifying protein coding regions within these genes.
- Understanding the function of the proteins encoded by the genes.

The important third problem, namely understanding the function of a newly sequenced gene, requires the solution of the second problem, identification of critical subregions which code for protein. Protein coding regions have different statistical characteristics from noncoding regions, and it is primarily this feature which enables us to distinguish them. An important aspect of work on the problem is the need to characterize these statistical differences and possibly explain their biological underpinnings.

1.5 The Computational Challenges in Gene Annotation

The computational task we are concerned with is that of determining from an experimentally determined sequence of nucleotides, of length on the order of 100,000, where the genes are, and what proteins these genes produce. We may also be interested in further annotations, describing specific features of the genes, such as repetitive regions, or sites of biological significance. This endeavor has three parts, though in practice one handles them together: the first two are determining where each gene is, and determining which parts of its sequence are exons and which are introns. Concurrently, it is necessary to annotate regions in an attempt to find features useful for the first two problems. In this thesis we focus on the problem of distinguishing exons from introns, although along the way we address some of the other annotation issues that arise.

Fortunately, we are not restricted to using only the obvious biological signals available to nature. Of primary importance is the use of repeats, which occur throughout the human genome, but very rarely in coding exons (see biology background above)

Secondly, the codons (and consequently amino acids) that code for protein, are not uniformly distributed, and their distribution differs from the distribution triplets in introns. This can help in distinguishing introns from exons. We can also use information from other organisms to enhance our signals (Chapter 6). Other restrictions such as consistency in coding frame between exons greatly reduces the number of possible parses in a given gene. Indeed, even though in principle the number of parses is exponential in the number of potential splice sites identified (Chapter 4), in practice many genes exhibit only a few possible parses after these numerous constraints are introduced.

1.5.1 Exon Prediction

In this section, we suggest a number of alternative definitions of what the exon prediction problem constitutes. The range of possible problems one might try to solve, combined with the vast differences in their difficulty, makes the selection of a goal an important issue. Indeed, results can vary greatly with the same test data depending on assumptions that have been made. The following assumptions are ones that may be appropriate in certain contexts:

1.	Consists of one complete gene on the forward strand.
2.	Consists of multiple complete genes on the forward strand.
3.	Consists of multiple complete genes on either strand.
4.	May consists of multiple complete genes, perhaps with partial genes on the ends, on either strand.

Table 1.1: Assumptions about the DNA in which one is to find coding exons

Solutions to the exon prediction problem have tended to concentrate on the model in which one can assume that there is only one complete gene. While this is useful in many cases, the technology by which genes are sequenced results in large genomic fragments which contain all the generalities listed above (in other words one has to assume that there are multiple genes, pseudogenes, *etc.*) In this thesis we have designed solutions based on different assumptions. For example, the dictionary method in Chapter 5 is based on the "one complete gene" assumption, whereas the comparative genomics chapter deals with the more general problem.

1.5.2 Other Problems

The exon prediction problem is complicated by a number of biological phenomena, many of which lead to interesting annotation problems in their own right:

Pseudogenes

A **pseudogene** is defined to be a DNA sequence significantly homologous (75 to 80%) to a functional gene, which has been altered so as to prevent any normal function. Pseudogenes can be classified into two categories, those that possess all the structural features of a gene (promoters, exons, introns, *etc.*) but have been mutated so that they are no longer functional (often a result of duplication of a gene and its silencing), and those that lack introns but do have a poly A tail. The latter type are believed to be the result of reverse transcription of mRNA, followed by integration of the resulting cDNA into a chromosomal site. In Chapter 5, we discuss the possibility of detecting reverse transcribed pseudogenes, by finding two adjacent regions in a gene that look like exons, but that cannot be frame consistent unless they have an intron in between them (of length not 0 modulo 3).

Alternative splice sites

Many genes can be spliced into a number of different variants, depending on environmental or developmental conditions in the gene. Often, alternative splicings lead to diseases or defects in an organism. Alternative splicing happens when a particular splice site can be selected in a different position, or when it is not used at all. Examples are given in Chapter 5.

The presence of an alternative splice site in a gene, renders the exon prediction problem ill-defined. Since there is no "correct" solution to how a gene is parsed, it makes no sense to try an return a predicted answer.

Thus, annotation of possible alternative splice sites is of great importance, especially since they seem to be abundant.

Branchpoints

The weak consensus of branch points in higher eukaryotes was mentioned in the biology background section of this chapter. This weak consensus makes it difficult to annotate branchpoints, or to effectively use them for exon prediction. The computational problem of annotating branchpoints is unsolved.

Chapter 2

Previous Work on Gene Annotation

Many computational methods have been developed for the purposes of gene annotation (Batzoglou *et al.* [7]). The different approaches that have been undertaken in an attempt to solve the gene recognition problem can be broadly classified as statistical (some may prefer the term AI/learning based) or homology based. In the past few years, the growing abundance of EST (expressed sequence tag) and protein data has resulted in a combination of both approaches being used in newer programs. In what follows, we attempt to provide a brief summary of the techniques that have been used in the past so that the reader can place our work in the appropriate context. In particular, we mention that the dictionary approach in Chapter 5 is an attempt to bridge the gap between statistical and homology based approaches. The work in Chapter 6 represents a novel way of tackling the gene recognition problem; we have coined a new term for it, *comparative prediction.*

2.1 Similarity Searching and Gene Annotation

Of the many applications of computer science in biology, perhaps the most successful has been the implementation of algorithms for finding similarities between sequences (for a discussion see Waterman [88]). The most widely used program developed for this purpose is `BLAST` developed by Altschul and others [3, 4], which is an alignment tool. `BLAST` is often manually applied for the purposes of gene annotation, including exon prediction and repeat finding. Other similarity search approaches include the `FLASH` [74] program which is an example of a clever use of a hash table to keep track of matches and positions of pairs of nucleotides in a database. The resulting information can be used to extract close matches to a given sequence. Nevertheless, neither of these search approaches have been designed for gene annotation.

2.2 Statistical Approaches

The vast number of exon prediction programs in existence precludes the possibility of providing a comprehensive survey without an extended discussion. Our aim here is to merely point out *some* of the more popular programs. Statistically based programs include GENSCAN [12], GENIE [54], GENEMARK [61], VEIL [36] (all based on hidden Markov models [HMM's]), FGENEH [83] (an integration of various statistical approaches for finding splice sites, exons, *etc.*), GRAIL [91] (based on neural networks) and GeneParser [82] (based on dynamic programming and neural networks) . Other approaches include language based techniques such as GenLang [25].

These programs have a number of characteristics in common, perhaps the most important being their reliance on a training set (to learn transition probabilities in the case of HMM's, or weights for neural networks). The problem of understanding the implications of this fact and its relationship to the performance of the programs is very difficult due to the relatively small amounts of publicly available data. We briefly discuss this in the results section that follows.

2.3 Homology Approaches

Homology based approaches exploit the fact that protein sequences similar to the expressed sequence of a gene are often in databases. Using such a target, one can successfully identify the coding regions of a gene. The idea is to find the "best" way to parse an input gene so that it best matches the given target after translation. The alignment based PROCRUSTES [28, 86, 64] program represents a very successful implementation of this idea. When a related mammalian protein is available, this program gives 99% accurate predictions and guarantees 100% accurate predictions 37% of the time; however, the user supplies the target protein sequence.

It is important to note that there are a number of current difficulties that arise in the implementation of homology based approaches. The first problem is the identification of good targets. This problem has begun to be addressed (Section 2.4). Additionally, the databases used to find targets were not designed with gene recognition as a goal, and so are not easy to use. For example, the cDNA databases are not always properly oriented. The protein databases may contain translated repeats. These are all issues that need to be dealt with when looking for good targets.

2.4 Hybrids

The difficulty of finding good targets for the homology approach is addressed in a recent approach [38]. Specifically, the AAT tool addresses this by automatically using BLAST-like information from protein or EST databases for exon prediction. The INFO program [57] is based on the idea of finding similarity to long stretches of a sequence in a protein database, and then finding splice sites around these regions. Such programs are becoming more important as the size of protein and EST databases increase.

2.5 Results

The analysis and benchmarking of gene recognition tools has become a science in and of itself. Of the many articles addressing these issues, we mention the excellent surveys of Burset and Guigó [13, 30]. With the exception of GENSCAN, the non-homology based algorithms are not sufficiently accurate to be relied upon. Accuracy claims range from 60-90 percent per nucleotide, and 30-80 percent per entire exon with exact numbers dependent on who is making the claim. Table 2.1 is from the GENSCAN website, containing statistics obtained by GENSCAN [11] as well as Burset-Guigó [13]. The programs were tested on the Burset-Guigó dataset (abbreviated as the BG dataset, see Appendix B). For definitions of the different statistics computed (such as Sensitivity, Specificity, *etc.*), see [13]:

	Accuracy per nucleotide			Accuracy per exon			
Method	Sn	Sp	AC	Sn	Sp	ME	WE
GENSCAN	0.93	0.93	0.91	0.78	0.81	0.09	0.05
FGENEH	0.77	0.85	0.78	0.61	0.61	0.15	0.11
GeneID	0.63	0.81	0.67	0.44	0.45	0.28	0.24
GeneParser2	0.66	0.79	0.66	0.35	0.39	0.29	0.17
GenLang	0.72	0.75	0.69	0.50	0.49	0.21	0.21
GRAIL II	0.72	0.84	0.75	0.36	0.41	0.25	0.10
SORFIND	0.71	0.85	0.73	0.42	0.47	0.24	0.14
XPound	0.61	0.82	0.68	0.15	0.17	0.32	0.13

Table 2.1: Accuracy statistics for programs on the BG dataset

These numbers are probably very optimistic compared to the performance observed in practice [30]. The alarming aspect of the current state of the field is that these programs perform much worse when tested on new data, namely genes that have been sequenced, whose intron/exon structure is known experimentally. Indeed, on a new sequence set, the programs identified about 1 in 6 genes correctly and completely missed the exons in 25 percent of the sequences. This poor performance is probably due to a number of factors, the most significant of which is that current "learning" takes place on small data sets which are often filled with errors since they have been annotated by the very same programs that are learning from them! Furthermore, the learning sets are often redundant and are not really true representatives of genes in entire genomes.

In practice, those who find genes use a very different approach. They hope that the cDNA or protein (or a good part of these) that are produced by the gene lie in one of the corresponding data bases. They then submit their sequences to BLAST [3], a program that finds best matches to members of the data base. When it is possible to match parts of the gene with an entire protein, then one has the answer to the problem, either by examining the alignments by eye, or submitting the matches to a program such as PROCRUSTES [28]. As the databases grow, the likelihood of

good matches to new genes increases. When this approach fails, they turn to the algorithms mentioned, and seek consensus results from them. The process is tedious, time consuming and does not necessarily produce correct results.

Chapter 3

Identification of Introns and Exons

In this chapter we study various characteristics of introns and exons that help us distinguish them from each other. We begin with a detailed analysis of splice sites. These are of special importance in the discrimination of introns and exons because they occur at the boundaries between the two. We then turn examine the various properties of the introns and exons themselves that are of computational importance.

3.1 Splice Sites

In this section we begin by describing some computational/statistical analysis of pair correlations around splice sites. These results lead to interesting observations of possible biological significance (see sections 3.2.2, 3.2.3 and 3.2.4). We continue by describing the GENSCAN splice site detector, and our modification of it which we use in Chapters 5 and 6 for exon prediction.

3.1.1 Pairwise Correlations

We begin by defining some terminology which is essential for our study. The *position* of a nucleotide around the donor splice site of an intron is defined to be the distance (in nucleotides) to the start of the intron (following the convention used in [58]). Negative positions indicate nucleotides in the exon. For example, positions +1 and +2 in an intron are the well conserved GT nucleotides. Position -1 refers to the last nucleotide in the exon, which is usually a G (note: some authors prefer to start the labeling in the intron with 0).

We also define positions around the acceptor splice site in the same way. In this case however positions -2 and -1 refer to the last two nucleotides in the intron. Notice that positions are always defined relative to the donor or acceptor splice sites. For example, in an intron of length 30, position +12 from the donor site and position -19 from the acceptor site represent the same nucleotide.

Correlation Matrices

Given two positions r, t around the donor splice site we constructed a 4×4 contingency table based on a set of genes with intron and exon boundaries marked. The rows and columns in the table are labeled with the four bases A, C, G, T. The ijth entry in the table is the number of times base i appeared in position r with base j in position t.

Given such a contingency table, we then computed Cramer's Statistic to test the null hypothesis that the nucleotide in position r is independent of the nucleotide in position t.

Cramer's statistic is derived from Pearson's Chi-Squared test. Let f_{ij} be the entry in the ith row and jth column of a contingency table. The ith row sum f_{i+} is given by $f_{i+} = \sum_j f_{ij}$ and the jth column sum f_{+j} is given by $f_{+j} = \sum_i f_{ij}$. Define $f = \sum_{i=1}^{4} \sum_{j=1}^{4} f_{ij}$ to be the sum of all the entries and let $e_{ij} = \frac{f_{i+} f_{+j}}{f}$. The statistic we compute is

$$X^2 = \sum_{i=1}^{4} \sum_{j=1}^{4} \frac{(f_{ij} - e_{ij})^2}{e_{ij}} \tag{3.1}$$

and the null hypothesis is rejected if X^2 exceeds $\chi^2_{\alpha,9}$ (α is usually taken to be .05).

Pearson's Chi-Squared Test is good in detecting dependencies between positions, however it is not as useful for measuring relative strengths of correlations. For this purpose, we use Cramer's V^2 statistic [1], defined for $a \times b$ contingency tables by

$$V^2 = \frac{X^2}{n \; min(a-1, b-1)} \tag{3.2}$$

where n is the sample size.

Given some window, say from $-k$ to k, around the donor splice site, we computed Cramer's statistic for each pair of positions r, t ($-k \leq r, t \leq k$) (in our case $a = b = 4$). The V^2 values were tabulated in a $2k \times 2k$ symmetric matrix which we call the **donor correlation matrix**. A similar matrix was also constructed for the acceptor splice site. Figure 3-1 shows the correlation matrices for donor and acceptor splice sites, computed in for a window of size $k = 40$.

Observations

It is evident from (3.1) that the diagonal of any correlation matrix is not well defined. We therefore set the diagonal entries to be 0. Similarly, if all the introns in the dataset contain the AG and GT consensus there will be two rows and two columns whose χ^2 computations contained divisions by 0. Since we only considered splice sites with the AG, GT consensus, we set the appropriate rows and columns to 0.

Another important trait of our test was the fact that the values in the correlation matrix for a pair of fixed positions depended on the size of the window chosen. This is because introns and exons have finite sizes, so as the window size was increased, the number of introns and exons used in our calculations decreased (for a fixed data set). In addition, since the length distributions of the introns and exons considered

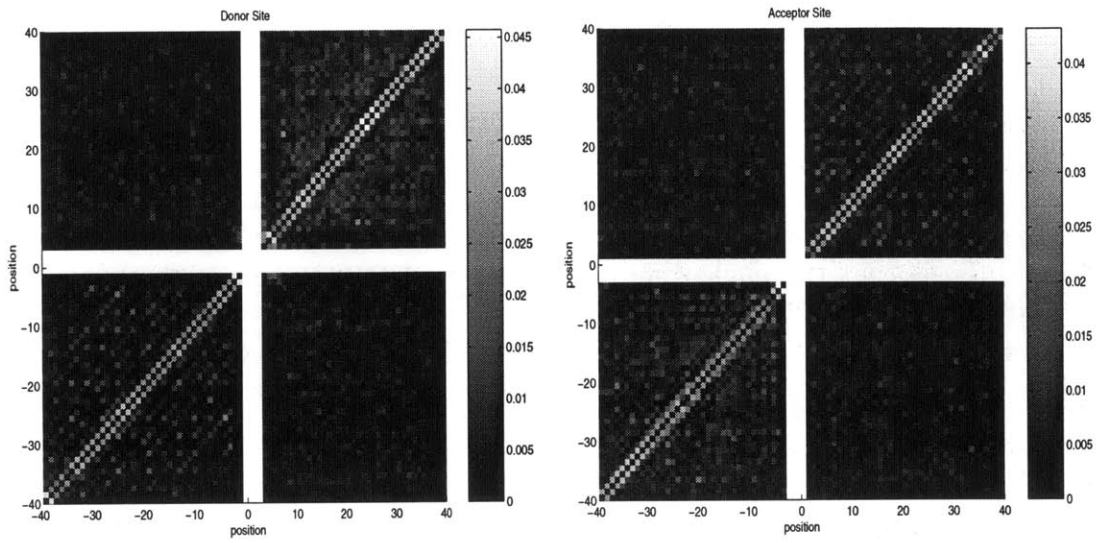


Figure 3-1: Correlation Matrices for donor and acceptor splice sites

changed with window size, we were actually sampling introns and exons with inherently different characteristics. We computed tables for a variety of window sizes, but chose to discuss only the $k = 40$ case in this thesis (and also only the human, as opposed to characteristics in other organisms). More detailed results will appear in follow up work.

As with all Chi-Squared tests for contingency tables, it is important that the values in the contingency table are "sufficiently" large. A commonly used guideline suggested by Cochran [17] is that at least 80 percent of the cells in the contingency table should have counts exceeding 5.0. We observed this to be the case in most of our tests. Nevertheless, for the purposes of our correlation matrices, we decided to use V^2 values rather than p-values computed from X^2 values. This is because we were more interested in the relative strength of correlations rather in an absolute measure of significance. It can be shown [21] that for an $a \times b$ contingency table the X^2 statistic cannot exceed the sample size multiplied by $min(a-1, b-1)$. Thus, the V^2 statistic provides a good measure of relative associations between datasets although it has no simple probabilistic interpretation.

The pair correlations were studied in terms of their applications to splice site prediction. In particular, the strong correlations between non-adjacent nucleotides close to the splicing site were observed to be the only significant correlations between exons and the introns, and so were examined with regards to possible connections to the splicing process. We mention that similar correlations (computed a bit differently) have recently been analyzed by Burge and Karlin [12].

In Table 3.1 we have listed the X^2 values for the matrix between positions -3 and 5 at the donor splice site (the data is from the -40,40 correlation matrix).

Burge and Karlin [12] provide an interesting analysis of the significance of the correlations, concluding that the absence of base pairings with U1 at certain positions is compensated for by base pairings in other positions. We refer the reader to their

0	179	37	0	0	15	38	30
179	0	103	0	0	29	63	70
37	103	0	0	0	12	55	37
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
15	29	12	0	0	0	96	101
38	63	55	0	0	96	0	184
30	70	37	0	0	101	184	0

Table 3.1: X^2 values for the (-3,5) donor window.

paper for a detailed analysis.

The lack of correlation between positions in the intron and the exon away from the splicing site suggest that material in the intron has evolved separately from material in the exon. Indeed, it seems plausible that the intron positions are free to mutate, while the exon positions are constrained by the structural requirements of the protein they code for (this is the basis for the results in Chapter 6). We remark that this lack of correlation can also be used to assist in splice site detection by explicitly measuring (using, say, a χ^2 test), the lack of correlation between positions in an intron and an exon.

We decided eventually to settle on a modified form of the splice site detector used in the GENSCAN program [12] (see Section 3.1.3), rather than scoring splice sites based on pair correlations alone. This decision was marginal, since the schemes do not appear to give vastly different results. The GENSCAN splice site detector has the advantage that it gives a score which has a direct, simple, probabilistic interpretation.

In sections 3.2 and 3.3, we present an overview of some other interesting facts and correlations we observed in our human correlation matrices.

3.1.2 The GENSCAN splice site detector

The splice site recognition method outlined in this section is based on the method by Burge described in [11]. The outline we provide is very vague, the reader should consult the thesis for exact details.

As an example, we consider the donor splice site detector (the method for the acceptor splice site detector is similar, and we refer the reader to [11] for details). Burge uses a technique he calls Maximal dependence decomposition (MDD). The method generates a decision tree as follows:

1. Calculate for each position i around the splice site, the sum

$$S_i = \sum_{j \neq i} \chi^2(C_i, X_j) \tag{3.3}$$

where C_i is the consensus at position i, and X_j is the nucleotide variable at

position j. Intuitively, this measures the amount of dependence between the consensus at a position, and the nucleotides in the local region around it.

2. Choose the value of i at which S_i is maximized, and partition the data into two subsets, the ones with the consensus at position i and the ones without the consensus.

3. Repeat the first two steps, thus building a binary tree, until there is no significant dependencies between the position, or until there is insufficient data for determining probabilities at the leaves.

The exact conditions for termination are described in the thesis. The entire procedure is heuristic in the sense that the decision for branching could be based on different criteria, the subdivision of the data could be different, and the termination conditions are heuristic. Given a sample sequence, a probability for it being a splice site can be obtained by traversing the tree and examining the probabilities at the leaf where it is contained. Figure 3-2 shows the distribution of scores obtained for true donor and acceptor sites, as well as the distribution of scores for "false" sites (sites containing the GT/AG consensus but that are not true splice sites).

Philosophical Note: We experimented with a variety of splice site detectors; indeed there are many available based on many techniques (neural nets, decision trees, heuristics, *etc.*) We developed a splice site detector based on the strict pair correlations described in the previous sections, as well as different metrics of distance between splice sites. Our approaches yielded similar results to the ones obtainable by the MDD detector, although they suffered from the problem that the scores associated with them had no probabilistic interpretation. Thus, one had to resort to a cutoff of the score to select splice sites, philosophically *ensuring* that certain splice sites would be lost, although ensuring good results on average. The MDD detector has the advantage that the scoring is probabilistic, and therefore no cutoffs are necessary in scoring schemes for gene prediction based on splice sites. The cutoff issue has been a recurring dilemma for us in many contexts: while one can improve average results with cutoffs, they ensure that there will be some false negatives.

3.1.3 Left Rules

The separation in distribution between scores (obtained with the `GENSCAN` detector) of true and false splice sites is impressive. Nevertheless, for reasons to be elaborated on in Chapter 6 it is useful to stretch the distributions as much as possible, so as to reduce further the score of true false splice sites.

We noticed a directional "effect" for donor and acceptor splice sites (the effect was also observed for translation initiation sites). The effect is that almost always, the score of a false splice site to the left (upstream in the 5′ direction) of a true splice site is smaller than the score of the true splice site. The effect is directional, because it is much stronger to the left, than to the right (downstream in the 3′ direction). We analyzed the effect for both donor and acceptor splice sites. It is interesting to note that the effect is not very dependent on the type of scoring used for splice sites.

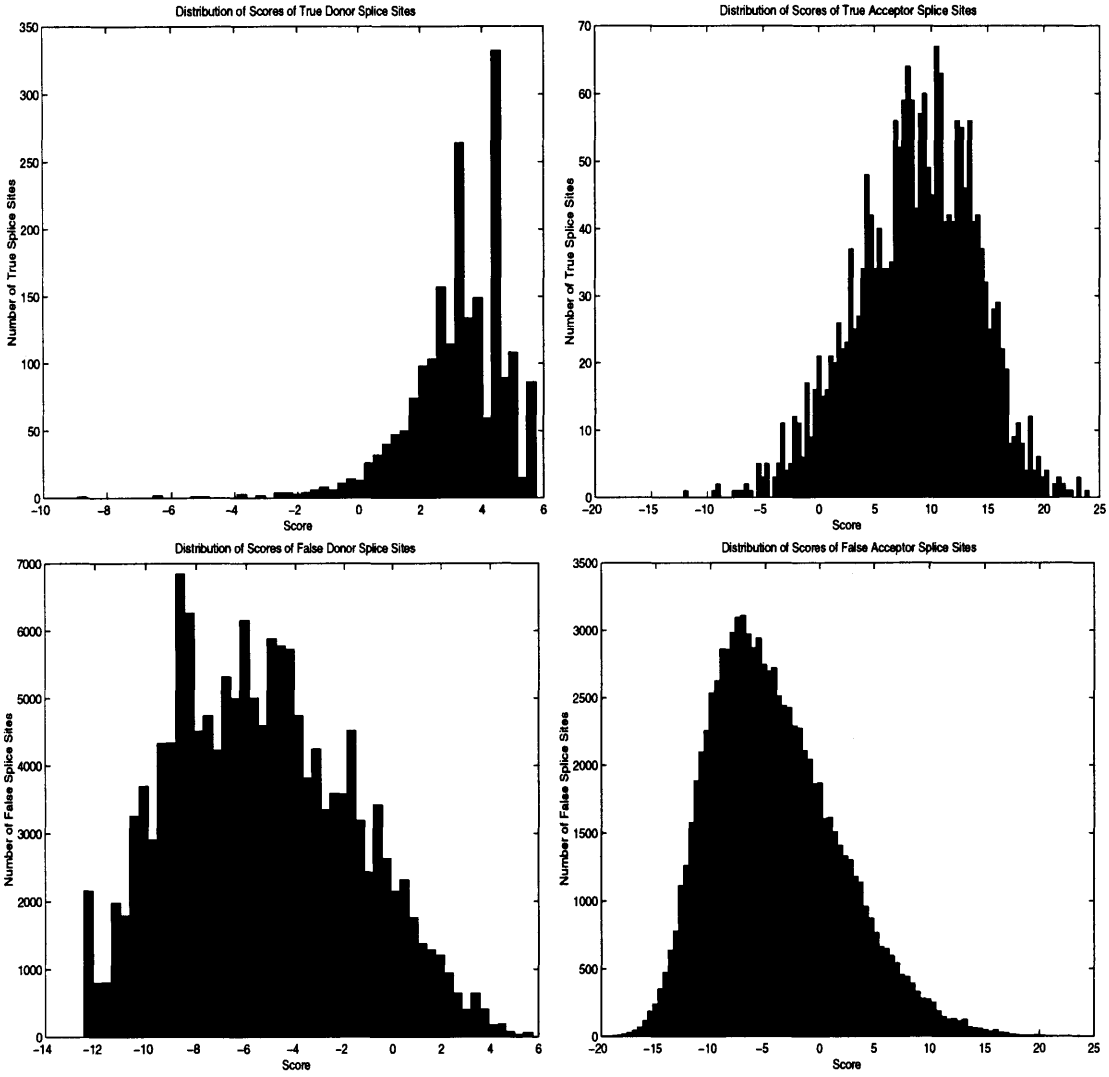


Figure 3-2: Scores of True/False Donor and Acceptor Splice Sites

Figures 3-3 and 3-4 show a detailed analysis of the strength of this effect as a function of distance from a true splice site. For a given parameter d and fixed direction (either left or right), we computed three quantities. First we computed $t(d)$, the total number of splice sites at most d away from a true splice site in the fixed direction, with the additional property that they had no intervening splice sites. Then we computed $b(d)$, the number of such splice sites with a better score than the closest true splice site. Finally, we computed $r(d)$, the ratio $\frac{b(d)}{t(d)}$. For example, Figure 3-3 contains the results computed for donor splice sites. The plots are labeled left and right depending on the direction tested. There is a tremendous difference between the left and right directions in terms of the number of adjacent splice sites with better scores.

A similar effect is evident for acceptor splice sites (Figure 3-4). Interestingly however, the situation is a bit different than for donor splice sites. In particular, there are *very few* potential acceptor splice sites immediately upstream of true acceptor sites.

For both acceptor and donor sites, it seems that there is a significant effect up to a distance of 30–40 basepairs away from the splice site. The directional effect at both splice sites strongly suggests that the splicing machinery has a directionality associated with it.

We can use these effects to enhance the separation between the true and false distributions for splice sites. Consider two neighboring splice sites s_1, s_2 with s_2 following s_1 (that is, s_1 is to the $5'$ side of s_2). We found that if the `GENSCAN` splice site scores of these splice sites were gs_1, gs_2 respectively, then a better separation was obtained by updating the score of s_2 to be gs_2' where gs_2' is defined by

$$gs_2' = gs_2 + MIN(0, gs_2 - gs_1).$$

In other words, we only penalize the score of a splice site, and only if it has a splice site to the left with a better score. Figure 3-5 shows the score distributions for true and false splice sites with the "left rules" added. Presumably, based on the empirical observations detailed above, a more elaborate scheme can be employed to assist in distinguishing between true and false splice sites. We decided to settle on a simple scheme in this thesis to avoid overtraining issues. Future work will include a detailed analysis of the left/right effects, biological verifications, and computational splice site detection schemes techniques based on these results.

3.2 Introns

In this section we describe various features of introns that are of biological and computational interest. We begin with a description of the length distribution of introns (which should be contrasted with the exon length distribution in section 3.3.1). This length information is used in Chapter 6 to assist in distinguishing introns from exons. We then proceed to examine various characteristics of introns that are correlated with the donor splice site. These investigations were motivated by the pair correlation

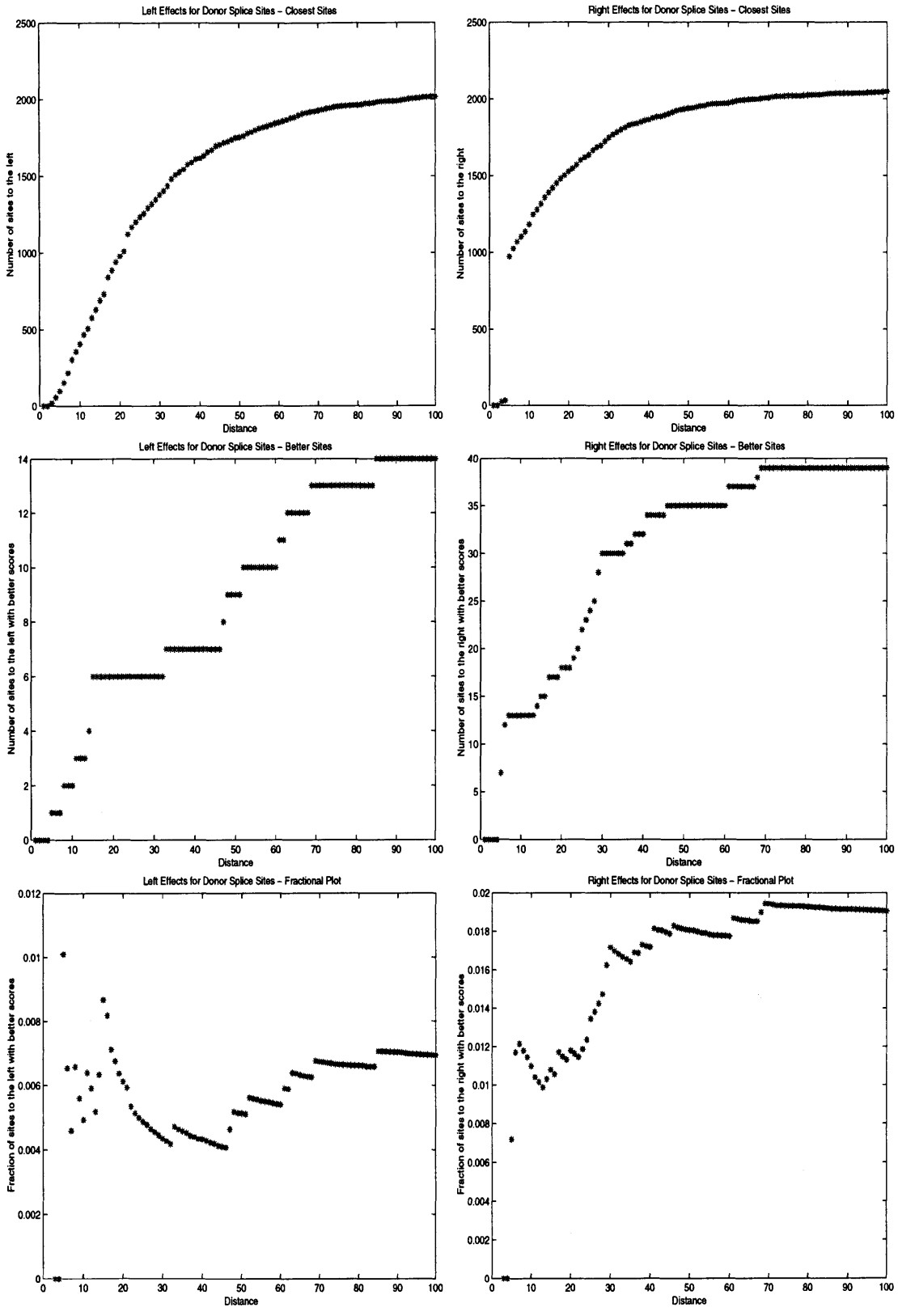


Figure 3-3: Left/Right effects for donor splice sites

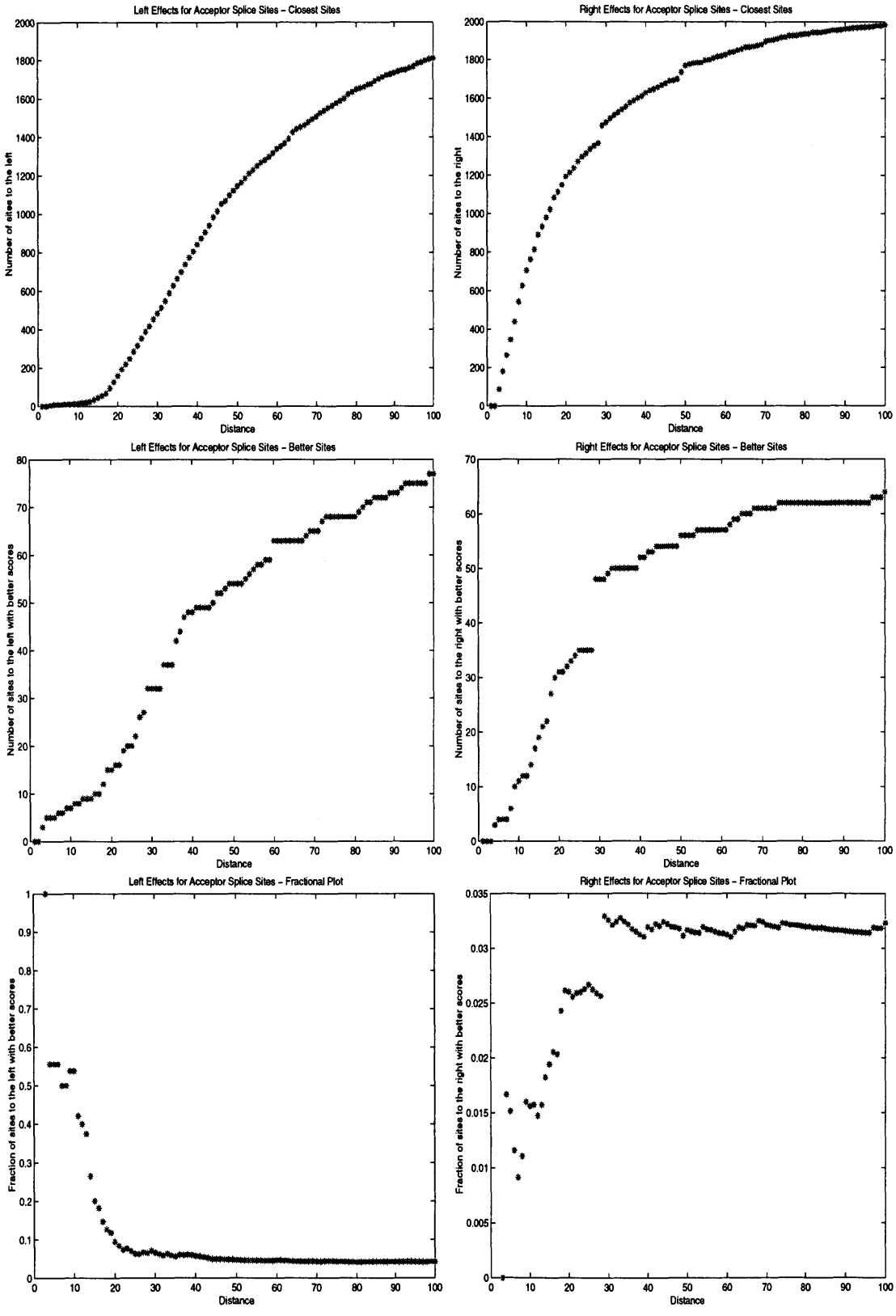


Figure 3-4: Left/Right effects for acceptor splice sites

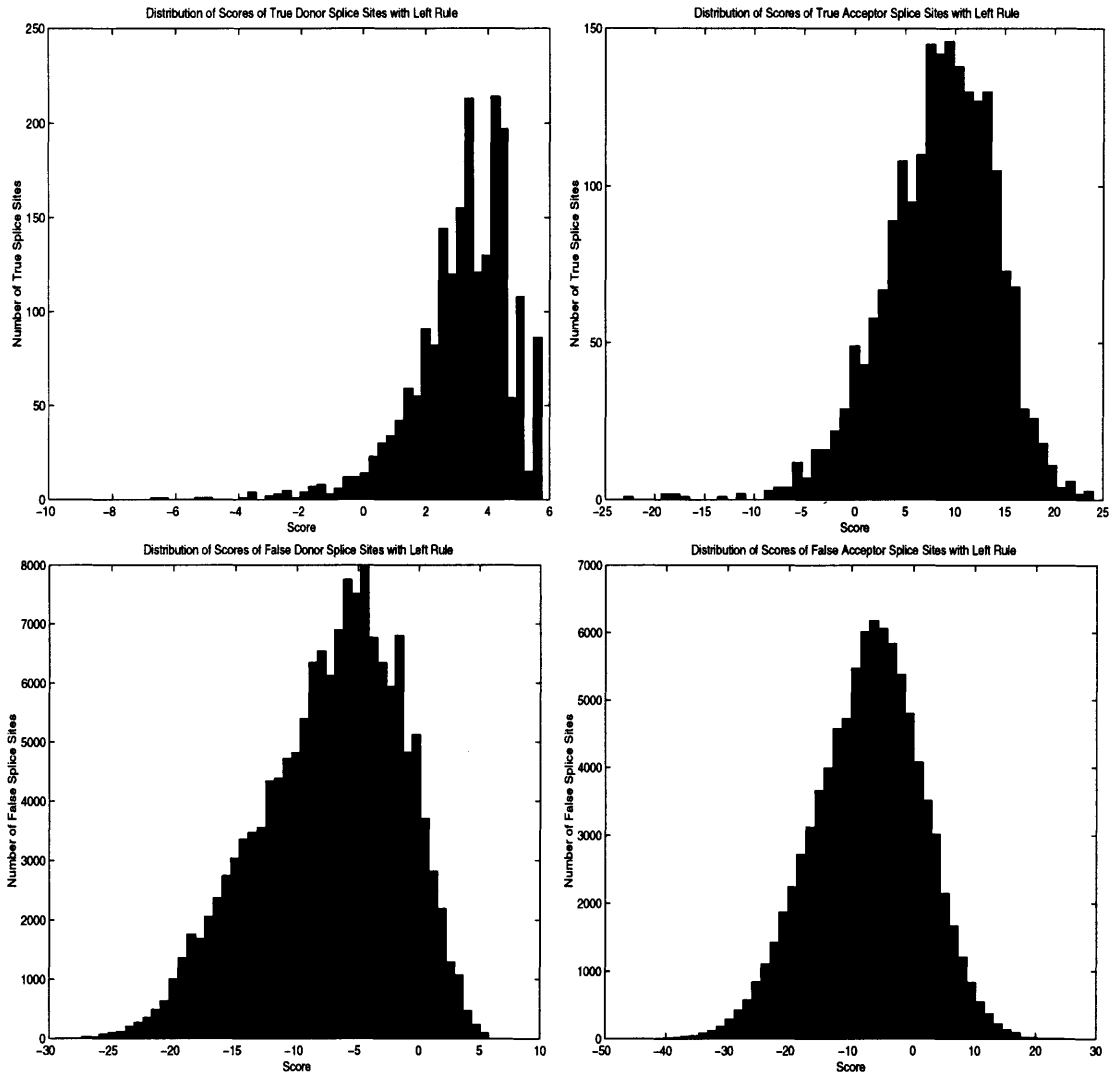


Figure 3-5: Scores of True/False Donor and Acceptor Splice Sites with the Left Rule

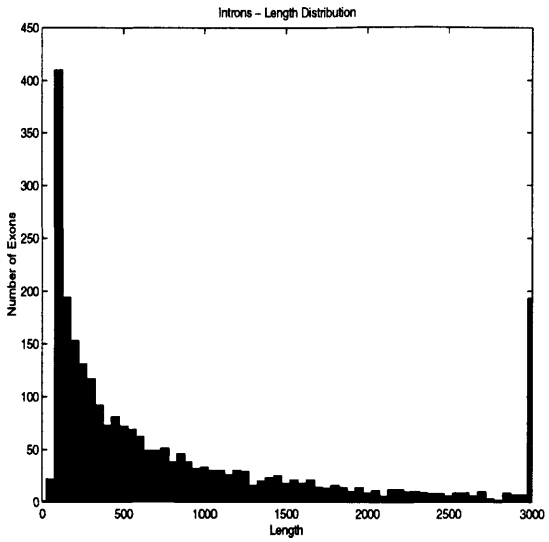


Figure 3-6: Length distribution of introns

study described in section 3.1. The results especially interesting in light of the fact that some of them have been verified experimentally.

3.2.1 Length Distribution

The length distribution of introns in human genes has been well documented [35]. We include a plot of the distribution of intron lengths, only for the sake of completeness and for reference. The distribution is of interest to us, mainly because it is different from the distribution of exon lengths (section 3.3.1).

3.2.2 Pair Correlations in Introns

Looking at the correlation matrices, we found significant correlations between almost all positions in introns. These correlations are due to the high G+C content of many genes, and the fact that this abundance of G+C is different for different genes in the same organism (discussed in more detail in the next subsection). Thus, the appearance of a G or a C in some position is an indicator that the entire intron is G+C rich. Strangely, we did not observe this phenomena in all organisms (for example, plants).

Also of interest is the fact that position +3 showed a correlation with distant positions while positions +4 and +5 did not. This means that positions +4 and +5 are severely constrained by splicing requirements, whereas position +3 is not! In fact, for any position $+k$ ($k > 5$) in the intron, a G in position +3 correlates with a G or C in position k and an A or T in position +3 correlates with an A or T in position k. So it seems that position +3 is constrained to be an A or G, but that in cases when the rest of the consensus is strong, it is essentially determined by the overall G+C content of the gene. Burge and Karlin [12] have noticed that when the overall consensus is strong position +3 tends to be a G rather than the A which would complement the

Correlation Coefficient	
Intron	Exon
0.76	0.64

Table 3.2: The correlation coefficient for GC content between an intron (exon) and its neighboring introns and exons.

RNA in the U1 spliceosome. It seems that this is simply due to the fact that most human genes tend to be G+C rich.

Strong correlations all around the area from positions +1 through +30 reveal that many important splicing interactions may be occurring in that region. Specifically, a very strong interaction between positions +23 and +24 suggests a possible splicing component involved with those specific sites. Adding credence to this observation is the fact that correlations (23,26), (21,24), (22, 25) and (21,25) are uncharacteristically *low*, which is significant because of the G+C effect mentioned above.

As a result of these observations we conjecture that there is a signal which is constrained only to a region (not a position), much like the branchpoint at the 3′ ends of introns. This signal is probably located somewhere between positions +22 and +26. We suggest there is a weak consensus for this signal although it seems that a G is recognized. The G is probably preceded by a T or a G.

Even more interesting is the fact that these biases disappeared when the entire GENBANK primate database was tested, suggesting that there might be a human specific splicing component (or else that the correlations are an artifact of the dataset we tested on).

3.2.3 G+C effects

The multitude of correlations in the intron can be attributed, in large part, to the GC isochore structure of the human genome [9]. Specifically, introns tend to appear in GC rich and GC poor flavors, and indeed, genes in general exhibit overall GC richness/poorness. As discussed previously, this effect is evident in the pair correlation test, which tests for independence between positions. If a nucleotide in an intron is a G, it is likely that another nucleotide, even far upstream or downstream in the same intron, is a G or a C. The relative lack of correlation between positions in the intron and positions in the exon is an indicator that the GC content of an intron is not as good a predictor of the GC content of its flanking exons, as it is for the adjacent introns (see Table 3.2). The fact that correlations in the intron are partly due to the GC effect, and the observation that the 3rd position of the donor splice site exhibits such correlations, led us to examine the relationship between the 3rd position and neighboring nucleotides. Figure 3 summarizes the findings. Even though it is apparent that the 3rd position is an indicator of GC content, the abundance of G's near the splice site when G appears in position 3 suggests there is more involved than just the GC effect.

	GC poor		GC rich	
	G_3	A_3	G_3	A_3
GC content	0.43	0.39	0.60	0.59
G triplets	0.69	0.34	1.54	1.48
Length	1531	1644	568	580
> 2 triplets length	2121	2337	643	568

Table 3.3: GC content and the number of G triplets.
A_3 and G_3 denote an A in position 3 and a G in position 3 respectively. > 2 triplets length is the average length of introns that contained more than two G triplets between positions 5 and 25.

3.2.4 G triplets near the donor splice site

Based on reports of many G triplets near donor splice sites, McCullough and Berget [63] set out to experimentally test the hypothesis that the triplets are involved in splicing, and not merely an artifact of GC rich introns. The intron they tested (the second one in the human α-globin gene) was short (129bp), in a GC rich intron (although other introns in the gene were GC poor) and happened to have a G in position 3. They found that the number of G triplets next to the donor site additively enhanced splicing of the intron (this is somewhat of a simplification, the reader should consult the paper for exact details). Interestingly, they found that G triplets help in splicing in the absence of a strong pyrimidine tract at the acceptor splice site. Indeed, improvement of the tract with contiguous uridines suppressed the requirements of G triplets for maximal splicing. Nevertheless, the G triplets still seemed to be involved in donor splice site selection. The authors strongly suggest that it is the short length of the introns that is necessitating this additional splicing element, and that there may be a class of short introns which require numerous G triplets for correct splicing.

Having observed that the number of G triplets near the donor splice site was directly correlated with position 3, we computationally investigated whether length, or perhaps other factors were responsible for the numerous G triplets in some introns. The data set was divided according to the GC content of the introns. **GC rich** introns were defined to be introns with at least 50% G's and C's. **GC poor** introns were defined as those with less than 50% G's and C's. The number of G triplets was calculated within each of these data sets, further divided according to the nucleotide in position 3. The **number of G triplets** was defined to be the number of G triplets between positions 5–25 at the beginning of the intron. Figures 3-7, 3-8, and 3-9 indicate many of the relationships we found. Table 3.3 summarizes additional parameters, such as length, that were calculated for the various subclasses.

Figure 3-8 show that the number of G triplets near the donor splice site is more strongly correlated with the overall GC content of the gene, than with the nucleotide in position 3. One must however take into consideration the fact that the nucleotide in position 3 is strongly correlated with the overall GC content of the intron. Indeed, the

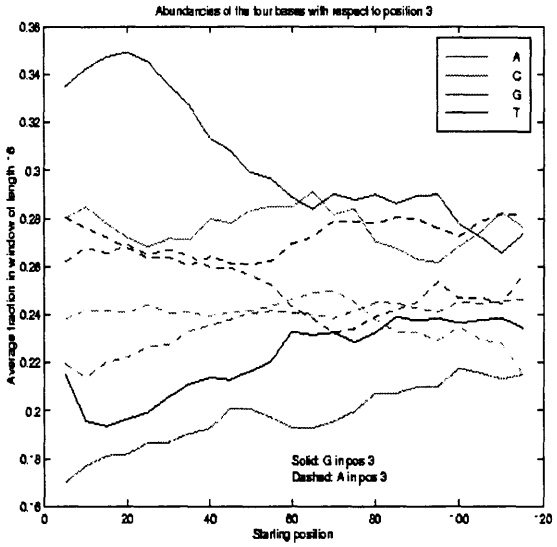


Figure 3-7: The effect of the nucleotide in the third position an bases downstream Notice the large number of G's around position 20 when there is a G in position 3.

plot in Figure 3-9, smoothed using local least squares regression, shows the correlation between overall GC content and the nucleotide in position 3. An A in position 3 is weighted 0 and a G is weighted 2 (C and T are weighted 1 and 3 respectively, however, the small number of sequences containing them makes their effect negligible). Considering again the information in Figure 3-8, we see that in GC poor introns, the role of G triplets becomes important when G appears in position 3. This is not at all the case in GC rich introns, where a G in the third position does not seem to affect the number of triplets by much. This suggests that a G in the third position is especially detrimental when the rest of the intron is not GC rich, and in such cases G triplets are a necessity.

3.3 Exons

Our main result concerning exons is a new procedure for identifying exons based on a differential frame test. This is described in section 3.3.3. We proceed to discuss the problem of frame identification in coding exons. These results are of practical importance in the exon prediction problem discussed in chapters 5 and 6. In particular, our frame prediction technique is applied in chapter 5.

3.3.1 Length Distribution

The distribution of exon lengths is interesting because unlike introns, the distribution appears normal and not exponential. Furthermore, terminal, initial and internal exons all have very different distributions, a fact that can be used to our advantage.

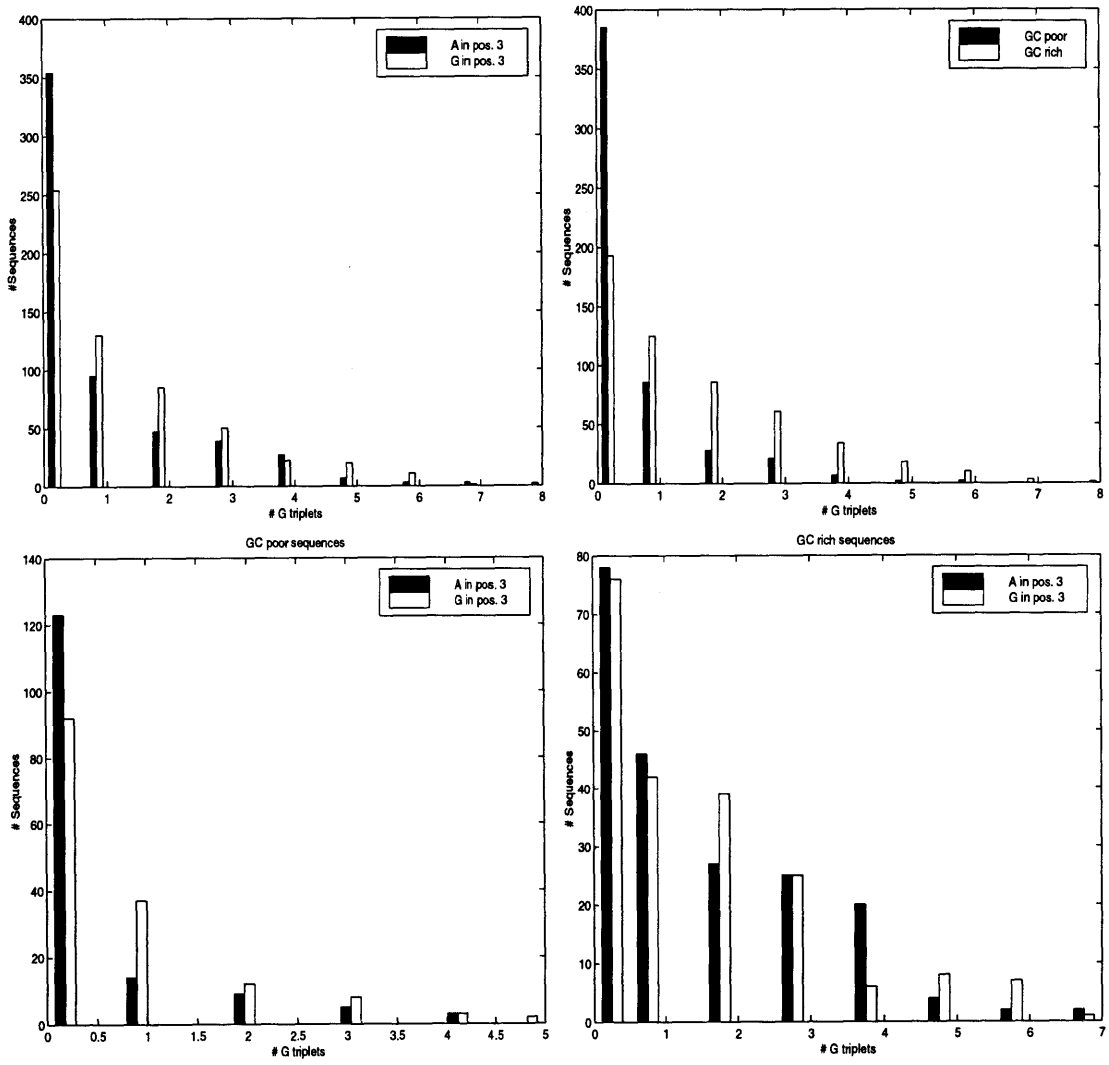


Figure 3-8: G triplets, GC content and Position 3

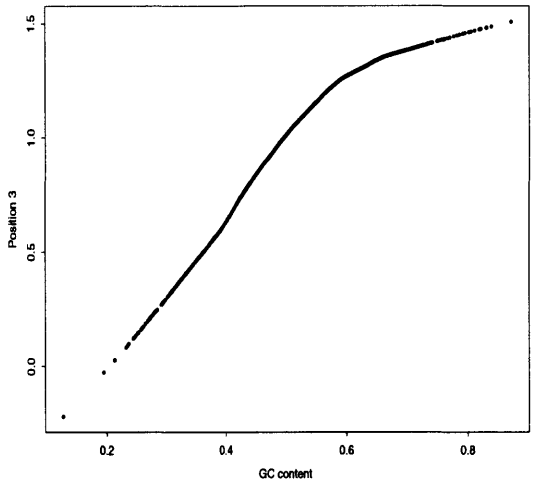


Figure 3-9: GC content and Position 3

3.3.2 Pair Correlations in Exons

Our correlation matrices provide immediate evidence of codon bias in genomes [43, 45, 44, 46], and also clearly reveal the triplet nature of the genetic code. Notice the correlations within exons along the off-diagonals, spaced distance three apart. It is interesting that the correlations do not appear to weaken significantly at large distances.

Of interest to the splicing problem, is the fact that we observed a correlation between positions -29 and -26 and somewhat of a correlation between positions -29 and -8 before the donor splice site. These correlations by themselves do not stand out as particularly significant however Chiara et. al. [15] have found experimental evidence that two spliceosomes (SRp20 and SRp30) interact with exons between positions -31 and -26.

3.3.3 The Frame

Frametests

It is well known that certain codons are overrepresented (or underrepresented) in the genes of different organisms [46]. Indeed, there are even substantial differences between organisms [44]. These biases have been used extensively in gene recognition programs.

We study the following two related problems:

- How can the frame of an exon be predicted?
- How can exons and introns be distinguished based on the tuples that occur in exons?

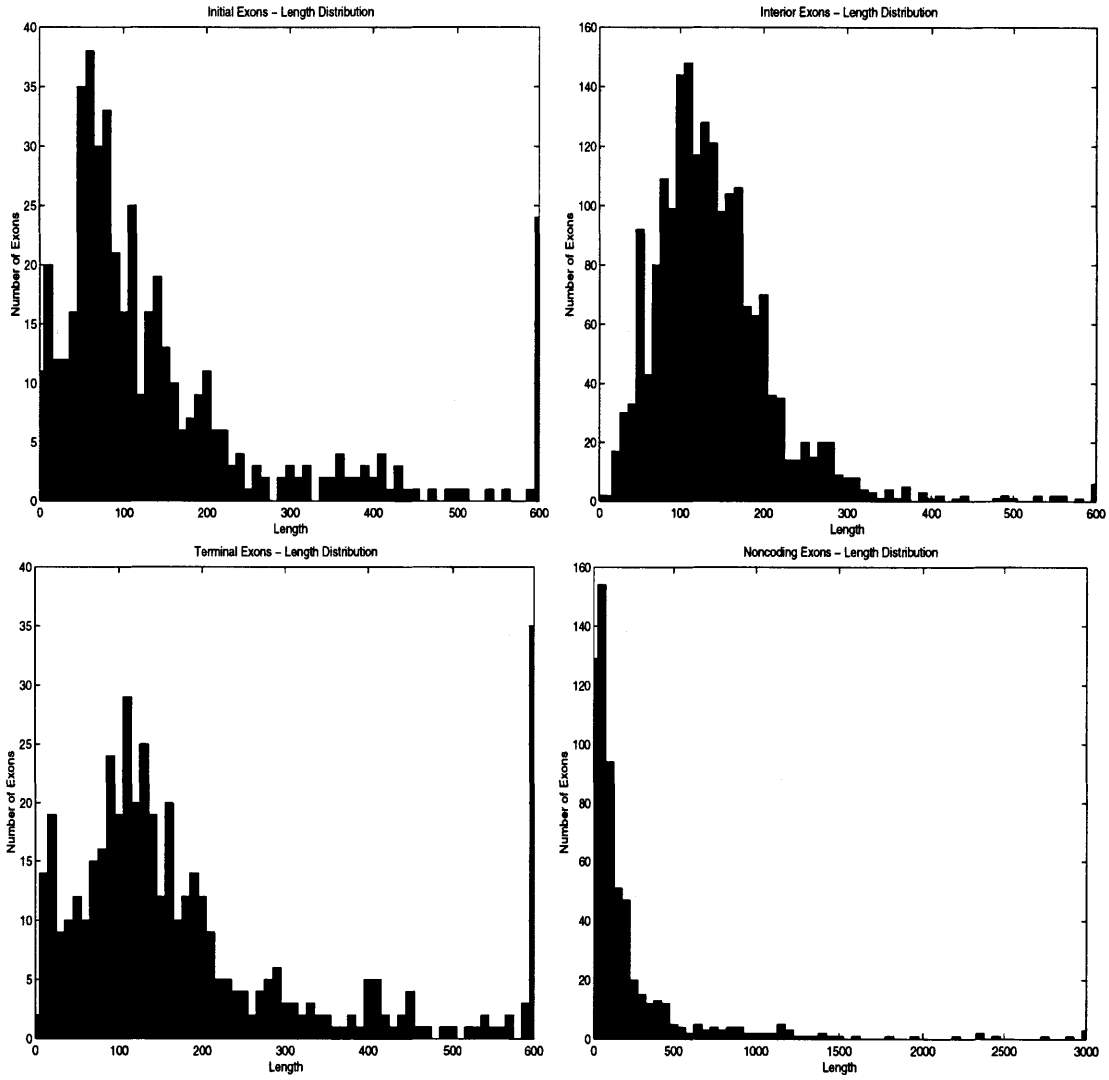


Figure 3-10: Length distributions of coding and noncoding exons in genes with multiple exons

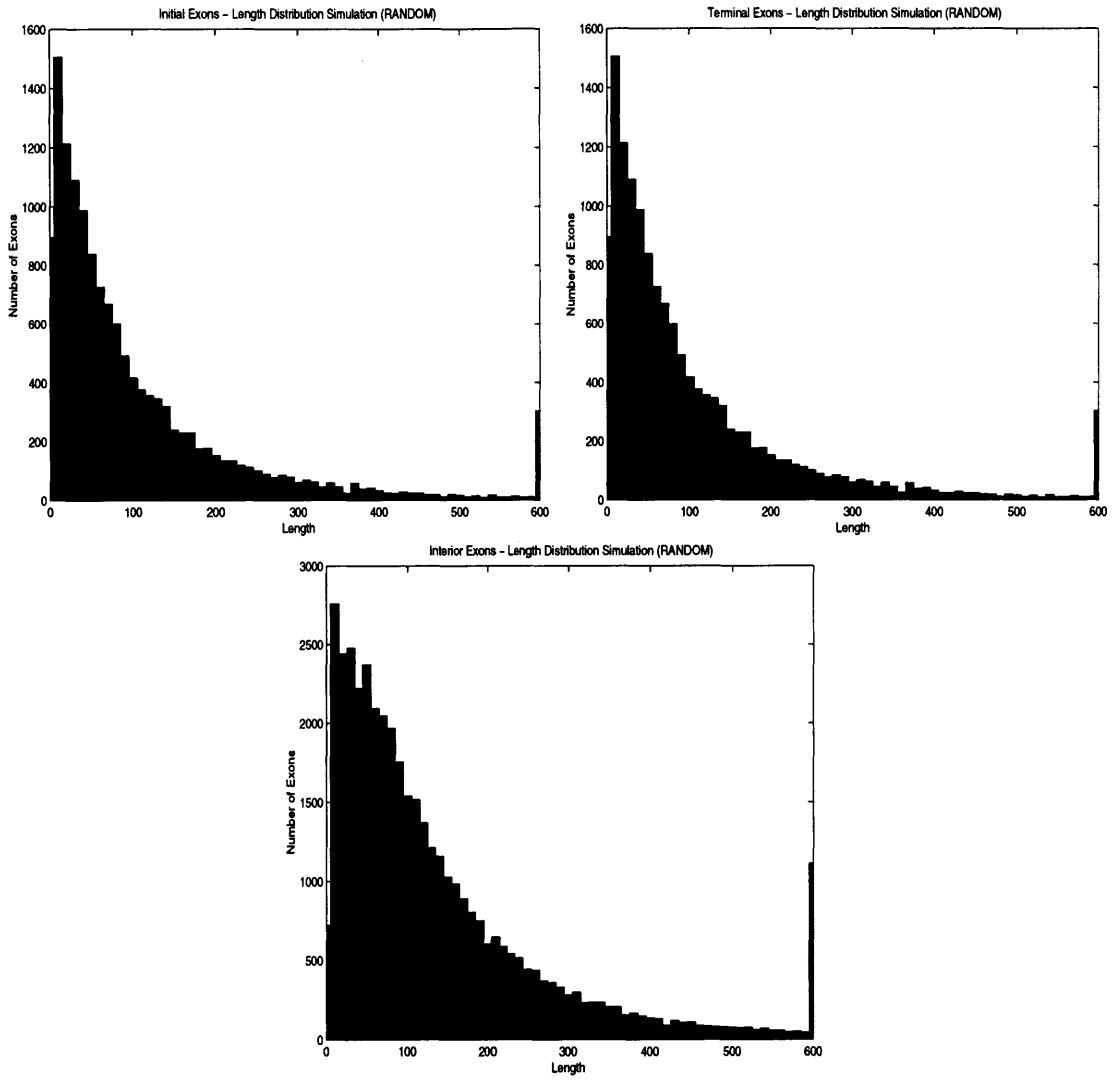


Figure 3-11: Length distributions of simulated exons in multiple exon genes

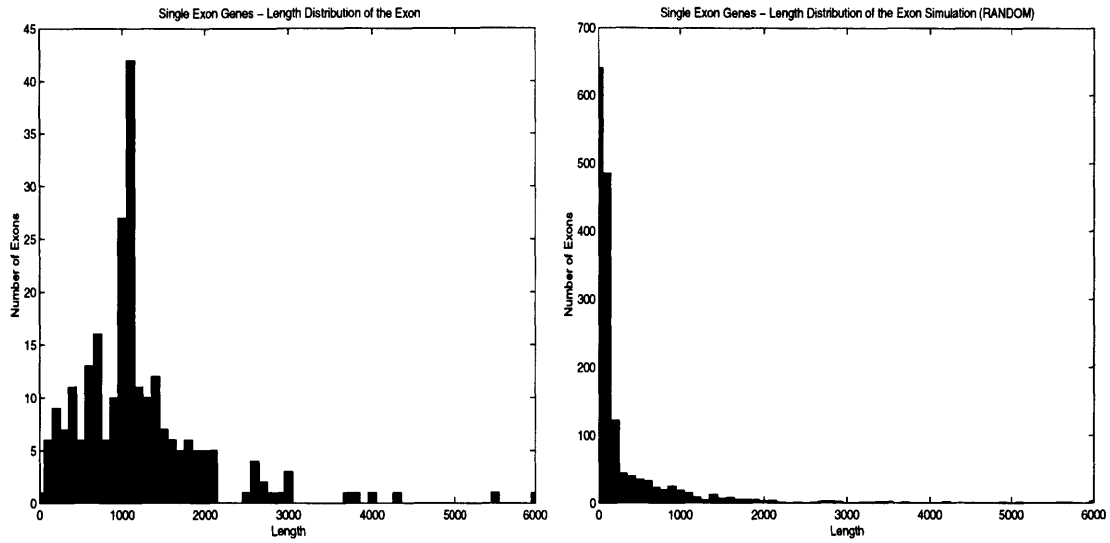


Figure 3-12: Length distributions of true and simulated exons in single exon genes

The solution of the above problems is possible because of the empirical observation that k-tuples are not uniformly distributed between the true frame and the shifted frames of an exon (for example, stop codons never occur in frame except in the last coding exon).

We begin with some notation and definitions:

Definition 3.1 *The absolute frame of a tuple in a coding exon is defined to be the position* mod3 *of the first nucleotide of the tuple in the exon. The absolute frame of an exon is defined to be the absolute frame of 3-tuples that are codons.*

Let t be a k-tuple. For a given learning set of coding DNA, define t_i $(i \in (0,1,2)$ to be the relative frequency of occurrence of t "shifted by frame i". That is, for every exon E in the learning set, t_i is computed by

$$t_i = \frac{\sum_E \sum_{j \in E} \chi_j(t)}{Total_i(t)}$$

where j ranges over the positions in the exon, and $\chi_j(t) = 1$ if the tuple t begins in position j and $(j-i)$ mod 3 is the absolute frame of the exon, $\chi_j(t) = 0$ otherwise, and $Total(t)$ is the total number of positions j in coding exons E satisfying $(j-i)$ mod 3 is the absolute frame of E. Define t_I to be the relative frequency of occurrence of t in a learning set of introns. For convenience, in any exon E, let t^r denote the tuple beginning at position r.

We define a frametest F to be a pair of maps $T : E \to R^3$ and $S : R^3 \to R$ where E is a sequence of genomic DNA. The idea is to construct functions T that for a given region, return three numbers which represent the "score" of the region in each of the three possible frames (under the assumption that it is coding). This information can already be used to guess the frame of the exon, by selecting the frame with the best score. In order to distinguish exons from introns, another function S

is constructed that uses the three frame scores to determine one score for the region. The function S can be used to base the test on codon usage bias (reference) (*e.g.* taking the maximum of the three numbers), or on the fact that coding exons tend to have one frame look more exonish as opposed to the other two (*e.g.* taking the difference between the largest and smallest score).

A number of different frametests were investigated. For a given exon E, the following tests were considered ($T : E \to (p_1, p_2, p_3)$, $S : (p_1, p_2, p_3) \to R$):

1. $(F = \#1)$ $p_i = \sum_{r \in E} (\log \frac{t^r_{r-i \bmod 3}}{t_I})$, $S = MAX(p_0, p_1, p_2)$.
2. $(F = \#2)$ $p_i = \sum_{r \in E} \frac{1}{t^r_{r-i \bmod 3}}$, $S = MIN(p_0, p_1, p_2)$.
3. $(F = \#3)$ $p_i = \sum_{r \in E} (t^r_{r-i \bmod 3} - t_I)$, $S = MAX(p_0, p_1, p_2)$.
4. $(F = \#4)$ $p_i = \sum_{r \in E} (t^r_{r-i \bmod 3} - t_I)$, $S = MAX(p_0, p_1, p_2) - (p_0 + p_1 + p_2)$.
5. $(F = \#5)$ $p_i = \sum_{r \in E} (t^r_{r-i \bmod 3} - t_I)$, $S = MAX'(p_0, p_1, p_2) - (p_0 + p_1 + p_2)$.
6. $(F = \#6)$ $p_i = \sum_{r \in E} \frac{t^r_{r-i \bmod 3} - t_I}{t_0 + t_1 + t_2}$, $S = MAX(p_0, p_1, p_2)$.
7. $(F = \#7)$ $p_i = \sum_{r \in E} \frac{t^r_{r-i \bmod 3} - t_I}{t_0 + t_1 + t_2}$, $S = MAX(p_0, p_1, p_2) - (p_1 + p_2 + p_3)$.
8. $(F = \#8)$ $p_i = \sum_{r \in E} \frac{t^r_{r-i \bmod 3} - t_I}{t_0 + t_1 + t_2}$, $S = MAX'(p_0, p_1, p_2) - (p_1 + p_2 + p_3)$.

The function MAX is the maximum and the function MAX' is the maximum where $p_i = -\infty$ if the exon E has a stop codon in frame i.

Testing

Tests were conducted by constructing two data sets, one consisting only of coding exon and the other only of intron. The coding exons from the HKR dataset (Appendix B) were glued together (with stop codons of the genes removed) to make one large open reading frame of length about 250,000. Similarly, the introns were glued together, although only a portion of roughly the same length as the total coding exon material was constructed. The same data was used for learning and for testing, although to remove the cheating that would be caused by this, a "minus one" option was implemented. In particular, for each test sequence considered, the tuples were removed from the learning table (t_0, t_1, t_2), except in the case when one of these entries was zero or one, in which case the entry was left unchanged if there was no stop codon in frame. For a fixed length L and frametest F, the coding exon segment and intron segment were each tested with F across every region of length L. The results were used to compute two histograms H_E and H_I for the respective segments. The separation $s(L, F)$ between the histograms was then calculated by the formula

$$s(L, F) = \frac{2A(H_E \cap H_I)}{A(H_E) + A(H_I)}.$$

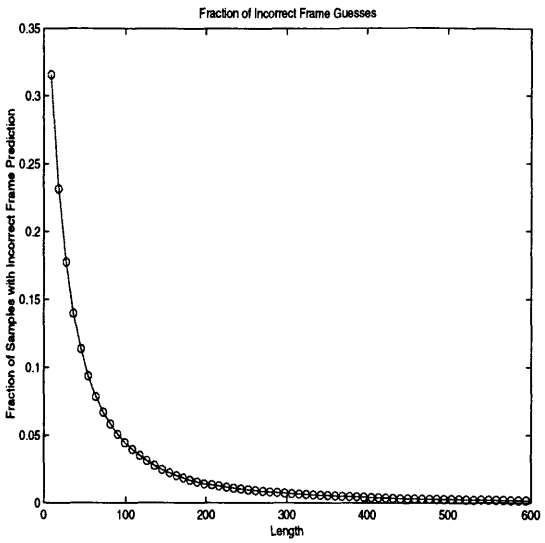


Figure 3-13: Frame Prediction

Optimal Frametest	
Length	Frametest
9–81	$F = \#6$
81–162	$F = \#8$
162–360	$F = \#7$
360 –	$F = \#3$

Table 3.4: The Best Frametests

where A(R) is the area of the region R.

Frame prediction was tested in a similar fashion, by examining the number of incorrect frame guesses using a particular function T from a frametest.

Results

The best function (in comparison with the functions defined above) for frame prediction was obtained by using the function $T : E \to R^3$ defined by

$$p_i = \sum_{r \in E} \frac{t^r_{r-i \bmod 3} - t_I}{t_0 + t_1 + t_2}.$$

and guessing the frame to be f where $p_f \geq p_i$ for $i \in \{0, 1, 2\}$. Figure 3-13 shows the performance of this method: Remarkably, the method above was the best for all lengths. Tests based on $F = \#1$ and $F = \#2$ performed only marginally worse.

The situation for frametests proved to be much more complicated. Depending on the lengths of the exons, different frametests are optimal: The results are depicted in Figure 3-14. Analysis of the plots is interesting: Notice that for short lengths,

the best frametest is obtained by taking S to be a MAX. In other words, the best frametest score is based on setting the score *to be* the score of the best frame. On the other hand, for slightly longer exons, the best test is derived from MAX', which is based on the *differential* between the different frames. Also interestingly, the best test for exons of length 81–162 is much worse than the best test in the region 9–81. For longer exons, once again MAX seems to be better, and normalizing at each step by the sum relative frequencies of the tuple in each frame is no longer better. The difference between the tests becomes much smaller as one examines longer exons. Indeed, *any* test is good for long exons, which is probably why many programs have little difficulty in identifying such exons. The fact that different tests are good for different lengths, and that the more sophisticated tests are in general better than the standard $F = \#1$, proves that there is no panacea, and that in this case the simplest test (with the easiest probabilistic interpretation), is not the best.

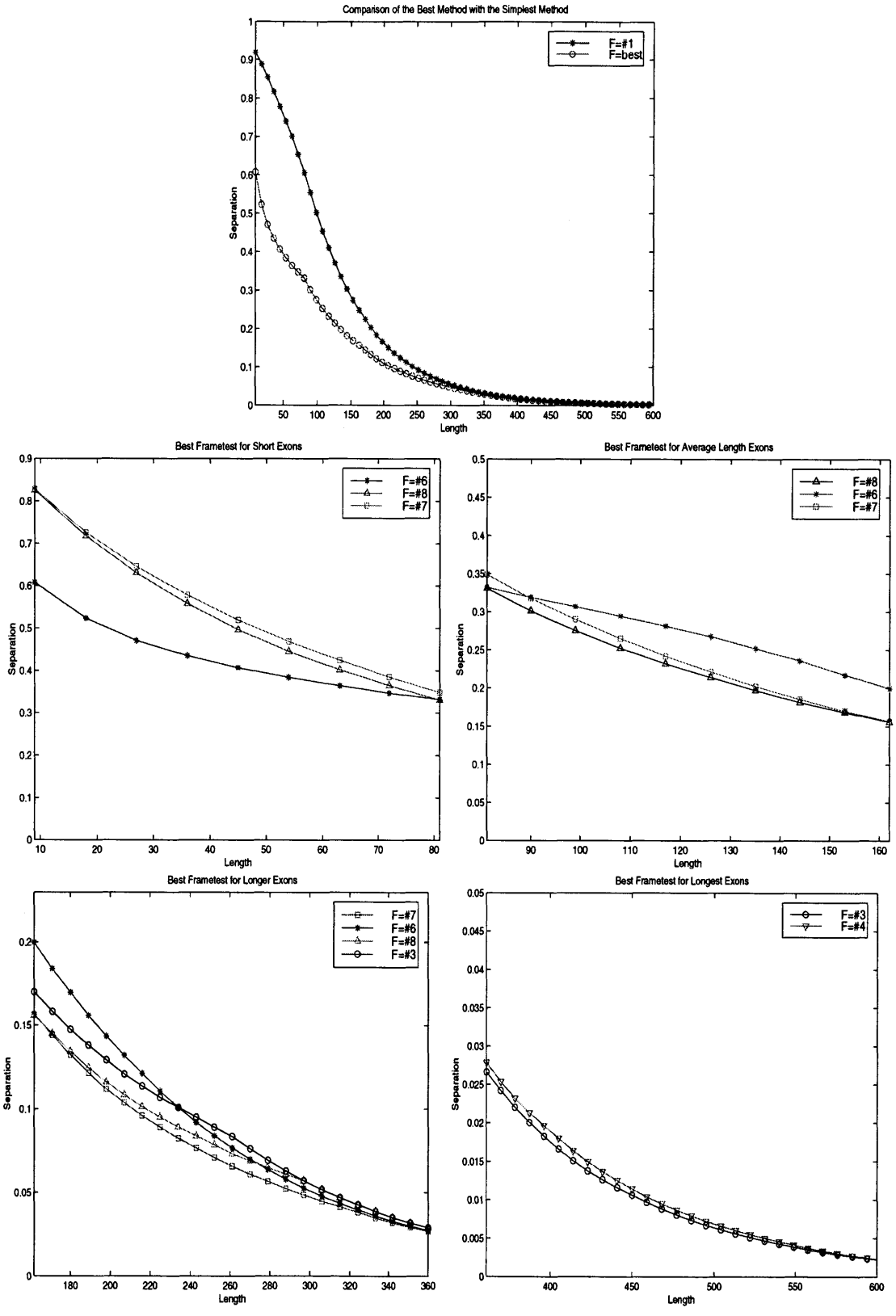


Figure 3-14: Separations for the different Frametests

Chapter 4

Assembling a Parse

4.1 Introduction

The difficulty in accurately predicting splice sites means that for every true donor splice site, approximately ten incorrect ones must be considered. The situation is even worse for acceptor splice sites (where the ratio is about fifteen splice sites for every true one). The number of parses that must be considered is therefore very large. In this chapter we first give an upper bound on this number (which is exponential in the number of sites that need to be considered). In practice, many parses can be discarded because of stop codons in frame. Still, even for a small gene, the potential number of parses is still enormous. We use dynamic programming to overcome this difficulty. Our approach is detailed at the end of this chapter.

4.2 Complexity of the Problem

4.2.1 A visit with Fibonacci

It is clear that the number of parses of a gene increases dramatically with an increase in the number of potential donor and acceptor splice sites. An exact analysis of the number of parses of a gene with n potential acceptor sites and m donor sites is in fact possible; the answer is, surprisingly, related to the Fibonacci numbers!

We model the problem as follows:

Definition 4.1 *Given a string S of open (and closed) brackets, let $p(S)$ denote the number of parses of S. A parse of S is a sequence of alternating open and closed brackets (not necessarily consecutive).*

In our model, the open brackets represent potential acceptor splice sites, and the closed brackets represent donor splice sites.

Theorem 4.1 *Let S be a string with n open brackets and m closed brackets.*

$$p(S) \leq F_{n+m+1}. \tag{4.1}$$

where F_{n+m+1} is the $(n+m+1)$th Fibonacci number.

Proof: We will use the notation $|S|$ for the length of a string S of open and closed brackets. Assume that the theorem is true by induction for $|S| \leq n-1$. Let S be a string with $|S| = n$. Observe that S must have an open bracket somewhere that is followed immediately by a closed bracket. Otherwise we have that $p(S) = 1$ (the empty parse is considered to be a parse). We thus have $S = S_1()S_2$ where S_1 has k open and r closed brackets respectively, and S_2 has $n-k-r-s-2$ open and s closed brackets respectively. Now notice that

$$p(S) \leq F_{r+k+1}F_{n-k-r} + F_{r+k+2}F_{n-k-r-1} + F_{n-1} - F_{r+k+1}F_{n-k-r-1}. \tag{4.2}$$

This is obtained by taking any parse in S_1 and appending to it a parse in the string "(S_2". Then one adds in parses in "S_1)" together with parses in S_2. Parses omitting the pair () are counted twice but one of these counts corresponds to all parses with the pair (). The last two terms correct for parses whose last element in S_1 is an open bracket and whose first element in S_2 is a closed bracket.

Using the Fibonacci identity (for an elegant proof see Chapter 6)

$$F_{n+m} = F_{n+1}F_m + F_nF_{m-1}, \tag{4.3}$$

we have

$$p(S) \leq F_n + F_{r+k+1}F_{n-k-r-1} + F_{n-1} - F_{r+k+1}F_{n-k-r-1} = F_{n+1}. \tag{4.4}$$

The bound is attained for certain configurations. For example, the string $S = ()()\cdots()$ with n open and n closed brackets has $p(S) = F_{2n+1}$.

4.2.2 Average case analysis

When examining real genes it is clear that we will not necessarily be dealing with strings S that conform to the worst case scenario in terms of the number of parses. Unfortunately, an average case analysis reveals that even random strings will have an exponential number of parses.

Theorem 4.2

$$\sum_{|S|=n} p(S) = \frac{1}{2}(3^n + 1) \tag{4.5}$$

Proof: Let $a_n = \sum_{|S|=n} p(S)$. We will establish that $a_n = 3a_{n-1} - 1$. The strings of length n can be partitioned into two classes, those that start with ")" and those that start with "(". The strings starting with ")" contribute (by induction) a_{n-1} to a_n since the first bracket cannot be used. Strings beginning with "(" contribute a_{n-1} when the first bracket is not used. When the first bracket is used, it must eventually be matched up with a closed bracket. Suppose there are i brackets between the first

bracket and this closed bracket. The i brackets can be chosen arbitrarily contributing a factor of 2^i. The remaining brackets after the closed bracket contribute a total of a_{n-2-i} total parses. Thus we have

$$a_n = 2a_{n-1} + \sum_{i=0}^{n-2} a_i 2^{n-2-i}, \tag{4.6}$$

from which the recursion $a_n = 3a_{n-1} - 1$, and hence the theorem, follow.

Theorem 4.2 immediately implies

Corollary 4.1 *The average number of parses of a string of length n is*

$$\frac{1}{2^{n+1}}(3^n + 1). \tag{4.7}$$

Indeed, this shows that on average the number of parses of a string of length n is almost as large as in the worst case.

4.2.3 Mitigating factors

The number of possible parses for an actual gene with n potential acceptor splice sites and m potential donor splice sites is, in practice, actually far lower than the above estimates. The reason for this is that many parses can be eliminated for biological reasons. These may include the existence of a stop codon in frame, or an extremely unlikely long exon produced by the parse. Indeed, this suggests that for short genes, an exhaustive enumeration of the parses is a reasonable idea.

4.3 A Dynamic Programming Approach

4.3.1 General Framework

Definition 4.2 *A valid parse is defined to be a subdivision of a gene into coding exons and introns satisfying the following requirements:*

- *The coding region begins with the codon ATG.*
- *The coding region ends with a stop codon (TAG, TGA, and TAA).*
- *All donor splice sites contain the GT consensus. All acceptor splice sites contain the AG consensus* [1].
- *The frame between adjacent coding exons is consistent.*

Definition 4.3 *A partial valid parse is a subset of a valid parse. That is, a partial valid parse is a subdivision of a subset of a gene into coding exons and introns, and thus the first coding exon is not required to begin with ATG, nor is the last required to end with a stop codon.*

[1] This requirement can be relaxed, as discussed in Chapter 6

noncoding $\rightarrow$ | $\longleftarrow$ Exon 1 $\longrightarrow$ | $\longleftarrow$ Intron 1 $\longrightarrow$ | $\leftarrow$ Exon 2 $\rightarrow$
noncoding $\longrightarrow$ | $\longleftarrow$ Exon 3 $\longrightarrow$ | $\longleftarrow$ Intron 2 $\rightarrow$ | $\leftarrow$ Exon 2 $\rightarrow$

Figure 4-1: The reason for frame consistent dynamic programming

Notice that a partial valid parse may actually be a valid parse (if it contains an initiation and termination codon).

We applied dynamic programming [10] in a variety of contexts (Chapters 5,6) to find "optimal" valid parses which maximized the sum of scores of exons in a parse. We experimented with a variety of scoring schemes for exons ranging from the frametest (Chapter 3), to comparative based scores described in Chapter 6. Formally, our dynamic programming optimized the function

$$f = \sum_{e} s_1(e) + \sum_{i} s_2(i).$$

where e ranges over positions in exons in the parse, and i ranges over positions in introns. The functions s_1, s_2 assign scores to positions in the exons and introns respectively. We found (empirically) that assigning exon scores was more beneficial than assigning intron scores or scores to both. The particular scoring functions we used were based on both heuristics (Chapter 5), and more formal probabilistic scores (Chapter 6). In general, we found that heuristics performed best in most situations, and were to be embraced rather than avoided in the context of gene recognition. Perhaps a reason for this is the multitude of exceptions that arise in the sequences (due to rare biological phenomena), and thus probabilistic scoring schemes based on statistical assumptions are often invalid because the assumptions are violated.

4.3.2 Frame Consistent Dynamic Programming

To guarantee that a parse is optimal, it is necessary to keep track of the best parses *in every possible frame.* Consider the example illustrated in Figure 4-1 (from Salzberg in [77]). Consider the ends of introns 1 and 2 (which are the same). At this position, it is necessary to save the values of the best parses ending in the three different frames in the previous exons. This is because, even though exon 1 might score better than exon 3, it may be that exon 1 is not frame consistent with exon 2. Thus, our dynamic programming keeps track of the best parse in every frame.

4.3.3 Technical Issues

The dynamic programming algorithm described above is quadratic in the number of splice sites.

We found that precomputing information necessary to parse the genes was both fast and efficient. For example, we precomputed a "stop matrix" which allowed us to do an $O(1)$ lookup to determine whether there was an open reading frame in a given

region. Clearly such a precomputation can be done in time linear in the length of the sequence.

Chapter 5

Dictionary Approaches

5.1 Introduction

In this chapter we describe a dictionary based method to scoring potential exons, using the dynamic programming approach described in Chapter 4. The dictionary method is a technique for rapidly identifying matches of an input sequence to sequences in a protein or cDNA database and has many additional applications beyond exon prediction; we outline some of these in the latter sections of this chapter.

Previous targeted approaches developed for exon prediction suffer from the lack of integration of statistical and homology based approaches. Our approach generalizes many of the previous approaches, and maintains the flexibility of adding further signals into the computations (such as promoters). Furthermore, the speed of our techniques enable us to look for alternative splice sites, and we can also consider other applications which have not been investigated to date.

The method is best described as applied to a particular database. Using the nonredundant protein OWL database [96], our dictionary consists of 4 tuples of amino acids, and for each, the protein sequences in OWL which contain it. We can construct such dictionaries very quickly from the OWL and dbEST databases [95]. We used the OWL dictionary to find the longest common subsequences of length at least k (for any $k \geq 4$) between sequences in OWL and the translation of the genomic sequence under investigation. A similar approach was applied to the dbEST database, after orienting the dbEST database [84] and eliminating non-coding exon regions. Given such information, we used our dynamic programming algorithm to produce a parse of the gene into introns and exons (Chapter 4). There are a continuum of scoring schemes for the exons based on frequencies of occurrence in the database of subsequences of each size.

We tested our method on two of the benchmark data sets described in Appendix B. On the HKRM data set we found exons with 88% nucleotide sensitivity and 99% specificity. Our exon sensitivity/specificity is 81%/82%. When testing on the BG data set we found nucleotides with 82%/97% sensitivity/specificity. Our exon sensitivity / specificity was 73%/75%. These results were obtained after removing sequences from the database with exact amino acid homology to genes in the data set.

5.2 Methods

5.2.1 Dictionary Lookups and Fragment Matching

A central component of our gene annotation approach is the **fragment matching problem**. That is, given a gene and a database (for example the dbEST database), we would like to find all the matches of length above some threshold between the gene and the database. This is a classic string matching problem, and there are linear-time algorithms for it. The problem with such an approach is that the size of the databases we are interested in matching against precludes the possibility of real time computation. Instead, we do some precomputation on the database so that we do not have to look at all the sequences in the database whenever we are looking up the matches for a particular gene.

Dictionary Construction

The data structure we precompute is a **dictionary**. Conceptually, the idea is to record for each tuple (either of DNA or of protein, depending on the dictionary one is building), the list of sequences in the database in which it appears.

Formally, a **dictionary** is based on a "plain" sequence file, consisting only of accession codes (identifying codes) and corresponding sequences of strings, from an alphabet of size 4 for DNA and size 20 for proteins. A **tuple** is a sequence of length 11 for DNA sequences and length 4 for protein sequences. A **hit** is a match between some segment in the input sequence and a **target sequence** in the database.

Sequences and tuples are indexed by integers for the purpose of lookups in the dictionary. The dictionary is organized into six components which collectively enable the following operations to be performed in $O(1)$ time:

- Find a sequence given its number.
- List all the sequences that contain a given tuple.
- Find the accession code of a sequence from its number.

Finally, the accession code of a sequence can be used to find the sequence number in $O(\log n)$ time using binary search. The first two sequence and tuple lookup functions are used by the end-user. The last function is a helper utility for enabling the $O(1)$ lookups.

Two dictionaries were constructed, one from the OWL database [96] and another from the dbEST database [95]. The more difficult step was constructing the tuple lookup table, *i.e.*, for each tuple, a list of the sequences in which it occurs. The naive approach to building the dictionary would be to first construct a matrix indexed by tuples and the accession numbers of the sequences. For instance, in the case of the dbEST database this would be a 4 million by 1 million entry binary matrix (the OWL database consists of roughly 250,000 sequences). The entries of the matrix would be flagged according to which tuples occur in which sequences, and the dictionary would

be built by reading off the entries for each tuple. Unfortunately, this would require too much space (terrabytes) or time (trillions of operations).

Instead, the dictionary was constructed by sorting pairs of sequence/tuple identifiers by the tuple coordinate. Specifically, every occurrence of a tuple in a sequence was recorded as a pair (s_i, t_i), where the s_i's are integers ranging from 1 to the number of sequences, and the t_i's are integers ranging from 1 to the number of tuples. The list was originally ordered by the sequences from the database. A linear time **radix sort** [20] was used to sort the list according to the second coordinate. The large number of pairs necessitated that the large list was sorted in pieces and then merged at the end. The size of the individual pieces to be sorted was set as a command line parameter, so that the dictionary construction could be tuned to take full advantage of the memory of our machine. The final list of sequence/tuple pairs ordered by tuple numbers was used to look up the sequences in which a specified tuple occurs in $O(1)$ time.

Using the Dictionary to Find Matches

The dictionaries were used to quickly find exact matches of subsequences between a given input sequence and a database. This information was used to compute hits. For a given hit, the following information was returned:

- The position in the input sequence where the hit began.
- The length of the hit.
- The accession number of the target sequence.
- The position in the target sequence where the hit began.

Returned hits correspond to longest segments in the input sequence that matched segments of each target. These hits were also required to be longer than a threshold k. The first three pieces of information were useful for obvious reasons. The position in the target sequence was used to determine if nonadjacent hits in the input sequence corresponded to consecutive segments in the target (thus indicating the presence of an intron).

This information was computed in two phases. In the first phase, the dictionary was used to find, for each tuple appearing in the input sequence, the list of target sequences containing that tuple. The input sequence was then scanned from the beginning to the end to find all segments longer than k that contained tuples from the same target sequence in the database. This resulted in a list of candidate segments.

A second phase was necessary to ensure that the tuples in these candidate segments were actually consecutive in the database sequence which they matched. This was accomplished by loading the database sequence for each candidate segment, and then finding the longest subsegments of the candidate segment appearing in the loaded database sequence. This final procedure was divided into two substeps. The first consisted of building a mini-dictionary, used to return in $O(1)$ time a list of the positions in the sequence where a given tuple occurred. This dictionary was then

used in the second step in a manner analogous to the first phase described above to scan through the candidate segment to find all subsegments consisting of *consecutive* tuples in the database sequence.

5.2.2 Dynamic Programming

We used the dynamic programming approach described in Chapter 4. Our assumption was that our input is genomic DNA from one gene only.

The score of a parse was defined to be the sum of the scores of the exons. For example, one simple score for a potential exon was computed by

$$score(exon) = \sum_{p} f(p)$$

where p ranges over all the positions in an exon. The function $f(p)$ was defined by

$$f(p) = \begin{cases} 1 & h(p) > 8 \\ -1 & 4 < h(p) \leq 8 \\ -2 & h(p) < 5 \end{cases}$$

where $h(p)$ was the length of the maximal hit at position p. Other scoring schemes were tested, but we found the heuristic scheme described above worked well.

When using the OWL database, the sequence was converted into protein in all three possible frames to look for matches (computed with a threshold of $k = 5$), and it was this information that was used to ascertain the frame of a potential exon.

Parses were constructed by first determining all potential splice sites. This was done using modified versions of the WAM, WWAM and MDD techniques described in Burge and Karlin [12]. For details see Chapter 3. The potential splice sites were then used to dynamically construct a parse using the scoring scheme described above. In addition, repeats were masked (see section 5.3.4), reducing considerably the locations available for exons.

Tests were conducted on the BG and HKRM datasets (for a detailed description of these datasets see Appendix B) The tests were performed once using the entire database, and once where sequences in the database matching all of the exons in the input sequence were removed.

5.3 Results and Discussion

5.3.1 Output of the Program

The program gives a list of maximal exact matches of length at least k, which is a parameter one may choose, in the database in question in various frames. The results depend on k and some matches may occur in introns and some in exons. Typically, those in introns tend to be scattered and it is usually easy to distinguish them from those in exons. Table 5.1 contains an example output obtained for the Id3

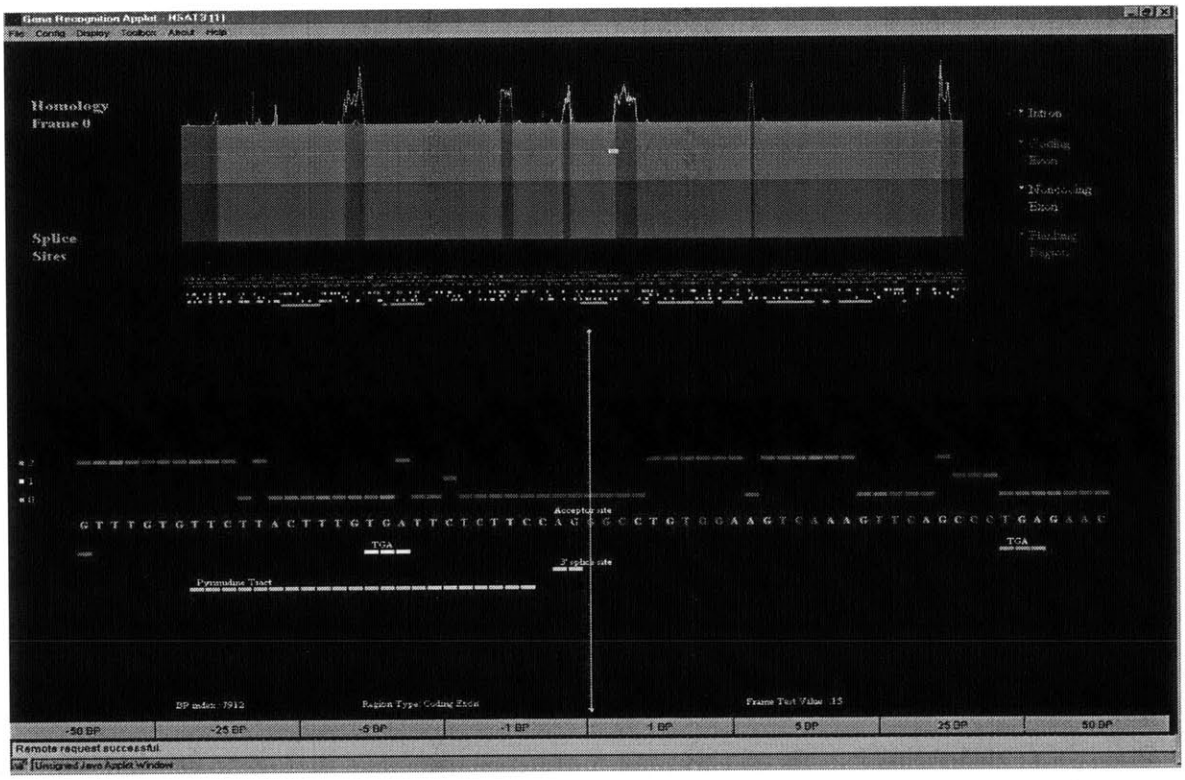


Figure 5-1: Java applet display

The sequence is displayed in a diagram on top. The darker interior regions correspond to coding exons. The number of hits to the OWL database at every position is displayed in the graph above the diagram. Subsequences of the actual sequence can be displayed below together with a variety of computed annotations. These include: potential splice sites, pyrimidine tracts, stop codons (with different colors corresponding to different frames), tuples much more common in introns than exons, repeats, as well as tuple frequencies in exons with respect to frame.

gene (an HLH type transcription factor, GENBANK Accession: X73428) using the OWL dictionary with a minimum threshold of $k = 8$. The number of hits returned was 46. A corresponding table for $k = 7$ contained 440 hits. The table for $k = 6$ contained more than 8000 hits. The frame of a hit was defined by its position in the DNA sequence *modulo 3*. Note that positional information in the target sequence has been omitted. The annotated structure of the gene in GENBANK is represented in Figure 5-2.

Figure 5-1 contains a screenshot of a java applet used to visualize genes, as well as various types of information including the the dictionary hits described above.

5.3.2 Alternative Splice Sites

Our methods are particularly amenable to showing where exons can be read in different frames. For example, consider the Id3 gene illustrated in Figure 5-2.

Using the protein dictionary, we found that the second exon matched in two

Position in the given sequence	Length in nucleotides	Frame	Locus in the OWL database
171	27	0	Y338 MYCGE
687	24	0	S43230
738	90	0	ID3 MOUSE
738	90	0	ID3 RAT
927	24	0	I51316
927	24	0	I51278
954	27	0	ID2 HUMAN
954	27	0	ID2 MOUSE
954	27	0	ID2 RAT
954	27	0	ID4 HUMAN
954	27	0	ID4 MOUSE
954	27	0	JC2007
954	27	0	A41689
954	27	0	AF049135
954	27	0	OMID2PROT
954	27	0	HSU 16153
957	24	0	HUMID2X
990	24	0	A27280
738	300	0	ID3 HUMAN
837	201	0	ID3 RAT
840	198	0	ID3 MOUSE
1053	57	0	S71404
1152	24	0	CELF21H11
738	480	0	S71405
1128	90	0	S71404
1506	24	0	D89624
1671	24	0	PABL STRGR
1953	24	0	Y4EF RHISN
2310	24	0	AE0006769
438	24	1	MVIM SALTY
438	24	1	STYFLGA5
810	27	1	SCBLACABL
954	24	1	GUN1 TRILO
954	24	1	GUN1 TRIRE
1239	24	1	AF041044
1239	24	1	AFO41045
1548	24	1	CYAA SACKL
1596	24	1	DMU08282
1839	24	1	IMH1 YEAST
2283	24	1	S49247
2283	24	1	BTU04364
126	24	2	YC26 PORPU
210	24	2	CEC50F44
1146	60	2	ID3 HUMAN
1716	24	2	CEB03655
1785	24	2	AF005632

Table 5.1: OWL hits returned with a minimum length cutoff of $k = 8$ amino acids.

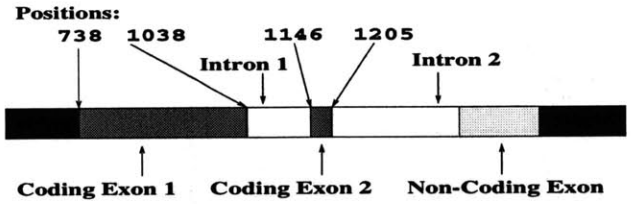


Figure 5-2: The Id3 gene.

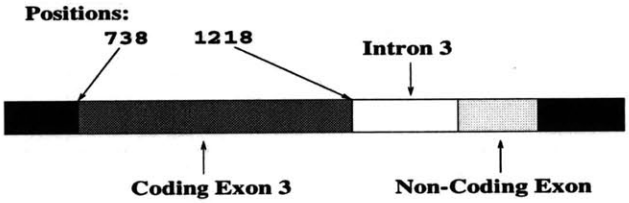


Figure 5-3: An alternative form of the Id3 gene.

different frames. There is an alternatively spliced form of the gene which is shown in Figure 5-3. In coding exon 3, the entire first intron is used to make protein, and the fact that its length is not divisible by 3 means that coding exon 2 is expressed in a different frame. The alternatively spliced version occurs only 10% of the time [24]. The algorithm we used to find exons in genes, given the protein matches, can be forced to select either alternative and return both answers, thus automatically identifying an alternative splice site. This particular example also illustrates the difficulty in finding a universal "good" target. Even though the alternatively spliced version of the gene is somewhat rare, `BLAST` reports it as a better match to the whole gene because the expressed protein is longer. Furthermore, analysis of the above gene with the `dbEST` dictionary revealed coding exons 1 and 2, and also the noncoding exon, but the alternatively spliced variant was not evident. Thus it can be useful to analyze genes using different databases.

Table 5.2 contains a list of all the genes we found in the Burset–Guigó data set that contained exons that matched proteins in two distinct frames. The criteria used was that the candidate exon had to have two segments (each at least 24 base pairs long) overlap it in two distinct frames, with the overlap between the segments and the exon being at least half the length of the exon. Furthermore, the overlap between the two segments was also required to be half the length of the exon. The candidate exon was at least 30 base pairs long. The strict criteria were chosen to ensure that the overlapping hits observed in the selected exons were statistically significant.

5.3.3 Exon Prediction

The results of tests using the entire `OWL` protein database are reported in Table 5.3 for the Burset–Guigó and Haussler–Kulp–Reese data sets. The sensitivity and specificity are based on the definitions in Burset and Guigó [13]. The results of tests with exact matches of each test sequence removed from the `OWL` database are also shown in Table 5.3. The results for the Burset–Guigó runs may have been affected by the presence

Locus	Description	Exon Position
HS1D3HLH	Id3 gene	1146 – 1205
HSCYP216	CYP21 gene	2036 –2214
HSDAO	diamine oxidase gene	4856 – 6425
HSGROW2	germ line for growth hormone	1602 – 1799
HSMT1H	MT1H gene	1409 – 1474
HSPRB3L	PRB3L gene	2087 – 2916
HSPRB4S	PRB4 gene	2213 – 2793
HSPSAG	DNA for prostate specific antigen	1688 – 1847
HSU12421	mitochondrial benzodiazepine receptor gene	3684 – 3872
HUMADAG	adenosine deaminase gene	35100 – 35202
HUMCP21OH	21-hydroxylase B gene	2728 – 2906
HUMCP21OHC	mutant 21-hydroxylase B gene	2729 – 2907
HUMGHN	growth hormone gene	1827 – 2024
HUMGHV	growth hormone variant gene	1834 – 2031
HUMLYTOXBB	lymphotoxin-beta gene	3559 – 3630
HUMMCHEMP	monocyte chemotactic protein gene	1472 – 1589
HUMMET2	metallothionein-II gene	1167 – 1232
HUMMET2	metallothionein-II gene	1436 – 1527
HUMNTRI	neutrophil peptide-1 gene	2627 – 2801
HUMNTRI	neutrophil peptide-1 gene	3382 – 3491
HUMNTRIII	neutrophil peptide-3 gene	2627 – 2801
HUMNTRIII	neutrophil peptide-3 gene	3382 – 3491
HUMPRCA	protein C gene	10516 – 11105
HUMTHROMA	thrombopoietin gene	5053 – 5718
LEBGLOB	Lepus europaeus adult beta-globin gene	2492 – 2620
MMGK5	Mouse glandular kallikrein gene	1645 – 1804
OAMTIB	Sheep metallothionein MT-Ib gene	1655 – 1720
OAMTIC	Sheep metallothionein MT-Ic gene	1229 – 1294

Table 5.2: Genes from the Burset–Guigó database with exons expressed in two frames. Unless otherwise specified, the genes are human.

Data Set	Nucl. Sn.	Nucl. Sp.	Exon Sn.	Exon Sp.
BG (matches removed)	82	97	73	75
BG	93	97	87	86
HKRM (matches removed)	88	99	81	82
HKRM	97	99	92	91

Table 5.3: Statistics for the `OWL` protein database.
BG=Burset–Guigó, HKRM=Haussler–Kulp–Reese, modified.

of non-human genes. The parameters for the program were calibrated on a human training set.

The power of combining dictionary hits with a gene recognition program is emphasized by the following statistic (computed with the removal of exact matches of test sequences from the database): Out of the 10% of intron positions covered by matches of 8-tuples to the `OWL` database, only 0.5% were predicted to be in exons. Only .05% of the total intron base pairs were incorrectly classified as coding.

The results in Table 3 compare favorably with other statistical methods. Estimates for sensitivity and specificity per nucleotide position range from 60–90%. Predictions of exact exons also vary between the programs, with estimates between 30–70% specificity and sensitivity. The homology based `AAT` approach predicts nucleotides with a sensitivity of 94% and specificity of 97%, and exons exactly with a sensitivity of 74% and specificity of 78%.

The quality of our results is, of course, directly related to the presence or absence of related matches to our test genes in the database. The larger the minimum tuple length threshold k, the more the results become dependent on the general redundancy of the database. For genes with few matches, one can resort to a smaller tuple size and take the *number* of hits into account. Thus, our approach can be tuned to work either as a statistical method or as a homology based method, as well as all the hybrids in between. Furthermore, as the size of the databases grow, the results can be expected to improve.

5.3.4 Other Applications

The dictionary approach we have described lends itself to a number of other applications:

- Repeat masking: We have built dictionaries from repeat databases and used these for rapidly finding repeat segments in genes. This technique provides an alternative to alignment based repeat maskers such as `RepeatMasker` (A. F. A. Smit and P. Green [97]). The method is especially useful for exon prediction, where it is advantageous not only to mask complete repeats, but to mask segments (perhaps from repeats) that do not occur in exons.

- Different tuple patterns: The construction of the dictionary can be based on

arbitrary tuple patterns and does not need to be restricted to consecutive tuples. Such patterns may be important biologically. For example, the third position in codons is less conserved in exons than the other two, so a pattern skipping every third position may lead to interesting results. Another example is the Kozak consensus (Chapter 1) for translation starts, which involves positions -3 and $+4$ around the ATG (recall that the consensus is AGXX**ATG**G).

- Pseudogenes: Reverse transcribed genes which lack introns are often pitfalls for gene recognition programs. The identification of neighboring exons in inconsistent frames with no room for an intron immediately suggests the presence of a pseudogene. This can be easily checked and automated, in the same vein as the alternative splicing detection. Indeed, we discovered two such examples in a newly sequenced genomic segment (`GENBANK` Accession: AC001226).

5.3.5 Discussion

The dictionary approach has a number of advantages over standard similarity search techniques. Despite the unprecedented success of alignment algorithms in biology, the algorithms are all handicapped by the problem that short matching segments can be difficult to find in certain cases [8, 94]. For example, Figure 5-4 illustrates a case where two subsequences of a large sequence agree in small regions (dark areas) and differ elsewhere. An alignment between them may be overlooked if mismatches are penalized less than gaps. Such a penalty scheme will produce an alignment of the two subsequences with each other where the dark regions are not aligned, rather than the desired alignment where the dark regions are superimposed. Even though this problem can be addressed by suitably modifying the alignment parameters, the number of such extreme examples, combined with the myriad of parameters necessary to address biological phenomena involved in sequence evolution (*e.g.,* `BLAST` [3]), creates a fundamental difficulty. Every given problem has a set of parameters associated with it that provides a "good" alignment, but there is no universal set of parameters that works for every problem. An advantage of our dictionary method is that when performing a database search, *all* the exact matches of segments in a sequence are rapidly detected.

The `FLASH` program is also designed to find similarities of segments to sequences in databases. Unlike our dictionary method, which involves $O(1)$ time lookups for tuples of a given size (4 tuples for protein dictionaries and 11 tuples for DNA dictionaries), the `FLASH` program relies upon storing the *positions* of shorter tuples in a hash table. The use of larger tuple sizes in a dictionary renders this unnecessary because longer tuples appear in fewer sequences. The advantages of our dictionary method become apparent when the databases involved become very large, which is the case with the current `dbEST` database. Furthermore, this method enables rapid calculation of frequency counts of arbitrary length tuples, which can be applied to statistics based programs that rely on such information.

This approach also has advantages over other exon prediction methods. The entire process of database search and exon prediction is automated, and the prediction

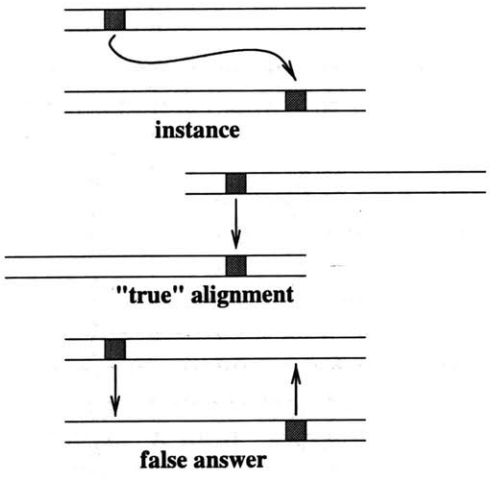


Figure 5-4: A difficult alignment problem.

is based on many good target sequences (and their fragments) simultaneously. Furthermore, in contrast to the INFO program, exons that have partial or no matches in a database can sometimes be accurately predicted by using the fact that exons are constrained independently in many ways: they require splice sites, must be frame consistent and cannot contain high complexity repeats [69]. Also in contrast to INFO, the method takes full advantage of the presence of long *and* short hits. The INFO program uses 26 tuples of amino acids, whereas the dictionary based approach can use all the tuple information starting with 4 tuples of amino acids. The AAT [38] tool is designed specifically to automate the process of finding a good target sequence. However, the reliance on one good target limits the ability of the program to predict exons from fragments.

The effectiveness of the dictionary method we propose is not at all obvious *apriori*. Indeed, the method can be inferior to alignment in the case where one wants to compare two similar DNA sequences, perhaps from different species. A few gaps and mismatches every 5–6 nucleotides will tend to exclude 11-tuple DNA hits. On the other hand, the method has proved to be very effective for exon prediction, especially when used in conjunction with a protein database (where mutations in the third position of codons do not necessarily alter the resultant protein), or an EST database (where exact matching fragments of a gene provide excellent candidates for the dictionary to find).

5.3.6 Running Times

The OWL dictionary was built on a Pentium 2 (400MhZ) in 10 minutes, and the dbEST dictionary, in 40 minutes. The total space occupied by the dictionaries was manageable on modern drives. The dbEST database occupied about 1.75GB. The analysis of a typical gene (computation of dictionary hits and solution of the best parse) was accomplished on the order of seconds.

The construction of the dictionaries, a routine exercise for small databases, was

complicated by the extremely large size of the databases. The solution described in the methods section solves the problem by finding a tradeoff between space and time that enables a realistic solution to the problem.

Chapter 6

Comparative Genomics

6.1 Introduction

In this chapter we present a new system for exon prediction based on comparative genomics. Our main result is the implementation and testing of a program which is based on the approaches outlined in the previous chapters, but which is greatly enhanced by the ability to compare information between genomes. We begin by introducing the new paradigm we have established, and then proceed to detail our algorithms. Finally, we discuss the results of tests performed with our program.

6.1.1 The Rosetta Stone

In September 1822, Jean-Francois Champollion uncovered the techniques which led to our current understanding of hieroglyphics. He managed to decipher the hieroglyphic language by using the Rosetta stone, which had been discovered in 1799.

The Rosetta stone is a record of the priestly honors and economic privileges given to Ptolemy V for the services he performed for Egypt. It contains text written in three languages: 14 lines of Egyptian hieroglyphics, 32 lines of Egyptian demotic and 54 lines of ancient Greek. Because scholars in the 19th century knew ancient Greek, Champollion was able to use the three languages simultaneously to reconstruct the meaning of the Egyptian hieroglyphics, and more importantly, to understand the structure of the language (for example, he deduced that there was a phonetic component to hieroglyphics).

In a sense, the problem of finding exons and introns in DNA, is a bit like deciphering a language, one that was composed by nature. But unlike the Rosetta stone, which was written in 196 BC, our own DNA is a record of events that have taken place during millions of years. Furthermore, when it comes to DNA, we have no ancient Greek that we understand and can use to begin the translation process. Nevertheless, evolution has been kind to us, and has acted differently on the different parts of the DNA code. In particular, while exons have remained relatively stable during evolution (possibly because mutations are often very damaging to the organism), introns have mutated wildly. Thus, the organisms of today are, in some sense, a Rosetta stone that we can use to learn about introns and exons. If our goal is to

Figure 6-1: The Rosetta stone.

distinguish introns and exons, we can use multiple organisms to see where they have been preserved (most likely exon) and where they have diverged (probably intron or intergenic material). Indeed, even though we cannot necessarily find the introns and exons in one organism alone, by using many organisms at once we can enhance our signals and hope to simultaneously identify the signals we are interested in. This idea is the backbone of what follows in this chapter [1].

6.1.2 A New Paradigm for Gene Annotation

The use of two or more organisms to distinguish introns and exons, or to annotate more complex features such as promoters seems to be a fundamentally new way of approaching the gene annotation problem. While the utility of two organism comparisons has been discussed [62, 33], little attention has been given explicitly to the problem of gene annotation, and to the algorithms necessary to accomplish this task.

In what follows we describe a detailed approach to exon prediction based on using two organisms (in our case, the human and mouse). Our algorithms are based on the following observations, which are almost always true (there are exceptions, and these are discussed at the end of this chapter):

- There is a bijective correspondence between coding exons in the human and the mouse.
- Coding exons in the human and the mouse are the same length modulo 3. Usually they are exactly the same length.

[1] Hieroglyphics are now extinct, and the Rosetta stone has been conveniently relocated to the British museum in London (where it has been since 1802). We hope that the DNA code, and the organisms that contains it (us), do not endure similar fates.

- The DNA identity between corresponding exons is about 75%. At the protein level, identity is around 85%.
- Introns are a lot less similar to each other than are coding exons, exhibiting large variations in length, and having DNA identity of less than 50%.

Based on these observations, we predict coding exons using the following general algorithm, which is described in the subsequent sections in detail:

1. Find the map between a human and its corresponding mouse gene. This entails globally aligning the two sequences so that the coding exons correspond.
2. Find good exon pairs in regions of good alignment by assuming an underlying Markov model and finding the best possible parse in the region.
3. Globally piece together good exon pairs to be frame consistent in both species.

We tried to keep the algorithm simple, yet as general as possible. For example, our method for predicting good exon pairs can be generalized to handle organism pairs other than human and mouse, simply by changing the PAM matrix used. We also tried to keep the algorithm symmetric with respect to the organism, so that the role of the human and mouse in the algorithm could be interchanged without effect. Finally, we avoided, as much as possible, techniques based on "learned" rules or probabilities, so that our algorithms are as robust as possible.

Each section below describes a main part of the overall algorithm and is self contained. We mention that the alignment method should be useful for purposes other than the exon prediction for which it is used in this thesis.

6.2 Alignments

6.2.1 Background

In order to use human/mouse comparisons for exon prediction, we need to align long, similar sequences of human and mouse DNA. This problem has been considered before [39, 31], but previous approaches suffer from a number of problems. Most methods do not compute an *entire map*, rather settling for finding regions of locally good alignment. Furthermore, the algorithms are slow; Hardison *et al.* [33] suggest that an alignment of the entire human and mouse genomes should take a month. Our approach is novel in that we produce a detailed alignment map for every position in the sequences being aligned. Our recursive approach means the algorithm is easily parallelized, and hence also fast. We define a **good alignment** to be one that satisfies the following two criteria:

- Corresponding exons between the human and mouse are mapped to each other.
- The map is well preserved even at the boundaries of coding exons.

Our alignment algorithm is based on an iterative mapping approach. The idea is to use large regions of homology to fix pieces of the alignment, and then to iteratively find the alignment for the regions in between the fixed ones. Thus we obtain a global alignment by successively fixing regions with good local alignment.

Formally, an alignment is a pair of maps $f : S_1 \to S_2 \cup \{-1\}$ and $g : S_2 \to S_1 \cup \{-1\}$ where S_1 and S_2 are the sets $\{1, \ldots, n\}$ and $\{1, \ldots, m\}$ respectively. The maps must be injective when restricted to elements that do not map to -1. In our case, we will think of S_1 and S_2 as representing positions in sequences of DNA (*i.e.*, there are additional maps N_1, N_2 from S_1, S_2 to $\{A, C, G, T\}$), and an element mapping to -1 will be a nucleotide that is "gapped".

Alignments have been well-studied in the context of biological sequences (for a comprehensive reference see Gusfield [31]). Our algorithm is a heuristic which is based on the two classic global alignment techniques. These will be referred to as **MMG** and **MMGG**:

ALGORITHM $MMG(m_a, m_s, g)$:
Input: Two sequences (S_1, S_2, N_1, N_2), real numbers m_a, m_s, g. Output: An "optimal" alignment of S_1 with S_2 where optimality is defined as the alignment maximizing the quantity

$$\sum_{i \in S_1, f(i) \neq -1, N(i) = N(f(i))} m_a + \sum_{i \in S_1, f(i) \neq -1, N(i) \neq N(f(i))} m_s + \sum_{i \in S_1, f(i) = -1} g + \sum_{i \in S_2, g(i) = -1} g. \tag{6.1}$$

This algorithm is known as the classical "Needleman-Wunsch" algorithm [66], and forms one of the original cornerstones of computational biology. The algorithm is based on dynamic programming. The simplest implementation runs in $O(nm)$ time and $O(nm)$ space where $|S_1| = n, |S_2| = m$. At the cost of doubling the running time, space requirements can be improved to $O(n), n < m$ [37, 65]. Reduction of space is especially important in our applications since the strings we are considering may be very long (hundreds of kilobases).

ALGORITHM $MMGG(m_a, m_s, g, g_o, g_e)$:
Exactly the same as algorithm **MMG** with two extra parameters g_o, g_e, where g_o is the penalty for opening a gap, and g_e is the penalty for a gap that is preceded by another one.

The **MMGG** algorithm is a simple extension of the **MMG** algorithm. Different gap opening and extension penalties are needed in our biological context because we would prefer to keep gaps clustered together.

6.2.2 Nested Alignments

Our main alignment algorithm, as described before, is based on an iterative, or nested, alignment procedure. The central ingredient in this algorithm is **ALGORITHM PARTIALALIGN** described next, which is the procedure that is called recursively. We begin by finding exact matches tuples of a fixed size between the two input sequences. The tuples are then aligned, with the match score for the alignment taken to be the alignment quality of the local regions around the matching tuples. That

is, two 1-tuple alignments are computed (one in each direction) and their scores are added. The resulting matching tuples are then "fixed", subject to there not being any inconsistencies. **ALGORITHM GLOBALALIGN** calls **ALGORITHM PARTIALALIGN** recursively, thereby fixing regions with increasingly higher resolution until the entire map is resolved.

ALGORITHM PARTIALALIGN:

Input: Two sequences (S_1, S_2, N_1, N_2), integers $k \geq 1, e_{Length} \geq 0$, and real numbers $e_C, e_S, e_{gap}, e_{mismatch}, e_{match}, e_{open}, e_{extend}, e'_{gap}, e'_{mismatch}, e'_{match}, e'_{open}, e'_{extend}$. Output: The algorithm outputs two partial maps f, g.

1. Find all the k-tuples in S_1 that match k-tuples in S_2. We now construct two sequences T_1, T_2 where T_1 consists of the tuples in S_1 that have a match in S_2 and T_2 consists of the (ordered) tuples in S_2 with a match in S_1. Notice that the number of distinct elements in T_1 is exactly the number of distinct tuples in S_1.

2. Align the sequences T_1 and T_2 using an **MMG** alignment where the parameters are: *Mismatch=0, Gap=0, Match=L+R.* L is defined to be the score of the alignment of a sequence of length e_{Length} to the left of a considered element of T_1, with a region of size e_{Length} to the left of the (corresponding) element in T_2. The alignment is an **MMGG** alignment with parameters $e_{gap}, e_{mismatch}, e_{match}, e_{open}, e_{extend}$. Similarly R is the score of an **MMGG** alignment of two sequences of length e_{Length} with parameters $e_{gap}, e_{mismatch}, e_{match}, e_{open}, e_{extend}$.

3. The alignment generated between T_1 and T_2 is really a partial map of k-tuples in S_1 to k-tuples in S_2. Inconsistencies in this map are removed in this step (*i.e.*, inconsistent tuples are unfixed). Such inconsistencies may arise from two consecutive overlapping tuples mapped to distant locations.

4. For each remaining mapped tuple, two **MMGG** alignments are performed on sequences on each side of the tuple (one on each side, with sequences are of length e_{Length}). These alignments are performed with the parameters $e'_{gap}, e'_{mismatch}, e'_{match}, e'_{open}, e'_{extend}$. Mapped tuples whose resulting alignments have a score $L + R < e_C$ are removed, and thus no longer mapped.

5. Of the remaining tuples, if the score of the alignment on *one* side is greater than e_S, the map is extended to cover the nucleotides on that side.

ALGORITHM GLOBALALIGN:

Input: Two sequences S_1, S_2, N_1, N_2, a collection of tuples sizes (positive integers) $p_1 > p_2 > p_3 > \cdots > p_n = 1$, as well as one set of parameters for each call to **PARTIALALIGN** (there will be n such calls). Output: Alignment between S_1 and S_2, (*i.e.*, the maps f, g).

1. Call **PARTIALALIGN** with tuple size $k = p_i$ and the corresponding set of parameters associated with p_i.

2. Repeat step 1 on each contiguous unmapped segment in S_1, and its corresponding segment in S_2 determined by the images of the endpoints of the unmapped segment in S_1.

6.3 Finding Coding Exons

Definition 6.1 *Given two sequences S_1 and S_2 and the alignment maps f, g, we define an exon pair to be a pair of subsequences $E_1 \in S_1$ and $E_2 \in S_2 = f(E_1)$. In addition, we impose the requirement that E_1 and E_2 can potentially be coding exons, i.e. they either have possible splice sites on the ends, or else initiation or stop codons.*

By considering all potential exon pairs with reasonable alignment in a dynamic programming context (Chapter 4), and by scoring the protein alignments between potential exons, we were able to reliably find coding exons in human (and mouse) genes. The dynamic programming was designed to operate under different assumptions. For example, we allowed for the different assumptions that the genomic region being analyzed contained only one gene, many genes, and even genes on either strand. In addition, we investigated the "local" performance of the program; that is, in the case where only information from neighboring exons was used. We found that the weak splice site consensus signals in the human were greatly enhanced by using an additional organism, as was information about codons *etc.*

6.3.1 Removing Regions with Bad Alignment

Given an alignment between two sequences S_1 and S_2, we began finding coding exons by removing regions with a *very* bad alignment. A badly aligning region is defined to be one with either too many gaps or too few matches:

Definition 6.2 *Consider a region P of length l. A position $i \in R$ is said to be in a* **bad** *alignment if* any *of the following conditions are true:*

1. *In a window of size w around i, there are fewer than m matches.*
2. *Position i is inside a gap of length at least g.*

If P contains no positions in a bad alignment, and l is greater than a cutoff parameter L, then P is said to be a **well-aligned** region. We chose the parameters $l = 25, w = 37, g = 30, m = 10$. Any positions not in a well-aligned region were forced to be non-coding. The choice of parameters was arbitrary; we chose to be lenient so as to avoid the possibility of removing any regions containing coding exons.

In addition to the local screening described above, we also removed any exon pairs that did not have a sufficient good overall alignment. Given an alignment map between two sequences, we define a score for the alignment as follows:

Definition 6.3 *$MMGGE(m_a, m_s, g, \alpha, \beta, cap)$: Given two strings S_1, S_2 together with the alignment maps f, g we define the* **MMGGE** *score as follows:*

$$MMGGE(m_a, m_s, g, \alpha, \beta.cap) = MAX_{j\in\{0,1,2,\}} \left(\sum_{i\in S_1, f(i)\neq -1, N(i)=N(f(i))} cm_j(m_a, i) \right)$$

$$+ \sum_{i\in S_1, f(i)\neq -1, N(i)\neq N(f(i))} m_s + \sum_{i\in S_1, f(i)=-1} cg(g,i) + \sum_{i\in S_2, g(i)=-1} cg(g,i).$$

The function $cm_j(m_a, i)$ is defined as follows: Let $th(i)$ be the total number of gaps until position i in S_1. Similarly, define $tm(i)$ to be the total number of gaps until position i in S_2. If $th(i) - tm(i) \equiv j \bmod 3$ then

$$cm_j(m_a, i) = m_a \beta^{MIN(l_m, cap)} \tag{6.2}$$

where l_m is the length of the longest consecutive string of matches ending at i. Otherwise,

$$cm_j(m_a, i) = -\frac{m_a}{2} \tag{6.3}$$

The function $cg(g, i)$ is defined in a similar fashion; it is given by

$$cg(g, i) = g\alpha^{MIN(l_g, cap)} \tag{6.4}$$

where l_g is the length of the longest consecutive string of gaps ending at i.

In principle, the alignment maps could be constructed by optimizing the function **MMGGE** described above. We found that in practice, the simpler alignment scoring functions sufficed to find the map, and that the more complicated scoring scheme was useful for post-processing.

We removed any exon pairs where the **MMGGE(2,-1,-3,0.84,1.2,7)** score was less than half the length of the human exon.

6.3.2 Scoring a Pair of Exons

The score of an exon pair $E_h \in$ human and $E_m \in$ mouse is given by

$$s(E_h, E_m) = Sb_h + Sb_m + Se_h + Se_m + EP(E_h, E_m) + EC(E_h, E_m) \tag{6.5}$$

where Sb_h, Sb_m, Se_h and Se_m (e for end, b for begin) are scores assigned to the endpoints of the exons. The scores depend on whether the endpoints are splice sites, initiation, or termination codons. We set the score of termination and initiation codons to 0. The score of a donor or acceptor splice site (either Sb_h or Sb_m for acceptor, Se_h or Se_m for donor) was given by the left rule modified `GENSCAN` splice site detector (Chapter 3). Donor splice site scores were multiplied by a factor of $\frac{1}{2}$, and acceptor splice site scores were multiplied by 3.5. This was done to balance the scores; the factors were designed to equate the mean true splice site scores.

$EP(E_h, E_m)$ is defined to be the protein alignment score of the two exons. Formally,

$$EP(E_h, E_m) = \sum_{i \in E_h} PAM(ch_i, f(ch_i)), \tag{6.6}$$

where i ranges over the codons in E_h, ch_i is the ith codon in the human sequence, and $f(ch_i)$ is the image of ch_i under the alignment map f (thus, $f(ch_i)$ is a codon, or a gap, in the mouse). The function $PAM(a, b)$ is the value of a PAM matrix for two codons a, b (see Appendix A). Note that as i ranges through E_h, the codon ch_i might be a gap. We used the PAM20 matrix (see Appendix A), normalized so that the average diagonal entry was 2. Future improvements will include constructing a specialized "PAM" matrix for the exact problem of determining homologies between human and mouse sequences. We chose to use a precomputed matrix so as to avoid the problem of training on the test set. PAM20 was selected (as opposed to a different PAMn matrix) because 20 point mutations per one hundred nucleotides seemed to agree with our results about human/mouse alignments (see discussion).

$EC(E_h, E_m)$ is defined to be the codon usage score of the two exons. Similarly to $EP(E_h, E_m)$, it is given by

$$EC(E_h, E_m) = \sum_{i \in E_h} CODON_h(ch_i) + \sum_{i \in E_m} CODON_m(cm_i). \tag{6.7}$$

As before, ch_i, cm_i represent the ith codons. $CODON_h$ and $CODON_m$ are log odds ratios that for a codon c, return the value

$$CODON_{h,m}(c) = \frac{\log f_c}{\log e_c}. \tag{6.8}$$

That is, depending on whether one is examining human or mouse, the function returns the logarithm of the frequency of occurrence of the codon in an exon database, divided by the logarithm of the expected frequency of occurrence. Tables for human and mouse codon usage are listed in Appendix A.

Cutoffs

As mentioned previously, cutoffs were employed to remove regions of bad alignment. Similarly, cutoffs were used to remove very bad splice sites. Coding exons with a combined splice site score less than -10 were not allowed. These cutoffs are heuristic, and can be tuned to remove false positives at the expense of false negatives. Similar cutoffs were experimented with for protein alignments and codon usage. We decided not to implement such cutoffs at this time.

Penalties

Penalties were implemented to account for the different length distributions of introns and exons. These penalties will be improved with time, both to be more probabilistic

and to take into account various parameters such as GC richness, G triplets near splice sites *etc.* (see Chapter 3).

A penalty was also imposed for exons with different lengths, with the penalty dependent on the difference of the lengths modulo 3. For internal exons, the penalty for being of different length modulo 3 was -27, and a penalty of -9 was imposed for length which were different, but the same modulo 3. Initial and terminal exons received similar penalties of -9 and -3 respectively.

6.3.3 Piecing together Exons

The dynamic programming algorithm described in Chapter 4 was used to find the optimal parse maximizing the exon pair scores from equation (6.5). A minor modification to the basic dynamic programming routine was added to deal with the special case of human/mouse based parsing:

Since the exon pair score is based on an exon pair E_h, E_m, and the dynamic programming was based on the human alone, it was necessary to find a matching exon pair in the mouse for a potential exon E_h in the human. This was accomplished by defining E_m to be $f(E_h)$ if the endpoints of E_m matched equivalent endpoints in the mouse. That is, if splice sites in the human mapped directly to splice sites in the mouse, they were used. In the case of no map, splice sites (or initiation/stop codons) within windows of size 15 on either side of the image were considered (if the case of no images inside these windows the potential exon in the human was rejected). Thus, a potential exon E_h in the human was paired with either $1, 2$ or 4 exons in the mouse. The best pairing was considered to be the true pair.

Frame consistency between predicted exons was required both in the human and the mouse. This was particularly important in the context of the different assumptions (Chapter 2) underlying the dynamic programming. We implemented the ability to operate under all possible assumptions, ranging from the single gene assumption, to multiple genes.

6.4 Results

We tested our methods on the HUMCOMP/MUSCOMP comparison test set (see Appendix B for details on how the test set was constructed). The program described in this chapter, which we call **Rosetta**, was tested with a variety of underlying assumptions:

1. The genomic regions we were analyzing consisted of only one gene (we call this the *single gene assumption*). The dynamic programming algorithm was set to find the optimal *valid parse* (see Chapter 4).

2. We allowed for multiple genes in the genomic regions analyzed (we call this the *multiple gene assumption*). The dynamic programming algorithm was set to find the optimal collection of *partial valid parses*, with the added restriction that a

partial valid parse could begin with an initiation codon only if the previous partial valid parse ended with a stop codon.

3. We also considered the problem of finding exons *locally.* We defined this to be the problem of using dynamic programming to find optimal partial valid parses for each *well-aligned* region. Information about the parse obtained in a well-aligned region was not used when finding parses for its neighbors. We call this the *parsed in pieces assumption.*

4. We also considered the *double strand assumption,* where the optimal parse was found on both strands.

The third assumption was used primarily as an intermediate step towards the goal of parsing with the multiple gene, double strand assumption. We state the results for this assumption in Table 6.5 because of its applicability for finding exons in short genomic fragments.

Results with the single gene assumption appear in Table 6.8. We include these for comparison with most other gene recognition programs which operate with this assumption. It is also interesting to note the large gain in accuracy obtained by adding this assumption.

Table 6.6 contains the results with the multiple gene assumption.

In the double strand assumption we did not allow for coding exons simultaneously appearing on both strands in the same position. If this happened, we picked a strand by examining a region of size 2000 on each size of both exons (on both strands) and picking for our prediction the strand in which we predicted the most coding exon in this region. Results with this assumption, as well as the multiple gene assumption (our most general result) appear in Table 6.7. The results of running `GENSCAN` operating under the same assumptions on the same test set appear in Table 6.9. Out of the 117 genes we tested on, 40 appear in `GENSCAN`'s training set. Surprisingly, we noticed that `GENSCAN`'s performance did not deteriorate when removing these genes.

The main results of our runs are summarized in the tables below. Table 6.1 shows a comparison between the `Rosetta` program and `GENSCAN`, with the most general assumptions in place (multiple genes, double strand).

	`Rosetta`	`GENSCAN`
Nucleotide sensitivity	95%	98%
Nucleotide specificity	97%	89%
Exon sensitivity	84%	83%
Exon specificity	83%	76%
Predicted exons not overlapping coding exon	26	**68**

Table 6.1: Summary of results for all coding exons.

Tables 6.2 and 6.3 show the statistics for internal and external coding exons.

	Rosetta	GENSCAN
% exons correct on both ends	93%	91%
% exons correct on one end only	4%	6%
% exons completely missed	3%	3%

Table 6.2: Summary of results for interior coding exons.

	Rosetta	GENSCAN
% exons correct on both ends	71%	73%
% exons correct on one end only	19%	15%
% exons completely missed	8%	10%

Table 6.3: Summary of results for exterior coding exons.

In Table 6.4 we show all of the genes in the HUMCOMP/MUSCOMP test set, together with the DNA and protein alignments for each exon as well as the results of our predictions. The first two rows contain the loci of the genes, the number of coding exons in each, the total length of the coding regions of the genes, and then the length of the individual exons. The alignment percentages correspond to the percentage of positions in the exons that match. The DNA alignment was done as follows: we used the exon annotations in the human and mouse to find the corresponding exons. If our map did not correspond to the annotations, we returned an alignment of 0. Otherwise we return the alignment given by our map (percentage of matches). For the protein we recompute an alignment based solely on the annotations. Thus, even if our map did not correspond to the annotations, we could compute an alignment. This was done so as to be able to see when our map did not correspond to the annotations. Unfortunately, not all the genes in MUSCOMP contained complete annotations, for these the alignment results are invalid. Furthermore, in genes with different numbers of exons in the human and mouse it was sometimes difficult to establish the exact correspondence. Finally, we show for each gene the results of our predictions for each exon. A $\surd$ indicates an exon correctly identified on both ends. A $\times$ indicates the exon was completely missed. $\times - \surd$, $\surd - \times$ and $\times - \times$ indicate that the exon was covered but at least one of the ends was not predicted correctly. In this latter case, the $\surd$ indicates which end was correct, if any at all.

6.5 Discussion

A number of observations are immediate from the results, and are interesting:

- Our results are robust in that they did not seem to differ much from gene to gene.

- We find external exons in genes about as accurately as **GENSCAN**, despite not incorporating promoter detection or other signals that help fix gene boundaries.
- Our main advantage at this stage is our superior specificity.
- Our errors are mostly restricted to external exons. Upon analysis, we see that our mistakes on internal exons are the result of a few, unrelated pathological examples. There does not seem to be a clear trend.

Our results also clearly demonstrate the advantages of using multiple organisms to enhance the signal to noise ratio in coding exon detection. Our technique clearly relies on the empirical observation that coding exon regions have good alignments, as opposed to introns. In Figures 6-3 and 6-4 we show the distributions of matches and gaps in windows of size 39 in exons and introns (taken from the HUMCOMP/MUSCOMP datasets). The figures clearly show excellent separation; the exact figures for overall alignment agree with those in the literature [52, 51, 67, 33, 62].

Unfortunately, it is not the case that the alignments alone allow us to distinguish exons from introns. In Figure 6-2 we show an example of a typical gene, and how there can be a lot more well-aligned material, than there is coding exon. The top bar represents the human gadd45 gene, and the bottom bar the orthologous mouse gene. The light regions within the bars are the coding exons. Arrows originate on splice sites and show where they map to. The painted region between the bars indicates regions of good alignment.

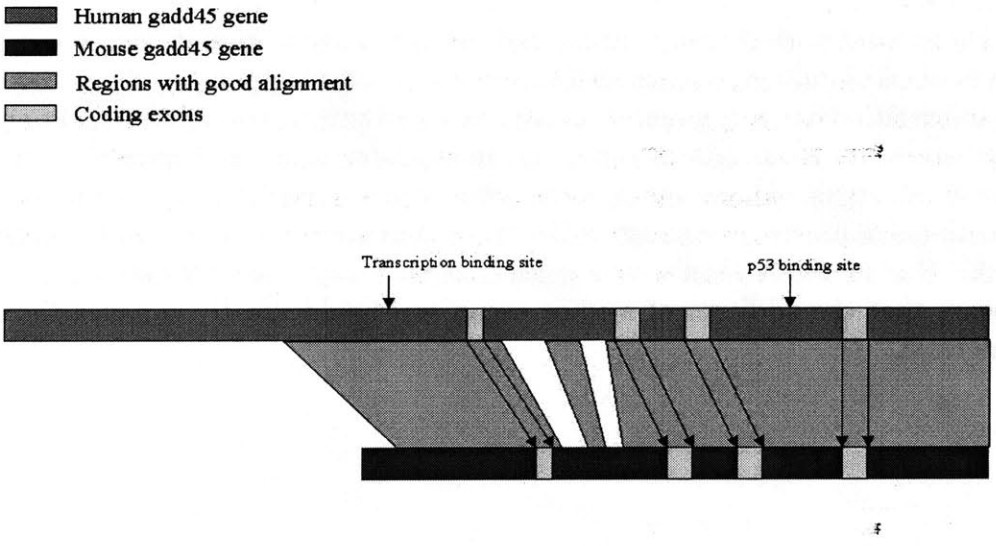


Figure 6-2: The gadd45 gene

Much of the power of our algorithm comes from our EP function, which is measuring how the amino acids mutated (using the PAM matrix). Indeed, it seems that what is helping us more is the *type* of mutation, rather than the *place* of mutation. In this work, we have strived to keep our tests fair and unbiased, and have therefore **not** trained on the HUMCOMP and MUSCOMP datasets. However, after testing, we are now in a position to use this data to measure exactly the types of mutations that occur, and considerably refine our PAM matrix.

The nature of the Rosetta technique also enables analysis other than gene recognition (and exon detection) to be performed on pairs of orthologous genes from two organisms. The fact that a global alignment is obtained, in conjunction with the exon prediction, allows for an accurate measurement of DNA and protein similarity between corresponding genes. This should prove useful in making statements about the functions of the genes being analyzed. As an example, consider the pair of genes HSU12202 and MMMRPS24 (see Table 6.4). These genes encode for the ribosomal protein S24. There are three interior coding exons in these genes (of lengths 66, 210 and 111 nucleotides) which have the property that each is preserved 100% at the protein level! In contrast, the DNA similarity is only about 90% for each exon. This would strongly suggest that this particular protein is structurally *very* constrained. Indeed, it seems reasonable that further analysis will show that various functional conclusions about genes can be drawn from examining extremely well preserved exons (and also perhaps exons which have diverged).

In addition to the applications outlined above, a detailed analysis of the types of mutations in promoters, enhancers and other signals in genes should yield numerous insights into the biological processes associated with the signals (a la the techniques of Chapter 3).

There is no apriori reason to restrict our methods to human-mouse comparisons. The method should generalize well to the use of other organisms simply by changing the splice site detection method, as well as recalibrating the PAM matrix and length parameters. It is an interesting question to determine the "optimal" evolutionary distance for signal enhancement. Organisms that are too far apart will be too different to distinguish preserved regions, while those that are too similar will be preserved too much. Our methods should also generalize to n organism comparisons. Indeed, it appears that the addition of a third organism should greatly enhance the signal to noise ratio.

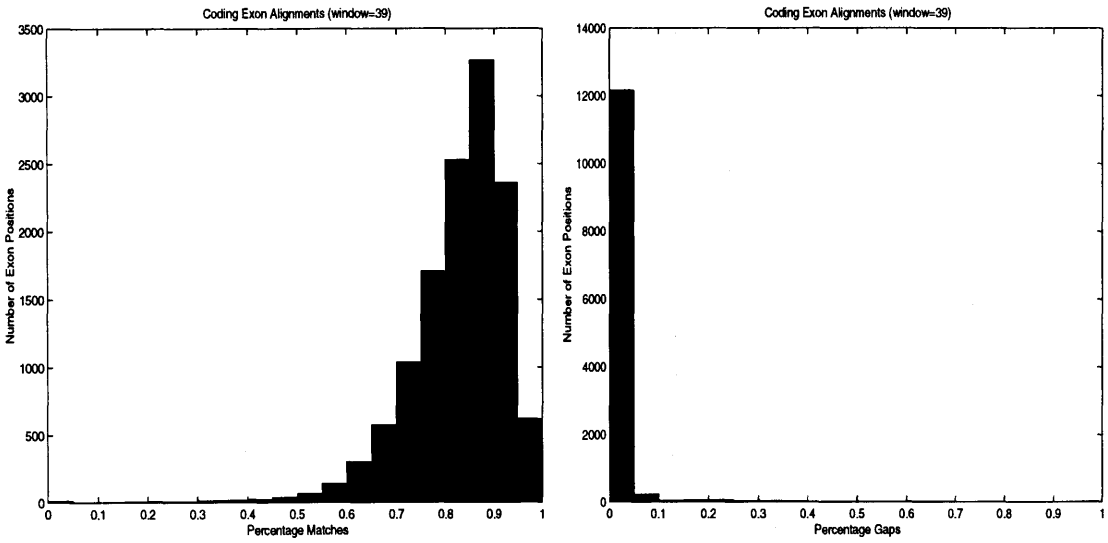


Figure 6-3: Alignment statistics in coding exons

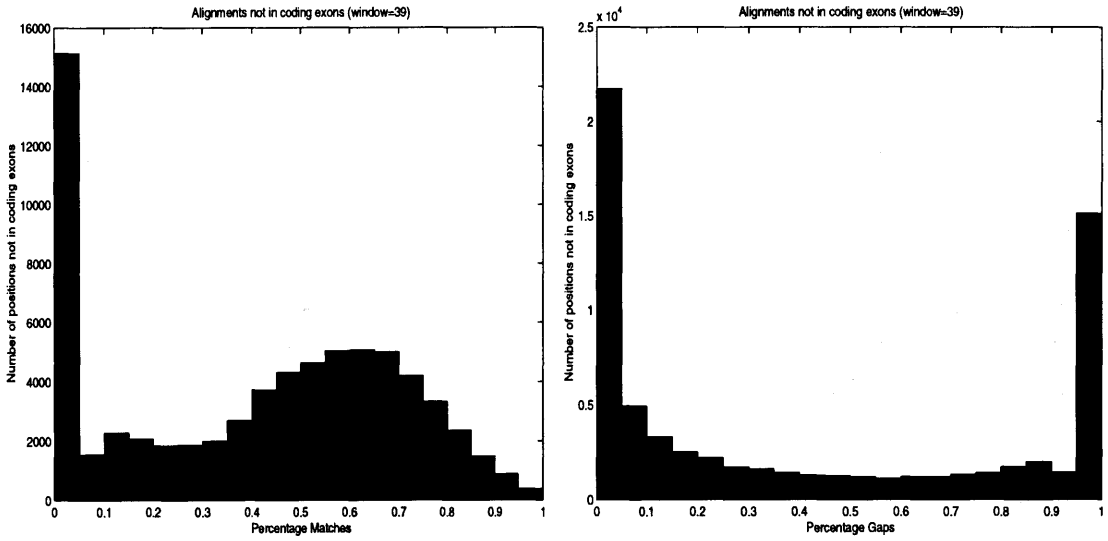


Figure 6-4: Alignment statistics outside of coding exons

HSCKIIBE	6	648	72	103	116	76	190	91		
MMGMCK2B	6	648	72	103	116	76	190	91		
DNA %		93.83	94.44	96.12	90.52	93.42	93.68	95.6		
Protein %		98.61	100	99.03	98.28	98.68	97.89	98.9		
Predictions			× − √	√	√	√	√	√		
HUMSAACT	6	1134	129	325	162	192	182	144		
MUSACASA	6	1134	129	325	162	192	182	144		
DNA %		90.83	89.92	89.23	90.12	90.62	92.86	93.75		
Protein %		99.21	100	99.69	98.15	98.44	98.9	100		
Predictions			× − √	√	√	√	√	√		
HSH4EHIS	1	312	312							
MMHIS412	1	312	312							
DNA %		89.74	89.74							
Protein %		100	100							
Predictions			√							
HSU12202	5	393	3	66	210	111	3			
MMMRPS24	5	396	3	66	210	111	6			
DNA %		89.06	100	86.36	89.52	91.89	0			
Protein %		100	100	100	100	100	100			
Predictions			×	√	√	√	×			
HUMHIS4	1	312	312							
MUSHIST4	1	312	312							
DNA %		87.5	87.5							
Protein %		100	100							
Predictions			× − √							
HSHISH3	1	411	411							
MMHIST31	1	411	411							
DNA %		86.13	86.13							
Protein %		100	100							
Predictions			√							
HSHSC70	8	1941	205	206	153	556	203	199	233	186
MMU73744	8	1941	205	206	153	556	203	199	233	186
DNA %		89.39	86.83	89.81	90.85	89.75	92.61	85.43	90.13	89.25
Protein %		99.23	99.51	99.03	100	99.82	97.54	99.5	97.85	100
Predictions			× − √	√	√	√	√	√	√	√
HUMNOCT	1	1332	1332							
MUSPOUDOMB	1	1338	1338							
DNA %		93.62	93.62							
Protein %		99.1	99.1							
Predictions			√							
HUMTROC	6	486	24	31	147	115	137	32		
MUSCTNC	6	486	24	31	147	115	137	32		
DNA %		91.15	95.83	87.1	91.84	93.04	87.59	96.88		
Protein %		96.91	100	96.77	97.96	96.52	96.35	93.75		
Predictions			√	√	√	√	√	√		
HSINT1G	4	1113	104	254	266	489				
MUSINT1A	4	1113	104	254	266	489				
DNA %		91.46	91.35	89.37	91.73	92.43				
Protein %		98.38	89.42	98.03	99.25	100				
Predictions			× − √	√	√	√				

HUMSRI1A	1	1176	1176									
MUSSRI1A	1	1176	1176									
DNA %		89.12	89.12									
Protein %		98.72	98.72									
Predictions			√									
HSMIMAR	1	1383	1383									
MUSACHRM1	1	1383	1383									
DNA %		90.31	90.31									
Protein %		98.05	98.05									
Predictions			× − √									
HSFAU1	4	402	75	145	56	126						
MUSFAUA	4	402	75	145	56	126						
DNA %		91.54	94.67	88.97	91.07	92.86						
Protein %		97.01	96	95.17	96.43	100						
Predictions			× − √	√	√	√						
HUMCRYABA	3	528	201	123	204							
MUSALPBCRY	3	528	201	123	204							
DNA %		91.29	92.54	84.55	94.12							
Protein %		97.73	95.52	100	98.53							
Predictions			× − √	√	√							
HSENO3	11	1305	85	96	59	70	134	223	198	202	109	59	70
MMENO3G	11	1305	85	96	59	70	134	223	198	202	109	59	70
DNA %		89.81	95.29	89.58	88.14	92.86	85.82	94.62	85.86	89.6	89.91	88.14	87.14
Protein %		96.09	98.82	93.75	81.36	98.57	96.27	99.55	93.94	96.53	99.08	96.61	94.29
Predictions			× − √	√	√	√	√	√	√	√	√	√	√
HUMPPIB	1	558	558									
MUSPPIA	1	558	558									
DNA %		93.01	93.01									
Protein %		97.31	97.31									
Predictions			√									
HUMKCHN	1	1572	1572									
MUSMK3A	1	1587	1587									
DNA %		90.71	90.71									
Protein %		97.14	97.14									
Predictions			√									
HUMPCNA	6	786	221	98	68	195	124	80				
MMPCNAG	6	786	221	98	68	195	124	80				
DNA %		88.42	88.69	85.71	89.71	88.72	88.71	88.75				
Protein %		95.8	95.02	97.96	97.06	96.92	94.35	93.75				
Predictions			√	√	√	√	√	√				
HSU73304	1	1419	1419									
MMU22948	1	1422	1422									
DNA %		90.49	90.49									
Protein %		97.25	97.25									
Predictions			√									
AF007876	7	873	112	129	105	206	57	99	165			
MMATPB2	7	873	112	129	105	206	57	99	165			
DNA %		90.95	97.32	91.47	93.33	88.35	91.23	85.86	90.91			
Protein %		96.22	99.11	95.35	91.43	96.12	94.74	93.94	100			
Predictions			√	√	√	√	√	√	√			

HUMNT3A	1	774	774							
MMNT3	1	777	777							
DNA %		89.79	89.79							
Protein %		96.12	96.12							
Predictions			× − √							
HSCKBG	7	1146	193	155	133	172	124	190	179	
MUSCRKNB	7	1146	193	155	133	172	124	190	179	
DNA %		90.23	87.56	89.03	92.48	88.95	94.35	93.16	87.71	
Protein %		95.55	91.71	90.97	99.25	95.93	96.77	96.32	98.88	
Predictions			× − √	√	√	√	√	√	√	
HUMACHRM4	1	1437	1437							
MMM4ACHR	1	1440	1440							
DNA %		89.42	89.42							
Protein %		95.2	95.2							
Predictions			√							
HUMMHHSP2	1	1926	1926							
MUSHSP7A2	1	1929	1929							
DNA %		91.38	91.38							
Protein %		95.33	95.33							
Predictions			√							
HUMAPEXN	4	957	58	188	193	518				
MUSAPEX	4	954	55	188	193	518				
DNA %		86.73	74.14	89.36	87.05	87.07				
Protein %		94.04	62.07	89.36	97.93	97.88				
Predictions			√	√	√	√				
HUMGAD45A	4	498	44	102	238	114				
MUSGAD45	4	498	44	102	238	114				
DNA %		90.56	97.73	88.24	88.24	94.74				
Protein %		93.37	88.64	91.18	92.02	100				
Predictions			√	√	√	√				
HUMMHHSPHO	1	1926	1926							
MUSHSC70T	1	1926	1926							
DNA %		84.94	84.94							
Protein %		94.39	94.39							
Predictions			√							
HSHOX51	2	768	433	335						
MMU77364	2	753	427	326						
DNA %		88.54	87.99	89.25						
Protein %		92.19	90.76	94.03						
Predictions			× − √	√						
HUMHISAC	1	660	660							
MUSH1EH2B	2	987	660	327						
DNA %		85.76	85.76							
Protein %		94.09	94.09							
Predictions			√							
HUMSPERSYN	8	909	167	121	93	154	84	146	123	21
MMSPERSYN	7	909	167	121	93	154	230	123	21	
DNA %		88.67	93.41	90.08	89.25	87.66	89.29	88.36	83.74	76.19
Protein %		93.73	97.01	99.17	100	93.51	92.86	90.41	85.37	85.71
Predictions			√	√	√	√	√	× − √	× − √	√

HSHIS10G	1	585	585										
MMU18295	1	585	585										
DNA %		91.11	91.11										
Protein %		94.87	94.87										
Predictions			√										
HSCFOS	4	1143	141	252	108	642							
MMCFOS	4	1143	141	252	108	642							
DNA %		88.28	90.07	84.13	91.67	88.94							
Protein %		93.7	97.87	86.9	100	94.39							
Predictions			√	√	√	√ − ×							
HUMHGCR	1	1173	1173										
MUS5HT1B	1	1161	1161										
DNA %		88.83	88.83										
Protein %		92.07	92.07										
Predictions			√										
HUMUDPCNA	1	1338	1338										
MUSGLCNACT	1	1344	1344										
DNA %		84.75	84.75										
Protein %		90.36	90.36										
Predictions			√										
HSODCG	10	1386	102	174	173	135	82	84	163	113	215	145	
MUSODCC	10	1386	102	174	173	135	82	84	163	113	215	145	
DNA %		86.36	86.27	86.78	90.17	87.41	84.15	88.1	87.12	84.07	88.37	78.62	
Protein %		89.61	82.35	94.83	90.17	93.33	87.8	89.29	92.02	92.92	93.49	74.48	
Predictions			√	√	√	√	√	√	√	√	√	√	
HUMGALTB	11	1140	82	170	76	49	130	57	123	133	84	155	81
MMU41282	11	1083	25	170	76	49	130	57	123	133	84	155	81
DNA %		77.02	23.17	88.82	84.21	77.55	95.38	94.74	81.3	90.98	94.05	82.58	0
Protein %		85.79	25.61	97.06	86.84	73.47	96.92	100	85.37	90.23	96.43	87.1	81.48
Predictions			× − √	√	√	√	√ − ×	×	×	×	× − √	√	×
HSU29185	1	738	738										
MUSPRNPA	1	765	765										
DNA %		81.3	81.3										
Protein %		83.74	83.74										
Predictions			× − √										
HSU01212	1	492	492										
MMU01213	1	492	492										
DNA %		88.62	88.62										
Protein %		89.63	89.63										
Predictions			√										
HUMMIF	3	348	108	173	67								
MMU20156	3	348	108	173	67								
DNA %		88.51	89.81	87.28	89.55								
Protein %		88.79	94.44	83.24	94.03								
Predictions			√	√	√								
AF049259	7	1263	495	83	157	162	126	221	19				
MMU13921	8	1314	471	83	157	162	126	221	23	71			
DNA %		84.8	80.61	86.75	91.08	88.89	90.48	90.05	0				
Protein %		86.46	80.61	79.52	93.63	92.59	92.86	96.38	0				
Predictions			√	√	√	√	√	√	×				

HSH12	1	642	642						
MUSHIS1A	1	639	639						
DNA %		76.95	76.95						
Protein %		84.11	84.11						
Predictions			× − √						
HSACTHR	1	894	894						
MUSACTHR	1	891	891						
DNA %		84.12	84.12						
Protein %		88.26	88.26						
Predictions			√ − ×						
AF027148	3	963	630	79	254				
MMMYOD1	3	957	627	79	251				
DNA %		85.25	89.05	81.01	77.17				
Protein %		87.85	94.76	68.35	76.77				
Predictions			√	√	√				
HUMKER18	7	1293	417	83	157	165	126	224	121
MUSENDOBA	7	1272	396	83	157	165	126	224	121
DNA %		83.45	79.86	89.16	88.54	87.88	80.95	83.93	80.99
Protein %		86.54	81.29	93.98	95.54	89.09	80.95	88.39	86.78
Predictions			√	√	√ − ×	√	√	√	√
HUMADRA	1	1353	1353						
MUSALP2ADB	1	1353	1353						
DNA %		87.29	87.29						
Protein %		88.25	88.25						
Predictions			√						
HSMHCPU15	6	660	60	68	132	130	142	128	
MUSLMP2A	4	642	281		132	130	99		
DNA %		54.24	0	89.71	85.61	80.77	55.63	0	
Protein %		58.18	65	92.65	88.64	90	19.01	16.41	
Predictions			×	√	√	√	√	√	
HSU72648	1	1386	1386						
MUSADRA	1	1377	1377						
DNA %		86.65	86.65						
Protein %		88.1	88.1						
Predictions			√						
HUMMK	4	432	76	168	162	26			
MUSMKPG	4	423	76	159	162	26			
DNA %		86.57	78.95	85.71	90.74	88.46			
Protein %		84.72	82.89	82.14	87.04	92.31			
Predictions			× − √	√	√	√			
HSMYF4G	3	675	471	81	123				
MUSMYOGEN	3	675	471	82	122				
DNA %		90.67	90.87	90.12	90.24				
Protein %		94.67	96.82	85.19	92.68				
Predictions			× − √	√	√				
HUMHISAB	1	666	666						
MMHISTH1	1	666	666						
DNA %		79.43	79.43						
Protein %		87.84	87.84						
Predictions			√						

HUMOTNPI	3	375	120	199	56														
MUSOXYNEUI	3	378	120	202	56														
DNA %		87.47	87.5	90.45	76.79														
Protein %		87.2	90	91.96	64.29														
Predictions			✓	✓	✓														
HUMTKRA	7	705	66	32	111	94	90	120	192										
MUSTKM	7	702	66	32	111	94	90	120	189										
DNA %		84.11	89.39	93.75	89.19	86.17	83.33	81.67	78.65										
Protein %		86.38	90.91	93.75	91.89	89.36	86.67	92.5	75										
Predictions			✓	✓	✓	✓	✓	✓	✓										
HUMADRBRA	1	1242	1242																
MMB2ARG	1	1257	1257																
DNA %		85.43	85.43																
Protein %		87.2	87.2																
Predictions			✓																
HUMMETIII	3	207	31	66	110														
MUSMETIII	3	207	31	66	110														
DNA %		88.41	93.55	84.85	89.09														
Protein %		84.06	87.1	77.27	87.27														
Predictions			✓	✓	✓														
HUMXRCC1G	17	1902	51	93	111	159	75	112	110	112	259	117	94	133	55	140	91	76	114
MUSXRCC1G	17	1890	51	93	117	159	75	112	110	100	259	117	94	130	55	137	91	76	114
DNA %		84.23	96.08	92.47	89.19	89.94	80	88.39	80.91	68.75	79.54	91.45	80.85	80.45	90.91	80	82.42	85.53	89.47
Protein %		84.7	94.12	100	100	98.11	88	91.07	79.09	61.61	81.08	94.87	70.21	63.16	87.27	72.86	89.01	86.84	97.37
Predictions			× − ✓	✓	✓	✓	✓	× − ✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
HUMG0S24B	2	981	24	957															
MUSZPF36G	2	960	24	936															
DNA %		83.08	91.67	82.86															
Protein %		84.4	87.5	84.33															
Predictions			✓	✓															
HSAGL1	3	429	95	205	129														
MUSHBA	3	429	95	205	129														
DNA %		80.42	73.68	79.51	86.82														
Protein %		85.31	72.63	86.34	93.02														
Predictions			✓	✓	✓														
HSPGK2G	1	1254	1254																
MUSPGK2	1	1254	1254																
DNA %		84.93	84.93																
Protein %		86.12	86.12																
Predictions			✓																
HSU57623	4	402	73	173	102	54													
MMU02884	4	402	73	173	102	54													
DNA %		83.33	90.41	85.55	77.45	77.78													
Protein %		85.07	90.41	84.97	82.35	83.33													
Predictions			✓	✓	✓	✓													
HSARYLA	8	1524	218	241	219	170	125	128	103	320									
MMDNAASFA	8	1521	215	241	219	170	125	128	103	320									
DNA %		82.61	79.36	86.31	82.19	85.29	79.2	88.28	88.35	78.12									
Protein %		84.65	83.94	92.12	87.67	88.24	84	91.41	81.55	74.06									
Predictions			✓	✓	✓	✓	✓	✓	✓	✓									

S63168	1	810	810												
MUSCRP3A	1	804	804												
DNA %		85.31	85.31												
Protein %		85.19	85.19												
Predictions			√ – ×												
HSHSP27	3	600	364	64	172										
MUSHSP25A	3	630	375	66	189										
DNA %		84	84.89	90.62	79.65										
Protein %		83.5	84.89	93.75	76.74										
Predictions			√	√	√										
HUMROD1X	3	1056	590	247	219										
MUSROM1X	3	1056	590	247	219										
DNA %		85.04	84.24	85.83	86.3										
Protein %		84.38	83.39	88.66	82.19										
Predictions			× – ×	√	√										
HUMSSTR3X	1	1257	1257												
MUSSSTR3A	1	1287	1287												
DNA %		82.74	82.74												
Protein %		85.2	85.2												
Predictions															
HSPNMTB	3	849	202	208	439										
MUSPNMT	3	888	235	208	445										
DNA %		82.92	77.72	83.65	84.97										
Protein %		84.45	75.74	86.54	87.47										
Predictions			× – √	√	√										
HSIGF2G	2	394	157		237										
MMU71085	3	543	157	149	237										
DNA %		85.53	87.26		84.39										
Protein %		45.69	85.99		18.99										
Predictions			× – √	√											
HUMADAG	12	1092	33	62	123	144	116	128	72	102	65	130	103	14	
MMU73107	11	1059	33	62	123	144	116	128	72	102	65	130	84		
DNA %		78.85	87.88	83.87	82.11	86.81	83.62	79.69	72.22	79.41	80	83.85	59.22	0	
Protein %		80.22	90.91	87.1	75.61	91.67	77.59	75	83.33	85.29	73.85	92.31	58.25	42.86	
Predictions			√	√	√	√	√	√	√	√	√	√	×	×	
HUMSMPD1G	6	1890	312	773	172	77	146	410							
MMASM1G	6	1884	306	773	172	77	146	410							
DNA %		82.49	70.51	83.31	91.28	88.31	86.3	83.9							
Protein %		82.06	54.81	87.32	92.44	89.61	90.41	84.15							
Predictions			× – √	√	√	√	√	√							
HUMCOX5B	4	390	103	74	100	113									
MUSCYTCOVB	4	387	100	74	100	113									
DNA %		85.9	84.47	89.19	88	83.19									
Protein %		80.77	75.73	89.19	87	74.34									
Predictions			√	√	√	√									
HUMNUCLEO	14	2124	18	117	478	198	87	142	125	124	158	124	134	127	215	77
MMNUCLEO	14	2124	18	117	496	186	87	142	125	124	155	124	122	127	224	77
DNA %		84.32	94.44	88.03	79.92	72.22	88.51	80.99	90.4	83.87	85.44	91.94	84.33	89.76	88.84	90.91
Protein %		82.34	100	87.18	75.31	66.67	86.21	69.72	96	77.42	79.75	84.68	82.84	94.49	99.07	93.51
Predictions			√	√	√	√	√	√	√	√	√	√	√	√	√	√

S45332	8	1527	115	136	176	158	154	88	88	612										
MMERYPR	8	1524	115	133	176	158	154	88	88	612										
DNA %		82.19	82.61	85.29	77.84	77.85	87.01	80.68	94.32	81.05										
Protein %		80.75	78.26	79.41	80.11	75.95	89.61	85.23	92.05	78.43										
Predictions			√	√	√	√	√	√	√	√										
HUMINSPR	2	333	187	146																
MMINSIIG	2	333	186	147																
DNA %		82.28	79.68	85.62																
Protein %		81.08	80.21	82.19																
Predictions			√	√																
HUMHST	3	621	340	104	177															
MMKFGF	3	609	328	104	177															
DNA %		81.16	77.65	86.54	84.75															
Protein %		79.23	70.59	86.54	91.53															
Predictions			√	√	√ − ×															
HUMPLPSPC	5	594	42	159	123	111	159													
MUSPSPC	5	582	42	159	123	111	147													
DNA %		79.63	85.71	84.28	80.49	86.49	67.92													
Protein %		78.79	85.71	90.57	78.05	78.38	66.04													
Predictions			√	√	√	√ − ×	√													
HSCOSEG	3	495	120	202	173															
MUSVASNEU	3	507	132	202	173															
DNA %		85.05	85	92.08	76.88															
Protein %		81.21	82.5	93.56	65.9															
Predictions			× − √	√	√															
HSB3A	2	1227	1205	22																
MMB3A	2	1203	1163	40																
DNA %		78	79.42	0																
Protein %		80.2	81.41	13.64																
Predictions			√ − ×	×																
HUMLORI	1	951	951																	
MMU09189	1	1446	1446																	
DNA %		78.86	78.86																	
Protein %		79.18	79.18																	
Predictions																				
HSU37055	18	2136	52	148	113	115	137	121	119	169	131	103	137	36	121	78	147	107	140	162
MUSHEPGFA	18	2151	52	148	113	115	137	121	119	196	131	103	137	36	109	78	147	107	140	162
DNA %		80.9	78.85	84.46	78.76	81.74	83.94	76.86	80.67	77.51	76.34	84.47	76.64	88.89	71.9	87.18	81.63	85.98	82.86	84.57
Protein %		78.51	69.23	81.08	76.99	78.26	83.21	76.86	83.19	71.01	75.57	84.47	76.64	91.67	76.86	84.62	81.63	78.5	68.57	83.33
Predictions			√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√	√
HSH11	1	648	648																	
MUSH1X	1	642	642																	
DNA %		73.15	73.15																	
Protein %		79.17	79.17																	
Predictions			√																	
HSU66875	3	294	73	137	84															
MMU63716	3	294	73	137	84															
DNA %		80.27	83.56	76.64	83.33															
Protein %		77.55	73.97	72.26	89.29															
Predictions			√	√	√															

HSINT2	3	720	220	104	396							
MMINT2	3	738	220	104	414							
DNA %		81.39	87.73	83.65	77.27							
Protein %		82.5	94.09	92.31	73.48							
Predictions			√	√	√ − ×							
HSBGL3	3	444	92	223	129							
MUSHBBMAJ	3	444	92	223	129							
DNA %		82.43	84.78	82.96	79.84							
Protein %		79.73	78.26	82.06	76.74							
Predictions			√	√	√							
HUMALIFA	3	609	19	179	411							
MUSALIFA	3	612	19	182	411							
DNA %		81.44	94.74	88.83	77.62							
Protein %		79.8	94.74	90.5	74.45							
Predictions			√	√	√							
HUMFABP	4	399	67	173	108	51						
MUSFABPI	4	606	103	173	108	222						
DNA %		81.2	85.07	81.5	82.41	72.55						
Protein %		60.15	13.43	79.77	66.67	41.18						
Predictions			× − √	√	√	√						
HUMLYTOXBB	4	735	162	46	72	455						
MMU16984	3	921	162	316		443						
DNA %		81.22	86.42	65.22	83.33	80.66						
Protein %		75.51	85.19	45.65	70.83	75.82						
Predictions			× − √	×	× − √	√						
HUMLYL1B	3	804	296	92	416							
MMLYL1	3	837	333	90	414							
DNA %		79.23	76.35	88.04	79.33							
Protein %		76.87	73.99	88.04	76.44							
Predictions			× − √	√	√							
HUMANFA	3	456	123	327	6							
MUSANF	3	459	120	327	12							
DNA %		80.92	75.61	82.87	83.33							
Protein %		78.95	63.41	84.4	100							
Predictions			√	√	×							
HUMG0S19A	3	279	73	115	91							
MMSCIMIP	3	279	76	112	91							
DNA %		78.85	82.19	78.26	76.92							
Protein %		75.27	82.19	75.65	69.23							
Predictions			√	√	√							
HUMALPI	11	1587	67	117	116	175	173	135	73	135	192	117	287
MUSIAP	11	1680	67	117	116	175	173	135	73	135	192	114	383
DNA %		79.84	89.55	85.47	83.62	85.71	80.92	80	68.49	78.52	81.25	69.23	76.31
Protein %		78.07	94.03	84.62	85.34	82.29	86.71	82.22	57.53	68.89	84.38	64.1	70.03
Predictions			× − √	√	√	√	√	√	√	√	√	√	√ − ×
HUMFPR1A	1	1053	1053										
MUSNFORREC	1	1095	1095										
DNA %		78.54	78.54										
Protein %		77.49	77.49										
Predictions			√										

HSSPRO	8	1437	64	120	345	140	157	153	345	113		
MMVITRO	8	1437	64	120	342	140	157	153	351	110		
DNA %		79.12	90.62	83.33	66.38	90.71	86.62	86.93	75.65	82.3		
Protein %		73.9	79.69	75	54.78	94.29	87.9	86.27	69.57	79.65		
Predictions			√	√	√	√	√	√	√	√		
HSGCSFG	5	624	40	164	108	147	165					
MMGCSFG	5	627	40	173	108	147	159					
DNA %		77.24	77.5	75.61	87.04	76.87	72.73					
Protein %		72.12	52.5	64.02	86.11	83.67	65.45					
Predictions			√	√ − ×	√	√	×					
HUMTNFBA	3	618	99	106	413							
MMTNFBG	3	609	96	100	413							
DNA %		79.13	72.73	69.81	83.05							
Protein %		71.84	57.58	56.6	79.18							
Predictions			× − √	√	√							
HSU16720	5	537	165	60	153	66	93					
MUSIL10Z	5	537	165	60	153	66	93					
DNA %		81.01	73.94	78.33	85.62	90.91	80.65					
Protein %		73.18	61.82	70	82.35	90.91	67.74					
Predictions			√	√	√	√	√					
HUMCP21OH	10	1488	202	90	155	102	102	87	201	179	104	266
MUS21OHA1	10	1464	202	90	143	96	102	87	201	170	104	269
DNA %		76.21	79.7	75.56	75.48	76.47	70.59	77.01	78.11	73.74	79.81	74.81
Protein %		70.56	78.71	70	73.55	76.47	47.06	68.97	71.64	70.39	72.12	68.8
Predictions			√	√	√	×	√	√	√	√	√	√
HUMMIS	5	1683	412	143	109	160	859					
MMAMH	5	1668	403	143	109	160	853					
DNA %		74.63	68.69	72.73	77.06	71.25	78.11					
Protein %		71.48	56.8	67.13	74.31	65.62	79.98					
Predictions			√	√	√	√	√					
HUMAPOE4	3	954	43	193	718							
MUSAPE	3	936	43	169	724							
DNA %		77.57	83.72	70.98	78.97							
Protein %		71.07	69.77	65.28	72.7							
Predictions			× − √	√	√							
HUMREGB	5	501	64	119	138	112	68					
MUSREGI	5	498	61	119	138	112	68					
DNA %		77.84	75	75.63	83.33	76.79	75					
Protein %		71.86	56.25	75.63	80.43	75	57.35					
Predictions			√	√	√	√	√					
HUMPROLA	1	1002	1002									
MUSPROL	1	1005	1005									
DNA %		76.45	76.45									
Protein %		71.56	71.56									
Predictions			√									
HSU29874	7	708	33	111	54	144	139	179	48			
MMU44024	7	699	33	122	46	144	144	189	21			
DNA %		70.48	93.94	65.77	74.07	76.39	79.86	72.63	8.333			
Protein %		67.37	90.91	70.27	72.22	75	73.38	62.01	18.75			
Predictions			√	×	√	√	√	√	×			

HSA6693	1	510	510									
MUSSER1	1	693	693									
DNA %		61.96	61.96									
Protein %		59.41	59.41									
Predictions			× – ×									
HUMIL2RGA	8	1110	115	154	185	140	163	97	70	186		
MMU21795	8	1110	115	154	185	143	163	97	67	186		
DNA %		80.63	71.3	82.47	79.46	77.86	80.98	82.47	82.86	86.02		
Protein %		69.73	44.35	74.03	71.35	64.29	64.42	71.13	77.14	85.48		
Predictions			× – √	√	√	√	√	√	√	√		
HUMCRPGA	2	675	61	614								
MMCRPG	2	678	64	614								
DNA %		76.15	72.13	76.55								
Protein %		69.78	54.1	71.34								
Predictions			√	√								
HSBCDIFFI	4	405	144	33	129	99						
MMIL5G	4	402	141	33	129	99						
DNA %		78.02	72.92	78.79	85.27	75.76						
Protein %		70.37	60.42	72.73	74.42	78.79						
Predictions			× – √	√	√	√						
HUMTHY1A	3	486	37	336	113							
MUSTHY1GC	3	489	37	339	113							
DNA %		77.57	86.49	73.81	85.84							
Protein %		69.14	72.97	63.39	84.96							
Predictions			√	√	√							
HSUPA	10	1296	57	28	108	175	92	220	149	141	149	177
MUSUPAA	10	1302	57	31	108	175	92	223	149	141	149	177
DNA %		77.47	73.68	60.71	79.63	82.29	84.78	76.36	71.81	77.3	80.54	75.14
Protein %		68.06	52.63	21.43	66.67	70.29	71.74	69.55	62.42	65.96	74.5	76.27
Predictions			√	√	√	√	√	√	√	√	√	× – √
HUMSAP01	2	672	64	608								
MUSSAPRB	2	675	67	608								
DNA %		74.4	84.38	73.36								
Protein %		68.75	65.62	69.08								
Predictions			√	√								
HUMPAP	5	528	76	119	138	127	68					
D63360	5	528	76	119	138	127	68					
DNA %		75.57	81.58	75.63	79.71	69.29	72.06					
Protein %		68.75	78.95	60.5	73.91	66.14	66.18					
Predictions			× – √	√	√	√	√					
HUMIL1B	6	810	47	52	202	165	131	213				
MMIL1BG	6	810	47	49	202	171	131	210				
DNA %		78.27	70.21	78.85	73.76	72.73	83.97	84.98				
Protein %		67.78	51.06	63.46	56.44	60	84.73	78.87				
Predictions			× – √	√	√	√	√	√				
HUMCAPG	5	768	55	148	136	255	174					
MUSCATHG	5	786	55	148	136	255	192					
DNA %		74.22	78.18	83.78	78.68	69.02	68.97					
Protein %		66.41	70.91	79.05	75	61.18	55.17					
Predictions			√	√	√	√	×					

Table 6.4: Analysis of alignments and results on the HUMCOMP/MUSCOMP test set

HUMCTLA1	5	744	55	148	136	261	144	
MUSSPCTLS	5	744	55	148	136	261	144	
DNA %		75.67	65.45	79.73	80.88	71.65	77.78	
Protein %		66.94	43.64	72.97	77.21	57.47	77.08	
Predictions			✓	✓	✓	✓	✓	
HSLACTG	4	429	133	159	76	61		
MUSALCALB	4	432	136	159	76	61		
DNA %		75.52	72.93	74.21	82.89	75.41		
Protein %		66.43	60.9	69.81	71.05	63.93		
Predictions			✓	✓	✓	✓		
HUMOSTP	6	945	54	39	81	42	324	405
MMOESTEOP	6	885	54	36	81	42	282	390
DNA %		71.22	75.93	71.79	82.72	88.1	64.51	71.85
Protein %		64.44	66.67	76.92	77.78	85.71	60.19	61.48
Predictions			✓	✓	✓	✓	× − ✓	✓
HSCD14G	2	1128	3	1125				
MMCD14	2	1101	3	1098				
DNA %		71.81	100	71.73				
Protein %		65.16	100	65.07				
Predictions			×	✓				
HSGAPIGNA	3	348	75	208	65			
MMU60528	3	351	75	211	65			
DNA %		75.86	78.67	71.63	86.15			
Protein %		65.52	68	59.13	83.08			
Predictions			✓	✓	✓			
HSBGPG	4	297	64	33	70	130		
MUSOGC	4	288	64	33	58	133		
DNA %		69.02	70.31	75.76	70	66.15		
Protein %		61.62	60.94	72.73	55.71	62.31		
Predictions			✓	✓	✓	✓		
HSAPOAIA	3	804	43	157	604			
MUSAICIIIA	1	9060	9060					
DNA %		75.5	83.72	80.89	73.51			
Protein %		13.06	41.86	11.46	11.42			
Predictions			✓	✓	✓			

Number of internal coding exons: 279
Number of internal coding exons predicted correctly: 263
Number of internal coding exons predicted correctly only on 5' end: 7
Number of internal coding exons predicted correctly only on 3' end: 5
Number of internal coding exons predicted correctly on neither end but partially covered: 0
Number of initial coding exons: 84
Number of initial coding exons predicted correctly: 52
Number of initial coding exons predicted correctly only on 5' end: 1
Number of initial coding exons predicted correctly only on 3' end: 26
Number of initial coding exons predicted correctly on neither end but partially covered: 1
Number of terminal coding exons: 84
Number of terminal coding exons predicted correctly: 71
Number of terminal coding exons predicted correctly only on 5' end: 4
Number of terminal coding exons predicted correctly only on 3' end: 2
Number of terminal coding exons predicted correctly on neither end but partially covered: 0
Number of genes with one coding exon: 33
Number of single gene coding exons predicted correctly: 23
Number of single gene coding exons predicted correctly only on 5' end: 3
Number of single gene coding exons predicted correctly only on 3' end: 6
Number of single coding exons predicted correctly on neither end but partially covered: 5
Number of genes: 117
Number of perfect genes: 51
Number of coding exons: 480
Number of predicted exons: 517
Number of coding exons of length greater than 50: 445
Number of predicted exons of length greater than 50: 462
Number of predicted exons overlapping no coding exon: 48
Number of splice sites in noncoding exons that are predicted: 19
Number of false negatives completely uncovered: 16
Number of nucleotides predicted to be coding: 103082
Number of nucleotides that are coding: 104557
Number of nucleotides predicted to be coding that are coding: 99599
Wrong exons (WE): 0.0928433
Missing exons (ME): 0.0333333
Nucleotide sensitivity: 0.952581
Nucleotide specificity: 0.966211
Nucleotide approximate correlation (AC): 0.950923
Exact exon sensitivity: 0.852083
Exact exon specificity: 0.791103
Covered exon sensitivity: 0.966667
Exact internal exon sensitivity: 0.942652

Table 6.5: Results with the multiple genes assumption, parsed in pieces.

Number of internal coding exons: 279
Number of internal coding exons predicted correctly: 262
Number of internal coding exons predicted correctly only on 5' end: 4
Number of internal coding exons predicted correctly only on 3' end: 6
Number of internal coding exons predicted correctly on neither end but partially covered: 0
Number of initial coding exons: 84
Number of initial coding exons predicted correctly: 52
Number of initial coding exons predicted correctly only on 5' end: 1
Number of initial coding exons predicted correctly only on 3' end: 27
Number of initial coding exons predicted correctly on neither end but partially covered: 1
Number of terminal coding exons: 84
Number of terminal coding exons predicted correctly: 70
Number of terminal coding exons predicted correctly only on 5' end: 4
Number of terminal coding exons predicted correctly only on 3' end: 1
Number of terminal coding exons predicted correctly on neither end but partially covered: 0
Number of genes with one coding exon: 33
Number of single gene coding exons predicted correctly: 23
Number of single gene coding exons predicted correctly only on 5' end: 3
Number of single gene coding exons predicted correctly only on 3' end: 5
Number of single coding exons predicted correctly on neither end but partially covered: 4
Number of genes: 117
Number of perfect genes: 57
Number of coding exons: 480
Number of predicted exons: 484
Number of coding exons of length greater than 50: 445
Number of predicted exons of length greater than 50: 449
Number of predicted exons overlapping no coding exon: 21
Number of splice sites in noncoding exons that are predicted: 17
Number of false negatives completely uncovered: 19
Number of nucleotides predicted to be coding: 101726
Number of nucleotides that are coding: 104557
Number of nucleotides predicted to be coding that are coding: 99391
Wrong exons (WE): 0.0433884
Missing exons (ME): 0.0395833
Nucleotide sensitivity: 0.950592
Nucleotide specificity: 0.977046
Nucleotide approximate correlation (AC): 0.956306
Exact exon sensitivity: 0.847917
Exact exon specificity: 0.840909
Covered exon sensitivity: 0.960417
Exact internal exon sensitivity: 0.939068

Table 6.6: Results with the parsed in pieces assumption.

Number of internal coding exons: 279
Number of internal coding exons predicted correctly: 260
Number of internal coding exons predicted correctly only on 5' end: 4
Number of internal coding exons predicted correctly only on 3' end: 6
Number of internal coding exons predicted correctly on neither end but partially covered: 2
Number of initial coding exons: 84
Number of initial coding exons predicted correctly: 50
Number of initial coding exons predicted correctly only on 5' end: 1
Number of initial coding exons predicted correctly only on 3' end: 26
Number of initial coding exons predicted correctly on neither end but partially covered: 2
Number of terminal coding exons: 84
Number of terminal coding exons predicted correctly: 69
Number of terminal coding exons predicted correctly only on 5' end: 4
Number of terminal coding exons predicted correctly only on 3' end: 1
Number of terminal coding exons predicted correctly on neither end but partially covered: 2
Number of genes with one coding exon: 33
Number of single gene coding exons predicted correctly: 23
Number of single gene coding exons predicted correctly only on 5' end: 3
Number of single gene coding exons predicted correctly only on 3' end: 5
Number of single coding exons predicted correctly on neither end but partially covered: 4
Number of genes: 117
Number of perfect genes: 55
Number of coding exons: 480
Number of predicted exons: 487
Number of coding exons of length greater than 50: 445
Number of predicted exons of length greater than 50: 449
Number of predicted exons overlapping no coding exon: 26
Number of splice sites in noncoding exons that are predicted: 17
Number of false negatives completely uncovered: 21
Number of nucleotides predicted to be coding: 101693
Number of nucleotides that are coding: 104557
Number of nucleotides predicted to be coding that are coding: 98881
Wrong exons (WE): 0.0533881
Missing exons (ME): 0.04375
Nucleotide sensitivity: 0.945714
Nucleotide specificity: 0.972348
Nucleotide approximate correlation (AC): 0.950529
Exact exon sensitivity: 0.8375
Exact exon specificity: 0.825462
Covered exon sensitivity: 0.95625
Exact internal exon sensitivity: 0.9319

Table 6.7: Results with the multiple gene assumption and double strand assumption

Number of internal coding exons: 279
Number of internal coding exons predicted correctly: 251
Number of internal coding exons predicted correctly only on 5' end: 8
Number of internal coding exons predicted correctly only on 3' end: 10
Number of internal coding exons predicted correctly on neither end but partially covered: 1
Number of initial coding exons: 84
Number of initial coding exons predicted correctly: 71
Number of initial coding exons predicted correctly only on 5' end: 2
Number of initial coding exons predicted correctly only on 3' end: 7
Number of initial coding exons predicted correctly on neither end but partially covered: 0
Number of terminal coding exons: 84
Number of terminal coding exons predicted correctly: 72
Number of terminal coding exons predicted correctly only on 5' end: 1
Number of terminal coding exons predicted correctly only on 3' end: 2
Number of terminal coding exons predicted correctly on neither end but partially covered: 1
Number of genes with one coding exon: 33
Number of single gene coding exons predicted correctly: 27
Number of single gene coding exons predicted correctly only on 5' end: 1
Number of single gene coding exons predicted correctly only on 3' end: 2
Number of single coding exons predicted correctly on neither end but partially covered: 1
Number of genes: 117
Number of perfect genes: 79
Number of coding exons: 480
Number of predicted exons: 476
Number of coding exons of length greater than 50: 445
Number of predicted exons of length greater than 50: 428
Number of predicted exons overlapping no coding exon: 19
Number of splice sites in noncoding exons that are predicted: 6
Number of false negatives completely uncovered: 24
Number of nucleotides predicted to be coding: 97818
Number of nucleotides that are coding: 104557
Number of nucleotides predicted to be coding that are coding: 96683
Wrong exons (WE): 0.039916
Missing exons (ME): 0.05
Nucleotide sensitivity: 0.924692
Nucleotide specificity: 0.988397
Nucleotide approximate correlation (AC): 0.947591
Exact exon sensitivity: 0.877083
Exact exon specificity: 0.884454
Covered exon sensitivity: 0.95
Exact internal exon sensitivity: 0.899642

Table 6.8: Results with the single gene assumption.

Number of internal coding exons: 279
Number of internal coding exons predicted correctly: 253
Number of internal coding exons predicted correctly only on 5' end: 10
Number of internal coding exons predicted correctly only on 3' end: 6
Number of internal coding exons predicted correctly on neither end but partially covered: 3
Number of initial coding exons: 84
Number of initial coding exons predicted correctly: 55
Number of initial coding exons predicted correctly only on 5' end: 2
Number of initial coding exons predicted correctly only on 3' end: 19
Number of initial coding exons predicted correctly on neither end but partially covered: 1
Number of terminal coding exons: 84
Number of terminal coding exons predicted correctly: 68
Number of terminal coding exons predicted correctly only on 5' end: 3
Number of terminal coding exons predicted correctly only on 3' end: 2
Number of terminal coding exons predicted correctly on neither end but partially covered: 1
Number of genes with one coding exon: 33
Number of single gene coding exons predicted correctly: 20
Number of single gene coding exons predicted correctly only on 5' end: 2
Number of single gene coding exons predicted correctly only on 3' end: 9
Number of single coding exons predicted correctly on neither end but partially covered: 2
Number of genes: 117
Number of perfect genes: 50
Number of coding exons: 480
Number of predicted exons: 524
Number of coding exons of length greater than 50: 445
Number of predicted exons of length greater than 50: 503
Number of predicted exons overlapping no coding exon: 68
Number of splice sites in noncoding exons that are predicted: 16
Number of false negatives completely uncovered: 26
Number of nucleotides predicted to be coding: 114646
Number of nucleotides that are coding: 104557
Number of nucleotides predicted to be coding that are coding: 102460
Wrong exons (WE): 0.129771
Missing exons (ME): 0.0541667
Nucleotide sensitivity: 0.979944
Nucleotide specificity: 0.893708
Nucleotide approximate correlation (AC): 0.92242
Exact exon sensitivity: 0.825
Exact exon specificity: 0.755725
Covered exon sensitivity: 0.945833
Exact internal exon sensitivity: 0.90681

Table 6.9: `GENSCAN` results on the HUMCOMP dataset

Part II

Combinatorics

Overview

The remainder of this thesis is devoted to combinatorial problems that originate from domino tiling questions. The bulk of the work is unrelated to the gene recognition problem discussed in the previous chapters, although there is one mathematical link between the two sections which the avid reader will undoubtedly find.

Chapter 7 is about the forcing numbers of matchings on different graphs. The forcing number of a perfect matching M of G is defined as the smallest number of edges in a subset $S \subset M$, such that S is in no other perfect matching. The forcing number of a matching was first defined by Harary, Klein and Živković [32], although it had already been considered by chemists [49, 50, 72] because of its applicability for resonance energy estimators of molecules. For example, the *innate degree of freedom* of a graph (the sum of forcing numbers over all matchings) was compared to classical resonance energy estimators in [49].

We consider forcing numbers of matchings on square grids (as opposed to polyhexes analyzed by chemists) and show that for the $2n \times 2n$ square grid, the forcing number of any perfect matching is bounded below by n and above by n^2. Both bounds are sharp. We also establish a connection between the forcing problem and the minimum feedback set problem.

Finally, in Chapter 8, we develop some of the combinatorial tools we used for forcing problems and discuss their applicability to proving combinatorial "power of 2" results for the number of domino tilings of certain classes of polyominoes. The enumeration of domino tilings of square grids was first undertaken by Kasteleyn [47], in an attempt to understand the absorption of dimers on a two dimensional lattice (and also because of its relationship to the Ising problem). While Kasteleyn succeeded in providing a closed form solution for the number of tilings of the $2n \times 2n$ square grid, it is not obvious from his solution that the number of tilings is of the form $2^n(2k+1)^2$ (a fact partially explained by Jockusch [41] and others). We give the first complete combinatorial proof of this fact thus settling a question raised in [75]. Despite our success with this problem, the ubiquitous appearance of powers of 2 in enumeration formulas for the number of domino tilings of polyominoes remains a mystery, which we perpetuate by adding a number of new conjectures to the existing literature.

Chapter 7

Forcing Matchings

7.1 Introduction

The notion of the forcing number of a matching was introduced by Harary, Klein and Živković in [32]:

Definition 7.1 *Let G be a graph that admits a perfect matching. The forcing number of a perfect matching M of G is defined as the smallest number of edges in a subset $S \subset M$, such that S is in no other perfect matching. The forcing number of M is denoted $\varphi(M)$. A subset S with the property above is said to force M.*

The concept of forcing is related to some problems in chemistry (see [49, 50, 72]). The investigation of forcing in the context of chemistry has led to the extensive study of forcing in hexagonal systems (see for example [32, 93, 92]). Surprisingly, few other classes of graphs have been considered. Here we consider the forcing problem for the square grid:

Our main result is an upper and lower bound for the forcing numbers of perfect matchings of R_n ($R_n = P_{2n} \oplus P_{2n}$):

Theorem 7.1 *Let M be a perfect matching of R_n. The forcing number of M is bounded by*

$$n \leq \varphi(M) \leq n^2. \tag{7.1}$$

The upper bound is the easier of the two bounds, we give both a constructive proof (in Section 7.3) and a nonconstructive proof (Section 7.5). The lower bound, which is more difficult, is proved in Section 7.4.

7.2 Preliminaries

We will be using the following definitions and conventions in the article:

Definition 7.2 *An alternating path in a matching M is a sequence $v_1, e_1, v_2, e_2, v_3, e_3, v_4, e_4, \ldots, v_{n-1}, e_{n-1}, v_n$ satisfying:*

v_i and v_{i+1} belong to the edge e_i.
$e_i \in M$ when i is odd and $e_i \notin M$ when i is even.
Alternating cycles are alternating paths where the final vertex is the same as the initial vertex.

Edges in an alternating path which are not in the matching will be called *alternate edges.* If all the edges in an alternating path (respectively. cycle) are distinct, the alternating path (respectively. cycle) will be called *simple.*

Definition 7.3 *We shall denote by c(M) the maximum number of disjoint, simple, alternating cycles in a matching M of a graph G.*

Proposition 7.1 *Let G be a graph with a perfect matching M. Then $\varphi(M) > 0$ if and only if $c(M) > 0$.*

Proof: Every simple alternating cycle in M must contain a forcing edge because otherwise we can replace the matching edges of the cycle with the alternate edges of the cycle to obtain a new perfect matching. It follows that $\varphi(M) \geq c(M)$ and so $c(M) > 0 \Rightarrow \varphi(M) > 0$. Now suppose that $\varphi(M) > 0$. Consider an edge $e_1 = (v_1, v_2)$ in M that is not forced. Examine the neighbors of v_2. If all the vertices in the neighborhood are endpoints of forced edges, then e_1 is forced. Therefore, there must be a neighbor (other than v_1) that is the endpoint of an edge that is not forced. Call this vertex v_3 and the edge e_2. We can now extend our alternating path by examining the neighborhood of the other endpoint of e_2. Since G is finite, and we can keep extending our alternating path, we must eventually return to a vertex already in the alternating path. It follows that $c(M) > 0$.

7.3 The Upper Bound

The perfect matching M in which every edge lies in the same direction shows that the upper bound is sharp. There are n^2 disjoint alternating cycles and since every alternating cycle must contain a forced edge, $\varphi(M) \geq n^2$. Figure 7-1 shows M together with the edges which force it. Forced edges are dark.

Proof of the upper bound: Embed R_n in R^2 so that the boundaries are parallel to the xy-axes and each edge has unit length. Place R_n in the first quadrant with one corner at $(0, 0)$. Every vertex has a label (i, j) where $1 \leq i, j \leq 2n$. Let $F \subset M$ be the collection of edges in M which contain a vertex whose co-ordinates are both even.

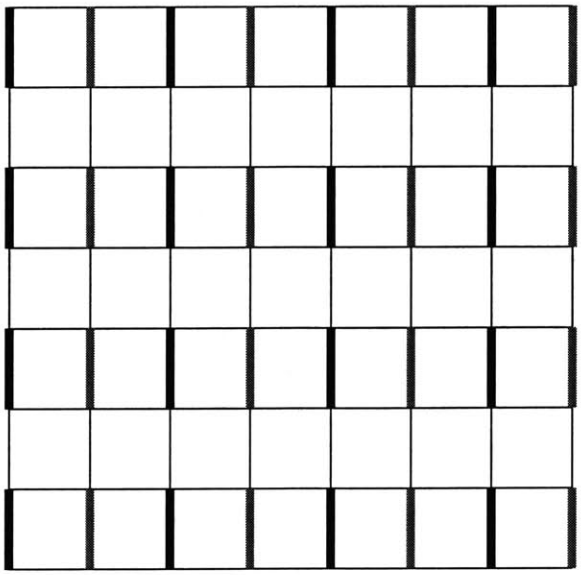


Figure 7-1: Forced tiling (upper bound)

Lemma 7.1 *$M - F$ has no simple alternating cycles.*

Proof of the lemma: Let $C = v_1, v_2, \ldots, v_n$ be any cycle in G that does not contain any vertex with both co-ordinates even. Suppose, also, that all the vertices in C are distinct (except for the first an final vertex) so that C encloses some region S in the plane. Let the number of vertices in C be b, the number of vertices in S be i, and let the area of S be A. Notice that for all odd j, the edges (v_j, v_{j+1}) and (v_{j+1}, v_{j+2}) lie in the same direction because C avoids all vertices with both co-ordinates even. We can thus encode C with a sequence whose elements consist of Up (U), Down (D), Left (L) and Right (R). Each symbol represents two edges in C. Note that the number of U's must equal the number of D's, and the number of L's must equal the number of R's. This defines a bijection between simple cycles avoiding vertices with both co-ordinates even and simple cycles on the unit lattice (otherwise known as polyominoes or animals). Graphically the bijection is given by:

It follows that b and A are divisible by four. We can conclude that i is odd (this follows by induction on A). Alternatively, the result follows from the more general result of Pick [29]:

Theorem 7.2 (Pick) *The area of a simple lattice polygon P is given by*

$$A = i + \frac{b}{2} - 1 \tag{7.2}$$

where i is the number of lattice points in the interior of P, and b is the number of lattice points on the boundary of P.

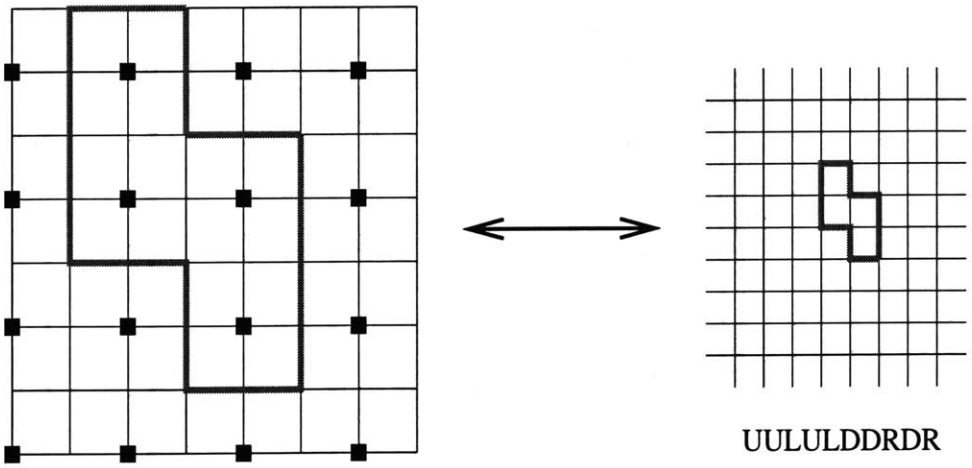


Figure 7-2: The bijection

If $M - F$ contained a simple alternating cycle C, there would be no way to match all the vertices in S (because i is odd).

Since $c(M - F) = 0$ it follows from Proposition 7.1 that F forces M.

Figure 7-2 shows an example of a cycle C and its corresponding encoding under the bijection. In this case, $i = 7, b = 20$ and $A = 16$. The dark vertices are the ones which C must avoid.

7.4 The Lower Bound

We begin by showing that the lower bound is sharp. Figure 7-3 shows a perfect matching M of R_4 with $\varphi(M) = 4$. As before, the forced edges are dark. The general construction for a perfect matching M of R_n with $\varphi(M) = n$ consists of n concentric simple alternating cycles arranged as in Figure 7-3. The forcing edges are staggered in a stepwise fashion upwards towards the center, beginning with a horizontal edge in the corner. A total of n edges are used for R_n. Notice that by construction, M has n simple alternating cycles; they are just the concentric rings. It follows that $\varphi(M) \geq n$. It is easy to verify that the forcing edges described above do in fact force M so that $\varphi(M) = n$.

Theorem 7.3 *Any perfect matching M in R_n can be decomposed into at least n simple, disjoint, alternating cycles.*

Proof of the theorem: We apply the method of proof introduced by Ciucu in his factorization theorem [16]. Embed R_n in the plane so that all edges have the same length and are parallel to the x, y axes. Let l be the diagonal line from the bottom left hand corner to the upper right hand corner. Notice that l is an axis of symmetry for R_n. Let the vertices which l intersects be labelled alternately $a_1, b_1, a_2, b_2, \ldots, a_n, b_n$ (See Figure 7-4).

Let M be any perfect matching of R_n. Let M' be the matching obtained by reflecting M across the line l and define $D = M \cup M'$ (D is allowed to have multiple

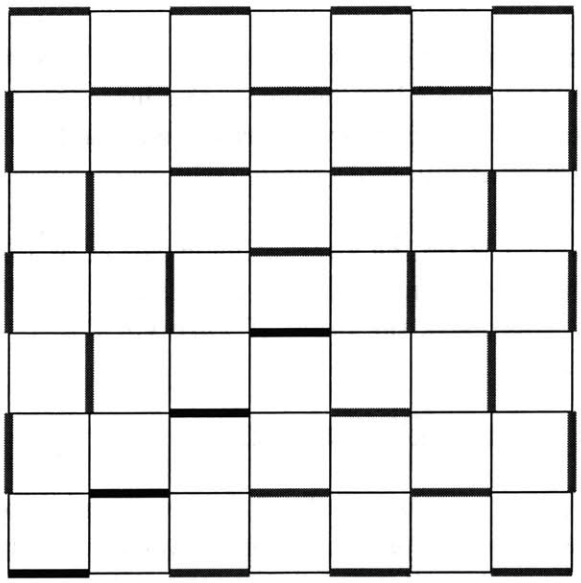


Figure 7-3: Forced tiling (lower bound)

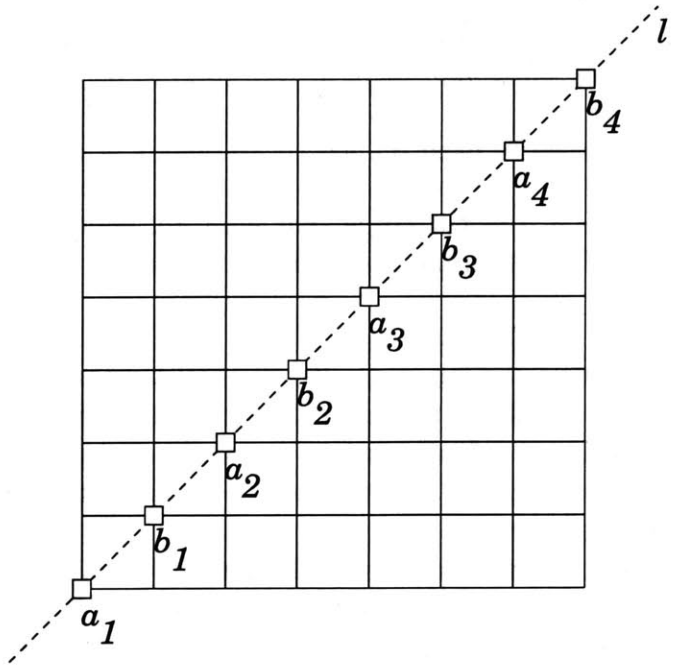


Figure 7-4: The square grid with its axis of symmetry and labelled diagonal

edges). Notice that D is a 2-factor of G and is therefore a disjoint union of even-length cycles. Furthermore, since D is symmetric across l, any cycle maps to another cycle under the reflection.

Now define C_i' to be the cycle containing a_i. C_i' can have at most one other vertex on l because every vertex in C_i' has degree 2. Furthermore, such a vertex must be of the type b_j, for otherwise the number of vertices enclosed by C is odd (contradicting the fact that D is a disjoint union of even length cycles). It follows that all the cycles C_i' are distinct.

Finally, let $C_i = C_i' \cap M$ be the alternating cycles in M obtained from C_i'. By the above arguments, the alternating cycles C_i are disjoint and there are n of them.

The above result completes the proof of Theorem 7.1.

7.5 A Min Max Theorem

We rely on Proposition 7.1 in our proofs of the lower and upper bound. Even though it is all that is necessary in our proofs, a much stronger result can be proved that has consequences for a large class of graphs other than R_n.

Definition 7.4 *Let G be a finite directed graph. A feedback set is a set of edges in G that contains at least one edge of each directed cycle of G.*

The following theorem of Lucchesi and Younger [60] relates the number of disjoint cycles in a directed graph to the minimal size of a feedback set:

Theorem 7.4 *For a finite planar directed graph, a minimum feedback set has cardinality equal to that of a maximum disjoint collection of directed cycles.*

The theorem has recently been refined by Barahona et al. [6]:

Theorem 7.5 *If D is a directed graph that does not contain a subdivision of $K_{3,3}$ then the cardinality of a minimum feedback set is equal to the maximum number of edge disjoint cycles.*

Using the terminology of Alon et al. [2], we shall say that a directed graph G has the *cycle-packing property* if the maximum size of a collection of edge disjoint cycles equals the minimum size of a feedback set. An undirected graph will be said to have the cycle-packing property if every orientation of the edges results in a directed graph with the cycle-packing property.

We now apply the above theorems to the forcing problem. Construct a digraph $D(M)$ from a perfect matching M of a bipartite graph G with the cycle-packing property in the following way: Let the vertex set of $D(M)$ be the vertex set of G. Since G is bipartite, the vertices can be naturally partitioned into two disjoint sets. Label the sets A and B. If $e \in M$, direct e from A to B. If $e \notin M$, direct e from B to A.

The following observation is trivial:

Lemma 7.2 *There is a one to one correspondence between alternating cycles in M and directed cycles in $D(M)$. Furthermore, two cycles in $D(M)$ intersect on an edge or not at all.*

There is also a natural correspondence between forcing sets in M and feedback sets in $D(M)$. In particular, we have:

Lemma 7.3 *For every feedback set in $D(M)$ there exists a forcing set in M of the same cardinality.*

Proof: Let F_M be a feedback set in $D(M)$. If all the edges in F_M lie in M, then by Proposition 7.1, F_M is a forcing set for M. If there exists an edge $e \in F_M$ and $e \notin M$ then there exists a unique edge $f \in M$ with the head of f being the tail of e. Any cycle in $D(M)$ passing through e must pass through f. We can therefore remove e from F_M and add in f. We repeat this process until all the edges in F_M lie in M.

The converse of the lemma is also true because any forcing set for M must be a feedback set in $D(M)$. Furthermore, $D(M)$ has the cycle-packing property because G has the cycle-packing property.

Theorem 7.6 *For any perfect matching M of a bipartite graph G with the cycle-packing property,*

$$\varphi(M) = c(M). \tag{7.3}$$

The above theorem and lemmas show that for bipartite graphs with the cycle-packing property the forcing problem for matching is equivalent to determining the number of disjoint alternating cycles in a matching. We can use Theorem 7.6 to obtain an upper bound for the forcing number of any perfect matching in such a graph.

Proposition 7.2 *Let G be a bipartite graph of girth g with the cycle-packing property. Suppose also that G has p vertices and admits a perfect matching. For any perfect matching M of G,*

$$\varphi(M) \leq \left\lfloor \frac{p}{g} \right\rfloor. \tag{7.4}$$

Proof: Any simple alternating cycle must contain at least g vertices. From the second part of Lemma 7.3 it follows that the maximum number of disjoint simple alternating cycles in M is $\lfloor \frac{p}{g} \rfloor$. Applying Theorem 7.6, $\varphi(M) \leq \lfloor \frac{p}{g} \rfloor$.

Applying Proposition 7.2 to R_n, we obtain an alternative proof of the upper bound in Theorem 7.1. We can also use Proposition 7.2 to obtain an upper bound for the hexagonal systems discussed in [93, 92, 32]. Since any hexagonal system has girth 6, for any perfect matching M of an hexagonal system with p vertices, $\varphi(M) \leq \lfloor \frac{p}{6} \rfloor$.

We also remark that for planar graphs there exist polynomial time algorithms for finding feedback sets. In [27], Gabow presents an $O(n^3)$ algorithm. We can use such algorithms on $D(M)$ to find forcing sets for perfect matchings in bipartite planar graphs.

7.6 Other Problems

Our results can be extended to arbitrary rectangular grids of the form $P_n \oplus P_m$ where mn is even. However, the behavior of the lower bound for such graphs seems more complicated than that of the square grid. Also of interest are the forcing numbers of perfect matchings of graphs other than rectangular grids. Let $T_n = C_{2n} \oplus C_{2n}$ be the $2n \oplus 2n$ torus:

Conjecture 7.1 *Let M be a perfect matching of T_n. The forcing number of M satisfies*

$$\varphi(M) \geq 2n. \tag{7.5}$$

The n dimensional hypercube, Q_n, also seems interesting.

Problem 7.1 *Let M be a perfect matching of Q_n. Find good upper and lower bounds on $\varphi(M)$.*

Some investigation of the forcing numbers of matchings for low dimensional hypercubes suggests that for any perfect matching M in the hypercube Q_n,

$$\varphi(M) = \frac{2^n}{4} \tag{7.6}$$

The following counting argument (due to Noga Alon) dispels the possibility that (7.6) holds for all n:

Lemma 7.4 *For sufficiently large n, there exists perfect matchings M of Q_n with*

$$\varphi(M) > \frac{2^n}{4} \tag{7.7}$$

Proof: Let $f(n)$ denote the number of perfect matchings of Q_n. We use the well known fact that

$$f(n) \geq \left(\frac{n}{e}\right)^{2^{n-1}} \tag{7.8}$$

If we assume that $\varphi(M) \leq \frac{2^n}{4}$ for every perfect matching M of Q_n, we see that

$$f(n) \leq \binom{2^n}{2^{n-1}} n^{2^{n-2}} \tag{7.9}$$

Taking the 2^{n-1}th root of (7.8) and (7.9) we obtain a contradiction.

Given the above result, it would be interesting to find the extremal perfect matchings (with respect to the forcing number) in Q_n. Presumably the perfect matching consisting of edges all in the same direction is the extremal configuration for a lower bound of $\varphi(M) \geq 2^{n-2}$.

Both Q_n and T_n do not have the cycle-packing property (except for small cases), so Theorem 7.6 does not apply. Furthermore, there is no known analogue of Theorem 7.3 for these graphs.

Chapter 8

Tilings of Grids and Power of 2 Conjectures

8.1 Introduction

The number of domino tilings of the $n \times m$ square grid was first calculated in a seminal paper by Kasteleyn [47]. He showed that, for n, m even, the number of tilings $N(n, m)$ is given by

$$N(n,m) = \prod_{j=1}^{\frac{n}{2}} \prod_{k=1}^{\frac{m}{2}} (4\cos^2 \frac{\pi j}{n+1} + 4\cos^2 \frac{\pi k}{m+1}). \tag{8.1}$$

This result, while interesting in its own right, does not reveal all of the properties of $N(n, m)$ at first glance. For example, $N(2n, 2n)$ is either a perfect square or twice a perfect square (such a number is called **squarish**). This was first proved by Montroll [59] using linear algebra and later proved by Jokusch [41] and others. Another interesting observation is that

$$N(2n, 2n) = 2^n(2k+1)^2. \tag{8.2}$$

A derivation of this fact from (8.1) has been obtained independently by a number of authors; we refer the reader to [42]. A combinatorial proof of (8.2) has proved more elusive, although partial results have been established [16]. Our main result is a direct combinatorial proof of (8.2). Our proof illuminates the combinatorics behind $N(2n, 2n)$ and leads directly to generalizations.

Interestingly, perhaps because of the closed form of equation (8.1), observations other than the ones mentioned above have been scarce. Propp has remarked [71] that "Aztec diamonds and their kin have (so far) been much more fertile ground for exact combinatorics than the seemingly more natural rectangles".

We show that there is a rich source of problems to be found in the enumeration of perfect matchings of rectangular grids. In fact, it seems that the tools needed to resolve many of the problems have yet to be discovered.

8.2 The square grid

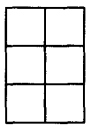
8.2.1 Even Squares

Theorem 8.1 *Let $N(2n, 2n)$ be the number of domino tilings of the $2n \times 2n$ square grid.*

$$N(2n, 2n) = 2^n(2k+1)^2. \tag{8.3}$$

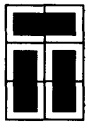
Our proof is broken down into two parts. The first part is not new, in fact it appears as a very special case in a theorem in [16]. Since we are interested in this special case only, we provide a simplified version of the proof in [16] that sacrifices much of the generality but illustrates the elegant combinatorial nature of the argument.

We begin by introducing the notation we will use. Rather than discussing perfect matchings of graphs, we will use the dual graph and think of edges in the perfect matching as dominoes covering two adjacent squares. We will, on occasion, use the two descriptions interchangeably. For an arbitrary region R, we will use the notation $\#\ R$ for the number of domino tilings of R. For example,

$\#$  $= 3.$

We will use the notation $\#_2 R$ for the parity of the number of domino tilings of R.

The **direction** of a domino from a fixed square is either up, down, left or right. We shall say that a domino is **oriented** in the **positive** (respectively **negative**) direction from a given square if its direction is up or to the right (respectively down or to the left). For example, in the tiling below, the top left square has a domino that is *positively oriented* and whose direction is *right.*



Lemma 8.1 *Label the diagonal squares on the $2n \times 2n$ square grid from the bottom left to the top right with the labels $a_1, b_1, a_2, b_2, \ldots, a_n, b_n$. The number of domino tilings of the square grid with dominoes placed at $a_1, a_2, \ldots, a_n$ is dependent only on the orientation of the dominoes and not their direction.*

Proof of lemma: Let M be any domino tiling of the $2n \times 2n$ square grid. Let M' be the tiling obtained by reflecting M across the diagonal and define $D = M \cup M'$ (D is allowed to consist of multiple dominoes). Notice that in the dual graph of the $2n \times 2n$ square grid, D is a 2-factor and is therefore a disjoint union of even-length cycles. Furthermore, since D is symmetric across across the diagonal, any cycle maps to another cycle under the reflection.

Now define C_i' to be the cycle containing a_i. C_i' can have at most one other vertex on the diagonal because every vertex in C_i' has degree 2. Furthermore, such a vertex

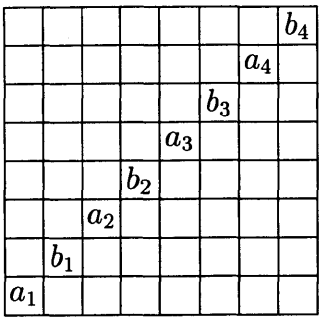
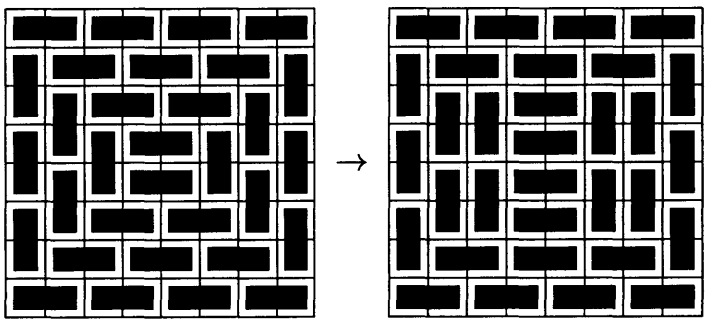


Figure 8-1: Labeling of the diagonal

must be of the type b_j, for otherwise the number of vertices enclosed by C is odd (contradicting the fact that D is a disjoint union of even length cycles). It follows that all the cycles C_i' are distinct.

Finally, let $C_i = C_i' \cap M$ be the alternating cycles (cycles in the dual graph alternating between edges in the tiling and edges not in the tiling) in M obtained from C_i'. By the above arguments, the alternating cycles C_i are disjoint. Thus, there is a bijection between any two sets of tilings with fixed dominoes of the same orientation on the a_i's. We simply select all the dominoes on the a_i's that have switched direction and rotate the appropriate alternating cycles.

Example 8.1 *Changing the direction of the domino at a_2 we have*



We now define a class of grids, H_n (first introduced by Ciucu [16]), as follows:

Notation 8.1 *H_1 consists of two adjacent horizontal squares. H_n is defined from H_{n-1} by adding a grid of size $2 \times (2n-1)$ to the left of H_{n-1}.*

Lemma 8.2 *The number of domino tilings of the square grid is given by*

$$N(2n, 2n) = 2^n (\#H_n)^2. \tag{8.4}$$

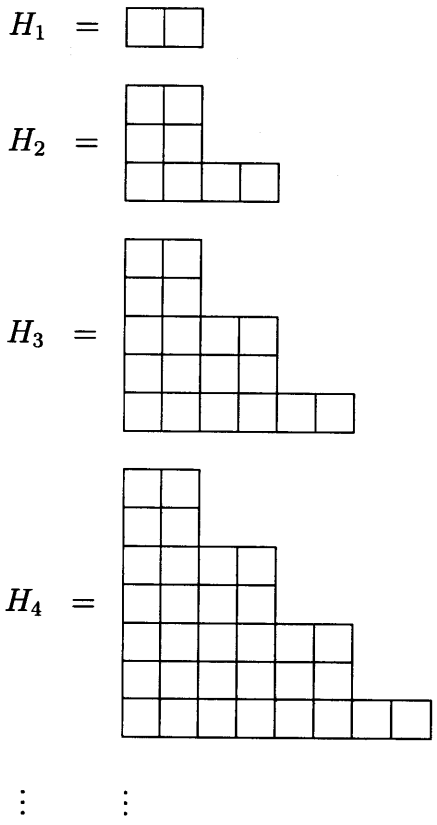


Figure 8-2: The grids H_n

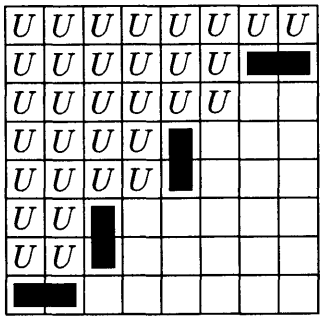


Figure 8-3: A reduced configuration

Proof of lemma: Consider a fixed orientation for the dominoes covering the a_i's. We can assume (using Lemma 8.1) that the directions of the dominoes are all either down or to the right (call such a configuration **reduced**). Notice that the square grid decomposes naturally into two halves. Figure 8-3 illustrates an example of a reduced configuration.

Notice that the region filled with U is equivalent to H_n, as is its complement. Now consider the standard checkerboard 2-coloring of the square grid. All the U's which are adjacent to empty squares have the same color. It follows that in any reduced configuration, every domino covers either two U's or none at all. We have from Lemma 8.1 that

$$N(2n, 2n) = 2^n \sum_C \#C \tag{8.5}$$

where C ranges over all reduced configurations. From the remarks above it follows that

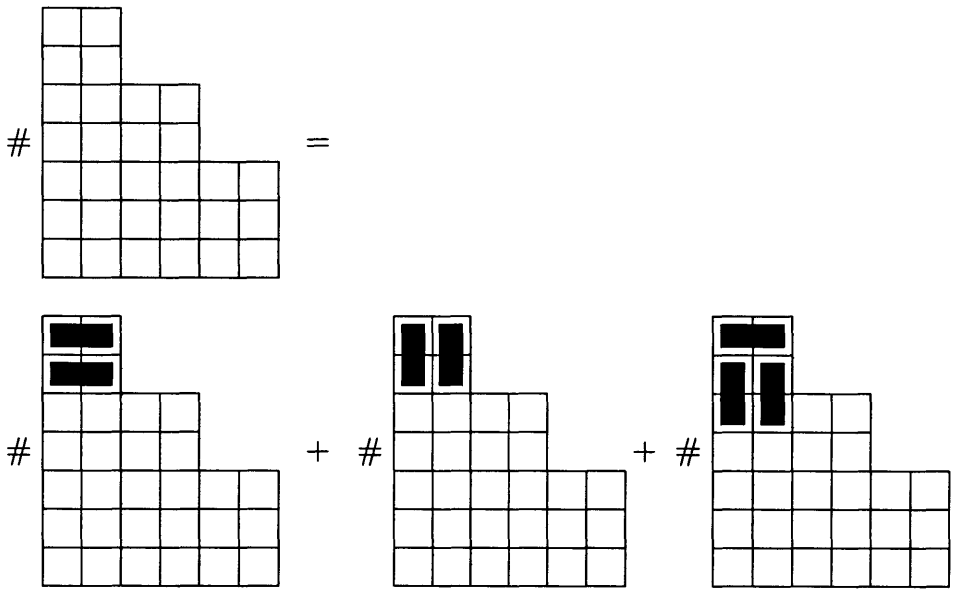
$$\sum_C \#C = (\#H_n)^2, \tag{8.6}$$

which completes the proof of the lemma.

Lemma 8.3 *$\#H_n$ is odd.*

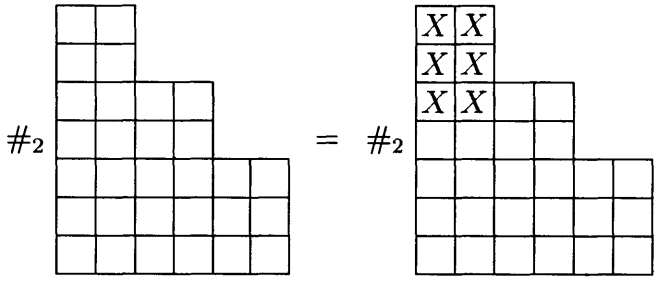
Proof of lemma: Our proof is by induction. The case when $n = 1, 2$ is trivial. We illustrate the general case by showing the step $n = 3 \Rightarrow n = 4$.

Begin by observing that



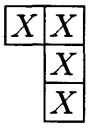
(8.7)

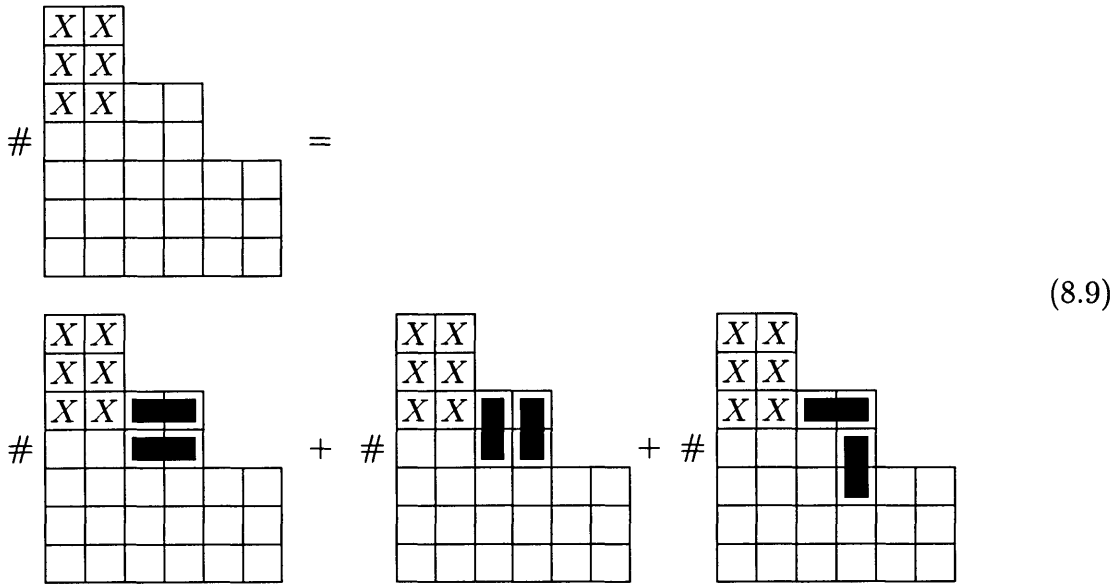
The first two terms in (8.7) are equal, so we have



(8.8)

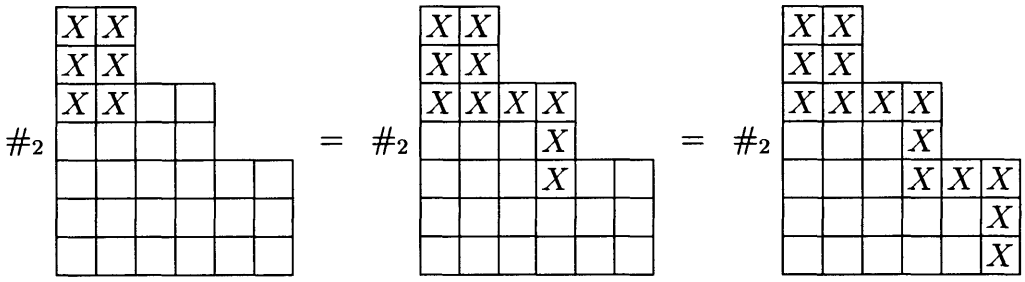
where the X's denote squares that cannot be used.

We now begin removing shapes of the form  from the diagonal, using a similar idea:



(8.9)

Hence, we can conclude that



(8.10)

Our last shape is H_{n-1} (minus the forced domino on the bottom right), flipped and rotated by 90°! It follows that

$$\#_2 H_n = \#_2 H_{n-1}. \tag{8.11}$$

Proof of theorem: The theorem follows immediately by applying Lemmas 8.2 and 8.3.

8.2.2 Odd Squares

Odd Squares clearly have no tilings with dominoes, but by deleting one square from the border we can make them tileable (Figure 8-4). It follows from Temperley's bijection, that the removal of any appropriate border square (one must remove a square that leaves a balanced bipartite graph), results in the same number of tilings for the resulting region [48] We propose a "squarish" conjecture for these regions and suggest a combinatorial approach.

We will use the notation $\overline{N}(n,m)$ to denote an $n \times m$ rectangular grid with a border square removed (if n, m are odd we assume that the removed square is of the

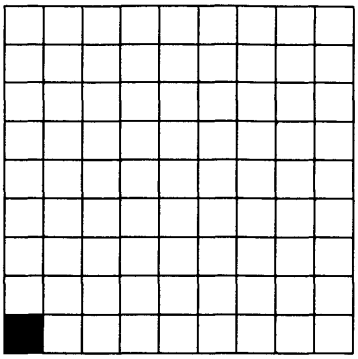


Figure 8-4: An Odd Square with a corner removed

appropriate type leaving a balanced bipartite graph).

Conjecture 8.1 $\frac{N(2n+1,2n+1)}{n+1}$ *is squarish.*

We may assume without loss of generality that we have removed the bottom left hand border square. The resulting figure is symmetric and thus we can apply Lemma 8.2 to understand the number of tilings in terms of the number of tilings of two smaller regions.

These smaller regions belong to two classes of grids (similar to H_n), which we denote by S_n ("symmetric") and D_n ("deleted"):

Notation 8.2 *S_1 is a 2×2 square grid. S_n is defined from S_{n-1} by adding a grid of size $2 \times (2n)$ to the left of S_{n-1}.*

Notation 8.3 *D_1 is an "L" shaped 4-omino (see Figure 8-6). D_n is defined from D_{n-1} by adding a grid of size $2 \times (2n-1)$ to the left of D_{n-1}, together with two additional squares on the top left hand corner.*

Lemma 8.4 *The number of domino tilings of the square grid with a corner removed is given by*

$$\overline{N}(2n+1, 2n+1) = 2^n(\#S_n)(\#D_n). \tag{8.12}$$

Proof: The proof is identical to that of Lemma 8.2.

Conjecture 8.2

$$\#S_n = (n+1)\#D_n \tag{8.13}$$

Conjecture 8.1 follows from Conjecture 8.2

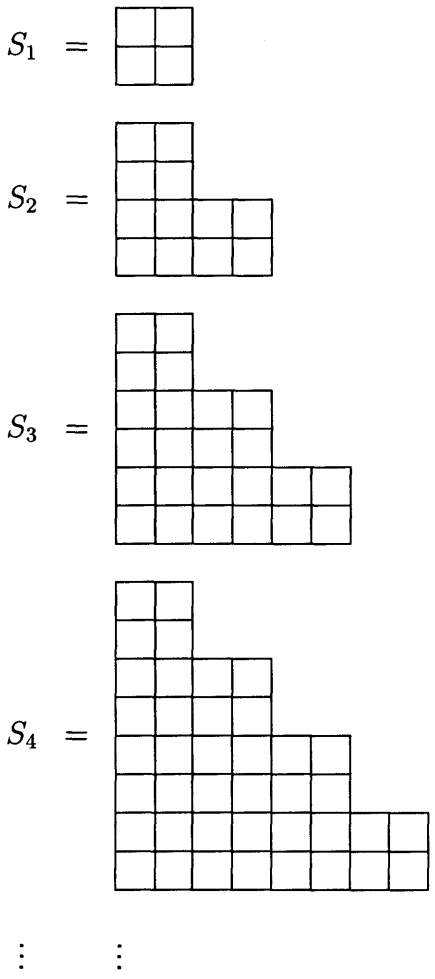


Figure 8-5: The grids S_n

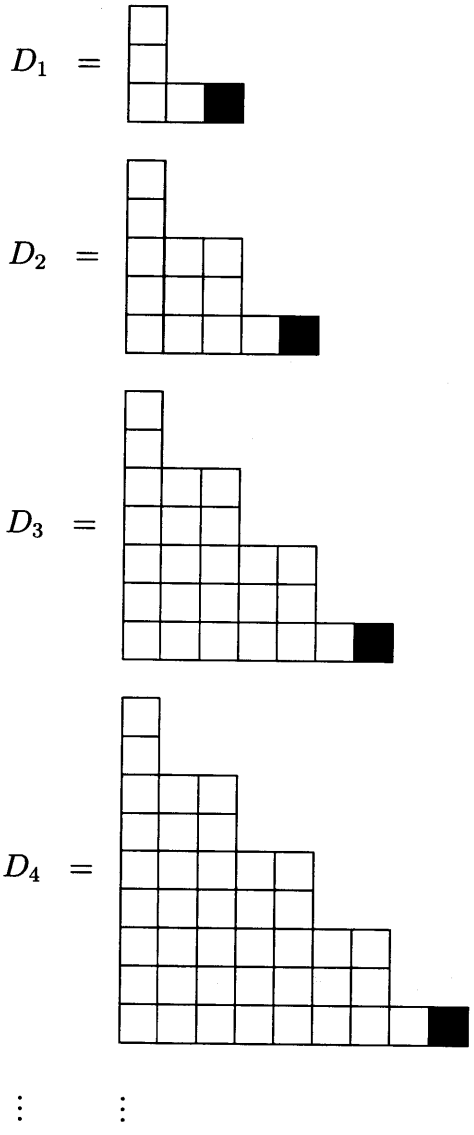


Figure 8-6: The grids D_n

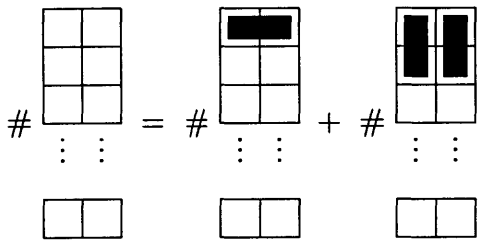
8.3 Rectangular Grids

8.3.1 $2 \times n$ grids

We begin with some simple combinatorial observations about $2 \times n$ grids, which lead to elegant proofs of Fibonacci identities.

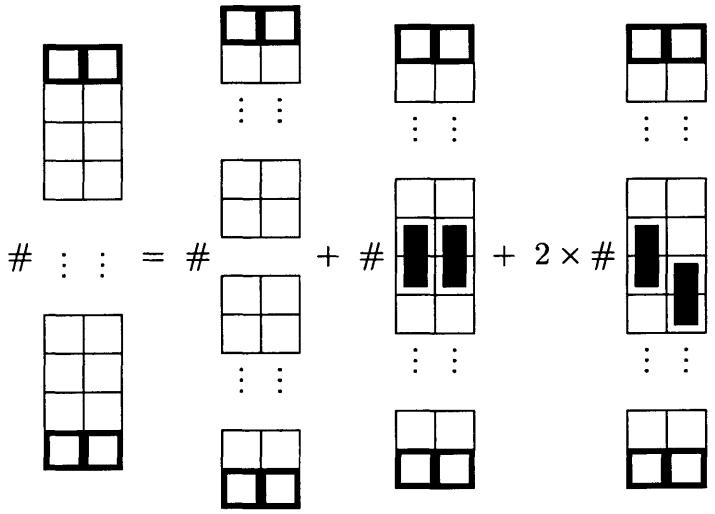
The art of discovering, or constructing problems whose solution is expressed in terms of Fibonacci numbers, has probably been in existence since the discovery of the numbers themselves. One of the many elegant interpretations of the Fibonacci numbers is in terms of domino tilings of the $2 \times n$ rectangle [59]. The correspondence is quickly established as follows:

Using the symbol # to denote the number of tilings of a grid, we have



Since the number of tilings of the 2×1 grid is 1, we have that the number of tilings of the $2 \times n$ grid is F_{n+1}.

The Lucas numbers can be recovered from a very similar problem. Consider the $2 \times n$ ring depicted below. The two bold squares on the top are identified with the two bold squares at the bottom.



(8.14)

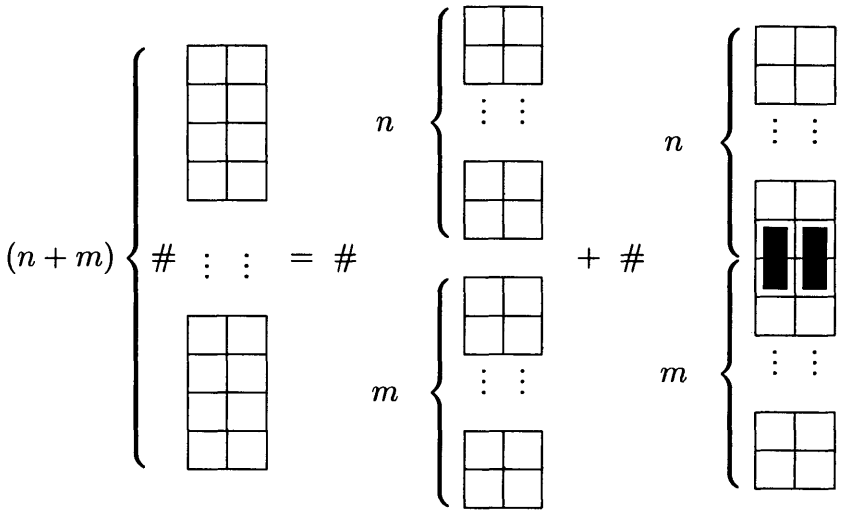
In other words, we choose a line cutting the ring, and we consider the number of tilings avoiding the line together with the number of tilings crossing the line. The third term in the expression on the right hand side has only one tiling, so we obtain that the number of tilings of the $2 \times n$ ring is

$$F_{n+1} + F_{n-1} + 2 = L_{n+1} + 2.$$

The cutting technique employed above can be used to provide an immediate "picture proof" of the identity

$$F_{n+m} = F_{m-1}F_n + F_mF_{n+1}. \tag{8.15}$$

Consider the $2 \times (n+m)$ rectangle:



We cannot have just one tile crossing a given line (see the third term in the right hand side of (8.14)) because the resulting region would not be tileable. We have thus shown that

$$F_{n+m+1} = F_{n+1}F_{m+1} + F_nF_m. \tag{8.16}$$

This is equivalent to equation (8.15) by replacing m with $m-1$.

Combinatorial proofs of the above fact have been discovered before (for a graph theoretical proof see [78]); Undoubtedly, many different bijective proofs are known, probably even the one given above. The search for bijective proofs of different identities, while interesting to some extent, is sometimes unnecessary and redundant. In this case however, the tiling proof is much clearer and shorter than the standard inductive proof. Indeed, it seems that elegant, succint tiling proofs can replace inductive proofs in many cases.

The cutting method described above is just one of many tools that can be used to extract proofs of identities. In what follows, we present a number of different examples illustrating the variety of identities one can deduce using tilings alone. Different algebraic and inductive proofs can be found in [87].

Note: The fact that the number of tilings of the $2 \times n$ rectangle is F_{n+1} means that many of the identities we derive need to be re-indexed (as in the proof of (8.15)). We omit this from our proofs.

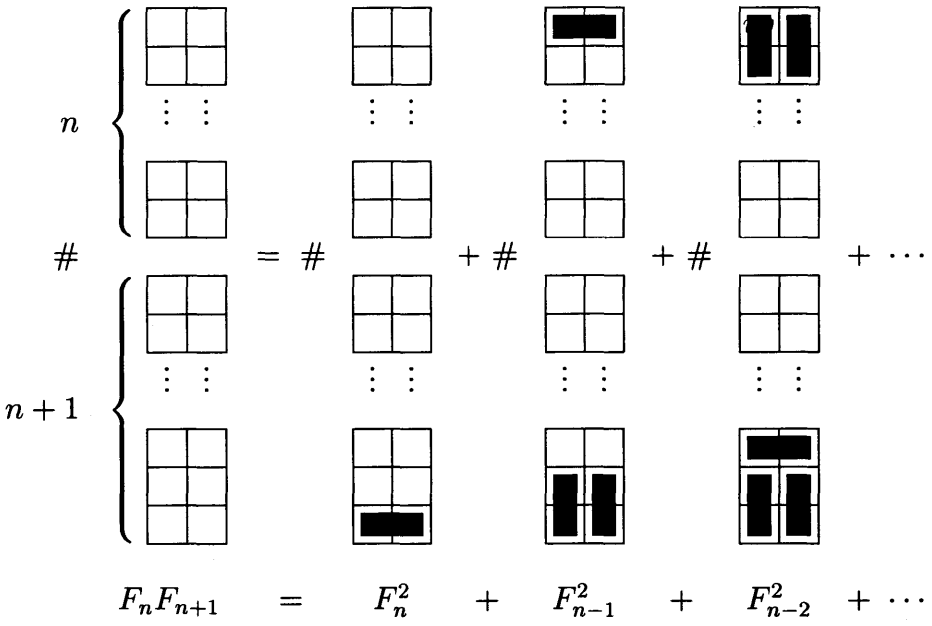
$F_n = \sum_{i=0}^{\infty} \binom{n-i-1}{i}$:

We use the fact that any horizontal line cutting the $2 \times n$ rectangle crosses two dominoes or non at all. It follows that any tiling can be specified by a sequence of H's

and V's (corresponding to horiontal and vertical dominoes) such that $\#H+2\times\#V = n$. Notice that the number of ways of selecting i positions from n positions so that no two are adjacent is exactly $\binom{n+1-i}{i}$. Since we can only place vertical dominoes on $n-1$ of the rows, and the condition is that no two are adjacent, it follows that

$$F_{n+1} = \sum_{i=0}^{\infty} \binom{n-i}{i}. \tag{8.17}$$

$\sum_{i=1}^{n} F_i^2 = F_n F_{n+1}$:



(8.18)

$F_{n+1}F_{n-1} - F_n^2 = (-1)^n$:

The proof of this identity using tilings is implicit in the bijective proof of Zeilberger [90]. The proof in his paper can, in principle, be translated into the language of tilings.

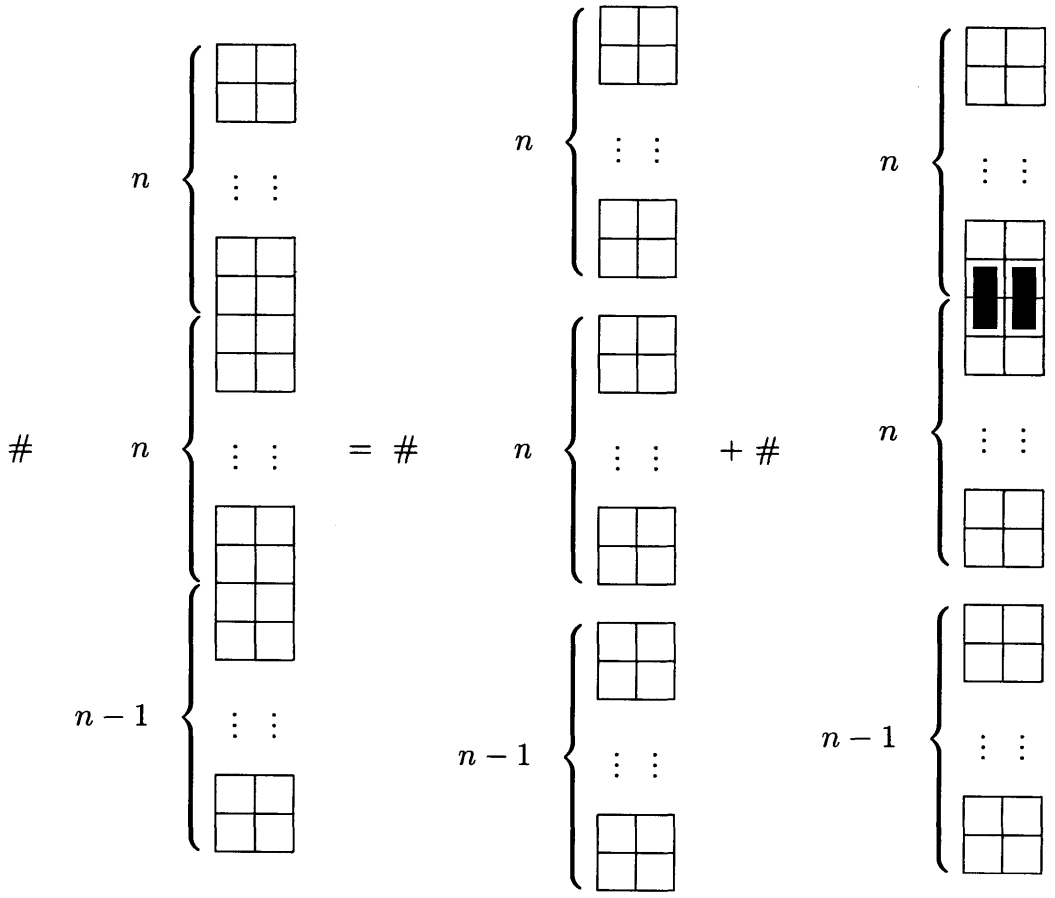
$F_{n+tp} = \sum_{i=0}^{p} \binom{p}{i} F_{n+(t-1)p-i}$:

Consider a grid of size $2 \times (m+tp)$. We begin by selecting a total of p H's and V's. Suppose that we have selected i V's. Then we have a tiling of the first $p+i$ rows of the original rectangle. we now complete the rest of the rows in $F_{n+(t-1)p-i}$ ways.

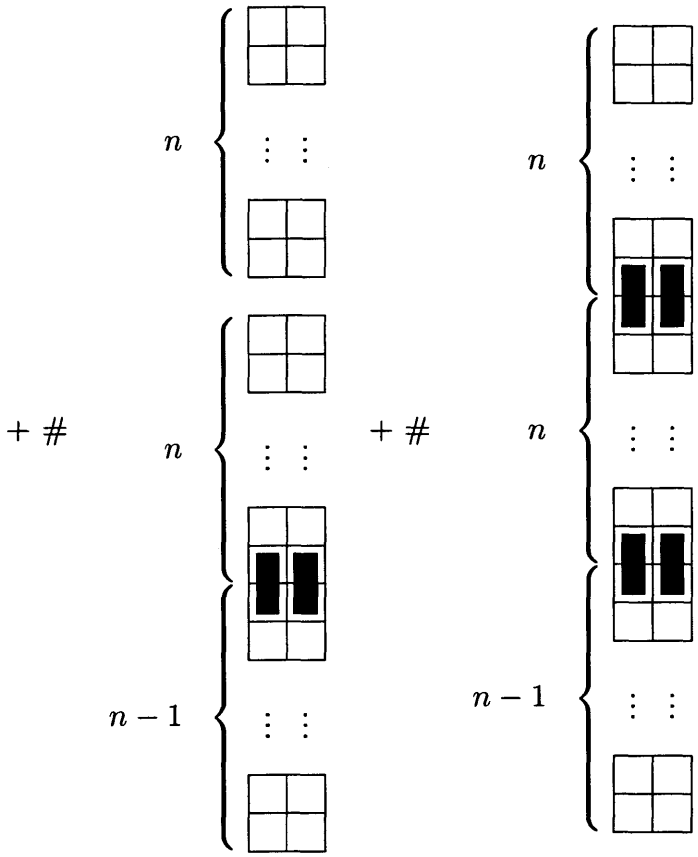
$F_{3n} = F_{n+1}^3 + F_n^3 - F_{n-1}^3$:

In this proof we cheat a bit and use some algebra, although only the basic identity (definition)

$$F_n = F_{n-1} + F_{n-2}.$$



(continued on next page)



From these diagrams it is apparent that

$$\begin{aligned}
F_{3n} &= F_{n+1}^2F_n + F_n^3 + F_{n-1}F_nF_{n+1} + F_{n-1}^2F_n && (8.19)\\
&= F_n^3 + F_{n-1}^3 - F_{n-1}^3 + F_{n+1}^2F_n + F_{n-1}F_nF_{n+1} + F_{n-1}^2F_n && (8.20)\\
&= F_n^3 + F_{n-1}^2(F_{n-1} + F_n) - F_{n-1}^3 + F_{n+1}^2F_n + F_{n-1}F_nF_{n+1} && (8.21)\\
&= F_n^3 - F_{n-1}^3 + F_{n-1}F_{n+1}(F_{n-1} + F_n) + F_{n+1}^2F_n && (8.22)\\
&= F_n^3 - F_{n-1}^3 + F_{n+1}^3 && (8.23)
\end{aligned}$$

Other Identities

Many of the above proofs easily generalize to give many more identities. For example, it is not hard to see how to modify example 2.1 to obtain the identity

$$\sum_{i=1}^{n} F_iF_{i+d} = \begin{cases} F_nF_{n+d+1} & n \text{ even} \\ F_nF_{n+d+1} - F_d & n \text{ odd} \end{cases} \qquad (8.24)$$

The variety of identities tackled above shows that in, some sense, the framework of tilings unifies many of the Fibonacci identities known to date. From a more practical standpoint, the tiling method should clearly be considered whenever trying to prove an identity about Fibonacci numbers. The proof may often be much simpler than an

inductive argument.

8.3.2 $n \times m$ grids

The exact formula for the largest power of 2 appearing in $N(2n, 2n)$ suggests an investigation of the same question for $n \times m$ rectangular grids.

We use the notation (a, b) to denote the greatest common divisor of a and b.

Problem 8.1 *Let $N(n, m)$ be the number of domino tilings of the $n \times m$ rectangular grid. Prove combinatorially that*

$$N(2n, 2m) = 2^{\frac{(2n+1,2m+1)-1}{2}}(2r_1 + 1) \tag{8.25}$$

$$N(2n+1, 2m) = 2^{\frac{(n+1,2m+1)-1}{2}(3+j)}(2r_2 + 1) \tag{8.26}$$

where j is defined by $n + 1 = 2^j(2t + 1)$. (In the above r_1, r_2, t are natural numbers that may vary for different n, m.)

Equation (8.25) follows using the methods introduced in [42]. (This has been observed by Saldanha [76]). Indeed, the other case should follow by similar methods. A combinatorial proof is not known for either case. Combinatorial proofs are important in this context because other methods fail for regions that are more complicated. Section 4 contains numerous examples where an analogous formula to (8.1) is lacking, and therefore there is no closed form formula from which to work.

Stanley [85] has conjectured that for fixed m (and n varying), $N(n, m)$ satisfies a linear recurrence with constant coefficients that is of order $2^{\frac{m+1}{2}}$ (he established this when $m + 1$ is an odd prime). Such recurrences have been obtained for small m and can be used to provide proofs of special cases of Problem 8.1. Indeed, Bao [5] has used such recurrences together with the reduction techniques we use above to establish combinatorial proofs for the formulas in Problem 8.1 for $n \leq 2$. Unfortunately, the difficulty in establishing recurrences for $N(n, m)$ combinatorially probably precludes the general applicability of the above method for finding combinatorial proofs for (8.25) and (8.26).

Equation (8.26), has recently been established algebraically by Chapman [14].

8.4 Conjectures

8.4.1 Deleting From Diagonals

We begin with an intriguing "power of 2" conjecture for a new type of region we call the **spider**.

Define the (n, k) spider to be the region obtained by deleting k consecutive squares (from the corner) along each diagonal of the $2n \times 2n$ square grid.

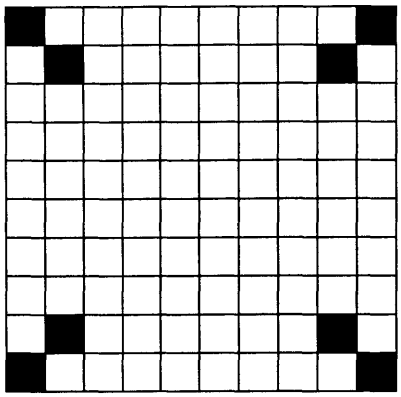


Figure 8-7: The $(5, 2)$ spider

Conjecture 8.3 *Let $S(n, k)$ be the number of domino tilings of the (n, k) spider.*

$$S(n, k) = 2^{n+k(2k-1)}(2r + 1), \quad k \leq \lfloor \frac{n}{2} \rfloor. \tag{8.27}$$

When $k > \lfloor \frac{n}{2} \rfloor$ the region reduces to an Aztec diamond after the removal of forced dominoes (for a definition and extensive discussion of Aztec diamonds see [26]). If n is even we see that (8.27) reduces to the formula for the number of domino tilings of the Aztec diamond when $k = \frac{n}{2}$. Conjecture 8.3 has been checked numerically for $n \leq 10$.

8.4.2 Deleting From Step Diagonals

The acute reader will have noticed that the arguments in Lemma 8.1 establish that any domino tiling of the $2n \times 2n$ square grid contains at least n disjoint alternating cycles. The tiling in Example 8.1 illustrates that this is the best result possible (for other results along these lines see the Chapter on forcing). Figure 8-8 shows how to place n dominoes so as to ensure the remaining figure has only one tiling (the n dominoes "block" the n cycles).

We shall call the set of the first n stepwise horizontal edges in the $2n \times 2n$ square grid the **step-diagonal.**

The above observation has led Propp [71] to ask whether removal of only half the dominoes from the bottom of the step diagonal results in a graph whose number of tilings is interesting. Indeed, drawing on his idea, we have formulated the following remarkable conjecture:

Conjecture 8.4 *Let G be the grid obtained after the removal of* **any** *k edges from the step-diagonal of the $2n \times 2n$ square grid. Then the number of domino tilings of G is of the form*

$$\#G = 2^{n-k}(2r + 1). \tag{8.28}$$

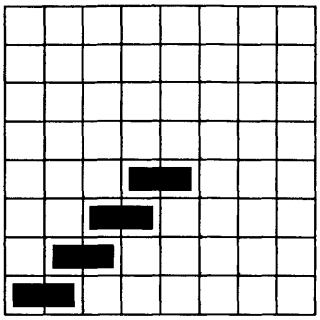


Figure 8-8: Tiles on the step-diagonal

In addition, if the k edges removed are consecutive from the lower left corner then $2r+1$ is a perfect square.

Conjecture 8.4 was checked extensively for $n \leq 10$ (the exponential growth of the number of configurations to be tested precluded exhaustive checking of this conjecture).

Also related to the step-diagonal is the following theorem:

Theorem 8.2 *Let G be the grid obtained after the removal of one edge from the step-diagonal of the $2n \times 2n$ square grid. Using the notation that $N(2n, 2n) = 2^n(2k+1)^2$, the number of domino tilings of G satisfies:*

$$\#G = c(2k+1) \tag{8.29}$$

where c is a constant which depends upon which edge was removed.

Proof: We can assume without loss of generality that the removed edge covers one of the a_i's. Furthermore, we can assume that the removed edge has direction either down or to the right. This can be arranged by flipping and/or rotating the grid. Just as in the proof of Lemma 8.2, we see that

$$\#G = 2^{n-1} \sum_C \#C \tag{8.30}$$

where C ranges over all reduced configurations (in this case, configurations for the remaining $(n-1)$ a_i's). Furthermore, from Figure 8-3, we have that $\sum_C \#C$ is $\#H_n$ (corresponding to the region filled with U), multiplied (this time) by a value *not equal* to $\#H_n$.

Edward Early has considered the number of tilings of **holey squares**. The holey square $H(n, m)$ is a $2n \times 2n$ square with a hole of size $2m \times 2m$ removed from the center. He has conjectured

Conjecture 8.5

$$\#H(n, m) = 2^{n-m}(2k+1)^2. \tag{8.31}$$

The fact that $2^{n-m}|H(n,m)$ is easily obtained using Lemma 8.1 (the fact that $H(n,m)$ is either a perfect square or twice a perfect square also follows). The fact that $n-m$ is the highest power of 2 dividing $H(n,m)$ does not follow inductively in this case. Bao [5] has established that the conjecture is true for $m=1,2$ by showing that a region similar to H_n has an odd number of domino tilings. Unfortunately, algebraic methods using (8.1) fail in this case since no analogous formulas from which to work are known.

Finally, based on numerical evidence, we present the following conjecture:

Conjecture 8.6 *Conjecture 8.4 is true for all holey squares (with n replaced by $n-m$ in (8.28)).*

Note: Theorem 8.2 is true for all holey squares (with $(2k+1)$ replaced by the square root of the odd part of $\#H(n,m)$).

8.5 Discussion

The results and conjectures of the previous sections point to an underlying combinatorial principle which is most likely the basis of the nice patterns of powers of 2. While such a result eludes us, the following old (somewhat forgotten) result which appears in [59] may hint at an algebraic approach to "power of 2" conjectures:

Proposition 8.1 *A graph G has an even number of perfect matchings iff there is a non-empty set $S \subseteq V(G)$ such that every point is adjacent to an even number of points of S.*

Henry Cohn [18] has recently proved an interesting result about the function $f(n)$ appearing in $N(2n,2n)=2^n f(n)^2$:

Theorem 8.3 (Cohn) *$f(n)$ is uniformly continous under the 2-adic metric, and satisfies the functional equation $f(-1-n)=\pm f(n)$ where the sign is positive iff $n \equiv 0,3 \bmod 4$.*

Appendix A

Biology Tables

The genetic code is the map used by ribosomes to convert codons into amino acids. The map we have included below is for eukaryotes. It includes the codons and the amino acids into which they are converted (together with the abbreviated alphabet letter). Notice that ATG (Methionine) is the standard initiation codon, and that TAG, TGA and TAA are the stop codons.

TTT	F	Phe	TCT	S	Ser	TAT	Y	Tyr	TGT	C	Cys
TTC	F	Phe	TCC	S	Ser	TAC	Y	Tyr	TGC	C	Cys
TTA	L	Leu	TCA	S	Ser	TAA	*	Ter	TGA	*	Ter
TTG	L	Leu	TCG	S	Ser	TAG	*	Ter	TGG	W	Trp
CTT	L	Leu	CCT	P	Pro	CAT	H	His	CGT	R	Arg
CTC	L	Leu	CCC	P	Pro	CAC	H	His	CGC	R	Arg
CTA	L	Leu	CCA	P	Pro	CAA	Q	Gln	CGA	R	Arg
CTG	L	Leu	CCG	P	Pro	CAG	Q	Gln	CGG	R	Arg
ATT	I	Ile	ACT	T	Thr	AAT	N	Asn	AGT	S	Ser
ATC	I	Ile	ACC	T	Thr	AAC	N	Asn	AGC	S	Ser
ATA	I	Ile	ACA	T	Thr	AAA	K	Lys	AGA	R	Arg
ATG	M	Met	ACG	T	Thr	AAG	K	Lys	AGG	R	Arg
GTT	V	Val	GCT	A	Ala	GAT	D	Asp	GGT	G	Gly
GTC	V	Val	GCC	A	Ala	GAC	D	Asp	GGC	G	Gly
GTA	V	Val	GCA	A	Ala	GAA	E	Glu	GGA	G	Gly
GTG	V	Val	GCG	A	Ala	GAG	E	Glu	GGG	G	Gly

Table A.1: The Genetic Code

The **PAM** matrix series was developed by Margaret Dayhoff et. al. [23] (for a thorough discussion about such matrices and their relatives see [31]). We used the PAM20 matrix to score exon pairs, having determined that this point mutation distance was suitable for assessing human/mouse homology. The exact matrix we used is the one shown below (obtained from the NCBI website), normalized so that

the average value on the diagonal is 2. We mention that true log odds ratios can be obtained by multiplying every entry in the matrix by $\frac{\log 2}{2}$. In Chapter 6 we describe the different possibilities for normalization, and how we selected one.

The codon usage tables are from the Transterm database [22]:

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	6	-8	-5	-4	-8	-5	-3	-3	-8	-6	-7	-8	-6	-9	-2	-1	-1	-16	-9	-3	-5	-4	-4	-19
R	-8	9	-7	-12	-9	-2	-11	-11	-3	-6	-10	-1	-5	-10	-5	-4	-8	-3	-11	-9	-9	-5	-7	-19
N	-5	-7	8	1	-13	-5	-3	-4	-1	-6	-8	-2	-11	-10	-7	-1	-3	-9	-5	-9	6	-4	-4	-19
D	-4	-12	1	8	-16	-4	2	-4	-5	-9	-15	-6	-13	-17	-9	-5	-6	-17	-13	-9	6	0	-7	-19
C	-8	-9	-13	-16	10	-16	-16	-11	-8	-7	-17	-16	-16	-15	-9	-4	-9	-18	-5	-7	-14	-16	-11	-19
Q	-5	-2	-5	-4	-16	9	0	-8	0	-9	-6	-4	-5	-15	-4	-6	-7	-15	-14	-8	-4	7	-6	-19
E	-3	-11	-3	2	-16	0	8	-5	-6	-6	-10	-5	-8	-16	-7	-5	-7	-19	-9	-8	0	6	-6	-19
G	-3	-11	-4	-4	-11	-8	-5	7	-10	-13	-12	-8	-10	-10	-7	-3	-7	-17	-16	-7	-4	-6	-6	-19
H	-8	-3	-1	-5	-8	0	-6	-10	9	-11	-7	-8	-13	-7	-5	-7	-8	-8	-4	-7	-2	-2	-6	-19
I	-6	-6	-6	-9	-7	-9	-6	-13	-11	9	-2	-7	-2	-3	-10	-8	-3	-16	-7	1	-7	-7	-6	-19
L	-7	-10	-8	-15	-17	-6	-10	-12	-7	-2	7	-9	0	-4	-8	-9	-8	-7	-8	-3	-10	-8	-7	-19
K	-8	-1	-2	-6	-16	-4	-5	-8	-8	-7	-9	7	-3	-16	-8	-5	-4	-14	-10	-10	-3	-5	-6	-19
M	-6	-5	-11	-13	-16	-5	-8	-10	-13	-2	0	-3	11	-5	-9	-6	-5	-15	-13	-2	-12	-6	-6	-19
F	-9	-10	-10	-17	-15	-15	-16	-10	-7	-3	-4	-16	-5	9	-11	-7	-10	-6	1	-9	-12	-16	-9	-19
P	-2	-5	-7	-9	-9	-4	-7	-7	-5	-10	-8	-8	-9	-11	8	-3	-5	-16	-16	-7	-8	-5	-6	-19
S	-1	-4	-1	-5	-4	-6	-5	-3	-7	-8	-9	-5	-6	-7	-3	7	0	-6	-8	-8	-2	-6	-4	-19
T	-1	-8	-3	-6	-9	-7	-7	-7	-8	-3	-8	-4	-5	-10	-5	0	7	-15	-7	-4	-4	-7	-5	-19
W	-16	-3	-9	-17	-18	-15	-19	-17	-8	-16	-7	-14	-15	-6	-16	-6	-15	13	-6	-18	-11	-17	-13	-19
Y	-9	-11	-5	-13	-5	-14	-9	-16	-4	-7	-8	-10	-13	1	-16	-8	-7	-6	10	-8	-7	-11	-9	-19
V	-3	-9	-9	-9	-7	-8	-8	-7	-7	1	-3	-10	-2	-9	-7	-8	-4	-18	-8	7	-9	-8	-6	-19
B	-5	-9	6	6	-14	-4	0	-4	-2	-7	-10	-3	-12	-12	-8	-2	-4	-11	-7	-9	6	-1	-6	-19
Z	-4	-5	-4	0	-16	7	6	-6	-2	-7	-8	-5	-6	-16	-5	-6	-7	-17	-11	-8	-1	6	-6	-19
X	-4	-7	-4	-7	-11	-6	-6	-6	-6	-6	-7	-6	-6	-9	-6	-4	-5	-13	-9	-6	-6	-6	-6	-19
*	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	-19	1

Table A.2: The PAM20 matrix.

AmAcid	Codon	Number	/1000	Fraction
Gly	GGG	67052.00	16.26	0.24
Gly	GGA	69289.00	16.80	0.25
Gly	GGT	46385.00	11.25	0.17
Gly	GGC	95488.00	23.15	0.34
Glu	GAG	167908.00	40.71	0.58
Glu	GAA	122622.00	29.73	0.42
Asp	GAT	94296.00	22.86	0.46
Asp	GAC	108702.00	26.36	0.54
Val	GTG	117559.00	28.50	0.47
Val	GTA	29241.00	7.09	0.12
Val	GTT	45257.00	10.97	0.18
Val	GTC	60686.00	14.71	0.24
Ala	GCG	29796.00	7.22	0.10
Ala	GCA	65536.00	15.89	0.23
Ala	GCT	76404.00	18.53	0.27
Ala	GCC	116551.00	28.26	0.40
Arg	AGG	45260.00	10.97	0.20
Arg	AGA	46298.00	11.23	0.20
Ser	AGT	49552.00	12.01	0.15
Ser	AGC	80527.00	19.52	0.24
Lys	AAG	137842.00	33.42	0.58
Lys	AAA	100951.00	24.48	0.42
Asn	AAT	70840.00	17.18	0.46
Asn	AAC	82995.00	20.12	0.54
Met	ATG	90548.00	21.95	1.00
Ile	ATA	28777.00	6.98	0.16
Ile	ATT	66222.00	16.06	0.36
Ile	ATC	90043.00	21.83	0.49
Thr	ACG	26394.00	6.40	0.12
Thr	ACA	61755.00	14.97	0.28
Thr	ACT	53911.00	13.07	0.24
Thr	ACC	81000.00	19.64	0.36

AmAcid	Codon	Number	/1000	Fraction
Trp	TGG	50400.00	12.22	1.00
End	TGA	3922.00	0.95	0.49
Cys	TGT	41047.00	9.95	0.45
Cys	TGC	50739.00	12.30	0.55
End	TAG	1739.00	0.42	0.22
End	TAA	2301.00	0.56	0.29
Tyr	TAT	50282.00	12.19	0.44
Tyr	TAC	65051.00	15.77	0.56
Leu	TTG	49783.00	12.07	0.13
Leu	TTA	29222.00	7.09	0.07
Phe	TTT	68820.00	16.69	0.46
Phe	TTC	82290.00	19.95	0.54
Ser	TCG	18711.00	4.54	0.06
Ser	TCA	48119.00	11.67	0.15
Ser	TCT	60222.00	14.60	0.18
Ser	TCC	71945.00	17.44	0.22
Arg	CGG	47772.00	11.58	0.21
Arg	CGA	25689.00	6.23	0.11
Arg	CGT	19735.00	4.79	0.09
Arg	CGC	44225.00	10.72	0.19
Gln	CAG	142629.00	34.58	0.74
Gln	CAA	49865.00	12.09	0.26
His	CAT	42331.00	10.26	0.41
His	CAC	61169.00	14.83	0.59
Leu	CTG	159724.00	38.73	0.40
Leu	CTA	27535.00	6.68	0.07
Leu	CTT	51426.00	12.47	0.13
Leu	CTC	76817.00	18.63	0.19
Pro	CCG	29150.00	7.07	0.11
Pro	CCA	70590.00	17.12	0.28
Pro	CCT	72473.00	17.57	0.28
Pro	CCC	82909.00	20.10	0.32

Table A.3: Codon Usage in Humans

AmAcid	Codon	Number	/1000	Fraction	AmAcid	Codon	Number	/1000	Fraction
Gly	GGG	35492.00	15.80	0.23	Trp	TGG	28319.00	12.61	1.00
Gly	GGA	39668.00	17.66	0.26	End	TGA	2307.00	1.03	0.49
Gly	GGT	27188.00	12.10	0.18	Cys	TGT	24262.00	10.80	0.46
Gly	GGC	52529.00	23.38	0.34	Cys	TGC	28288.00	12.59	0.54
Glu	GAG	91181.00	40.59	0.59	End	TAG	1076.00	0.48	0.23
Glu	GAA	62167.00	27.67	0.41	End	TAA	1282.00	0.57	0.27
Asp	GAT	49356.00	21.97	0.44	Tyr	TAT	27170.00	12.09	0.42
Asp	GAC	62116.00	27.65	0.56	Tyr	TAC	38296.00	17.05	0.58
Val	GTG	64700.00	28.80	0.47	Leu	TTG	27575.00	12.27	0.13
Val	GTA	15715.00	7.00	0.11	Leu	TTA	13397.00	5.96	0.06
Val	GTT	22840.00	10.17	0.17	Phe	TTT	36046.00	16.04	0.43
Val	GTC	34878.00	15.52	0.25	Phe	TTC	47781.00	21.27	0.57
Ala	GCG	15804.00	7.03	0.10	Ser	TCG	10127.00	4.51	0.06
Ala	GCA	34247.00	15.24	0.22	Ser	TCA	24811.00	11.04	0.14
Ala	GCT	44581.00	19.84	0.29	Ser	TCT	34282.00	15.26	0.19
Ala	GCC	59633.00	26.54	0.39	Ser	TCC	39640.00	17.64	0.22
Arg	AGG	25775.00	11.47	0.21	Arg	CGG	23262.00	10.35	0.19
Arg	AGA	25602.00	11.40	0.21	Arg	CGA	14906.00	6.63	0.12
Ser	AGT	27061.00	12.05	0.15	Arg	CGT	10958.00	4.88	0.09
Ser	AGC	44351.00	19.74	0.25	Arg	CGC	22672.00	10.09	0.18
Lys	AAG	78551.00	34.96	0.61	Gln	CAG	77131.00	34.33	0.74
Lys	AAA	49491.00	22.03	0.39	Gln	CAA	26417.00	11.76	0.26
Asn	AAT	35666.00	15.88	0.42	His	CAT	22105.00	9.84	0.39
Asn	AAC	48946.00	21.79	0.58	His	CAC	34030.00	15.15	0.61
Met	ATG	50169.00	22.33	1.00	Leu	CTG	85763.00	38.17	0.40
Ile	ATA	15025.00	6.69	0.15	Leu	CTA	16396.00	7.30	0.08
Ile	ATT	33369.00	14.85	0.33	Leu	CTT	27064.00	12.05	0.13
Ile	ATC	51779.00	23.05	0.52	Leu	CTC	42912.00	19.10	0.20
Thr	ACG	13636.00	6.07	0.11	Pro	CCG	15291.00	6.81	0.11
Thr	ACA	34913.00	15.54	0.29	Pro	CCA	38695.00	17.22	0.28
Thr	ACT	29650.00	13.20	0.24	Pro	CCT	41530.00	18.49	0.30
Thr	ACC	44121.00	19.64	0.36	Pro	CCC	42600.00	18.96	0.31

Table A.4: Codon Usage in Mice

Appendix B

Datasets

B.1 Description of the Databases

The data available to us comes from a number of data bases, which contain examples of various kinds of biological sequences, as follows:

- The protein database `OWL` [96]; it contains proteins whose amino acid sequences have been determined.
- The cDNA database `dbEST` [95]; it consists of what are essentially the DNA sequences of the exons of a gene only, and fragments thereof.
- Databases of genes whose splicings into introns and exons are known (`GENBANK`).
- Databases of genes of various species without such information (`GENBANK`).

There are numerous complications that arise when using publicly available data, perhaps the most significant of which is the unreliable nature of the annotated data. Errors frequently occur in the data itself, while sometimes errors appear in the annotation of the exons and introns. The errors are difficult to find, because sometimes what may look like an apparent error (for example a GC instead of a GT in a splice site), is really a valid annotation, albeit rare.

B.1.1 Learning

We used a number of different databases for our various training experiments, depending on whether or not the data was to be used again for testing.

In Chapter 3 we describe χ^2 tests for which we used the **HKR** (Haussler-Kulp-Reese) dataset. This database is distributed freely by the creators of the `GENIE` program [34] and consisted of 353 genes selected from `GENBANK`. The criteria for selection included heterogeneity among the genes, reliable intron/exon boundary annotations *etc.* The dataset has since been expanded to include over 400 genes, and this dataset was used to train the splice site detector used in Chapter 6.

B.1.2 Testing

Tests for the dictionary approach in Chapter 5 were performed on a test set derived from the HKR benchmark dataset. The Haussler–Kulp–Reese test set was filtered with the additional following criteria (beyond what the authors had already checked for): Genes were required to have a *single* annotation covering the whole sequence, the CDS annotation was checked for consistency with the annotated nucleotides, and sequences with "unknown basepairs" were removed. This resulted in a reduced set of genes, described in Appendix B, which we call the **HKRM** dataset.

Additional tests were conducted on a data set of 570 vertebrate genes, compiled by Burset and Guigó [13]. This is a standard test set that has been used as a benchmark for comparing gene recognition programs. We refer to the dataset as **BG**.

The human-mouse comparison test sets (HUMCOMP/MUSCOMP)

This dataset consisted of 119 human genes (homo sapiens) together with the corresponding genes in the mouse (Mus musculus). The dataset was obtained by starting with a list of 1196 orthologous mouse and human cDNAs as described by Makolowski *et al.* in [62]. For each cDNA, we selected genomic matches from `GENBANK` that contained the entire cDNA (this was done rapidly using the dictionary method described in Chapter 5). We screened the human matches by hand for genes with complete intron exon annotations. This pruning and filtering procedure resulted in 119 genes, for which we had the necessary annotations in the human, and for which we could find the corresponding murine genomic DNA. In this thesis, we refer to the datasets as HUMCOMP and MUSCOMP. The genes together with information about their coding exons are listed in Table B.1.

B.2 Tables

The following pages contain the various datasets described above. Locus names have been provided. To find the genes in `GENBANK`, search in
`http://www.ncbi.nlm.nih.gov/Entrez/nucleotide.html`

We mention that Table B.1 begins on the next page and is forty pages long. The annotations for the genes numbered 56,71,72 and 80 have been truncated for formatting reasons; in these genes only the lengths of the first 13 exons are displayed. Some of the annotations in the mouse sequences (specifically exon lengths and number of exons) may be inaccurate because of lack of annotations in the mouse database.

Gene number: 1
Human locus: HSCKIIBE 5917 bp DNA PRI 10-SEP-1998
DEFINITION Human gene for casein kinase II subunit beta (EC 2.7.1.37).
ACCESSION X57152
–
Mouse locus: MMGMCK2B 7874 bp DNA ROD 12-APR-1996
DEFINITION M.musculus gMCK2-beta gene.
ACCESSION X80685
–
Exon Lengths:

H:	72	103	116	76	190	91
M:	72	103	116	76	190	91

	Human	Mouse
Number of coding exons	6	6
Total coding length	648	648

Gene number: 2
Human locus: HUMSAACT 3778 bp DNA PRI 09-JAN-1995
DEFINITION Human skeletal alpha-actin gene, complete cds.
ACCESSION M20543
–
Mouse locus: MUSACASA 4007 bp DNA ROD 10-OCT-1991
DEFINITION Mouse skeletal alpha-actin gene, complete cds.
ACCESSION M12347
–
Exon Lengths:

H:	129	325	162	192	182	144
M:	129	325	162	192	182	144

	Human	Mouse
Number of coding exons	6	6
Total coding length	1134	1134

Gene number: 3
Human locus: HSH4EHIS 859 bp DNA PRI 09-NOV-1992
DEFINITION H.sapiens H4/e gene for H4 histone.
ACCESSION X60484
–
Mouse locus: MMHIS412 637 bp DNA ROD 08-DEC-1995
DEFINITION Mouse histone H4 gene (clone 12).
ACCESSION X13235
–
Exon Lengths:

H:	312
M:	312

	Human	Mouse
Number of coding exons	1	1
Total coding length	312	312

Gene number: 4
Human locus: HSU12202 4942 bp DNA PRI 13-SEP-1996
DEFINITION Human ribosomal protein S24 (rps24) gene, complete cds.
ACCESSION U12202
–
Mouse locus: MMMRPS24 5499 bp DNA ROD 15-MAR-1994
DEFINITION M.musculus MRP S24 gene.
ACCESSION X71972
–
Exon Lengths:

H:	3	66	210	111	3
M:	3	66	210	111	6

	Human	Mouse
Number of coding exons	5	5
Total coding length	393	396

Gene number: 5
Human locus: HUMHIS4 1098 bp DNA PRI 08-NOV-1994
DEFINITION Human histone H4 gene, complete cds, clone FO108.
ACCESSION M16707
–
Mouse locus: MUSHIST4 968 bp DNA ROD 26-MAR-1994
DEFINITION Mouse histone H4 gene, complete cds.
ACCESSION J00422 V00753
–
Exon Lengths:

H:	312
M:	312

	Human	Mouse
Number of coding exons	1	1
Total coding length	312	312

Gene number: 6
Human locus: HSHISH3 698 bp DNA PRI 12-SEP-1993
DEFINITION Human histone H3 gene.
ACCESSION X00090
–
Mouse locus: MMHIST31 592 bp DNA ROD 12-SEP-1993
DEFINITION Murine H3.1 gene for histone H3.1.
ACCESSION X16496
–
Exon Lengths:

H:	411
M:	411

	Human	Mouse
Number of coding exons	1	1
Total coding length	411	411

Gene number: 7
Human locus: HSHSC70 5408 bp DNA PRI 09-MAY-1995
DEFINITION Human hsc70 gene for 71 kd heat shock cognate protein.
ACCESSION Y00371
–
Mouse locus: MMU73744 4270 bp DNA ROD 07-NOV-1996
DEFINITION Mus musculus heat shock 70 protein (Hsc70) gene, complete cds.
ACCESSION U73744
–
Exon Lengths:

H:	205	206	153	556	203	199	233	186
M:	205	206	153	556	203	199	233	186

	Human	Mouse
Number of coding exons	8	8
Total coding length	1941	1941

Gene number: 8
Human locus: HUMNOCT 4878 bp DNA PRI 19-JAN-1996
DEFINITION Homo sapiens POU-domain transcription factor (N-Oct-3), complete cds.
–
Mouse locus: MUSPOUDOMB 3864 bp DNA ROD 18-MAY-1992
DEFINITION Mouse brain-2 POU-domain protein, complete cds.
ACCESSION M88300
–
Exon Lengths:

H:	1332
M:	1338

	Human	Mouse
Number of coding exons	1	1
Total coding length	1332	1338

Gene number: 9
Human locus: HUMTROC 4567 bp DNA PRI 11-JAN-1991
DEFINITION Human slow twitch skeletal muscle/cardiac muscle troponin C gene, complete cds.
–
Mouse locus: MUSCTNC 4194 bp DNA ROD 15-MAR-1990
DEFINITION M.musculus slow/cardiac troponin C (cTnC) gene, complete cds.
ACCESSION J04971
–
Exon Lengths:

H:	24	31	147	115	137	32
M:	24	31	147	115	137	32

	Human	Mouse
Number of coding exons	6	6
Total coding length	486	486

Gene number: 10
Human locus: HSINT1G 4522 bp DNA PRI 03-JAN-1991
DEFINITION Human int-1 mammary oncogene.
ACCESSION X03072
–
Mouse locus: MUSINT1A 5607 bp DNA ROD 21-DEC-1990
DEFINITION Mouse mammary proto-oncogene Wnt-1 (int-1), complete cds.
ACCESSION K02593 M34750
–
Exon Lengths:
H: 104 254 266 489
M: 104 254 266 489

	Human	Mouse
Number of coding exons	4	4
Total coding length	1113	1113

Gene number: 11
Human locus: HUMSRI1A 1634 bp DNA PRI 29-DEC-1994
DEFINITION Human somatostatin receptor isoform 1 gene, complete cds.
ACCESSION M81829
–
Mouse locus: MUSSRI1A 1265 bp DNA ROD 29-DEC-1994
DEFINITION Mus musculus somatostatin receptor isoform 1 gene, complete cds.
ACCESSION M81831
–
Exon Lengths:
H: 1176
M: 1176

	Human	Mouse
Number of coding exons	1	1
Total coding length	1176	1176

Gene number: 12
Human locus: HSMIMAR 2100 bp DNA PRI 01-OCT-1996
DEFINITION H. sapiens M1 gene for muscarinic acetylcholine receptor.
ACCESSION Y00508 M35128
–
Mouse locus: MUSACHRM1 1574 bp DNA ROD 29-MAR-1994
DEFINITION Mouse muscarinic acetylcholine receptor M1 gene, complete cds, clone M1/ZEM228.
–
Exon Lengths:
H: 1383
M: 1383

	Human	Mouse
Number of coding exons	1	1
Total coding length	1383	1383

Gene number: 13
Human locus: HSFAU1 2016 bp DNA PRI 21-JUL-1993
DEFINITION H.sapiens fau 1 gene.
ACCESSION X65921 S45242
–
Mouse locus: MUSFAUA 2850 bp DNA ROD 07-MAR-1996
DEFINITION Mus musculus Fau gene, complete cds.
ACCESSION L33715
–
Exon Lengths:

H:	75	145	56	126
M:	75	145	56	126

	Human	Mouse
Number of coding exons	4	4
Total coding length	402	402

Gene number: 14
Human locus: HUMCRYABA 4206 bp DNA PRI 01-NOV-1994
DEFINITION Human alpha-B-crystallin gene, 5' end.
ACCESSION M28638
–
Mouse locus: MUSALPBCRY 4181 bp DNA ROD 22-MAY-1995
DEFINITION Mouse alpha-B2-crystallin gene, complete cds.
ACCESSION M73741
–
Exon Lengths:

H:	201	123	204
M:	201	123	204

	Human	Mouse
Number of coding exons	3	3
Total coding length	528	528

Gene number: 15
Human locus: HSENO3 7194 bp DNA PRI 25-JUN-1997
DEFINITION H.sapiens ENO3 gene for muscle specific enolase.
ACCESSION X56832
–
Mouse locus: MMENO3G 5472 bp DNA ROD 09-OCT-1991
DEFINITION M.musculus gene for beta-enolase.
ACCESSION X61600
–
Exon Lengths:

H:	85	96	59	70	134	223	198	202	109	59	70
M:	85	96	59	70	134	223	198	202	109	59	70

	Human	Mouse
Number of coding exons	11	11
Total coding length	1305	1305

Gene number: 16
Human locus: HUMPPIB 1083 bp DNA PRI 08-JAN-1995
DEFINITION Homo sapiens 21 kDa protein gene, complete cds, clone D4S234.
ACCESSION M98529
–
Mouse locus: MUSPPIA 576 bp DNA ROD 13-AUG-1992
DEFINITION Mouse 19 kDa protein gene, complete cds, clone D4S234E.
ACCESSION M98530
–
Exon Lengths:
H: 558
M: 558

	Human	Mouse
Number of coding exons	1	1
Total coding length	558	558

Gene number: 17
Human locus: HUMKCHN 2397 bp DNA PRI 03-DEC-1997
DEFINITION Homo sapiens voltage-gated potassium channel (HGK5) gene, complete cds.
–
Mouse locus: MUSMK3A 1994 bp DNA ROD 15-NOV-1992
DEFINITION Mouse intronless potassium channel gene MK3.
ACCESSION M30441
–
Exon Lengths:
H: 1572
M: 1587

	Human	Mouse
Number of coding exons	1	1
Total coding length	1572	1587

Gene number: 18
Human locus: HUMPCNA 6340 bp DNA PRI 07-JAN-1995
DEFINITION Human proliferating cell nuclear antigen (PCNA) gene, complete cds.
ACCESSION J04718
–
Mouse locus: MMPCNAG 4970 bp DNA ROD 07-MAY-1991
DEFINITION Murine PCNA gene for proliferating cell nuclear antigen (DNA polymerase delta auxiliary protein).
–
Exon Lengths:
H: 221 98 68 195 124 80
M: 221 98 68 195 124 80

	Human	Mouse
Number of coding exons	6	6
Total coding length	786	786

Gene number: 19
Human locus: HSU73304 5665 bp DNA PRI 05-NOV-1996
DEFINITION Human CB1 cannabinoid receptor (CNR1) gene, complete cds.
ACCESSION U73304
–
Mouse locus: MMU22948 1654 bp DNA ROD 28-MAR-1995
DEFINITION Mus musculus CB1 cannabinoid receptor gene, complete cds.
ACCESSION U22948
–
Exon Lengths:
H: 1419
M: 1422

	Human	Mouse
Number of coding exons	1	1
Total coding length	1419	1422

Gene number: 20
Human locus: AF007876 7894 bp DNA PRI 07-APR-1998
DEFINITION Homo sapiens Na,K-ATPase beta 2 subunit gene, complete cds.
ACCESSION AF007876
–
Mouse locus: MMATPB2 7179 bp DNA ROD 20-MAY-1992
DEFINITION Mouse Na/K-ATPase beta 2 subunit gene.
ACCESSION X56007
–
Exon Lengths:
H: 112 129 105 206 57 99 165
M: 112 129 105 206 57 99 165

	Human	Mouse
Number of coding exons	7	7
Total coding length	873	873

Gene number: 21
Human locus: HUMNT3A 1029 bp DNA PRI 07-MAR-1995
DEFINITION Human neurotrophin-3 gene, complete cds, from 1.8 kb HindIII fragment.
–
Mouse locus: MMNT3 1284 bp DNA ROD 30-NOV-1992
DEFINITION M.musculus NT-3 gene for neurotrophin-3.
ACCESSION X53257
–
Exon Lengths:
H: 774
M: 777

	Human	Mouse
Number of coding exons	1	1
Total coding length	774	777

Gene number: 22
Human locus: HSCKBG 4200 bp DNA PRI 24-APR-1993
DEFINITION Human gene for creatine kinase B (EC 2.7.3.2).
ACCESSION X15334
–
Mouse locus: MUSCRKNB 4521 bp DNA ROD 24-JUN-1993
DEFINITION Mouse creatine kinase B gene, complete cds.
ACCESSION M74149
–
Exon Lengths:

H:	193	155	133	172	124	190	179
M:	193	155	133	172	124	190	179

	Human	Mouse
Number of coding exons	7	7
Total coding length	1146	1146

Gene number: 23
Human locus: HUMACHRM4 2595 bp DNA PRI 30-OCT-1994
DEFINITION Human m4 muscarinic acetylcholine receptor gene.
ACCESSION M16405
–
Mouse locus: MMM4ACHR 1707 bp DNA ROD 19-JUL-1993
DEFINITION M.musculus m4 muscarinic acetylcholine receptor.
ACCESSION X63473
–
Exon Lengths:

H:	1437
M:	1440

	Human	Mouse
Number of coding exons	1	1
Total coding length	1437	1440

Gene number: 24
Human locus: HUMMHHSP2 2876 bp DNA PRI 07-MAR-1995
DEFINITION Human MHC class III HSP70-2 gene (HLA), complete cds.
ACCESSION M59830 M34269
–
Mouse locus: MUSHSP7A2 3518 bp DNA ROD 26-MAR-1994
DEFINITION Mouse heat shock protein 70.1 (hsp70.1) gene, complete cds.
ACCESSION M35021
–
Exon Lengths:

H:	1926
M:	1929

	Human	Mouse
Number of coding exons	1	1
Total coding length	1926	1929

Gene number: 25
Human locus: HUMAPEXN 3730 bp DNA PRI 18-JAN-1995
DEFINITION Human APX gene encoding APEX nuclease, complete cds.
ACCESSION D13370
–
Mouse locus: MUSAPEX 4042 bp DNA ROD 08-FEB-1996
DEFINITION Mouse gene for APEX nuclease, complete cds.
ACCESSION D38077
–
Exon Lengths:
H: 58 188 193 518
M: 55 188 193 518

	Human	Mouse
Number of coding exons	4	4
Total coding length	957	954

Gene number: 26
Human locus: HUMGAD45A 5378 bp DNA PRI 25-JAN-1994
DEFINITION Human gadd45 gene, complete cds.
ACCESSION L24498
–
Mouse locus: MUSGAD45 3100 bp DNA ROD 23-JUL-1998
DEFINITION Mus musculus GADD45 protein (gadd45) gene, complete cds.
ACCESSION U00937
–
Exon Lengths:
H: 44 102 238 114
M: 44 102 238 114

	Human	Mouse
Number of coding exons	4	4
Total coding length	498	498

Gene number: 27
Human locus: HUMMHHSPHO 3330 bp DNA PRI 07-MAR-1995
DEFINITION Human MHC class III HSP70-HOM gene (HLA), complete cds.
ACCESSION M59829 M34268
–
Mouse locus: MUSHSC70T 2295 bp DNA ROD 09-JAN-1995
DEFINITION Mouse heat shock protein 70 (Hsc70t) gene, complete cds.
ACCESSION L27086
–
Exon Lengths:
H: 1926
M: 1926

	Human	Mouse
Number of coding exons	1	1
Total coding length	1926	1926

Gene number: 28
Human locus: HSHOX51 6305 bp DNA PRI 25-JUN-1997
DEFINITION Human HOX 5.1 gene for HOX 5.1 protein.
ACCESSION X17360
–
Mouse locus: MMU77364 7627 bp DNA ROD 23-JAN-1997
DEFINITION Mus musculus homeodomain-containing transcription factor (Hoxd4) gene, complete cds.
–
Exon Lengths:
H: 433 335
M: 427 326

	Human	Mouse
Number of coding exons	2	2
Total coding length	768	753

Gene number: 29
Human locus: HUMHISAC 1978 bp DNA PRI 07-MAR-1995
DEFINITION Human histone H1 (H1F4) gene, complete cds.
ACCESSION M60748
–
Mouse locus: MUSH1EH2B 3605 bp DNA ROD 04-AUG-1994
DEFINITION Mouse histone H1e gene, complete cds and histone H2b pseudogene.
ACCESSION L26163
–
Exon Lengths:
H: 660
M: 660 327

	Human	Mouse
Number of coding exons	1	2
Total coding length	660	987

Gene number: 30
Human locus: HUMSPERSYN 7623 bp DNA PRI 13-JAN-1995
DEFINITION Human spermidine synthase gene, complete cds.
ACCESSION M64231
–
Mouse locus: MMSPERSYN 3915 bp DNA ROD 07-DEC-1995
DEFINITION M.musculus spermidine synthase gene.
ACCESSION Z67748
–
Exon Lengths:
H: 167 121 93 154 84 146 123 21
M: 167 121 93 154 230 123 21

	Human	Mouse
Number of coding exons	8	7
Total coding length	909	909

Gene number: 31
Human locus: HSHIS10G 2530 bp DNA PRI 12-SEP-1993
DEFINITION Human gene for histone H1(0).
ACCESSION X03473
–
Mouse locus: MMU18295 2893 bp DNA ROD 15-JUL-1995
DEFINITION Mus musculus histone H1(0) gene, complete cds.
ACCESSION U18295
–
Exon Lengths:
H: 585
M: 585

	Human	Mouse
Number of coding exons	1	1
Total coding length	585	585

Gene number: 32
Human locus: HSCFOS 3565 bp DNA PRI 21-NOV-1994
DEFINITION Human cellular oncogene c-fos (complete sequence).
ACCESSION V01512
–
Mouse locus: MMCFOS 3967 bp DNA ROD 01-OCT-1996
DEFINITION Mouse c-fos oncogene.
ACCESSION V00727
–
Exon Lengths:
H: 141 252 108 642
M: 141 252 108 642

	Human	Mouse
Number of coding exons	4	4
Total coding length	1143	1143

Gene number: 33
Human locus: HUMHGCR 2635 bp DNA PRI 17-SEP-1992
DEFINITION Human gene for serotonin 1B receptor, complete cds.
ACCESSION D10995
–
Mouse locus: MUS5HT1B 2348 bp DNA ROD 11-DEC-1995
DEFINITION Mus musculus 5HT1B gene, complete cds.
ACCESSION M85151
–
Exon Lengths:
H: 1173
M: 1161

	Human	Mouse
Number of coding exons	1	1
Total coding length	1173	1161

Gene number: 34
Human locus: HUMUDPCNA 4705 bp DNA PRI 19-SEP-1995
DEFINITION Human alpha-1,3-mannosyl-glycoprotein beta-1,
2-N-acetylglucosaminyltransferase (MGAT) gene, complete cds.
–
Mouse locus: MUSGLCNACT 1894 bp DNA ROD 08-DEC-1992
DEFINITION Mouse N-acetylglucosaminyltransferase I (GlcNAc-T1) gene, complete cds.
–
Exon Lengths:
H: 1338
M: 1344

	Human	Mouse
Number of coding exons	1	1
Total coding length	1338	1344

Gene number: 35
Human locus: HSODCG 9043 bp DNA PRI 24-APR-1993
DEFINITION Human gene for ornithine decarboxylase ODC (EC 4.1.1.17).
ACCESSION X16277
–
Mouse locus: MUSODCC 7100 bp DNA ROD 15-DEC-1988
DEFINITION Mouse ornithine decarboxylase gene, complete cds.
ACCESSION J03733
–
Exon Lengths:
H: 102 174 173 135 82 84 163 113 215 145
M: 102 174 173 135 82 84 163 113 215 145

	Human	Mouse
Number of coding exons	10	10
Total coding length	1386	1386

Gene number: 36
Human locus: HUMGALTB 4286 bp DNA PRI 14-AUG-1995
DEFINITION Homo sapiens galactose-1-phosphate uridyl transferase (GALT) gene, complete cds.
–
Mouse locus: MMU41282 4023 bp DNA ROD 19-DEC-1995
DEFINITION Mus musculus galactose-1-phosphate uridyltransferase (GALT) gene, complete cds.
–
Exon Lengths:
H: 82 170 76 49 130 57 123 133 84 155 81
M: 25 170 76 49 130 57 123 133 84 155 81

	Human	Mouse
Number of coding exons	11	11
Total coding length	1140	1083

Gene number: 37
Human locus: HSU29185 35522 bp DNA PRI 19-FEB-1998
DEFINITION Homo sapiens prion protein (PrP) gene, complete cds.
ACCESSION U29185
–
Mouse locus: MUSPRNPA 38418 bp DNA ROD 19-FEB-1998
DEFINITION Mus musculus short incubation prion protein Prnpa gene, complete cds.
–
Exon Lengths:
H: 738
M: 765

	Human	Mouse
Number of coding exons	1	1
Total coding length	738	765

Gene number: 38
Human locus: HSU01212 3718 bp DNA PRI 03-AUG-1994
DEFINITION Human olfactory marker protein (OMP) gene, complete cds.
ACCESSION U01212
–
Mouse locus: MMU01213 3279 bp DNA ROD 03-AUG-1994
DEFINITION Mus musculus 129 olfactory marker protein (OMP) gene, complete cds.
ACCESSION U01213
–
Exon Lengths:
H: 492
M: 492

	Human	Mouse
Number of coding exons	1	1
Total coding length	492	492

Gene number: 39
Human locus: HUMMIF 2167 bp DNA PRI 29-SEP-1994
DEFINITION Homo sapiens macrophage migration inhibitory factor (MIF) gene, complete cds.
–
Mouse locus: MMU20156 920 bp DNA ROD 08-MAR-1996
DEFINITION Mus musculus macrophage migration inhibitory factor (MIF) gene, complete cds.
–
Exon Lengths:
H: 108 173 67
M: 108 173 67

	Human	Mouse
Number of coding exons	3	3
Total coding length	348	348

Gene number: 40
Human locus: AF049259 5698 bp DNA PRI 16-SEP-1998
DEFINITION Homo sapiens keratin 13 gene, complete cds.
ACCESSION AF049259
–
Mouse locus: MMU13921 4678 bp DNA ROD 24-JAN-1995
DEFINITION Mus musculus cytokeratin 13 (MK13) gene, complete cds.
ACCESSION U13921
–
Exon Lengths:

H:	495	83	157	162	126	221	19	
M:	471	83	157	162	126	221	23	71

	Human	Mouse
Number of coding exons	7	8
Total coding length	1263	1314

Gene number: 41
Human locus: HSH12 1391 bp DNA PRI 09-NOV-1992
DEFINITION H.sapiens H1.2 gene for histone H1.
ACCESSION X57129
–
Mouse locus: MUSHIS1A 1877 bp DNA ROD 26-MAR-1994
DEFINITION Mouse histone (H1-.1) gene, complete cds.
ACCESSION M25365
–
Exon Lengths:

H:	642
M:	639

	Human	Mouse
Number of coding exons	1	1
Total coding length	642	639

Gene number: 42
Human locus: HSACTHR 1850 bp DNA PRI 18-SEP-1992
DEFINITION H.sapiens ACTH-R gene for adrenocorticotropic hormone receptor.
ACCESSION X65633
–
Mouse locus: MUSACTHR 1100 bp DNA ROD 24-JAN-1996
DEFINITION Mouse gene for adrenocorticotropin receptor, complete cds.
ACCESSION D31952
–
Exon Lengths:

H:	894
M:	891

	Human	Mouse
Number of coding exons	1	1
Total coding length	894	891

Gene number: 43
Human locus: AF027148 12825 bp DNA PRI 08-AUG-1998
DEFINITION Homo sapiens myogenic determining factor 3 (MYOD1) gene, complete cds.
–
Mouse locus: MMMYOD1 2627 bp DNA ROD 27-JAN-1992
DEFINITION M.musculus myoD1 gene for MyoD1 protein.
ACCESSION X61655
–
Exon Lengths:
H: 630 79 254
M: 627 79 251

	Human	Mouse
Number of coding exons	3	3
Total coding length	963	957

Gene number: 44
Human locus: HUMKER18 6520 bp DNA PRI 09-JAN-1995
DEFINITION Human keratin 18 (K18) gene, complete cds.
ACCESSION M24842 M19353 X12799
–
Mouse locus: MUSENDOBA 7879 bp DNA ROD 26-AUG-1994
DEFINITION Mus musculus cytokeratin (endoB) gene, complete cds.
ACCESSION M22832
–
Exon Lengths:
H: 417 83 157 165 126 224 121
M: 396 83 157 165 126 224 121

	Human	Mouse
Number of coding exons	7	7
Total coding length	1293	1272

Gene number: 45
Human locus: HUMADRA 1521 bp DNA PRI 30-OCT-1994
DEFINITION Human platelet alpha-2-adrenergic receptor gene, complete cds.
ACCESSION M18415
–
Mouse locus: MUSALP2ADB 1454 bp DNA ROD 20-AUG-1992
DEFINITION Mouse alpha-2 adrenergic receptor, complete cds.
ACCESSION M99377
–
Exon Lengths:
H: 1353
M: 1353

	Human	Mouse
Number of coding exons	1	1
Total coding length	1353	1353

Gene number: 46
Human locus: HSMHCPU15 5833 bp DNA PRI 29-JUL-1993
DEFINITION H.sapiens gene for major histocompatibility complex encoded proteasome subunit LMP2.
–
Mouse locus: MUSLMP2A 6101 bp DNA ROD 18-AUG-1993
DEFINITION Mus musculus proteasome (lmp2) gene, complete mRNA.
ACCESSION L11613
–
Exon Lengths:
H: 60 68 132 130 142 128
M: 281 132 130 99

	Human	Mouse
Number of coding exons	6	4
Total coding length	660	642

Gene number: 47
Human locus: HSU72648 4850 bp DNA PRI 19-DEC-1996
DEFINITION Human alpha2-C4-adrenergic receptor gene, complete cds.
ACCESSION U72648
–
Mouse locus: MUSADRA 2409 bp DNA ROD 22-JUL-1993
DEFINITION Mouse alpha-2 adrenergic receptor gene, complete cds.
ACCESSION M97516
–
Exon Lengths:
H: 1386
M: 1377

	Human	Mouse
Number of coding exons	1	1
Total coding length	1386	1377

Gene number: 48
Human locus: HUMMK 4638 bp DNA PRI 12-SEP-1992
DEFINITION Human midkine gene, complete cds.
ACCESSION D10604 D90540
–
Mouse locus: MUSMKPG 2929 bp DNA ROD 15-SEP-1990
DEFINITION Mouse retinoic acid-responsive protein (MK) gene, complete cds.
ACCESSION M34094 J05447
–
Exon Lengths:
H: 76 168 162 26
M: 76 159 162 26

	Human	Mouse
Number of coding exons	4	4
Total coding length	432	423

Gene number: 49
Human locus: HSMYF4G 2804 bp DNA PRI 27-SEP-1996
DEFINITION Human myf4 gene for skeletal muscle-specific transcription factor.
ACCESSION X62155
–
Mouse locus: MUSMYOGEN 4145 bp DNA ROD 03-SEP-1992
DEFINITION Mus musculus myogenin gene, complete cds.
ACCESSION M95800
–
Exon Lengths:
H: 471 81 123
M: 471 82 122

	Human	Mouse
Number of coding exons	3	3
Total coding length	675	675

Gene number: 50
Human locus: HUMHISAB 1314 bp DNA PRI 07-MAR-1995
DEFINITION Human histone H1 (H1F3) gene, complete cds.
ACCESSION M60747
–
Mouse locus: MMHISTH1 1943 bp DNA ROD 22-AUG-1996
DEFINITION M.musculus gene for histone H1.
ACCESSION Z38128
–
Exon Lengths:
H: 666
M: 666

	Human	Mouse
Number of coding exons	1	1
Total coding length	666	666

Gene number: 51
Human locus: HUMOTNPI 1338 bp DNA PRI 03-MAY-1996
DEFINITION Human prepro-oxytocin-neurophysin I (OXT) gene, complete cds.
ACCESSION M11186
–
Mouse locus: MUSOXYNEUI 2003 bp DNA ROD 11-MAR-1992
DEFINITION Mouse oxytocin-neurophysin I gene, complete cds.
ACCESSION M88355
–
Exon Lengths:
H: 120 199 56
M: 120 202 56

	Human	Mouse
Number of coding exons	3	3
Total coding length	375	378

Gene number: 52
Human locus: HUMTKRA 13500 bp DNA PRI 14-JAN-1995
DEFINITION Human thymidine kinase gene, complete cds, with clustered Alu repeats in the introns.
–
Mouse locus: MUSTKM 2939 bp DNA ROD 24-SEP-1992
DEFINITION Mouse thymidine kinase gene, complete cds.
ACCESSION M68489
–
Exon Lengths:

H:	66	32	111	94	90	120	192
M:	66	32	111	94	90	120	189

	Human	Mouse
Number of coding exons	7	7
Total coding length	705	702

Gene number: 53
Human locus: HUMADRBRA 3458 bp DNA PRI 13-FEB-1996
DEFINITION Human beta-2-adrenergic receptor gene, complete cds.
ACCESSION J02960
–
Mouse locus: MMB2ARG 4928 bp DNA ROD 22-MAR-1991
DEFINITION Mouse gene for beta-2-adrenergic receptor.
ACCESSION X15643
–
Exon Lengths:

H:	1242	
M:	1257	

	Human	Mouse
Number of coding exons	1	1
Total coding length	1242	1257

Gene number: 54
Human locus: HUMMETIII 2167 bp DNA PRI 01-JUL-1992
DEFINITION Human metallothionein-III gene, complete cds.
ACCESSION M93311
–
Mouse locus: MUSMETIII 2649 bp DNA ROD 01-JUL-1992
DEFINITION Mouse metallothionein-III gene, complete cds.
ACCESSION M93310
–
Exon Lengths:

H:	31	66	110
M:	31	66	110

	Human	Mouse
Number of coding exons	3	3
Total coding length	207	207

Gene number: 55
Human locus: HUMXRCC1G 37785 bp DNA PRI 30-JAN-1995
DEFINITION Human XRCC1 DNA repair gene, genomic.
ACCESSION L34079
–
Mouse locus: MUSXRCC1G 37349 bp DNA ROD 30-JAN-1995
DEFINITION Mouse XRCC1 DNA repair gene, genomic.
ACCESSION L34078
–
Exon Lengths:

H:	51	93	111	159	75	112	110	112	259	117	94	133	55
M:	51	93	117	159	76	112	100	258	116	94	130	55	137

	Human	Mouse
Number of coding exons	17	16
Total coding length	1902	1779

Gene number: 56
Human locus: HUMG0S24B 3889 bp DNA PRI 09-MAY-1997
DEFINITION Homo sapiens zinc finger transcriptional regulator (GOS24) gene, complete cds.
–
Mouse locus: MUSZPF36G 7493 bp DNA ROD 24-OCT-1995
DEFINITION Mus musculus tristetraprolin (zpf-36) gene, complete cds.
ACCESSION L42317
–
Exon Lengths:

H:	24	957	
M:	24	936	

	Human	Mouse
Number of coding exons	2	2
Total coding length	981	960

Gene number: 57
Human locus: HSAGL1 1138 bp DNA PRI 24-APR-1993
DEFINITION Human alpha-globin germ line gene.
ACCESSION V00488
–
Mouse locus: MUSHBA 1441 bp DNA ROD 01-SEP-1988
DEFINITION Mouse alpha-globin gene, complete cds.
ACCESSION J00410 M13126
–
Exon Lengths:

H:	95	129	
M:	95	205	129

	Human	Mouse
Number of coding exons	2	3
Total coding length	224	429

Gene number: 58
Human locus: HSPGK2G 1911 bp DNA PRI 12-SEP-1993
DEFINITION Human testis-specific PGK-2 gene for phosphoglycerate kinase (ATP:3-phospho-D-glycerate 1-phosphotransferase, EC 2.7.2.3).
–
Mouse locus: MUSPGK2 2147 bp DNA ROD 02-NOV-1992
DEFINITION Mouse testis-specific phosphoglycerate kinase (pgk-2) gene, complete cds.
–
Exon Lengths:
H: 1254
M: 1254

	Human	Mouse
Number of coding exons	1	1
Total coding length	1254	1254

Gene number: 59
Human locus: HSU57623 9170 bp DNA PRI 16-JUN-1996
DEFINITION Human fatty acid binding protein FABP gene, complete cds.
ACCESSION U57623
–
Mouse locus: MMU02884 8765 bp DNA ROD 03-FEB-1995
DEFINITION Mus musculus mammary-derived growth inhibitor (MDGI) gene, complete cds.
–
Exon Lengths:
H: 73 173 102 54
M: 73 173 102 54

	Human	Mouse
Number of coding exons	4	4
Total coding length	402	402

Gene number: 60
Human locus: HSARYLA 3637 bp DNA PRI 24-APR-1993
DEFINITION Human DNA for arylsulphatase A (EC 3.1.6.1).
ACCESSION X52150
–
Mouse locus: MMDNAASFA 4342 bp DNA ROD 20-NOV-1997
DEFINITION M.musculus gene for arylsulfatase A.
ACCESSION X73231
–
Exon Lengths:
H: 218 241 219 170 125 128 103 320
M: 215 241 219 170 125 128 103 320

	Human	Mouse
Number of coding exons	8	8
Total coding length	1524	1521

Gene number: 61
Human locus: S63168 1594 bp DNA PRI 23-AUG-1993
DEFINITION CCAAT/enhancer-binding protein delta=transcription factor CRP3 homolog [human, prostate carcinoma cell line LNCaP, Genomic, 1594
–
Mouse locus: MUSCRP3A 1146 bp DNA ROD 27-JUN-1994
DEFINITION Mouse C/EBP-related protein (CRP3) gene.
ACCESSION M85144
–
Exon Lengths:
H: 810
M: 804

	Human	Mouse
Number of coding exons	1	1
Total coding length	810	804

Gene number: 62
Human locus: HSHSP27 2496 bp DNA PRI 28-MAR-1995
DEFINITION Human gene for 27kDa heat shock protein (hsp 27).
ACCESSION X03900
–
Mouse locus: MUSHSP25A 3058 bp DNA ROD 04-AUG-1993
DEFINITION Mus musculus small heat shock protein (HSP25) gene,.
ACCESSION L07577
–
Exon Lengths:
H: 364 64 172
M: 375 66 189

	Human	Mouse
Number of coding exons	3	3
Total coding length	600	630

Gene number: 63
Human locus: HUMROD1X 2841 bp DNA PRI 09-JAN-1995
DEFINITION Human rod outer segment membrane protein 1 (ROM1) gene exons 1-3, complete cds.
–
Mouse locus: MUSROM1X 2787 bp DNA ROD 14-JUL-1993
DEFINITION Mouse rod outer segment membrane protein 1 (Rom1) gene exons 1-3, complete cds.
–
Exon Lengths:
H: 590 247 219
M: 590 247 219

	Human	Mouse
Number of coding exons	3	3
Total coding length	1056	1056

Gene number: 64
Human locus: HUMSSTR3X 1413 bp DNA PRI 13-JAN-1995
DEFINITION Human somatostatin receptor subtype 3 (SSTR3) gene, complete cds.
ACCESSION M96738
–
Mouse locus: MUSSSTR3A 1797 bp DNA ROD 01-APR-1993
DEFINITION Mouse somatostatin receptor (SSTR3) gene, complete cds.
ACCESSION M91000
–
Exon Lengths:
H: 1257
M: 1287

	Human	Mouse
Number of coding exons	1	1
Total coding length	1257	1287

Gene number: 65
Human locus: HSPNMTB 3799 bp DNA PRI 24-APR-1993
DEFINITION Human gene for phenylethanolamine N-methylase (PNMT) (EC 2.1.1.28).
ACCESSION X52730
–
Mouse locus: MUSPNMT 3144 bp DNA ROD 28-JUN-1995
DEFINITION Mouse phenylethanolamine N-methyltransferase gene, complete cds.
ACCESSION L12687
–
Exon Lengths:
H: 202 208 439
M: 235 208 445

	Human	Mouse
Number of coding exons	3	3
Total coding length	849	888

Gene number: 66
Human locus: HSIGF2G 8837 bp DNA PRI 27-JAN-1993
DEFINITION Human gene for insulin-like growth factor II.
ACCESSION X03562 M13970 M14116 M14117 M14118
–
Mouse locus: MMU71085 27874 bp DNA ROD 17-DEC-1997
DEFINITION Mus musculus insulin-like growth factor II (Igf2) gene, complete cds.
–
Exon Lengths:
H: 157 237
M: 157 149 237

	Human	Mouse
Number of coding exons	2	3
Total coding length	394	543

Gene number: 67
Human locus: HUMADAG 36741 bp DNA PRI 04-OCT-1995
DEFINITION Human adenosine deaminase (ADA) gene, complete cds.
ACCESSION M13792
–
Mouse locus: MMU73107 29807 bp DNA ROD 30-MAR-1998
DEFINITION Mus musculus adenosine deaminase (ADA) gene, complete cds.
ACCESSION U73107
–
Exon Lengths:

H:	33	62	123	144	116	128	72	102	65	130	103	14
M:	33	62	123	144	116	128	72	102	65	130	84	

	Human	Mouse
Number of coding exons	12	11
Total coding length	1092	1059

Gene number: 68
Human locus: HUMSMPD1G 5588 bp DNA PRI 31-AUG-1995
DEFINITION Homo sapiens acid sphingomyelinase (SMPD1) gene, complete cds, ORF's 1-3, complete cds's.
–
Mouse locus: MMASM1G 4775 bp DNA ROD 28-APR-1994
DEFINITION M.musculus (balb-c) gene for sphingomyelin phosphodiesterase.
ACCESSION Z14132 Z31654
–
Exon Lengths:

H:	312	773	172	77	146	410
M:	306	773	172	77	146	410

	Human	Mouse
Number of coding exons	6	6
Total coding length	1890	1884

Gene number: 69
Human locus: HUMCOX5B 2593 bp DNA PRI 01-NOV-1994
DEFINITION Homo sapiens cytochrome c oxidase subunit Vb (COX5B) gene, complete cds.
–
Mouse locus: MUSCYTCOVB 2540 bp DNA ROD 04-AUG-1994
DEFINITION Mouse cytochrome c oxidase Vb subunit gene, complete cds.
ACCESSION M77040 M38201
–
Exon Lengths:

H:	103	74	100	113
M:	100	74	100	113

	Human	Mouse
Number of coding exons	4	4
Total coding length	390	387

Gene number: 70
Human locus: HUMAE1ERY 21319 bp DNA PRI 01-DEC-1994
DEFINITION Human anion exchanger (AE1) gene, exons 1-20.
ACCESSION L35930
–
Mouse locus: MUSBAND32 12519 bp DNA ROD 15-JUN-1989
DEFINITION Mouse band 3 anion exchange protein gene, complete cds.
ACCESSION J02756
–
Exon Lengths:

H:	15	91	62	181	136	124	85	182	211	195	149	195	174
M:	6	121	74	181	136	124	82	182	211	213	149	195	171

	Human	Mouse
Number of coding exons	19	18
Total coding length	2736	2699

Gene number: 71
Human locus: HUMNUCLEO 10942 bp DNA PRI 07-JAN-1995
DEFINITION Human nucleolin gene, complete cds.
ACCESSION M60858 J05584
–
Mouse locus: MMNUCLEO 11478 bp DNA ROD 27-AUG-1998
DEFINITION Mouse nucleolin gene.
ACCESSION X07699
–
Exon Lengths:

H:	18	117	478	198	87	142	125	124	158	124	134	127	215
M:	18	117	496	186	87	142	125	124	155	124	122	127	224

	Human	Mouse
Number of coding exons	14	14
Total coding length	2124	2124

Gene number: 72
Human locus: S45332 8647 bp DNA PRI 23-DEC-1992
DEFINITION erythropoietin receptor [human, placental, Genomic, 8647 nt].
ACCESSION S45332
–
Mouse locus: MMERYPR 6561 bp DNA ROD 13-MAY-1992
DEFINITION Mouse gene for erythropoietin receptor.
ACCESSION X53081
–
Exon Lengths:

H:	115	136	176	158	154	88	88	612
M:	115	133	176	158	154	88	88	612

	Human	Mouse
Number of coding exons	8	8
Total coding length	1527	1524

Gene number: 73
Human locus: HUMINSPR 3943 bp DNA PRI 06-JAN-1995
DEFINITION Human alpha-type insulin gene and 5' flanking polymorphic region.
ACCESSION M10039
–
Mouse locus: MMINSIIG 2408 bp DNA ROD 09-APR-1993
DEFINITION Mouse preproinsulin gene II.
ACCESSION X04724
–
Exon Lengths:
H: 187 146
M: 186 147

	Human	Mouse
Number of coding exons	2	2
Total coding length	333	333

Gene number: 74
Human locus: HUMHST 6616 bp DNA PRI 22-AUG-1995
DEFINITION Human transforming protein (hst) gene, complete cds.
ACCESSION J02986 M16338
–
Mouse locus: MMKFGF 3989 bp DNA ROD 11-NOV-1994
DEFINITION Mouse hst/kFGF genomic DNA.
ACCESSION X14849 M28516
–
Exon Lengths:
H: 340 104 177
M: 328 104 177

	Human	Mouse
Number of coding exons	3	3
Total coding length	621	609

Gene number: 75
Human locus: HUMPLPSPC 3409 bp DNA PRI 17-AUG-1998
DEFINITION Human pulmonary surfactant protein C (SP-C) and pulmonary surfactant protein C1 (SP-C1) genes, complete cds.
–
Mouse locus: MUSPSPC 3633 bp DNA ROD 24-JAN-1991
DEFINITION Mouse pulmonary surfactant protein SP-C (SFTP2) gene, complete cds.
ACCESSION M38314
–
Exon Lengths:
H: 42 159 123 111 159
M: 42 159 123 111 147

	Human	Mouse
Number of coding exons	5	5
Total coding length	594	582

Gene number: 76
Human locus: HSCOSEG 3521 bp DNA PRI 15-JAN-1992
DEFINITION H.sapiens coseg gene for vasopressin-neurophysin precursor.
ACCESSION X62890
–
Mouse locus: MUSVASNEU 3494 bp DNA ROD 11-MAR-1992
DEFINITION Mouse vasopressin-neurophysin II gene, complete cds.
ACCESSION M88354
–
Exon Lengths:
H: 120 202 173
M: 132 202 173

	Human	Mouse
Number of coding exons	3	3
Total coding length	495	507

Gene number: 77
Human locus: HSB3A 3683 bp DNA PRI 18-MAY-1993
DEFINITION H.sapiens gene for beta-3-adrenergic receptor.
ACCESSION X72861
–
Mouse locus: MMB3A 3438 bp DNA ROD 18-MAY-1993
DEFINITION M.musculus gene for beta-3-adrenergic receptor.
ACCESSION X72862
–
Exon Lengths:
H: 1205 22
M: 1163 40

	Human	Mouse
Number of coding exons	2	2
Total coding length	1227	1203

Gene number: 78
Human locus: HUMLORI 3321 bp DNA PRI 06-OCT-1992
DEFINITION Human loricrin gene exons 1 and 2, complete cds.
ACCESSION M94077
–
Mouse locus: MMU09189 6530 bp DNA ROD 30-NOV-1995
DEFINITION Mus musculus loricrin gene, complete cds.
ACCESSION U09189
–
Exon Lengths:
H: 951
M: 1446

	Human	Mouse
Number of coding exons	1	1
Total coding length	951	1446

Gene number: 79
Human locus: HSU37055 9980 bp DNA PRI 16-MAY-1996
DEFINITION Human hepatocyte growth factor-like protein gene, complete cds.
ACCESSION U37055 M74179
–
Mouse locus: MUSHEPGFA 6751 bp DNA ROD 07-OCT-1994
DEFINITION Mouse hepatocyte growth factor-like protein gene, complete cds.
ACCESSION M74180
–
Exon Lengths:

H:	52	148	113	115	137	121	119	169	131	103	137	36	121
M:	52	148	113	115	137	121	119	196	131	103	137	36	109

	Human	Mouse
Number of coding exons	18	18
Total coding length	2136	2151

Gene number: 80
Human locus: HSH11 1034 bp DNA PRI 19-APR-1993
DEFINITION H.sapiens H1.1 gene for histone H1.
ACCESSION X57130
–
Mouse locus: MUSH1X 1781 bp DNA ROD 04-AUG-1994
DEFINITION Mouse histone H1, complete cds.
ACCESSION L26164
–
Exon Lengths:

H:	648
M:	642

	Human	Mouse
Number of coding exons	1	1
Total coding length	648	642

Gene number: 81
Human locus: HSU66875 1569 bp DNA PRI 31-MAY-1997
DEFINITION Homo sapiens cytochrome oxidase subunit VIa heart isoform precursor (COX6AH) gene, complete cds.
–
Mouse locus: MMU63716 2324 bp DNA ROD 02-FEB-1997
DEFINITION Mus musculus cytochrome C oxidase subunit VIa heart isoform (coxVIaH) gene, nuclear gene encoding mitochondrial protein,
–
Exon Lengths:

H:	73	137	84
M:	73	137	84

	Human	Mouse
Number of coding exons	3	3
Total coding length	294	294

Gene number: 82
Human locus: HSINT2 11608 bp DNA PRI 25-JUN-1997
DEFINITION Human int-2 proto-oncogene.
ACCESSION X14445
–
Mouse locus: MMINT2 8283 bp DNA ROD 29-NOV-1994
DEFINITION Mouse int-2 gene.
ACCESSION Y00848 M26284 X68450
–
Exon Lengths:
H: 220 104 396
M: 220 104 414

	Human	Mouse
Number of coding exons	3	3
Total coding length	720	738

Gene number: 83
Human locus: HSBGL3 2052 bp DNA PRI 07-OCT-1996
DEFINITION Human germ line gene for beta-globin.
ACCESSION V00499
–
Mouse locus: MUSHBBMAJ 6532 bp DNA ROD 15-JUN-1988
DEFINITION Mouse beta-globin major gene.
ACCESSION J00413 K01748 K03545 X00624
–
Exon Lengths:
H: 92 223 129
M: 92 223 129

	Human	Mouse
Number of coding exons	3	3
Total coding length	444	444

Gene number: 84
Human locus: HUMALIFA 7614 bp DNA PRI 31-OCT-1994
DEFINITION Human leukemia inhibitory factor (LIF) gene, complete cds.
ACCESSION M63420 J05436
–
Mouse locus: MUSALIFA 8757 bp DNA ROD 10-APR-1991
DEFINITION Mouse leukemia inhibitory factor (LIF) gene, complete cds.
ACCESSION M63419 J05435
–
Exon Lengths:
H: 19 179 411
M: 19 182 411

	Human	Mouse
Number of coding exons	3	3
Total coding length	609	612

Gene number: 85
Human locus: HUMFABP 5204 bp DNA PRI 08-NOV-1994
DEFINITION Human, intestinal fatty acid binding protein gene, complete cds, and an Alu repetitive element.
–
Mouse locus: MUSFABPI 5039 bp DNA ROD 25-FEB-1992
DEFINITION Mouse Fabpi gene, exons 1-4.
ACCESSION M65033
–
Exon Lengths:

H:	67	173	108	51
M:	103	173	108	222

	Human	Mouse
Number of coding exons	4	4
Total coding length	399	606

Gene number: 86
Human locus: HUMLYTOXBB 6305 bp DNA PRI 14-MAY-1996
DEFINITION Homo sapiens lymphotoxin-beta gene, complete cds.
ACCESSION L11016
–
Mouse locus: MMU16984 6914 bp DNA ROD 20-JUL-1995
DEFINITION Mus musculus lymphotoxin-beta (LT-beta) gene, complete cds.
ACCESSION U16984
–
Exon Lengths:

H:	162	46	72	455
M:	162	316	443	

	Human	Mouse
Number of coding exons	4	3
Total coding length	735	921

Gene number: 87
Human locus: HUMLYL1B 4569 bp DNA PRI 18-MAR-1996
DEFINITION Human LYL-1 protein gene, complete cds.
ACCESSION M22638
–
Mouse locus: MMLYL1 3678 bp DNA ROD 02-AUG-1991
DEFINITION Mouse Lyl-1 gene.
ACCESSION X55055
–
Exon Lengths:

H:	296	92	416
M:	333	90	414

	Human	Mouse
Number of coding exons	3	3
Total coding length	804	837

Gene number: 88
Human locus: HUMANFA 2710 bp DNA PRI 01-NOV-1994
DEFINITION Human atrial natriuretic factor (PND) gene, complete cds.
ACCESSION K02043
–
Mouse locus: MUSANF 1983 bp DNA ROD 16-DEC-1985
DEFINITION Mouse PND gene encoding atrial natriuretic factor, complete cds.
ACCESSION K02781
–
Exon Lengths:
H: 123 327 6
M: 120 327 12

	Human	Mouse
Number of coding exons	3	3
Total coding length	456	459

Gene number: 89
Human locus: HUMG0S19A 4102 bp DNA PRI 07-JAN-1991
DEFINITION Human homologue-1 of gene encoding alpha subunit of murine cytokine (MIP1/SCI), complete cds.
–
Mouse locus: MMSCIMIP 1988 bp DNA ROD 08-MAY-1993
DEFINITION Mouse SCI/MIP-1a gene for stem cell inhibitor/macrophage inflammatory protein 1a.
–
Exon Lengths:
H: 73 115 91
M: 76 112 91

	Human	Mouse
Number of coding exons	3	3
Total coding length	279	279

Gene number: 90
Human locus: HUMALPI 5291 bp DNA PRI 29-APR-1996
DEFINITION Human intestinal alkaline phosphatase (ALPI) gene, complete cds.
ACCESSION J03930
–
Mouse locus: MUSIAP 5293 bp DNA ROD 02-DEC-1991
DEFINITION Mouse intestinal alkaline phosphatase (IAP) gene, complete cds.
ACCESSION M61705 M35029
–
Exon Lengths:
H: 67 117 116 175 173 135 73 135 192 117 287
M: 67 117 116 175 173 135 73 135 192 114 383

	Human	Mouse
Number of coding exons	11	11
Total coding length	1587	1680

Gene number: 91
Human locus: HUMFPR1A 6931 bp DNA PRI 18-MAR-1994
DEFINITION Human N-formyl peptide receptor (FPR1) gene, complete cds and Alu repeats.
–
Mouse locus: MUSNFORREC 1524 bp DNA ROD 25-JAN-1994
DEFINITION Mouse N-formyl peptide chemotactic receptor gene, complete cds.
ACCESSION L22181
–
Exon Lengths:
H: 1053
M: 1095

	Human	Mouse
Number of coding exons	1	1
Total coding length	1053	1095

Gene number: 92
Human locus: HSSPRO 5296 bp DNA PRI 20-MAY-1992
DEFINITION Human S-protein gene, complete cds.
ACCESSION X05006
–
Mouse locus: MMVITRO 5004 bp DNA ROD 17-FEB-1997
DEFINITION M.musculus gene for vitronectin.
ACCESSION X72091
–
Exon Lengths:
H: 64 120 345 140 157 153 345 113
M: 64 120 342 140 157 153 351 110

	Human	Mouse
Number of coding exons	8	8
Total coding length	1437	1437

Gene number: 93
Human locus: HSGCSFG 2960 bp DNA PRI 24-APR-1993
DEFINITION Human gene for granulocyte colony-stimulating factor (G-CSF).
ACCESSION X03656
–
Mouse locus: MMGCSFG 3054 bp DNA ROD 10-JAN-1991
DEFINITION Murine G-CSF gene for granulocyte colony stimulating factor precursor.
–
Exon Lengths:
H: 40 164 108 147 165
M: 40 173 108 147 159

	Human	Mouse
Number of coding exons	5	5
Total coding length	624	627

Gene number: 94
Human locus: HUMTNFBA 2140 bp DNA PRI 14-JAN-1995
DEFINITION Human tumor necrosis factor-beta (TNFB) gene, complete cds.
ACCESSION M55913
–
Mouse locus: MMTNFBG 3219 bp DNA ROD 08-MAY-1993
DEFINITION Mouse tumor necrosis factor-beta (lymphotoxin) gene.
ACCESSION Y00137
–
Exon Lengths:

H:	99	106	413
M:	96	100	413

	Human	Mouse
Number of coding exons	3	3
Total coding length	618	609

Gene number: 95
Human locus: HSU16720 8868 bp DNA PRI 28-OCT-1995
DEFINITION Human interleukin 10 (IL10) gene, complete cds.
ACCESSION U16720
–
Mouse locus: MUSIL10Z 7207 bp DNA ROD 30-JUN-1992
DEFINITION Mouse interleukin 10 (IL10) gene, complete cds.
ACCESSION M84340
–
Exon Lengths:

H:	165	60	153	66	93
M:	165	60	153	66	93

	Human	Mouse
Number of coding exons	5	5
Total coding length	537	537

Gene number: 96
Human locus: HUMCP21OH 4042 bp DNA PRI 01-NOV-1994
DEFINITION Human 21-hydroxylase B gene, complete cds.
ACCESSION M26856 X05448
–
Mouse locus: MUS21OHA1 3307 bp DNA ROD 15-MAR-1990
DEFINITION Mouse steroid 21-hydroxylase A (21-OHase A) gene, complete cds.
ACCESSION M15009
–
Exon Lengths:

H:	202	90	155	102	102	87	201	179	104	266
M:	202	90	143	96	102	87	201	170	104	269

	Human	Mouse
Number of coding exons	10	10
Total coding length	1488	1464

Gene number: 97
Human locus: HUMMIS 3100 bp DNA PRI 03-MAY-1996
DEFINITION Human Mullerian inhibiting substance gene, complete cds.
ACCESSION K03474
–
Mouse locus: MMAMH 2870 bp DNA ROD 22-JAN-1992
DEFINITION M.musculus mAmh gene for anti-Mullerian hormone.
ACCESSION X63240
–
Exon Lengths:

H:	412	143	109	160	859
M:	403	143	109	160	853

	Human	Mouse
Number of coding exons	5	5
Total coding length	1683	1668

Gene number: 98
Human locus: HUMAPOE4 5515 bp DNA PRI 09-NOV-1994
DEFINITION Human apolipoprotein E (epsilon-4 allele) gene, complete cds.
ACCESSION M10065 J03053 J03054
–
Mouse locus: MUSAPE 4856 bp DNA ROD 26-SEP-1998
DEFINITION Mus musculus gene for apolipoprotein, exons 1,2,3,4, complete cds.
ACCESSION D00466
–
Exon Lengths:

H:	43	193	718
M:	43	169	724

	Human	Mouse
Number of coding exons	3	3
Total coding length	954	936

Gene number: 99
Human locus: HUMREGB 4251 bp DNA PRI 15-SEP-1990
DEFINITION Human regenerating protein (reg) gene, complete cds.
ACCESSION J05412
–
Mouse locus: MUSREGI 3756 bp DNA ROD 18-APR-1996
DEFINITION Mouse reg I gene for regenerating protein I, complete cds.
ACCESSION D14010
–
Exon Lengths:

H:	64	119	138	112	68
M:	61	119	138	112	68

	Human	Mouse
Number of coding exons	5	5
Total coding length	501	498

Gene number: 100
Human locus: HUMPROLA 1404 bp DNA PRI 19-MAY-1995
DEFINITION Human cathepsin L gene, complete cds.
ACCESSION M20496
–
Mouse locus: MUSPROL 1413 bp DNA ROD 17-MAY-1994
DEFINITION Mouse cathepsin L gene, complete cds, clones a-H-ras-1 and RIT-1.
ACCESSION M20495
–
Exon Lengths:
H: 1002
M: 1005

	Human	Mouse
Number of coding exons	1	1
Total coding length	1002	1005

Gene number: 101
Human locus: HSU29874 6155 bp DNA PRI 29-FEB-1996
DEFINITION Human Flt3 ligand gene and Flt3 ligand alternatively spliced isoform gene, complete cds.
–
Mouse locus: MMU44024 4799 bp DNA ROD 02-APR-1996
DEFINITION Mus musculus Flt3 ligand gene, complete cds.
ACCESSION U44024
–
Exon Lengths:
H: 33 111 54 144 139 179 48
M: 33 122 46 144 144 189 21

	Human	Mouse
Number of coding exons	7	7
Total coding length	708	699

Gene number: 102
Human locus: HSA6693 3448 bp DNA PRI 13-JUN-1998
DEFINITION Homo sapiens UHS KerA gene.
ACCESSION AJ006693
–
Mouse locus: MUSSER1 3366 bp DNA ROD 11-JAN-1991
DEFINITION Mouse serine 1 ultra high sulfur protein gene, complete cds.
ACCESSION M37759
–
Exon Lengths:
H: 510
M: 693

	Human	Mouse
Number of coding exons	1	1
Total coding length	510	693

Gene number: 103
Human locus: HUMIL2RGA 4038 bp DNA PRI 18-OCT-1993
DEFINITION Human (IL2RG) gene, complete cds with repeats.
ACCESSION L19546
–
Mouse locus: MMU21795 5267 bp DNA ROD 25-MAR-1995
DEFINITION Mus musculus common cytokine receptor gamma chain gene, complete cds.
–
Exon Lengths:

H:	115	154	185	140	163	97	70	186
M:	115	154	185	143	163	97	67	186

	Human	Mouse
Number of coding exons	8	8
Total coding length	1110	1110

Gene number: 104
Human locus: HUMCRPGA 2480 bp DNA PRI 01-NOV-1994
DEFINITION Human C-reactive protein gene, complete cds.
ACCESSION M11725
–
Mouse locus: MMCRPG 2140 bp DNA ROD 09-APR-1993
DEFINITION Murine crp gene for C-reactive protein.
ACCESSION X13588
–
Exon Lengths:

H:	61	614
M:	64	614

	Human	Mouse
Number of coding exons	2	2
Total coding length	675	678

Gene number: 105
Human locus: HSBCDIFFI 3230 bp DNA PRI 30-MAR-1992
DEFINITION H.sapiens gene for B cell differentiation factor I.
ACCESSION X12706
–
Mouse locus: MMIL5G 6727 bp DNA ROD 10-APR-1993
DEFINITION Murine gene for interleukin 5 (eosinophil differentiation factor).
ACCESSION X06271
–
Exon Lengths:

H:	144	33	129	99
M:	141	33	129	99

	Human	Mouse
Number of coding exons	4	4
Total coding length	405	402

Gene number: 106
Human locus: HUMTHY1A 2806 bp DNA PRI 14-JAN-1995
DEFINITION Human Thy-1 glycoprotein gene, complete cds.
ACCESSION M11749
–
Mouse locus: MUSTHY1GC 3257 bp DNA ROD 01-SEP-1988
DEFINITION Mouse Thy-1.2 gene, clones pcT108 and pcT34.
ACCESSION M11160
–
Exon Lengths:
H: 37 336 113
M: 37 339 113

	Human	Mouse
Number of coding exons	3	3
Total coding length	486	489

Gene number: 107
Human locus: HSUPA 7258 bp DNA PRI 07-FEB-1997
DEFINITION H.sapiens uPA gene.
ACCESSION X02419
–
Mouse locus: MUSUPAA 9950 bp DNA ROD 15-MAR-1990
DEFINITION Mouse Murine urokinase-type plasminogen activator protein gene, complete cds.
–
Exon Lengths:
H: 57 28 108 175 92 220 149 141 149 177
M: 57 31 108 175 92 223 149 141 149 177

	Human	Mouse
Number of coding exons	10	10
Total coding length	1296	1302

Gene number: 108
Human locus: HUMSAP01 1394 bp DNA PRI 11-MAR-1998
DEFINITION Homo sapiens gene for serum amyloid P component, complete cds.
ACCESSION D00097
–
Mouse locus: MUSSAPRB 1350 bp DNA ROD 15-MAR-1990
DEFINITION Mouse serum amyloid P component gene, complete cds.
ACCESSION M29535
–
Exon Lengths:
H: 64 608
M: 67 608

	Human	Mouse
Number of coding exons	2	2
Total coding length	672	675

Gene number: 109
Human locus: HUMPAP 4497 bp DNA PRI 07-JAN-1995
DEFINITION Homo sapiens pancreatits-associated protein (PAP) gene, complete cds.
–
Mouse locus: D63360 4292 bp DNA ROD 02-APR-1997
DEFINITION Mouse DNA for regIIIbeta/PAP protein, complete cds.
ACCESSION D63360
–
Exon Lengths:

H:	76	119	138	127	68
M:	76	119	138	127	68

	Human	Mouse
Number of coding exons	5	5
Total coding length	528	528

Gene number: 110
Human locus: HUMIL1B 7824 bp DNA PRI 09-AUG-1995
DEFINITION Human interleukin 1-beta (IL1B) gene, complete cds.
ACCESSION M15840
–
Mouse locus: MMIL1BG 7100 bp DNA ROD 16-SEP-1994
DEFINITION Murine interleukin 1-beta gene.
ACCESSION X04964
–
Exon Lengths:

H:	47	52	202	165	131	213
M:	47	49	202	171	131	210

	Human	Mouse
Number of coding exons	6	6
Total coding length	810	810

Gene number: 111
Human locus: HUMCAPG 3734 bp DNA PRI 31-OCT-1994
DEFINITION Human cathepsin G gene, complete cds.
ACCESSION J04990
–
Mouse locus: MUSCATHG 3438 bp DNA ROD 03-JUN-1993
DEFINITION Mus musculus cathepsin G gene, complete cds.
ACCESSION M96801
–
Exon Lengths:

H:	55	148	136	255	174
M:	55	148	136	255	192

	Human	Mouse
Number of coding exons	5	5
Total coding length	768	786

Gene number: 112
Human locus: HUMCTLA1 4505 bp DNA PRI 23-MAY-1995
DEFINITION Human cytotoxic T-lymphocyte-associated serine esterase 1 (CTLA1) gene, complete cds.
–
Mouse locus: MUSSPCTLS 4348 bp DNA ROD 30-JUN-1997
DEFINITION Mouse cytotoxic T lymphocyte-specific serine protease CCPI gene, complete cds.
–
Exon Lengths:

H:	55	148	136	261	144
M:	55	148	136	261	144

	Human	Mouse
Number of coding exons	5	5
Total coding length	744	744

Gene number: 113
Human locus: HSLACTG 3310 bp DNA PRI 24-APR-1993
DEFINITION Human alpha-lactalbumin gene.
ACCESSION X05153
–
Mouse locus: MUSALCALB 3045 bp DNA ROD 15-OCT-1992
DEFINITION Mouse alpha-lactalbumin gene, complete cds.
ACCESSION M87863
–
Exon Lengths:

H:	133	159	76	61
M:	136	159	76	61

	Human	Mouse
Number of coding exons	4	4
Total coding length	429	432

Gene number: 114
Human locus: HUMOSTP 10881 bp DNA PRI 30-MAY-1996
DEFINITION Human DNA for osteopontin, complete cds.
ACCESSION D14813
–
Mouse locus: MMOESTEOP 5782 bp DNA ROD 17-FEB-1997
DEFINITION Murine gene for osteopontin.
ACCESSION X51834
–
Exon Lengths:

H:	54	39	81	42	324	405
M:	54	36	81	42	282	390

	Human	Mouse
Number of coding exons	6	6
Total coding length	945	885

Gene number: 115
Human locus: HSCD14G 1570 bp DNA PRI 23-JUN-1993
DEFINITION Human gene for CD14 differentiation antigen.
ACCESSION X06882
–
Mouse locus: MMCD14 2404 bp DNA ROD 21-AUG-1997
DEFINITION Mouse CD14 gene.
ACCESSION X13987
–
Exon Lengths:
H: 3 1125
M: 3 1098

	Human	Mouse
Number of coding exons	2	2
Total coding length	1128	1101

Gene number: 116
Human locus: HSGAPIGNA 2609 bp DNA PRI 10-AUG-1996
DEFINITION H.sapiens gap-I gene.
ACCESSION X74322
–
Mouse locus: MMU60528 5416 bp DNA ROD 08-AUG-1996
DEFINITION Mus musculus guanylin precursor gene, promoter region and complete cds.
–
Exon Lengths:
H: 75 208 65
M: 75 211 65

	Human	Mouse
Number of coding exons	3	3
Total coding length	348	351

Gene number: 117
Human locus: HSBGPG 1675 bp DNA PRI 24-APR-1993
DEFINITION Human gene for bone gla protein (BGP).
ACCESSION X04143
–
Mouse locus: MUSOGC 949 bp DNA ROD 17-FEB-1994
DEFINITION Mus musculus osteocalcin gene, complete cds.
ACCESSION L24429
–
Exon Lengths:
H: 64 33 70 130
M: 64 33 58 133

	Human	Mouse
Number of coding exons	4	4
Total coding length	297	288

Gene number: 118
Human locus: HSAPOAIA 2209 bp DNA PRI 03-NOV-1994
DEFINITION Human fetal gene for apolipoprotein AI precursor.
ACCESSION X01038

Mouse locus: MUSAICIIIA 9060 bp DNA ROD 08-MAR-1993
DEFINITION Mouse apolipoprotein A-I/CIII gene.
ACCESSION L04149

Exon Lengths:
H: 43 157 604
M: 9060

	Human	Mouse
Number of coding exons	3	1
Total coding length	804	9060

Table B.1: The HUMCOMP/MUSCOMP Datasets

Number	Locus
1.	HSACKI10
2.	HSAPOA2
3.	HSAPOAIA
4.	HSAPOC2G
5.	HSARYLA
6.	HSASML
7.	HSAT3
8.	HSATPCP1
9.	HSBCDIFFI
10.	HSBGPG
11.	HSBSF2
12.	HSC1INHIB
13.	HSCBMYHC
14.	HSCD1R3
15.	HSCKBG
16.	HSCKIIBE
17.	HSCOSE
18.	HSCPH70
19.	HSCST3G
20.	HSCYCLA
21.	HSCYP216
22.	HSDAO
23.	HSDNAMIA
24.	HSENO2
25.	HSERPG
26.	HSFAU1
27.	HSFBRGG
28.	HSFESFPS
29.	HSGCSFG
30.	HSGEBCMA
31.	HSGLUCG2
32.	HSHAP1
33.	HSHH3X3B
34.	HSHLADMBG
35.	HSHNRNPA
36.	HSHOX3D
37.	HSHSC70
38.	HSIFNG
39.	HSIGK12
40.	HSIL1AG
41.	HSIL1B
42.	HSL7A
43.	HSLCATG
44.	HSMECDAG
45.	HSMED
46.	HSMT1H
47.	HSNCAMX1
48.	HSNFM
49.	HSODCG
50.	HSODF2
51.	HSP53G
52.	HSPAT133
53.	HSPLAPL
54.	HSPNMTB
55.	HSPROPG
56.	HSPSAG
57.	HSRPS6G
58.	HSRPS7
59.	HSRPS8
60.	HSSHBG
61.	HSTNFA
62.	HSTPI1G
63.	HSTUBAG
64.	HSU20325
65.	HSU20982
66.	HSUBA52G
67.	HSUBR
68.	HSZNGP1
69.	HUMA1GLY2
70.	HUMADAG
71.	HUMADPRF02
72.	HUMAFP
73.	HUMAK1
74.	HUMANFA
75.	HUMANT1
76.	HUMATPSYB
77.	HUMBFXIII
78.	HUMBNPA
79.	HUMCEL
80.	HUMCFVII
81.	HUMCRPG
82.	HUMCRYABA
83.	HUMCYC1A
84.	HUMCYP2DG
85.	HUMEDHB17
86.	HUMEDN1B
87.	HUMEPOHYDD
88.	HUMFABP
89.	HUMFIXG
90.	HUMG0S8PP
91.	HUMGFP40H
92.	HUMHMG14A
93.	HUMHMG2A
94.	HUMHMGIY
95.	HUMHSKPQZ7
96.	HUMHSP89KD
97.	HUMIL4A
98.	HUMIMPDH
99.	HUMIRBPG
100.	HUMLHDC
101.	HUMLYTOXBB
102.	HUMMGPA
103.	HUMMHCD8A
104.	HUMMHDC3B
105.	HUMMIF
106.	HUMMIS
107.	HUMNKG5PRO
108.	HUMOPS
109.	HUMOSTP
110.	HUMP45C17
111.	HUMPALC
112.	HUMPBIPB
113.	HUMPCNA
114.	HUMPDHAL
115.	HUMPDHBET
116.	HUMPEPYYA
117.	HUMPGAMMG
118.	HUMPIM1A
119.	HUMPROT1B
120.	HUMPROT2
121.	HUMRBPA
122.	HUMRIGA
123.	HUMRIGBCHA
124.	HUMROD1X
125.	HUMRPS17A
126.	HUMSEMI
127.	HUMSOMI
128.	HUMTBGA
129.	HUMTDGF1A
130.	HUMTFPB
131.	HUMTHY1A
132.	HUMTKRA
133.	HUMTPALBU
134.	HUMTRHYAL
135.	HUMTROC
136.	HUMTS1
137.	HUMUBILP
138.	HUMVIPAA

Table B.2: The HKRM dataset

Number	Locus	Number	Locus	Number	Locus
1.	ACU08131	51.	CHEBGLI	101.	GGRIHBGEN
2.	AGGGLINE	52.	CHEBGLII	102.	GGVITIIG
3.	AGU04852	53.	CHKALDB	103.	GIBPROTP1A
4.	ALOEGLOBIM	54.	CHKAPOII	104.	GORAFPA
5.	ALOEGLOBIN	55.	CHKDPCP	105.	GORPROTP1A
6.	ALOPROTP1A	56.	CHKMAX	106.	GPIINS
7.	AMU12024	57.	CHKMYOD	107.	HAMAMYLP
8.	AMU12025	58.	CHKOVAL	108.	HAMHG5
9.	AOIRHODOPS	59.	CHKRPL30	109.	HROAMD1
10.	ASPROP2	60.	CHKRPL37A	110.	HROMA2
11.	ASYRVISP	61.	CHKRPL5G	111.	HROMA4A
12.	ATREGLOBIN	62.	CHKTUB4B	112.	HRSPRMI
13.	BABAPOE	63.	CHKY	113.	HS1D3HLH
14.	BATROX	64.	CHPPHYGEN	114.	HS2OXOC
15.	BCHEGLOBIN	65.	CHPPROTP1A	115.	HSABLGR1
16.	BOVANPA	66.	CHPPROTP1B	116.	HSALDCG
17.	BOVCOX7AL	67.	CHWEGLOBIM	117.	HSAPC3A
18.	BOVGAS	68.	CIGH	118.	HSAPOA2B
19.	BOVIAP	69.	CITEGLOBIM	119.	HSAPOAIA
20.	BOVIRBP	70.	CJAEGLOBIN	120.	HSAPOAIT
21.	BOVLHB	71.	CJINTERFG	121.	HSAPOC2G
22.	BOVLYSOZMA	72.	CL54K	122.	HSAQUPN2
23.	BOVLYSOZMB	73.	CMBGA2B2	123.	HSARYLA
24.	BOVLYSOZMC	74.	CMEGA2E2	124.	HSAT3
25.	BOVMYF5A	75.	CMU11711	125.	HSATPCP1
26.	BRHOX22	76.	CPPROT1GN	126.	HSATPCP2
27.	BRHOX34A	77.	CRASEQB	127.	HSB3A
28.	BRU12895	78.	CRUCH7AHYD	128.	HSBGPG
29.	BTATPAAA	79.	CRUGAD45A	129.	HSBSF2
30.	BTEBGL2	80.	DMPROTP1	130.	HSCBMYHC
31.	BTEBGL4	81.	DMU11712	131.	HSCD14G
32.	BTGL01	82.	DOGCOLIP	132.	HSCKBG
33.	BTHOR02	83.	DOGIL8T	133.	HSCOMT2
34.	BTKER6B	84.	ECGLOAP1	134.	HSCPH70
35.	BTSPDNA	85.	ECPZA2GL	135.	HSCSF1PO
36.	BTSVSP109	86.	ECZGL1	136.	HSCST3G
37.	BTTNP2G	87.	ECZGL2	137.	HSCYCLA
38.	BTU02285	88.	ELMETL	138.	HSCYP216
39.	CALEGLOBIM	89.	FDFELDI2	139.	HSCYP45C
40.	CBUEGLOBIM	90.	FDTNFA	140.	HSCYTOK17
41.	CCALAC	91.	GDMYF5G	141.	HSCYTOK20
42.	CCBEGLOBIM	92.	GGAC01	142.	HSDAO
43.	CCBEGLOBIN	93.	GGACTAC	143.	HSERPG
44.	CCGHG	94.	GGACTI	144.	HSFAU1
45.	CCGONBS1	95.	GGCALB	145.	HSFESFPS
46.	CCGONBS2	96.	GGCRYD1	146.	HSGCSFG
47.	CCGTHA1	97.	GGGL03	147.	HSGEBCMA
48.	CCT64CLU	98.	GGGNRHIA	148.	HSGGL2
49.	CEPPINS	99.	GGNFM1D	149.	HSGLA
50.	CHBLG	100.	GGPROP2	150.	HSGROW2

Table B.3: The BG dataset, part 1

Number	Locus	Number	Locus	Number	Locus
151.	HSGSTM4	201.	HUMATPGG	251.	HUMLHDC
152.	HSGTRH	202.	HUMBETGLOA	252.	HUMLUCT
153.	HSHMG17G	203.	HUMBETGLOB	253.	HUMLYTOXBB
154.	HSHNRNPA	204.	HUMBETGLOC	254.	HUMMCHEMP
155.	HSHOX51	205.	HUMBETGLOD	255.	HUMMET2
156.	HSHSC70	206.	HUMBETGLOE	256.	HUMMGPA
157.	HSIFNAR	207.	HUMBETGLOF	257.	HUMMIF
158.	HSIFNG	208.	HUMBETGLOG	258.	HUMMIS
159.	HSIL1AG	209.	HUMBETGLOH	259.	HUMMKXX
160.	HSIL1B	210.	HUMBETGLOI	260.	HUMNKG5PRO
161.	HSINT2	211.	HUMBETGLOJ	261.	HUMNTRI
162.	HSL7A	212.	HUMBETGLOK	262.	HUMNTRIII
163.	HSLCATG	213.	HUMBETGLOL	263.	HUMOSTP
164.	HSMECDAG	214.	HUMBETGLOM	264.	HUMPAP
165.	HSMED	215.	HUMBETGLON	265.	HUMPCI
166.	HSMT1H	216.	HUMBETGLOO	266.	HUMPEM
167.	HSNCAMX1	217.	HUMBETGLOP	267.	HUMPEPYYA
168.	HSNFM	218.	HUMBETGLOR	268.	HUMPF4V1A
169.	HSODCG	219.	HUMCBRG	269.	HUMPHOSA
170.	HSODF2	220.	HUMCHYMASE	270.	HUMPRCA
171.	HSPAT133	221.	HUMCP21OH	271.	HUMPREELAS
172.	HSPLAPL	222.	HUMCP21OHC	272.	HUMPRPHX
173.	HSPRB3L	223.	HUMCS1	273.	HUMRCC1
174.	HSPRB4S	224.	HUMCS3	274.	HUMREGHOM
175.	HSPSAG	225.	HUMCSN2A	275.	HUMROD1X
176.	HSRPII145	226.	HUMCSPA	276.	HUMRPIB2
177.	HSRPS7	227.	HUMCTLA1	277.	HUMRPS6B
178.	HSSAA1B	228.	HUMDEF5A	278.	HUMSEMI
179.	HSSHBG	229.	HUMDZA2G	279.	HUMSEMIIB
180.	HSSSPN1AG	230.	HUMELAFIN	280.	HUMSTATH2
181.	HSSURF3	231.	HUMEPOHYDD	281.	HUMTBGA
182.	HSTNFB	232.	HUMG0S24B	282.	HUMTCRBRA
183.	HSTPI1G	233.	HUMG0S8PP	283.	HUMTHROMA
184.	HSTUBAG	234.	HUMGAD45A	284.	HUMTNP2SS
185.	HSU01102	235.	HUMGARE	285.	HUMTPALBU
186.	HSU04357	236.	HUMGCK	286.	HUMTRHYAL
187.	HSU04636	237.	HUMGFP40H	287.	HUMTSHB2
188.	HSU05259	238.	HUMGHG	288.	HUMV2R
189.	HSU07807	239.	HUMGHN	289.	HYPSCGH
190.	HSU07983	240.	HUMGHV	290.	LAYEGLOBIN
191.	HSU08198	241.	HUMGSTM4A	291.	LEBGLOB
192.	HSU12421	242.	HUMHMGIY	292.	LNOEGLOBIN
193.	HSUBR	243.	HUMHOX13G	293.	MACHBGA1
194.	HSXBXVIII	244.	HUMHPARS1	294.	MACHBGA2
195.	HSZNGP1	245.	HUMIBP3	295.	MAMGLUTRA
196.	HUMADAG	246.	HUMIDS	296.	MAU09941
197.	HUMADPRF02	247.	HUMIGERA	297.	MFAPOA2A
198.	HUMAPEXN	248.	HUMIL2RGA	298.	MFAPOC3A
199.	HUMAPOCII	249.	HUMIL4A	299.	MMA2IXCOA
200.	HUMAPOE4	250.	HUMIRBPG	300.	MMACLGNA

Table B.4: The BG dataset, part 2

Number	Locus	Number	Locus	Number	Locus
301.	MMAPOG	351.	MMU02298	401.	MUSKE3A
302.	MMASM1G	352.	MMU02884	402.	MUSLMP7A
303.	MMCASEIB	353.	MMU04056	403.	MUSLTA
304.	MMCD24A	354.	MMU04827	404.	MUSMC26
305.	MMCFOS	355.	MMU07568	405.	MUSOGC
306.	MMCOL10A	356.	MMU07808	406.	MUSOGCA
307.	MMCRPG	357.	MMU09741	407.	MUSOGCB
308.	MMCYTOKNA	358.	MMU09964	408.	MUSPNMT
309.	MMDM2RR	359.	MMU10098	409.	MUSPROTA
310.	MMDNAASFA	360.	MMU11054	410.	MUSPROTEB
311.	MMEZGL	361.	MMU11713	411.	MUSPRPC2
312.	MMG37	362.	MMU12029	412.	MUSPRPMPB
313.	MMGBCRYA	363.	MMU12273	413.	MUSREGI
314.	MMGCCRYA	364.	MMU12559	414.	MUSREGII
315.	MMGFAPD	365.	MMU12560	415.	MUSROM1X
316.	MMGGLINE	366.	MMU12561	416.	MUSRPL30
317.	MMGK5	367.	MMU12562	417.	MUSRPS16
318.	MMH2D4Q5	368.	MMU12565	418.	MUSS100B
319.	MMHOX13	369.	MMU13921	419.	MUSSERPROA
320.	MMHOX24I	370.	MMU14421	420.	MUSSSATGN
321.	MMIL1BG	371.	MMVITRO	421.	MUSTCP
322.	MMIL3G	372.	MMVPREB2	422.	MUSTHY1
323.	MMIL5G	373.	MMVPREB3	423.	MUSTLAG
324.	MMKFGF	374.	MNKEGLOBIM	424.	MUSTSHBA2
325.	MMLDHAG	375.	MNKEGLOBIN	425.	MUSY1GLOB
326.	MMMMP9A	376.	MNKHGBGGAG	426.	OABBGLOB
327.	MMMRPS24	377.	MNPROP2	427.	OABCGLOB
328.	MMMTIX	378.	MUS8HS20	428.	OAINIGFII1
329.	MMMUPBS6	379.	MUSAP	429.	OAINTL3
330.	MMMYCL	380.	MUSAP5A	430.	OAKRT213
331.	MMNFMG	381.	MUSAPEX	431.	OALGB
332.	MMNMYC	382.	MUSAPOAII	432.	OAMETIAG
333.	MMNUCLEO	383.	MUSAPOIVA	433.	OAMTIB
334.	MMODC2	384.	MUSBMP4	434.	OAMTIC
335.	MMOESTEOP	385.	MUSBNP	435.	OAMTII
336.	MMPO	386.	MUSCD14A	436.	OAPROTP1
337.	MMPPSOMA	387.	MUSCRKNB	437.	OATRICH
338.	MMPROP2	388.	MUSCYTCOVB	438.	OCAPOAIG
339.	MMPROT1	389.	MUSENDOBA	439.	OCTNFBETA
340.	MMPROT2	390.	MUSFAUA	440.	OCUTGLOB
341.	MMSAP	391.	MUSFERHC	441.	OOGH
342.	MMSCIMIP	392.	MUSFKBP13X	442.	OPOP1
343.	MMSYNDE1A	393.	MUSGAD45	443.	ORAPROTP1A
344.	MMTHY1G	394.	MUSGFJE	444.	PAA1GL
345.	MMTLAC	395.	MUSGPOAD	445.	PAAFPG
346.	MMTNFBG	396.	MUSHBBH0	446.	PAU03674
347.	MMTROPIB	397.	MUSHBBH1	447.	PIGAPAI
348.	MMTUM198	398.	MUSHES1	448.	PIGAPCIII
349.	MMU01530	399.	MUSHOX35A	449.	PIGCNP
350.	MMU02278	400.	MUSIL1RN	450.	PIGFSB

Table B.5: The BG dataset, part 3

Number	Locus
451.	PIKEGLOBIN
452.	PPGLTG
453.	PPPROP2
454.	PPYPROP2
455.	PTAZGLO
456.	PTGGGLOG
457.	PTPPINS
458.	PTPROP2
459.	PVU11715
460.	QULNFLW
461.	QULTROPIA
462.	RABCRP
463.	RABDNP3AA
464.	RABSURFA
465.	RABTNF
466.	RATCYC
467.	RATCYSS
468.	RATDCCOVB
469.	RATGRG
470.	RATGROS
471.	RATGSTY
472.	RATIGFBA
473.	RATJE
474.	RATKALA
475.	RATLHB
476.	RATLITHOST
477.	RATLYSOZYM
478.	RATLYSZYM
479.	RATOSCAL
480.	RATPAP
481.	RATPAPIIB
482.	RATPPP
483.	RATPTBZR02
484.	RATRGPI
485.	RATRPIII
486.	RATSVSIV1
487.	RATTSHB
488.	RN0B2GLOB
489.	RN2BGLOB
490.	RN3B2GLOB
491.	RN3B3GLOB
492.	RN3BGLOB
493.	RNAC01
494.	RNAC02
495.	RNANTANT
496.	RNAPOEG
497.	RNCALBD9
498.	RNGAMT
499.	RNGMTG
500.	RNGROW3

Number	Locus
501.	RNHSC73
502.	RNLALB01
503.	RNLPKG
504.	RNODC
505.	RNP9KA
506.	RNPBPG
507.	RNPGMUT
508.	RNPROST22
509.	RNREVSV2A
510.	RNSDHG
511.	RNSVFG
512.	RNTHYCSG
513.	RNU03551
514.	RNU09193
515.	RNU12250
516.	RNUCPG
517.	RNVEGP1
518.	RNVEGP2C
519.	RRU04320
520.	S46763
521.	S49651
522.	S57980
523.	S62287
524.	S63697
525.	S66606
526.	S67057
527.	S69277
528.	S69278
529.	S69350
530.	SAIEGLOBIM
531.	SMIGCU
532.	SMNPRP1A
533.	SOEEGLOBIN
534.	SSBAT1G
535.	SSIKBAGE
536.	SSPINT1B
537.	TAPROTP1
538.	TARHBE
539.	TARHBG
540.	TFLPA2P1
541.	TFLPA2P2
542.	TFLPA2P3
543.	TFLPA2P4
544.	TGLHBB
545.	TIOPRP1A
546.	U00432
547.	U00433
548.	U00454
549.	U00938
550.	U01026

Number	Locus
551.	U01027
552.	U01844
553.	XELCRPGA
554.	XELCRYB
555.	XELHBBC
556.	XLACTA
557.	XLACTCAG
558.	XLBGL3
559.	XLGLAA1
560.	XLGLAA2
561.	XLK81A1G
562.	XLRPL14
563.	XLRPL1AG
564.	XLS8
565.	XLTUBAG
566.	XLXANF1A
567.	XTGLA
568.	XTGLAA
569.	XTGLB
570.	ZEFB2MICB

Table B.6: The BG dataset, part 4

Appendix C

Numerical Evidence for Tiling Conjectures

The following values were computed with the program `vaxmacs` by David B. Wilson. In many cases, the large numbers obtained made the presentation impractical so in most examples we have restricted ourselves to grids of size at most 12×12.

$k\backslash n$	2	3	4	5	6
0	$2^2 3^2$	$2^3 29^2$	$2^4 17^2 53^2$	$2^5 241^2 373^2$	$2^6 5^4 31^2 53^2 701^2$
1	2^3	$2^4 7^2$	$2^5 13^4$	$2^6 3^4 11^2 139^2$	$2^7 5^2 744397^2$
2	–	–	2^{10}	$2^{11} 31^2$	$2^{12} 3617^2$
3 –	–	–	–	–	2^{21}

Table C.1: Values of $S(n,k)$ for $n = \{2,\ldots,6\}$, $k \leq \lfloor \frac{n}{2} \rfloor$

$k\backslash n$	2	3	4	5	6
0	$2^2 3^2$	$2^3 29^2$	$2^4 17^2 53^2$	$2^5 241^2 373^2$	$2^6 5^4 31^2 53^2 701^2$
1	$2^1 3^2$	$2^2 29^2$	$2^3 17^2 53^2$	$2^4 241^2 373^2$	$2^5 5^4 31^2 53^2 701^2$
2	2^0	$2^1 11^2$	$2^2 3^6 13^2$	$2^3 3^4 3923^2$	$2^4 3^4 5^2 61^2 4133^2$
3	–	2^0	$2^1 3^4 5^2$	$2^2 5009^2$	$2^3 3^2 5^4 7^2 3187^2$
4	–	–	2^0	$2^1 197^2$	$2^2 233^2 347^2$
5	–	–	–	2^0	$2^1 3^2 7^2 43^2$
6	–	–	–	–	2^0

Table C.2: Number of tilings of the $2n \times 2n$ square grid with k edges removed from the lower left corner

In the following computation we have removed one edge from the step-diagonal. The value r, indicates which edge was removed (the lower left edge is at $r = 1$ and the last edge on the step-diagonal is at $r = n$).

$r\backslash n$	2	3	4	5	6
1	$2^1 3^2$	$2^2 29^2$	$2^3 17^2 53^2$	$2^4 241^2 373^2$	$2^5 5^4 31^2 53^2 701^2$
2	$2^1 3$	$2^2 11^1 29^1$	$2^3 3^3 13^1 17^1 53^1$	$2^4 3^2 241^1 373^1 3923^1$	$2^5 3^2 5^3 31^1 53^1 61^1 701^1 4133^1$
3	–	$2^2 19^1 29^1$	$2^3 5^1 7^1 17^2 53^1$	$2^4 3^1 5^1 29^1 137^1 241^1 373^1$	$2^5 5^3 31^1 53^1 701^1 3824333^1$
4	–	–	$2^3 17^1 53^1 373^1$	$2^4 3^4 5^1 97^1 241^1 373^1$	$2^5 5^3 11^1 31^1 37^1 53^1 701^1 6317^1$
5	–	–	–	$2^4 241^1 373^1 52861^1$	$2^5 5^3 7^1 31^1 53^1 227^1 701^1 2143^1$
6	–	–	–	–	$2^5 3^1 5^3 31^1 53^1 701^1 845099^1$

Table C.3: Number of tilings of the $2n \times 2n$ square grid with the rth edge removed from the step-diagonal

n	Number of tilings
0	1
1	2^2
2	$2^6 3^1$
3	$2^{11} 7^2$
4	$2^{12} 3^2 5^3 11^2$
5	$2^{18} 3^5 5^2 11^2 13^2$
6	$2^{18} 7^3 13^2 29^4 43^2$
7	$2^{36} 7^2 17^2 47^2 79^2 97^2$
8	$2^{24} 3^6 17^4 19^4 37^4 53^2 109^2$
9	$2^{34} 3^2 5^3 11^4 19^4 41^2 59^2 101^2 181^2 281^2$
10	$2^{30} 11^5 23^4 89^2 109^2 199^2 241^2 373^2 397^2 419^2$
11	$2^{51} 3^5 5^4 7^6 11^2 13^2 23^6 71^2 73^2 97^2 193^4 263^2$

Table C.4: Number of tilings of a $(2n+1) \times (2n+1)$ square grid with one square removed from the border

Bibliography

[1] A. Agresti. *Analysis of Ordinal Categorical Data.* Wiley Series in Probability and Mathematical Statistics. John Wiley and Sons, 1984.

[2] N. Alon, C. McDiarmid, and M. Molloy. Edge-disjoint cycles in regular directed graphs. *Journal of Graph Theory*, 22(3):231–237, 1996.

[3] S. F. Altschul, S. F. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[4] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[5] J. Bao. On the number of domino tilings of the rectangular grid, the holey square and related problems. *Final Report, Research Summer Institute at MIT*, 1997.

[6] F. Barahona, J. Fonlup, and A. Mahjoub. Compositions of graphs and polyhedra IV: Acyclic spanning subgraphs. *SIAM Journal of Discrete Math*, 7(3):390–402, August 1994.

[7] S. Batzoglou, B. Berger, D. J. Kleitman, E. S. Lander, and L. Pachter. Recent developments in computational gene recognition. *Documenta Mathematica, Extra Volume ICM 1998*, 1:649–658, 1998.

[8] G. Benson. Sequence alignment with tandem duplication. *Journal of Computational Biology*, 4(3):351–367, 1997.

[9] G. Bernardi. The isochore organization of the human genome. *Annu. Rev. Genet.*, 23:637–661, 1989.

[10] D. Bertsekas. *Dynamic Programming: Deterministic and Stochastic Models.* Academic Press, 1976.

[11] C. Burge. *Identification of genes in Human Genomic DNA.* PhD dissertation, Stanford University, Department of Mathematics, March 1997.

[12] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic DNA. *Journal of Molecular Biology*, 268:78–94, 1997.

[13] M. Burset and R. Guigó. Evaluation of gene structure prediction programs. *Genomics*, 34(3):353–367, 1996.

[14] R. J. Chapman. Personal communication.

[15] M. Chiara, O. Gozani, M. Bennett, P. Championarnaud, and L. Palandjian. Identification of proteins that interact with exon sequences, splice sites, and the branchpoint sequence during each stage of spliceosome assembly. *Molecular and Cellular Biology*, 16(7):3317–3326, 1996.

[16] M. Ciucu. Enumeration of perfect matchings in graphs with reflective symmetry. *Journal of Combinatorial Theory Ser. A*, 77(1):67–97, 1997.

[17] W. G. Cochran. Some methods of strengthening the common χ^2 tests. *Biometrics*, 10:417–451, 1954.

[18] H. Cohn. 2-adic behavior of numbers of domino tilings. *Electronic Journal of Combinatorics*, 6(1, R14), 1999.

[19] T. P. Coleman and J. R. Roesser. Secondary structure- an important CIS-element in rat Calcitonin/CGRP premessenger RNA splicing. *Biochemistry*, 37(45):15941–15950, 1998.

[20] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* MIT press, 1990.

[21] H. Cramer. *Mathematical Methods of Statistics.* Princeton University Press, 1946.

[22] M. E. Dalphin, P. A. Stockwell, W. P. Tate, and C. M. Brown. Transterm, the translational signal database, extended to include full coding sequences and untranslated regions. *Nucleic Acids Research*, 27:293–294, 1999.

[23] M. Dayhoff, R. M. Schwartz, and B. C. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5:345–352, 1978.

[24] R. W. Deed, M. Jasiok, and J. D. Norton. Attenuated function of a variant form of the helix-loop-helix protein, Id-3, generated by an alternative splicing mechanism. *FEBS Lett.*, 393:113–116, 1996.

[25] S. Dong and D. B. Searls. Gene structure prediction by linguistic methods. *Genomics*, 23:540–551, 1994.

[26] N. Elkies, G. Kuperberg, M. Larsen, and J. Propp. Alternating sign matrices and domino tilings. *Journal of Algberaic Combinatorics*, 1(2, 3):111–132 and 219–234, 1994.

[27] H. Gabow. Centroids, representations, and submodular flow. *Journal of Algorithms*, 18:586–628, 1995.

[28] M. S. Gelfand, A. A. Mironov, and P. A. Pevzner. Gene recognition via spliced sequence alignment. *Proc. Natl. Acad. Sci. USA*, 93:9061–9066, 1996.

[29] B. Grünbaum and G. C. Shephard. Pick's theorem. *American Mathematical Monthly*, 100:150–161, 1993.

[30] R. Guigó. Computational gene identification: an open problem. *Computers and Chemistry*, 21(4):215–222, 1997.

[31] D. Gusfield. *Algorithms on Strings, Trees and Sequences.* Cambridge, 1997.

[32] F. Harary, D. J. Klein, and T. P. Živković. Graphical properties of polyhexes: Perfect matching vector and forcing. *Journal of Mathematical Chemistry*, 6:295–306, 1991.

[33] R. C. Hardison, J. Oeltjen, and W. Miller. Long human-mouse sequence alignments reveal novel regulatory elements: A reason to sequence the mouse genome. *Genome Research*, 7:959–966, 1997.

[34] D. Haussler, D. Kulp, and M. Reese. A representative benchmark gene data set. `http://www-hgc.lbl.gov/inf/genesets.html`, 1996.

[35] J. Hawkins. A survey on intron and exon lengths. *Nucleic Acids Research*, 16:9893–9908, 1988.

[36] J. Henderson, S. Salzberg, and K. H. Fasman. Finding genes in DNA with a hidden Markov model. *Journal of Computational Biology*, 4(2):127–141, 1997.

[37] D. S. Hirschberg. Algorithms for the longest common subsequence problem. *J. ACM*, 24:664–675, 1977.

[38] X. Huang, M. D. Adams, H. Zhou, and A. Kerlavage. A tool for analyzing and annotating genomic sequences. *Genomics*, 46:37–45, 1997.

[39] X. Huang, R. Hardison, and W. Miller. A space-efficient algorithm for lcoal similarities. *Comput. Appl. Biosc.*, 6:373–381, 1990.

[40] G. B. Hutchinson and M. R. Hayden. SORFIND: A computer program that predicts exons in vertebrate genomic DNA. *BSC93*, 1993.

[41] W. Jockusch. Perfect matchings and perfect squares. *Journal of Combinatorial Theory Ser. A*, 67:100–115, 1994.

[42] P. John, H. Sachs, and H. Zernitz. Problem 5. domino covers in square chessboards. *Zastosowania Matematyki (Applicationes Mathematicae)*, XIX(3-4):635–641, 1987.

[43] S. Karlin and C. Burge. Dinucleotide relative abundance extremes: a genomic signature. *Trends Genet.*, 11(7):283–290, 1995.

[44] S. Karlin and I. Ladunga. Comparisons of eukaryotic genomic sequences. *Proceedings of the National Academy of Sciences*, 91(26):12832–12836, 1994.

[45] S. Karlin, I. Ladunga, and B. Blaisdell. Heterogeneity of genomes- measures and values. *Proceedings of the National Academy of Sciences*, 91(26):12837–12841, 1994.

[46] S. Karlin and J. Mrazek. What drives codon choices in human genes. *Journal of Molecular Biology*, 262(4):459–472, 1996.

[47] P. W. Kasteleyn. The statistics of dimers on a lattice, I: The number of dimer arrangements on a quadratic lattice. *Physica*, 27:1209–1225, 1961.

[48] R. Kenyon, D. B. Wilson, and J. Propp. A connection between perfect matchings and spanning trees of planar graphs. *preprint*, 1999.

[49] D. J. Klein and M. Randić. Innate degree of freedom of a graph. *Journal of Computational Chemistry*, 8:516–521, 1987.

[50] D. J. Klein, T. P. Živković, and R. Valenti. Topological long-range order for resonating-valence-bond structures. *Phys. Rev. B*, 43 A:723–727, 1991.

[51] B. F. Koop. Human and rodent DNA sequence comparisons: a mosaic model of genomic evolution. *Trends in Genetics*, 11(9):367–371, September 1995.

[52] B. F. Koop and L. Hood. Striking sequence similarity over almost 100 kilobases of human and mouse T-cell receptor DNA. *Nature Genetics*, 7:48–53, May 1994.

[53] M. Kozak. Possible role of flanking nucleotides in recognition of the AUG initiator codon by eukaryotic ribosomes. *Nucleic Acids Research*, 9:5233–5252, 1981.

[54] D. Kulp, D. Haussler, M. G. Reese, and F. H. Eeckman. A generalized hidden Markov model for the recognition of human genes in DNA. *Proceedings of the 4th Conference on Intelligent Systems in Molecular Biology*, 1996.

[55] E. S. Lander. The new genomics- global views of biology. *Science*, 274(5287):536–539, 1996.

[56] E. S. Lander and M. S. Waterman. *Calculating the Secrets of Life: Applications of the Mathematical Sciences in Molecular Biology.* National Academy Press, 1995.

[57] M. Laub and D. W. Smith. Finding intron/exon splice junctions using INFO, INterruption finder and organizer. *Journal of Computational Biology*, 5(2):307–321, 1998.

[58] H. Lodish, A. Berk, P. Matsudaira, D. Baltimore, L. Zipursky, and J. Darnell. *Molecular Cell Biology.* Scientific American Books, Inc., 1995.

[59] L. Lovász. *Combinatorial Problems and Exercises.* North Holland Publishing Company, 1979.

[60] C. L. Lucchesi and D. H. Younger. A minimax theorem for directed graphs. *J. London Mathematical Society*, 17:369–374, 1978.

[61] A. V. Lukashin and M. Borodovsky. GENEMARK.HMM: new solutions for gene finding. *Nucleic Acids Research*, 26(4):1107–1115, 1998.

[62] W. Makolowski, J. Zhang, and M. S. Boguski. Comparative analysis of 1,196 orthologous mouse and human full-length mRNA and protein sequences. *Genome Research*, 6(9):846–857, September 1996.

[63] A. J. McCullough and S. M. Berget. G triplets located throughout a class of small vertebrate introns enforce intron borders and regulate splice site selection. *Molecular and Cellular Biology*, 17(8):4562–4571, 1997.

[64] A. Mironov, M. A. Roytberg, P. A. Pevzner, and M. S. Gelfand. Performance-guarantee gene predictions via spliced alignment. *Genomics*, 51:332–339, 1998.

[65] E. W. Myers and W. Miller. Optimal alignments in linear space. *Comp. Appl. Biosciences*, 4:11–17, 1988.

[66] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48:443–453, 1970.

[67] J. C. Oeltjen, T. M. Malley, D. M. Muzny, W. Miller, R. A. Gibbs, and J. W. Belmont. Large-scale comparative sequence analysis of the human and murine Bruton' Tyrosine Kinase loci reveals conserved regulatory domains. *Genome Research*, 7:315–329, 1997.

[68] L. Pachter. Combinatorial approaches and conjectures for 2-divisibility problems concerning domino tilings of polyominoes. *Electronic Journal of Combinatorics*, 4(1, R29), 1997.

[69] L. Pachter, S. Batzoglou, E. Banks, W. Beebee, N. Feamster, V. I. Spitkovsky, T. Tyan, B. Wallis, E. S. Lander, D. J. Kleitman, and B. Berger. Unpublished manuscript. 1999.

[70] L. Pachter and P. Kim. Forcing matchings on square grids. *Discrete Mathematics*, 190:287–294, 1998.

[71] J. Propp. Twenty open problems in enumeration and matchings. *Preprint*, 1996.

[72] M. Randić and D. J. Klein. Kekule valence structures revisited. In N. Trinasjstic, editor, *Innate Degrees of Freedom of π-electron Couplings in Mathematical and Computational Concepts in Chemistry.* Wiley, New York, 1986.

[73] R. Rieger, A. Michaelis, and M. M. Green. *Glossary of Genetics - Classical and Molecular 5th ed.* Springer-Verlag, 1991.

[74] I. Rigoutsos and A. Califano. Searching in parallel for similar strings. *Computational Science and Engineering*, 1(2):60–75, 1994.

[75] H. Sachs and H. Zernitz. Remark on the dimer problem. *Discrete Applied Math*, 51:171–179, 1994.

[76] N. Saldanha. Personal communication.

[77] S. L. Salzberg. Decision trees and markov chains for gene finding. In S. L. Salzberg, D. B. Searls, and S. Kasif, editors, *Computational Methods in Molecular Biology*, number 32 in New Comprehensive Biochemistry, chapter 10, pages 187–203. Elsevier, 1998.

[78] L. K. Sanders. A proof from graph theory for a Fibonacci identity. *Fibonacci Quarterly*, 28(1):48–55, 1990.

[79] P. A. Sharp. Split genes and RNA splicing. *Cell*, 77(6):805–815, 1994.

[80] P. A. Sharp and C. B. Burge. Classification of introns: U2-type or U12-type. *Cell*, 91:875–879, 1997.

[81] A. F. A. Smit. Origin of interspersed repeats in the human genome. *Curr. Opin. Genet. Devel.*, 6(6):743–749, 1996.

[82] E. Snyder and G. Stormo. Identification of protein coding regions in genomic DNA. *Journal of Molecular Biology*, 248:1–18, 1995.

[83] V. V. Solovyev, A. A. Salamov, and C. B. Lawrence. Identfication of human gene structure using linear discriminant functions and dynamic programming. *Proceedings of ISMB*, pages 367–375, 1995.

[84] V. I. Spitkovsky. Dictionary approaches to gene recognition. Master's thesis, MIT, 1999.

[85] R. Stanley. On dimer coverings of rectangles of fixed width. *Discrete Applied Mathematics*, 12(1):81–87, 1985.

[86] S. Sze and P. A. Pevzner. Las Vegas algorithms for gene recognition: Suboptimal and error-tolerant spliced alignment. *Journal of Computational Biology*, 4(3):297–309, 1997.

[87] S. Vajda. *Fibonacci and Lucas numbers, and the golden section. Theory and Applications. Ellis Horwood Series: Mathematics and its Applications.* Ellis Horwood Ltd., 1989.

[88] M. S. Waterman. *Introduction to Computational Biology: Maps, sequences and genomes.* Chapman and Hall, 1998.

[89] J. Watson, N. Hopkins, J. Roberts, J. Steitz, and A. Weiner. *Molecular Biology of the Gene.* The Benjamin/Cummings Publishing Company Inc., 1987.

[90] M. Werman and D. Zeilberger. A bijective proof of Cassini's Fibonacci identity. *Discrete Math*, 58(1):109, 1986.

[91] Y. Xu, R. J. Mural, M. Shah, and E. C. Uberbacher. Recognizing exons in genomic sequences using GRAIL II. In J. Setlow, editor, *Genetic Engineering: Principles and Methods*, volume 16. Penum Press, 1994.

[92] F. Zhang, X. Guo, and R. Chen. The z-transformation graphs of perfect matchings of hexagonal system. *Discrete Math*, 72:405–415, 1988.

[93] F. Zhang and X. Li. Hexagonal systems with forcing edges. *Discrete Math*, 140:253–263, 1995.

[94] Z. Zhang, W. R. Pearson, and W. Miller. Aligning a DNA sequence with a protein sequence. *Journal of Computational Biology*, 4(3):339–349, 1997.

[95] `dbEST. http://www.ncbi.nlm.nih.gov/dbEST/index.html`, 1998.

[96] `OWL. http://bmbsgi11.leeds.ac.uk/bmb5dp/owl.html`, 1998.

[97] `RepeatMasker. http://ftp.genome.washington.edu/cgi-bin/RepeatMasker`, 1998.