Report Number: WUCSE-2007-5

2007

Leveraging EST Evidence to Automatically Predict Alternatively Spliced Genes, Master's Thesis, December 2006

Robert Zimmermann

Current methods for high-throughput automatic annotation of newly sequenced genomes are largely limited to tools which predict only one transcript per gene locus. Evidence suggests that 20-50% of genes in higher eukariotic organisms are alternatively spliced. This leaves the remainder of the transcripts to be annotated by hand, an expensive time-consuming process. Genomes are being sequenced at a much higher rate than they can be annotated. We present three methods for using the alignments of inexpensive Expressed Sequence Tags in combination with HMM-based gene prediction with N-SCAN EST to recreate the vast majority of hand annotations in the D.melanogaster... **Read complete abstract on page 2.**

Leveraging EST Evidence to Automatically Predict Alternatively Spliced Genes, Master's Thesis, December 2006

Robert Zimmermann

Complete Abstract:

Current methods for high-throughput automatic annotation of newly sequenced genomes are largely limited to tools which predict only one transcript per gene locus. Evidence suggests that 20-50% of genes in higher eukariotic organisms are alternatively spliced. This leaves the remainder of the transcripts to be annotated by hand, an expensive time-consuming process. Genomes are being sequenced at a much higher rate than they can be annotated. We present three methods for using the alignments of inexpensive Expressed Sequence Tags in combination with HMM-based gene prediction with N-SCAN EST to recreate the vast majority of hand annotations in the D.melanogaster genome. In our first method, we “piece together” N-SCAN EST predictions with clustered EST alignments to increase the number of transcripts per locus predicted. This is shown to be a sensitve and accurate method, predicting the vast majority of known transcripts in the D.melanogaster genome. We present an approach of using these clusters of EST alignments to construct a Multi-Pass gene prediction phase, again, piecing it together with clusters of EST alignments. While time consuming, Multi-Pass gene prediction is very accurate and more sensitive than single-pass. Finally, we present a new Hidden Markov Model instance, which augments the current N-SCAN EST HMM, that predicts multiple splice forms in a single pass of prediction. This method is less time consuming, and performs nearly as well as the multi-pass approach.

2007-5

Leveraging EST Evidence to Automatically Predict Alternatively Spliced Genes, Master's Thesis, December 2006

Authors: Robert Zimmermann

Corresponding Author: rpz@cse.wustl.edu

Web Page: http://nijibabulu.org/bz

Abstract: Current methods for high-throughput automatic annotation of newly sequenced genomes are largely limited to tools which predict only one transcript per gene locus. Evidence suggests that 20-50% of genes in higher eukariotic organisms are alternatively spliced. This leaves the remainder of the transcripts to be annotated by hand, an expensive time-consuming process. Genomes are being sequenced at a much higher rate than they can be annotated. We present three methods for using the alignments of inexpensive Expressed Sequence Tags in combination with HMM-based gene prediction with N-SCAN EST to recreate the vast majority of hand annotations in the D.melanogaster genome. In our ﬁrst method, we “piece together” N-SCAN EST predictions with clustered EST alignments to increase the number of transcripts per locus predicted. This is shown to be a sensitve and accurate method, predicting the vast majority of known transcripts in the D.melanogaster genome. We present an approach of using these clusters of EST alignments to construct a Multi-Pass gene prediction phase, again, piecing it together with clusters of EST alignments. While time consuming, Multi-Pass gene prediction is very accurate and more sensitive than single-pass. Finally, we present a new Hidden Markov Model instance, which augments the current N-SCAN EST HMM, that predicts multiple splice forms in a single pass of prediction. This method is less time consuming, and performs nearly as well as

Type of Report: Other

SEVER INSTITUTE OF TECHNOLOGY

MASTER OF SCIENCE DEGREE

THESIS ACCEPTANCE

(To be the first page of each copy of the thesis)

DATE: December 12, 2006

STUDENT'S NAME: Bob Zimmermann

This student's thesis, entitled Leveraging EST Evidence to Automatically Predict Alternatively Spliced Genes has been examined by the undersigned committee of three faculty members and has received full approval for acceptance in partial fulfillment of the requirements for the degree Master of Science.

APPROVAL: _____ Chairman

_____

_____

WASHINGTON UNIVERSITY

THE HENRY EDWIN SEVER GRADUATE SCHOOL

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LEVERAGING EST EVIDENCE TO AUTOMATICALLY PREDICT ALTERNATIVELY SPLICED GENES

by

Bob Zimmermann, B.S.

Prepared under the direction of Michael Brent

A thesis presented to the Henry Edwin Sever Graduate School of
Washington University in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

December 2006

Saint Louis, Missouri

WASHINGTON UNIVERSITY
THE HENRY EDWIN SEVER GRADUATE SCHOOL
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

ABSTRACT

LEVERAGING EST EVIDENCE TO AUTOMATICALLY PREDICT
ALTERNATIVELY SPLICED GENES

by

Bob Zimmermann

ADVISOR: Michael Brent

December 2006

Saint Louis, Missouri

Current methods for high-throughput automatic annotation of newly sequenced genomes are largely limited to tools which predict only one transcript per gene locus. Evidence suggests that 20-50% of genes in higher eukariotic organisms are alternatively spliced. This leaves the remainder of the transcripts to be annotated by hand, an expensive time-consuming process. Genomes are being sequenced at a much higher rate than they can be annotated. We present three methods for using the alignments of inexpensive Expressed Sequence Tags in combination with HMM-based gene prediction with N-SCAN_EST to recreate the vast majority of hand annotations in the *D.melanogaster* genome. In our first method, we "piece together" N-SCAN_EST predictions with clustered EST alignments to increase the number of transcripts per locus predicted. This is shown to be a sensitve and accurate method, predicting the vast majority of known transcripts in the *D.melanogaster* genome. We present an approach of using these clusters of EST alignments to construct a Multi-Pass gene prediction phase, again, piecing it together with clusters of EST alignments. While time consuming, Multi-Pass gene prediction is very accurate and more sensitive than single-pass. Finally, we present a new Hidden Markov Model instance, which augments the current N-SCAN_EST HMM, that predicts multiple splice forms in a single pass of prediction. This method is less time consuming, and performs nearly as well as the multi-pass approach.

to Laura,
whose unparalleled bravery and positivity
will forever inspire me

Contents

List of Figures

Acknowledgments

I would like to thank all of the excellent faculty I had the privilege of interacting with along the way, especially (in alphabetical order) Michael Brent, Jeremy Buhler, Gary Stormo, and Sean Eddy. Additionally, I would like to thank the present and past members of my research group, Randy Brown, Sam Gross, Mani Arumugam, Jeltje Van Baren, Beth Frazier, Aaron Tenney, Chauchun Wei, Suman Kumar, Catherine Beauheim and Laura Langton for all of their support, advice and tolerance. Thanks for the less direct (but equally vital) support to my parents Imo and Paul Zimmermann, my sister Laura Kokesh, my friends in St. Louis and Ann Arbor for their patience, especially Matthew Cunningham, Sam Moyerman and Gretta Treiber for cleaning my apartment and taking care of my cats, for whom I am also endebted for their patience. Alan Kwan has not only offered me boundless advice and support but also stayed up for hours doing projects with me, intermittently stealing away to the diner for grease and coffee. Justin Levine, Morgan Deters, my billiards team, Angela Duff, Derrick Mosley, Sarah Roland, Ben West and others I'm probably forgetting all deserve credit for the best times I've had in St. Louis, keeping me afloat during crazy work schedules. Finally, a special thanks goes to Brian Koebbe for giving me ranch059 last weekend, and to Charles Comstock for doing Primer Design this summer, so I didn't have to.

With direct relation to this work, it should be noted that many of the ideas for N-SCAN_MP_EST were Jeltje van Baren's. If I were generous to myself, I might call it a collaboration, but I was never one to split hairs.

Bob Zimmermann

Washington University in Saint Louis
December 2006

Preface

Since the earliest sequences of the human genome were released, many computer scientists, mathematicians and biologists have been developing a new interdisciplinary synergy called bioinformatics. Indeed, it has not been a straight, well-paved path. With the bombardment of genetic and biological data has come research that challenges our understanding of the workings of molecular biology of the cell. The data we acquire demand new ideas and approaches. While this can create a lot of confusion amongst all camps of biologists and bioinformaticians alike, it presents an interesting challenge for the future of academics: can we as scientists grapple with a field that has few constants, harnessing all the data that comes our way? Can we consistently adapt our methods and invent new approaches as quickly as our understandings change?

Unlike many subfields of bioinformatics, the core problem of prediction of protein coding genes has essentially remained the same. How can we most accurately predict the locations of protein coding genes with a given amount of evidence? Over the last five years, however, what the "given amount of evidence" signifies has changed drastically. There are many more genomes sequenced, EST data has increased hugely, mass spectrometry has improved, and chromatin structure is being explored. Sequencing is becoming cheaper, microarrays are improving, lessening the need for pure *ab initio* gene prediction. Given the massive data available for some genomes, the utility of gene predictors becomes questionable.

Fortunately for us, genome sequencing is not likely to halt any time soon, and wide support for sequencing-based research in all genomes will probably not be available. The question now becomes, when we are presented with a new genome, how much evidence do we need to reliably annotate the protein-coding regions? What kind of data best guides our predictors to find the most correct exon-intron boundaries? Can the shortcomings of gene predictors be overcome by the intelligent use of extrinsic evidence?

We certainly hope that this question will be more closely studied. The future of genomics will necessarily revolve around a close interplay between computational and experimental methods if it is to continue to thrive. Hand annotation simply cannot keep up with the pace of genome sequencing.

In this thesis, we address one facet of this complicated problem: how can we recreate hand annotations in an automated fashion, given limited data from the target genome? How close can we get to the original annotations? What direction should we take to come closer to realizing automated full genome annotation?

One of the biggest challenges facing gene predictors today is alternate splice form prediction. Given that these events are frequent and play a large role in genome diversity, a good modern gene predictor should address this issue. In this thesis, we take a look at some methods for tackling this.

Additionally, we take a look at further harnessing the evidence from EST sequences to predict additional gene structures, probabilistically and heuristically.

Chapter 1

Introduction

The large aim of genome annotation is to predict, align, sequence and otherwise guess the locations of features in the DNA of a genome which have some meaningful biological function. More narrowly, protein genome annotation, the one of interest to this thesis, is the subfield in which protein-coding DNA features are annotated. While determining pairs of coordinates in a genome seems simple enough, the task is everything but simple: in the case of human, the task is to determine the protein coding regions occupying 1–2% of the total length of the genome.

Further, many red herrings exist in the form of retrotransposed elements (genes "re-copied" into the genome) and untranslated elements (transcribed gene-like sequences that do not become a protein) as well as oddballs such as selenocystine-encoding genes (ones which appear as though they should stop before the end of actual translation). It is also discovered that the phenomenon of alternative splicing, where one region can account for more than a single protein product, further complicating the problem.

It is widely undisputed that the most reliable annotations are the ones that are human-verified. Computer programs, while getting better and better may never reach this level of accuracy. But, as the case was with genome sequencing, the sheer mass of the data demands automation. The rate of genome sequencing is increasing, and not all genomes will receive the same attention that the earlier genomes of human, mouse, fruit fly and worm received.

How can we address this issue without further relegating duties of annotation refinement to the "sequence gazers"? The answer, as it often happens to be, may lie in the use of computer systems to make educated guesses.

1.1 Bioinformatics and genome annotation

The role of the bioinformatician is to develop, prove and explain computer-accelerated methods for handling biological data. This is extraordinarily open-ended. Unlike many other fields, no pretense for profound solutions over simple solutions exists. This is good, since this encourages research by any means necessary to analyze the massive amount of biological data incoming.

Only scratching the surface of the current research of bioinformatics can be overwhelming. Methods, algorithms and models are developed and published daily en masse. A lot of this research overlaps in its aims and usefulness, confusing the user.

The field of genome annotation is no exception. Many probabilistic, comparative, and discriminative methods exist, and no single "leader" has emerged. This is troubling, since in many instances, antiquated, but reputable, methods are favored over modern ones, and redundant research is continually carried out. While it is not the business of research to keep a leaderboard which may marginalize the subtleties of a given research project, gene prediction has come of age, nearing the 10th anniversary of the advent of Genscan. The time may have come to consolidate our resources and find an exhaustive method, combining as many sources of information as possible.

1.2 The spectrum of gene prediction

Foissac and Schiex propose two major categories of bioinformatics-driven genome annotation methods: *intrinsic* and *extrinsic* [12]. *Intrinsic* methods are those which use genomic DNA of one or more species to predict genes in a target genome. These annotators are usually founded on probabilistic or discriminative models exclusively. *Extrinsic* methods are those which use known transcribed sequences and intelligently match those sequences to their putative originating genomic regions. These generally include RT-PCR sequencing of RNA or the use of DNA mircoarray data.

The major advantage of *intrinsic* methods is the low cost and the lack of expression bias. *Intrinsic* methods do not inherently discriminate against a gene which is under-expressed,[1] a persistent problem with *extrinsic* data. *Intrinsic* methods, however, use models which are biased toward average genes, and will miss "oddball" genes: genes embedded in other genes[2], genes with non-canonical boundary signals[3], and those "red herrings" and oddballs listed above. These are the situations in which *extrinsic* methods work best.

In the same paper, they point out that these two methods are not mutually exclusive to each other. Those methods "in-between" might prove as our best approximation to hand annotation.

1.3 ESTs and their uses

Expressed Sequence Tags (ESTs) are short sequences from expressed RNA, and are considered strong evidence for transcription. They usually represent a partial sequencing of a gene except in cases where the gene is very short. ESTs can be systematically and inexpensively produced, making them an attractive alternative to full-length gene sequencing.

In order to take full advantage of the data they imply, ESTs must be matched to their most likely place in the source genome from where they were transcribed. This is known as alignment. Many packages have been built specifically to this purpose, including `EST_GENOME` [25].

While this is helpful in detecting a likely transcribed region EST alignments alone do not suffice for gene finding. The problem is that the transcription initiation and termination[4] sites cannot be reliably determined with ESTs. Many programs have been designed to intelligently classify and cluster ESTs for gene prediction, including EbEST. Even these results do not prove to be as strong as *intrinsic* methods such as `FGENEH` [18].

[1] Although it is certainly possible that there is bias in gene predictors due to training methods, it does not require that the input data include all genes that are to be predicted.

[2] I refer here to genes in introns, but forsake the terminology for the purposes of introduction.

[3] Specifically, splice sites.

[4] Barring the use of poly-A sequences.

More recently, a program called PASA was developed to update genome annotations, harnessing full gene sequences, the original annotation as well as ESTs to produce a full picture of the genome annotation [17]. This greatly improved the *Arabidopsis* genome annotation. A large part of its strength in specificity is that it only relies on full gene sequences and known annotations to have complete information on a transcribed protein-coding gene, rather than allowing ESTs alone to predict a gene.

1.4 The crux of the modeling problem

One of the most popular *intrinsic* methods for computational gene prediction employs probabilistic modeling of the structure of genes. Predicting genes with probabilistic models is trainable to all genomes, and in many cases does not even require supervised training on the target genome [23]. Data from related genomes can be used to bootstrap annotations of new genomes.

Probabilistic modeling of gene structures is also not limited to expression data. In *de novo* gene prediction, the predictor is not constrained to regions where EST alignments exist, and is capable of predicting novel genes [32].

The flexibility of these methods come at a price: only one transcript is predicted per gene. This is problematic since genes in higher organisms take many different forms in the same region. If a computer model is to provide a similar basis for annotation improvement as it does in hand annotation, this must be accounted for.

1.5 The proposed solution

Modern gene predictors are highly accurate. It is not difficult to imagine using the gene predictions, rather than hand annotations, as a seed for the transcript updates in PASA. This might compensate for many of the weaknesses of probabilistic gene predictors, including their inability to accurately predict alternatively spliced genes and aberrant genes, using only compute time instead of human verification. In this thesis, we show that this is a effective solution.

With the aid of full-length gene sequences, however, PASA's job is made much easier, and the contribution of the gene predictor comes into question. In this thesis, we further address the question, can the system work with EST sequences alone, relying on the gene predictor to "fill in the gaps" that ESTs leave?

1.6 Goals

The goal of this work is to find a method for producing annotations similar in quality to hand-generated ones as cheaply as possible and in a streamlined fashion. In doing so, we hope to bring the computational genomics community closer to the sequencing community to work in concert to simplify and accelerate genome annotation.

More specifically, we aim to

- create a trainable, adaptable system,
- make the system accessible to computer scientists and biologists alike,
- and minimize the sequencing requirements for new genome annotation
- while producing a an annotation of close-to-identical quality to hand annotations.

Along the way, we explore some novel ways of modeling alternative splicing.

1.7 Major contributions

In this work we demonstrate that the use of gene predictions combined with EST alignment assemblies generated by PASA reproduce a vast majority of hand-annotated gene sets. Additionally, we propose two new methods for predicting alternative splices with a probabilistic model. We demonstrate that on their own, they are more sensitive than older methods in predicting genes in the *D.melanogaster* genome.

We set ourselves apart from most of the current methods by emphasizing high-throughput annotations with that require little human interaction to produce.

Chapter 2

Background

Genetic information has been a popular topic of discussion in the last 20 years. The advent of high-throughput sequencing technologies and the many cultural and life changes that harnessing biological information can imply has struck a chord with society. Some have stated that understanding it will prove more significant and important than computer technology advances will in the next few decades. The pedestrian lexicon for genetics is often limited to the anecdotal, but sometimes proves more accurate than expected. To this day, although in places technically inaccurate, Mendelian genetics still holds water, even with our vastly larger knowledge base.

Here we explain some of the major biological and computational concepts involved in this work.

2.1 DNA

The cells of eukaryotic organisms by definition contain a nucleus. Genetic information is stored there in the form of DNA, consisting of a chain of Nucleoside molecules. Each of these molecules has one of four[1] bases attached. These chains are divided into chromosomes, which are further divided into arms. The strands themselves are very long (each cell of the human genome has no less than 2 meters of DNA when stretched out). To compact these long strands, the cell has complex of proteins which surround the DNA to package it into a smaller space, called chromatin [2].

The sequence of these DNA bases is a major mechanism for the transfer of hereditary genetic information. Among its more important functions is the code to produce

[1]Non-canonical bases exist, but for our purposes, are too rare to consider.

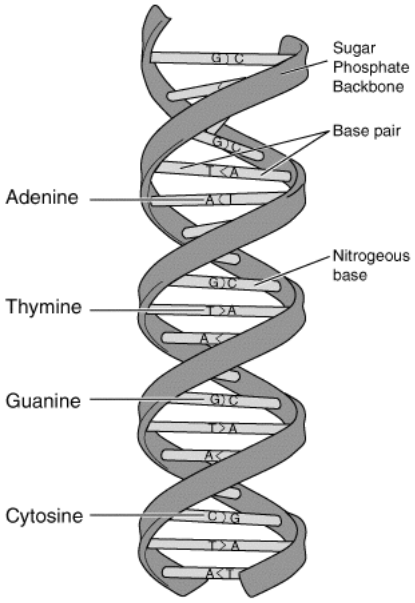

Figure 2.1: The DNA double-helix structure.

proteins in the cell, including signals that govern where, when and to what degree protein synthesis should occur.

The DNA molecule takes the form of a doubly helix. The pairs of bases on both strands are referred to as "base pairs" and as a rule[2] each of the four bases will pair to their complementary base: Adenine to Thiamine, Guanine to Cytosine, and vice versa. Pertinent information can be carried on either strand.

For the purposes of bookkeeping, one of the two strands in DNA databases is stored, and referred to as the "forward", "positive" or "plus" strand. The "reverse", "negative" or "minus" strand is implied by the forward strand sequence, and can be derived by taking the "reverse complement" of the forward strand sequence: reverse the order of the bases, and translate every base into its complementary base.

The reverse complement is useful because, all reactions occur in one direction, which is governed by the shape of the nucleoside molecule. It consists of a phosphate and a sugar, along with its information-encoding base. The sugar takes the form of pentose, a 5-carbon sugar which forms a ring. Each point on the ring is labeled with a number, from 1' to 5' (pronounced one-prime to five-prime). On one end is the 5' part of the

[2]As usual, exceptions exist, but none of interest to this work.

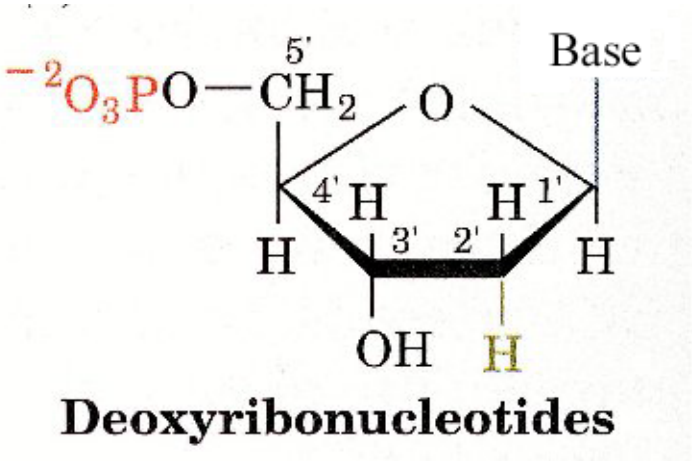

Figure 2.2: The sugar-phosphate portion of the nucleoside molecule. The 5-carbon ring determine the directionality of DNA. The 5' end can be seen as the "left" end, and the 3' end is the "right" end. The base forms a carbon-nitrogen bond to the 1' end of the carbon ring. The phosphate attached to the 5' and 3' end serves the purpose, in the case of DNA, of linking the molecules together.

ring, and at the other end is the 3' part of the ring. Enzymes (protein molecules) and ribosomes (RNA molecules) use the structure of this sugar-phosphate complex to determine the direction in which to interpret the DNA (illustrated in figure 2.2).

Important to note is that while the double strandedness of DNA provides two different directions to read from, (almost) all reactions happen from the 5' to the 3' end of the molecule. Therefore we use the terminology "upstream" for a base that is attached via the 5' end of the DNA sugar (and read before the current base), and downstream if attached via the 3' end.

2.2 RNA

RNA is almost identical to DNA except that it contains an additional oxygen atom at the 2' side of the sugar. (RNA stands for ribonucleic acid and DNA stands for *deoxy*ribonucleic acid.) This makes it a less stable molecule, and thus susceptible to degradation. All functions of RNA are still not known, but it is more recently theorized that before DNA and protein, RNA served as the primary catalyst for cellular genetic information and reactions [14].

In the modern DNA world, the most visible function of RNA is in the production of proteins from the code of DNA. RNA serves as a "go-between", carrying information from genomic DNA out to be translated into proteins. This type of RNA is referred to as messenger RNA, and is of primary interest to this work.

2.3 DNA Transcription and Translation

One major key to understanding the inner workings of the cell is the further understanding of expressed proteins in the cell. The canonical interpretation of this process is quite simple and seemingly elegant:

1. Some signal is sent to the cell, perhaps an environmental cue such as a nutrient.

2. A transcription factor is activated, which binds to DNA near the site of transcription.

3. The DNA is transcribed in a reaction to form another molecule called the premature messenger RNA.

4. The messenger RNA (mRNA) is edited in parts where unused sequence is present to form a concise mature mRNA molecule.

5. This mature mRNA is then translated into an amino acid chain, ultimately producing a protein.

This is often referred to as the "central dogma" of biology (illustrated in figure 2.3). The fallout of this is a simple model for gene structure (illustrated in figure 2.4).

DNA translation to protein occurs in a three-base degenerate code for 20 possible amino acids. These three-base sequences are called codons. The locations of codon positions in an RNA sequence are said to make up the "reading frame" of DNA. Translation occurs with the Methionine codon, often called the start codon. Translation typically stops at one of three possible stop codons, TAA, TGA, or TAG, and are typically not translated. A full translation from start codon to stop codon is called the "open reading frame".

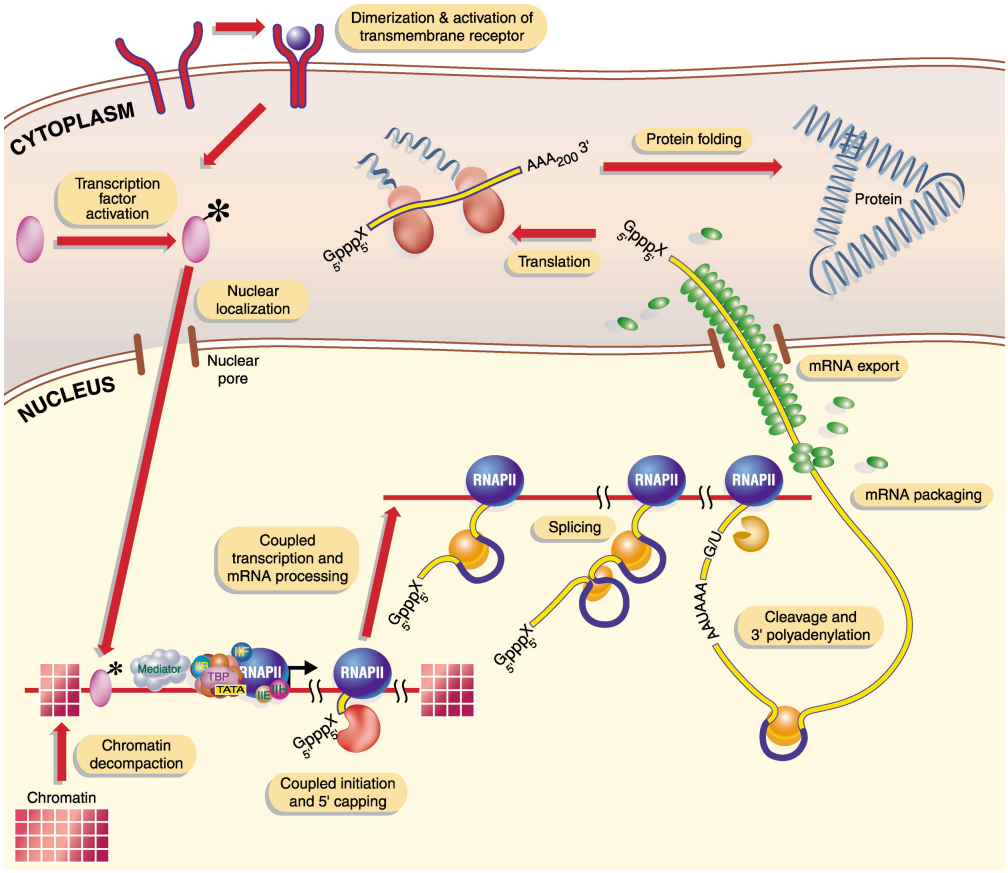

Figure 2.3: A high-level view of the genetic signal pathway of protein production, referred to as the "central dogma" of molecular biology.

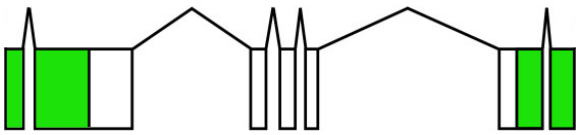

Figure 2.4: A typical illustration of a spliced gene. White portions are exons, angled lines are introns. Upstream of the start of translation is the 5' untranslated region (UTR, green), which may consist of more than one exon. Downstream of the stop codon is the 3' UTR (also green), which also may consist of more than one exon.

2.3.1 Splicing

Processing of mRNA causes some of the sequence to be spliced out, as in step 4. These sequences which are not present in the mature mRNA are referred to as introns, and the segments between are called exons. A splice site may be a donor (on the 5' end of the intron) or an acceptor (3'). Splicing is an important process, since it governs the definition of exons, the expressed sequence.

2.4 Alternative splicing events

A gene is often (loosely) defined by the part of the DNA which is transcribed to premature mRNA. This area is called its "locus" (plural: loci). Although a gene may consistently come from the transcription of the same section of DNA, the resulting mature RNA might not always be the same. The reason for this is that splicing does not consistently occur at the same locations, giving rise to much of the transcript diversity in higher eukaryotic organisms.

Alternative splicing (AS) is the cause of many different phenomena in the cell, including alternative protein products, alternative promoter regions and nonsense mediated decay, the case of an in frame stop codon signaling the cell to destroy the transcript.

While in theory, these might lead us to think of these alternate transcripts as separate genes, they are referred to as alternate "isoforms" of the same gene.

Although not everything is understood about alternative splicing, one of its major functions appears to be transcript and protein product regulation, making it a significant mechanism for cell dynamics. Alternative splicing is potentially brought about by the reduced or increased presence of proteins which promote or suppress splicing.

Several forms of alternative splicing manifest themselves in mRNA processing. Some of these are illustrated in figure 2.5. These include (in the order pictured):

1. exon skipping events, where an exon is skipped in one transcript and included in another (these are sometimes referred to as "cassette" exons),

2. mutually exclusive exons, where two or more exons are only present in the absence of the other exons in any given isoform,

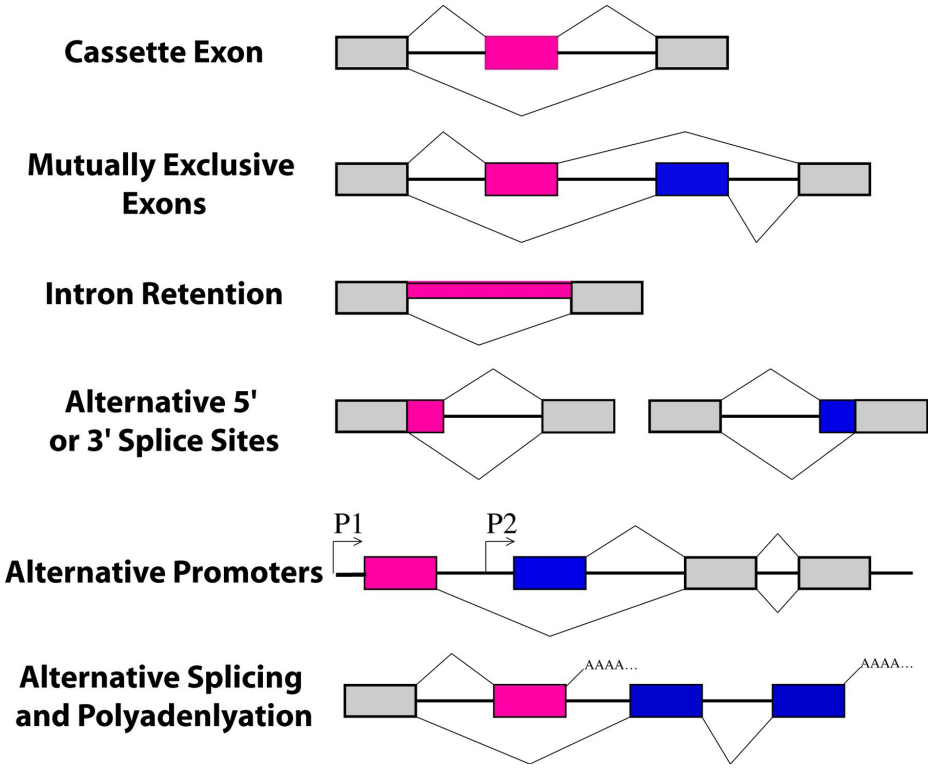

Figure 2.5: Categories of alternative splicing events.

3. intron retention, where an intron is at times not spliced out,

4. exon extensions, where an alternate donor or acceptor site is present which changes the length of a given exon,

5. alternative promoters, where an exon's altered location changes the site of translation promotion and

6. alternative poly-adenylation sites, where an exon's altered location changes the site of poly-adenylation (the chaining of adenosine molecules to the 3' end of a messenger RNA).

2.5 Methods for Predicting Alternative Splices

Several different approaches have been taken to the prediction of multiple splice forms. Some methods favor a direct modeling of alternative splicing. These have mostly been explored on the exon-level.

2.5.1 Sequence-based methods

Alternative splicing prediction dates back at least to Genscan in 1997 [5], which predicts "suboptimal exons"; exons which did not score as well as the predicted exons and are, thus, also potential exons. Cawley and Pachter more carefully address this potential method [6], however the accuracy of this method and its potential high-throughput genome annotator is not discussed in either publication.[3]

Another recent work uses a hidden Markov model designed to predict exons which might be alternatively spliced [3]. This method predicts a single exon at a time. This represents a step further in alternative splicing prediction. We present a generalization of this method to full-length transcripts.

More recently, discriminative models such as Support Vector Machines have been used to predict optionally included coding regions [8] [30]. This method is trainable and has high recall and precision. Combined with a *de novo* gene predictor, this could represent a strong method. As of now this is limited to certain types of alternative splicing, namely cassette exons, exons which are included in one transcript and skipped in another.

2.5.2 Evidence-based methods

Ensembl is a well-known genome annotation pipeline which annotates novel genomes, mostly by Protein, EST and mRNA alignment coupled with correction heuristics to the resulting annotation [7]. This method is highly sensitive, but unlike ours, relies on lots of high-cost data in order to produce its good results.

[3] We have carried out informal experiments in our reimplementation of Genscan. This showed poor results.

UNCOVER aligns homologous human and mouse introns with a pair hidden Markov model to detect skipped exons, alternate splice sites and retained introns [26]. This takes advantage of the conserved nature of alternative splicing events. This is limited, however, in that the alternative splicing event must be conserved in order to be detected, and the user must select introns which are likely to contain an alternate exon or exon extension.

LOCUS was introduced this year, and it uses a dynamic programming algorithm to combine the information of mRNA lengths and sequence evidence to predict multiple splice forms of a single gene [1]. The method relies on the use of low-cost RT-PCR reactions to determine the lengths of several transcribed mRNAs.

ÈuGene-M predicts exon boundaries in places in the putative transcript where EST evidence is incompatible with the prediction [12]. This works similarly to a cruder approach presented in this work. We hope to provide a more complete analysis of how this method performs.

These last two methods most closely resemble the methods presented here, in that we use the external evidence of ESTs to guide us into predicting genes, but do not limit ourselves to EST evidence. Here we hope to take things a step further by examining how cheaply we can perform automated annotation, and additionally use evidence directly from EST alignments to update our predictions, as a post-processing step.

Chapter 3

Gene Prediction

This work uses two well-known methods for predicting genes: hidden Markov Model-based gene finding and EST alignment. The focus of the work is the former of the two, since no novel methods in the latter are presented. Our approach to gene modeling and prediction is based heavily on the Genscan model, an HMM-based *ab initio* gene predictor. N-SCAN is an extension of the Genscan model which models patterns in multiple alignments as evidence for transcription. N-SCAN_EST uses both conservation evidence from multiple alignments and EST alignments from the target genome to predict genes.

This chapter will cover a brief introduction to gene finding algorithms and give a working knowledge of the problem in order to understand the methods presented in this thesis.

3.1 Markov Chains

A Markov chain is a discrete-time stochastic process following the Markov property, that an event at time $t + 1$ is conditional on only the current event at time t. Its purpose is to approximate the conditional probability of an observation given all previous observations, but assumes only the current observation is dependent, i.e.

$$\Pr(O_{t+1}|O_0..O_t) = \Pr(O_{t+1}|O_t). \tag{3.1}$$

An example of this might be weather prediction. One could assert that if on one day the temperature was below freezing, the next day would be highly likely see temperatures between 25 and 40 degrees Fahrenheit.

This model breaks down during the transitional periods of the year, such as spring or fall. In order to compensate for this, a higher-order Markov chain, one which conditions on more than the current observation, is applied. A Markov chain weather predictor which conditions on the current day and the previous four days to predict tomorrow's weather, a 5^{th} order Markov chain, will probably serve as a better predictor than our original 1^{st} order Markov chain in the spring.

3.2 HMMs and the Viterbi Algorithm

The hidden Markov model was introduced in the late 1960s in a series of technical reports by Lawrence Baum. The reports were highly esoteric, and most researchers outside of the field were unaware of their presence [28] until the mid-to-late 1980s, when Lawrence Rabiner published an accessible review on the applications of hidden Markov models to speech recognition.

The goal of a hidden Markov model is to find a correct labeling of an input sequence. The input sequence is the "uncovered" part of the model, and the labeling, which is not apparent upon input, is the "hidden" part of the model. Hidden Markov models can be represented as a probabilistic finite automata, and are thus constrained to have a finite state set, as well as having presupposed labels. The model is a Markov model because, like Markov chains, they follow the Markov property (see equation 3.1) that the probability of transitioning from the current state to the next is conditionally independent of all previous states at previous timepoints.

A hidden Markov model falls under the category of generative probabilistic models. The model is purported to have generated the input sequence. Thus, the result of decoding a hidden Markov model is to find the most likely sequence of states in the model which generated the input sequence.

The canonical example of a hidden Markov model is the fair/unfair coin example. Supposing you have a friend who uses a fair coin 90% of the time and an unfair coin 10% of the time, can you detect the times when he is using the unfair coin? The model is illustrated in figure 3.1. The events occur over a period of time, and the state and output of the model is dependent on the time an event occurs and the context under which it occurs.

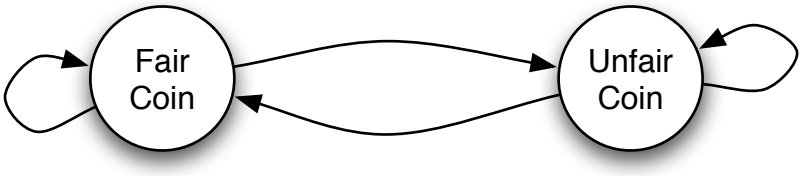

Figure 3.1: The fair/unfair coin example of a hidden Markov model.

The parameterization of a hidden Markov model is as follows:

- An output alphabet α. This is the observable part of the model. In this case it is an heads and tails flip of the coin.
- A set of states S. This is the hidden part of the model, describing the underlying labeling. These generally carry some real-world significance. In this case it is the times when your friend is using the fair coin and when your friend is using the unfair coin.
- Prior probabilities on these states π_i. These probabilities are often referred to as the "initial" probabilities, because they are used when deciding which state the system began in, however, they represent the likelihood of being in either given state at any time.
- Emission probabilities on these states e_{ic}. These probabilities represent the conditional likelihood of emitting a letter c given that the state is in state i. In our case, tails are never emitted from the unfair coin state, and heads and tails are equiprobable in the fair coin state.
- Transition probabilities between these states τ_{ij}. These are conditional probabilities of transitioning to a state given that the system is in state i. Note this follows the Markov property for stochastic processes: the probability of an event occurring at time t is dependent only on the event immediately preceding it. This is the key assumption which makes finding an optimal path computationally tractable.

Given training data, the parameters can be estimated in closed-form, using maximum likelihood.

One major shortcoming of the above model, an "ordinary" hidden Markov model, is that the likelihood of staying in any state for a length of time is distributed geometrically. This assumption assigns the most likely length of stay (all inputs being equal) to be the shortest possible one. In the real world, this is not often the case. A generalization of this model, called the hidden semi-Markov model or generalized hidden Markov model (gHMM) attempts to compensate for this by allowing the likelihood of duration of stay in a state to be any arbitrary probability density function. This provides the additional duration parameters:

- d_{il} where i is the state and l is the duration of stay in the state.

A gHMM still maintains the property that the probability of transitioning to a different state is only dependent on the current state.

Decoding the most likely state sequence is done with the Viterbi algorithm, the one most commonly used for gene prediction. This is a dynamic programming algorithm based around the Open Shortest Path First (OSPF) algorithm. The nodes of the graph represent time points in particular states in the system, i.e. at position i in the sequence, the probability of being in state j is represented by a node at v_{ij}. These nodes are connected only to states at the next time point, forming a topologically sorted graph. The goal of the Viterbi algorithm is to find the most probable path, or the path with the least cost, as is defined by OSPF. Thus the algorithm can be computed in $O(E)$ time.

In the ordinary case, edges connect only from the current node to the node in the time point immediately in ahead of it. Therefore E is equal to S^2L, where S is the number of states and L is the length of the input sequence.

With generalized hidden Markov models, all assumptions about the length of stay in a state are lifted, so gHMM has links connecting the current time point to all future time points in the current state. This gives the graph S^2L^2 edges.

It is interesting to note, however, that since optimal lengths of stay are computed at all possible lengths, that it becomes possible to integrate more complicated models into the probability of a duration in a state. In the case of gene finding, it is particularly

useful to know where the beginning and end of a putative exon lie, so that the probability of a particular donor-acceptor site pair can be computed along with the coding content probabilities of the exon.

Like OSPF, each node has a cost associated with it (inversely, the probability) of the shortest path up to that node. This can be computed recursively. Given a sequence G the computation for a gHMM is defined as follows:

$$v_{1j} = \pi_j e_{jG_1} \tag{3.2}$$

$$v_{ij} = \max_k[\tau_{kj} \max_l[d_{jl}\Pi_{n=1}^{i} e_{kG_{i-n}}]] \tag{3.3}$$

Each v_{ij} value represents the most probable path passing through state j at the position in the input sequence G. Traceback, therefore, begins at the end of the sequence l, with $\arg\max_j v_{lj}$.

Note that for most practical purposes, the value for $\Pi_{n=1}^{i} e_{kG_{i-n}}$ can be computed with multiple models. If using an acceptor sequence model, for example, you might compute the first 3 bases of the exon as being the probability of the acceptor model given the sequence.

3.3 Genscan

By the mid 1990s, many hidden Markov model gene predictors had been developed, but none had yet tackled the problem of recovering full length transcripts from human DNA. Christopher Burge designed and parameterized a semi-hidden Markov model for predicting human genes, which have complex coding patterns, deceptive splice sites, among other things. The Genscan hidden Markov model is illustrated in 3.2.

The design of the Genscan HMM was met with many engineering challenges, due in no small part to the length of DNA databases. While an algorithmic time complexity of $O(S^2L^2)$ is usually considered fast, DNA sequences are quite long, and speedups needed to be implemented.

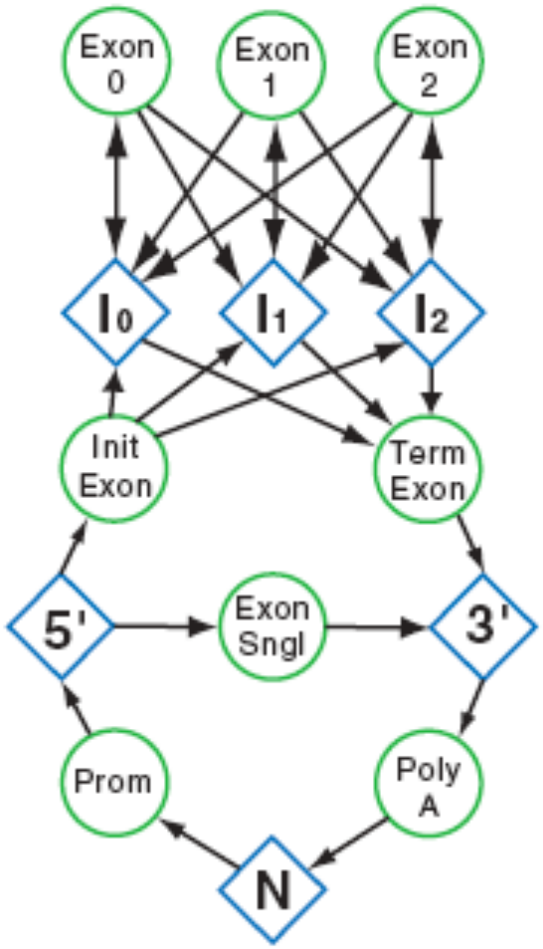

Figure 3.2: The Genscan hidden Markov model for gene prediction. This model also includes (unpictured) states for minus strand features. The durations circle-shaped states are modeled with explicit distributions, and the diamond states are modeled with geometric length distributions. The 5' and 3' states represent UTRs.

3.3.1 Duration distributions

In the general case, no restrictions are placed on the length function of a stay in any given state in an HMM. Naïvely, at any given position in the sequence i, the optimal length of stay in that state would need to be computed for all lengths up to i at each time point. Genscan takes advantage of the properties of exons and introns which make practical constraints on the length of stay:

- The probability of an exon length is modeled empirically, up to a fixed length. Most exons are not longer than 6000 bases, so a low, fixed probability is applied to all exons beyond that length[1]
- The probability of an intron length is modeled with a geometric distribution. Human introns roughly follow a geometric length distribution, with the notable exception of short introns. Using a geometric distribution cuts all edges in the graph up before the current time point, speeding calculation up.
- Promoters and PolyA tails are assumed to have a short range of possible lengths. This reduces the length computation to a small constant.

When no states are allowed to stay for any length beyond a certain constant, the computation algorithm is reduced again to $O(S^2L)$, since only a constant number of previous links are possible.

3.3.2 Content models

Since the input alphabet is only four letters, an emission model which simply reflects the probabilities of a letter in that state would be seriously underfit to the model. That is, the differences between the intron distribution and the coding exon distribution would be quite different. To compensate for this, Markov chains are used to model content.

[1]While mixing empirical distributions with other length distributions is normal, it is technically incorrect to apply a fixed probability beyond a certain length. In practical terms, however, it is very unlikely to find an exon in a given reading frame with no in-frame stop codon preventing it from being a usable exon, and further that with the low probability, this exon will be chosen. Therefore this fact is "swept under the rug". Estimating a geometric tail beyond this empirical distribution is also problematic given the limited data.

In the case of introns and intergenic sequence, the sequence is modeled with a 5^{th}-order Markov chain. Coding sequence is modeled with a special 3-periodic 5^{th}-order Markov chain. This means that each of the three codon positions are modeled with a separate Markov chain. Genscan borrows this model from a study which did an ad nauseum search for the best discriminator of protein-coding and non-coding sequence [10].

3.3.3 Signal models

Another speed up and source of accuracy is the addition of Weight Matrix Models and their generalizations, Weight Array Models (WAMs), to the exon emission model. WMMs assign a odds-ratio to each position in the sequence for a given, fixed window. These are especially useful for signal models, where, for example, an "AG" consensus sequence is expected at the exon-intron boundary. This allows all potential exons without such a consensus to be ignored, since it will have zero probability.

The odds-ratio is a common idiom in models for biological sequences. Each parameter in the positive model (the model for the correct predictions) is paired with a parameter in the negative or "null" model, as in

$$S(o) = \log \frac{\Pr(o|+)}{\Pr(o|-)} \tag{3.4}$$

For practical purposes, the scores are converted to logs and added together rather than multiplied.

In the case of signal models, Burge modeled the positive model after all donors, acceptors, start and stop codons for each model, and the negative model were the respective "pseudosites". A pseudosite is any instance in non-coding sequence where a consensus sequence occurs (for example, a donor would be all "GT"s in introns). The model will assign a score greater than 0 to all sites that stand out from the pseudosites.

Weight Array Models (WAMs) can be viewed as a generalization of WMMs. At each position of the WMM, parameters are estimated for a base in a specific position with no context, making it a 0^{th} order Markov chain. A WAM increases the order at each

Pos	-3	-2	-1	0	1	2	3	4	5
A	-2	12	-16	-∞	-∞	17	13	4	-3
C	1	-9	-24	-∞	-∞	-32	-10	-7	-5
G	-8	-11	23	22	-∞	-8	-11	-1	-1
T	-43	-12	-28	-∞	18	-28	-14	-1	4

Figure 3.3: An example Weight Matrix model for a donor site. Positions are relative to the exon boundary on the forward strand. Positive numbers indicate that the event occurs more commonly in the positive model (true donor sites) than the negative model (pseudosites). A score is assigned to each position, and being log-odds ratios, they are added rather than multiplied. For example, a potential donor site of CAAGTGAAG would receive a score of 45. A negative score does not preclude a potential donor from being considered. The reject threshold is a heuristically chosen constant. The score of the "GT" consensus sequence in the negative model are estimated from the composition bias of the training sequence.

position of the WMM, effectively making the parameterization a string of Markov chains, estimated for specific positions in a signal site.

3.3.4 Model inflexibility

If we return back to the formalism of the gHMM , emission probabilities for a feature are defined as

$$\Pi_{n=1}^{i} e_{kG_{i-n}} \tag{3.5}$$

The likelihood of any given feature will be very low, since several small numbers will be multiplied together. Computers have a limited capacity for storing very small numbers, so this particular calculation is not made. Instead, a log-odds ratio is used [2]:

$$\log \Pi_{n=1}^{i} \frac{e_{kG_{i-n}}}{e_{\mathrm{NULL}G_{i-n}}} \tag{3.6}$$

[2]The scores are not computed as the log of the ending product of the probabilities, but instead as the sum of the log-odds probabilities, which is equivalent

Since the log function is monotonically increasing, the ordering of probabilities under the positive model will be the same. The null model is intron and intergenic sequence. Since the null model is scored the same under the sequence in all states, the denominator can be treated as a constant.

However, with a few observations, we see that this is not the case in the Genscan model. The null model is scored differently in our example above with the donor site, as a 0-th order position-specific probability. Additionally, the model has been empirically determined to perform at a near maximum when the null model is estimated from the first 1000 bases of the first intron after the initial exon.[3]

Several attempts at extending the Genscan model in significant ways have consistently shown to be detrimental to the performance of this model. While several alternatives to using the Genscan model exist, we have not found one that performs better at modeling gene structure from an input DNA sequence.

3.4 Twinscan and additional sequences

Here we briefly discuss Twinscan, the precursor to N-SCAN. Twinscan augmented the Genscan model to incorporate conservation information, improving predictive accuracy [24]. Here two genomes were used as evidence for transcriptions in the target genome, the second being called the "informant" genome. This genome is assumed to be evolutionarily related but distant enough so that non-coding sequence tends not to be conserved. In the original paper, human was the target species and mouse was the informant species.

For this project, the conservation sequence was invented, a sequence of 3 characters indicating whether the target genome is matched, mismatched, or unaligned to the informant genome with WU-BLASTN [15]. This sequence was input in addition to the target DNA sequence and modeled in a similar manner as described above.

Formally, this can be seen as adding several new characters to the input alphabet, i.e. "an A which matched informant sequence", "an A which mismatched the informant sequence", "an A which is unaligned to the target sequence" and so on. However, these

[3]This was reverse engineered for Twinscan and N-SCAN, and is not necessarily part of the original Genscan model.

two sequences are assumed to be uncorrelated and thus probabilistically independent. The probability of emission can thus be calculated as

$$\Pr(O_t) = \Pr(N_t) \Pr(C_t) \tag{3.7}$$

where O_t is the combined observation of the DNA letter and conservation letter at position t, N_t is the DNA letter at position t and C_t is the conservation sequence letter at position t.

This paradigm of adding probabilistically independent data via a parallel sequence has manifested in many forms, including the Twinscan_EST, N-SCAN and N-SCAN_EST variants.

3.5 N-SCAN and UTR prediction

N-SCAN is a phylogenetic HMM founded on Genscan's models for gene prediction, which aims to generalize Twinscan's conservation model into any number of informant species. N-SCAN uses Bayesian networks to model conservation levels in the features it predicts. Incorporating conservation information across species makes N-SCAN improve on the accuracy of Genscan [16].

One of the key elements of N-SCAN's predictive accuracy is its probability factorization. By rooting the Bayesian network on the target sequence, N-SCAN is able to compute the probabilities of the target sequence model independently of the conservation model, using the Genscan model. As a result, the conservation probabilities are computed independently and multiplied together, as in equation 3.7.

N-SCAN also introduced the modeling of 5' UTR exons. This is illustrated in figure 3.4. N-SCAN accurately predicts 5' UTRs and this might be reason for higher start codon prediction performance [4].

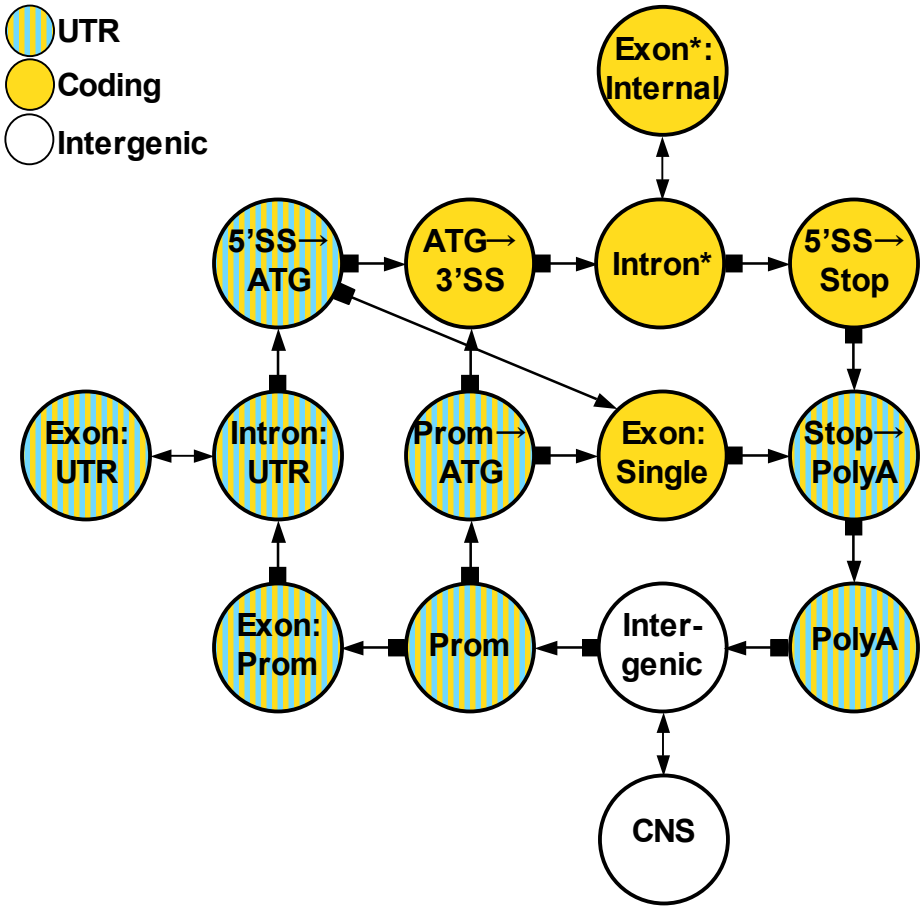

Figure 3.4: The N-SCAN hidden Markov model. All states labeled with a * track the reading frame of the state. The four states on the left represent the new UTR-predicting states.

1. Align all ESTs to the target sequence.
2. For each alignment,
   - label all aligned regions as exons and
   - label all unaligned regions between exons as introns.
3. At every position in the target sequence,
   - if no alignment exists, mark the position with an `N`
   - otherwise, if two alignments exist, one with an putative exon and one with a putative intron, mark the position with an `N`
   - otherwise, if any exon alignment exists, mark the position with an `E`
   - otherwise, mark the position with an `I` (there must be an intron).

Figure 3.5: The ESTSEQ algorithm.

3.6 N-SCAN_EST

N-SCAN_EST takes prediction another step further to model EST alignment presence in the target genome [31]. In order to fit this into a hidden Markov model context, another sequence is generated in parallel to the target DNA sequence. The sequence consists of three characters, 'E' representing an exon aligned at that location, 'I' representing an intron aligned and 'N' representing intergenic or unknown (including cases where two alignments overlap, one containing an exon and another containing an intron). This sequences is again assumed probabilistically independent of the target sequence, and can be computed as in equation 3.7.

The algorithm for computing this sequence is illustrated in figure 4.8 and described in figure 3.5.

The intention of using an "N" in the regions where an exon and intron are present in two different alignments is to avoid bias of bad alignments at intron-exon boundaries in predicting splice sites.

Chapter 4

Methods

Three methods were used to predict alternatively spliced genes in *D.melanogaster* genome, UCSC version 2 [9]. All of these methods are based around the N-SCAN_EST model and improvements thereof.

4.1 Methods overview

Three extended pipelines ("superpipelines") are compared in this thesis: N-SCAN_EST, N-SCAN_MP_EST and N-SCAN_AS_EST . These pipelines have five major steps. They are alignment, sequence generation, parameter estimation, gene prediction and alignment post-processing. This is illustrated in figure 4.1.

The N-SCAN_EST superpipeline recycles a current method [31]. We present this method, which only predicts a single splice form per locus, as a performance base line to the novel methods.

The N-SCAN_MP_EST "multi-pass" method updates N-SCAN_EST by using multiple, conflicting ESTs in separate runs on the same target sequence. The general idea is to encourage the N-SCAN predictor to predict different transcripts with each successive pass.

Finally the N-SCAN_AS_EST method works in a single pass, but has an HMM capable of predicting alternately spliced genes. The N-SCAN_AS_EST method also uses an updated EST sequence, one which indicates whether alternative splicing was indicated by the EST alignments.

Several of these steps involve the PASA pipeline, a suite of tools designed to intelligently align full-length cDNA and EST sequences to a genome and cluster them into

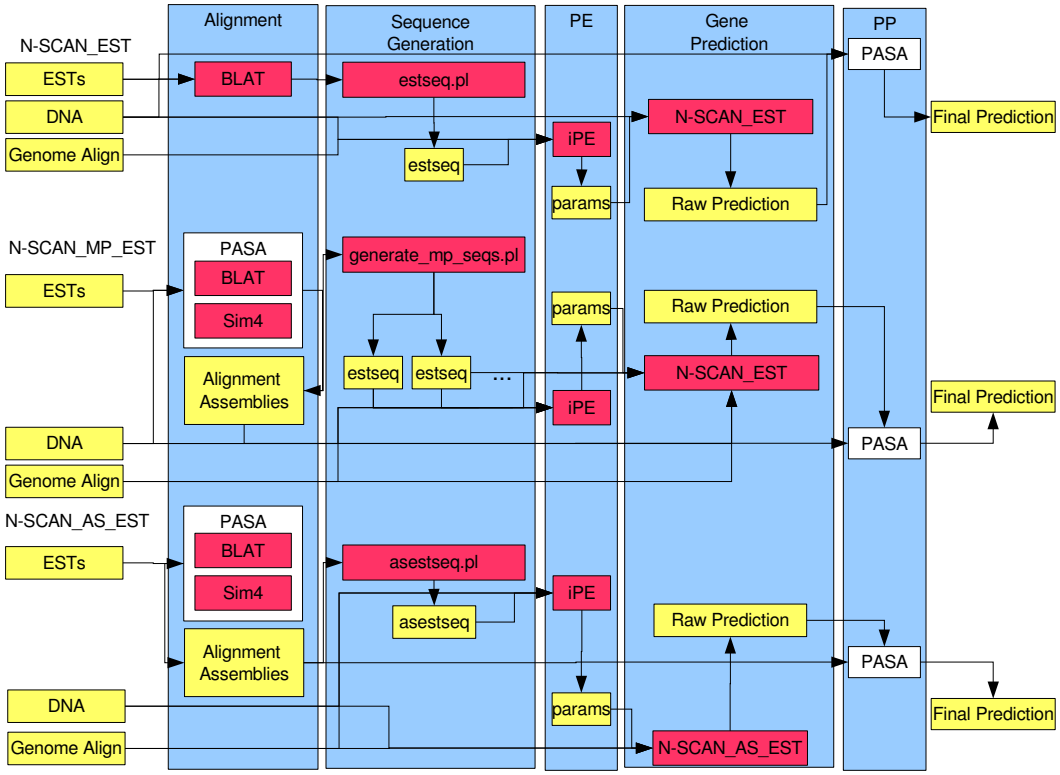

Figure 4.1: Here the data flow for each method is illustrated. PE stands for parameter estimation and PP stands for post processing. Red boxes are standalone programs or scripts, yellow boxes are data and white boxes are pipelines of several programs and/or scripts (namely, PASA).

"alignment assemblies" [17]. This is a heuristic EST alignment processing suite which carefully checks the validity of alignments and attempts to realign the sequence if the current alignment is insufficient.

The PASA pipeline will also update an annotation based on these alignment assemblies. After PASA has built the database of alignment assemblies, the program can accept annotations, compare them to the alignment assemblies, and intelligently "merge" them together in an annotation update. This is done in all post processing steps of the superpipelines.

The goal of these experiments is to evaluate the probabilistic gene prediction phases of N-SCAN_EST and friends in the context of the PASA annotation updater.

4.2 Parameter estimation

Parameters were trained from the Reference Sequence annotations [27] as downloaded from the UCSC genome browser [22]. For simplicity, only chromosomes 2L, 2R, 3L, 3R, 4 and X were used. The annotations were filtered for incorrect stop- or start-codons, inframe stop codons, non-canonical donor sites (GT, GC or AT) and non-canonical acceptor sites (AG or AC) using updated parts of the Eval software package [20].

All overlapping sequence features were given a count-weight proportional to the number of distinct features at each position. Null models for DNA sequence were estimated from the first 1,000 intronic bases downstream of the start codon, and this model was tied to both intergenic and intronic regions for decoding. All other DNA parameters were trained as described in [5].

Conservation parameters were trained with BLASTZ [29] alignments between *D.melanogaster* UCSC version 2 and *D.ananassae* UCSC version 2. The methods used here are identical to those described in [16]. We chose to use only a two-way alignment because the results are close to those with a 3-way alignment, and because we were comparing relative performance of the different EST methods, not conservation.

EST parameters for N-SCAN_EST were trained from BLAT alignments of ESTs to the *D.melanogaster* UCSC version 2 genome. These were downloaded from the UCSC browser EST track of the *D.melanogaster* genome (a total of 255,654 sequences).

BLAT was run using the same parameters as described in the methods section of [31].

All other variants were trained from EST assemblies generated by the PASA pipeline as detailed below. In addition to the ESTs, full length mRNA tracks were downloaded (a total of 21,069 seqeneces).

4.3 EST processing with the PASA pipeline

Gene predictions were modified and improved using the PASA (Program to Assemble Spliced Alignments) pipeline. A large schematic of the intended use of the PASA pipeline is shown in figure 4.2. This program was originally developed to improve the annotations of the *A.thaliana* genome by (in a broad sense) merging EST evidence with current gene predictions and hand-curated annotations to produce a more complete annotation. The process is completely automatic. Some modifications were made to the pipeline, which will be discussed later.

We use this pipeline here as an example of how EST alignments can be automatically incorporated with N-SCAN_EST gene predictions and used in training N-SCAN_EST to accurately predict the majority of genes. The following post processing steps were performed on all N-SCAN predictions to combine them with EST alignments with the aid of the PASA pipeline.

The PASA pipeline, as originally written, has four main phases: alignment, assembly, comparison and update.

4.3.1 Alignment

In preparation for aligning the raw ESTs to the *D.melanogaster* genome, the sequences were cleaned using the `seqclean` tool [13]. A custom vector database which contained only the vectors used for *D.melanogaster* EST sequencing was used.

All *D.melanogaster* EST and full-length cDNA sequences were downloaded from the UCSC genome browser, as described above. These sequences were then aligned to *D.melanogaster* UCSC version 2 genome.

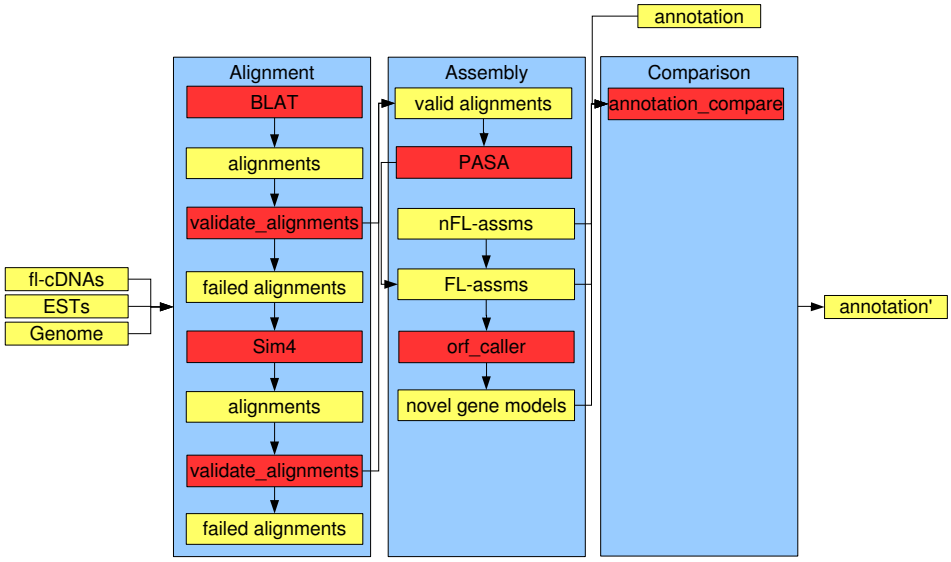

Figure 4.2: A schematic view of the PASA pipeline. Programs and data are represented the same way as in figure 4.1. The pipeline relies almost exclusively on a MySQL database schema for storage and retrieval of data, including the data presented here. The comparison and update phase is run separately from the other two phases, and can be repeated on the same database of alignment assemblies several times.

The alignments were filtered for validity after each alignment wave is performed. Alignments were required to have major splice sites: GT or GC donors and AG acceptors. Alignments are first discarded if their introns are unreasonably long. Only 36 of the 15,069 introns in the Gold Set[1] are longer than 500,000 base pairs, so we use this as the maximum intron length. The alignments are further are filtered for a minimum percent identity, i.e. a minimum matches per aligned nucleotide, and a minimum percent of the total cDNA length aligned. These are set to 95 and 90, respectively. No single-exon EST or FL-cDNA alignments were considered.

All EST and FL-cDNA sequences were aligned using the BLAT [21] and Sim4 [11] alignment programs. BLAT was used first, and if the alignment did not pass the above criteria, they were passed on to Sim4. All sequences whose Sim4 alignments did not meet these criteria were discarded.

4.3.2 Assembly

The PASA assembly algorithm is a dynamic programming algorithm designed to find optimal clusters of EST alignments. A cluster of alignments represents entirely redundant EST alignments, ones whose internal splice sites all match in coordinates. The core algorithm is designed to find "maximal transcript alignment assemblies". These are found by searching, for each alignment A_a, for the assembly with the most EST and cDNA alignments that contains A_a.

The algorithm is performed by computing the compatibility of all pairs of overlapping alignments first. For each alignment A_a, find all compatible alignments contained within the span of A_a, and record it as C_a. The assemblies are then computed in an $n \times n$ dynamic programming matrix, where n is the number of alignments. Incompatible alignments are marked as such. The order of the alignments in the matrix is ascending genomic position. The best assembly for each pair of alignments a and b, at cell (a, b), is computed twice, once in a bottom-up fashion ($(0, 0)$ to $(n-1, n-1)$) and once in a top-down fashion ($(n-1, n-1)$ to $(0, 0)$). The bottom-up values are stored in an L_{ab} matrix (since all alignments are either contained or strictly left of higher-indexed alignments), and the top-down values in R_{ab}.

[1] This is a separately produced hand-annotated set derived from the same sequences as the reference sequence project used, however requires support from ESTs and fl-cDNAs to become part of the set. This is therefore a smaller set than the reference sequence set.

At each cell (a, b), the L-value is computed as $L_{ab} = \max_{ij}[C_a, L_{ij}+C_{a\setminus i}]|i \leq a \wedge j \leq b$, where $C_{a\setminus i}$ is the number of compatible alignments contained in the span of a but not in the span of i (this is done to avoid double-counting alignments). A link to the cell from which the maximum computation originated is stored. Additionally a forward link is stored in each cell for the last maximum computation that was found.

The assemblies are recovered from the matrix by finding the maximum $R_{ab}+L_{ab}-C_a$ value of any cell, then tracing the links to the maximum paths through the matrix. Afterward, the next highest cell not contained in any alignment assembly is found, until every alignment belongs to some assembly, or there are no new assemblies. Assuming the computation of L, R, and C values is $O(1)$, the algorithm takes $O(n^3)$ time and $O(n^2)$ space where n is the number of transcripts. This algorithm is illustrated in figure 4.3, taken from [17].

The FL-cDNA containing assemblies are considered as putative confirmed or novel gene models. This means that if an assembly contains a FL-cDNA, then it may alone be considered a gene, without any evidence from the input annotation. So, with FL-cDNAs, if no hand annotation is input to the pipeline, several genes may be output. All other assemblies are considered as possible extensions for gene models or evidence for alternative splicing. This means that ESTs will only serve to extend FL-cDNA alignments and input hand annotations. Without FL-cDNAs, the pipeline requires evidence from the input annotations for the existence of a gene locus.

4.3.3 Comparison

Here, the assembled alignments are compared to the input annotation set that PASA augments. There are five types of updates: UTR extension, coding sequence extension, gene structure alteration, alternative splicing and internal exon addition. All these updates can take place with the comparison of an input annotation and an assembly with or without FL-cDNAs. Novel genes, ones that do not overlap anything in the input annotations, require FL-cDNA-containing assemblies.

As alluded to, there are two major kinds of assemblies: FL-cDNA-containing assemblies (FL-assms) and non-FL-cDNA-containing assemblies (nFL-assms). The comparison process is a series of comparisons of one PASA assembly to one input annotation. At each step, the comparison program determines if a pair is part of the same gene

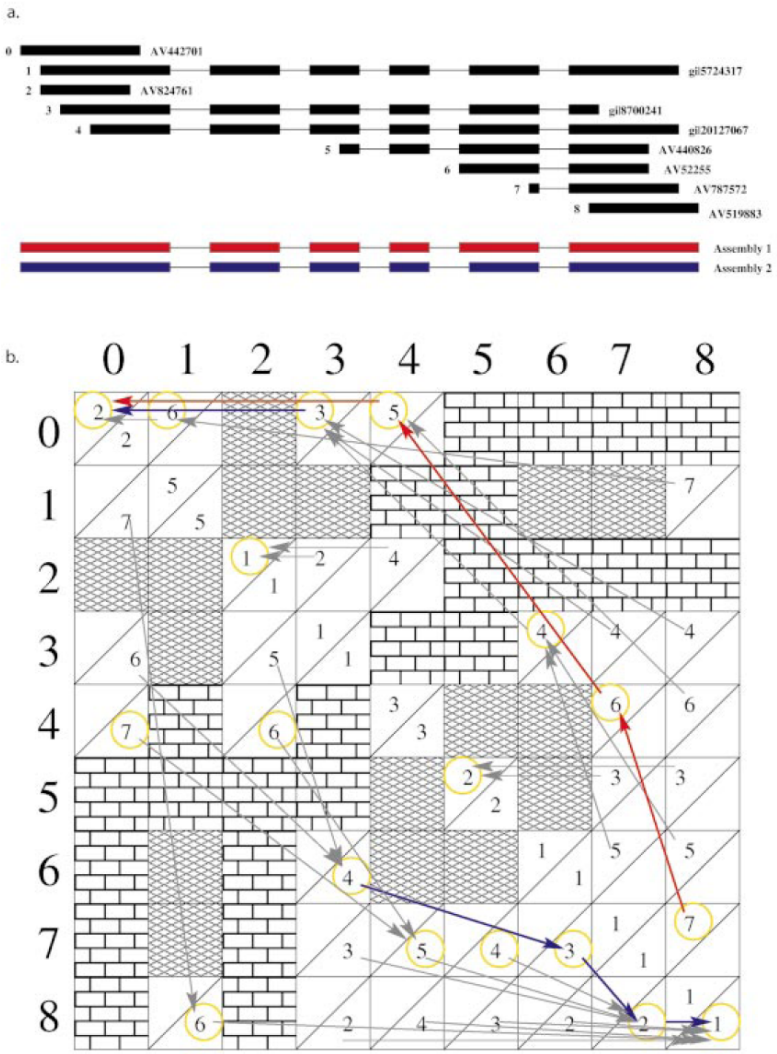

Figure 4.3: The PASA algorithm as illustrated in [17]. This algorithm goes both forward (upper left numbers) and backward (lower right).

with the use of some user modifiable heuristics. If they are determined to be a part of the same gene, then an update is flagged.

Here are the criteria for an update to be considered true:

- For any two cDNA sequences from the annotation and assembly set each, that pair is part of the same gene only if they overlap 50% of each others' neucleotides and the assembly sequence was derived from a FL-cDNA sequence.
- For any cDNA sequence in the assemblies containing a FL-cDNA, an assembly is valid only if its maximum length predicted open reading frame covers at least 40% of the total sequence.
- For any cDNA sequence from an assembly that does not contain a FL-cDNA alignment, a sequence is considered valid only if 70% of the protein coding sequence in both alignments overlap.
- For any annotation and cDNA sequence from an assembly, the pair is considered part of the same gene only if 70% of the protein coding sequences are aligned in both sequences and 70% of the aligned sequence is identical.
- For any cDNA sequence from an assembly containing a FL-cDNA, a sequence may be considered an alternate isoform of an annotated gene only if the overall length of the sequence is at least 70% of the overall length of the annotated sequence.
- For any cDNA sequence from an assembly, a sequence is considered valid only if it contains 2 or fewer UTR exons.
- For any annotation in the original set, it may be replaced by a FL-assm only if multiple annotated genes overlap it by 80% or more. The purpose of this is to prevent the situation where a few inaccurate EST alignments would alter the original annotation, and thus to give more credence to the input hand annotation set.

Assemblies are rejected as possible updates if they overlap more than one gene in the annotation, if a novel gene has some amount of overlap with another gene, if an assembly is marked as an alternate isoform of a novel gene, if the genes must be

merged but are only partially described by the assembly or if the assembly does not fit within the above 7 classifications.

4.3.4 Update

The update phase is a simple extension of the above. All valid updates are combined with the current annotations, and replaced genes are removed from the current annotations.

4.3.5 Modifications to the PASA pipeline

The expressed purpose of the PASA pipeline is to update a current genome annotation with evidence from assembled EST alignments. We used the PASA pipeline for three other purposes, described in detail below:

- **Update of gene predictions.** We use PASA to increase the number of annotated transcripts per locus. This is similar to its original use, except that the input to PASA is not a hand annotation.
- **Evidence of EST presence.** We use PASA as a more intelligent EST sequence generator by using alignment assemblies to generate EST sequence. (For a description of EST sequence see section 3.6.)
- **Evidence of Alternative Splicing.** PASA will assemble ESTs into very likely alternate splice forms by using several heuristics as described above. We use this evidence to generate a new EST sequence, the Alternative Splice EST sequence, or AS_ESTSEQ sequence.

4.4 N-SCAN_EST + PASA

PASA serves as a tool to automatically update annotations which appear to be incomplete or to lack compelling evidence for their correctness. Bad annotations could be inserted into the collection for several reasons, such as bad alignments, sequencing error, identification of a pseudogene or exclusive reliance on a gene predictor for

evidence of transcription. In our case, we take our own, automatically generated annotations which may be incomplete due to shortcomings of the model itself (e.g. atypical transcripts left unpredicted) or due to the fact that N-SCAN_EST can only predict a single isoform at a locus.

There is no difference in the usage or the code for this method. This simply differs in concept rather than implementation. Using N-SCAN_EST (or one of the variants), we seed loci for PASA to pick up additional annotations. PASA will have more confidence in using any of the EST assemblies which contain an N-SCAN_EST prediction. With only ESTs for evidence, PASA will not predict a full-length mRNA, since it has no metric with which to gage the completeness of an EST assembly. N-SCAN_EST fills that gap.

The goal of this work is to improve predictions by predicting multiple splice forms of the gene. Given that we have the PASA tool, the relative virtues of N-SCAN_EST and its variants must be compared as it relates to whether the predictions can improve the ultimate outcome of the pipeline. In essence, this serves as our "control" or baseline method.

4.5 N-SCAN_MP_EST + PASA

As described, PASA outputs many putative splice variants given EST alignments. Many of these are incomplete, and thus PASA will not rely on partial EST alignments alone to assert the existence of a full-length transcript. N-SCAN_EST has in several cases shown to be able to complete the picture of these rough sketches of transcripts.

N-SCAN_MP_EST is a slight variant on N-SCAN_EST . Rather than running N-SCAN_EST once, we run it several times, each time with different EST sequences. If an EST assembly overlaps another EST assembly, the overlapping assembly is used in a separate EST sequence for a separate N-SCAN_EST run. The idea is to "cue" N-SCAN_EST to predict different splice forms by presenting it with different EST sequences that each represent conflicting EST alignment assemblies in some places.

In order to do this, we first generate the multiple EST sequences with the following algorithm in figure 4.4. It is important to note that in this case, there are no cases of overlapping introns and exons, since there are no overlapping alignment assemblies.

1. For each EST alignment assembly,
    - label all aligned regions as exons and
    - label all unaligned regions between exons as introns.
2. Generate EST layers such that
    - no two EST assemblies overlap and
    - any EST assembly only exists on one layer
3. At every position in the target sequence, for each EST layer,
    - if an exon exists, mark it with an '`E`'
    - otherwise, if an intron exists, mark it with an '`I`'
    - otherwise, mark it with an '`N`'. (It must be intergenic.)

Figure 4.4: The algorithm for generating MS_ESTSEQs.

It is also important to note that each EST alignment assembly only associates with one EST sequence. The intention is to encourage prediction of all detected ESTs, completing a plausible ORF for the potentially partial EST. Should an antisense EST exist in the intron of another gene, this can be predicted with this method.

The layers are ordered such that earlier EST layers will contain more EST alignment assemblies, and the later EST layers will contain fewer.[2]

Training was done on the first 10% of the sequences for each chromosome. The training set was specially altered by using the PASA program to associate EST alignments which match the training data with those EST sequences in training. This was intended to simulate the prediction of the correct matching transcript underlying the EST alignment assembly, biasing the parameters toward exons matching '`E`' characters in the EST sequences.

The remaining parameters, the Genscan and N-SCAN conservation parameters, were trained along with the EST sequence parameters in the same manner as usual, except

[2]It is not the case that this is a complete ordering. The algorithm simply defaults to the first EST layer that has no overlapping EST alignment assembly. A greater EST layer could have multiple EST alignment assemblies overlapping a single EST alignment assembly in an earlier layer. In a very pathological case, the ordering could be completely reversed, but with our data, this was not so.

that the specially altered training data was used, rather than the original RefSeq annotations.

N-SCAN_MP_EST takes an order of magnitude more time to run than standard N-SCAN_EST does. One can envision a faster version, which only takes into account conflicting ESTs as they are presented, and adds a new sub-path when a change occurs. Similar, unpublished work has been done with SNPs [19], and ÈuGene-M uses a similar algorithm, but the effects are not equivalent [12].

4.6 N-SCAN_AS_EST

N-SCAN_AS_EST is yet another variant on N-SCAN_EST that attempts to resolve the speed issue created by running several N-SCAN_EST passes. Instead of running multiple passes, a single pass is run on a new hidden Markov model which includes states that indicate the presence of optional coding sequence. A new HMM is developed for this purpose, and the ESTSEQ algorithm is again reworked, with new characters included to indicate the presence of optionally transcribed regions.

4.6.1 Design and choice of additional HMM states

The large goal was to design a hidden Markov model which generates optional coding sequence. Of the major types of alternative splicing, we chose to model cassette and mutually exclusive exons and alternate 5' and 3' extensions to internal exons. Several decisions were made based on practical reasons.

Alternate splice sites

Statistics were gathered on the presence of 3' and 5' extensions to exons in the RefSeq annotation set for *D.melanogaster* UCSC version 2, shown in table 4.5. Based on the low level of multiple exon extensions present in the data set, we chose to model only one exon extension per internal exon.

	3'	5'
Total extensions	251	328
Reading frame preserving extensions	138	226
Reading frame shifting extensions	127	121
Average length of extensions	356.02	235.19
Exons with 1 extension	238	310
Exons with 2 extensions	12	15
Exons with 3 extensions	1	2
Exons with 4 extensions	0	1

Figure 4.5: Alternative splicing statistics for exon extensions on RefSeq annotations for *D.melanogaster* UCSC version 2.

All coding states in the N-SCAN gHMM have an associated frame, indicating the codon position of the last base of the feature. In order to correctly predict a frame-shifting extension to an exon, some other alternate site must compensate for the shift in frame of the alternate isoform. Since tracking frame for multiple isoforms gives rise to a combinatorial explosion of the number of gHMM states, the only feasible way to predict frame-shifting exon extensions would be to relax reading-frame constraints on the extensions.

Unfortunately, reading frame is very crucial to correct prediction of exons in spliced transcripts.[3] Given this, it is hard to envision a gHMM with no frame constraints on exon extensions correctly predicting multiple isoforms of a gene. For these reasons, we chose not to model frame-shifting exon extensions. While the number frame-shifting exon extensions are fewer than expected in random annotations, they represent a significant portion of exon extensions, leaving this method with a serious handicap.

Given the average length of the extensions, we chose to model their length with an empirical distribution up to 500 bases long. All longer extensions are modeled with a geometric distribution, however very few longer exons were found.

[3]This was verified in an experiment where frame constraints were relaxed on all spliced exons. Fewer than 18% of the spliced exons were predicted exactly correctly on chromosomes 2R and 3R of *D.melanogaster* UCSC version 2.

Optional exons

Additionally, we chose to model optional coding exons. Like exon extensions, only frame-preserving optional exons are modeled. While optional exons have been shown to exhibit a longer average length and higher variance [33], this did not affect our choice of length distribution. We used the same empirical distribution used for internal exons in the N-SCAN model. The maximum length for this distribution, 6,000, is much greater than the average length found in the RefSeq set (353.2), and thus will probably not affect the robustness of the parameterization. The distribution for optional exons was made separate from the distribution for standard internal exons.

The ASHMM

This new HMM adds three major states[4] to the current N-SCAN hidden Markov model, as illustrated in figure 4.6. The MD state represents an additional donor site off the 3' end of the exon, the MA state represents an additional acceptor site up the 5' end of the exon, and the OptExon state represents an exon which is sometimes included in the mature mRNA sequence and other times is not.

These states each represent two features, one coding and one non-coding (intronic). For each time the optimal path passes through one of the optional coding states, two transcripts are output, one including the feature and one excluding the feature. Thus the total number of transcripts output will be 2^N, where N is the number of optional coding features.

4.6.2 The use of AS_ESTSEQ

PASA helps us to sharpen the picture of which ESTs belong together as a complete transcript, and also which are alternatively spliced. With N-SCAN_MP_EST we take advantage of this indirectly, closing the predictor in on mutually exclusive EST alignment assemblies. Here we hope to more directly model these features. In order to aid this process, we make the addition of a new EST sequence that indicates the presence of alternatively spliced features.

[4]Several more states are added in addition for the sake of frame tracking and strand information.

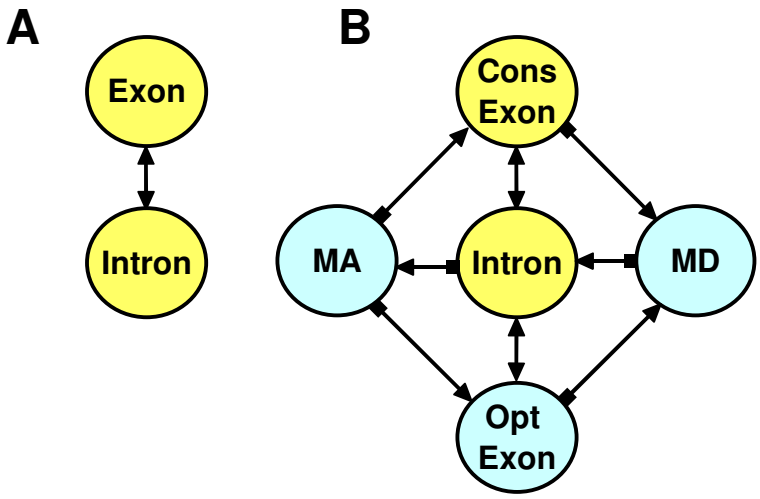

Figure 4.6: A is the original N-SCAN internal exon and intron state structure. B shows the new optional-coding states added to the original N-SCAN HMM. All the blue states represent optional coding sequence. Since the open reading frame is maintained across all internal exons, all features with blue states are constrained to have a length divisible by 3.

1. Generate layers of EST alignment assemblies, as in the MS_ESTSEQ algorithm
2. Detect all optional exons and exon extensions and replace the features in those positions with the appropriate alternative splice feature
3. For each position in the target sequence, if in any of the layers
   - an optional exon exists, mark it with an '`O`'
   - otherwise, if an exon extension exists, mark it with an '`M`'
   - otherwise, if an exon exists, mark it with an '`E`'
   - otherwise, if an intron exists, mark it with an '`I`'
   - otherwise, mark it with an '`N`' (There must be an intergenic region.)

Figure 4.7: The algorithm for generating AS_ESTSEQ

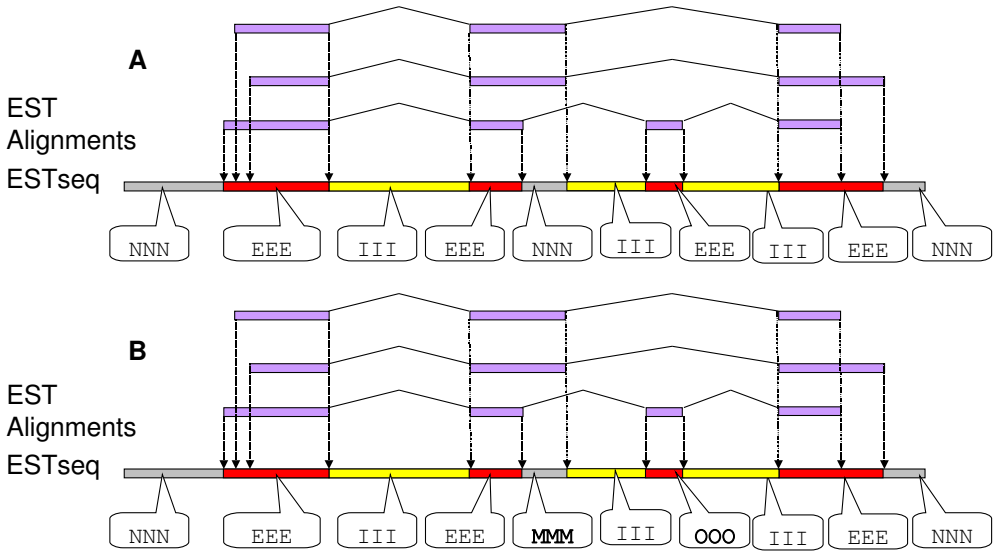

Figure 4.8: Differing definitions of EST sequence. In the current method (A), all conflicting overlapping exons and introns are marked as 'unknown' or intergenic. The new method (B) marks a character for all optional coding sequence.

Two new characters are added: one which indicates an exon extension in the EST alignment assembly ('M' for multiple splice site) and one which indicates an optional exon ('O'). The algorithm for creating these is shown in figure 4.7. Note that the algorithm favors marking an alternative splicing event even if it is an ambiguous case. The differences between the old method and the new method are visually compared in figure 4.8.

4.6.3 Definition of alternative splicing events for training purposes

An alternative splicing event may have many aberrant manifestations, some of which may not fit into a clear category. There is a clear need to carefully classify the different types of events, as well as distinguish true alternate isoforms from possible sequencing or alignment error in the annotation process. This is not the focus of this

work, however, and for the purposes of training and generating AS_ESTSEQ , we chose the following definitions for the modeled alternative splicing events. Note that the same detection algorithms are used both for parameter estimation and generation of the AS_ESTSEQ .

Optional Exons

Optional exons are sometimes called "cassette" exons. This name comes from the visual similarity to a cassette: the cassette exon can be inserted in between two constituent exons. Optional exons can exist as a difference between two transcripts, one containing the exon and the other not. In order to have a proper cassette exon, however, the two transcripts must share a supercassette in common. That is, the internal splice sites of the flanking exons in the cassette-exon-containing isoform must share the coordinates of an intron in the cassette-exon-lacking isoform. This is illustrated in figure 4.9.

In summary, an exon is considered optional if and only if there exists two transcripts which share identical boundaries in a supercassette and one transcript contains an optional exon and another does not.

Whenever an optional exon is detected, all transcripts containing that supercasette are converted to transcripts containing an optional exon at the splice sites indicated by the optional exon.

Exon Extensions

I often refer to exon extensions as "multi-sites", for the fact that they are simply additional splice sites to an exon in the same region. A multi-site extends a "constituent" exon, which in this case simply comes to mean shorter exon. It is defined by a pair of exons, one of which has at least one different splice sites. The only real constraint on extension-pair candidates is that the exons overlap, since the exons may differ on both splice sites and still be considered a valid multi-site. In order to simplify the problem of algorithmically determining the constituent exon, only one end–5' or 3'–is considered at a time and the innermost splice site is considered that of the constituent

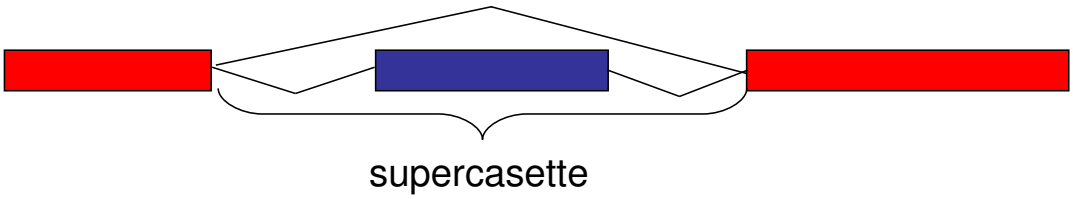

Figure 4.9: Illustration of an optional exon. Here both splice forms are merged into a single transcripts, with the downward facing intron lines representing the splice path of the optional-exon-containing isoform.

exon. It is possible for the constituent exon to vary from one multisite to another, as illustrated in figure 4.10.

Whenever an exon extension is detected between a pair of exons, both transcripts add a feature for the exon extension, shoving aside any features in its way.

Interpretation of Alternate Splice Events in Training vs. AS_ESTSEQ Generation

When training the parameters of an HMM, training data is converted into the feature sequence that the HMM would generate given the annotations. This is a fairly straightforward task until multiple isoforms per gene are taken into consideration.

The default behavior with our training algorithm is to weight the features based on how many other transcripts exist in that position of the sequence according to the training data. If alternate splice states are present in the HMM, our training algorithm takes an additional step after recasting the annotations as feature sequences to detect alternative splicing events. Each of the alternative splicing states may have one of the two above events associated with it. If an event is detected, the state associated with that event replaces the features in that region of the transcript, as described above. The same thing occurs with the AS_ESTSEQ generation, as described in the algorithm.

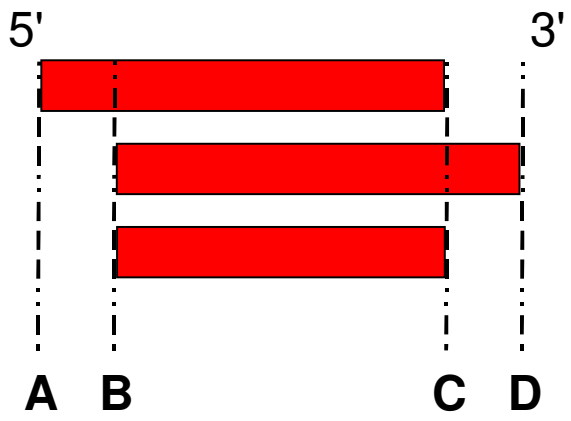

Figure 4.10: Illustration of two exon extensions. The intervals from A-B and C-D represent exon extensions. Algorithmically, when considering the 5' end of the top two exons, the middle exon of the three is considered "constituent", even though it is not necessarily the smallest nor constituent for that matter. This is irrelevant when considering just the 5' end, though.

The difference between the usage of these two is in connotation. 5' and 3' extensions are treated equally in the AS_ESTSEQ and strand information is ignored. For training, the HMM goes through separate states when on the plus or minus strand, and also depending on which end of the exon is extended. Thus a different feature sequence is necessary for training. Similarly, strand and frame information are a part of the ASHMM states, whereas the AS_ESTSEQ carries none of this information.

4.6.4 Training the new ASHMM

When parameterizing an HMM, several decisions can be made. With regards to scoring any state, the content, duration and transition model are independent, and can be tied to any model. For instance, in the Genscan HMM, the Intron and Intergenic states have the same content model, but have different locations in the topology and use different length distributions. The features with these two states manifest in completely different ways, even though the content model is identical.

Given this, it is easy to imagine that similar things can be done for these new alternative splicing states.

DNA Content Models

A brief investigation into retraining the DNA content models proved bad: the models are overfit to the data. Several of the parameters had no examples, even without cross validation. Additionally, experiments showed that using new DNA content models worsened performance as compared to N-SCAN_EST . Therefore, all DNA content models were tied to their constituent parameters, i.e. 5' extensions were tied to the original acceptor and coding region parameters, 3' extensions were tied to the original donor and coding parameters and optional exons were tied to the original donor, acceptor and coding parameters.

Conservation Models

New conservation models were added for all optional features: extension donor, extension acceptor, extension coding and all the analogous models for optional coding sequence. The parameter sets for these models are identical to their analogous constituent models, however the actual parameters were estimated separately.

AS_ESTSEQ Models

Similarly, new AS_ESTSEQ models were added. These models were estimated similarly to the N-SCAN_EST models, except that a new 5-character alphabet was being used rather than a 3-character alphabet. Separate, 5^{th}-order Markov chains were estimated for content in the exon extension regions, as well as optional coding region. Acceptor and donor sites for exon extensions and optional exons were estimated separately each using a 2^{nd} order WAM.

The NULL model was estimated from intergenic region. Intergenic region was considered the bases from 1000 to 250 bases upstream of the annotated transcription start site and 250 to 1000 bases downstream of the last annotated 3' UTR. If any of these annotated regions overlapped a non-intergenic feature, the region was not used to estimate intergenic parameters.

Chapter 5

Results and Conclusion

In this chapter, we discuss some results in an attempt to estimate the relative accuracy of the methods presented in this thesis. Until now, N-SCAN_EST has reported the best accuracy results of any gene predictor in human and *C.elegans*. We thus use N-SCAN_EST as our baseline.

We tested three major methods, the original N-SCAN_EST method, the novel N-SCAN_MP_EST method which runs N-SCAN_EST on mutually exclusive EST alignments and the N-SCAN_AS_EST method, which adds new alternative splicing states to the original N-SCAN HMM. We tested the predictive accuracy of these methods alone, and then after using PASA to augment the predictions.

5.1 Sensitivity and specificity measures

The typical metric for predictive accuracy is a measure of sensitivity and specificity of several levels of features: gene, transcript and exon. Sensitivity is a measure of how much of the test set was predicted, and specificity is a measure of how much of the predicted set was correct. More formally,

$$\mathrm{Sn} = \frac{\mathrm{TP}}{\mathrm{FN} + \mathrm{TP}} \tag{5.1}$$

$$\mathrm{Sp} = \frac{\mathrm{TP}}{\mathrm{FP} + \mathrm{TP}} \tag{5.2}$$

where TP is true positives, FP is false positives and FN is false negatives.

TP + FN is calculated as the size of the testing set, and FP + TP is the total number of predictions.

5.2 Cross validation

In all cases where accuracy measures are reported, we used a 4-fold cross validation procedure to eliminate the possibility of underfit models. First, we clean the annotation sets for non-canonical signals, excessively short introns (less than 20 base pairs) and inframe stop codons. We then cluster the annotations into genes by the following definition: if any two transcripts have at least onat least one identical coding exon, then both are considered a part of the same cluster.

We then randomly partition the genes into four sets. These make up a testing set. The examples in the test set are left out in training. We run training on all four sets, make predictions on all four sets, and take an average of the four for sensitivity measures,

The notable exception is the N-SCAN_MP_EST results. N-SCAN_MP_EST is a proof of concept, and is largely impractical to use. The methods for training N-SCAN_MP_EST are mostly identical to those of N-SCAN_EST and the accuracy of that training method is proved in its cross validation results. The difference between N-SCAN_MP_EST and N-SCAN_EST in training is that several additional EST sequences are used against training data that is optimized for each of the sequences as they match to the EST alignments. Since the parameter set is the same, and the training examples are only increased by this method, there is no risk of underfitting, provided N-SCAN_EST is proved.

Cross validation would have been carried out, however, the idea was to take the method to its maximum capacity. The result is that N-SCAN_EST is run 16, 80, 20, 25 and 15 times each[1] with the same target DNA sequence but different EST sequences. As shown in what follows, this is very time consuming and not practical for cross-validation.

The sets were still evaluated over the same four testing sets as the other two methods, and the results were averaged similarly.

[1] on their respective chromosomes in lexicographical order

Method	Tx Sn	Tx Sp	Ex Sn	Ex Sp
N-SCAN_EST	.4963	**.4887**	.8081	**.7069**
N-SCAN_MP_EST	**.5471**	.2744	**.8338**	.4260
N-SCAN_AS_EST	.5197	**.4786**	.8125	**.6998**

Figure 5.1: The accuracy of each method.

5.3 How to read UCSC annotation pictures

The illustrations which follow are images generated from the UCSC genome browser [22]. We uploaded the predictions from or *D.melanogaster* runs as a custom track to compare alongside the reference sequence annotations as well as the BLAT EST alignments. It is worth noting that the BLAT alignments on the browser do not necessarily match those used in these experiments.

The UCSC browser illustrates exons with thick horizontal lines, and introns with thin horizontal lines with arrowheads superimposed, indicating the direction of transcription. In some cases, single exon transcripts have arrowheads imposed on the exon itself.

The illustrations of all our predictions (always shown on top)exclusively show coding exons and the introns between them. UTR exons are shown on all other annotations with thinner lines than the coding exon lines.

5.4 *Sans* PASA, N-SCAN_MP_EST performs best

Initially, we evaluated the performance of each of the methods on their own, using our four-fold cross validation method. Chromosomes 2L, 2R, 3L, 3R, 4 and X were used.

N-SCAN_MP_EST returned the most correct genes, transcripts and exons, however was not nearly as specific as the other methods. Many additional transcript predictions were made that did not have a match in the testing set. The results are shown in figure 5.1.

Clearly, N-SCAN_MP_EST draws from the strong predictive accuracy of N-SCAN_EST and harnesses it to make more predictions than N-SCAN_EST does, through its multiple passes. N-SCAN_AS_EST is certainly a close contender, and substantially improves on N-SCAN_EST levels.

5.4.1 N-SCAN_AS_EST shows no AS predictive power

We took a closer look at the predictions from one of the N-SCAN_AS_EST test runs, filtering out only transcripts which were alternatively spliced. In the entire run, no exon extensions were predicted. True exon extensions are often short, and given the low transition probability to the extension states and likely overfitting of the empirical length distribution, paths through these extension states did not score highly.

Optional exons were frequently predicted, but not nearly as much as in the annotation set. 128 alternate isoforms were predicted in all, while in the full reference sequence set there are 5,457. Only 4 of the alternately spliced transcripts were correctly predicted from the test set, and none of them correctly predicted more than one isoform of the gene.

In this test, the N-SCAN_AS_EST HMM has no effect on the number of transcripts per gene that it correctly predicts. Why are the predictions better, then? The likely answer lies in PASA's intelligent use of the BLAT and Sim4 alignment tools, as well as its assembly algorithm. In order to test this hypothesis, we took a look at one layer of the N-SCAN_MP_EST runs (rather than multiple runs concatenated together), showing the following results:

Transcript Sn	.5141
Transcript Sp	.5180
Exon Sn	8174
Exon Sp	.1880

These are better than the N-SCAN_EST results in figure 5.1. It would appear that PASA's good alignments play a role in increasing the N-SCAN_EST gene prediction accuracy.

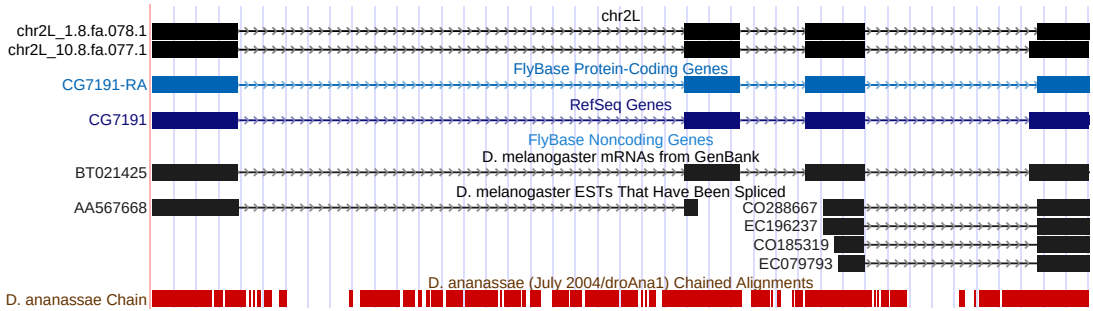

Figure 5.2: An example where N-SCAN_MP_EST correctly predicts an alternately spliced gene with incomplete EST information.

5.4.2 N-SCAN_MP_EST predicts complex, subtle ASs

N-SCAN_MP_EST predicted 4,458 alternate isoforms of genes. This is much closer to the 5,457 isoforms in the reference sequence set. Upon inspection, the forms varied quite a bit more, often predicting exon extensions and optional exons simultaneously.

Exon extensions were successfully predicted in some cases. In figure 5.2, the extension is correctly predicted, even though no full length cDNA gives evidence for the additional isoform.

N-SCAN_MP_EST is also capable of predicting mutually exclusive exons, as shown in figure 5.3. Additionally, we see a correctly predicted antisense intronic gene in figure 5.4, a feat not previously done with a single probability-based gene predictor.

The low specificity level of N-SCAN_MP_EST is likely caused by frequent over-splitting of transcripts. In many of these predictions, alternate forms were returned, however they either split one true transcript into two transcripts or joined two separate true transcripts with a long intron, as shown in figure 5.5. This is likely the consequence of a lack of exon type information from the EST sequence. Since the EST sequence does not distinguish between the different types of exons, N-SCAN_MP_EST will join separate transcripts.

Given that N-SCAN_MP_EST predicts so many exons correctly, this gives it prime candidacy for updates with PASA. PASA was designed to split and join genes in places where they had seemingly been incorrectly joined or split, respectively. An example of PASA updating an N-SCAN_MP_EST prediction correctly is shown in figure 5.6.

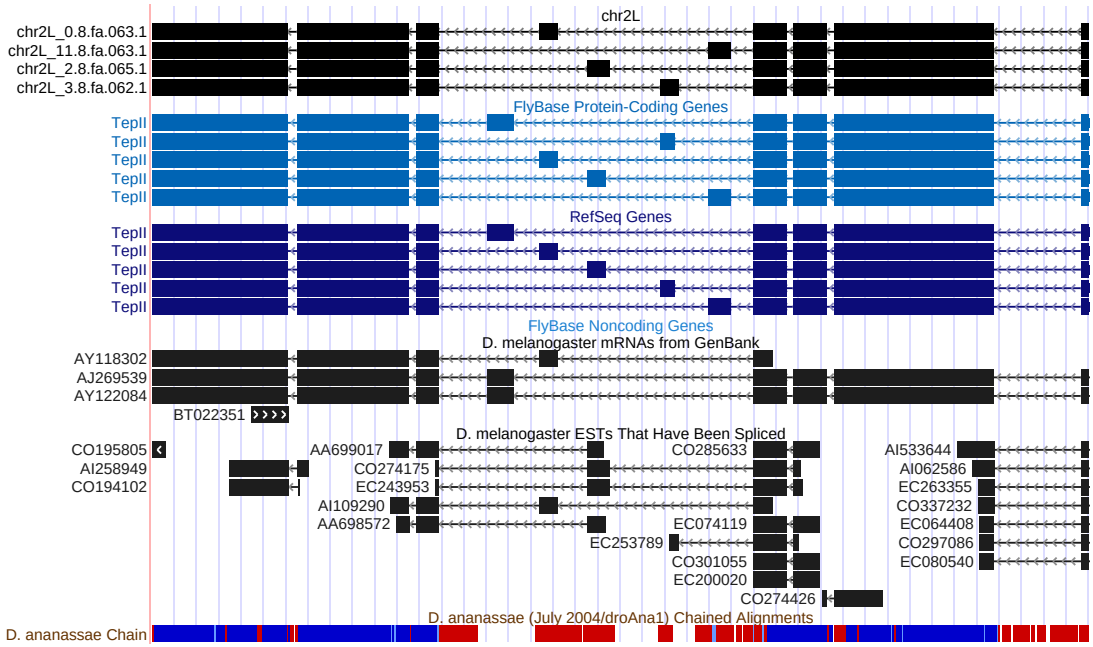

Figure 5.3: Here we see a case where N-SCAN_MP_EST was able to reconstruct several mutually exclusive exon-containing transcripts, mostly relying on EST evidence.

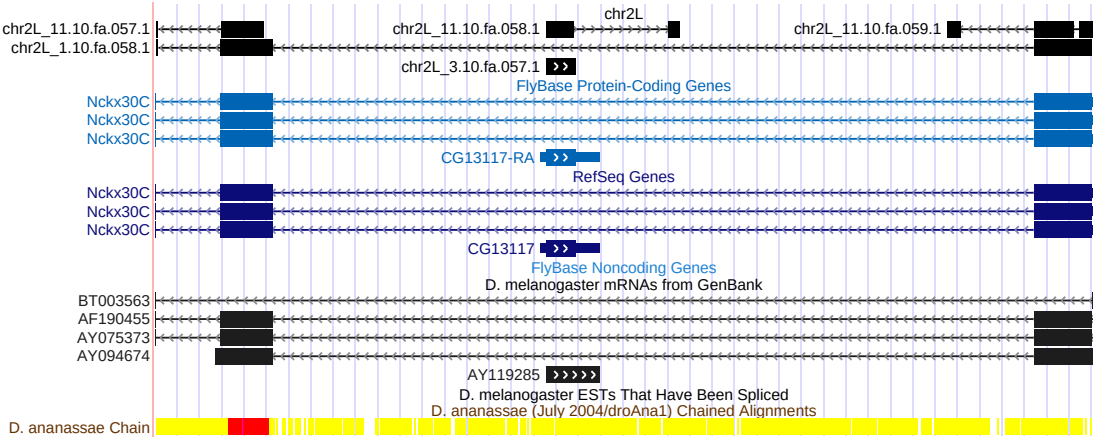

Figure 5.4: Here we see an instance of N-SCAN_MP_EST predicting an antisense gene in the intron region. This is generally not possible with gene predictors.

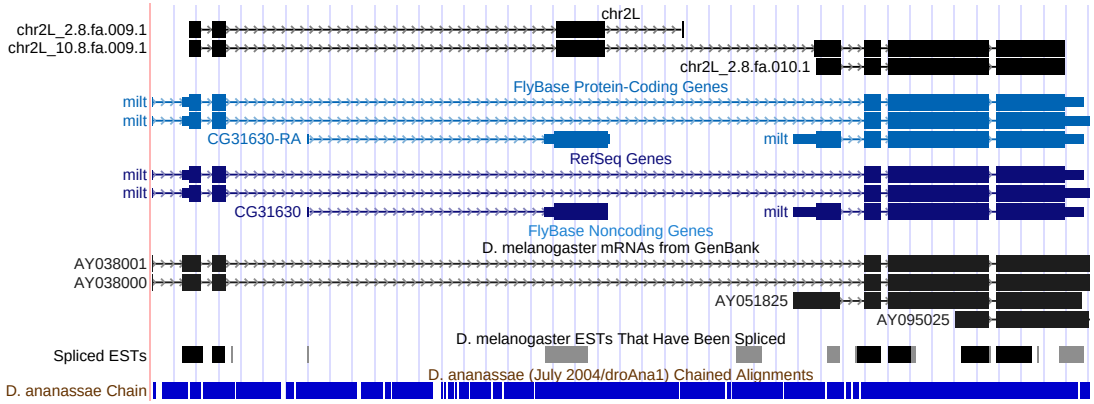

Figure 5.5: Here N-SCAN_MP_EST incorrectly predicts three separate transcripts as a single transcript.

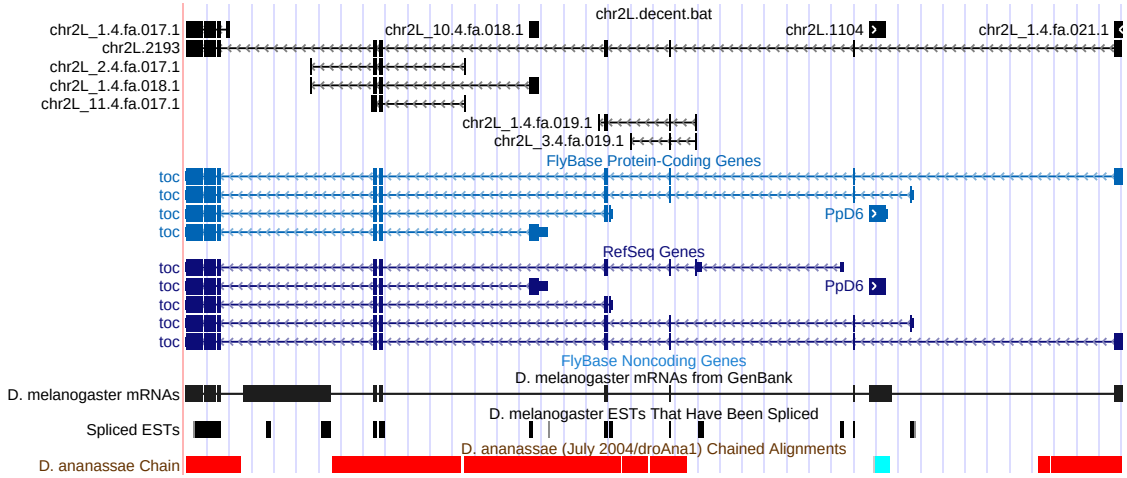

Figure 5.6: In this picture, chr2L.2193 represents a new transcript generated by a PASA update to all the other transcripts, predicted by N-SCAN_MP_EST . This is an example of scattered, but correctly predicted N-SCAN_MP_EST transcripts of are joined together into a long transcript, chr2L.2193.

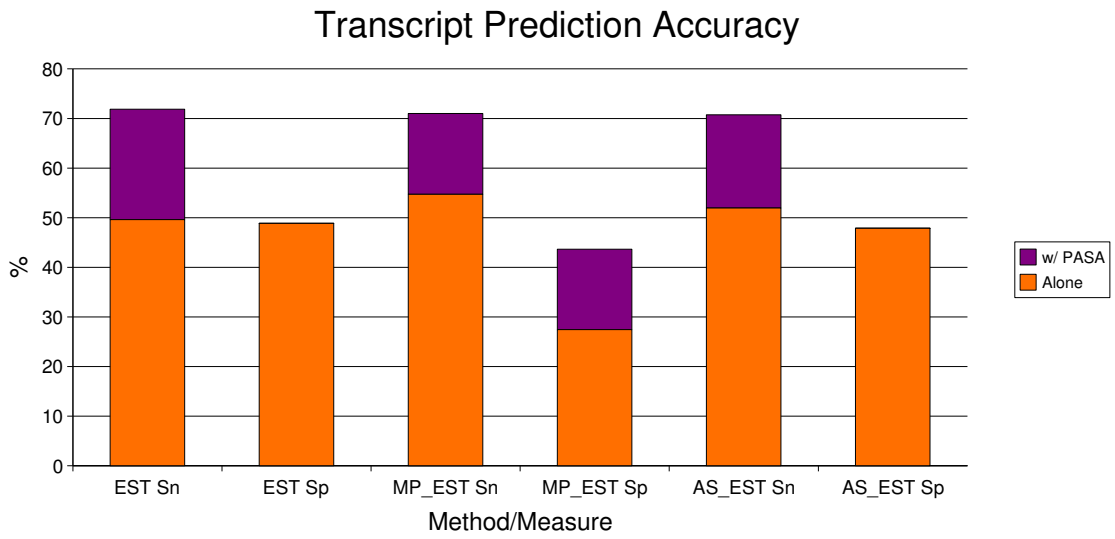

Figure 5.7: Transcript prediction accuracy of the three methods. A transcript is considered correct only if all coding exons predicted correctly.

5.5 With PASA, all methods perform comparably

We used all *D.melanogaster* full-length cDNAs and ESTs available from the UCSC browser against UCSC version 2 of the genome to update our predictions with PASA.

We expect in general that our gene predictor covers the vast majority of all gene loci, but the exact gene sensitivity suffers from cases where splice sites or translation initiation and termination signals are incorrectly called. PASA, which updates the annotations with additional splice forms as well as novel genes (from full-length cDNAs), should compensate for this shortcoming, and indeed it does.

The use of PASA as a post-processing step negates all distinctions between the predictive ability, as shown in figure 5.7. PASA will remove alternate isoforms of a gene if there is little evidence of their existence. This seems to compensate for the excessive transcripts N-SCAN_MP_EST predicts.

Indeed, the combined use of PASA and any of these gene predictors makes for a strong annotation system, given the mass of EST and cDNA evidence for transcription.

5.6 Reducing ESTs lowers PASA's improvement

While the above figures for gene accuracy are quite impressive for complete *de novo* gene prediction, a lot of these correct predictions can be recovered from the cDNAs and ESTs alone.

In order to simulate the annotation of newer genomes with fewer EST sequences associated to them, we artificially reduced the number of ESTs considered for training and annotation update with PASA. Beginning with the 255,654 ESTs, we pared the initial set down to 250,000 ESTs at random. Each progressively smaller set is a subset of the larger one, with 50,000 more ESTs randomly removed. No full-length cDNA sequences were introduced to the system.

In all steps along the way, including the generation of EST sequence, training, and PASA post-processing, the number of ESTs used was limited to the indicated amount. Figure 5.8 shows the results.

Bearing in mind that PASA would never predict any isoforms without evidence from a full length cDNA (not presented here) or a prior annotation, it becomes clear that the stronger the gene predictor is, the more predictions can be made. In genomes with fewer than 1,000 ESTs, we might say PASA is of very little effect, and the use of ESTs seems to hinder the N-SCAN_EST prediction power.

The rate of change of the margin between N-SCAN_EST only predictions and PASA-aided predictions appears to peak somewhere between 10,000 and 100,000 ESTs. While one might jump to say that this is a "sweet spot" for an EST sequencing project, one must account for the number of ESTs in proportion to the number of genes expected in the genome. Expression bias and other factors might contribute to the success of these methods.

Regardless of the optimum, N-SCAN_EST predicts roughly half of the transcripts in *D.melanogaster* with only 10,000 ESTs, representing an inexpensive solution to annotation.

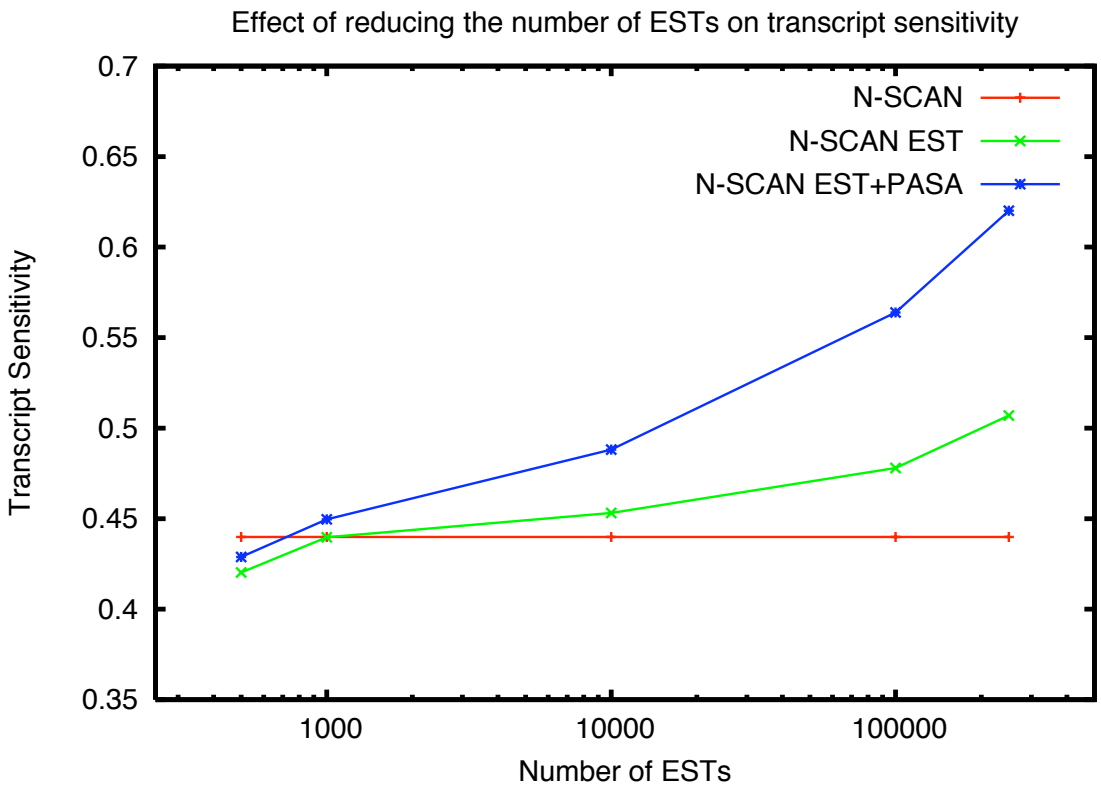

Figure 5.8: Transcript sensitivity is plotted as a function of the number of ESTs in the system. N-SCAN remains the same because it does not use ESTs. Note the x-axis has a log scale.

5.7 Discussion

We have tested a few methods for more completely annotating genomes without the use of human interaction. We have shown that using EST evidence can help predict alternate isoforms, both with a generative probabilistic model as well as with heuristic integration of EST alignments.

5.7.1 All predictors perform well with PASA

It is clear that the use of all available EST and mRNA evidence in the *D.melanogaster* genome, we can closely reconstruct the EST-supported examples in the hand-curated reference sequence. This method, unlike sequence-only methods, still allows for unsupported or underexpressed genes to be predicted, unlike in EST-only methods.

Further, we come very close to these results using only a limited number of ESTs that we integrate int N-SCAN_EST predictions. This indicates that new genomes, at least at the level of complexity of the *D.melanogaster* genome, can be reliably annotated with sparse data on transcription.

Our goal was to create a trainable, adaptable system for automated annotation with limited sequencing requirements. We believe we accomplished this with the method of combining N-SCAN_EST with the PASA pipeline. Even with as few as 10,000 ESTs, this system was able to predict 49% of all transcripts in the annotation set. The payoff for EST evidence is much higher with this method than it is with N-SCAN_EST alone, as well. In order to approach 49% sensitivity with N-SCAN_EST alone, 100,000 ESTs were required.

5.7.2 Few ESTs are required for a strong annotation

In the experiment above, we were able to predict roughly half of the transcripts in the reference sequence set of the *D.melanogaster* genome using only 10,000 ESTs. This represents an inexpensive sequencing project when compared to full-length cDNA sequencing solutions. Newer genomes can additionally use EST alignments of sequences from related genomes to increase the supporting evidence.

5.7.3 AS modeling is difficult in an HMM context

While directly modeling is the most attractive and elegant-sounding solution to predicting alternative splicing, it seems that in practice it does not work. Previous efforts have been made on our part to model optional exons in the following ways (among others): modeling the increased conservation levels in the flanking introns, modeling the codon bias (should be enriched for exonic splice enhancer hexamers) and modeling the DNA splice content (should have less information as compared to constituent exons). As alluded to in section 3.3.4, the Genscan DNA content model is quite delicate, and adding sensible models to it is not a non-destructive action. The direct use of EST evidence for alternative splicing, apart from these previous ones, is a rational, definite and believable method that *should* work, however we see that it does not.

Several difficulties arise, namely that many alternative splicing events shift the frame of the transcript and also that multiple events happen in concert. The Markov property prevents modeling of these higher-order effects of alternative splicing. Additionally, as compared to the total mass of the coding sequence, these events occupy a small fraction. Using our models they therefore do not represent an "average" event, whereas translation initiation, splice signals and translation termination signals are consistent events for a gene sequence.

While one project that we know of, ExAlt, does use an HMM to model alternative splicing events, there is no confirmation of the frequency of the correct discrimination between optional and constituent coding sequence reported in this paper [3]. Furthermore, the system only predicts a single constituent exon at a time, allowing it to predict frame shifts, and otherwise non-Markov friendly effects of alternative splicing.

We highly doubt that a solution for modeling full-length transcripts with alternative splicing will employ an HMM to determine optional and constituent features.

5.7.4 Running time makes N-SCAN_MP_EST not viable

In the above analysis of limiting EST evidence, the N-SCAN_MP_EST method was left out entirely, and if you recall from the methods section, N-SCAN_MP_EST was not cross validated. This was done primarily because the running time for this method is unreasonable (see figure 5.9).

	Time (h)
N-SCAN_EST	22
N-SCAN_MP_EST	1875
N-SCAN_AS_EST	55

Figure 5.9: Total CPU hours for whole genome runs on *D.melanogaster* UCSC version 2. This only includes the running time of the gene predictor. Parameter Estimation, EST alignment, and post-processing are additional steps in the process, but are close in running time for all 3 methods.

This makes running N-SCAN_MP_EST difficult to experiment on. Furthermore, it is impractical for most modest compute clusters to run. For this reason, we simply consider N-SCAN_MP_EST a proof-of-concept method answering the question, if we suggest to N-SCAN_EST that it should predict all forms of a gene with EST evidence, will it do so? It seems that the answer is "mostly".

5.8 Future work

Despite the running time problems of the N-SCAN_MP_EST method, the predictive power of this method is quite impressive, when taking into account the complex and subtle differences between isoforms that it successfully models. With this as a starting point for annotation update, we might see even stronger annotations with little EST evidence.

Currently work is being done on memory optimized versions of N-SCAN and N-SCAN_EST [19]. These algorithms enable us to run multiple instances of a Viterbi trellis on a single CPU. Additionally, the differing paths through these trellises can be found with a small fraction of the total computations for the trellis, since the parallel trellises often share the same path in the vast majority of positions in the input sequence. Thus the algorithm for N-SCAN_MP_EST could be completely reproduced efficiently, only computing additional probabilities where EST alignments disagree with each other.

Additionally, the EST assemblies produced by PASA contain information about which splice sites (those internal to the assembly) are reliable and which are not (the flanking

exons). This information could be used to more carefully select splice sites in the prediction phase.

Appendix A

Implementation

A.1 `iscan` and `zoe`

`zoe` is the implementation of the research projects of Twinscan, N-SCAN, N-SCAN_EST and Pairagon. `iscan` is the front-end where prediction runs can be invoked in any of the above modes, as well as Genscan mode, which is identical to Twinscan mode except that no conservation model is present. `zoe` contains many optimizations which shortcut some of the slower parts of the Viterbi algorithm by pruning paths which are impossible (incorrect splice signal) and pre-computing all models to eschew redundant emissions calculations.

The `zoe` library is written in C and is open source, available at http://mblab.wustl.edu. The design and much of the initial implementation of the code was done by Ian Korf, and includes contributions from Daniel Duan, Paul Flicek, Charles Vaske, Samuel Gross, Evan Keibler, Manimozhiyan Arumugam, Chaochun Wei and Randall Brown.

In order to carry out the above experiments, a few minor modifications to the code were made.

A.1.1 Addition of new feature factories

When running the Viterbi algorithm, `zoe` generates exons at each position and takes the maximum of all features and the remaining states with an object referred to as a feature factory. (This name comes from the fact that it "manufactures" features.) The feature factories provide a speed-up to the algorithm as well, ignoring all possible

features which cannot exist at the current position or length because of a missing signal.

Additional feature factories were created for the Multi Acceptor and Multi Donor states, which unlike the normal exon states, were bounded by two of the same type of splice site.

A.1.2 Addition of traceback interpretation algorithm

N-SCAN_AS_EST reports all possible traces implied by all optional coding sequences. For each optional coding region, 2 transcripts are implied, one including the sequence in the putative transcript and one leaving it out. Thus 2^N transcripts are reported, where N is the number of predicted optional coding regions.

In order to correctly report all 2^N isoforms in GTF format, the annotations are treated as a tree, branching at every point where an optional coding region is predicted, from low to high coordinate on the plus strand sequence. The left side of the tree includes all features, and the right side of the tree excludes all optional features, and all subtrees follow this pattern. The root of the tree is empty, and a tree with no optional features will have a left leaf node only, representing the full transcript. The tree is visited in post-order.

A.2 iParameterEstimation

iParameterEstimation (iPE) is a system for estimating parameters for generalized hidden Markov models. iPE was specifically written and designed by the author for use with the Brent Lab `zoe` code base, however will perform maximum likelihood estimation for any type of hidden Markov model. The framework was designed for customization and accessibility on all levels: user, model designer and coder.

iPE is written in about 60,000 lines of Perl and C source code and is Open Source under the Perl license. It is currently in beta version and available for download at http://mblab.wustl.edu. An extensive user guide is available [34].

A.2.1 Interface

iPE receives all its configuration through XML files provided by the user. All command line options are eliminated, and instead input through the "instance" file, which defines an entire run of parameter estimation. The purpose of this feature is to more require the user to leave a "paper trail" to the experiments carried out. The resulting parameter file includes the name and location of the instance file used to generate it.

Additionally, a gHMM file is required and pointed to by the instance file. The gHMM file includes a complete description of all the models included in the gHMM. This includes initial probabilities, transition probabilities and duration distribution functions. All "hard-coded" parameters, such as initial probabilities, tweaked transition probabilities, numbers generated from previous experiments (e.g. signal peptide parameters and promoter models) are detailed here. No numbers are hard-coded into the iPE code base.

Finally, a feature map file is provided by the user. This file describes how to translate annotation files into state sequences. This can be useful, for example, in the case where you want to consider the flanking bases of an annotation to be the intergenic null model, as in the above experiments.

A.2.2 Annotation Engine

The system features an engine which minimizes code rewriting. This includes a portion of code which will translate annotations to gHMM state sequences by user-defined translation rules. Additional annotation formats can be incorporated through the use of the Annotation Plugin system. The Plugins are simple Perl objects which are expected to populate a list of iPE transcript objects with the data from a given annotation. The Plugin does not have to translate the coordinates of the input file; this is done with utility routines.

In the gHMM file, the user can specify the use of one or more "`altsplice`" states. These states are programmed to detect certain alternative splicing events, and can be coded with a minimal amount of effort. The AltSplice modules are expected to convert the features to altsplice states if the appropriate alternative splices is presented, adding additional features when necessary.

All the remaining overlapping features which are not designated as alternatively spliced are weighted by the percentage of transcripts at a given position that share that feature.

Each region to be counted is subdivided into regions where different submodels exist for each feature, and then into submodels for those submodels, and so on. All of this is done internally, and no calculations need to be performed by the user. The locations of the submodels are defined as being relative to a feature's begin or end coordinate.

A.2.3 Performance

The program has been thoroughly validated and no users have met any major bugs since the beta release. In all cases, this program has equaled or surpassed the predictive performance of previous implementations of parameter estimation.

The program can estimate parameters using all of the current Reference Sequences for the human genome in about two and a half hours using under four gigabytes of memory. (This memory requirement will be drastically reduced before version 1.0.)

A.3 The PASA pipeline

The PASA pipeline is a large set of Perl scripts, C and C++ programs which run on the Linux operating system with i386 line processors. It utilizes MySQL and CGI scripting for web output. It is available on the TIGR website at `http://www.tigr.org`.

Few modifications were made, except to provide facilities to exchange formats between the Brent Lab GTF annotation format, and PASA's native MySQL and GFF3 formats. Additional, external scripts to generate the multiple EST sequences for N-SCAN_MP_EST and the AltSplice EST sequences for N-SCAN_AS_EST were written in perl. These will eventually be merged into the PASA code base, as we are actively working with the primary author, Brian Haas, on developing the software.

A.4 Eval

All evaluations were made with Eval software written by Evan Keibler [20]. Since its inception, this software has been expanded to include validation scripts for GTF-formatted files and FASTA-formatted sequence files. This validation pipeline was used to clean all annotation sets, including code by Randy Brown, Jeltje van Baren and Mikhail Velikanov.

A.5 Biological Annotation Tool

The most trying challenge to our current code base was the N-SCAN_MP_EST outputs. In order to properly evaluate the results, all redundant transcripts must be eliminated, and all transcripts belonging to a parent gene must be merged into one. Our current implementation was in Perl as a part of the Eval package, and naïvely compared all transcripts and exons to each other to determine the gene membership.

In order to combat this problem, we decided to reimplement this algorithm in C. Along the way, we began work the framework for a larger annotation tool, which is a current sourceforge (`http://www.sourceforge.net`) project.

The new algorithm takes $O(NM + GF)$ time, where N is the number of features in the annotation, and M is the maximum number of overlapping features in the annotation, G is the number of true genes in the annotation, and F is the maximum number of features in any gene in the annotation. The algorithm has not run over 5 minutes for any input on a 2GHz AMD processor, as compared to the previous performance of one day per chromosome.

References

[1] Ritesh Agrawal and Gary D. Stormo. Using mrnas lengths to accurately predict alternatively spliced gene products in *Caenorhabditis elegans*. *Bioinformatics*, 22(10), 2006.

[2] Bruce Alberts, Alexander Johnson, Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter. *Molecular Biology of the Cell*. Garland, fourth edition, 2002.

[3] Jonathan E. Allen and Steven L. Salzberg. A phylogenetic generalized hidden markov model for predicting alternatively spliced exons. *Algorithms for Molecular Biology*, 1, August 2006.

[4] Randall Brown, Samuel Gross, and Michael R. Brent. Begin at the beginning: Predicting genes with 5' utrs. 15, 2005.

[5] Christopher Burge and Samuel Karlin. Prediction of complete gene structures in human genomic dna. *Journal of Molecular Biology*, 268(1), 1997.

[6] Simon L. Cawley and Pachter Lior. Hmm sampling and applications to gene finding and alternative splicing. *Bioinformatics*, 19, 2003.

[7] Val Curwen, Eduardo Eyras, T. Daniel Andrews, Laura Clarke, Emmanuel Mongin, Steven M.J. Searl, and Michele Clamp. The ensembl automatic gene annotation system. *Genome Research*, 14, 2004.

[8] Gideon Dror, Rotem Sorek, and Ron Shamir. Accurate identification of alternatively spliced exons using support vector machine. *Bioinformatics*, 21(7), 2005.

[9] Rachel A. Drysdale, Madeline A. Crosby, and The FlyBase Consortium. Flybase: genes and gene models. *Neucleic Acids Research*, 33, September 2004.

[10] James W. Ficket and Chang-Shung Tunk. Assessment of protein coding measures. *Neucleic Acids Research*, 20(24), November 1992.

[11] Liliana Florea, George Hartzell, Zheng Zhang, Gerald M. Rubin, and Webb Miller. A computer program for aligning a cdna sequence with a genomic dna sequence. *Genome Research*, 8(9), 1998.

[12] Sylvain Foissac and Thomas Schiex. Integrating alternative splicing detection into gene prediction. *BMC Bioinformatics*, February 2005.

[13] The Institute for Genome Research. seqclean. http://www.tigr.org.

[14] L.J. Gibson. Did life begin in an rna world? *Origins*, 20(1), 1993.

[15] Warren Gish. Wu-blast. http://blast.wustl.edu.

[16] Samuel Gross and Michael Brent. Using multiple alignments to improve gene prediction. *Journal of Computational Biology*, 13(2), 2006.

[17] Brian J. Haas, Arthur L. Delcher, Stephen M. Mount, Jennifer R. Wortman, Rojer K. Smith Jr., Linka I. Hannkick, Rama Maiti, Rusch Ronning, Catherine M., Town Douglas B., Christopher D., Steven L. Salzberg, and Owen White. Improving the *Arabidopsis* genome annotation using maximal transcript alignment assemblies. *Neucleic Acids Research*, 1(19), 2003.

[18] Jian Jiang and HowardJ. Jacob. Ebest: An automated tool using expressed sequence tags to delineate gene structure. *Genome Research*, 8, 1998.

[19] Evan Keibler, Manimozhiyan Arumugam, and Michael R. Brent. The treeterbi and parallel treeterbi algorithms: Efficient, optimal decoding for ordinary, generalized, and pair hmms. Unpublished.

[20] Evan Keibler and Michael Brent. Eval: A software package for analysis of genome annotations. *BMC Bioinformatics*, 3(50), 2003.

[21] W. James Kent. Blat–the blast-like alignment tool. *Genome Research*, 12, 2002.

[22] W.J. Kent, C. W. Sugnet, T. S. Furey, K.M. Roskin, T. H. Pringle, A. M. Zahler, and D. Haussler. The human genome browser at ucsc. *Genome Research*, 12(6), 2002.

[23] Ian Korf. Gene finding in novel genomes. *BMC Bioinformatics*, 4, May 2004.

[24] Ian Korf, Paul Flicek, Daniel Duan, and Michael R. Brent. Integrating genomic homology into gene structure prediction. *Bioinformatics*, 17, April 2001.

[25] Richard Mott. Est_genome: a program to align spliced dna sequences to unspliced genomic dna. *Computer Applications in the Biosciences*, 13(4), 1997.

[26] Uwe Ohler, Noam Shomron, and Christopher Burge. Recognition of unknown conserved alternatively spliced exons. *PLoS Computational Biology*, 1(2), 2005.

[27] K.D Pruitt, T. Tatusova, and D.R Maglott. Ncbi reference sequence (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Neucleic Acids Research*, 33(1), 2005.

[28] L.R. Rabiner and B.H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, January 1986.

[29] Scott et al. Schwartz. Human–mouse alignments with blastz. *Genome Research*, 13, 2002.

[30] Rotem Sorek, Ronen Shemesh, Yuval Cohen, Ortal Basechess, Gil Ast, and Ron Shamir. A non-est-based method for exon-skipping prediction. *Genome Research*, 2004.

[31] Chaochun Wei and Michael R Brent. Using ests to improve the accuracy of *de novo* gene prediction. *BMC Bioinformatics*, July 2006.

[32] Jia Qian Wu, David Schteynberg, Manimozhiyan Arumugam, Richard A. Gibbs, and Michael R. Brent. Identification of rat genes by twinscan gene prediction, rt-pcr, and direct sequencing. *Genome Research*, 14, 2004.

[33] Mihaela et al. Zavolan. Impact of alternative initiation, splicing and termination on the diversity of the mrna transcripts encoded by the mouse transcriptome. *Genome Research*, 13, 2003.

[34] Bob Zimmermann. iparameterestimation user guide. Available at http://mblab.wustl.edu/.

Vita

Robert Zimmermann

rpz@cse.wustl.edu

Office
Campus Box 1045
One Brookings Drive
St. Louis, Missouri 63130
(314) 935-9187

Home
750 Westgate
Apartment 8
University City, MO 63130
(314) 413-8797

Interests

Computational Biology; Alternative Splicing Events

Education

2004–2006 Master of Science, Computer Science (to be completed 12-2006)
Washington University in St. Louis
Specialization: Computational Biology, Gene Prediction
Thesis Topic: Leveraging EST Evidence to Automatically Predict Alternatively Spliced Genes
Chair: Michael R. Brent, Computer Science
Committee: Jeremy D. Buhler, Computer Science
Gary Stormo, Genetics

1999–2003 Bachelor of Science, Computer Science
University of Michigan in Ann Arbor
Specialization: Software Development, Educational Software
Elected to Phi Beta Kappa

Research

2004–present **Laboratory for Computational Genomics**
Michael R. Brent, Department of Computer Science and Genetics
Washington University in St. Louis
Development of new parameterization of the N-SCAN model for prediction of exon extension and exon skipping events. Involvement in several projects, including revising the N-SCAN model, and adopting new conservation alphabets for Twinscan.

2002–2004 **Center for Highly Interactive Computing in Education**
Elliot Soloway, Department of Computer Science and Education
University of Michigan, Ann Arbor
Research in new user interfaces for the PalmOS for use in the classroom. Authored an organizer, a chemistry modeling/animation tool, and a participatory simulations backend, and maintained over seven other packages.

Teaching

2004,2005 **Algorithms for Computational Biology**
Washington University in St. Louis
Teaching Assistant
Responsible for design and implementation of all lab and homework assignments. Lead several help sessions and assisted students with office hours. Received excellent reviews.

Work Experience

2003–2004 **GoKnow, Inc.**
Software Engineer/Tester
Responsible for versioning, testing, and maintaining code bases for several software packages for release.

2004 **University of Michigan Transportation**
Web Designer/Programmer
Designed and implemented RosterManager, a PHP/MySQL package which maintained a roster system for the student drivers. Included authentication, profile information, and the capability for supervisors to view and upload rosters, and students to view and sign off for their shifts.

2002–2003 **University of Michigan**
Department of Mechanical Engineering
Software Engineer/Programmer
Finalized a large suite of programs to evaluate, chart, and otherwise visualize a number of statics and dynamics equations for students.

Computer Languages

C, C++, Perl, PHP, HTML, XML, DTD, Java, MySQL, Shell Scripting, Unix/Linux System Administration

Honors

2005 Graduate Assistantship, Laboratory for Computational Genomics

2004 Distinguished Master's Fellowship, Washington University in St. Louis

2002 Phi Beta Kappa Award

Publications

Fu, Yan, Chenhong Zhang, Zimmermann, Bob, Barbazuk, Brad and Brent, Michael. *Species-specific TWINSCAN significantly improves the ab initio gene predictions in maize and rice. In progress.*

W. Brad Barbazuk, Bob Zimmermann, Michael R. Brent (2006, January). *Ab initio gene finding in maize.* Poster presentation preseented at the Plant and Animal Genomes XIV Conference.

Lisa Ann Scott, Robert Zimmermann, Hsin-Yi Chang, Mary Heitzman, Joseph Krajcik, Kate Lynch McNeill, Christ Quintana, Elliot Soloway. *Chemation: a handheld chemistry modeling and animation tool.* Proceeding of the 2004 conference on interaction design and children: building a community. 145–146.

References

Michael R. Brent
Professor of Computer Science
Campus Box 1045
One Brookings Drive
Washington University
St. Louis, MO 63130
(314) 935-6621
brent@cse.wustl.edu

Jeremy D. Buhler
Assistant Professor
Campus Box 1045
One Brookings Drive
Washington University
St. Louis, MO 63130
(314) 935-6180
jbuhler@cse.wustl.edu

Sean R. Eddy
Alvin Goldfarb Distinguished Professor of Computational Biology
Center for Genome Sciences
HHMI/Dept. of Genetics, Box 8510
Washington University School of Medicine
4444 Forest Park Blvd.
Saint Louis MO 63108
(314) 362-7666
eddy@genetics.wustl.edu