EXON AND INTRON DETECTION IN HUMAN GENOMIC DNA

By

JAMES KEITH MILLER

A dissertation submitted in partial fulfillment of
the requirements for the degree of

DOCTOR OF PHILOSOPHY

WASHINGTON STATE UNIVERSITY
Department of Pure and Applied Mathematics

MAY 2005

To the Faculty of Washington State University:

The members of the Committee appointed to examine the dissertation of JAMES KEITH MILLER find it satisfactory and recommend that it be accepted.

Chair

ACKNOWLEDGEMENTS

I would like to express my thanks to my advisor, Professor Richard Gomulkiewicz for his support, both financial and otherwise, and to my other committee members, Jave Pascual, and Robert Dillon for their suggestions throughout. I am also grateful to the faculty, staff, and students of the Department of Pure and Applied Mathematics at Washington State University and at the University of Arizona; they have all helped me along the way in more ways than even I realize. A special thanks to Andy Felt for his countless tips on life, love, and LaTeX (or something like that), as well as a myriad of other computer related issues, and to Indika Rajapakse for all the good talks. The U of A crew of Lou Rossi, Andy Long, and Dan McGee made Tucson a wonderful place. Thanks also to Lior Pachter, University of California at Berkeley, for his talks with me on hidden Markov models. Kevin Cooper has been an invaluable source of information both computing and otherwise, and I am deeply indebted to him. Thanks also to my father, Keith Miller, for helping me through the earlier stages of graduate school and for mulling over issues on my dissertation. Of course my entire family has been of tremendous support through the entire process, and they have my undying gratitude; a special thanks to Sonja for her countless edits of my dissertation – thanks HBW.

EXON AND INTRON DETECTION IN HUMAN

GENOMIC DNA

Abstract

by James Keith Miller, Ph.D.

Washington State University

May 2005

Chair: Richard Gomulkiewicz

The exponential growth of raw genomic data demands a shift from biological methods of gene annotation to more computational and mathematical methods. We present a novel computational approach using likelihood ratios which we call the multi-window method. DNA n-tuple frequencies are collected from a training set of known exons and introns. Likelihood ratios, based on these n-tuple frequencies within a window of nucleotides, are used to predict the position of a nucleotide. This position either indicates the location within a codon for exon nucleotides, or indicates that the nucleotide is from an intron.

We also compare the sensitivity and specificity of this method with a simple hidden Markov model which captures many of the same features as our multi-window method.

Contents

1 Introduction **1**

2 Exon and Intron Detection using Likelihood Ratios **8**

2.1 Abstract 8

2.2 Introduction 8

2.3 Biological Background 11

2.3.1 Transcription 12

2.3.2 Splicing 17

2.3.3 Alternative Splicing 19

2.3.4 Translation 22

2.4 Current Methods 23

2.4.1 Biological Methods 25

2.4.2 Computational Methods 30

2.5 New Method 46

2.6 Discussion 59

3 Sensitivity and Specificity of Exon Detection using Likeli-

hood Ratios **73**

3.1 Abstract 73

3.2 Introduction 74

3.3 Biological Background 76

3.4 Methods 82

3.5 Results 90

3.6 Discussion 97

4 Hidden Markov Models as a Means of Analyzing Likelihood Ratios **102**

4.1 Abstract 102

4.2 Introduction 103

4.2.1 Overview of Markov Chain Models and Hidden Markov Models 103

4.3 Methods 107

4.3.1 Viterbi Algorithm 110

4.3.2 Forward and Backward Algorithms 115

4.4 Results 118

4.5 Discussion 122

5 Exon Detection using Likelihood Ratios with the Incorporation of GeneSplicer **127**

5.1 Abstract 127

5.2 Introduction 128

5.3 Biological Background 129

5.4 Overview of GeneSplicer 135
5.4.1 Maximal Dependence Decomposition 135
5.4.2 Sequence Statistic in Larger Windows 138
5.4.3 Local Score Optimality Feature 139
5.5 Methods 139
5.6 Results 140
5.7 Discussion 142

List of Figures

1.1 Number of nucleotides and sequences in GenBank. http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html. Revised: May 4, 2004. Used with permission from the National Center for Biotechnology Information. 2

2.1 Number of nucleotides and sequences in GenBank. http://www.ncbi.nlm.nih.gov/Genbank/genbankstats.html. Revised: May 4, 2004. Used with permission from the National Center for Biotechnology Information. 10

2.2 Transcription and Translation – schematic. ©1997 by John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc. 13

2.3 Consensus sequences for regions of an intron. E_k denotes the k^{th} exon. | denotes an exon/intron or intron/exon boundary. R - a puRine (an A or G base), Y - a pYrimidine (a C or T/U), N - aNy nucleotide. The subscripts give the percentage occurrences of these bases at the indicated position relative to the splice sites. Subscripts of 100 are rounded, and there are many known exceptions (and many more may be found when introns are searched for without assuming that they start and end with these sequences. See http://www.ebi.ac.uk/asd/altextron/pre-release-dist-data.html for current percentages). 18

2.4 Splicing. Used by permission of Oxford University Press – Free permission. 20

2.5 Transcription and Translation – diagram. ©1997 by John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc. 24

2.6 Cutting of DNA sequence by the restriction enzyme EcoRI ("echo-R-one"). The boldface nucleotides represent the six base-pair "restriction site" for EcoRI 27

2.7 A perceptron with an input of dimension n and threshold T. Used by permission of the author – http://www.iiit.ac.in/~vikram/nn_intro.html. 35

2.8 Vertical axis – relative frequency of intron triplets, horizontal axis – the 64 triplets in alphabetical order. Frequencies of all six frames are superimposed with vertical line indicating range. Note the similarity of frequencies in all frames. 52

2.9 Idealized exon, intron, exon, intron sequence. Plots a, b, and c show the lambdas corresponding to starting at position one, two, and three respectively. Exons at positions 1 - 30 and 81 - 110, introns elsewhere. 58

3.1 Consensus sequences for regions of an intron. E_k is the k^{th} exon of the gene. | denotes an exon/intron or intron/exon boundary. R - a puRine (an A or G base), Y - a pYrimidine (a C or T/U), N - aNy nucleotide. The subscripts give the percentage occurrences of these bases. Subscripts of 100 are rounded, and there are many known exceptions (and many more may be found when introns are searched for without assuming that they start and end with these sequences. See http://-www.ebi.ac.uk/asd/altextron/pre-release-dist-data.html for current percentages of donor/acceptor splice sites). 79

3.2 Vertical axis – relative frequency of intron triplets, horizontal axis – the 64 triplets in alphabetical order. Frequencies of all six frames are superimposed with vertical line indicating range. Note the similarity of frequencies in all frames. 84

3.3 Length distribution of exons in our data set within intron containing genes. Exons of length greater than 400 nucleotides, which represent approximately 4 percent of these exons, are not included in the graph. 92

3.4 Ordered ranking of sensitivities for individual genes for first base of codons. 94

3.5 Ordered ranking of specificity for individual genes for first base of codons. 94
3.6 Ordered ranking of sensitivities for individual genes for intron bases. 95
3.7 Ordered ranking of specificity for individual genes for intron bases. 95

4.1 Sensitivity of position 1 detection with altered bases. See text for the ordering of the plotted points. 123

5.1 Consensus sequences for regions of an intron. E_k is the k^{th} exon of the gene. | denotes an exon/intron or intron/exon boundary. R - a puRine (an A or G base), Y - a pYrimidine (a C or T/U), N - aNy nucleotide. The subscripts give the percentage occurrences of these bases. Subscripts of 100 are rounded, and there are many known exceptions (and many more may be found when introns are searched for without assuming that they start and end with these sequences. See http://-www.ebi.ac.uk/asd/altextron/pre-release-dist-data.html for current percentages of donor/acceptor splice sites). 132
5.2 Frequency distribution of distances to donor splice sites 141
5.3 Frequency distribution of distances to acceptor splice sites . . 142

List of Tables

2.1 Codons, alphabetical ranking and frequency ranking. a) ranked alphabetically b) ranked by frequency. 54

2.2 Intron triplets, alphabetical ranking and frequency ranking. a) ranked alphabetically b) ranked by frequency. 55

2.3 Lambdas, alphabetical ranking and value ranking. a) ranked alphabetically b) ranked lowest to highest. 56

3.1 EV_{jk} is the expected value of $\log \frac{L(H_j|Data)}{L(H_k|Data)}$ where Data is a triplet from frame j. $j = 1, 2, 3, i \quad j \neq k$. 88

3.2 Sensitivity and specificity of method detecting various categories of nucleotides. 93

3.3 Correlation coefficients for characteristic vs. sensitivity of first base of codon detection. 96

3.4 Correlation coefficients for characteristic vs. specificity of first base of codon detection. 96

4.1 Emission probabilities = $b_j(k)$. Each value under state j in the row $N_1N_2N_3$ gives $P(N_3|$ prior 2 bases are N_1N_2, (state of N_3) = $j)$. Of particular note are the three lowest values which correspond to the stop codons in the reading frame (TAA, TAG, TGA – state 3.) Note that the probabilities which must sum to one are $\sum_{N_3} P(N_3|$ prior 2 bases are N_1N_2, (state of N_3) = $j)$, and not the probabilities along a given row. 111

4.2 Ranked sensitivity and specificity of the three methods: vit (Viterbi method), fb (forward/backward method), mw (multi-window method). In the Exon columns the highest and lowest values for each of the three positions of a codon are shown. The median values are calculated by finding the sensitivities and specificities for each of the 4074 sequences and ranking them from lowest to highest, and then finding the median sensitivities and specificities. The mean values are found by considering all 27,085,898 nucleotides in the 4074 sequences, and finding the sensitivities and specificities of the methods on these nucleotides. 119

5.1 Sensitivity and specificity of the multi-window method (mw) with and without the use of GeneSplicer. 143

Dedication

This dissertation is dedicated to my immediate family – both pre-Sonja, and post-Kianna. Thank you all for your unyielding support through not only this dissertation, but through the much bigger issues of life itself. I love you all dearly, and without you this would be a hollow accomplishment. Danica's desire to see me "hydrate" since a time when she could barely speak has been a constant impetus to finish while keeping me grounded with life's simple pleasures.

Chapter 1

Introduction

As of October 2004, there were more than 38 million sequence entries totaling over 43 billion base pairs entered into GenBank [Benson et al., 2000] (http://www.ncbi.nlm.nih.gov/Web/Newsltr/SummerFall04/GBrel.html). This staggering amount of data has been doubling almost every year for the last 20 years, and shows no signs of slowing - figure 1.1 At the same time, however, computational power (CPU speed and memory size) grows at a slower rate [Livstone et al., 2003]. Thus, to keep pace with the analysis of this data, new techniques are needed. For a review of current methods, see Mathe *et al.* [Mathe et al., 2002] and the website of Wentian Li (http://linkage.rockefeller.edu/wli/gene/).

Significant biological questions may be answered using this sequence data, including questions regarding phylogeny [Townsend et al., 2004] (the evolutionary history of an organism, and thus its relatedness to other organisms), population genetics [Won and Hey, 2005], and medical research

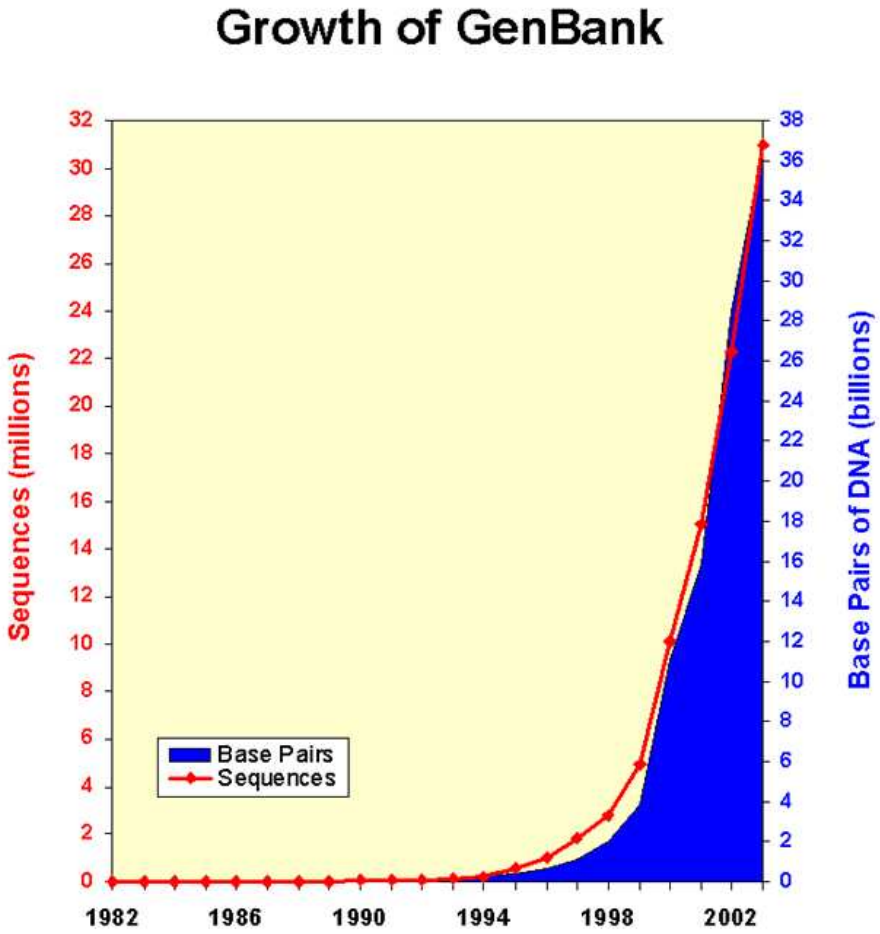

Figure 1.1: Number of nucleotides and sequences in GenBank. http://www.-ncbi.nlm.nih.gov/Genbank/genbankstats.html. Revised: May 4, 2004. Used with permission from the National Center for Biotechnology Information.

[Sridhar, 2001]. Often, but not always, a first step in the analysis of this raw genomic data is to computationally identify biologically significant regions of these sequences. While there are many subregions of interests, (telomeres and centromeres for example may play roles in aging and chromosome replication, respectively) the genes of an organism are the key players since they contain the genetic code which instructs the cell how to make the building blocks of the cell itself – the proteins. This dissertation presents a novel approach to locating the exons, or the regions which are translated into proteins, of a gene.

In Chapter 2, after introducing the relevant biological background, summaries of both biological annotation methods and the major current computational annotation programs are given. Next, we present a novel computational approach using likelihood ratios. DNA n-tuple frequencies are collected from a training set of known exons and introns. Likelihood ratios, based on these n-tuple frequencies within a window of nucleotides, are used to predict the position of a nucleotide. This position either indicates the location within a codon for exon nucleotides, or indicates that the nucleotide is from an intron.

Chapter 3 gives a more thorough description of the method introduced in Chapter 2. Next, the results from this method, i.e. the sensitivities and specificities (accuracies), are enumerated for n-tuples of length one, two and three. Finally, an extension (the multi-window method) is introduced which uses triplet frequencies and three consecutive overlapping windows of DNA data. This extension increased the method's overall sensitivity and specificity. The

multi-window method, unlike many of the more widely used current methods, uses only local information to predict the position of a nucleotide. While this may lower the method's overall sensitivity and specificity, it still performs reasonably well as a stand alone method. The results from our multi-window method may also be used as additional evidence of the position of a nucleotide when used in conjunction with another method. Another possible use for our multi-window method is finding regions that were exonic in the past, but no longer produce proteins. These so-called "pseudogenes" do not have all of the DNA sequence information contained in a true gene, and thus methods which rely on this entire set of true gene information will miss these pseudogenes. While missing pseudogenes certainly can not be considered a fault of a method which seeks to identify currently active genes, there are instances where finding pseudogenes is of interest [Zhang and Gerstein, 2003], [Zhang et al., 2003].

A simple hidden Markov model is developed in Chapter 4. It incorporates many of the same features as the multi-window method, and is used to give a more rigorous mathematical framework to the analysis of the sensitivity and specificity of this latter method. Under this hidden Markov model framework, both the Viterbi algorithm and the forward-backward algorithms are employed to make exon/intron predictions. Then the sensitivities and specificities of the multi-window, the Viterbi and the forward-backward methods are compared.

Finally, in Chapter 5, we attempt to increase the overall sensitivity and specificity of the multi-window method by using a splice site prediction

method – GeneSplicer. Under ideal conditions the multi-window method will correctly predict the position of a base when the three overlapping windows are within the same region (exon or intron) as the base itself. As the leading tail of the windows move through a splice site (a change from exon to intron or visa versa) though, this tail contains data from a different region than the nucleotide in question. This causes the predictions to become increasingly poor until the windows move entirely into the new region. This inability to resolve splice sites was partially alleviated by incorporating the splice site predictions made by GeneSplicer.

Bibliography

[Benson et al., 2000] Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., Rapp, B. A., and Wheeler, D. L. (2000). GenBank. *Nucleic Acids Res*, 28(1):15–18.

[Livstone et al., 2003] Livstone, M. S., van Noort, D., and Landweber, L. F. (2003). Molecular computing revisited: a Moore's Law? *Trends Biotechnol*, 21(3):98–101.

[Mathe et al., 2002] Mathe, C., Sagot, M.-F., Schiex, T., and Rouze, P. (2002). Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res*, 30(19):4103–4117.

[Sridhar, 2001] Sridhar, G. R. (2001). Impact of human genome project on medical practice. *J Assoc Physicians India*, 49:995–998. Historical Article.

[Townsend et al., 2004] Townsend, T., Larson, A., Louis, E., and Macey, J. (2004). Molecular phylogenetics of squamata: the position of snakes, amphisbaenians, and dibamids, and the root of the squamate tree. *Syst Biol*, 53(5):735–757.

[Won and Hey, 2005] Won, Y.-J. and Hey, J. (2005). Divergence population genetics of chimpanzees. *Mol Biol Evol*, 22(2):297–307.

[Zhang and Gerstein, 2003] Zhang, Z. and Gerstein, M. (2003). Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes. *Nucleic Acids Res*, 31(18):5338–5348.

[Zhang et al., 2003] Zhang, Z., Harrison, P. M., Liu, Y., and Gerstein, M. (2003). Millions of years of evolution preserved: a comprehensive catalog of the processed pseudogenes in the human genome. *Genome Res*, 13(12):2541–2558.

Chapter 2

Exon and Intron Detection using Likelihood Ratios

2.1 Abstract

In order to distinguish between the exon and intron regions within genes in the human genome, nucleotide n-tuple frequencies in these two regions are analyzed. The differences in these frequencies gives an exploitable method, using likelihood ratios, to characterize these regions, and to find the correct "reading frame" within the exons.

2.2 Introduction

As of October 2004, there were more than 38 million sequence entries totaling over 43 billion base pairs entered into GenBank [Benson et al., 2000] (http:/-

/www.ncbi.nlm.nih.gov/Web/Newsltr/SummerFall04/GBrel.html). This staggering amount of data has been doubling approximately yearly over the last 20 years, and shows no signs of slowing - figure 2.1. At the same time, computational power (CPU speed and memory size) is growing at a slower rate [Livstone et al., 2003]. To keep pace with the analysis of this data, new analytical techniques are needed.

Many biological questions can be addressed using genome sequence data. Some examples include questions regarding phylogeny [Townsend et al., 2004] (the evolutionary history of an organism, and thus its relatedness to other organisms), population genetics [Won and Hey, 2005], and medical research [Sridhar, 2001]. Often, but not always, a first step in the analysis of raw genomic data is to identify biologically significant regions of these sequences. There are many subregions of interests. For example, telomeres on the ends of chromosomes may play a role in aging and centromeres play a role in chromosome replication. However, the genes of an organism are the key players since they contain the genetic code which instructs each cell how to make the building blocks of the cell itself – the proteins.

This chapter begins with an overview of the pertinent biology and then gives a discussion of current methods of identifying exons and introns. Finally, a new method of exon and intron detection, using likelihood ratios, is introduced.

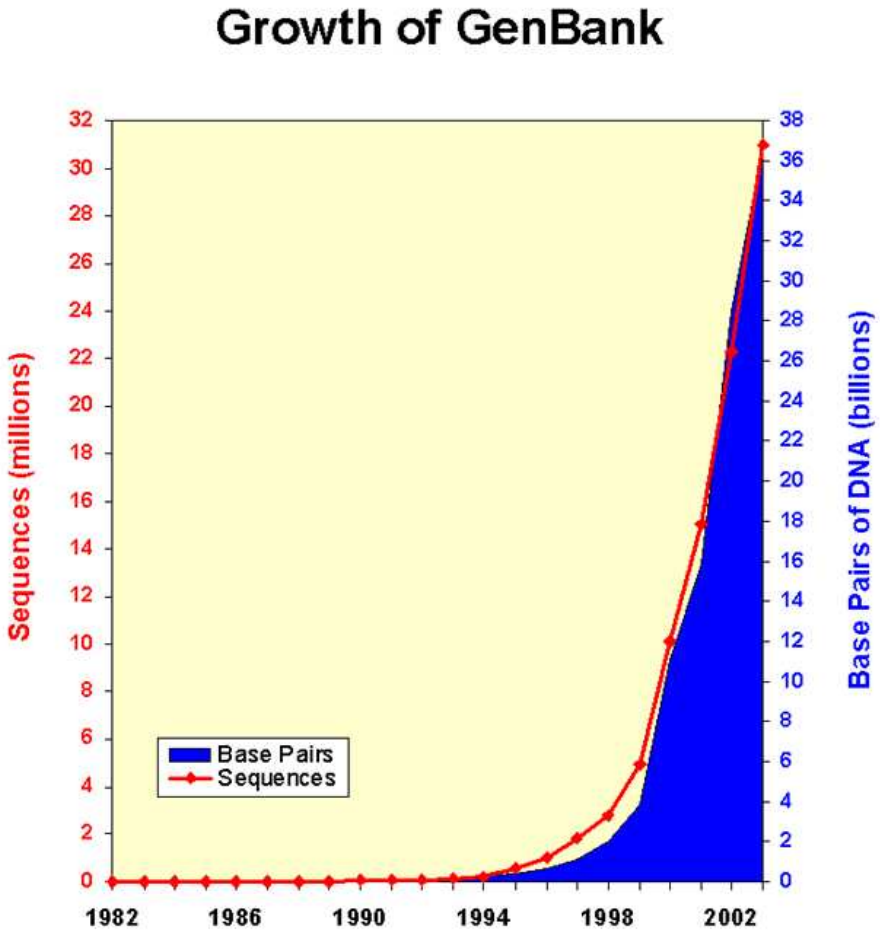

Figure 2.1: Number of nucleotides and sequences in GenBank. http://www.-ncbi.nlm.nih.gov/Genbank/genbankstats.html. Revised: May 4, 2004. Used with permission from the National Center for Biotechnology Information.

2.3 Biological Background

This section contains a summary of the pertinent biology. More detailed accounts of the basic processes of transcription, slicing, and translation can be found in any of the following general texts of genetics: [Lewin, 1994], [Fairbanks and Anderson, 1999] and [Snustad et al., 1997].

Human chromosomes are composed of tightly coiled threads of deoxyribonucleic acid (DNA) [Watson and Crick, 1953] and associated protein molecules which aid in the structured packing of the DNA. The DNA itself is often compared to a twisted ladder with the sides of the ladder being the sugar-phosphate backbone of the DNA, and the rungs being the two complementary nucleotides that bind to one another - one from each of the two strands of DNA. A single strand of DNA may be thought of as a sequence of four nucleotides: adenine (A), cytosine (C), guanine (G) and thymine (T). The nucleotides that bind to one another to form the "rungs" are known as complementary pairs: A binds with T by a double hydrogen bond, and C binds with G by a triple hydrogen bond. The A and G nucleotides are known as purines and are larger than the C and T nucleotides, which are known as pyrimidines. Thus all "rungs" are composed of one purine and one pyrimidine. The DNA is always read by the cell machinery in the same orientation. That is, the sequence AATCGTA of nucleotides (or bases) along a strand of DNA would always be read in the order indicated above or in the reverse as ATGCTAA, but not in both orders. The end of the sequence where the reading starts is known as the 5' end, and the other is the 3' end.

The complementary strand always has the reverse orientation. Thus if one strand of a chromosome had the sequence 5'- AATCGTA - 3', then this would be bound to the sequence 3' - TTAGCAT - 5' as shown below:

5' - AATCGTA - 3'

3' - TTAGCAT - 5'

The genes within the DNA (which in higher eukaryotes comprise only a small percentage of the entire genome - the entire DNA sequence of an organism - which in humans is some three billion nucleotides long) are the genetic code used by the cell to make proteins. A typical gene is a few thousand bases long. There are many genes on both strands of the DNA of a chromosome (humans have 23 pairs of chromosomes and somewhere on the order of 30,000 genes), but if a particular gene were on the top strand of the above diagram, it would be read from left to right, whereas if it were on the bottom, it would be read from right to left.

2.3.1 Transcription

An initial stage of protein synthesis is the transcription of the DNA into messenger RNA (mRNA). This mRNA will transfer the information from the DNA in the nucleus of the cell out into the cytoplasm of the cell where the protein is synthesized. See figure 2.2 [Snustad et al., 1997, Figure 11.28] for a schematic of transcription and translation (explained below), and figure 2.5 [Snustad et al., 1997, Figure 11.5] for a diagram of them.

RNA is a molecule very similar in structure to DNA, except that thymine

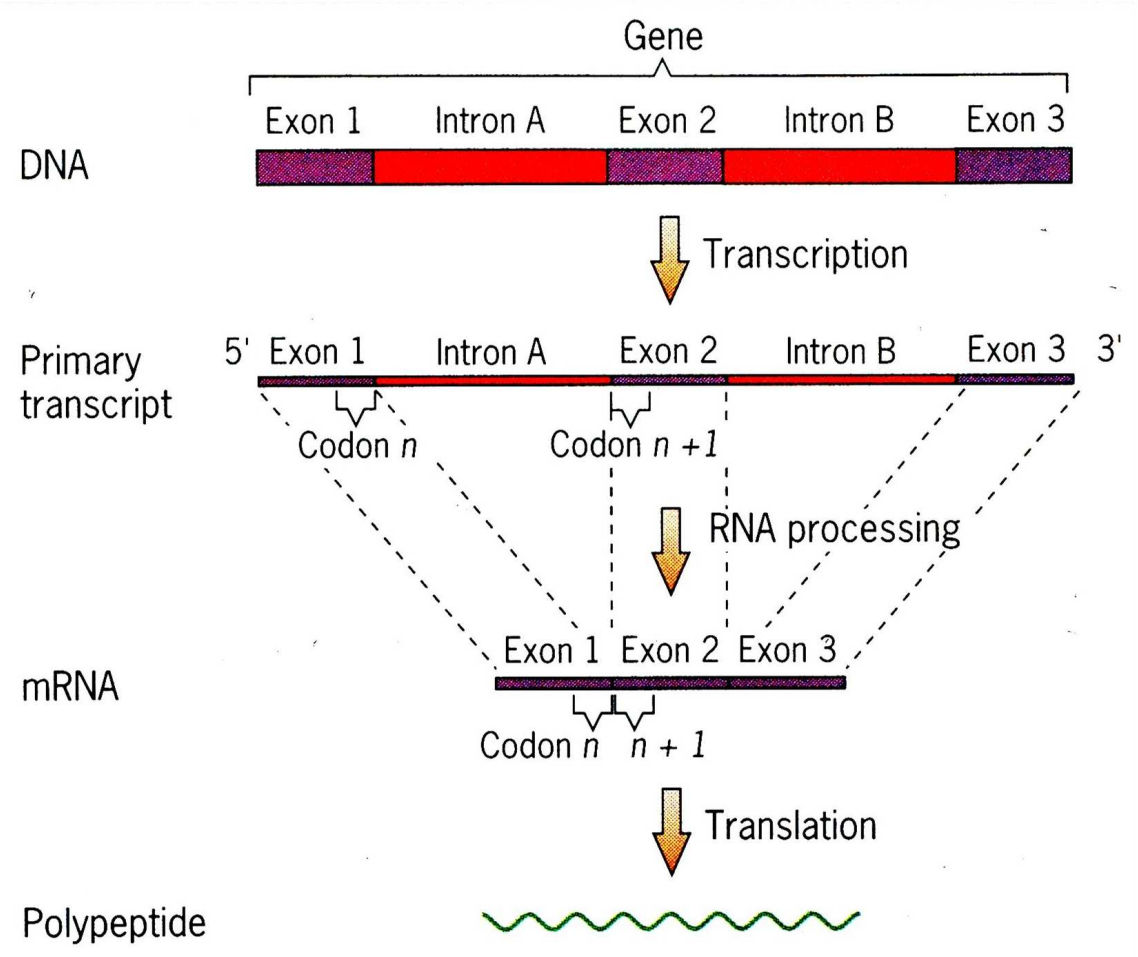

Figure 2.2: Transcription and Translation – schematic. ©1997 by John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc.

is replaced by the pyrimidine uracil (U) and RNA uses the sugar ribose instead of deoxyribose for its sugar-phosphate backbone. If, in the above example, the bottom strand were part of a gene it would be called the sense strand for this portion of the double helix. The top strand would be used by an enzyme (a catalytic protein) known as RNA polymerase II to synthesize the mRNA; thus this top strand is known as the template or antisense strand. The bottom strand, the non-template strand, would have the sequence in the orientation in which, by convention, genes are reported. The growing mRNA would be synthesized in the 5' to 3' direction as the reverse complement of the top strand, so would be synthesized from "right to left" here:

5' - ... AATCGTA ... - 3' The DNA template strand

3' - ...← UAGCAU ... - 5' The newly synthesized mRNA (pre-mRNA)

In order for RNA polymerase II to synthesize mRNA, the double helix of the DNA must be locally unwound in the area around the transcription initiation site. This unwinding, in eukaryotes, is performed not by the RNA polymerase II, but by other proteins associated with transcription initiation. Regulatory elements are located to either side of the transcription initiation site. These are sequences which can bind to various proteins which then increase (up regulate) or reduce transcription of the gene, thereby controlling the amount of the corresponding protein in the cell. By definition, the regulatory element adjacent and upstream (to the 5' side) to the initiation site

is known as the promoter. The promoter is comprised of various promoter elements known as the TATA box (or the Goldber-Hogness box after its discoverers), the CAAT box, and the GC box. Although no single promoter element is found in every promoter, and their actual sequences vary, their position and sequences are as follows in many genes. The TATA box has the consensus sequence 5'-TATAAA-3', and is located 30 bases upstream from the transcription initiation site. This sequence appears to have little effect on gene expression, but influences where transcription starts and thus serves as a reference point for the protein complex involved in translation. The CAAT box has a consensus sequence of 5'-GGCCAATCT-3' and is often 75 bases upstream of the initiation site. This sequence does have a significant effect on gene expression, with mutations down regulating gene expression. The CG box has a consensus sequence of 5'-GGGCGG-3' and is found 90 bases upstream of the initiation site. This promoter element can face towards or away from the gene, and thus can also appear as 5'-CCGCCC-3'; there are often multiple copies of this promoter element.

In addition to these promoter elements, there are other regulatory elements known as enhancer and silencer elements. As the names imply, these elements up or down regulate gene expression. Although most are located upstream of the initiation site, they can be found downstream as well. These elements tend to be further away from the gene – often more than 1,000 bases from the promoter. Regulatory proteins bind to these sites, often causing the DNA to fold, bringing the regulatory element bound protein(s) adjacent to the start of transcription which then enhances or represses gene expression.

A gene's promoter is not recognizable by the RNA polymerase, and thus various transcription factors (TFs) are necessary for mRNA synthesis – a typical example follows. First TFIID binds to the TATA box. TFIIB then binds to this complex and brings in the RNA polymerase and TFIIF. Next TFIIE and TFIIH bind to the complex, and transcription begins. The rate of mRNA production may then be regulated by various enhancer or silencer elements with their bound proteins.

The RNA polymerase synthesizes the mRNA from the point of transcription initiation through all the nucleotides which will be translated into proteins, and past. The mechanism for termination of mRNA synthesis is poorly understood. A highly conserved sequence, known as the poly (A) signal (see below), of AAUAAA is synthesized towards the end of the gene. Eleven to 30 bases past this sequence, the growing mRNA is cleaved. The RNA polymerase continues to synthesize mRNA for hundreds to thousands of bases past this site, but this mRNA is degraded, and no part of it codes for a protein.

The newly synthesized mRNA is known as pre-mRNA at this stage since it must still undergo chemical modifications. In addition to the sequence of bases which will be translated into amino acids, the pre-mRNA contains a 5' untranslated region (5' UTR) that contains a sequence which helps the ribosome (see below) bind to the mRNA, and a 3' untranslated region which contains the poly (A) signal. The 5' end has a chemically modified guanine base added to it which also helps the ribosome to bind to the mRNA for translation. The 3' end of the mRNA is modified by the addition of a poly(A)

tail. This tail of usually 50 to 250 A bases helps regulate the degradation of the mRNA out in the cytoplasm of the cell. There is no DNA template for the poly(A) tail, that is, there is no sequence of corresponding T's on the DNA template strand, but there is a short sequence in the DNA (which is then transcribed to the mRNA) that indicates where the poly(A) tail should be added.

2.3.2 Splicing

Often chemical modification of the mRNA is followed by "splicing" where precise, predefined, subsequences are spliced out and degraded. These subsequences are known as introns (INTeRvening sequences), and the subsequences which are joined together are known as exons (EXpressed sequences). The joined exons, called "mature mRNA" or simply "mRNA," will pass out of the nucleus of the cell to the cytoplasm where protein synthesis occurs.

Most splicing occurs in a splicing complex known as a spliceosome. The spliceosome is a complex of the mRNA bound to small nuclear ribonucleoprotein particles (snRNPs – or "snurps"). The snRNPs are small nuclear RNAs (snRNAs) associated with between six to ten proteins. There are six principle snRNA called U1-U6 (the U designated they were "unusual" when first discovered), and all but U3 are involved with splicing. Introns that are spliced out by spliceosomes usually start with the dinucleotide GU, end with the sequence AG, and have a branch point sequence anywhere from 18 to 38 bases upstream from the end of the intron. Within this branch point

5' $\ldots E_n$ | donor splice site $\ldots$ branch pt. seq. $\ldots$ acceptor splice site | E_{n+1} $\ldots$ 3'

$A_{64}G_{73}$ | $G_{100}U_{100}A_{62}A_{68}G_{84}U_{63}$ $\ldots Y_{80}NY_{80}Y_{87}R_{75}A_{100}Y_{95}$ $\ldots 12YNC_{65}A_{100}G_{100}$ | N

Figure 2.3: Consensus sequences for regions of an intron. E_k denotes the k^{th} exon. | denotes an exon/intron or intron/exon boundary. R - a puRine (an A or G base), Y - a pYrimidine (a C or T/U), N - aNy nucleotide. The subscripts give the percentage occurrences of these bases at the indicated position relative to the splice sites. Subscripts of 100 are rounded, and there are many known exceptions (and many more may be found when introns are searched for without assuming that they start and end with these sequences. See http://www.ebi.ac.uk/asd/altextron/pre-release-dist-data.html for current percentages).

sequence, there is an A nucleotide known as the branch point. The start and end of the intron are known as the donor or 5' and acceptor or 3' splice sites respectively. Although the initial GU and terminal AG are the only highly conserved sequences from intron to intron, figure 2.3 shows that there are longer, less well conserved sequences at the donor and acceptor splice sites as well as the branch point sequence. Although this is useful information, the sequences given at the donor splice site and branch-point occur only 10 and 40 percent of the time respectively (and the branch-point sequence has only a single unique base represented), making these moderately conserved signals of limited value in splice site detection.

The assembly of the spliceosome begins with the U1 snRNP binding to the donor splice site. This is done by complementary base pairing of the snRNA in the U1 snRNP with the donor splice site sequence. Next the U2 snRNP binds in a similar fashion to the branch point sequence, causing the

intron to fold and bringing the 3' end of E_n in proximity to the 5' end of E_{n+1} – see figure 2.4 [Lewin, 2000, Figure 22.10]. The U4, U5, and U6 snRNPs join the complex, and then U4 snRNP dissociates from the complex forming an active spliceosome. The intron is cleaved at the donor splice site, and then this free end is ligated to the branch-pt. Next the acceptor splice site is cleaved and the ends of the exons joined. The spliceosome dissociates, the donor splice site end is released from the branch point, and then the intron is degraded.

Finally, it should be noted that there are other classes of introns (Group I and Group II introns) which undergo self-splicing [Davies et al., 1982], [Waring and Davies, 1984]. That is, the splicing is a protein-independent reaction, and little is known about conserved sequences in these introns. Thus, due to the different classes of introns, the incomplete understanding of spliceosomal activity, the short and poorly conserved sequences in the donor and acceptor splice sites, and the branch-point sequence, splice site detection remains an open problem.

2.3.3 Alternative Splicing

To add to the challenges of splice site detection, it is estimated that half of the human genes that are spliced can undergo alternative splicing [Mironov et al., 1999], [Brett et al., 2000], [Lander et al., 2001]. Alternative splicing yields different (viable) proteins through a variety of means: alternate donor splice site, alternate acceptor splice site, exon skipping, and splice

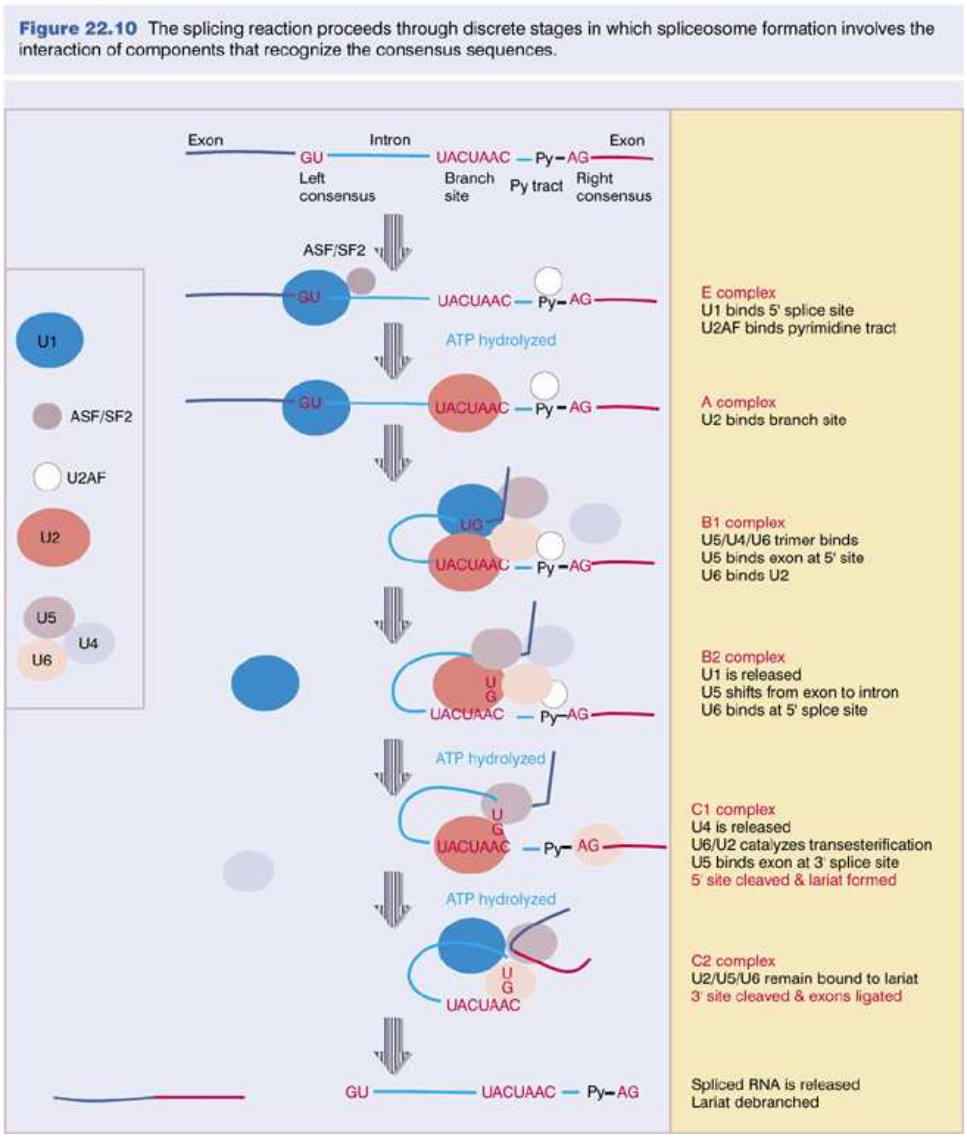

Figure 2.4: Splicing. Used by permission of Oxford University Press – Free permission.

versus no splice. When the spliceosome splices the intron at a different starting point, this is known as alternative donor splice site splicing. If these alternate starts to the splice site are off by a multiple of three nucleotides, then amino acids corresponding to the differing bases added or lost will be added or deleted from the final protein. If on the other hand the alternate starts are not off by a multiple of three, then there is a "frame shift" in the codons (see section on translation), and thus all subsequent corresponding amino acids can be different. A similar situation holds for alternate acceptor splice sites.

As the name implies, exon skipping occurs when an exon is skipped in the splicing process. Thus one less exon is incorporated into the mature mRNA. If the exon has a length that is a multiple of three, then a certain region of the resulting protein is excised, whereas if its length is not a multiple of three, then a frame shift occurs with the above mentioned consequences.

Splicing versus no splicing is similar to exon skipping, but instead of excising an exon, an intron is incorporated versus being spliced out. Once again the length of the intron determines a possible frame shift.

Some believe that exons exist because they confer upon the organisms carrying them the ability to more easily transfer discrete chunks of information (single exons) between genes [Gilbert, 1978]. Often proteins have certain regions, known as active sites, which carry out the important function of the protein. An organism with the ability to transfer the DNA corresponding to these active sites between genes might be given an evolutionary advantage, and although this transfer does not occur under regular

circumstances in a cell, through evolutionary time, it would become relatively common.

2.3.4 Translation

The information in the mRNA is used to synthesize protein in a process known as translation. The genetic code of this mRNA is read in consecutive, non-overlapping sets of three nucleotides. Each of these three nucleotides codes for a single amino acid. Thus a sequence of DNA has three frames, only one of which is used to make a particular protein. Consider the sequence ...TACGGTAATCCGGGT .... Since the sequence is read in triplets, it could be read as:

...TAC GGT AAT CCG GGT...,

...T ACG GTA ATC CGG GT... or

...TA CGG TAA TCC GGG T....

Each of these would code for an entirely different amino acid sequence. The triplets in the proper frame, which are used for protein synthesis, are known as codons.

There are 64 codons (four possible nucleotides in each of the three locations). Three of the codons, (TAA, TAG, and TGA, or their more commonly used mRNA counterparts: UAA, UAG, and UGA) signal that protein synthesis should stop, and are thus known as stop codons. The remaining 61 each code for one of the 20 amino acids. Some amino acids are coded for by only a single codon, while other amino acids have as many as six codons

which code for them – known as redundancy. This single codon coding for a single amino acid, which is highly conserved throughout all organisms, is known as the "universal code."

Protein synthesis occurs on a cellular organelle known as a ribosome. When one of the many ribosomes in the cytoplasm of the cell comes in contact with the 5' end of the mRNA, the ribosome becomes attached to it. The ribosome "reads," or moves down the mRNA three bases, or one codon, at a time. There are two sites on the ribosome, each of which can hold both a codon from the mRNA, and a transfer RNA (tRNA), and the tRNA's associated amino acid. Each tRNA has a three base anticodon on it. Only a tRNA with an anticodon that matches the mRNA codon (by complementary base pairing) which is docked on the ribosome can dock at the ribosome site; this assures that the correct amino acids are placed in proximity to one another. The amino acid (residue) from the most recently attached tRNA is then attached to the growing polypeptide chain, and thus the DNA message is faithfully transferred via the mRNA to the final polypeptide – see figure 2.5 [Snustad et al., 1997, Figure 11.5].

2.4 Current Methods

Most genomic data consists of "raw" DNA sequences whose effective components and functions are completely unknown. For a given DNA sequence, it is not known *a priori* whether the sequence came from (or spanned) a gene or not. If it is from a gene, we would like to know if it is from an exon or an

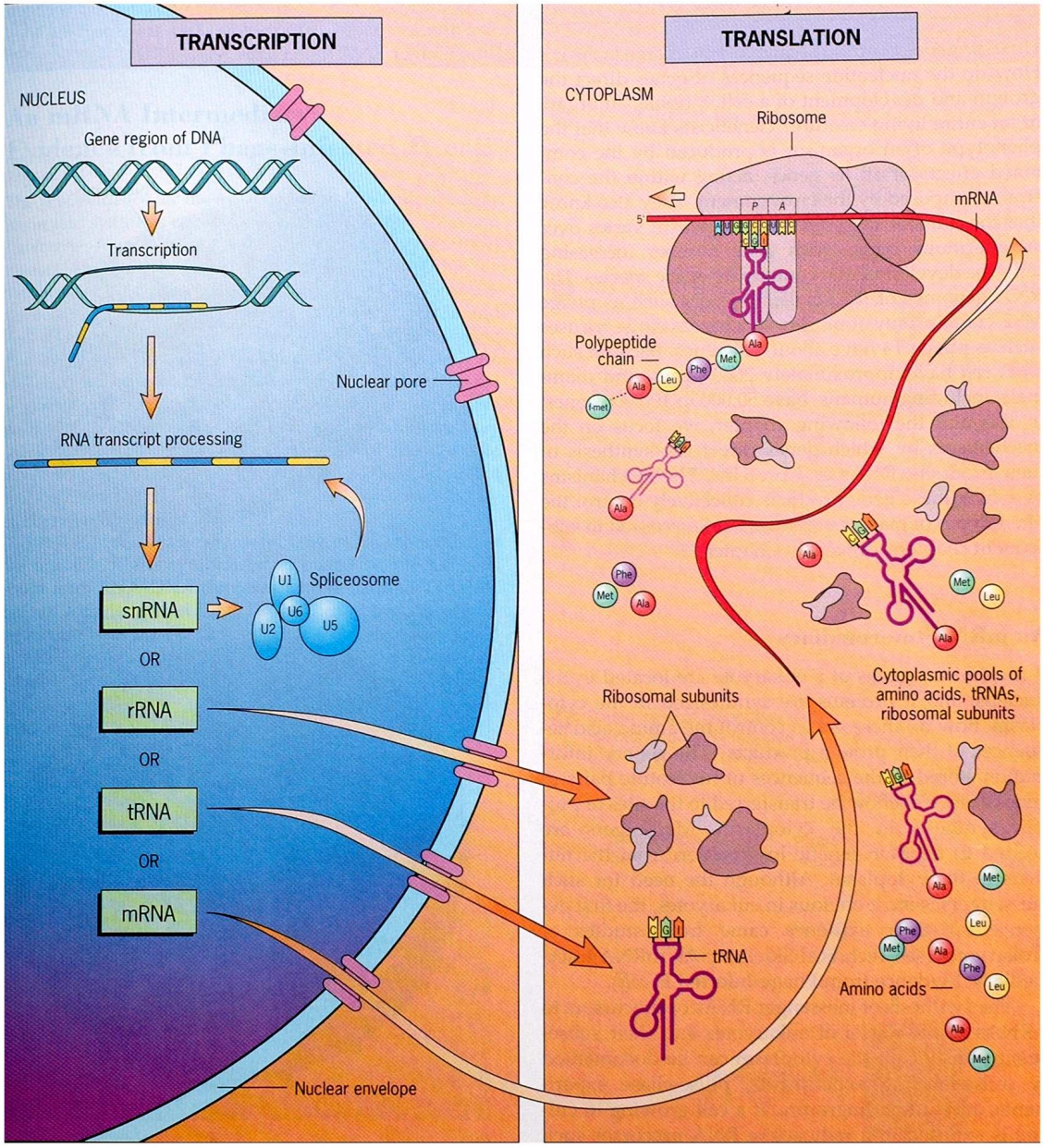

Figure 2.5: Transcription and Translation – diagram. ©1997 by John Wiley & Sons, Inc. Reprinted with permission of John Wiley & Sons, Inc.

intron, and if it is from an exon, we would like to know which of the three frames it is read in in order to find the protein product for which it codes.

Current methods for finding genes and distinguishing between their exons and introns can be grouped into computational or biological methods. While the computational approach is crude as compared to biological experiments, it is much faster. Often the two methods are used in conjunction with one another, with the computational predictions giving a starting point for further biological experimentation. It should also be noted that the basic underlying information used by computational methods is obtained through biological methods. A brief overview of both the biological and computational approaches to finding the function of a given DNA sequence follows.

2.4.1 Biological Methods

When a biologist seeks the sequence of a gene, she or he has an underlying question that must be addressed. Often, a biologist is interested in identifying a specific gene or genes. These genes may code for protein(s) that affect the phenotype under study. In other cases, a biologist may be less interested in what a gene does than what it indicates about, say, the evolutionary relatedness of organisms.

Cystic fibrosis (CF) provides a classic example of the biological methods used to locate and sequence a gene for a known phenotype (gene product). It is the most prevalent genetic disorder among whites in the United States. The most common cause of CF is a single three-base deletion which disrupts

the function of a ion channel protein. This malfunctioning protein causes an accumulation of mucus on the surface of certain cells, often resulting in chronic infections and malnutrition. In 1940, the life expectancy of a newborn with CF was two years. With advances in the understanding of the disorder, this has grown to over 30 years. The cystic fibrosis gene was found by positional cloning, or reverse genetics – a process that took four years and the efforts of many laboratories [Rommens et al., 1989].

The first scientific step in finding a cure for a disease is to identify the causative agent itself. This agent may have an environmental or genetic basis. If it is genetic, then many members of the same family – some with the disease, and others without, can be analyzed. If some "genetic marker" (a DNA signature that is found more frequently in those individuals with the disease than without) can be linked to the disease, then one may be able to locate the genes whose variants cause the disease.

The genetic marker can be any detectable genetic difference between individuals. Occasionally these differences can be seen under a light microscope. The DNA-containing chromosomes of a cell are condensed at a phase of cell division known as metaphase. If the chromosomes are stained at this stage, they are visible under the microscope. This complete set of chromosomes is called a "karyotype" (literally, nucleus type).

Karyotypes can be visualized with different staining dyes. Some of these show not only the condensed chromosome, but also a banding pattern within the chromosomes. These bands have an average length on the order of 10 million nucleotides. Although other staining techniques are available, they

5'...CCATGAT**GAATTC**CTTTAC...3'

3'...GGTACTA**CTTAAG**GAAATG...5'

$\Downarrow$

5'...CCATGAT**G** **AATTC**CTTTAC...3'

3'...GGTACTA**CTTAA** **G**GAAATG...5'

Figure 2.6: Cutting of DNA sequence by the restriction enzyme EcoRI ("echo-R-one"). The boldface nucleotides represent the six base-pair "restriction site" for EcoRI

all show very large scale differences in chromosomes.

The CF gene did not show any linkage to visible karyotype differences. Instead, CF was linked to a restriction fragment length polymorphism (RFLP) [Rommens et al., 1989]. A RFLP is a difference in the way two individual genomes are "cut" by a "restriction enzyme." Restriction enzymes are naturally occurring enzymes in bacteria that are used to defeat invading viruses by literally cutting the invading virus's genetic code, and thus rendering it inactive. Biologists have learned to use many different restriction enzymes [Russell, 1998], each of which cuts specific nucleotide sequences (usually 4 to 8 base pairs in length – e.g. EcoRI cuts DNA as shown in figure 2.6). If similar pieces of a chromosome are cut at different places, the resulting segments have different lengths. The pattern of segment lengths can be observed using a procedure known as Southern blot analysis.

Cystic fibrosis was linked to a particular RFLP in an analysis of a group of related individuals (a "pedigree"), some of who had CF [Davies et al., 1987].

Using a radioactively labeled RFLP probe, it was shown that the CF gene was on chromosome 7. Then known markers on this chromosome were found (again with linkage analysis) that flanked the CF gene. These two markers are approximately 1.5 million base pairs (bp) apart – a large distance. Subsequent research found two new markers that flanked the gene that were about 500 thousand base pairs apart. Through additional processes, known as chromosomal walking and chromosomal jumping, many clones (copies of DNA made by various replicating agents) of the DNA in this region were made, and their union spanned 500 thousand base pairs containing the CF gene.

There can, however, be many genes in a region of this size and finding the gene of interest poses additional problems. One technique of finding genes is to take the DNA clones, which are labeled by some method, and try to hybridize them (join through complementary base pairing of the two strands of DNA/RNA) with DNA from various animals in a procedure known as a zoo blot. The reasoning behind this procedure is that there is less among species variation in gene sequences than in intergenic sequences. This is because random mutations in intergenic regions have no or little phenotypic affect on an individual, and are expected to accumulate more rapidly than mutations in a gene where they are presumably culled from a population by natural selection [Rogic et al., 2001].

In the CF studies, five probes (clones that are labeled by some method – radioactively, fluorescently, etc.) hybridized with the DNA from other organisms – identifying them as possible CF genes. Two of these were excluded

as containing the CF gene by linkage analysis, and one was found to be a pseudogene – a sequence that had been a gene in the past, but became inactivated by mutations to regions that control expression of the gene. Of the two remaining probes, one was ruled out for its inability to hybridize with any mRNA extracted from human cells. The remaining probe was shown to hybridize with a cDNA (complementary DNA) made from mRNA extracted from the cytoplasm of the cell. This cDNA was about 6,500 bp long – indicating the mRNA was also about 6,500 bp long. Subsequent work showed this mRNA was indeed the CF gene, and that it spanned 250 kb of DNA and involved 24 exons (Exons can be found hybridizing the mRNA – or cDNA made from it – with the chromosomal DNA. As only the exons are present in the mRNA, the introns are spliced out, the mRNA/cDNA will hybridize only where the exons are located) [Riordan et al., 1989].

Cystic fibrosis was the first gene found by positional cloning. This technique is also known as reverse genetics since the gene is found without any knowledge of the gene product, the protein, itself. As cystic fibrosis is the most common recessive disease of Caucasians, locating the gene responsible was a major breakthrough for geneticists in the latter 1980's.

The hunt for the cystic fibrosis gene ultimately concentrated on a minuscule fraction of the human genome. Working with karyotypes, RFLPs, labeling DNA probes, and hybridizing DNA is both effective and accurate, but it is also a slow and costly process. By comparison, the Human Genome Project has created a draft which covers the entire three billion base pairs of our genome and is 99.99 percent accurate. The challenge is to now locate

the genes within this data through more efficient, namely, computational means.

2.4.2 Computational Methods

A large number of computational approaches (around 100) have been created for locating genes in genome sequence data. For a review, see Mathe *et al.* [Mathe et al., 2002], and the website of Wentian Li (http://linkage.rockefeller.edu/wli/gene/). This section will describe the major categories of methods.

Computational methods of gene identification and exon/intron detection can broadly be classified into three main groups: signal sensors, content sensors, and similarity searches. Signal sensors focus on subsequences that may indicate the character of the DNA. For example, a series of promoter elements may indicate that a gene is present downstream. Content sensors seek intrinsic properties of a sequence of interest. For instance, a content sensor could look for the nucleotide triplet frequencies in a sequence, and compare the frequencies to known intron and exon triplet frequencies. Similarity searches look for subsequence similarities between two sets of data. The most basic similarity search is to search a sequence for a particular relatively short subsequence. Suppose for example, that a gene sequence has been found in an experimental organism such as the yeast *Saccharomyces cerevisae*. One would then search the human genome for a similar gene sequence (up to a third of yeast genes and about

half the genes of the fruit fly *Drosophila melanogaster* have human counterparts).

Signal Sensors

Detecting biological sequence signals (promoter elements, donor splice sites, acceptor splice sites, branch point sequences, poly (A) signals, ...) is probably the most intuitive method for detecting genes, exons and introns. Indeed it is undoubtedly the underlying method used by the cell itself. Unfortunately it has met with limited success in inferring the function of raw genomic data. The variability of the consensus sequences (the most common sequence – see figure 2.3 for an example) in the promoter (they are not "well" or "highly" conserved), the lack of any promoter element being present in all promoters, and the variability in the number of any specific promoter element have all led to difficulties with this approach in predicting the transcription initiation site of genes. Pedersen *et al.* [Pedersen et al., 1999] gives a review of the biology of promoters and some of the computational difficulties in locating them. Ohler *et al.* [Ohler et al., 2000], [Ohler and Niemann, 2001] review the literature on computational promoter prediction. Signal sensors have been more successful in detecting splice sites between exons and introns. SplicePredictor [Kleffe et al., 1996] (http://www.bioinformatics.-iastate.edu/cgi-bin/sp.cgi) and SPLICEVIEW [Rogozin and Milanesi, 1997] (http://l25.itba.mi.cnr.it/~webgene/wwwspliceview.html – n.b. the character in l25 is an "el") try to predict splice sites by identifying consensus sequences. Other signal sensors predict splice sites using one or more

of the following techniques: maximal dependence decomposition (MDD), hidden Markov models (HMM), and neural networks (NN), which are briefly described below. GeneSplicer [Pertea et al., 2001](http://www.tigr.-org/tdb/GeneSplicer/index.shtml) uses both HMM and MDD, NETGENE2 [Tolstrup et al., 1997] (http://www.cbs.dtu.dk/services/NetGene2/) uses NN and HMM, and NNSPLICE0.9 [Reese et al., 1997] (http://www.fruitfly.-org/seq_tools/splice.html) uses NN.

Maximal Dependence Decomposition Maximal dependence decomposition [Burge and Karlin, 1997] was developed to identify the most significant dependencies between positions of a splice site. It is a generalization of the weight array model [Zhang and Marr, 1993], which is itself a generalization of the weight matrix method [Staden, 1984]. The weight matrix method uses the frequencies p_j^i of the j^{th} nucleotide at position i to estimate the probability

$$Prob(X) = \Pi_{i=1}^{n} p_{x_i}^i$$

of generating the sequence $X = x_1, x_2, \ldots, x_n$. The weight array model, which takes into account dependencies between adjacent sites, calculates the probability as

$$Prob(X) = p_{x_1}^1 \Pi_{i=2}^{n} p_{x_{i-1},x_i}^{i-1,i}$$

where $p_{j,k}^{i-1,i}$ is the conditional probability of nucleotide x_k at position i given that the nucleotide at position $i-1$ is x_j.

Maximal dependence decomposition starts with a set D of N aligned sequences of length k. These sequences could be any type of biological signal

for which dependencies between nucleotides is sought. Burge and Karlin [1997] used the nine nucleotide sequence that corresponds to the last three bases of an exon, and the first six bases of the intron of a donor splice site. The positions were denoted -3, -2 -1, 1, 2, 3, 4, 5, and 6 with positions 1 and 2 always being the canonical GT (or GU in the tRNA) in the set D. The most frequently occurring base(s) at each position is/are called the consensus base(s), and an indicator variable C_i is assigned the value 1 if the i^{th} base of a given sequence of D is equal to the consensus base(s), and 0 otherwise. The nucleotide indicator X_j identifies the nucleotide at position j. For each pair of i, j with $i \neq j$, a contingency table is formed. The χ^2 values with i or j equal to one or two were omitted from their table as these positions did not have any variability in their data set. Of the remaining 42 i, j pairs, 31 had a significant χ^2 value at the relatively stringent level of $P < 0.001, df = 3$. This demonstrated that there was a great deal of dependence among these nine nucleotides. Next, the sum

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

is calculated, which gives a measure of the dependence between C_i and the nucleotides at the other positions. A binary decision tree is then used to subdivide their set as follows. Choose the value i_1 such that S_{i_1} is maximal, and partition D into two subsets, D_{i_1} and D_{i_1}-. D_{i_1} contains all the sequences from D which have the consensus nucleotide(s) at position i_1 and D_{i_1}- contains the sequences which do not.

Each of these subsets is recursively subdivided until one of the following

three conditions is met: i) the $k-1^{th}$ level of the tree is reached (and thus no further subdivision is possible); ii) no significant dependencies between positions is found; or iii) the size of the subset is small enough that further subdivision would result in weight matrix method frequencies that would be unreliable. Burge and Karlin derive a separate weight matrix method model for each subset of the tree, and use them in their larger hidden Markov model.

Neural Networks A neural network (or more precisely an artificial neural network) can be thought of as a weighted directed graph with the nodes and edges being the neurons, and weighted edges connecting the neurons [Agatonovic-Kustrin and Beresford, 2000]. A neural network is characterized by:

- network architecture (or the topology of the network)
- network node properties (threshold values, ...)
- weights of the edges between the neurons (the strength of their association)
- learning or updating algorithm used by the network.

Figure 2.7 illustrates an example of a "perceptron." The input, or pattern, $x_1, x_2, \ldots, x_n$, and their associated edge weights $w_1, w_2, \ldots, w_n$ will cause the neuron to "fire" (produce a 1) if $\sum_{i=1}^{n} w_i x_i > T$. If this threshold is not reached, then the neuron does not fire (a 0 is produced). The "training" of the perceptron (or of a more complicated neural network), is accomplished

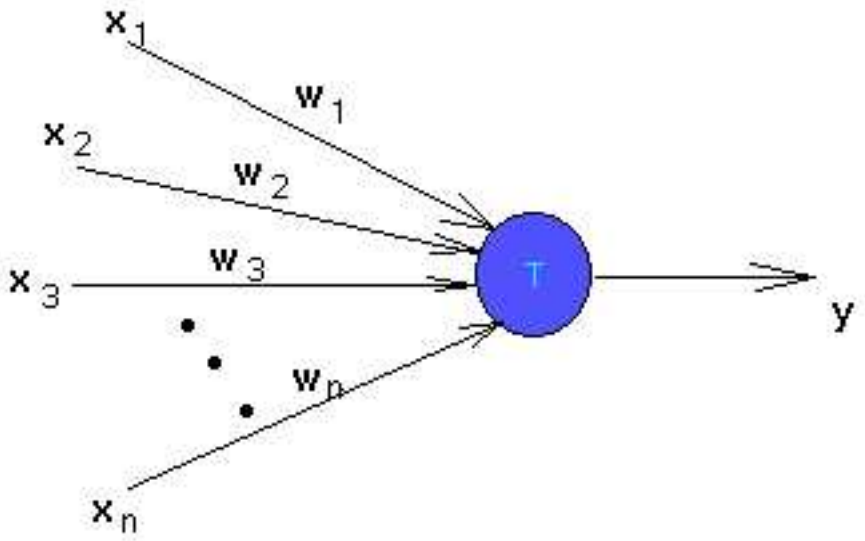

Figure 2.7: A perceptron with an input of dimension n and threshold T. Used by permission of the author – http://www.iiit.ac.in/~vikram/nn_intro.html.

by supplying it with inputs from a "training set" where the "answer" is known. If the perceptron produces the correct answer, then the weights are not modified, but if it produces the wrong answer, they are updated according to the learning algorithm. A simple algorithm is to decrease w_i by cx_i if the neuron fired when it should not have, and to increase w_i by cx_i if the neuron failed to fire when it should have. Thus the "knowledge" of the system is stored in the weights of the edges (and this is thought to be a component of biological brains as well). The condition $\sum_{i=1}^{n} w_i x_i = T$ defines a hyperplane, and thus training this simple perceptron is equivalent to finding the values w_i such that the set of patterns with one answer are separated from those

with the other answer by the hyperplane.

More complicated neural networks can have many hidden layers with many neurons in each layer. If all signals are passed in the same direction, then it is said to be a "feed forward" network, but if any loops occur, then it is known as a "recurrent" (feedback) network.

Genie [Reese et al., 2000] is a popular hidden Markov model gene identification program that uses neural networks to help identify promoter regions. GRAIL [Uberbacher and Mural, 1991] on the other hand uses neural networks as the main scheme in gene detection. GRAIL's coding recognition module is an artificial neural network with seven input nodes, two layers of hidden nodes, and an output node. The seven input nodes correspond to weights, or levels of confidence, that a subsequence is a coding region. Each input nodes weight is derived from a distinct algorithm which analyzes the coding potential of the sequence. This neural net is trained on known coding and noncoding regions. If the output from the net then exceeds a given threshold, it is predicted that the subsequence comes from a coding region.

Hidden Markov Models A Markov model is a set of states and a corresponding set of values which give the probabilities of changing from one state to another [Rabiner, 1989]. One can consider the weather at noon on consecutive days as being modeled by a Markov model where there are, for example, three distinct states: sunny, overcast, and precipitating. Call these states 1, 2, and 3 respectively. Then $a_{ij}, 1 \leq i \leq 3, 1 \leq j \leq 3$ is the probability of the weather being in state j given that it was in state i the previous day.

For a N state model we have $a_{ij} \geq 0$ and $\sum_{i=1}^{N} a_{ij} = 1$. These are examples of discrete first order Markov chains. If the transition values are dependent not only on the current state, but also on prior states, then a higher order Markov model may be used.

In the above example, the states of the system are observable – one can tell if it is sunny or rainy. In a hidden Markov model, some signal is observable, but the underlying state of interest which generated the signal is hidden. Consider, for example, three urns that are hidden behind a curtain in a room. Each urn has a specific (hopefully sufficiently different) proportion of red, green, and blue balls in it. A transition matrix $A = a_{ij}, 1 \leq i \leq 3, 1 \leq j \leq 3$ is given which gives the probability of drawing a ball from urn j given that the previous ball was drawn from urn i.

A person behind the curtain selects the urn using the transition matrix A, and then randomly draws a ball from that urn. The person then shows the color of the ball draw (the observation), but does not divulge the urn from which it was drawn (the state). The goal is to surmise from the observation sequence, the state sequence.

More generally, consider a model with M observable signals, N states and T observations. Let q_t and O_t represent the state and observation, respectively, at step t. Then we wish to determine $Q = q_1 q_2 \ldots q_T$ from $O = O_1 O_2 \ldots O_T$.

Let π be the N dimensional initial probability state vector with $\pi_i = P(q_1 = S_i)$ giving the probability that the process starts in state i, and $b_j(k) = P(O_t = v_k | q_t = S_j)$ giving the probability that the t^{th} observation is

v_k, given that the process is in state j. Thus the matrix $B = b_j(k), 1 \leq j \leq M, 1 \leq k \leq N$ simply gives the probability of each observation from each state.

The hidden Markov model is characterized by the triplet $\lambda = (A, B, \pi)$. Given the model λ, we need to calculate $P(O|\lambda)$, choose the state sequence Q which "best" explains the observation sequence O, and update the model parameters λ to maximize $P(O|\lambda)$. For a thorough discussion of these three problems, see the tutorial by Rabiner on hidden Markov model [Rabiner, 1989].

The problem of finding $P(O|\lambda)$ is critical. It provides a way to compare different models (λ's) for a given observation sequence O, and to choose the model which best matches the observations. It is a straightforward task to calculate the probability of a particular observation sequence O given λ and a particular state sequence $Q = q_1 q_2 \ldots q_T$:

$$\begin{aligned} P(O|Q,\lambda) &= \Pi_{t=1}^{T} P(O_t|q_t,\lambda) \\ &= b_{q_1}(O_1) b_{q_2}(O_2) \ldots b_{q_T}(O_T), \end{aligned}$$

assuming observations are statistically independent for a given sequence Q. We can also easily find the probability of any state sequence Q given λ:

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T}.$$

The product of these two gives the joint probability of O and Q:

$$P(O,Q|\lambda) = P(O|Q,\lambda) P(Q|\lambda). \tag{2.1}$$

The probability of the observation sequence O given the model λ is found by summing the joint probability (2.1) over all possible state sequences Q:

$$\begin{aligned} P(O|\lambda) &= \sum_{allQ} P(O,Q|\lambda) \\ &= \sum_{allQ} P(O|Q,\lambda)P(Q|\lambda) \\ &= \sum_{q_1 q_2 \ldots q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \ldots a_{q_{T-1} q_T} b_{q_T}(O_T). \end{aligned}$$

Unfortunately, this is impractical to compute unless observations are limited to only a small number of states (i.e. most $b_j(k) = 0$ for most j). In general, if all of the N states can yield M observations, then each of the N^T possible state sequences Q, of length T, could yield the observation sequence O of length T. So for all but the smallest T's, a new method is needed to calculate $P(O|\lambda)$. Rabiner [Rabiner, 1989] shows that by finding the probability of a partially observed sequence (starting with the first observation, and using induction), the number of calculations to compute $P(O|\lambda)$ can be cut to order of N^2T.

The Viterbi algorithm [Viterbi, 1967], [Forney, 1973] is often used to calculate a "most probable" state sequence Q. This algorithm finds the state sequence which maximizes $P(Q|O,\lambda)$, that is, it maximizes the probability of the entire state sequence. This is very different than trying to choose the most likely state individually at each step t. To appreciate this difference, consider the three urns again. All three urns contain each of the colors blue, green, and red, but urns one (S_1), two (S_2) and three (S_3) have a prepon-

derance of blue, green, and red balls respectively. Assume also that $a_{13} = 0$, that is one can not move from urn one to urn three directly. If we were to observe the sequence blue, blue, red, to maximize $P(q_i|O_i, \lambda)$, for each individually, we would choose a state sequence $q_1 = S_1, q_2 = S_1, q_3 = S_3$, even though this state sequence is impossible. The Viterbi method, by contrast, would assign this sequence a probability of zero. At times it may be desirable to calculate the most probable state at a given step t, and this can be done with two algorithms known as the forward and backward algorithms, which are described in the methods section of chapter 4.

Although there is no known method of adjusting λ to maximize $P(O|\lambda)$, the Baum-Welch [Baum, 1972] algorithm does well in practice, and assures us of finding a local maximum. This is an iterative algorithm which uses a training set of data to update the model's parameter values. For example, a_{ij} is updated using (the expected number of transitions from state S_i to state S_j)/(expected number of transitions from state S_i), and similarly for the other model parameters. The reader is referred to Rabiner [Rabiner, 1989] for further details.

GENSCAN [Burge and Karlin, 1997] is one of the most accurate gene prediction programs [Zhang and Zhang, 2002]. The hidden Markov model they use has 27 states corresponding to various functional units within the genome. Exons, for example, are treated as separate states depending on whether they are initial, terminal, or internal exons. Additionally, internal exons are broken into three states as determined by the "phase" (where the intron falls in relation to the three bases of a codon) of the preceding intron.

Content Sensors

As high throughput DNA sequencers started to flood databases with human sequences, content sensor methods became the primary means of identifying putative novel genes. Previously unknown human genes for which there were known gene sequences from other organisms were often identified through similarity searches.

Content sensors classify sequences based on statistical differences between different categories of sequences. Nucleotides C and G, for example, are known to occur at higher frequencies in exons than in introns [Mathe et al., 2002], and thus give a clue to the classification of a sequence. Both neural networks and hidden Markov models have been employed in content sensors.

Content sensors typically do a fairly accurate job of finding coding sequences, but often find numerous false positives as well [Guigo et al., 2000], [Guigo et al., 2003]. That is, they have high "sensitivity," but low "specificity." Delimiting exact boundaries between regions (exons and introns for example) is also often a problem. For both of these reasons, computational programs often combine techniques. For example, a program may search for exons with a content sensor, and then identify splice sites using signal sensors.

GENSCAN by Burge and Karlin [Burge and Karlin, 1997] (http://genes.mit.edu/GENSCAN.html) is one of the most widely used gene finders. It combines hidden Markov models for exon detection, with

maximal dependence decomposition and weight array model methods to find the donor and acceptor splice sites, respectively. GeneParser [Snyder and Stormo, 1993], [Snyder and Stormo, 1995] (http://beagle.-colorado.edu/~eesnyder/GeneParser.html) and Genie [Reese et al., 1997] (http://www.fruitfly.org/seq_tools/genie.html) both use neural networks to detect coding regions, and then check these putative coding sequences against either expressed sequence tags (short sequences of DNA recovered from the cytoplasm of the cell, and thus sequences which have been transcribed and have not been spliced out) in the former, and proteins from a protein database in the latter.

All three of these programs also use dynamic programming to find optimal gene models. That is, from all the possible donor and acceptor splice sites, they find a consistent set (donor site followed by acceptor site) which gives the best entire gene prediction. The reader is referred to Krogh [Krogh, 1998] for a review of dynamic programming.

Similarity Searches

Similarity searches have been one of the major techniques of gene identification, since even before computational approaches were employed. The zoo blot mentioned in the biological methods section is accomplished through binding of similar (or complementary) nucleotide sequences – see for example [Russell, 1998, page 485]. The same idea is used in similarity searches: find a sequence in uncharacterized DNA that is similar to a known DNA sequence (a gene or exon for example). BLAST (Basic Local Alignment Search Tool)

[Altschul et al., 1990] compares a so-called query sequence with a database of nucleotide sequences, to find the highest scoring match. There are numerous variations such as BLASTP [Altschul et al., 1997], which compares an amino acid query sequence against a protein sequence database. Still others compare amino acid/nucleotide query sequences to nucleotide/protein database. These are some of the most commonly used gene locating programs in computational biology. A limitation to BLAST is that the query sequence can only be on the order of a gene's length. For the above mentioned similarity searches to a known sequence, this poses no problem, but with the completion of other vertebrate genome sequences, entirely new techniques of gene identification are being developed.

Another strategy is to search for exons by finding similar DNA sequences in two species genomes [Zhang et al., 1998]. This approach relies on the premise that exon sequences are more highly conserved than intron sequences. The basis for this premise that random mutations in exons are expected to have a deleterious effect on an individual and are thus removed from a population by natural selection, whereas random mutations in introns cause no phenotypic change in an individual [Sunyaev et al., 2003].

Algorithms which use similarity searches include "global alignment" algorithms which compare an optimal similarity score over the entire length of the two sequences. In contrast, "local alignment" algorithms start with very short exact matches, and then extend the sequence out as far as possible.

AVID [Bray et al., 2003], [Couronne et al., 2003] and LAGAN [Brudno et al., 2003] are global alignment programs that compare the

genomes of two organisms. The mouse is thought to be an ideal model organism for global alignment with human genome data due to its relatively close evolutionary proximity. Results using global alignment methods are reported for the human-mouse comparison in [Bray et al., 2003], [Couronne et al., 2003] and [Brudno et al., 2003]. Organisms that are even more closely related, like primates and humans, may not have sufficiently divergent introns; the exons from more distantly related organisms may be too dissimilar to make comparisons meaningful.

SLAM [Alexandersson et al., 2003] is another global sequence aligner that predicts exons from conserved sequences as well as conserved noncoding sequences (CNS) using a CNS state in its hidden Markov model. SLAM [Alexandersson et al., 2003] was run both with and without CNS information. The study concluded that both the sensitivity and specificity for exon detection were increased by considering CNS data.

Another global alignment algorithm is described in the paper by Boffelli *et al.* [Boffelli et al., 2003]. They employ a novel approach in that they use multiple closely related species in their alignments instead of the more commonly used two distantly related species. Boffelli *et al.* coin the term "phylogenetic shadowing" to describe their use of a phylogenetic tree which includes humans. The tree is used to compare similar regions of DNA sequences with up to 17 primate species. Phylogenetic shadowing is a variation of phylogenetic "footprinting" which considers conserved sequences, but does not take into account their phylogenetic tree [Tagle et al., 1988], [Gumucio et al., 1992]. As Gibbs and Nelson put it [Gibbs and Nelson, 2003], rather than make the

standard assumption that "what is important is conserved," phylogenetic shadowing takes the point of view "what is not critical can vary – at least some of the time."

One of the main benefits of phylogenetic shadowing is the ability to find newly evolved genes in humans, or genes which have become inactive in a model organism. For example Apolipoprotein (a) is a newly evolved primate gene product that is also of considerable biomedical importance (its presence at high level in the plasma is a cardiovascular disease risk predictor). Phylogenetic shadowing showed that the exon regions as well as the TATA box and another previously characterized promoter region were highly conserved in this proteins gene sequence [Boffelli et al., 2003]. In addition, 8 short, previously uncharacterized, regions upstream from the promoter site showed a high degree of conservation. It was hypothesized that these regions play a role in gene expression. To test this hypothesis, an electrophoretic mobility–shift assay was performed with sequences from both highly and poorly conserved regions. This test showed that DNA binding proteins bound tightly to the conserved regions, and weakly or not at all to the nonconserved regions. As further support, experiments performed with the conserved and nonconserved regions individually deleted from the sequence showed that gene expression was affected by deletion of the conserved regions, but not by the deletion of the nonconserved regions [Boffelli et al., 2003]. An important caveat of computational gene identification methods was addressed by Boffelli et al. here: although these methods can vastly accelerate answering many biological questions, the results need to be verified with biological experiments.

2.5 New Method

This section will introduce a new method of exon and intron detection. Using only local data it makes predictions as to the function of the raw sequence data using previously collected frequency counts of small DNA subsequences in known exon and intron data. Using these frequencies, and likelihood ratios, we hope to increase the overall accuracy of exon and intron detection either in general, or under specific circumstances.

In 1925 Sir Ronald Fisher coined the term "likelihood" in the following passage, in regards to comparing different hypotheses [Edwards, 1972, page 9]:

> What has now appeared is that the mathematical concept of probability is inadequate to express our mental confidence or diffidence in making such inferences, and that the mathematical quantity which appears to be appropriate for measuring our order of preference among different possible populations does not in fact obey the laws of probability. To distinguish it from probability, I have used the term 'Likelihood' to designate this quantity.

Let $P(R|H)$ denote the probability of obtaining result R given hypothesis H. Then the *likelihood*, $L(H|R)$, of the hypothesis H given the result R is proportional to $P(R|H)$, with the constant of proportionality being *arbitrary*. Although $P(R|H)$ and $L(H|R)$ are functions of both R and H, in the former, H is usually considered to be fixed; then $P(R|H)$ is a function of only the result (data, outcome, etc.). On the other hand, in $L(H|R)$, R is

considered to be fixed, and thus $L(H|R)$ is a function of various *hypotheses* under consideration.

As Edwards points out [Edwards, 1972, page 9], this distinction is fundamental. While the arbitrary constant of proportionality makes the quantity $L(H|R)$ itself of little interest, and it does not give rise to a statistical distribution, Edwards [Edwards, 1972, pages 9-10] reveals that this is in fact not a shortcoming at all:

> The arbitrary constant of proportionality enables us to use the same definition of likelihood for discrete and continuous variables alike, and is no impediment to its use, which invariably involves the *comparison* of likelihoods. Though it is a constant in any one application, involving many different hypotheses but the same data and probability model, it is, of course, not necessarily the same constant in another application. This, too, is no hindrance, for we shall not be attempting to make an absolute comparison of *different* hypotheses on *different* data.

Here, the quantity of interest is a "likelihood ratio," defined as $L(H_1|R)/L(H_2|R)$ for two hypotheses H_1 and H_2; the arbitrary (fixed) constant cancels out. While a given value of a likelihood ratio does not correspond to a particular probability or confidence interval, it does compare the credibility of two hypotheses for given data.

Our interest is in using likelihood ratios to detect introns and exons in a sequence of DNA. The orientation of the sequence of DNA (i.e. the direction

in which it is "read"), will be known from the sequencing technique. If exons are in the sequence, however, the reading frame will not be known, and thus allowance must be made for all three frames. The complementary DNA strand can (and must) be checked as well. The reverse complement of the given sequence is easily formed and similarly analyzed.

Our likelihood method for distinguishing exons from introns is as follows. Let the DNA sequence of interest have length L. We proceed by looking at overlapping subsequences of length wl (window length), and computing the likelihood ratio for each of these subsequences. More precisely we will look at a subsequence of length wl (seq_{wl}), where wl is divisible by n – the length of the DNA subsequences whose frequencies were previously calculated. The seq_{wl} can then be broken into these n-tuples, and we can compute $L(H_e|Data)/L(H_i|Data)$ where H_e and H_i are the hypotheses that seq_{wl} came from an exon or intron respectively, and where $Data$ is the seq_{wl} itself. Assuming the probability of an n-tuple at position k is independent of the n-tuple at position $k-1$, they can be thought of as coming from a multinomial distribution with 4^n (there are 4^n DNA sequences of length n) outcomes, and thus we have

$$L(H_e|Data) = k * Prob(Data|H_e) = k * \frac{(wl/n)!}{a_1!a_2!\ldots a_{4^n}!} p_1^{a_1} p_2^{a_2} \ldots p_{4^n}^{a_{4^n}} \quad (2.2)$$

where wl/n is the total number of n-tuples in seq_{wl}, a_i is the number of occurrences of the i^{th} triplet in seq_{wl}, and p_i is the probability (relative frequency) of the i^{th} n-tuple within exons, similarly for $L(H_i|Data)$. The likelihood ratio of interest is the ratio of equation 2.2 to its intron counterpart.

Upon cancellation of like terms and taking the log for computational ease, we have:

$$\lambda = \log \frac{L(H_e|Data)}{L(H_i|Data)} = \log \frac{p_1^{a_1} p_2^{a_2} \ldots p_{4^n}^{a_{4^n}}}{q_1^{a_1} q_2^{a_2} \ldots q_{4^n}^{a_{4^n}}} \tag{2.3}$$

or more simply:

$$\lambda = a_1 \log \frac{p_1}{q_1} + a_2 \log \frac{p_2}{q_1} + \ldots + a_{4^n} \log \frac{p_{4^n}}{q_{4^n}} \tag{2.4}$$

where q_i is the frequency of the i^{th} triplet within introns. Each of the 4^n p_i/q_i are computed only once, and so each λ takes only wl/n additions and multiplications.

For this study, n-tuple frequencies in human exons and in introns were extracted from the Exon-Intron Database (EID, http://www.mcb.harvard.-edu/gilbert/eid/) [Saxonov et al., 2000]. This is an exhaustive database of protein-coding intron-containing genes compiled from the GenBank 115 release [Benson et al., 2000]. The distribution of EID includes not only the database itself, but tools to extract and analyze sequences from the database. Using the EID perl program extract_species.pl [Saxonov et al., 2000], human genes were extracted from the database. The EID filter_exp_keyw1.pl [Saxonov et al., 2000] program was applied to this subdatabase, giving sequences that failed to contain certain keywords in their GenBank annotation. The keywords indicate whether the sequence or splice sites were found computationally or experimentally, and so this filter can be used to extract experimentally determined genes and splice sites from a subdatabase. It is generally accepted that experimentally found genes and their exon and in-

tron boundaries have a higher accuracy than those found with computational methods [Claverie, 1997], and thus this filter should give a more accurate training set. In addition sequences were removed that did not start with the canonical ATG translation start site, or that had any non-ACGT characters in them (there are various standard symbols to denote, any nucleotide (N), a Purine (R), a pyrimidine (Y), any non A (B), etc).

There is only one reading frame in exons which corresponds to how the DNA is read by the cell machinery to make proteins – the codons (see introduction). Introns on the other hand have $2n$ (where n is the length of the n-tuple frequencies) frames of possible interest. Consider the case of triplets for example. These frames will be denoted 1, 2, 3, -1, -2, and -3. For an intron 200 bases long, index the bases 1, 2, ..., 200. Then the six frames are as follows:

frame	frequencies using the following triplets	
1	1, 2, 3	4, 5, 6 ...
2	2, 3, 4	5, 6, 7 ...
3	3, 4, 5	6, 7, 8 ...
-1	... 195, 196, 197	198, 199, 200
-2	... 194, 195, 196	197, 198, 199
-3	... 193, 194, 195	196, 197, 198

In any one intron, frames -1, -2 and -3 always correspond (element-wise) to the frames 1, 2, and 3 as follows:

if the intron is length 0 mod 3, $\{1, 2, 3\} \rightarrow \{-1, -3, -2\}$

if the intron is length 1 mod 3, $\{1, 2, 3\} \rightarrow \{-2, -1, -3\}$

if the intron is length 2 mod 3, $\{1, 2, 3\} \rightarrow \{-3, -2, -1\}$

Significant triplet frequency differences could emerge between all six frames when the frequencies from all introns are examined. For example, if introns start or end with a particular sequence, the frequency in one of the frames could be significantly altered. In this study however, frequencies computed from all six frames yielded almost identical results – see figure 3.4. A two-factor fixed effects ANOVA (6 frames and 64 triplets) gave P values of 0.9067 and 7.4712^{-379} for the null hypotheses of no effect of frame and triplet, respectively. Thus we would reject the latter hypothesis, but not the former.

Tables 2.1, 2.2, and 2.3 show our computed triplet frequencies for codons, introns, and lambda values respectively for the 64 triplets. Our most notable finding is that all 8 intron triplets containing the sequence CG have less than 1/5 the average frequency of the other 56 intron triplets, and indeed have the 8 lowest frequencies of the 64 triplets. In one thousand randomly generated sequences of the same length as the total number of intron nucleotides in our database for example, the lowest dinucleotide frequency was never as low as the CG frequency observed in our dataset. This initially surprising result might be explained by the fact that the DNA dinucleotide CG often has a methylated C. That is, the C is chemically modified, and this modification

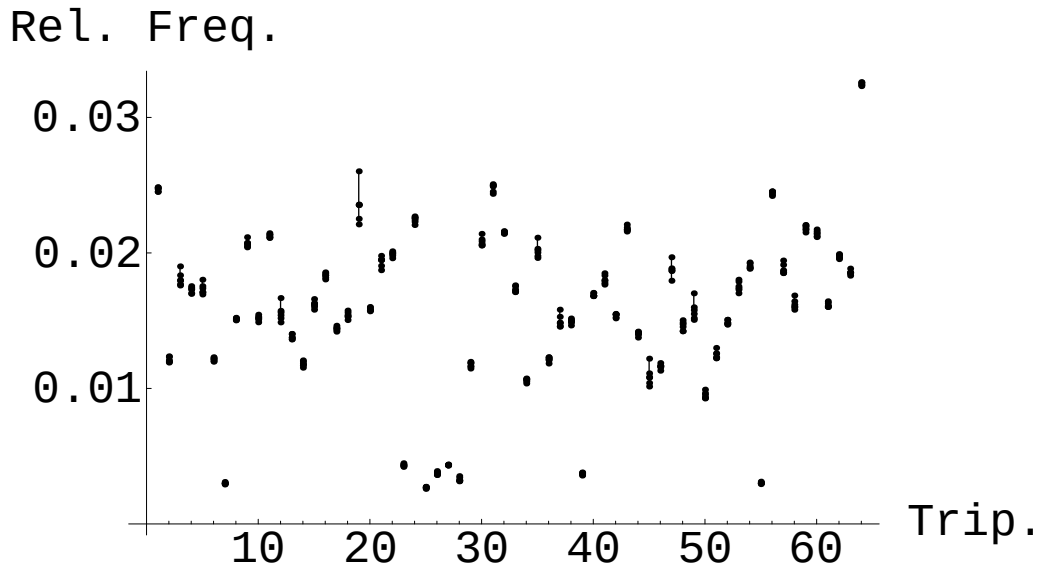

Figure 2.8: Vertical axis – relative frequency of intron triplets, horizontal axis – the 64 triplets in alphabetical order. Frequencies of all six frames are superimposed with vertical line indicating range. Note the similarity of frequencies in all frames.

results in a C which is more easily spontaneously mutated into a T. Since mutations in an intron cause no phenotypic change in an individual or its descendants, a CG to TG mutated intron is more likely to spread through the population than a similar mutation in an exon (where the mutations tend to be detrimental to the individual). This may explain why the CG dinucleotide is relatively rarer among introns than it is among exons (the mutation is "selected against" in exons, but not in introns). These 8 CG containing triplets give some of the largest likelihood ratios, and are thus very helpful in distinguishing between exons and introns (five of the six largest lambdas have a CG dinucleotide in them), see table 2.3. We say they have a strong (positive) signal.

Stop codons give a strong negative signal. That is, they have large negative values for lambda. This is because only one stop codon occurs in a gene, at the end of the last exon, and are thus expected to be rare in exons. Our results (table 2.1) show that stop codons have the three lowest codon frequencies by far – an average frequency of about 1/20 the other 61 codons. Similarly to the CG frequency simulations, we again generated one thousand random sequences of the same length as the total number of exon nucleotides in our database, and the lowest trinucleotide frequency was never as low as any of the stop codon frequencies observed in our dataset. These three triplets give the strongest signal (almost 3 times the strength of the next strongest triplet).

In general, the more positive values of lambda indicate more evidence that a subsequence seq_{wl} is from an exon, and that the sequence is being

aaa	1	23.6	gaa	33	28.6	tag	51	0.5	gtc	46	14.8
aac	2	20.2	gac	34	26.5	taa	49	0.7	cac	18	14.9
aag	3	33.2	gag	35	40.6	tga	57	1.3	gca	37	15.6
aat	4	16.8	gat	36	22.2	tcg	55	4.5	att	16	15.7
aca	5	14.7	gca	37	15.6	cgt	28	4.7	tac	50	16.3
acc	6	19.9	gcc	38	28.6	cga	25	6.2	gga	41	16.4
acg	7	6.4	gcg	39	7.7	acg	7	6.4	cca	21	16.5
act	8	12.7	gct	40	18.4	cta	29	6.8	ggg	43	16.5
aga	9	11.2	gga	41	16.4	gta	45	6.8	ttt	64	16.6
agc	10	19.3	ggc	42	23.1	ata	13	7.	aat	4	16.8
agg	11	11.1	ggg	43	16.5	tta	61	7.	cct	24	17.2
agt	12	11.7	ggt	44	10.9	ccg	23	7.1	tcc	54	17.7
ata	13	7.	gta	45	6.8	gcg	39	7.7	gct	40	18.4
atc	14	22.3	gtc	46	14.8	tgt	60	9.7	agc	10	19.3
atg	15	22.2	gtg	47	29.3	cat	20	10.1	ctc	30	19.3
att	16	15.7	gtt	48	10.7	gtt	48	10.7	acc	6	19.9
caa	17	11.8	taa	49	0.7	ggt	44	10.9	aac	2	20.2
cac	18	14.9	tac	50	16.3	cgc	26	11.	ccc	22	20.3
cag	19	34.4	tag	51	0.5	agg	11	11.1	ttc	62	20.7
cat	20	10.1	tat	52	12.1	aga	9	11.2	atg	15	22.2
cca	21	16.5	tca	53	11.4	tca	53	11.4	gat	36	22.2
ccc	22	20.3	tcc	54	17.7	cgg	27	11.6	atc	14	22.3
ccg	23	7.1	tcg	55	4.5	agt	12	11.7	ggc	42	23.1
cct	24	17.2	tct	56	14.5	caa	17	11.8	aaa	1	23.6
cga	25	6.2	tga	57	1.3	ttg	63	12.	gac	34	26.5
cgc	26	11.	tgc	58	12.4	tat	52	12.1	gaa	33	28.6
cgg	27	11.6	tgg	59	13.	ctt	32	12.4	gcc	38	28.6
cgt	28	4.7	tgt	60	9.7	tgc	58	12.4	gtg	47	29.3
cta	29	6.8	tta	61	7.	act	8	12.7	aag	3	33.2
ctc	30	19.3	ttc	62	20.7	tgg	59	13.	cag	19	34.4
ctg	31	40.	ttg	63	12.	tct	56	14.5	ctg	31	40.
ctt	32	12.4	ttt	64	16.6	aca	5	14.7	gag	35	40.6

a b

Table 2.1: Codons, alphabetical ranking and frequency ranking. a) ranked alphabetically b) ranked by frequency.

aaa	1	30.4	gaa	33	17.2	cga	25	2.5	atg	15	16.7
aac	2	12.	gac	34	9.6	acg	7	2.7	cat	20	16.7
aag	3	17.5	gag	35	18.8	tcg	55	2.8	tcc	54	16.9
aat	4	20.1	gat	36	13.	cgc	26	3.2	agt	12	16.9
aca	5	17.	gca	37	14.7	cgt	28	3.2	aca	5	17.
acc	6	12.	gcc	38	14.8	gcg	39	3.4	tat	52	17.2
acg	7	2.7	gcg	39	3.4	cgg	27	3.8	gaa	33	17.2
act	8	15.	gct	40	15.7	ccg	23	3.8	taa	49	17.4
aga	9	20.3	gga	41	16.6	gac	34	9.6	ggg	43	17.4
agc	10	15.	ggc	42	14.9	tac	50	10.1	aag	3	17.5
agg	11	19.9	ggg	43	17.4	gtc	46	10.5	gtg	47	18.3
agt	12	16.9	ggt	44	13.4	cta	29	11.4	tca	53	18.4
ata	13	16.2	gta	45	11.7	gta	45	11.7	ctc	30	18.8
atc	14	12.	gtc	46	10.5	aac	2	12.	gag	35	18.8
atg	15	16.7	gtg	47	18.3	atc	14	12.	ttc	62	19.1
att	16	22.3	gtt	48	15.4	acc	6	12.	tta	61	19.1
caa	17	15.	taa	49	17.4	tag	51	12.8	cca	21	19.1
cac	18	15.	tac	50	10.1	gat	36	13.	tga	57	19.3
cag	19	22.	tag	51	12.8	ggt	44	13.4	ttg	63	19.9
cat	20	16.7	tat	52	17.2	gca	37	14.7	agg	11	19.9
cca	21	19.1	tca	53	18.4	gcc	38	14.8	aat	4	20.1
ccc	22	16.5	tcc	54	16.9	ggc	42	14.9	aga	9	20.3
ccg	23	3.8	tcg	55	2.8	agc	10	15.	ctt	32	20.4
cct	24	20.7	tct	56	22.4	caa	17	15.	cct	24	20.7
cga	25	2.5	tga	57	19.3	cac	18	15.	tgg	59	21.3
cgc	26	3.2	tgc	58	15.7	act	8	15.	tgt	60	21.5
cgg	27	3.8	tgg	59	21.3	gtt	48	15.4	cag	19	22.
cgt	28	3.2	tgt	60	21.5	tgc	58	15.7	att	16	22.3
cta	29	11.4	tta	61	19.1	gct	40	15.7	tct	56	22.4
ctc	30	18.8	ttc	62	19.1	ata	13	16.2	ctg	31	23.3
ctg	31	23.3	ttg	63	19.9	ccc	22	16.5	aaa	1	30.4
ctt	32	20.4	ttt	64	39.1	gga	41	16.6	ttt	64	39.1

a b

Table 2.2: Intron triplets, alphabetical ranking and frequency ranking. a) ranked alphabetically b) ranked by frequency.

aaa	1	-0.11	gaa	33	0.22	tag	51	-1.41	cac	18	0.
aac	2	0.23	gac	34	0.44	taa	49	-1.39	ctc	30	0.01
aag	3	0.28	gag	35	0.33	tga	57	-1.17	tcc	54	0.02
aat	4	-0.08	gat	36	0.23	tta	61	-0.44	gca	37	0.03
aca	5	-0.06	gca	37	0.03	ttt	64	-0.37	ttc	62	0.04
acc	6	0.22	gcc	38	0.29	ata	13	-0.37	gct	40	0.07
acg	7	0.37	gcg	39	0.36	tgt	60	-0.35	ccc	22	0.09
act	8	-0.07	gct	40	0.07	aga	9	-0.26	agc	10	0.11
aga	9	-0.26	gga	41	-0.01	agg	11	-0.25	atg	15	0.12
agc	10	0.11	ggc	42	0.19	gta	45	-0.23	gtc	46	0.15
agg	11	-0.25	ggg	43	-0.02	cta	29	-0.23	cgt	28	0.16
agt	12	-0.16	ggt	44	-0.09	ttg	63	-0.22	ggc	42	0.19
ata	13	-0.37	gta	45	-0.23	cat	20	-0.22	cag	19	0.19
atc	14	0.27	gtc	46	0.15	ctt	32	-0.22	gtg	47	0.2
atg	15	0.12	gtg	47	0.2	tgg	59	-0.21	tac	50	0.21
att	16	-0.15	gtt	48	-0.16	tca	53	-0.21	tcg	55	0.21
caa	17	-0.1	taa	49	-1.39	tct	56	-0.19	acc	6	0.22
cac	18	0.	tac	50	0.21	agt	12	-0.16	gaa	33	0.22
cag	19	0.19	tag	51	-1.41	gtt	48	-0.16	aac	2	0.23
cat	20	-0.22	tat	52	-0.15	tat	52	-0.15	gat	36	0.23
cca	21	-0.06	tca	53	-0.21	att	16	-0.15	ctg	31	0.24
ccc	22	0.09	tcc	54	0.02	aaa	1	-0.11	ccg	23	0.27
ccg	23	0.27	tcg	55	0.21	caa	17	-0.1	atc	14	0.27
cct	24	-0.08	tct	56	-0.19	tgc	58	-0.1	aag	3	0.28
cga	25	0.39	tga	57	-1.17	ggt	44	-0.09	gcc	38	0.29
cgc	26	0.54	tgc	58	-0.1	cct	24	-0.08	gag	35	0.33
cgg	27	0.49	tgg	59	-0.21	aat	4	-0.08	gcg	39	0.36
cgt	28	0.16	tgt	60	-0.35	act	8	-0.07	acg	7	0.37
cta	29	-0.23	tta	61	-0.44	cca	21	-0.06	cga	25	0.39
ctc	30	0.01	ttc	62	0.04	aca	5	-0.06	gac	34	0.44
ctg	31	0.24	ttg	63	-0.22	ggg	43	-0.02	cgg	27	0.49
ctt	32	-0.22	ttt	64	-0.37	gga	41	-0.01	cgc	26	0.54

a b

Table 2.3: Lambdas, alphabetical ranking and value ranking. a) ranked alphabetically b) ranked lowest to highest.

read in the correct reading frame. Increasingly negative values of lambda are stronger evidence that seq_{wl} is from an intron. This window is slid along the DNA sequence one nucleotide at a time, with a lambda calculated for each consecutive wl nucleotides. These lambdas are calculated for all contiguous subsequences of length wl, and thus a sequence of length L would yield L-wl+1 lambdas.

Consecutive lambdas have little correlation, as there is a frame shift from one to the next, but there is a strong correlation between every third lambda because they have wl/3 -1 triplets in common – one triplet is lost from the window, and a new one is added. Thus we use three separate plots to display the lambdas for a sequence. The first plot shows lambdas for the positions one, four, seven etc., the second plot shows lambdas for positions two, five, eight, etc., and the third plot shows the lambdas for the remaining positions. Ideally, in a long sequence with introns in front of and behind an exon, the exon signal would be strongest in the plot corresponding to its reading frame. As the window moves into the exon, a series of lambdas would start to grow until the window was fully into the exon, and later diminish as the window moves out of the exon.

Figure 2.9 shows an idealized case with a sequence of 30 CGCs (the most exon-like triplet) followed by 50 TAGs (the most intron-like triplet); these 80 triplets are then repeated. The peaks in each plot indicate the possible presence of exons. Spurious peaks occur, however, when the window is in an intron region. Similarly, some exons yield surprisingly low values of lambda.

In the next chapter, we give the procedures and results using the afore-

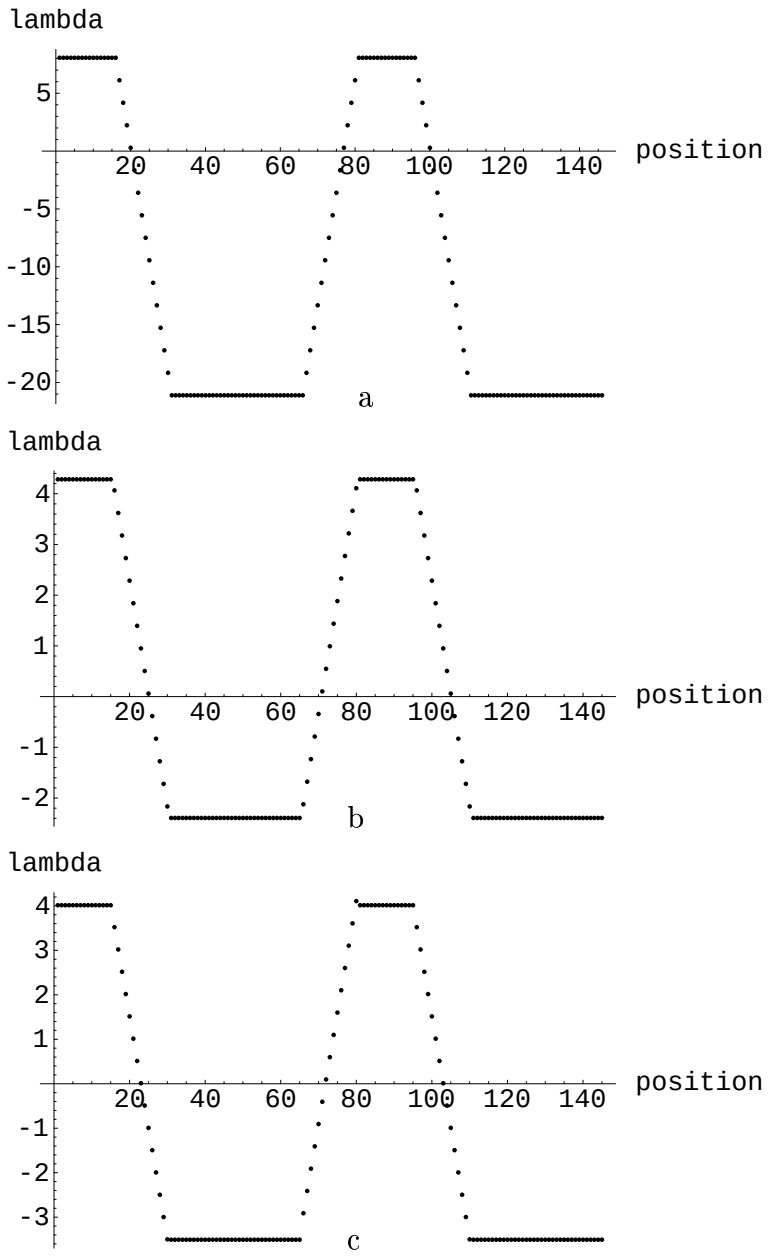

Figure 2.9: Idealized exon, intron, exon, intron sequence. Plots a, b, and c show the lambdas corresponding to starting at position one, two, and three respectively. Exons at positions 1 - 30 and 81 - 110, introns elsewhere.

mentioned method of n-tuple frequencies and likelihood ratios. We give the sensitivities (the number of true positives the method detects divided by *all* the positives it detects) and specificities (the number of true negatives the method detects divided by *all* the negatives it detects) of the method with n set equal to one, two and three. Finally, we explore an extension to the triplet case (the most successful) which we call the multi-window method. This extension uses the DNA data from three consecutive overlapping subsequences, and takes into account the cyclic nature of the frames in exons.

2.6 Discussion

Many current programs that annotate DNA sequence data use multiple techniques. These include hidden Markov models, neural networks, and maximal dependence decomposition, along with similarity matching techniques, and signal searches. Some methods also incorporate the full annotation efforts of multiple programs [Murakami and Takagi, 1998], [Rogic et al., 2002]. Apparently, integrated techniques that use a variety of information tend to achieve more accurate results than those that use only a single method [Rogic et al., 2001], [Rogic et al., 2002].

As more genes with canonical features (promoter elements, splice sites etc.) are found, more emphasis must be placed on locating genes with non-canonical features. Progress must also be made in the biological areas of signal recognition. Computational biology should not only help answer these questions, but should help decide which questions to address. Existing or

new procedures should be able to incorporate these new findings to further the progress in computational gene identification.

The method introduced here, using the maximum likelihood ratio with n-tuple frequencies, is an additional tool in our attempt to annotate genomes. Its incorporation into existing programs should increase overall sensitivity and specificity.

Bibliography

[Agatonovic-Kustrin and Beresford, 2000] Agatonovic-Kustrin, S. and Beresford, R. (2000). Basic concepts of artificial neural network (ANN) modeling and its application in pharmaceutical research. *J Pharm Biomed Anal*, 22(5):717–727.

[Alexandersson et al., 2003] Alexandersson, M., Cawley, S., and Pachter, L. (2003). SLAM: cross-species gene finding and alignment with a generalized pair hidden Markov model. *Genome Res*, 13(3):496–502.

[Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *J Mol Biol*, 215(3):403–410.

[Altschul et al., 1997] Altschul, S. F., Madden, T. L., Schaffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402.

[Baum, 1972] Baum, L. E. (1972). An inequality and association maximization technique in statistical estimation for probabilistic functions of a Markov process. *Inequalities*, 3:1–8.

[Benson et al., 2000] Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., Rapp, B. A., and Wheeler, D. L. (2000). GenBank. *Nucleic Acids Res*, 28(1):15–18.

[Boffelli et al., 2003] Boffelli, D., McAuliffe, J., Ovcharenko, D., Lewis, K. D., Ovcharenko, I., Pachter, L., and Rubin, E. M. (2003). Phylogenetic shadowing of primate sequences to find functional regions of the human genome. *Science*, 299(5611):1391–1394.

[Bray et al., 2003] Bray, N., Dubchak, I., and Pachter, L. (2003). AVID: A global alignment program. *Genome Res*, 13(1):97–102.

[Brett et al., 2000] Brett, D., Hanke, J., Lehmann, G., Haase, S., Delbruck, S., Krueger, S., Reich, J., and Bork, P. (2000). EST comparison indicates 38alternative splice forms. *FEBS Lett*, 474(1):83–86.

[Brudno et al., 2003] Brudno, M., Do, C. B., Cooper, G. M., Kim, M. F., Davydov, E., Green, E. D., Sidow, A., and Batzoglou, S. (2003). LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res*, 13(4):721–731. Evaluation Studies.

[Burge and Karlin, 1997] Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *J Mol Biol*, 268(1):78–94.

[Claverie, 1997] Claverie, J. M. (1997). Computational methods for the identification of genes in vertebrate genomic sequences. *Hum Mol Genet*, 6(10):1735–1744.

[Couronne et al., 2003] Couronne, O., Poliakov, A., Bray, N., Ishkhanov, T., Ryaboy, D., Rubin, E., Pachter, L., and Dubchak, I. (2003). Strategies and tools for whole-genome alignments. *Genome Res*, 13(1):73–80.

[Davies et al., 1987] Davies, K. A., Lorand, L., Waterfield, M., Wainwright, B., Farrall, M., and Williamson, R. (1987). Isolation of a polymorphic genomic clone from chromosome 7. Physical and genetic linkage studies to markers around the cystic fibrosis locus. *Hum Genet*, 77(2):122–126.

[Davies et al., 1982] Davies, R. W., Waring, R. B., Ray, J. A., Brown, T. A., and Scazzocchio, C. (1982). Making ends meet: a model for RNA splicing in fungal mitochondria. *Nature*, 300(5894):719–724.

[Edwards, 1972] Edwards, A. W. F. (1972). *Likelihood*. Cambridge University Press, New York, NY.

[Fairbanks and Anderson, 1999] Fairbanks, D. J. and Anderson, R. W. (1999). *Genetics*. Brooks/Cole.

[Forney, 1973] Forney, G. D. (1973). The Viterbi algorith. *Proceedings of the IEEE*, 61:268–278.

[Gibbs and Nelson, 2003] Gibbs, R. A. and Nelson, D. L. (2003). Human genetics. Primate shadow play. *Science*, 299(5611):1331–1333. Comment.

[Gilbert, 1978] Gilbert, W. (1978). Why genes in pieces. *Nature*, 271(5645):501.

[Guigo et al., 2000] Guigo, R., Agarwal, P., Abril, J. F., Burset, M., and Fickett, J. W. (2000). An assessment of gene prediction accuracy in large DNA sequences. *Genome Res*, 10(10):1631–1642.

[Guigo et al., 2003] Guigo, R., Dermitzakis, E. T., Agarwal, P., Ponting, C. P., Parra, G., Reymond, A., Abril, J. F., Keibler, E., Lyle, R., Ucla, C., Antonarakis, S. E., and Brent, M. R. (2003). Comparison of mouse and human genomes followed by experimental verification yields an estimated 1,019 additional genes. *Proc Natl Acad Sci U S A*, 100(3):1140–1145.

[Gumucio et al., 1992] Gumucio, D. L., Heilstedt-Williamson, H., Gray, T. A., Tarle, S. A., Shelton, D. A., Tagle, D. A., Slightom, J. L., Goodman, M., and Collins, F. S. (1992). Phylogenetic footprinting reveals a nuclear protein which binds to silencer sequences in the human gamma and epsilon globin genes. *Mol Cell Biol*, 12(11):4919–4929.

[Kleffe et al., 1996] Kleffe, J., Hermann, K., Vahrson, W., Wittig, B., and Brendel, V. (1996). Logitlinear models for the prediction of splice sites in plant pre-mRNA sequences. *Nucleic Acids Res*, 24(23):4709–4718.

[Krogh, 1998] Krogh, A. (1998). *Gene finding: putting the parts together.* Academic Press, 2nd edition.

[Lander et al., 2001] Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh,

W., Funke, R., Gage, D., Harris, K., Heaford, A., Howland, J., Kann, L., Lehoczky, J., LeVine, R., McEwan, P., McKernan, K., Meldrim, J., Mesirov, J. P., Miranda, C., Morris, W., Naylor, J., Raymond, C., Rosetti, M., Santos, R., Sheridan, A., Sougnez, C., Stange-Thomann, N., Stojanovic, N., Subramanian, A., Wyman, D., Rogers, J., Sulston, J., Ainscough, R., Beck, S., Bentley, D., Burton, J., Clee, C., Carter, N., Coulson, A., Deadman, R., Deloukas, P., Dunham, A., Dunham, I., Durbin, R., French, L., Grafham, D., Gregory, S., Hubbard, T., Humphray, S., Hunt, A., Jones, M., Lloyd, C., McMurray, A., Matthews, L., Mercer, S., Milne, S., Mullikin, J. C., Mungall, A., Plumb, R., Ross, M., Shownkeen, R., Sims, S., Waterston, R. H., Wilson, R. K., Hillier, L. W., McPherson, J. D., Marra, M. A., Mardis, E. R., Fulton, L. A., Chinwalla, A. T., Pepin, K. H., Gish, W. R., Chissoe, S. L., Wendl, M. C., Delehaunty, K. D., Miner, T. L., Delehaunty, A., Kramer, J. B., Cook, L. L., Fulton, R. S., Johnson, D. L., Minx, P. J., Clifton, S. W., Hawkins, T., Branscomb, E., Predki, P., Richardson, P., Wenning, S., Slezak, T., Doggett, N., Cheng, J. F., Olsen, A., Lucas, S., Elkin, C., Uberbacher, E., Frazier, M., Gibbs, R. A., Muzny, D. M., Scherer, S. E., Bouck, J. B., Sodergren, E. J., Worley, K. C., Rives, C. M., Gorrell, J. H., Metzker, M. L., Naylor, S. L., Kucherlapati, R. S., Nelson, D. L., Weinstock, G. M., Sakaki, Y., Fujiyama, A., Hattori, M., Yada, T., Toyoda, A., Itoh, T., Kawagoe, C., Watanabe, H., Totoki, Y., Taylor, T., Weissenbach, J., Heilig, R., Saurin, W., Artiguenave, F., Brottier, P., Bruls, T., Pelletier, E., Robert, C., Wincker, P., Smith, D. R., Doucette-Stamm, L., Rubenfield, M., Weinstock, K., Lee,

H. M., Dubois, J., Rosenthal, A., Platzer, M., Nyakatura, G., Taudien, S., Rump, A., Yang, H., Yu, J., Wang, J., Huang, G., Gu, J., Hood, L., Rowen, L., Madan, A., Qin, S., Davis, R. W., Federspiel, N. A., Abola, A. P., Proctor, M. J., Myers, R. M., Schmutz, J., Dickson, M., Grimwood, J., Cox, D. R., Olson, M. V., Kaul, R., Raymond, C., Shimizu, N., Kawasaki, K., Minoshima, S., Evans, G. A., Athanasiou, M., Schultz, R., Roe, B. A., Chen, F., Pan, H., Ramser, J., Lehrach, H., Reinhardt, R., McCombie, W. R., de la Bastide, M., Dedhia, N., Blocker, H., Hornischer, K., Nordsiek, G., Agarwala, R., Aravind, L., Bailey, J. A., Bateman, A., Batzoglou, S., Birney, E., Bork, P., Brown, D. G., Burge, C. B., Cerutti, L., Chen, H. C., Church, D., Clamp, M., Copley, R. R., Doerks, T., Eddy, S. R., Eichler, E. E., Furey, T. S., Galagan, J., Gilbert, J. G., Harmon, C., Hayashizaki, Y., Haussler, D., Hermjakob, H., Hokamp, K., Jang, W., Johnson, L. S., Jones, T. A., Kasif, S., Kaspryzk, A., Kennedy, S., Kent, W. J., Kitts, P., Koonin, E. V., Korf, I., Kulp, D., Lancet, D., Lowe, T. M., McLysaght, A., Mikkelsen, T., Moran, J. V., Mulder, N., Pollara, V. J., Ponting, C. P., Schuler, G., Schultz, J., Slater, G., Smit, A. F., Stupka, E., Szustakowski, J., Thierry-Mieg, D., Thierry-Mieg, J., Wagner, L., Wallis, J., Wheeler, R., Williams, A., Wolf, Y. I., Wolfe, K. H., Yang, S. P., Yeh, R. F., Collins, F., Guyer, M. S., Peterson, J., Felsenfeld, A., Wetterstrand, K. A., Patrinos, A., Morgan, M. J., de Jong, P., Catanese, J. J., Osoegawa, K., Shizuya, H., Choi, S., Chen, Y. J., and Szustakowki, J. (2001). Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921.

[Lewin, 1994] Lewin, B. (1994). *Genes V.* Oxford University Press.

[Lewin, 2000] Lewin, B. (2000). *Genes VII.* Oxford University Press.

[Livstone et al., 2003] Livstone, M. S., van Noort, D., and Landweber, L. F. (2003). Molecular computing revisited: a Moore's Law? *Trends Biotechnol*, 21(3):98–101.

[Mathe et al., 2002] Mathe, C., Sagot, M.-F., Schiex, T., and Rouze, P. (2002). Current methods of gene prediction, their strengths and weaknesses. *Nucleic Acids Res*, 30(19):4103–4117.

[Mironov et al., 1999] Mironov, A. A., Fickett, J. W., and Gelfand, M. S. (1999). Frequent alternative splicing of human genes. *Genome Res*, 9(12):1288–1293.

[Murakami and Takagi, 1998] Murakami, K. and Takagi, T. (1998). Gene recognition by combination of several gene-finding programs. *Bioinformatics*, 14(8):665–675.

[Ohler and Niemann, 2001] Ohler, U. and Niemann, H. (2001). Identification and analysis of eukaryotic promoters: recent computational approaches. *Trends Genet*, 17(2):56–60.

[Ohler et al., 2000] Ohler, U., Stemmer, G., Harbeck, S., and Niemann, H. (2000). Stochastic segment models of eukaryotic promoter regions. *Pac Symp Biocomput*, pages 380–391.

[Pedersen et al., 1999] Pedersen, A., Baldi, P., Chauvin, Y., and Brunak, S. (1999). The biology of eukaryotic promoter prediction - a review.

[Pertea et al., 2001] Pertea, M., Lin, X., and Salzberg, S. L. (2001). GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Res*, 29(5):1185–1190.

[Rabiner, 1989] Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Reese et al., 1997] Reese, M. G., Eeckman, F. H., Kulp, D., and Haussler, D. (1997). Improved splice site detection in Genie. *J Comput Biol*, 4(3):311–323.

[Reese et al., 2000] Reese, M. G., Kulp, D., Tammana, H., and Haussler, D. (2000). Genie–gene finding in Drosophila melanogaster. *Genome Res*, 10(4):529–538.

[Riordan et al., 1989] Riordan, J. R., Rommens, J. M., Kerem, B., Alon, N., Rozmahel, R., Grzelczak, Z., Zielenski, J., Lok, S., Plavsic, N., and Chou, J. L. (1989). Identification of the cystic fibrosis gene: cloning and characterization of complementary DNA. *Science*, 245(4922):1066–1073.

[Rogic et al., 2001] Rogic, S., Mackworth, A. K., and Ouellette, F. B. (2001). Evaluation of gene-finding programs on mammalian sequences. *Genome Res*, 11(5):817–832.

[Rogic et al., 2002] Rogic, S., Ouellette, B. F. F., and Mackworth, A. K. (2002). Improving gene recognition accuracy by combining predictions from two gene-finding programs. *Bioinformatics*, 18(8):1034–1045.

[Rogozin and Milanesi, 1997] Rogozin, I. B. and Milanesi, L. (1997). Analysis of donor splice sites in different eukaryotic organisms. *J Mol Evol*, 45(1):50–59.

[Rommens et al., 1989] Rommens, J. M., Iannuzzi, M. C., Kerem, B., Drumm, M. L., Melmer, G., Dean, M., Rozmahel, R., Cole, J. L., Kennedy, D., and Hidaka, N. (1989). Identification of the cystic fibrosis gene: chromosome walking and jumping. *Science*, 245(4922):1059–1065.

[Russell, 1998] Russell, P. J. (1998). *Genetics*. Benjamin/Cummings, an imprint of Addison-Wesley Longman, Inc., Fifth edition.

[Saxonov et al., 2000] Saxonov, S., Daizadeh, I., Fedorov, A., and Gilbert, W. (2000). EID: the Exon-Intron Database-an exhaustive database of protein-coding intron-containing genes. *Nucleic Acids Res*, 28(1):185–190.

[Snustad et al., 1997] Snustad, P. D., Simmons, M. J., and Jenkins, J. B. (1997). *Principles of Genetics*. Wiley.

[Snyder and Stormo, 1993] Snyder, E. E. and Stormo, G. D. (1993). Identification of coding regions in genomic DNA sequences: an application of dynamic programming and neural networks. *Nucleic Acids Res*, 21(3):607–613.

[Snyder and Stormo, 1995] Snyder, E. E. and Stormo, G. D. (1995). Identification of protein coding regions in genomic DNA. *J Mol Biol*, 248(1):1–18.

[Sridhar, 2001] Sridhar, G. R. (2001). Impact of human genome project on medical practice. *J Assoc Physicians India*, 49:995–998. Historical Article.

[Staden, 1984] Staden, R. (1984). Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res*, 12(1 Pt 2):505–519.

[Sunyaev et al., 2003] Sunyaev, S., Kondrashov, F. A., Bork, P., and Ramensky, V. (2003). Impact of selection, mutation rate and genetic drift on human genetic variation. *Hum Mol Genet*, 12(24):3325–3330.

[Tagle et al., 1988] Tagle, D. A., Koop, B. F., Goodman, M., Slightom, J. L., Hess, D. L., and Jones, R. T. (1988). Embryonic epsilon and gamma globin genes of a prosimian primate (Galago crassicaudatus). Nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints. *J Mol Biol*, 203(2):439–455.

[Tolstrup et al., 1997] Tolstrup, N., Rouze, P., and Brunak, S. (1997). A branch point consensus from Arabidopsis found by non-circular analysis allows for better prediction of acceptor sites. *Nucleic Acids Res*, 25(15):3159–3163.

[Townsend et al., 2004] Townsend, T., Larson, A., Louis, E., and Macey, J. (2004). Molecular phylogenetics of squamata: the position of snakes, amphisbaenians, and dibamids, and the root of the squamate tree. *Syst Biol*, 53(5):735–757.

[Uberbacher and Mural, 1991] Uberbacher, E. C. and Mural, R. J. (1991). Locating protein-coding regions in human DNA sequences by a multiple sensor-neural network approach. *Proc Natl Acad Sci U S A*, 88(24):11261–11265.

[Viterbi, 1967] Viterbi, A. W. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Infromat. Theory*, IT-13:260–269.

[Waring and Davies, 1984] Waring, R. B. and Davies, R. W. (1984). Assessment of a model for intron RNA secondary structure relevant to RNA self-splicing–a review. *Gene*, 28(3):277–291.

[Watson and Crick, 1953] Watson, J. D. and Crick, F. H. C. (1953). A structure for Deoxyribose Nucleic Acid. *Nature*, 171:737.

[Won and Hey, 2005] Won, Y.-J. and Hey, J. (2005). Divergence population genetics of chimpanzees. *Mol Biol Evol*, 22(2):297–307.

[Zhang and Zhang, 2002] Zhang, C.-T. and Zhang, R. (2002). Evaluation of gene-finding algorithms by a content-balancing accuracy index. *J Biomol Struct Dyn*, 19(6):1045–1052. Evaluation Studies.

[Zhang and Marr, 1993] Zhang, M. Q. and Marr, T. G. (1993). A weight array method for splicing signal analysis. *Comput Appl Biosci*, 9(5):499–509.

[Zhang et al., 1998] Zhang, Z., Schaffer, A. A., Miller, W., Madden, T. L., Lipman, D. J., Koonin, E. V., and Altschul, S. F. (1998). Protein sequence similarity searches using patterns as seeds. *Nucleic Acids Res*, 26(17):3986–3990.

Chapter 3

Sensitivity and Specificity of Exon Detection using Likelihood Ratios

3.1 Abstract

Our initial approach of distinguishing between exon and intron regions using consecutive windows of nucleotides, and predicting the character of the sequence based solely on the likelihood ratio of this subsequence gave low sensitivities and specificities. Herein a new method is developed using multiple windows, the frequencies of nucleotide triplets in all three frames of exons as well as triplet frequencies in introns, and the ordering of the three frames in exons. This new method gives substantially better results.

3.2 Introduction

There are various ways of analyzing the performance of a gene or exon/intron detection program [Zhang and Zhang, 2002]. "Correct" detection can be stringently defined as correctly identifying the entire gene with exact exon/intron boundaries identified at every donor and acceptor splice site [Burge and Karlin, 1997]. Alternative splicing of many genes makes this an ill-posed problem as the construct of the gene product is variable. One then may look for the most frequent splicing of the gene, or one of the known splicings. One can also consider exon/intron detection on an individual exon/intron basis [Zhang and Zhang, 2002]. That is, if the splice sites to either side of an exon/intron are correctly identified, then this exon/intron has been correctly identified.

Single nucleotides are also often considered when gauging a method's sensitivity and specificity [Zhang and Zhang, 2002]. In this case, sensitivity is equal to the number of true positives the method detects (a nucleotide which is correctly predicted to be in an exon) divided by all the positives it detects (the true positives *plus* the false positives). Similarly, specificity is equal to the number of true negatives (similarly, a nucleotide which is correctly predicted to not be in an exon) divided by all of the negatives. The method we develop below does not have the resolution to determine exact splice sites, so we use this single nucleotide definition of sensitivity and specificity.

Our original method (Chapter 2) used only a single window of nucleotides

from the sequence to characterize the subsequence as coming from an exon or intron. Consecutive, overlapping windows of a given length were analyzed individually by counting the number of consecutive, nonoverlapping n-tuples in the window and calculating the likelihood ratio

$$\lambda = \log \frac{L(H_e|Data)}{L(H_i|Data)} \tag{3.1}$$

where H_e and H_i are the hypotheses that the data came from an exon or intron region respectively. Sir Ronald Fisher's concept of "likelihood" and its history is reviewed in [Edwards, 1972].

This "single window" method uses the sign of λ in equation 3.1 to predict if the first nucleotide of the data is and exon (positive λ) or an intron (negative λ) nucleotide. It achieved very limited success with exon sensitivity/specificity for the mono, di, and tri nucleotide, respectively, as follows: .731851/.638492, .731516/.656525, and .562256/.815566. There were far too many spuriously high likelihood ratios in intron regions and low likelihood ratios within exons. The ability of the single window method to delineate the splice sites was very low and it did not obtain high nucleotide sensitivities or specificities. Therefore a new method was developed.

This new "multi-window" method uses the codon frequencies (used in the $L(H_e|Data)$ above) and the nucleotide triplet frequencies in the introns, but it also takes into account the nucleotide triplet frequencies within the exons that are not in the reading frame used for protein synthesis – i.e. the two non-codon frames. In addition to considering these two new triplet frequencies, the newer method analyzes three consecutive windows which are each shifted

from one another by a single nucleotide, and compares various likelihood ratios and their sums. The multi-window new method makes a prediction on a single nucleotide basis as to whether the nucleotide came from the first, second, or third base (nucleotide) of a codon, or if the nucleotide came from an intron. It attained higher accuracy than the single window method.

3.3 Biological Background

A brief summary of the pertinent biology is given here. For a more detailed account the reader is referred to the Biological Background section in Chapter 2. The following books also give further information on the topic: [Lewin, 1994], [Fairbanks and Anderson, 1999] and [Snustad et al., 1997].

Human chromosomes are composed of tightly coiled threads of deoxyribonucleic acid (DNA) and associated protein molecules which aid in the structural packing of the DNA. The DNA itself is often compared to a twisted ladder with the sides of the ladder being the sugar-phosphate backbone of the DNA, and the rungs being the two complementary nucleotides that bind to one another - one from each of the two strands of DNA [Watson and Crick, 1953]. A single strand of DNA may be thought of as a sequence of four nucleotides: adenine (A), cytosine (C), guanine (G) and thymine (T). The nucleotides that bind to one another to form the "rungs" are called complementary pairs: A binds with T and C binds with G.

The DNA is always read by the cell machinery in the same orientation. That is, the sequence AATCGTA of nucleotides (bases) along a strand of

DNA would always be read in the order indicated above, or in the reverse as ATGCTAA, but not in both orders. The end of the sequence, where the reading starts, is the 5' end and the other is the 3' end. The complementary strand always has the reverse orientation. Thus if one strand of a chromosome had the sequence 5'- AATCGTA - 3', then this would be bound to the sequence 3' - TTAGCAT - 5'.

The genes within the DNA are the genetic code used by the cell to make proteins. In higher eukaryotes these genes comprise only a small percentage of the entire genome – the entire DNA sequence of an organism – which in humans is some three billion nucleotides long. A typical human gene is a few thousand bases long. There are many genes on both strands of the DNA of a chromosome. Humans have 23 pairs of chromosomes and somewhere on the order of 30,000 genes.

Transcription

An initial stage of protein synthesis is the transcription of the DNA into messenger RNA (mRNA). This mRNA transfers the information from the DNA in the nucleus of the cell out into the cytoplasm of the cell where the protein is synthesized. RNA is a molecule very similar in structure to DNA, except that thymine is replaced by the nucleotide uracil (U), and RNA uses the sugar ribose instead of deoxyribose for its sugar-phosphate backbone. If a subsequence on one strand is a gene, then this strand is known as the sense strand for this portion of the double helix. The complementary DNA strand is used by an enzyme (a catalytic protein) known as RNA polymerase II to

synthesize the mRNA. This complementary strand is known as the template or antisense strand. The non-template, or sense, strand has the sequence in the orientation in which genes are reported.

Splicing

The newly synthesized mRNA is known as pre-mRNA at this stage as it must undergo chemical modifications to its beginning and end. Often chemical modification is followed by "splicing" where precise, predefined, subsequences are spliced out and degraded. These subsequences are called introns (INTeRvening sequences); the subsequences which are joined together to make the mature mRNA, are called exons (EXpressed sequences). The joined exons, called "mature mRNA" or simply "mRNA," pass out of the nucleus of the cell to the cytoplasm where protein synthesis occurs.

The start and end of the intron are known as the donor or 5' and acceptor or 3' splice sites, respectively. Although the initial GU and terminal AG of an intron are the only highly conserved sequences in introns, figure 5.1 shows that there are longer, less well conserved sequences. In particular, at the

donor and acceptor splice sites as well as at a sequence known as the branch point sequence which is generally 30 bases upstream from the acceptor splice site. Although this is useful information, the sequences given at the donor splice site and branch-point occur only 10 and 40 percent of the time respectively (and the branch-point sequence has only a single unique base represented), making these moderately conserved signals of limited value in splice site detection.

5' $\ldots E_n$ | donor splice site $\ldots$ branch pt. seq. $\ldots$ acceptor splice site | E_{n+1} $\ldots$ 3'

$A_{64}G_{73}$ | $G_{100}U_{100}A_{62}A_{68}G_{84}U_{63}$ $\ldots Y_{80}NY_{80}Y_{87}R_{75}A_{100}Y_{95}$ $\ldots 12YNC_{65}A_{100}G_{100}$ | N

Figure 3.1: Consensus sequences for regions of an intron. E_k is the k^{th} exon of the gene. | denotes an exon/intron or intron/exon boundary. R - a puRine (an A or G base), Y - a pYrimidine (a C or T/U), N - aNy nucleotide. The subscripts give the percentage occurrences of these bases. Subscripts of 100 are rounded, and there are many known exceptions (and many more may be found when introns are searched for without assuming that they start and end with these sequences. See http://www.ebi.ac.uk/asd/altextron/-pre-release-dist-data.html for current percentages of donor/acceptor splice sites).

Alternative Splicing

To add to the problem of splice site detection, it is estimated that one half of the human genes that are spliced can undergo alternative splicing. Alternative splicing yields different (viable) proteins through a variety of means: alternate donor splice site, alternate acceptor splice site, exon skipping, and splice vs. no splice. When the intron is spliced at a different starting point, this is known as alternative donor splice site splicing. If these alternate starts to the splice site are off by a multiple of three nucleotides, then amino acids corresponding to the differing bases added or lost will be added or deleted from the final protein. If, on the other hand, the alternate starts are not off by a multiple of three, then there is a frame "shift" in the codons (see section on translation), and thus all subsequent corresponding amino acids can be different. A similar situation holds for alternate acceptor splice sites.

As the name implies, exon skipping occurs when an exon is skipped in the

splicing process. Thus one less exon is incorporated into the mature mRNA. If the exon has a length that is a multiple of three, then a certain region of the resulting protein is excised; whereas if its length is not a multiple of three, then a frame shift occurs with the above mentioned consequences.

Splicing vs. no splicing is similar to exon skipping, but instead of excising an exon, an intron is incorporated instead of being spliced out. Once again the length of the intron determines a possible frame shift.

Translation

The information in the mRNA is used to synthesize protein in a process known as translation. The genetic code of this mRNA is read in consecutive, non-overlapping sets of three nucleotides. Each of these triplets codes for a particular amino acid – the subunits of a protein. Thus a sequence of DNA has three frames, only one of which is used to make a particular protein. Consider the sequence ... TACGGTAATCCGGGT .... Since the sequence is read in triplets, it could be read as

... TAC GGT AAT CCG GGT...,

... T ACG GTA ATC CGG GT... or

... TA CGG TAA TCC GGG T...,

each of which would code for an entirely different amino acid sequence. The triplets in the proper frame, which are used for protein synthesis, are called codons.

There are 64 codons (four possible nucleotides in each of the three locations). Three of the codons (TAA, TAG, and TGA, or their more commonly

used mRNA counterparts: UAA, UAG, and UGA) signal that protein synthesis should stop, and are thus known as stop codons. The other 61 each code for one of the 20 amino acids. Some amino acids are coded for by only a single codon, while others have as many as six. The correspondence between a codon and its associated amino acid, or function as a stop codon, is so consistent over all organisms (although exceptions exist), that it is known as the "universal code."

Protein synthesis occurs on a cellular organelle known as a ribosome. When one of the many ribosomes in the cytoplasm of the cell comes in contact with the 5' end of the mRNA, the ribosome becomes attached to it. The ribosome "reads," or moves down the mRNA three bases, or one codon, at a time. There are two sites on the ribosome each of which can hold a codon from the mRNA, the transfer RNA (tRNA), and the tRNA's associated amino acid. Each tRNA has a three base anticodon on it. Only a tRNA with an anticodon that matches the mRNA codon (by complementary base pairing), which is docked on the ribosome, can dock at the ribosome site. This assures that the correct amino acids are placed in close proximity. The amino acid from the most recently attached tRNA is then attached to the growing polypeptide chain. Thus the DNA message is faithfully transferred via the mRNA to the final polypeptide.

3.4 Methods

The differences between codon frequencies in exons and associated triplet frequencies in introns gives exploitable information for distinguishing between the two – see section 2.5. Human DNA in exons has been sequenced and analyzed for some time in the form of RNA and cDNA (complementary DNA which is made in the laboratory from collected mRNA), and thus codon frequency usage tables are available. With the human genome now sequenced, and with access to databases, for example EID (Exon-Intron Database, http://www.mcb.harvard.edu/gilbert/eid/) [Saxonov et al., 2000], that have intron sequences, we can now compute intron triplet frequencies. Assuming the frequencies we arrive at in these training sets are representative of the frequencies in the exons and introns we are looking for, we can look at a subsequence of DNA and determine if it more probably came from an exon than an intron.

There are three frames in an exon, only one of which is the reading frame used by the cell to code for protein synthesis. Consider, for example, the sequence ...AATGCCTA... in an exon. The first A in the sequence could be the first, second or third base in a codon which would result in the sequence having the following codons, or being read as

...AAT GCC TA...,

...AA TGC CTA... or

...A ATG CCT A....

The codon locations are known in the database (our training set), and the

triplets collected as codons are denoted by frame 1. The triplets collected as the second and third nucleotides of a codon, and the first nucleotide of the next codon are denoted by frame 2. Finally, the triplets collected as the final nucleotide in a codon followed by the first two nucleotides of the subsequent codon are denoted by frame 3. All three of these triplet frequencies are collected for every exon in the database. Similarly, the term "position k" is given to an individual nucleotide where k = 1, 2, 3, or i refers to the first, second, or third position in a codon, or an intron nucleotide respectively.

Intron nucleotide triplet frequencies were analyzed in various frames as well, but were all found to be similar – figure 3.4. A two-factor fixed effects ANOVA (6 frames and 64 triplets) gave P values of 0.9067 and 7.4712^{-379} for the null hypotheses of no effect of frame and triplet, respectively. Thus we would reject the latter hypothesis, but not the former. Also, the expected value, see equation 3.7 for the exact formula, of the likelihood ratios $\log \frac{L(H_j|Data)}{L(H_k|Data)},\ j,k = 1,2,3,-1,-2,-3,\ j \neq k$ was low enough that the simplifying assumption of using a single set of averaged intron triplet frequencies was used. Table 3.1 (later in this section) gives these expected values, which range from 7.1175 to 14.6614, for the three exon frames and the averaged intron frame. The expected values when comparing the six intron frames to one another, by comparison, ranged from .01036 to .02907 and thus comparing these six intron frames to one another gives only from .07 to .19 percent as much information as comparing the three exon frames and the averaged intron frame to one another.

An interesting note concerning the intron triplet frequencies is that all

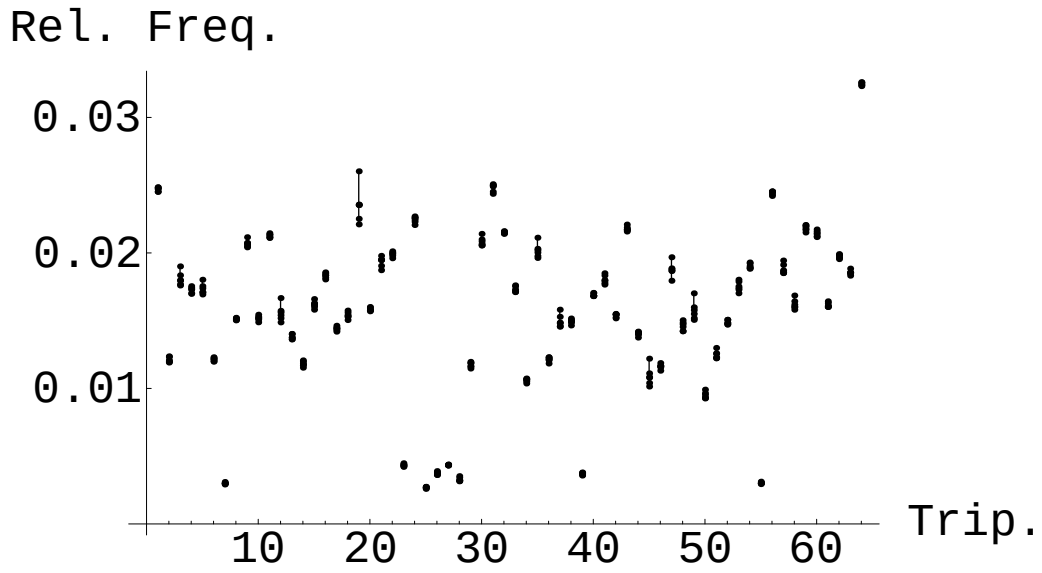

Figure 3.2: Vertical axis – relative frequency of intron triplets, horizontal axis – the 64 triplets in alphabetical order. Frequencies of all six frames are superimposed with vertical line indicating range. Note the similarity of frequencies in all frames.

eight intron triplets containing the dinucleotide CG are relatively scarce. In one thousand randomly generated sequences of the same length as the total number of intron nucleotides in our database for example, the lowest dinucleotide frequency was never as low as the CG frequency observed in our dataset. The CG frequencies are similar to one another, and have an average frequency less than 1/5 the frequency of the other 56 triplets. This initially surprising result may be explained by the fact that the DNA dinucleotide CG often has a methylated C – which may be useful to an organism because it helps prevent viruses from inserting their own DNA at this location along the DNA sequence [Simon et al., 1983]. The C is chemically modified, and this modification results in a C which is more easily spontaneously mutated into a T. Since mutations in an intron are not translated, they cause no phenotypic change in an individual or its descendants. A CG to TG mutated intron is therefore more likely to spread through the population than the same mutation in an exon (where the mutations tend to be detrimental to the individual). This may explain why the CG dinucleotide is relatively rarer among introns than it is among exons (the mutation is "selected against" in exons, but not in introns).

The single window method with the highest accuracy uses exon and intron triplet frequencies (as compared to mononucleotide or dinucleotide frequencies) in the single likelihood ratio

$$\lambda = \log \frac{L(H_e|Data)}{L(H_i|Data)} \tag{3.2}$$

where H_e and H_i are the hypotheses that the data came from an exon (with

the first base of the data being the first base of a codon) or intron region. Let seq_{wl} be the sequence of length wl (window length – divisible by 3) used in the likelihood ratio. Then the likelihood in the numerator of 3.2 is given by

$$L(H_e|Data) = k * Prob(Data|H_e) = k * \frac{(wl/3)!}{a_1!a_2!\ldots a_{64}!} p_1^{a_1} p_2^{a_2} \ldots p_{64}^{a_{64}} \quad (3.3)$$

where k is an arbitrary fixed constant of likelihoods, $wl/3$ is the total number of triplets in seq_{wl}, a_i is the number of occurrences of the i^{th} triplet in seq_{wl}, and p_i is the probability (relative frequency) of the i^{th} triplet within exons. A similar calculation gives $L(H_i|Data)$. The likelihood ratio of interest here is the ratio of equation 3.3 to its intron counterpart. Upon cancellation of like terms and taking the log for computational ease, we have:

$$\lambda = \log \frac{L(H_e|Data)}{L(H_i|Data)} = \log \frac{p_1^{a_1} p_2^{a_2} \ldots p_{64}^{a_{64}}}{q_1^{a_1} q_2^{a_2} \ldots q_{64}^{a_{64}}} \quad (3.4)$$

or more simply:

$$\lambda = a_1 \log \frac{p_1}{q_1} + a_2 \log \frac{p_2}{q_1} + \ldots + a_{64} \log \frac{p_{64}}{q_{64}} \quad (3.5)$$

where q_i is the frequency of the i^{th} triplet within introns. Each of the 64 p_i/q_i are computed only once, and so each λ takes only wl/3 additions and multiplications.

Let H_1, H_2, H_3 and H_i be the hypotheses that the first base of the data is the first, second, or third base of a codon, or a base in an intron, respectively. $L(H_1)$ was given in 3.3 (as $L(H_e)$). $L(H_2)$ and $L(H_3)$ are similarly computed using triplet frequencies where the triplet starts in the second and third

position of a codon respectively. That is H_1, H_2, and H_3 correspond to the triplet frequencies collected in frames 1, 2, and 3 respectively. The ability of the method to distinguish between the different frames in an exon or an intron comes from their differences in triplet frequencies. To obtain a measure of the method's ability to distinguish between any two of the four frames, we calculated the expected value of

$$\log \frac{L(H_j|Data)}{L(H_k|Data)} \tag{3.6}$$

where H_m is the hypothesis that the triplet came from frame m, one of the three frames in an exon, or an intron. These values indicate how much distinction between frames can be expected. Let EV_{jk} be the expected value of 3.6 *given that the data triplet came from frame j.* More precisely,

$$\begin{aligned} EV_{jk} &= \sum_{n=1}^{64} \Big[Prob(triplet_n|triplet_n \text{ came from frame j}) \\ &\quad * \; log \frac{L(H_j|triplet_n)}{L(H_k|triplet_n)} \Big], \end{aligned} \tag{3.7}$$

where $Prob(triplet_n|triplet_n$ came from frame j) is the number of $triplet'_n s$ found in frame j in the database divided by the total number of all *triplets* in frame j in the database.

Table 3.1 gives these values, and we can see that the difference in triplet frequencies of codons and introns (i.e. $EV_{1i} = 8.3313$, $EV_{i1} = 11.3518$) is not as great as the difference between certain other frames. In particular, the likelihood ratio containing H_1 and H_3 will, on average, more readily distinguish between these two frames (i.e. $EV_{13} = 11.3757$, $EV_{31} = 14.6614$). Certain other combinations of hypotheses also give higher likelihood ratios.

EV_{12}	10.5985	EV_{1i}	8.3313	EV_{2i}	7.1175
EV_{21}	12.0368	EV_{i1}	11.3518	EV_{i2}	7.5338
EV_{13}	11.3757	EV_{23}	9.7323	EV_{3i}	7.8890
EV_{31}	14.6614	EV_{32}	9.9501	EV_{i3}	8.1232

Table 3.1: EV_{jk} is the expected value of $\log \frac{L(H_j|Data)}{L(H_k|Data)}$ where Data is a triplet from frame j. $j = 1, 2, 3, i \quad j \neq k$.

As all of these differences in triplet frequencies contribute to the ability of the method to distinguish between frames, our new method uses the following likelihood ratios :

$$\lambda_{xy} = \log \frac{L(H_x|Data)}{L(H_y|Data)} \tag{3.8}$$

where $x, y = 1, 2, 3, i \quad x \neq y$. From these λs, the following Λs are defined to be:

$$\Lambda_x = \sum_{y=1,2,3,i \quad y \neq x} \lambda_{xy}. \tag{3.9}$$

Λ_x gives the sum of the three log likelihood ratios which each have the hypothesis that the data came from frame x in the numerator and one of the three alternative hypotheses in each of the denominators. Thus Λ_x gives the overall likelihood that the data came from frame x.

In order to capture maximal information regarding the first nucleotide position in the raw genomic subsequence of length wl, we consider additional biological constraints on the sequence. The cyclic "position within a codon" nature of the nucleotides in an exon and the nearly identical triplet frequencies for all frames in an intron gives the positions of the second and third

nucleotides of the subsequence (assuming no intervening splice site) for any assumed position of the first nucleotide. Calculations based on the four assumptions that the first base of the window is from a first, second, or third base of an exon, or a base in an intron are as follows:

$$\Lambda_1(wl_k) + \Lambda_2(wl_{k+1}) + \Lambda_3(wl_{k+2}) \tag{3.10}$$

$$\Lambda_2(wl_k) + \Lambda_3(wl_{k+1}) + \Lambda_1(wl_{k+2}) \tag{3.11}$$

$$\Lambda_3(wl_k) + \Lambda_1(wl_{k+1}) + \Lambda_2(wl_{k+2}) \tag{3.12}$$

$$\Lambda_i(wl_k) + \Lambda_i(wl_{k+1}) + \Lambda_i(wl_{k+2}) \tag{3.13}$$

where $\Lambda_j(wl_m)$ is Λ_j evaluated using the window of nucleotides starting with the m^{th} nucleotide. As for a basic likelihood ratio, these sums of Λs do not give us a statistical distribution (i.e. does not give the probabilities of being in each of the four states), but give us a way to compare our four hypotheses. This is exactly what we want – to be able to compare the relative merits of the hypotheses, and be able to choose the one in which we have the most confidence. To this end we make a prediction as to the position of the k^{th} nucleotide (the first base of the first of the three overlapping windows being looked at) as follows. If the first of the above four sums is largest, we predict the k^{th} nucleotide is a first base of a codon. Similarly, if the second, third

or fourth sum is largest, then a prediction is made that the k^{th} base comes from the second or third base of a codon, or from an intron, respectively.

3.5 Results

We tested our new "multi-window" method described in the last section on the human protein-coding gene sequences containing intron sequences from the Exon-Intron Database at http://www.mcb.harvard.edu/gilbert/-eid/) [Saxonov et al., 2000]. This web site has various Perl scripts to help extract relevant information from their 800mb database which was obtained from GenBank 115 (the 115^{th} release of GenBank). We used the EID Perl script extract_species.pl to extract human genes from EID's sub-database gb115.exp_mrna.dEID. This sub-database contains DNA sequences which have subsequences within them which match known cDNAs. These cDNAs, or complementary DNAs, are double-stranded DNA sequences constructed from mature mRNAs. Recall that mature mRNAs have been transcribed from an organism's DNA and then have undergone splicing to remove the sequences corresponding to the introns of the gene. Thus, using a sub-database which has known cDNAs in each sequence helps ensure that each sequence comes from a true gene [Russell, 1998]. We further filtered this sub-database by removing any sequences which did not start with the canonical start codon ATG, end with an exon, or contain any characters other than A, C, G, or T (sometimes reported sequences will have other characters such as N for any nucleotide, B for any non-A nucleotide, etc.). The remaining 3,085 gene

sequences were analyzed.

In choosing the number of nucleotides to observe per window (wl), a compromise must be made between a large wl which has more data (yielding better predictions) and a small wl which is more likely to be shorter than the exons themselves. If wl is longer than the length of the exon, it will lower the method's ability to detect the exon as it will contain both exon and intron data. Figure 3.3 shows the length distribution of exons in genes containing introns, from our database. Genes without introns (single exon genes) have longer average exon lengths [Chen et al., 2002].

In our database, only approximately 6 percent of exons in intron containing genes are shorter than 45 nucleotides, and this length still gave good sensitivities and specificities (see below), and thus wl was set to 45 for this analysis. Note that window length is a parameter that can be varied if there is any *a priori* knowledge of the exon length(s) for which the search is being employed.

As the multi-window method uses, for a particular position, three overlapping windows, of combined length $wl + 2$, for every nucleotide prediction, the last $wl + 1$ bases of a gene sequence are not given a prediction – due to there not being enough available data. A prediction for these final bases could be made such that they are consistent with the last prediction, but in this analysis these bases are simply omitted. They represent less than one half of one percent of the total nucleotides analyzed.

The sensitivity, Prob(method predicts nucleotide to be in frame j | nucleotide is in frame j), and specificity, Prob(method predicts nucleotide to

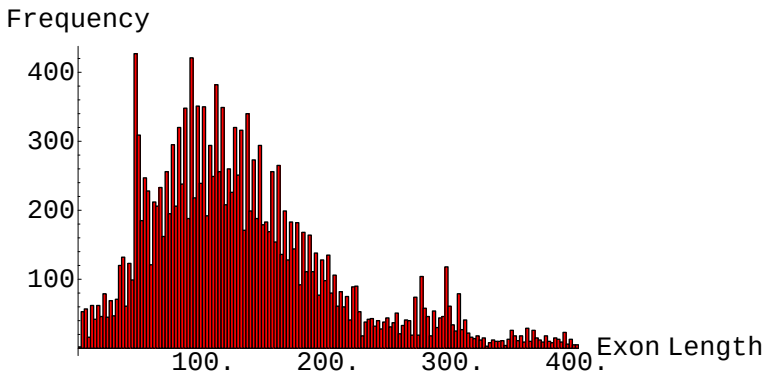

Figure 3.3: Length distribution of exons in our data set within intron containing genes. Exons of length greater than 400 nucleotides, which represent approximately 4 percent of these exons, are not included in the graph.

not be in frame j | nucleotide is not in frame j), of the multi-window method are given in table 3.2. Calculations were based on the 3,085 gene sequences which contain 31,400,012 bases for analysis. The sensitivities and specificities for individual genes were very similar for all three categories of bases in exons as can be seen in table 3.2.

Figures 3.4 through 3.7 show the ordered ranking of the sensitivities and specificities for the 3,085 individual gene sequences. The sensitivities and specificities were very similar for all three exon positions, thus only the graphs for the first base of an exon and for intron bases are shown. In Figure 3.4, for example, we see that approximately the lowest scoring 1,000 genes have a sensitivity of .6 or less and the approximately 1,000 highest scoring genes have a sensitivity of .8 or greater. This mean that for the 1,000 gene sequences with the lowest sensitivity, 60 percent or less of the actual bases which are

NUCLEOTIDE TYPE	SENSITIVITY (% true positive)	SPECIFICITY (% true negative)
First Base of Codon	66.3319%	87.9792%
Second Base of Codon	66.2278 %	87.9521%
Third Base of Codon	66.24%	87.9882%
Intron Base	63.0762%	81.7907%

Table 3.2: Sensitivity and specificity of method detecting various categories of nucleotides.

the first base of a codon are properly detected. Similarly, for the 1,000 gene sequences with the highest sensitivity, 80 percent of the true first bases in a codon are accurately predicted.

We attempted to find a simple characteristic, such as total or average exon length within a gene, that is highly correlated with the method's sensitivity or specificity. The aim here is to see if the method could be specifically trained on lower scoring sequences as described below. Correlation results for the sensitivities and specificities of first base of a codon are shown in tables 3.3 and 3.4. P-values for testing the null hypothesis that the correlation is equal to zero are also given. In table 3.4 we see that in two cases the null hypothesis would be rejected at the .01 significance level, and that in table 3.6 there is sufficient evidence to reject the null hypothesis at this significance level for all cases. Still, no characteristic examined is highly correlated with the method's performance. Total and average intron length

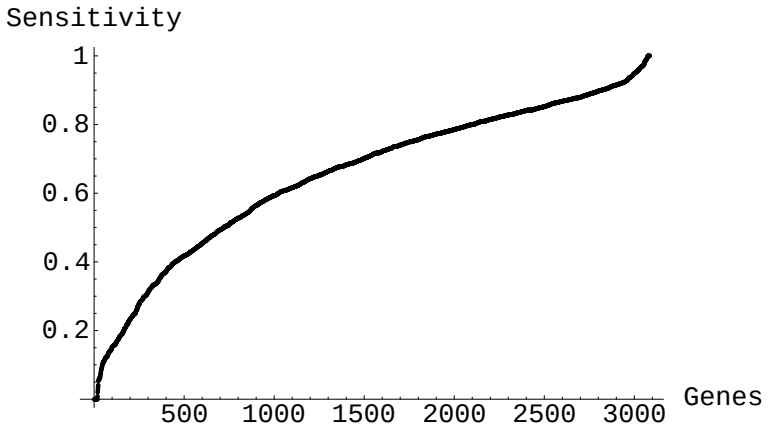

Figure 3.4: Ordered ranking of sensitivities for individual genes for first base of codons.

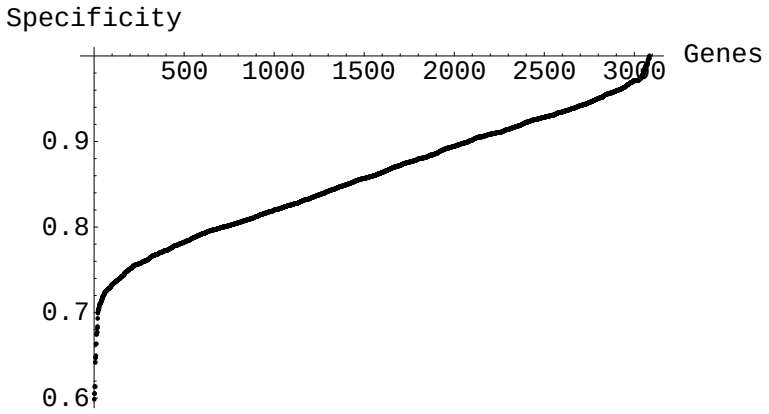

Figure 3.5: Ordered ranking of specificity for individual genes for first base of codons.

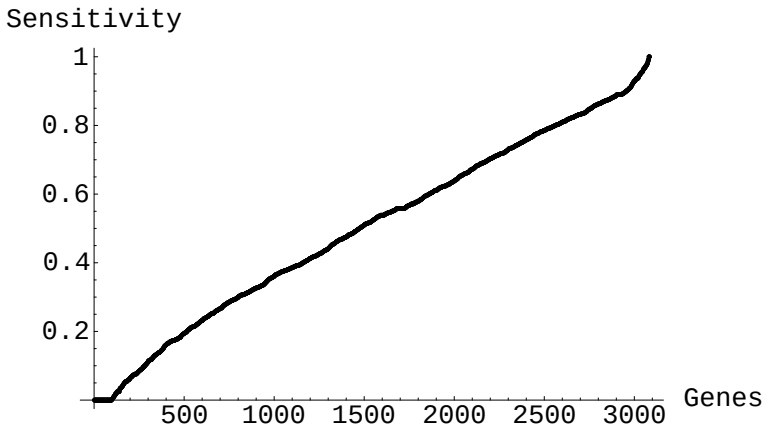

Figure 3.6: Ordered ranking of sensitivities for individual genes for intron bases.

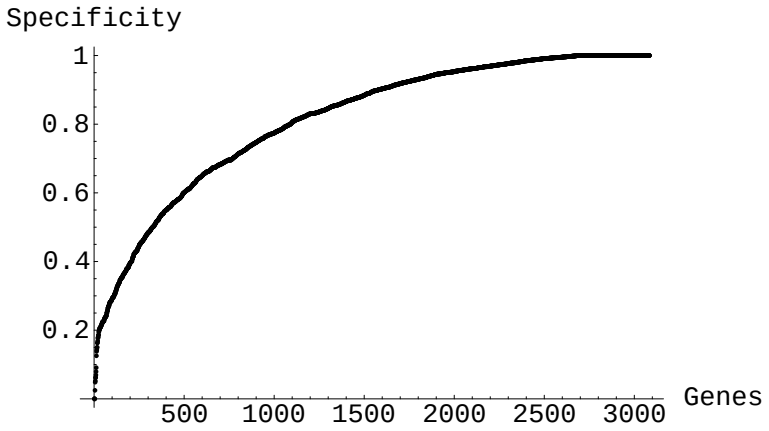

Figure 3.7: Ordered ranking of specificity for individual genes for intron bases.

CHARACTERISTIC	CORRELATION	P-Value
Total Exon Length	0.0368849	0.040421
Total Intron Length	-0.188104	$< 10^{-4}$
Average Exon Length	0.0327903	0.068508
Average Intron Length	-0.100098	$< 10^{-4}$
(Exon Length) / (Intron Length)	0.0245828	0.172138

Table 3.3: Correlation coefficients for characteristic vs. sensitivity of first base of codon detection.

CHARACTERISTIC	CORRELATION	P-Value
Total Exon Length	-0.050685	0.0048339
Total Intron Length	0.167440	$< 10^{-4}$
Average Exon Length	0.076865	$< 10^{-4}$
Average Intron Length	0.137882	$< 10^{-4}$
(Exon Length) / (Intron Length)	0.051086	0.0045079

Table 3.4: Correlation coefficients for characteristic vs. specificity of first base of codon detection.

are the most highly (negatively/positively) correlated characteristics for the sensitivity/specificity of the first base of a codon detection. Study of additional biological characteristics (for example if the gene codes for membrane bound proteins), which may be more highly correlated with the method's performance, should be pursued. This may give different categories of genes which have lower than average sensitivities and specificities. The method could then use the triplet frequencies specific to these categories of genes, which could in turn raise the sensitivity and specificity of the method for these currently low scoring genes.

3.6 Discussion

The multi-window method introduced in this chapter uses the information contained in three overlapping windows of nucleotides to classify the first base in the first of these windows. Thus it will tend to misclassify nucleotides when a large proportion of the latter part of the window is in, say, an exon region and the base for which the prediction is being made is in an intron region. A simple approach of scanning for the canonical start of an intron, the dinucleotide subsequence GT, in the vicinity of $wl/2$ nucleotides down from where the method starts to predict an exon, and likewise looking for the canonical end of an intron, the dinucleotide AG, a distance $wl/2$ nucleotides down from where the method starts to predict an intron, could lead to higher sensitivities and specificities for all nucleotide categories. However there are more sophisticated and accurate splice site

detectors [Pertea et al., 2001], [Burge and Karlin, 1997], and combining the multi-window method with them would likely lead to better results; future work on this is planned.

One possible advantage of the multi-window method over methods which rely more on information regarding canonical gene sequences (splice sites, promoter sequences – which signal the start of a gene, etc.) is that it is purely content based, and thus it is not biased against less common splice sites, for instance. This could prove useful in detection of genes containing these rarer types of splice sites.

Another possible benefit of the multi-window method is detecting sequences other than a typical gene. Pseudogenes are sequences which no longer code for proteins due to some mutation in their sequence. If pseudogenes have mutated such that many of the common "signals," which many other methods try to identify, have been lost, this method may be able to detect them at a higher rate (which could be a benefit or detriment depending on whether the user is interested in these other sequences). Similarly, subsequences of a chromosome are sometimes randomly translocated to a different location along the genome. If this breaks a gene into two parts it would disrupt the gene's protein synthesis, and thus the sequence would lose its classification as a gene. If genes are broken by these translocation events, this method should still be able to find the broken subsequences.

If *a priori* knowledge of the size of the exon(s) being searched for is known, the user can adjust wl to give the multi-window method higher sensitivity and specificity without compromising its ability to exclude the desired exon(s).

This could be the case, for example, if the user had *a priori* knowledge of the protein and/or the protein's active sites, which are often contained in a single exon [Gilbert, 1978], for which the gene coded.

Chapter 4 introduces hidden Markov models both as a means to more rigorously analyze the sensitivity and specificity of our prediction methods and to compare our multi-window method with two prediction methods offered by the hidden Markov model. Both our multi-window method, and the hidden Markov model use the triplet frequencies in our exon and intron training set as the most significant source of information.

Bibliography

[Burge and Karlin, 1997] Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *J Mol Biol*, 268(1):78–94.

[Chen et al., 2002] Chen, C., Gentles, A. J., Jurka, J., and Karlin, S. (2002). Genes, pseudogenes, and Alu sequence organization across human chromosomes 21 and 22. *Proc Natl Acad Sci U S A*, 99(5):2930–2935.

[Edwards, 1972] Edwards, A. W. F. (1972). *Likelihood.* Cambridge University Press, New York, NY.

[Fairbanks and Anderson, 1999] Fairbanks, D. J. and Anderson, R. W. (1999). *Genetics.* Brooks/Cole.

[Gilbert, 1978] Gilbert, W. (1978). Why genes in pieces. *Nature*, 271(5645):501.

[Lewin, 1994] Lewin, B. (1994). *Genes V.* Oxford University Press.

[Pertea et al., 2001] Pertea, M., Lin, X., and Salzberg, S. L. (2001). GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Res*, 29(5):1185–1190.

[Russell, 1998] Russell, P. J. (1998). *Genetics*. Benjamin/Cummings, an imprint of Addison-Wesley Longman, Inc., Fifth edition.

[Saxonov et al., 2000] Saxonov, S., Daizadeh, I., Fedorov, A., and Gilbert, W. (2000). EID: the Exon-Intron Database-an exhaustive database of protein-coding intron-containing genes. *Nucleic Acids Res*, 28(1):185–190.

[Simon et al., 1983] Simon, D., Stuhlmann, H., Jahner, D., Wagner, H., Werner, E., and Jaenisch, R. (1983). Retrovirus genomes methylated by mammalian but not bacterial methylase are non-infectious. *Nature*, 304(5923):275–277.

[Snustad et al., 1997] Snustad, P. D., Simmons, M. J., and Jenkins, J. B. (1997). *Principles of Genetics*. Wiley.

[Watson and Crick, 1953] Watson, J. D. and Crick, F. H. C. (1953). A structure for Deoxyribose Nucleic Acid. *Nature*, 171:737.

[Zhang and Zhang, 2002] Zhang, C.-T. and Zhang, R. (2002). Evaluation of gene-finding algorithms by a content-balancing accuracy index. *J Biomol Struct Dyn*, 19(6):1045–1052. Evaluation Studies.

Chapter 4

Hidden Markov Models as a Means of Analyzing Likelihood Ratios

4.1 Abstract

A hidden Markov model (HMM) which incorporated many of the same features as our multi-window method was used to give a more rigorous analysis of the sensitivity and specificity of the method. While there are some differences, many of the qualitative features are captured by both models.

4.2 Introduction

As a means to more rigorously analyze the sensitivity and specificity of the multi-window method, a hidden Markov model was constructed which captured as many similar features as possible. The most significant source of information in both comes from the collected triplet frequency data in each of the three frames of the exons and the triplet frequencies found in introns (which are approximately invariant under shifts in frame collection.) Another salient feature captured by both methods is the cyclic tendency to move through the codon "positions" when in exons, and to stay in exons (or introns) once there. While the hidden Markov model does not incorporate any parameter corresponding to the window length used in the multi-window method, it does use an algorithm that finds the most probable "state path" through the entire sequence, which allows it a "view" of the overall sequence even though the basic units of interest are only the small triplets.

4.2.1 Overview of Markov Chain Models and Hidden Markov Models

A Markov model is a set of states and a corresponding set of values which give the probabilities of changing from one state to another [Rabiner, 1989]. One can consider the weather at noon on consecutive days as being modeled by a Markov model where there are, for example, three distinct states: sunny, overcast, and precipitating. Call these states 1, 2, and 3 respectively. Then $a_{ij}, 1 \leq i \leq 3, 1 \leq j \leq 3$ is the probability of the weather being in state j

given that it was in state i the previous day. For an N state model we have $a_{ij} \geq 0$ and $\sum_{j=1}^{N} a_{ij} = 1$. These are examples of discrete, first order Markov chains. If the transition values are dependent not only on the current state, but also on prior states, then a higher order Markov model may be used [Rabiner, 1989].

In the above example, the states of the system are observable – one can tell if it is sunny or overcast. For hidden Markov models, some signal is observable, but the underlying state which generated the signal is hidden. Consider for example three urns, hidden behind a curtain, in a room. Each urn has a specific (hopefully sufficiently different) proportion of red, green, and blue balls in it. A transition matrix $A = a_{ij}, 1 \leq i \leq 3, 1 \leq j \leq 3$ is given which gives the probability of drawing a ball from urn j given that the previous ball was drawn from urn i.

A person behind the curtain selects the urn using the transition matrix A, and then randomly draws a ball from that urn. The person then shows the color of the ball drawn (the observation), but does not divulge the urn from which it was drawn (the state). The goal is to surmise from the observation sequence, the state sequence.

Consider a model with M observable signals, N states and T observations. Let q_t and O_t represent the state and observation, respectively, at step t. Then we wish to determine $Q = q_1 q_2 \ldots q_T$ from $O = O_1 O_2 \ldots O_T$.

Let π be the N dimensional initial state vector with $\pi_i = P(q_1 = S_i)$ giving the probability that the process starts in state i. Let $b_j(k) = P(O_t = v_k | q_t = S_j)$ denote the probability that the t^{th} observation is v_k, given that

the process is in state j. Thus $B = b_j(k), 1 \leq j \leq M, 1 \leq k \leq N$ simply gives the probability of each observation from each state.

This triplet, which we denote by $\lambda = (A, B, \pi)$, then characterizes the HMM. Given the model λ, we need to be able to calculate $P(O|\lambda)$ and find some method to choose the state sequence Q which in some way "best" explains the observation sequence O. Speech recognition has used HMMs since the early 1970s, and this discipline continues to produce the majority of papers on the subject [Durbin et al., 1998]. In this context, a recorded speech signal is broken into 10-20 millisecond "frames". These frames are assigned to predefined categories, and are the observations of the HMM. The goal is to infer the state sequence, or sequence of words, from these observations. For a thorough discussion of these issues, please see the detailed tutorial by Rabiner on HMMs [Rabiner, 1989].

It is a straightforward task to calculate the probability of a particular observation sequence O given λ and a particular state sequence $Q = q_1 q_2 \ldots q_T$:

$$\begin{aligned} P(O|Q,\lambda) &= \Pi_{t=1}^{T} P(O_t|q_t,\lambda) \\ &= b_{q_1}(O_1) b_{q_2}(O_2) \ldots b_{q_T}(O_T), \end{aligned}$$

with the assumption of statistical independence of observations. We can also find the probability of this state sequence Q:

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \ldots a_{q_{T-1} q_T}.$$

The product of these two gives us the joint probability of O and Q:

$$P(O,Q|\lambda) = P(O|Q,\lambda) P(Q|\lambda). \tag{4.1}$$

By summing the above joint probability over all possible state sequences Q, we may then find the probability of the observation sequence O given the model λ:

$$\begin{aligned} P(O|\lambda) &= \sum_{allQ} P(O|Q,\lambda)P(Q|\lambda) \\ &= \sum_{q_1 q_2 \ldots q_T} \pi_{q_1} b_{q_1}(O_1) a_{q_1 q_2} b_{q_2}(O_2) \ldots a_{q_{T-1} q_T} b_{q_T}(O_T). \end{aligned}$$

Unfortunately, this is computationally impractical unless specific observations are limited to only a very small number of states (i.e. most $b_j(k) = 0$ for most j). In general, if each of the N states can yield each of the M observations, then each of the N^T possible T length state sequences Q could yield the T length observation sequence O, and so for all but the smallest values of T, a new method is needed to calculate $P(O|\lambda)$. Rabiner [Rabiner, 1989] shows that by finding the probability of a partially observed sequence (starting with the first observation, and using then induction), that the number of calculations to compute $P(O|\lambda)$ can be cut to the order of N^2T.

The well known Viterbi algorithm [Viterbi, 1967], [Forney, 1973] is used to calculate a "most probable" state sequence Q. It is important to note that this algorithm finds the state sequence which maximizes $P(Q|O,\lambda)$; it maximizes the probability of the entire state sequence. This is very different than trying to choose the most likely state individually at each time t. To appreciate this difference, consider the three urns again. All three urns contain each of the colors blue, green, and red, but urns one (S_1), two (S_2) and three (S_3) have a preponderance of blue, green, and red balls respectively. Assume also that

$a_{13} = 0$, that is one can not move from urn one to urn three directly. If we were to observe the sequence blue, blue, red, to maximize $P(q_i|O_i, \lambda)$, for each individually, we would choose a state sequence $q_1 = S_1, q_2 = S_1, q_3 = S_3$, even though this state sequence is impossible. The Viterbi method, by contrast, would assign this sequence a probability of zero. At times it may be desirable to calculate the most probable state at a given step t, and this can be done with two algorithms known as the forward and backward algorithms, which are described in the methods section of chapter 4.

4.3 Methods

Let the first, second and third bases of a codon be represented by states 1, 2, and 3 respectively, and denote bases from introns by state 4. An attempt will be made to determine these hidden states from the observed sequence of nucleotides (A, C, G, T). In order to construct the hidden Markov model the $\lambda = (A, B, \pi)$ parameters were found from empirical data. The Exon-Intron Database (EID, http://www.mcb.harvard.edu/-gilbert/eid/) [Saxonov et al., 2000] based on Genbank release 132 was the original database upon which we made our refinements.

The human sequences were extracted from this database using EID's perl script extract_species.pl. Next the two perl scripts filter_exp_keyw1.pl and filter_exp_keyw2.pl were used on this human database. These three scripts are all available from the above EID website. The two "filter" scripts remove sequences that have certain keywords in their annotation which would lead

one to believe that they were found using gene locating programs as opposed to being found by experimentation. If the characters "evidence=experiment" are found in the coding sequence portion of the GenBank entry, then the sequence is not removed – regardless of the possible presence of any other keyword(s). This procedure gives a database with more accurate data. We then took the intersection of the sequences from the two keyword filtered scripts, and further removed any sequences that contained any non A, C, G, or T base predictions (standard notations for ambiguous nucleotides include: N - aNy nucleotide, Y - pYrimidine, U - pUrine, B - anything but A, etc.). This left 7328 sequences. It was further checked that the first base from each of these sequences was the first base of a codon (in "position 1") by checking the length of the first exon and the phase of the first intron (the phase of the intron tells whether it falls between codons, between the first and second, or second and third base of a codon). From these sequences, the triplet data was collected in each of the frames of the exons and introns. The triplet frequencies in the three frames in the exons are significantly different from one another, while those in the introns are almost identical, and thus an average of all the intron frames was used. For further information please see Chapter 2. Further refinement of this set of 7328 sequences was made to remove sequences which had incomplete intron data. This left 4074 sequences for analysis by the hidden Markov model.

Assuming a geometric length distribution for both exons and introns (which fairly closely models the data), and using the average length of the exons and introns in the 4074 sequences which had full exon/intron data,

the values of A and π were found. The average exon and intron length were 156.4 and 1344.7 bases respectively. The mean value of a geometric distribution is $1/p$ where p is the probability of success. Let p be the probability of transitioning from an exon to an intron. Then $1/p = 156.4$ and $p = .0064 = a_{14} = a_{24} = a_{34}$. For transitioning from an intron to an exon, $1/p = 1344.7$ or $p = .00074$. This value of p must be equally divided among a_{41}, a_{42} and a_{43} as a transition from the intron state to any of the exon states is equally likely. Exon states are only permitted to transition to the "next" exon state, or to the intron state, and thus $a_{11} = a_{13} = a_{21} = a_{22} = a_{32} = a_{33} = 0$. The transitions from certain exon states to intron states, and visa versa could also be modified to represent the bias in the phase of introns – see for example [Ruvinsky et al., 2005]. The remaining elements of A are found from the fact that the row sums are equal to one. This gives us the following transition matrix:

$$A = \begin{pmatrix} 0 & .9936 & 0 & .0064 \\ 0 & 0 & .9936 & .0064 \\ .9936 & 0 & 0 & .0064 \\ .00025 & .00025 & .00025 & .99926 \end{pmatrix}$$

The values for π, the initial state distribution, were found from these mean lengths as well. For raw genomic data, the probability of starting in an intron is greater than that of starting in an exon as introns are, on average, longer than exons. Thus we set $\pi_4 = 1344.7/(156.4 + 1344.7) = .8958$, and

$\pi_1 = \pi_2 = \pi_3 = (1 - .8958)/3$ or $\pi = (.0347, .0347, .0347, .8958)$.

The triplet frequency data collected in each of the three exon frames, and the averaged intron frames were used to compute $b_j(k) = P(O_t = v_k | q_t = S_j)$. While $b_j(k)$ would be a 4 by 4 matrix if we were to consider our observation, v_k, to be a single nucleotide, we wish to base $b_j(k)$ on not only the current nucleotide, but also on the two previous nucleotides. Thus $b_j(k)$ is a 64 by 4 matrix – see table 4.1. In this table the observations are $N_1N_2N_3$, any of the 64 triplets composed of A, C, G, and T. The states represent the (hidden) positions of the nucleotides or whether the nucleotides are in an exons (states 1, 2, and 3) or an intron (state 4).

As the first and second bases of a sequence do not have two prior bases, $b_j(k)$ was calculated by taking an average over all 16 possible dinucleotides that could precede the first base and over the 4 nucleotides which could precede the first two bases. More specifically, for the first nucleotide (considering it to be N_3) we calculate $b_j(k)$ as follows: $b_j(k) = \frac{1}{16} \sum_{all N_1 N_2} Prob(O_1 = N_1N_2N_3 | N_3$ is in state k) and for the second nucleotide (now considering the first nucleotide to be N_2 and the second nucleotide to be N_3) $b_j(k) = \frac{1}{4} \sum_{all N_1} Prob(O_1 = N_1N_2N_3 | N_3$ is in state k).

4.3.1 Viterbi Algorithm

The Viterbi algorithm was used to find the most probable state sequence, the sequence giving the states or positions of the nucleotides, through each of the 4074 given (observation) sequences. The state sequence consists of runs of

	State 1	State 2	State 3	State 4
AAA	.269388	.382126	.219406	.371615
AAC	.185139	.196308	.242734	.152143
AAG	.408017	.203866	.378459	.23024
AAT	.137457	.2177	.159401	.246002
ACA	.332993	.338198	.244462	.354702
ACC	.287529	.276706	.414487	.263816
ACG	.179303	.11281	.136263	.0644918
ACT	.200175	.272285	.204788	.31699
AGA	.283724	.377393	.196142	.275953
AGC	.250366	.220286	.383376	.210032
AGG	.324713	.235119	.224111	.289224
AGT	.141197	.167202	.196371	.224791
ATA	.165132	.190793	.0906704	.233022
ATC	.172207	.294239	.358234	.182435
ATG	.492161	.203599	.345294	.25747
ATT	.1705	.31137	.205802	.327072
CAA	.183022	.294728	.137	.208283
CAC	.209459	.237403	.217053	.220908
CAG	.471892	.21821	.52145	.336807
CAT	.135628	.249659	.124497	.234003
CCA	.335032	.27937	.249094	.302447
CCC	.28596	.214465	.350725	.289691
CCG	.158241	.168507	.136271	.0702144
CCT	.220767	.337658	.26391	.337647
CGA	.127893	.292008	.156662	.186917
CGC	.296032	.227164	.377694	.257561
CGG	.450858	.25394	.343966	.315997
CGT	.125217	.226888	.121678	.239525
CTA	.116417	.193118	.0691483	.144759
CTC	.24203	.264141	.261291	.262964
CTG	.488768	.255122	.544312	.323131
CTT	.152785	.287619	.125249	.269146

	State 1	State 2	State 3	State 4
GAA	.256553	.361951	.19616	.283744
GAC	.18903	.208475	.258713	.168451
GAG	.414651	.224655	.382294	.334773
GAT	.139765	.20492	.162833	.213032
GCA	.301328	.268598	.193233	.293181
GCC	.279853	.25132	.423574	.312555
GCG	.174793	.154284	.14178	.076183
GCT	.244026	.325799	.241414	.318081
GGA	.281533	.396803	.222869	.258681
GGC	.264274	.245537	.354624	.23717
GGG	.326264	.188887	.271117	.297985
GGT	.127928	.168774	.15139	.206164
GTA	.142402	.194075	.0895005	.194507
GTC	.210535	.273234	.246716	.196216
GTG	.503063	.260804	.519045	.338988
GTT	.144	.271887	.144739	.270289
TAA	.268658	.352386	.0167946	.301677
TAC	.226126	.215705	.60653	.17661
TAG	.34623	.161368	.0152494	.230564
TAT	.158986	.270542	.361426	.291148
TCA	.336363	.297786	.212337	.292369
TCC	.302024	.298326	.412501	.291204
TCG	.126005	.119248	.0944256	.0486453
TCT	.235608	.284641	.280737	.367782
TGA	.230402	.32646	.0315809	.243587
TGC	.248571	.208435	.350208	.204553
TGG	.390851	.262619	.388172	.281522
TGT	.130177	.202485	.230039	.270338
TTA	.163547	.181027	.0969691	.190044
TTC	.244999	.312557	.436992	.204237
TTG	.421817	.163998	.201273	.208356
TTT	.169636	.342418	.264766	.397363

Table 4.1: Emission probabilities $= b_j(k)$. Each value under state j in the row $N_1N_2N_3$ gives $P(N_3|$ prior 2 bases are $N_1N_2,$ (state of $N_3) = j)$. Of particular note are the three lowest values which correspond to the stop codons in the reading frame (TAA, TAG, TGA – state 3.) Note that the probabilities which must sum to one are $\sum_{N_3} P(N_3|$ prior 2 bases are $N_1N_2,$ (state of $N_3) = j)$, and not the probabilities along a given row.

... 1, 2, 3, 1, 2, 3 ... denoting the positions of the nucleotides within a codon, and runs of ... 4, 4, 4, 4, 4 ... denoting nucleotides in introns. The observation sequence consists of the nucleotides themselves, e.g. ... AAGTACCA .... Let $path$ and $path_k$ denote a state sequence through a given observation sequence and the k^{th} state in $path$, respectively. Also let

$$path^* = argmax_{path} P(O, path),$$

where $P(O, path)$, the joint probability of O and the $path$, both of length L, is given by

$$P(O, path) = \pi_{q_1} \Pi_{i=1}^{L} b_{q_i}(i) a_{q_i q_{i+1}}.$$

Thus $path^*$ is simply the state sequence with highest probability through the observed sequence. Suppose that the probability of the most probable path ending in state k with observation O_i is known for all states k, and denote this probability by $vit_k(i)$. The probabilities for the subsequent observation O_{i+1} are then calculated as follows:

$$vit_l(i+1) = b_l(O_{i+1}) max_k(vit_k(i) a_{kl}). \tag{4.2}$$

Consider the term $vit_k(i)$ on the right hand side, which is calculated for all states. This gives us the most probable path through $O_1, O_2 \ldots O_i$ that ends in state k. Moving to the next state involves the transition from state k to state l, a_{kl}. The only k dependence is in these two factors, and thus the max_k is applied to them. The max is over the product of these two because the quantity of interest is the most likely path through the the observation sequence which ends in state k *and then* transitions into state l.

The max_k factor is multiplied by the probability of observation O_{i+1} from state l, $b_l(O_{i+1})$, to give the most probable path through our given observation sequence $O_1, O_2 \ldots O_{i+1}$ which ends in state l, or $vit_l(i+1)$.

Thus if we know how *vit* starts, then the entire path can be found recursively. For notational convenience, without loss of generality, let there be a begin state of 0 in which all sequences start, and an end state of 0 in which all sequences end. By keeping track of "pointers," which point from all $vit_k(i)$s backwards to the previous states, the path can be found by backtracking. The Viterbi algorithm (see [Durbin et al., 1998]) is as follows:

Viterbi Algorithm

$$
\begin{array}{rlll}
Initialization(i=0): & vit_0(0) & = 1 \\
 & vit_k(0) & = 0 \text{ for k} > 0. \\
Recursion: (i=1,2,\ldots,L): & vit_l(i) & = b_l(O_i) max_k(vit_k(i-1)a_{kl}) \\
 & ptr_i(l) & = argmax_k(vit_k(i-1)a_{kl}). \\
Termination: & P(O, path^*) & = max_k(vit_k(L)a_{k0}) \\
 & path^*_L & = argmax_k(vit_k(L)a_{k0}). \\
Traceback: (i=L, L-1, \ldots, 1): & path^*_{i-1} & = ptr_i(path^*_i).
\end{array}
$$

The initiation step says that the path starts in state 0. The first equation in the recursion step is the previously explained equation 4.2, and the second equations shows how the pointers are calculated. These pointers are collected for every state at each time step. At $i = 1$ all pointers point back to the 0 state. Consider $argmax_k(vit_k(i-1)a_{kl})$ for $i \geq 2$. Then, given that the

process is in state l, we wish to find the most probable previous state. The k^{Max} corresponding to the largest $vit_k(i-1)$ is not what we desire as there may be a very small (or 0) probability of transitioning from this state k^{Max} to state l, and so the given equation yields the desired state. The termination step gives $P(O, path^*)$, and $path_L^*$, the first calculated state of $path^*$. In the traceback step we follow the pointers (backwards) from $path_L^*$, back through their most probable immediate predecessors to $path_0^*$.

The most probable state sequence $path^*$ produced by the Viterbi algorithm is a sequence of states. It consists of runs of ... 1, 2, 3, 1, 2, 3 ... interspersed with ... 4, 4, 4, 4, 4, 4 .... The predicted states corresponding to being in an exon (1, 2, 3) are forced by the transition matrix A to cycle through the three positions in a codon while in the exon. Once the state path has exited the exon, and entered an intron though, it is free to re-enter the exon in any state. This is not biologically reasonable, but it models the procedure followed by the multi-window method. In both of these models there is a tendency to start the exon in the correct state though, as there is local (multi-window) or local and global (Viterbi) information which helps the respective method correctly identify the particular state – it is just not an implied assumption of either model. In the multi-window method for example, if any of the three stop codons are present in frame k then the likelihood ratios that both have H_1 in the numerator and the the window broken into frame k, will be greatly reduced. This will in turn make the method less likely to mispredict the state of a nucleotide as the method makes a transition from an intron to an exon. Although biologically the position of a the

first base in an exon must be the "next" position after the last nucleotide of the previous exon, our method's not making this be a constraint could be useful in detecting an error in the collected sequence data. This is further examined in the Discussion section.

4.3.2 Forward and Backward Algorithms

Using a combination of the results from the so-called forward and backward algorithms, we are able to obtain both the probability of an observed sequence, $P(O)$, and the particular state sequence $\widehat{path}$ which maximizes the probability of the state at each individual observation. The aforementioned combination of results also yields an exact probability (or confidence) for each predicted state given our model $\lambda = (A, B, \pi)$ [Durbin et al., 1998].

The probability in the forward algorithm corresponding to $vit_k(i)$ is

$$f_k(i) = P(O_1, O_2 \ldots O_i, path_i = k)$$

which gives the joint probability of the observed sequence up through and including O_i, and the i^{th} state being state k. The recursion equation is :

$$f_l(i+1) = b_l(O_{i+1}) \sum_k (f_k(i) a_{kl}). \tag{4.3}$$

This is very similar to the recursion equation for the Viterbi method, equation 4.2. Here the forward variable replaces the Viterbi variable, and $\sum_k$ replaces max_k. Whereas in the Viterbi equation we used max_k to give the most probable path, here we want the total probability of the observed sequence,

and thus the sum of the probabilities (from all previous states) is used. The full forward algorithm is as follows:

Forward Algorithm

$$\begin{aligned} Initialization(i=0): \quad & f_0(0) && = 1 \\ & f_k(0) && = 0 \text{ for k} > 0. \\ Recursion: (i = 1, 2 \ldots L): \quad & f_l(i) && = b_l(O_i) \sum_k (f_k(i-1) a_{kl}). \\ Termination: \quad & P(O) && = \sum_k (f_k(L) a_{k0}). \end{aligned}$$

As in the Viterbi algorithm, the initiation step says that we start in state 0. The recursion step is an application of the above equation 4.3, and the termination step gives the total probability of the observation sequence O.

In order to find $P(path_i = k|O)$ we calculate the probability of the entire observed sequence O with the i^{th} state being state k.

$$\begin{aligned} P(O, path_i = k) &= P(O_1, O_2 \ldots O_i, path_i = k) \\ &\quad * P(O_{i+1}, O_{i+2} \ldots O_L | O_1, O_2 \ldots O_i, path_i = k) \\ &= P(O_1, O_2 \ldots O_i, path_i = k) \\ &\quad * P(O_{i+1}, O_{i+2} \ldots O_L | path_i = k). \end{aligned} \tag{4.4}$$

The first term is $f_k(i)$, and we will denote the second term by $\beta_k(i)$, which we can see is the probability of the later observations given that the preceding state was k. We calculate these $\beta_k(i)$s in a manner similar to that of the forward algorithm, and will use them, along with the $f_k(i)$s to obtain

the probability of a state at a given time step given the observation sequence O – equation 4.6. The backward algorithm follows:

Backward Algorithm

$$\begin{aligned} Initialization (i = L): \quad & \beta_k(L) = a_{k0} \text{ for all k.} \\ Recursion: (i = L-1, L-2 \ldots 1): \quad & \beta_k(i) = \sum_l a_{kl} b_l(O_{i+1}) \beta_l(i+1). \\ Termination: \quad & P(O) = \sum_l a_{0l} b_l(O_1) \beta_l(1). \end{aligned}$$

Here the initialization step starts at the end of the sequence, and $\beta_k(L)$ gives the probability of transitioning from state k to the end of the sequence. In the recursion step we have a_{kl} which gives the probability of transitioning from state k to state l. Next, once in state l, we need the probability of observing O_{i+1} which is $b_l(O_{i+1})$. The $\beta_l(i+1)$ gives the probability of the observation sequence $O_{i+2}, O_{i+3} \ldots O_L$ given that the $(i+1)^{st}$ state was l. Finally, as the observation O_{i+1} could, in general, come from any of the states, we must sum over all the states to obtain $\beta_k(i)$. As in the forward algorithm, the termination step gives $P(O)$, and so this step is unnecessary, but serves as a check on the two algorithms.

Using the results from both the forward and backward algorithms, we can now calculate equation 4.4 as

$$P(O, path_i = k) = f_k(i) b_k(i) \tag{4.5}$$

and from this we get the desired posterior probability

$$P(path_i = k | O) = \frac{f_k(i) b_k(i)}{P(O)} \tag{4.6}$$

where $P(O)$ can be obtained from either the forward or backward algorithm. From here it is a simple matter to find the state sequence, $\widehat{path}$, which maximizes the probability of each state at each observation. Let the i^{th} element of $\widehat{path}$ be denoted $\widehat{path}_i$, giving

$$\widehat{path}_i = argmax_k P(path_i = k|O). \tag{4.7}$$

In the Results section we compare the sensitivity and specificity of $path^*$, $\widehat{path}$, and the state path as predicted by the multi-window method.

4.4 Results

Table 4.2 gives the mean and median sensitivities, [Prob(method predicts nucleotide to be in frame j | nucleotide is in frame j)], and specificities, [Prob(method predicts nucleotide to not be in frame j | nucleotide is not in frame j)], for the state prediction of each of the three methods. Both the sensitivities and specificities were very similar when comparing the predictions of each of the three base positions in the exons, and thus they are grouped together in the table with the highest and lowest values shown. The rankings from the most to least accurate of the three methods were unchanged within each of the four sensitivity/specificity and exon/intron categories, regardless of whether the mean or median was used. All three methods are the highest scorer in at least one category, with the Viterbi Algorithm being highest in two categories. The Viterbi algorithm, along with the multi-window method, scored the lowest in two categories. The most striking difference between the

Exon Sensitivity	Intron Sensitivity	Exon Specificity	Intron Specificity
Median	Median	Median	Median
fb 0.782609 - 0.783417	vit 0.956239	vit 0.988076 - 0.988563	mw 0.884487
mw 0.709877 - 0.711712	fb 0.930134	fb 0.977855 - 0.977961	fb 0.822751
vit 0.653553 - 0.655051	mw 0.527244	mw 0.866776 - 0.867576	vit 0.721235
Mean	Mean	Mean	Mean
fb 0.684834 - 0.685108	vit 0.926559	vit 0.976135 - 0.976174	mw 0.8132
mw 0.659582 - 0.660767	fb 0.904385	fb 0.969498 - 0.969505	fb 0.717252
vit 0.590563 - 0.590908	mw 0.650498	mw 0.886748 - 0.887218	vit 0.625244

Table 4.2: Ranked sensitivity and specificity of the three methods: vit (Viterbi method), fb (forward/backward method), mw (multi-window method). In the Exon columns the highest and lowest values for each of the three positions of a codon are shown. The median values are calculated by finding the sensitivities and specificities for each of the 4074 sequences and ranking them from lowest to highest, and then finding the median sensitivities and specificities. The mean values are found by considering all 27,085,898 nucleotides in the 4074 sequences, and finding the sensitivities and specificities of the methods on these nucleotides.

methods is regarding the sensitivity of the introns. Here, the multi-window method gives a dramatically lower value of .5272 as compared to Viterbi's .9562 and forward/backward's .9301.

The forward/backward and multi-window methods would probably be considered the highest and lowest scoring methods, respectively, overall. Although the forward/backward was second highest in three categories, in the two categories where Viterbi outscored it, it was only by .011 to .026 for the median case and .007 to .022 for the mean case. The largest difference in all four categories between the highest scorer and the forward/backward method was for intron specificity where the multi-window method outscored the forward/backward method by .062 in the median case, and .096 in the mean case.

Given that both the forward/backward and the Viterbi method make state predictions based on the entire sequence, with the Viterbi giving the most probable state sequence, and the forward/backward maximizing the probability of correctly identifying the state at every observation, it is not surprising that they tend to do better than the multi-window method.

Though the multi-window method's ability to predict a state base solely on local sequence information lowers its sensitivity and specificity, it is also an advantage in certain situations. If there is an error in the actual nucleotide sequence data, for instance a single nucleotide might not get sequenced – resulting in a single base omission for the sequence data – then a method which makes predictions based only on local information would not be as likely to be thrown off for as as long as a method which uses information from the

entire sequence. Consider for example a long exon which has had a single base not reported in the sequence data. Then the "correct" state sequence, given this incorrect data, should have a single element removed from the list $\ldots 1, 2, 3, 1, 2, 3 \ldots$. Viterbi would not allow itself to ever make a prediction that had a single element removed from this list as the transition matrix A would assign such a sequence the probability of zero. Similarly, the forward/backward method would assign a low probability to a state sequence which had any such element removed. While the multi-window method would also generally give lower sensitivities and specificities around such a removed element, once it was making predictions more than "window length" nucleotides from this error, the error would have absolutely no effect on the method's predictions.

To show the effects of these possible "mis-reads" on a sequence, one hundred randomly selected sequences were each altered by a total of 18 "insertions," "deletions" and "mutations." Insertions consisted of inserting one, two or three bases at random exon and intron locations in the sequence. Analogous alterations were made for deletions and mutations (removing base(s) and randomly changing base(s), respectively). The order of these alterations, as shown in figure 4.1 is as follows:

(1 base, 2 base, 3 base), (exon, intron), (insertion, deletion, mutation)

where choices in parentheses move from left to right, and choices in latter parentheses cycle more rapidly. Thus the first and second sensitivities plotted are a 1 exon base insertion and a 1 exon base deletion. Of particular note are the relatively low sensitivities, in all three graphs, at the first, second,

seventh (2 base exon insertion), and eighth (2 base exon deletion) points plotted. This is exactly as would be expected. Neither 3 base insertions or deletions, nor mutations disrupt the period 3 cycling of nucleotide positions within an exon. Also, as there is no cycling of positions in an intron, no alterations dramatically affect the sensitivity. Although all three methods had a drop in sensitivity for the one and two base insertion and deletions, the multi-window method suffered the least from these alterations. In addition, these alterations in longer exons would lower the sensitivity of the forward/backward and Viterbi method more dramatically while leaving the drop in sensitivity of the multi-window method the same.

4.5 Discussion

We have introduced a simple Markov model which incorporated many of the same features as our multi-window method. This served not only to give a more rigorous analysis of the sensitivity and specificity of our method, but also to compare the sensitivity and specificity of various methods. Although the multi-window method does not give as high an overall sensitivity and specificity for exon and intron predictions as either the Viterbi or forward/backward method, it still has certain valuable characteristics. First, it gives a higher intron specificity than either of the other two methods. Second, it only looks locally around an observation (a nucleotide) to predict whether the nucleotide came from the first, second, or third position of a codon, or from an intron. While this can lower its overall sensitivity and specificity as

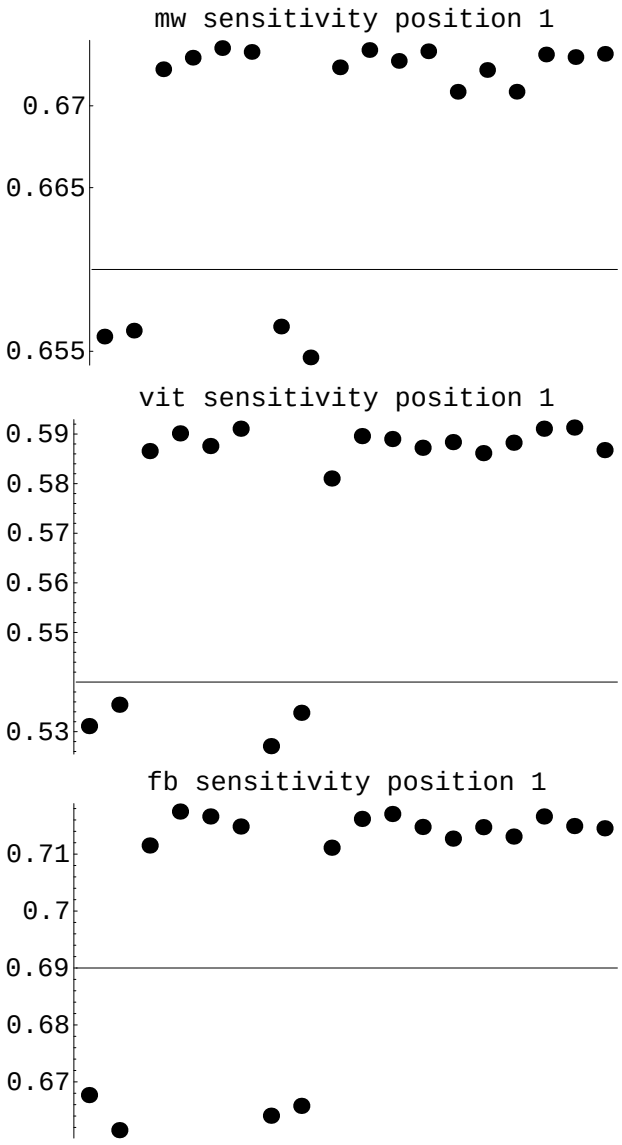

Figure 4.1: Sensitivity of position 1 detection with altered bases. See text for the ordering of the plotted points.

compared to similar methods which also incorporate information from the entire sequence, it also makes the method more robust to errors in the sequence data. Lastly, this method undoubtedly contains elements that more closely reproduce the actual biological functions which take place in the cell when the pre-mRNA is spliced by the spliceosome. That is, it is not very reasonable to believe that the spliceosome reads the entire sequence before splicing out the intron. Although there may be some interaction with distant DNA and the spliceosome, it is believed that the most dramatic signals used by the spliceosomes are local. As additional biological details on the interaction between DNA and spliceosomes become available, it may be that methods, such as our multi-window method, which more closely approximate the spliceosomes's biological role will be able to more readily incorporate this new information to produce methods with higher sensitivity and specificity.

Bibliography

[Durbin et al., 1998] Durbin, R., Eddy, S., Krogh, A., and Mitchison, G. (1998). *Biological sequence analysis*. Cambridge University Press.

[Forney, 1973] Forney, G. D. (1973). The Viterbi algorith. *Proceedings of the IEEE*, 61:268–278.

[Rabiner, 1989] Rabiner, L. R. (1989). A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE*, 77(2):257–286.

[Ruvinsky et al., 2005] Ruvinsky, A., Eskesen, S. T., Eskesen, F. N., and Hurst, L. D. (2005). Can codon usage bias explain intron phase distributions and exon symmetry? *J Mol Evol*, 60(1):99–104.

[Saxonov et al., 2000] Saxonov, S., Daizadeh, I., Fedorov, A., and Gilbert, W. (2000). EID: the Exon-Intron Database-an exhaustive database of protein-coding intron-containing genes. *Nucleic Acids Res*, 28(1):185–190.

[Viterbi, 1967] Viterbi, A. W. (1967). Error bounds for convolutional codes and an asymptotically optimal decoding algorithm. *IEEE Trans. Infromat. Theory*, IT-13:260–269.

Chapter 5

Exon Detection using Likelihood Ratios with the Incorporation of GeneSplicer

5.1 Abstract

A shortcoming of exon detection methods, including our multi-window method, which use relatively long nucleotide subsequences to make a prediction as to a nucleotide's biological function, is their poor ability to find exact boundaries between sequences with distinct functions. GeneSplicer [Pertea et al., 2001] is a splice site prediction method with high sensitivity and specificity. Combining our multi-window method with GeneSplicer's splice site predictions helps alleviate this problem of indistinct boundaries and gives our method higher overall sensitivity and specificity.

5.2 Introduction

The multi-window method described in Chapter 3 makes a prediction as to whether a nucleotide is from an exon or intron based on likelihood ratios which use data from three overlapping subsequences or "windows." Ideally, when these windows are entirely in an exon or intron, the likelihood ratios will correctly identify the base's region, but as the windows move from exon to intron, or vice versa, the windows contain data from both regions, and the method's performance drops. In an attempt to solve this problem, we incorporate the splice site predictor GeneSplicer, which can help predict the exact boundary between exons and introns.

GeneSplicer uses a decision tree method developed by Burge and Karlin [Burge and Karlin, 1997], called maximal dependence decomposition. In addition, Markov models are used to extract information in small windows around the splice sites and sequence statistics are used on larger windows to help distinguish between exons and introns. Finally, Brendel and Kleffe's [Brendel and Kleffe, 1998] local score optimality feature is used to increase the accuracy of the method.

By combining the predictions from our multi-window method with those from GeneSplicer, we were able to increase the overall sensitivity and specificity of our method.

5.3 Biological Background

A brief summary of the pertinent biology is given here. For a more detailed account the reader is referred to the Biological Background section in Chapter 2. The following books also give further information on the topic: [Lewin, 1994], [Fairbanks and Anderson, 1999] and [Snustad et al., 1997].

Human chromosomes are composed of tightly coiled threads of deoxyribonucleic acid (DNA) and associated protein molecules which aid in the structural packing of the DNA. The DNA itself is often compared to a twisted ladder with the sides of the ladder being the sugar-phosphate backbone of the DNA, and the rungs being the two complementary nucleotides that bind to one another - one from each of the two strands of DNA [Watson and Crick, 1953]. A single strand of DNA may be thought of as a sequence of four nucleotides: adenine (A), cytosine (C), guanine (G) and thymine (T). The nucleotides that bind to one another to form the "rungs" are called complementary pairs: A binds with T and C binds with G.

The DNA is always read by the cell machinery in the same orientation. That is, the sequence AATCGTA of nucleotides (bases) along a strand of DNA would always be read in the order indicated above, or in the reverse as ATGCTAA, but not in both orders. The end of the sequence, where the reading starts, is the 5' end and the other is the 3' end. The complementary strand always has the reverse orientation. Thus if one strand of a chromosome had the sequence 5'- AATCGTA - 3', then this would be bound to the sequence 3' - TTAGCAT - 5'.

The genes within the DNA are the genetic code used by the cell to make proteins. In higher eukaryotes these genes comprise only a small percentage of the entire genome – the entire DNA sequence of an organism – which in humans is some three billion nucleotides long. A typical human gene is a few thousand bases long. There are many genes on both strands of the DNA of a chromosome. Humans have 23 pairs of chromosomes and somewhere on the order of 30,000 genes.

Transcription

An initial stage of protein synthesis is the transcription of the DNA into messenger RNA (mRNA). This mRNA transfers the information from the DNA in the nucleus of the cell out into the cytoplasm of the cell where the protein is synthesized. RNA is a molecule very similar in structure to DNA, except that thymine is replaced by the nucleotide uracil (U), and RNA uses the sugar ribose instead of deoxyribose for its sugar-phosphate backbone. If a subsequence on one strand is a gene, then this strand is known as the sense strand for this portion of the double helix. The complementary DNA strand is used by an enzyme (a catalytic protein) known as RNA polymerase II to synthesize the mRNA. This complementary strand is known as the template or antisense strand. The non-template, or sense, strand has the sequence in the orientation in which genes are reported.

Splicing

The newly synthesized mRNA is known as pre-mRNA at this stage as it must undergo chemical modifications to its beginning and end. Often chemical modification is followed by "splicing" where precise, predefined, subsequences are spliced out and degraded. These subsequences are called introns (INTeRvening sequences); the subsequences which are joined together to make the mature mRNA, are called exons (EXpressed sequences). The joined exons, called "mature mRNA" or simply "mRNA," pass out of the nucleus of the cell to the cytoplasm where protein synthesis occurs.

The start and end of the intron are known as the donor (or 5') and acceptor (or 3') splice sites, respectively. The initial GU and terminal AG of an intron are the only highly conserved sequences in the introns. Less well conserved sequences are shown in figure 5.1; in particular, at the donor and acceptor splice sites as well as at a sequence known as the branch point sequence, which is generally 30 bases upstream from the acceptor splice site, we find longer less well conserved sequences. Although this is useful information, the sequences given at the donor splice site and branch-point occur only 22 and 40 percent of the time respectively (and the branch-point sequence was only determined up to two bases in most instances), making these moderately conserved signals of limited value in splice site detection.

5' $\ldots E_n$ | donor splice site $\ldots$ branch pt. seq. $\ldots$ acceptor splice site | E_{n+1} $\ldots$ 3'

$A_{64}G_{73}$ | $G_{100}U_{100}A_{62}A_{68}G_{84}U_{63}$ $\ldots Y_{80}NY_{80}Y_{87}R_{75}A_{100}Y_{95}$ $\ldots 12YNC_{65}A_{100}G_{100}$ | N

Figure 5.1: Consensus sequences for regions of an intron. E_k is the k^{th} exon of the gene. | denotes an exon/intron or intron/exon boundary. R - a puRine (an A or G base), Y - a pYrimidine (a C or T/U), N - aNy nucleotide. The subscripts give the percentage occurrences of these bases. Subscripts of 100 are rounded, and there are many known exceptions (and many more may be found when introns are searched for without assuming that they start and end with these sequences. See http://www.ebi.ac.uk/asd/altextron/-pre-release-dist-data.html for current percentages of donor/acceptor splice sites).

Alternative Splicing

To add to the problem of splice site detection, it is estimated that one half of the human genes that are spliced can undergo alternative splicing. Alternative splicing yields different (viable) proteins through a variety of means: alternate donor splice site, alternate acceptor splice site, exon skipping, and splice vs. no splice. When the intron is spliced at a different starting point, this is known as alternative donor splice site splicing. If these alternate starts to the splice site are off by a multiple of three nucleotides, then amino acids corresponding to the differing bases added or lost will be added or deleted from the final protein. If, on the other hand, the alternate starts are not off by a multiple of three, then there is a frame "shift" in the codons (see section on translation), and thus all subsequent corresponding amino acids can be different. A similar situation holds for alternate acceptor splice sites.

As the name implies, exon skipping occurs when an exon is skipped in the

splicing process. Thus one less exon is incorporated into the mature mRNA. If the exon has a length that is a multiple of three, then a certain region of the resulting protein is excised; whereas if its length is not a multiple of three, then a frame shift occurs with the above mentioned consequences.

Splicing vs. no splicing is similar to exon skipping, but instead of excising an exon, an intron is incorporated instead of being spliced out. Once again the length of the intron determines a possible frame shift.

Translation

The information in the mRNA is used to synthesize protein in a process known as translation. The genetic code of this mRNA is read in consecutive, non-overlapping sets of three nucleotides. Each of these triplets codes for a particular amino acid – the subunits of a protein. Thus a sequence of DNA has three frames, only one of which is used to make a particular protein. Consider the sequence ... TACGGTAATCCGGGT .... Since the sequence is read in triplets, it could be read as

... TAC GGT AAT CCG GGT...,

... T ACG GTA ATC CGG GT... or

... TA CGG TAA TCC GGG T...,

each of which would code for an entirely different amino acid sequence. The triplets in the proper frame, which are used for protein synthesis, are called codons.

There are 64 codons (four possible nucleotides in each of the three locations). Three of the codons (TAA, TAG, and TGA, or their more commonly

used mRNA counterparts: UAA, UAG, and UGA) signal that protein synthesis should stop, and are thus known as stop codons. The other 61 each code for one of the 20 amino acids. Some amino acids are coded for by only a single codon, while others have as many as six. The correspondence between a codon and its associated amino acid, or function as a stop codon, is so consistent over all organisms (although exceptions exist), that it is known as the "universal code."

Protein synthesis occurs on a cellular organelle known as a ribosome. When one of the many ribosomes in the cytoplasm of the cell comes in contact with the 5' end of the mRNA, the ribosome becomes attached to it. The ribosome "reads," or moves down the mRNA three bases, or one codon, at a time. There are two sites on the ribosome each of which can hold a codon from the mRNA, the transfer RNA (tRNA), and the tRNA's associated amino acid. Each tRNA has a three base anticodon on it. Only a tRNA with an anticodon that matches the mRNA codon (by complementary base pairing), which is docked on the ribosome, can dock at the ribosome site. This assures that the correct amino acids are placed in close proximity. The amino acid from the most recently attached tRNA is then attached to the growing polypeptide chain. Thus the DNA message is faithfully transferred via the mRNA to the final polypeptide.

5.4 Overview of GeneSplicer

GeneSplicer [Pertea et al., 2001] is a computational splice site prediction method which combines various techniques to provide a program which gives accuracies comparable to, or better than, the best alternative programs [Pertea et al., 2001].

GeneSplicer uses a binary decision tree method called maximal dependence decomposition which was first introduced by Burge and Karlin [Burge and Karlin, 1997], and improves it with the addition of Markov models which detect additional dependencies among nucleotides around the splice site. Only relatively small windows of nucleotides are examined here, but they seem to capture the majority of the biological information used by the spliceosomes. In addition, the method uses longer windows to either side of a splice site to detect statistical differences between the exon and intron regions on both sides of a splice site. Finally, it employs a local score optimality feature similar to that used by Brendel and Kleffe [Brendel and Kleffe, 1998] to exclude many false positive splice sites.

5.4.1 Maximal Dependence Decomposition

Maximal dependence decomposition was developed to identify the most significant dependencies between positions of a splice site. It is a generalization of the weight array model [Zhang and Marr, 1993], and the weight matrix method [Staden, 1984]. The weight matrix method uses the relative frequen-

cies p_j^i of the j^{th} nucleotide at position i to estimate the probability

$$Prob(X) = \Pi_{i=1}^{n} p_{x_i}^i$$

of generating the sequence $X = x_1, x_2, \ldots, x_n$. The weight array model, which takes into account dependencies between adjacent sites, calculates the probability as

$$Prob(X) = p_{x_1}^1 \Pi_{i=2}^{n} p_{x_{i-1},x_i}^{i-1,i},$$

where $p_{j,k}^{i-1,i}$ is the conditional probability of nucleotide x_k at position i given that the nucleotide at position $i-1$ is x_j.

Maximal dependence decomposition starts with a set D of N aligned sequences of length k. These sequences could be any type of biological signal for which dependencies between nucleotides is sought. Burge and Karlin use the nine nucleotide sequence that corresponds to the last three bases of an exon and the first six bases of the intron of a donor splice site. The positions are denoted -3, -2 -1, 1, 2, 3, 4, 5, and 6 with positions 1 and 2 always being the canonical GT (or GU in the tRNA) in the set D. The most frequently occurring base(s) at each position is/are called the consensus base(s), and an indicator variable C_i is assigned the value 1 if the i^{th} base of a given sequence of D is equal to the consensus base(s), and 0 otherwise. The nucleotide indicator X_j identifies the nucleotide at position j. For each pair of i, j with $i \neq j$, a contingency table is formed. The χ^2 values with i or j equal to one or two are omitted from their table as these positions do not have any variability in their data set. Of the remaining 42 i, j pairs, 31 have a significant χ^2 value at the relatively stringent level of $P < 0.001, df = 3$. This

shows that there is a great deal of dependence among these nine nucleotides. Next, the sum

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

is calculated, which gives a measure of the dependence between C_i and the nucleotides at the other positions. A binary decision tree is used to subdivide their set as follows. Choose the value i_1 such that S_{i_1} is maximal, and partition D into two subsets, D_{i_1} and D_{i_1}-. D_{i_1} contains all the sequences from D which have the consensus nucleotide(s) at position i_1 and D_{i_1}- contains the sequences which do not.

Each of these subsets is recursively subdivided until one of the following three conditions is met: i) the $k-1^{th}$ level of the tree is reached (and thus no further subdivision is possible); ii) no significant dependencies between positions are found; or iii) the size of the subset is small enough that further subdivision would result in weight matrix method frequencies that would be unreliable. Burge and Karlin derive a separate weight matrix method model for each subset of the tree, and use them in their larger hidden Markov model. Pertea *et al* use these final subsets (or "leaves") of the binary decision tree, but in addition construct first-order Markov chain models using a 16 base (29 base) region around the donor (acceptor) splice sites. First-order Markov chain models are also constructed around false donor and acceptor splice sites (GT and AG dinucleotides that did not correspond to true donor/acceptor splice sites). The score given to a potential splice site is given by the difference between the log-odds score of the site as computed by the true Markov model,

and that of the false Markov model. The Markov model score of a sequence is given in [Salzberg et al., 1998] and [Salzberg et al., 1999]. The score of a particular site $s_i, s_{i+1}, \ldots, s_j$ is computed for both the true and the false Markov model. Let the score of the site starting at i and ending at j be given by

$$S(i,j) = \Sigma_{k=i}^{j} M_{s,k},$$

where

$$M_{s,k} = \ln \frac{f((s_{k-2}, s_{k-1}, s_k), k)}{f((s_{k-2}, s_{k-1}), k-1)}$$

and $f(s,k)$ is the frequency of the subsequence s ending at position k.

5.4.2 Sequence Statistic in Larger Windows

Pertea *et al* [Pertea et al., 2001] next construct two second-order Markov models from the exon and intron regions to either side of the splice sites. They used subsequences of length 80 bases to either side of a splice site. Thus for exons or introns adjacent to the splice site that are shorter than 80 bases, some non-exon, or non-intron data is incorporated into their training set, but they state that this is a relatively rare event, and only slightly alters the Markov probabilities.

Let $S_{comb}(k,i)$ be the score given above computed from the maximal dependence decomposition with the added Markov chain, and let $S_{cod}(j)$ and $S_{noncod}(j)$ be the scores from the coding and non-coding Markov models respectively. Then the score given to a splice site is calculated as follows:

$$S(k) = S_{comb}(k,16) + [S_{cod}(k-80) - S_{noncod}(k-80)] + [S_{noncod}(k+1) - S_{cod}(k+1)]$$

and

$$S(k) = S_{comb}(k, 29) + [S_{noncod}(k-80) - S_{cod}(k-80)] + [S_{cod}(k+1) - S_{noncod}(k+1)]$$

where k is the position of the donor site in the first equation, and the position of the acceptor site in the second.

5.4.3 Local Score Optimality Feature

Finally, Pertea *et al* [Pertea et al., 2001] apply a local score optimality feature similar to that of Brendel and Kleff [Brendel and Kleffe, 1998] which eliminates many false positives. In particular, only the highest scoring splice site, of a particular type (donor or acceptor), is kept in any 60 base window.

5.5 Methods

To combine the information from our multi-window method and that from GeneSplicer, we found all locations where our predictions changed from exon to intron, and then checked to see if there was a GeneSplicer predicted donor splice site "close by." Recall that due to the length of the window, our predictions are based on nucleotides that are downstream from the nucleotide in question. Thus we expect our predictions to foreshadow the true classification of the sequence by about one half of the length of the window. Therefore we looked in a region centered one half a window length downstream from our change in prediction, and if there was a GeneSplicer predicted donor site with sufficiently high score, we changed all our predictions from intron

to exon for all bases in this region that are prior to GeneSplicer's predicted donor splice site. An analogous treatment was done for acceptor splice sites.

Both GeneSplicer cutoff score values and lengths of the aforementioned regions were analyzed to maximize the combined method's overall sensitivity and specificity.

5.6 Results

Although combining the multi-window method with GeneSplicer enhanced the accuracy of prediction of many bases, many inaccurately predicted bases were left unchanged for two main reasons. First the majority of inaccurate base predictions came not from bases close to a true splice site, but from regions in introns that had triplet composition more closely approximating that of the average exon rather than the average intron. To a lesser extent we also found intron-like regions in exons.

Figures 5.2 and 5.3 are histograms of the distribution of distances from inaccurately predicted bases to the nearest true splice site for a typical sequence. Distances are measured as (location of error - location of nearest splice site), and thus can be positive or negative in both the donor and acceptor histograms. In the donor histogram, the majority of distances are positive as these refer to errors in true introns; and as introns span a larger portion of the entire sequence, we find more errors here. Similarly, for the acceptor histogram, we find more negative distances. For both the donor and acceptor cases, we see that there are more positive and negative dis-

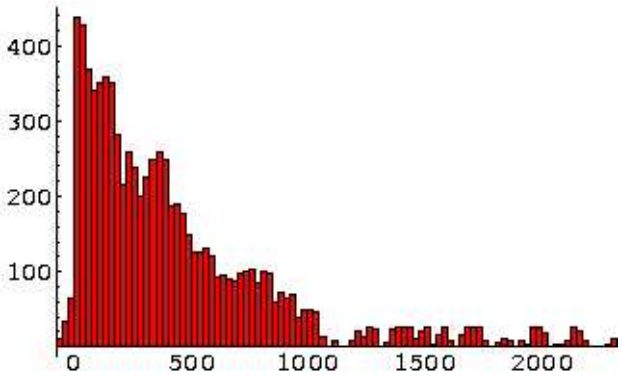

Figure 5.2: Frequency distribution of distances to donor splice sites

tances close to zero than any other category, but still the majority of errors are farther from zero than the multi-steps window length, and thus can not be correctly adjusted even if GeneSplicer correctly predicts all donor and acceptor sites.

The second main problem with combining the two methods is that GeneSplicer misses some true splice sites, and gives quite a few false predictions. In their paper, Pertea *et. al* [Pertea et al., 2001] give various values of true sites missed for both acceptor and donor site detection and their corresponding false positive values. At 20 percent of the true sites missed, for example, they report a false positive rate of 1.1 percent for acceptor site detection. This 1.1 percent is calculated as the number of false positives they predict divided by the number of what they term "false acceptors," – AG dinucleotides which are not true acceptor splice sites. If on the other hand one considers the number of acceptor sites they predict which are not true acceptor sites, divided by the

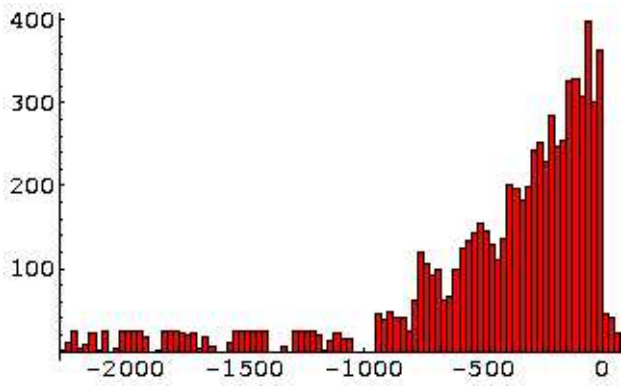

Figure 5.3: Frequency distribution of distances to acceptor splice sites

total number of acceptor sites they predict, (a common definition for the term "false positive"), then, for example, when 24 percent of the true acceptor sites are missed, 79 percent of their predictions are not true acceptor sites. For both of these reasons, with the former contributing more, many incorrectly predicted bases are not changed when these methods are combined.

Table 5.1 gives the exon and intron sensitivity and specificity both with and without the aid of GeneSplicer. Exon sensitivity and intron specificity are increased by .0167 and .0219, respectively. Exon specificity and intron sensitivity on the other hand are slightly lowered – .0022 and .0059, respectively.

5.7 Discussion

Overall, the performance of the union of these two methods gave poorer than expected results. Both the fact the GeneSplicer did not perform as well as

Exon Sensitivity	Intron Sensitivity	Exon Specificity	Intron Specificity
w/o GeneSplicer mw 0.659582 - 0.660767	w/o GeneSplicer mw 0.650498	w/o GeneSplicer mw 0.886748 - 0.887218	w/o GeneSplicer mw 0.8132
w/ GeneSplicer mw 0.676544 - 0.677179	w/ GeneSplicer mw 0.644608	w/ GeneSplicer mw 0.884756 - 0.88516	w/ GeneSplicer mw 0.835065

Table 5.1: Sensitivity and specificity of the multi-window method (mw) with and without the use of GeneSplicer.

anticipated, and that the majority of the incorrect predictions of the multi-window method were not due to the window overlapping regions of distinct biological function, contributed to this lower performance.

Bibliography

[Brendel and Kleffe, 1998] Brendel, V. and Kleffe, J. (1998). Prediction of locally optimal splice sites in plant pre-mRNA with applications to gene identification in Arabidopsis thaliana genomic DNA. *Nucleic Acids Res*, 26(20):4748–4757.

[Burge and Karlin, 1997] Burge, C. and Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. *J Mol Biol*, 268(1):78–94.

[Fairbanks and Anderson, 1999] Fairbanks, D. J. and Anderson, R. W. (1999). *Genetics*. Brooks/Cole.

[Lewin, 1994] Lewin, B. (1994). *Genes V*. Oxford University Press.

[Pertea et al., 2001] Pertea, M., Lin, X., and Salzberg, S. L. (2001). GeneSplicer: a new computational method for splice site prediction. *Nucleic Acids Res*, 29(5):1185–1190.

[Salzberg et al., 1998] Salzberg, S., Delcher, A. L., Fasman, K. H., and Henderson, J. (1998). A decision tree system for finding genes in DNA. *J Comput Biol*, 5(4):667–680.

[Salzberg et al., 1999] Salzberg, S. L., Pertea, M., Delcher, A. L., Gardner, M. J., and Tettelin, H. (1999). Interpolated Markov models for eukaryotic gene finding. *Genomics*, 59(1):24–31.

[Snustad et al., 1997] Snustad, P. D., Simmons, M. J., and Jenkins, J. B. (1997). *Principles of Genetics*. Wiley.

[Staden, 1984] Staden, R. (1984). Computer methods to locate signals in nucleic acid sequences. *Nucleic Acids Res*, 12(1 Pt 2):505–519.

[Watson and Crick, 1953] Watson, J. D. and Crick, F. H. C. (1953). A structure for Deoxyribose Nucleic Acid. *Nature*, 171:737.

[Zhang and Marr, 1993] Zhang, M. Q. and Marr, T. G. (1993). A weight array method for splicing signal analysis. *Comput Appl Biosci*, 9(5):499–509.