Predicting Intron Locations in Non-Model Organism Expressed Sequence Tags (ESTs) Using Comparative Homology with Divergent Model Organism Genomes

by

S.M. Al Mamun

A thesis submitted to
The Faculty of Graduate Studies of
The University of Manitoba
in partial fulfillment of the requirements
of the degree of

Master of Science

Department of Computer Science
The University of Manitoba
Winnipeg, Manitoba, Canada
January 2014

Thesis advisors

Michael Domaratzki and Barbara J. Sharanowski

Author

S.M. Al Mamun

Predicting Intron Locations in Non-Model Organism Expressed Sequence Tags (ESTs) Using Comparative Homology with Divergent Model Organism Genomes

Abstract

Finding the approximate location of short read genome sequences by comparing them to an already available closely related organism's complete genome sequence is a challenging research issue. Predicting intron locations in the short form of mRNA called Expressed Sequence Tags (ESTs) and the variability of intron lengths are the major challenges. More specifically, finding the intron positions in an EST sequence by comparing it with a reference genome sequence is a time consuming task, as currently it is done manually.

In my thesis, I designed a pipeline that can predict the intron positions in ESTs of non-model organisms. Initially, I compared the ESTs to the closest completely sequenced genome. The pipeline then finds the alignment of the ESTs, the reference genome sequence, and the coding region of the gene (known as Coding DNA Sequence or CDS) from the reference genome.

Contents

List of Figures

Acknowledgments

I would like to begin by thanking my advisors, my committee members, my parents, my significant other, and all the people who have supported me along the way.

This thesis is dedicated to my parents, my sister, and my love.
You know who you are and what you mean to me.

Chapter 1

Introduction

The genome sequence of an organism is the information that is embedded in its Deoxyribonucleic Acid (DNA), which stores all heritable information for that specific individual (see Chapter 2 for further explanation about DNA). The regions of DNA that control heritable traits are known as genes. Most of the living organisms can be sorted into two groups. These two groups are known as prokaryotes and eukaryotes. A eukaryotic gene is a combination of exons and introns, where an exon is a portion of the DNA sequence that codes for a protein, while an intron is not involved in coding for proteins. An intron is excised from a gene prior to transcription. As a result, in eukaryotes the information for forming a protein is a subsequence of the original gene sequence. Figure 1.1 shows the process that includes the conversion from a gene sequence in DNA to mRNA (known as transcription) and from mRNA to a protein (known as translation). Initially, the gene is shown as a combination of exons and introns in Figure 1.1. In the next step, all the introns are excised and

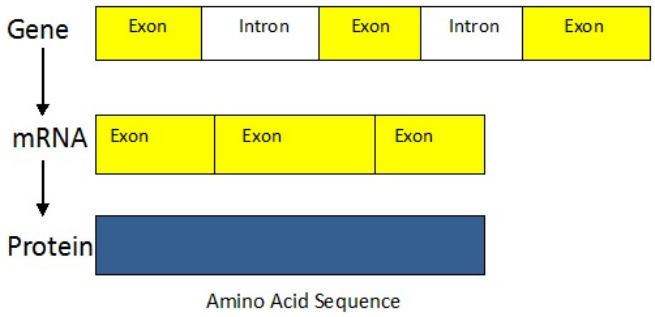

Figure 1.1: A diagram showing the fate of a DNA gene sequence through the processes of transcription and translation. In the diagram, transcription is happening during the process from first row to second row and translation is happening from second row to third row, based on a diagram from The Maize Full Length cDNA Project web page mai [2013].

mRNA is formed from the exons. Finally, the mRNA is translated into a protein which is a sequence of amino acids. In my thesis, my focus is on eukaryotic organisms only because prokariyotes do not have introns (see Chapter 2 for further information about mRNA).

DNA copies of mRNA can be made from mRNA using the enzyme reverse transcriptase. These copies are referred to as cDNAs. A partial sequence of cDNA is called an Expressed Sequence Tag (EST) (discussed in more detail in Chapter 2). cDNA or ESTs do not have introns like a regular gene and are more suitable than mRNA for biological experiments [Wilkeninga and Baderb, 2004] because cDNA is more chemically stable while being a highly similar copy of the mRNA. ESTs, sequenced from cDNA can be used to identify the gene, which is known as annotation. Figure 1.2 shows how an EST can be mapped to a genome sequence to identify a gene.

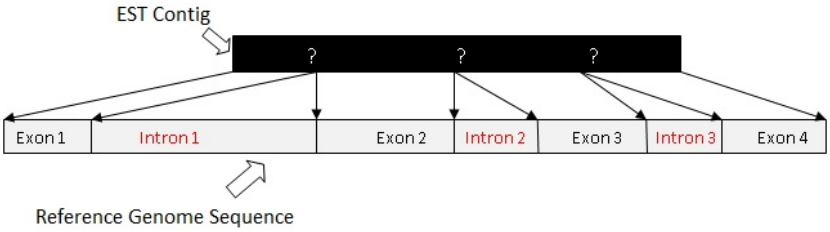

Figure 1.2: A diagram depicting a coding sequence (solid black rectangle) aligned to a reference genome sequence (transparent rectangle). The question marks indicate the possible location of the splicing position of the exons. The arrows indicate the boundaries of exons.

Unfortunately, complete genome sequences are available only for 4,129 species as indexed in The Genomes OnLine Database (GOLD) [Pagani et al., 2013] and among them only 183 are eukaryotes. Based on the available genome sequences, organisms are classified into two categories: model and non-model organisms. Organisms whose complete genome sequences are well studied, probably sequenced completely, and provide insight on the genomes of other organisms are known as model organisms [Nawy, 2012]. On the other hand, organisms whose complete genome sequences are not sequenced or not available are called non-model organisms [Nawy, 2012]. For instance, divergence of the honey bee, *Apis mellifera*, from parasitic wasps occurred in the family Ichneumonidae millions of years ago ($>$ 150 million years ago [Grimaldi and Engel, 2005]), which is the motivation of my thesis. More specifically, I am using the honey bee and ESTs from the parasitic wasps *Campoletis sonorensis* (Ichneumonidae) (from Sharanowski et al. [2010]) to predict the intron-exon boundaries in wasp ESTs.

Existing methods, such as in the est2genome [Mott, 1997] software, do not perform well for mapping an EST to a complete genome, to predict the location of introns

in that EST. The existing methods can sometimes identify the exons, but can not identify where the introns and the intron-exon junctions are located, since mutations in introns vary largely even among related species. The location of introns within genes is important for primer design, gene annotation, and comparative genomics. As a result, we need a new method to determine the intron positions in non-model organisms. Figure 1.2 shows an overview of the problem. We do not know the splicing sites in ESTs, indicated by the question marks in the figure. We need to find the positions where one exon ends and another intron starts.

Several questions come to mind while we are considering non-model organisms. Why is the intron location important for non-model organisms? Why is it difficult to find the intron positions in a sequence? Why can the available methods and softwares not produce results to predict the intron positions in ESTs for non-model organisms? The following few paragraphs in this chapter will answer the above questions to demonstrate the motivation for the research done in this thesis.

During the last few decades introns were considered as junk DNA [Ohno, 1972] that did not have any importance in gene-regulatory functions. The reality is different, as some introns may be functional [Comeron, 2001]. Jia et al. [2004] concluded that introns are functionally much more complex than exons.

Next, finding intron locations is difficult as they can vary largely in terms of length and position within a gene [Yandell et al., 2006]. Also, splice sites may not be present as intron loss may have occurred [Zhang, 1998] or there may be additional splice sites if intron gain has occurred.

Variation in the length of introns is an important factor associated with introns due to mutation [Comeron, 2001]. Moreover, intron length may help us to compare between similar phylogenetic organisms. In particular, the lengths of introns may differ largely even among the same genes in closely related organisms. According to Haddrill et al. [2005], there is no known consistent patterns of evolutionary constraint in introns.

There may be a correlation between exon and intron length. According to Zhu et al. [2009], there are correlations between sequence divergence, length, and the ordinal position of introns and exons among different organisms. The same authors also suggested that divergence rate is strongly and oppositely correlated with an intron's ordinal position, such that downstream introns demonstrate greater sequence divergence across organisms. Haddrill et al. [2005] suggested that functional elements might be ubiquitous in longer introns and these introns might have large roles in regulatory gene expression.

To find introns, the short reads (ESTs) of the species of the interest need to be compared against complete genome sequences of the available most closely related model organism. For insects and other invertebrates, the closest model organism may have diverged 100-200 million years ago from the species of interest. In addition, introns are not uniformly distributed across the genome of the complex eukaryotes [Cooper, 2000]. For example, more than 90% of the human blood clotting factor VIII coding gene is made up of introns [Cooper, 2000], whereas histone genes (which are very important for epigenetic regulation) do not contain any introns [Pandey et al., 1990]. Many biologists previously classified introns as junk DNA because it was accepted

that these sequences are not of any biological importance. However, Moran et al. [1999] strongly disagreed with this notion, stating that many of the exon swapping events (also known as exon shuffling) have been found overlapping in the exon-intron boundaries. In addition, introns have been found to harbour important regulatory information on gene expression [Rose, 2008]. Therefore, due to the evolving nature of all biological systems, it is very difficult to design an intron prediction tool, which takes all of these factors into consideration.

Some software packages are able to find intron-exon junctions with some limitations, such as est2genome [Mott, 1997], orf finder [Rombel et al., 2002], and ensemble [Shen and Chou, 2006] (discussed further in Chapter 3). The major limitation is that none of the software packages claim that they can be applicable for non-model organisms. For instance, est2genome can find the exon positions, ignoring small introns, for model organisms. One of the main problems in finding intron-exon junctions is the variability in intron-exon regions in the genome across divergent species. Most of the software (e.g., orf finder, ensemble) looks for certain conserved properties in the genome sequence that help to define intron-exon boundaries. Unfortunately, those signatures are not present in all genes. This is due to the variable length number, and distribution of the introns along the genome [Deutsch and Long, 1999]. Furthermore, during gene expression, these conserved signatures have been found to be altered by the complex gene regulatory networks of organisms [Mattick, 2009]. Therefore, due to the variability of the introns, rather than relying solely on software to define the exon-intron junctions, scientists tend to take advantage of techniques used in wet labs to verify results obtained from the software. In particular, software packages are a

good starting point for initial intron-exon junction prediction but must be followed up and confirmed by sequencing both transcripts and genomic DNA, to determine the intron-exon boundaries, and comparing the alignment to the results produced by the software packages.

Mapping ESTs of non-model organisms to organisms with completely sequenced genomes will potentially allow the prediction of intron-exon boundaries in the transcripts of the non-model organism. I designed a pipeline for aiding intron location predictions in transcripts of non-model organisms. A pipeline is a combination of tools which acts in sequence and transforms a set of raw inputs to a set of outputs. To the best of my knowledge, there has not been any major bioinformatics research on introns of non-model organisms. In my method, all the ESTs (see Chapter 2 for further explanation about ESTs) are given as input data for the pipeline and the output is an alignment of the predicted intron-exon junctions in those ESTs using the aid of a reference genome and mRNA transcripts from the same species as the reference genome.

For evaluation, I compare the accuracy of the pipeline with the accuracy of the results obtained by manual alignment of 52 genes with a reference genome. For the dataset of 52 genes, the sensitivity of the pipeline is 92.12% and the specificity is 100% for the total number of exons predicted in the aligned test genes (two closely related bee species) of *Apis mellifera* and *Bombus terrestris*. For introns, the specificity of the pipeline is 89.79% and the sensitivity is 100% for the total number of introns predicted in the test ESTs.

Chapter 2

Backgroud

2.1 Biology

All organisms, with the exception of certain viruses, contain a molecule called deoxyribonucleic Acid (DNA) that is used to encode genetic instructions in these organisms. DNA is made up of a hereditary sequence of four nucleotides: adenine (A), cytosine (C), guanine (G) and thymine (T). DNA can also be thought of as a sequence of elements from the alphabet A, C, G, T. These four nucleotides can form two different complementary pairs: adenine pairs with thymine, and cytosine pairs with guanine. This complementary rule is universal in all species and forms a backbone structure when DNA forms in a twisted pattern, which is known as the double stranded helix structure of DNA. The asymmetric nature of the genetic instructions in DNA results from a direction in the way the DNA can be read. A symbolic representation of a single strand of DNA sequence is shown in Figure 2.1.

Figure 2.1: A diagram depicting a 14bp nucleotide sequence; nucleotides are shown inside circles and the connected lines are the bonds between the adjacent two nucleotides.

The nucleotides in DNA make bonds with two complementary bases in opposite strands at the same position, forming base pairs (bp) in the double-strand of DNA. The bonds that occur in this form are commonly known as Watson-Crick bonds [Watson and Crick, 1953]. Those two strands go in opposite directions to each other in an 'anti-parallel' order, where one strand is known as the forward strand and the other is known as the reverse complement strand. The forward strand is directed in the $5'$ (five prime) to $3'$ (three prime) direction and the reverse complement strand is directed in the $3'$ (three prime) to $5'$ (five prime) direction of the forward strand. Figure 2.2 shows an example of 'anti-parallel' strands of DNA.

The genome of an organism consists of many genes that code for proteins. The genome also contains large portions of the sequence that do not code for any protein. The function of those large portions of non-coding sequence are mostly unknown [Woolfe et al., 2004]. The coding region of a gene, also known as Coding DNA Sequence (CDS), provides the instructions to produce a specific protein. On the other hand, from cDNA, RNA is generated and from RNA, Expressed Sequence Tags (ESTs) are sequenced [NCBI, 2013a]. ESTs may be used to identify gene transcripts, and are instrumental in gene discovery and gene sequence determination [Adams et al., 1991]. ESTs also help to identify unknown genes by using sequence compari-

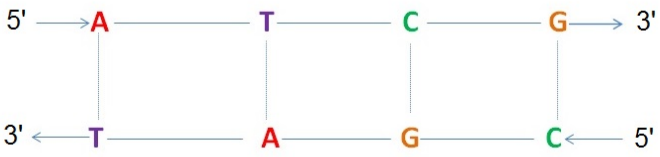

Figure 2.2: Anti-parallel strands of DNA. Top strand of the DNA sequence of the diagram is read in the forward direction as 'ATCG' and the DNA strand on the bottom of the diagram is read as 'CGAT' in the reverse complement.

son to determine gene homology in other organisms. The way ESTs are sequenced, many ESTs are overlapping fragments of the CDS. ESTs are simply assemblies of the raw reads into unique genes.

The conversion of DNA into a protein is shown in Figure 1.1. We already saw that DNA can be expressed as a sequence of nucleotides (A, C, G, and T). During *transcription*, the information in DNA is transcribed to mRNA, which contains only exons. Figure 2.3 shows that a typical mature mRNA is constructed with a 5′ cap, a 5′ UTR (Untranslated Region), a Coding DNA Sequence (CDS), a 3′ UTR, and a Poly-A tail. The 5′ cap and Poly-A tail mark the starting and the end point of the mRNA in *translation*. The 5′ UTR and 3′ UTR aids the stability of the mRNA and controls gene expression [Kuersten and Goodwin, 2003]. CDS is the sequence that actually codes for the protein.

In general, the only informational difference between DNA and mRNA is that mRNA is a combination of A, C, G, U. Here, thymine (T) in DNA has been replaced by uracil (U) in RNA. Next, the information in mRNA is translated into a protein, which is a sequence of 20 different amino acids and this process is known as translation.

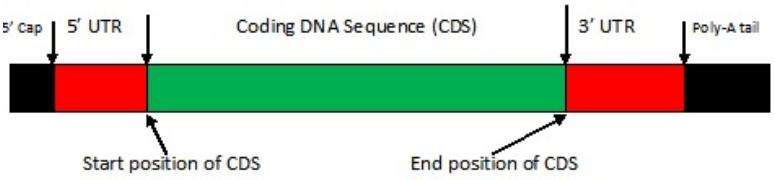

Figure 2.3: A typical mRNA structure. Here, UTR stands for Untranslated Region.

The amino acid sequence in a protein is determined by a combination of three-nucleotides, creating a codon. Each codon is translated into an amino acid. Based on the starting position of a codon, there are three possible reading frames if we consider the mRNA strand from left to right (5′ to 3′). Similarly, there are three other possible reading frames if we consider the strand from the opposite direction (3′ to 5′). Translation starts with an initiation codon called a start codon and ends with a terminator codon called a stop codon. For instance, if we translate the nucleotides in forward reading frames in Figure 2.1, amino acid sequences would be NR*SX, IDRVX, and SIEL. Here, N, R, S, I, D, E, L are different amino acids, * stands for a stop codon and X represents an ambiguous codon. Please see Appendix A for more details on the genetic code table.

Organisms typically have several chromosomes, which are collectively referred to as the complete genome sequence or genome. Genomes are usually measured in terms of their length of the sequence. The length of a genome is same as the number of base pairs in the genome of an organism. For example, the complete genome sequence of *Apis mellifera* consists of approximately 236 million base pairs [Munoz-Torres et al., 2010].

2.2 From Biology to Informatics

In this section, I will focus on the necessary terms and information to understand the following chapters.

2.2.1 Sequence Alignment

In bioinformatics, sequence alignment is an algorithmic approach of comparing two or more sequences (DNA, RNA, or protein) to find the similarity between them. The goal of the sequence alignment is to locate the homologous regions among the sequences by maximizing their similarity. The rationale behind the sequence alignment is evolution. During evolution, sequences can change due to mutation, whereby a nucleotide can change to a different nucleotide or a nucleotide may be inserted or deleted (indel) in the sequence. Alignments show regions where changes have occurred. When there are more than two sequences in an alignment, it is called a multiple sequence alignment. Sequence alignments also provide a score based on mismatches, insertions, or deletions between the two sequences and this score indicates the similarity of the sequences in that alignment. In sequence alignment, sequences are placed horizontally one after another. Figure 2.4 shows a multiple sequence alignment for three different protein sequences. In this figure, mismatches are represented by columns where the amino acids differ across organisms. Gaps are identified by dashes (-) and imply that a deletion or an insertion mutation has occurred (referred to as indel in short).

Sequence alignment is also used to locate the region of similarity between an EST

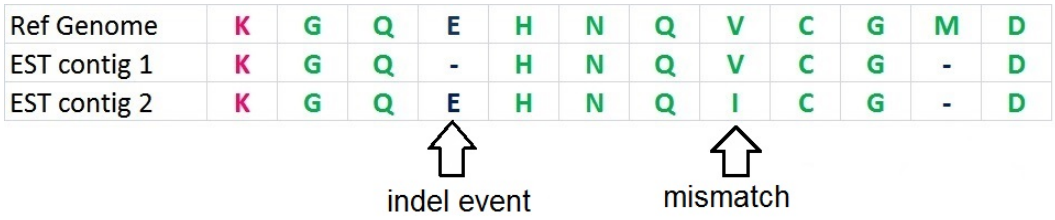

Figure 2.4: A sequence alignment between three amino acid sequences from a reference genome and two different ESTs. The alignment also shows indel events and a mismatch.

and a genomic sequence. Since both the ESTs and the genome are represented as strings of the alphabet {A, T, C, G} (in Figure 2.4, they are represented as strings, from a set of 20 different amino acids), we can think of EST mapping as an instance of *string matching*. Due to mutation, mapping ESTs to a genome sequence is an *approximate string alignment* problem, which is more complex than a simple *string matching* problem. Instead of $\Theta(n)$ matching time for a *string matching* problem, *approximate string alignment* takes at least $\Theta(mn)$ time; where m is the length of the pattern or ESTs (in number of base pairs) and n is the length of the genome sequence (in number of base pairs).

Most of the popular sequence alignment algorithms use dynamic programming approaches. For instance, dynamic programming is used by the gapped BLAST (Basic Local Alignment Search Tool) algorithm by Altschul et al. [1997]. BLAST is the most popular tool to search a DNA or a protein query sequence against a database of sequences for similarity. In the following subsection, I will discuss the function of BLAST.

2.2.2 BLAST

BLAST is an algorithm to compare protein or DNA queries with protein or DNA databases. In my thesis, I am using BLAST to preprocess the ESTs. In a BLAST search, the DNA or protein sequence that is submitted for comparison is called the *query sequence* and the sequence that is compared with the query sequence is known as *subject sequence* [Wheeler and Bhagwat, 2007]. A BLAST search returns a set of results when the score based on the match between the query and the subject sequence surpasses a minimum threshold. The minimum threshold parameter is also known as the expect (e) value. The e value is the default sorting metric in BLAST and typically gives the same sorted ordering as the maximum score of the sequence alignment. E value illustrates the number of hits that we can "expect" to find randomly while searching a database of a fixed size [NCBI, 2013b]. For a query sequence to be matched with database sequences, the default maximum e value is 10, which means that 10 matches between the query and the database sequence are expected merely by chance if the query sequence has that e value. The lower the e value is, the better is the match of the query in the BLAST search.

In a BLAST search, at first, sequence repeats are removed from the query sequence. Then, BLAST creates a k-letter word list containing substrings (words) of length k where k is 3 for protein sequences and k is 11 for DNA query sequences. For all the words in the list, BLAST organizes the high-scoring words into an efficient search tree. Target sequences in the target database are then scanned for these high-scoring words of the search tree. After that BLAST extends the exact matches

to high-scoring segment pairs. The corresponding matches in the previous step are considered as the region for Smith-Waterman [Smith and Waterman, 1981] alignment. If the e value is lower than the threshold expect value, then BLAST shows the alignment in the result. If several matches (hits) are identified, only the best hit on each strand is considered in the BLAST result.

In my thesis, I am using tblastx to map the ESTs to a genome sequence. tblastx is commonly known as "translated query vs. translated database" [NCBI, 2013b]. It takes a nucleotide sequence query and translates it in all six possible reading frames (both in forward and reverse order). Then, all of those six translated query sequences are compared to the target database sequences, dynamically translated in all six reading frames. tblastx is useful for identifying coding regions in a genome when ESTs are passed to tblastx as a query sequence. The correct reading frame of the ESTs might not be known; tblastx solves the problem by comparing all possible reading frames. More specifically, I am trying to maximize the regions of matches of amino acids between the ESTs and the reference genome in my thesis.

2.3 Statistics

2.3.1 Sensitivity, Specificity, and Accuracy

To measure the accuracy and the performance for a binary classification or identification problem, there are three different statistical measures.

Sensitivity (SN) measures the accuracy of positive classification and can be represented as SN = TP / (TP + FN), where TP and FN refer to the number of true positives (correctly identified) and the number of false negatives (incorrectly rejected), respectively.

Specificity (SP) measures the accuracy of negative classification and is represented as SP = TN / (TN + FP), where TN and FP refer to the number of true negatives (correctly rejected) and false positives (incorrectly identified), respectively.

Accuracy (ACC) represents the proportion of true classification using both positive and negative measurements and is represented as

ACC = (TP + TN) / (TP + FP + TN + FN)

Chapter 3

Related Work

To date, ESTs are still difficult to align with genome sequences containing introns. Currently, existing alignment programs can not adequately locate the position of intron-exon junctions in transcripts. Kent [2002] and Brudno et al. [2003] claimed that several tools, such as BLAST [Altschul et al., 1997], BLAT [Kent, 2002], DIALIGN [Morgenstern et al., 1998], MUMmer [Kurtz et al., 2004], AVID [Bray et al., 2003], and LAGAN [Brudno et al., 2003] can align exons quite well. Most of these tools function in a similar way. For example, BLAST aligns chunks of the query and database sequence which are similar to each other. These smaller chunks can be thought of as "hot-spots" that are indexed in a table to rapidly find the matches. The length of the initial similar sequences is called the word-size [NCBI, 2013b]. Word-size is the parameter that adjusts the size of the sequences and is essential for BLAST to initiate extensions to align the entire sequence. Finally, the program searches for near-perfect matches by increasing or decreasing the word-size.

On the other hand, Kurtz et al. [2004] developed an open-source software program that can compare large genome sequences with large evolutionary distances. Their target was to develop an efficient tool, called MUMmer, that could compare two sequences of lengths greater than 1,000,000bp using an optimized suffix-tree algorithm. Kurtz et al. [2004] found that MUMmer can maintain an optimal or near-optimal worst-case run time depending on the algorithm used. MUMmer can align two genomes gene-by-gene or shorter fragments of genomes and concatenate the results together. Besides, MUMmer can compare even large genome sequences that BLAST can not do. In my thesis, I tried MUMmer for mapping ESTs to complete genome sequences but it failed to map them appropriately. MUMmer is a combination of suffix-tree (which finds all distinct subsequences in a given sequence), the longest increasing subsequences (LIS), and Smith-Waterman alignment algorithms. The reason for using suffix-tree in MUMmer is to discard the sequence repeats efficiently. As MUMmer uses the longest increasing subsequence algorithm, it is suitable for large sequences having approximate matches. On the other hand, ESTs are shorter in length and the longest increasing subsequence algorithm is not suitable for these short ESTs. As a result, MUMmer is not capable of handling the intron-exon junctions in the non-model ESTs. Though MUMmer can produce both exact and approximate mapping between two large eukaryotic genome sequences, my focus is on ESTs and I have developed a pipeline that can predict the intron-exon junctions in those ESTs.

Q-PALMA [Bona et al., 2008] can align short reads of next generation sequencing (NGS) data. For preprocessing, the inputs are short reads, produced by the Illumina NGS platform, and a genomic sequence. Q-PALMA contains three parts: (i) the

splice site prediction model, (ii) the dynamic programming algorithm, and (iii) the optimization of the scoring function. To compute accurate spliced alignments, Q-PALMA uses a supervised machine learning approach called Support Vector Machines (SVMs) with the 'weighted degree' kernel (which computes the similarity between two sequences considering substrings occurring in both strings) and extensions of the Smith-Waterman algorithm. In Q-PALMA, short read sequences are first aligned to identify unspliced reads. Then, unmapped sequences are sent to Q-PALMA to align again, to identify reads of at least half of the read length to find seeds for using in Q-PALMA. For each seed position, Q-PALMA aligns the read and returns a score. The best scoring alignment is returned as the spliced alignment of the read when the intron is confirmed to be of length twice the read. For training, Q-PALMA depends on previously known splice junctions. As a result, Q-PALMA is mostly biased to these training datasets. Also, Bona et al. [2008] mentioned that the short length and inherent high error rate in short reads represent a significant challenge to align them over the intron boundaries. Bona et al. [2008] shows that if the spliced reads overlap significantly (>4 bp) into the next exon region near the intron-exon boundary, then the error rate is only 0.5%. On the other hand, the error rate can be as high as 12% when only 1-2 bp overlaps in the next exon region. Q-PALMA was designed specifically for aligning NGS short reads. On the other hand, my thesis deals with the alignment between EST and the reference genome sequence to identify the intron-exon junctions in the EST. So, Q-PALMA is not a suitable method to locate the intron-exon junctions in non-model EST.

TopHat [Trapnell et al., 2009] is another program that aligns short reads to a

genome in order to identify exon splice junctions. TopHat uses an efficient read-mapping algorithm designed to align reads from an RNA transcript experiment to a reference genome without relying on known splice sites. TopHat has advantages over Q-PALMA in two cases. First, TopHat does not need any training with previously known junctions, and second is TopHat's speed, which is 2.2 million reads per CPU hour on a single core CPU. The worst-case run-time for the algorithm in TopHat [Trapnell et al., 2009] is around 3 times longer than would be taken to align with a quadratic-space program. In practice, the maximal-scoring segment is often much shorter than the complete genome length. Trapnell et al. [2009] claimed that the program runs only about 1.5 times slower in practice. Thus, TopHat is only efficient at producing fast output only for short read alignments with only an accuracy of a minimum of 72% of the splice junctions, which is not appropriate for my thesis work as accuracy is more important than speed. In TopHat, reads that are mappable on the reference genome are grouped into distinct clusters so that the reads within each cluster are linked together through overlapping regions. Each cluster then defines a putative exonic regions that subsequently helps to search exon-exon junctions. So, TopHat was designed for an alternative purpose, only to predict exon splice junctions of NGS short read sequences. The main issue with Tophat is that it can align RNA-seq reads from the model organism to that model organisms genome. It would likely not work for non-model organisms with any significant evolutionary distance from the sequenced genome

Collins et al. [2008] presented a mapping approach for non-model organisms. They indicated that to work with non-model organisms, mapping is the only technique if

a suitable reference genome is available. In the software, Collins et al. [2008] used short read sequences produced by Next Generation Sequencing platforms and ESTs from the closest reference genome were also used for evaluation. Then, both of the sequences were mapped to the smallest evolutionary distance model reference genome and sorted using a relational database to locate the genome areas of interest, which can be extracted and visualized later on. The results are shown only for the reads greater than 85bp in length and they got 92.4% mapping accuracy by correlating ESTs to a gene and by using a set of mapping parameters. They used MySQL as their database. As genome sequences do not follow any specific rule for storage, genome sequences are normally handled as simple text files. In my thesis, I tried to avoid using a relational database to avoid complexity and maintain simplicity. Also, they have focused only on locating the approximate matching region but not predicting the intron-exon junctions. The main issue with this software is that it is comparing ESTs to ESTs and does not map intron-exon junctions.

Li et al. [2010] recently introduced another new tool for comparing the whole genome sequences of two or more species. In their method, they considered only organisms that have a large number of introns. To improve the quality of the result, Li et al. [2010] used coding DNA sequence as query sequences and a lower gap penalty score while making the comparison in BLAST. The lower penalty score allowed the CDS to be extended over occasionally low-matched regions. In their experiment, they used five different teleost fishes having a similarity of more than 85% across the whole genome sequence. Li et al. [2010] only focused on aligning genomes to genomes and they did not deal with missing introns. In my thesis, I am following a similar technique

to that of Li et al. [2010], which is using an organism's genome sequence that is not completely sequenced for comparison to large and complete genome sequences. Also, my thesis is predicting the position of the introns and exons in the non-model genome sequences.

Another alignment program called est2genome [Mott, 1997] predicts genes by using sequence similarity. It aligns a set of spliced nucleotide sequences (such as ESTs) to a complete genomic sequence, inserting some appropriate intron lengths. It then gives a set of introns and exons as output based on the alignment. est2genome [Mott, 1997] uses the Smith-Waterman local alignment [Smith and Waterman, 1981] and a subsequent divide and conquer strategy. est2genome aligns a set of spliced nucleotide sequences to an unspliced genome sequence, inserting introns of arbitrary length when needed. The insertion of arbitrary introns in the alignment prevents est2genome from predicting intron-exon boundaries of EST. Mott [1997] mentioned that the choice of boundaries which minimizes indel and mismatch costs does not coincide exactly. For these reasons, est2genome is not an ideal tool to identify intron-exon junctions in ESTs. Rather, it is useful to define the intron-exon region in the chromosome of the reference genome. In my thesis, I am using est2genome to map the homologous region of the CDS to the reference genome sequence from the same organism.

On the other hand, Spidey [Wheelan et al., 2001] computes an alignment of mRNA to a genomic sequence with increasing detail by performing successive BLAST runs at decreasing stringency levels. Spidey can align mRNA to genomic sequences quite well (only the exons) but can not locate the exact exon position in the genome sequence, nor can it predict the intron-exon boundaries. It also takes a much longer time for

a single alignment compared to est2genome. This is why, instead of Spidey, I am using est2genome to map the homologous region of the CDS to the reference genome sequence.

However, I did not find any tool that can predict the specific intron positions in a coding sequence. Some researchers (e.g., Wolf et al. [2009], Rose [2004]) claimed that finding the location of intron positions is highly complex due to the vast variation in intron length across species. But locating intron positions should not be impossible. If identifying the intron-exon junction is possible, then it would also be possible to predict the location of the introns in non-model organisms. Thus, my thesis provides a novel way to solve the problem of predicting junctions in ESTs using a reference genome sequence.

Chapter 4

Solution Methodology

To predict intron positions in non-model organisms, I developed a pipeline. The pipeline was implemented using python in the Ubuntu linux system. At the very beginning of the pipeline, EST contigs are given as input and after a series of computational steps an alignment is produced between the EST contig, an orthologous transcript (mRNA) from a model organism, and the genome region of that transcript from the model organism. The alignment depicts the putative intron-exon junctions of the EST contig introduced as input. The most closely related model organism is chosen ahead of time, and genome sequences are downloaded to a local hard drive.

The pipeline will start processing the EST contig by finding a similar gene from a model organism using a similarity search. When a similarity search is done between an EST contig using tblastx and the NCBI nucleotide (nt) database, I retrieve the most similar mRNA transcript from the model organism (the putative ortholog of the EST contig) and the chromosome number in which this gene resides.

However, I restrict unsuitable EST contigs in the next step of the pipeline by setting a maximum expect value between the EST contig and the mRNA of the related model organism to prevent paralogs. Once the EST sequences from the similarity searches are filtered, the est2genome [Rice et al., 2000] software is used to identify the region in the complete genome sequence that is most similar to the mRNA sequence. If the target genome region in the chromosome sequences covers a longer region than 10,000 bp, I discard that EST contig as it is extremely diffcult to map a short contig (a few hundred nucleotides or less) to a chromosome sequence that is over a hundred times longer. If the genome region where the transcript lies is less than 10,000bp, then it is extracted and used in the next step of the pipeline.

Once the Coding DNA Sequence (CDS) from the mRNA and the extracted genomic region are available to pass to the next step of the pipeline, the exon finder algorithm that I developed is used to define the intron-exon boundaries in the CDS and in the extracted genome sequence. Finally, this profile alignment (consisting of the CDS and reference genome sequence) is aligned with the EST contig using MAFFT [Katoh et al., 2009] and an alignment is produced between the EST contig, the mRNA, and the genome sequence of the model organism (Figure 4.1). The EST contig is only logically aligned across the exons of the genome sequence, and thus the intron-exon boundaries are illustrated in the EST contig sequence. It is important to note that the EST contig may or may not span the entire coding sequence, as by nature, they are contigs of fragments of genes. My solution method includes a detailed description of the pipeline. Figure 4.1 shows a brief overview of the solution methodology of my thesis. The following subsections give more details about my

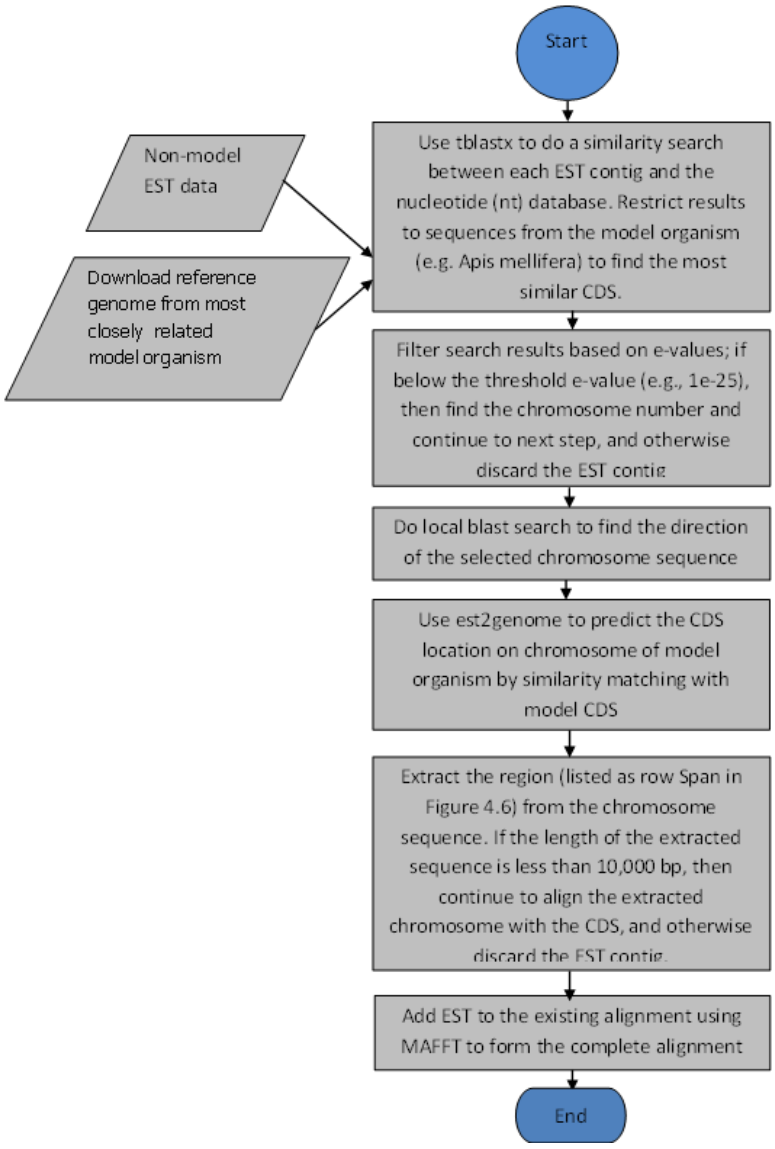

Figure 4.1: Detailed flow chart of solution methodology, showing the complete pipeline starting from raw data to the final output.

solution strategy. Figure 4.2 shows an overview of the structure of the final output of the pipeline.

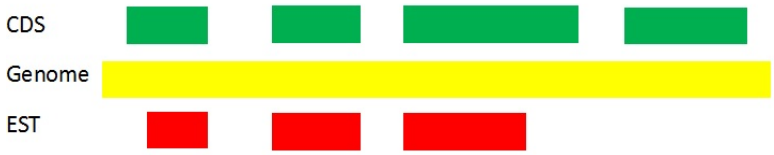

Figure 4.2: An overview of the final output.

4.1 Similarity searches using tblastx

The input to the first step are EST contigs from non-model organisms. Figure 4.3 shows an example EST contig that is the input to the pipeline. As EST contigs are not annotated with biological information, we need more information, such as the positions from where the exons start and end, to analyze the EST contig. The first step is adding information to the EST contig by performing a similarity search using tblastx [Altschul et al., 1990].

Initially, I compare an EST contig using tblastx against the NCBI nucleotide (nt) database to retrieve the most similar transcript from a chosen closely related species, which is specified by the user. In my thesis, I am using NCBI Entrez System [Maglott et al., 2010], the largest freely available database of any kind of genomic information for retrieving the orthologous CDS from the chosen model organism. The tblastx search translates the nucleotide query sequence in all six reading frames and compares those translated sequences to the translated database sequences. Identifying the correct reading frame helps to identify the correct intron-exon junction in the EST contigs.

This step is time consuming as searching for similar genes to the EST contig in

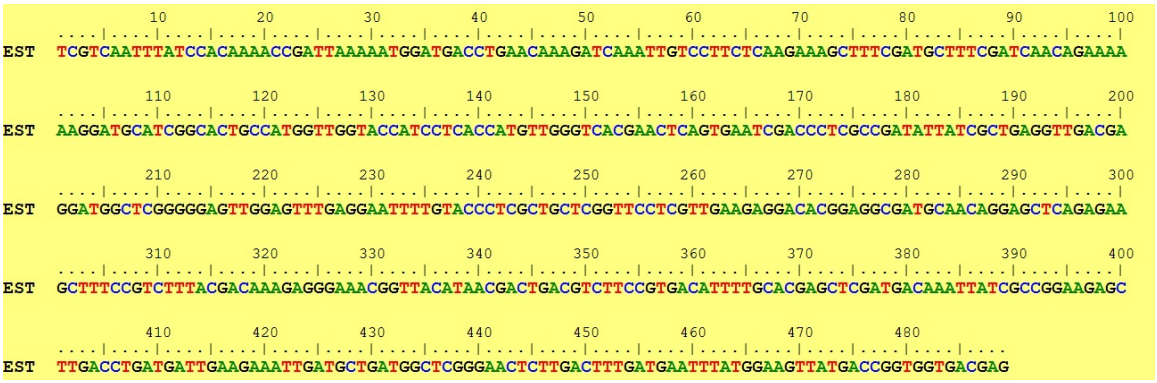

Figure 4.3: An example input of an EST contig to the pipeline shown using BioEdit [Hall, 1999].

the nucleotide database needs to handle a huge number of transcripts from the NCBI server. The more EST contigs that are given, the more time they will take to complete the similarity searches. To prevent the paralogs, I discard those EST contigs which are not suitable (based on different criterias described in later sections) to work with in the next step of the pipeline. My approach (from input to final output) takes approximately 5-15 minutes per sequence depending on the size of the CDS and the chromosome. After this step, for each EST contig, we now have an associated CDS to use in the next step of the pipeline.

4.2 Filtering the data

Coding DNA sequences identified by the NCBI tblastx need to be filtered to discard any sequences that are not suitable as input in this step of the pipeline. I filter the outputs from tblastx based on e values using python scripts. In my search, I restrict e values to an arbitrary maximum value of 1e-25. This step attempts to

remove highly variable genes, paralogs, or genes with no known homolog in the model organism and helps to reduce run time. As a result, it reduces the number of ESTs to be mapped against the genome and improves the quality of the result.

There is another option in the pipeline to examine the 'PREDICTED' keyword in similarity searches. Here, 'PREDICTED' means that it is a putative ortholog based on similarity searches with an annotated genome. Figure 4.4 shows a partial output of this step under different conditions. It also shows a sample snapshot of both kinds of mRNA sequences. This option can be used to filter any of these two kind of mRNAs.

4.2.1 Finding the exact reading frame for the chromosome

There are several features for a gene in the NCBI database. I use two important features to get the exact chromosome number to compare with the CDS. One is the 'chromosome' and another is the 'linkage group' (as shown in Figure 4.5). Either of these two features directly tell us that the CDS comes from a certain chromosome file. Specifically, a complete genome sequence is typically comprised of several chromosome files and the combination of all those files make a full genome sequence.

Figure 4.5 shows different features of a gene. We can see that LG12 is listed as the linkage group. That means I will compare the CDS of this gene ID to the extracted LG12 chromosome of the model organism, in this case *Apis mellifera*. As the chromosome contains both introns and exons of the genome, I am using chromosomes to map both the EST contigs and the CDSs. Instead of working with the whole genome sequence, I am working only on the mapped region from the specific chromosome

```
>gi|58585245|ref|NM_001011651.1| Apis mellifera troponin C type IIIa (TpnCIIIa), mRNA
ATGGATGATCTGACCAAGGATCAAATTGCTCTTTTGAAGAAAGCGTTCGACGCCTTCGATCATGACAAAA
AGGGTAGCATCGGTACCGACATGGTGGGCACGATATTGACCATGTTGGGTTACGAGCTCAGCGAGAAGAC
GTTGAAAGAGATCATCACTGAAGTTGATGAAGATGGATCCGGCCAATTAGAATTCGAAGAGTTCTGCACC
CTGGCCGCCCCGATTCTTAGTGGAGGAAGATTCGGAGGCTATGCAGCAGGAATTACGCGAGGCATTCCGAT
TGTACGACAAGGAAGGGAACGGCTACATAACCACTGCAGTATTCCGCGACATTCTTCACGAATTGGACGA
CAAATTGACGCCGCAAGAGTTGGACATGATGATAGAGGAAATTGATGCCGACGGTTCTGGAACGCTCGAC
TTCGACGAATTTATGGAAGTTATGACCGGTGGTGACGATTAA

Campoletis_sonorensis_5
>gi|328792733|ref|XM_003251720.1| PREDICTED: Apis mellifera hypothetical protein LOC551381,
 transcript variant 1 (LOC551381), mRNA
AGATTTATATTATTGAACGTTTTATTTTATCAATCTTCTCTTTAATACTGTCCAAAAATCTACGTATTAA
GCATTAAAAGTTGAATAATAAAAAATAAAAATGTTACGACATCTTTTCCAATCAATTACTCGAAACGCAA
AAAGTAGTATGCGATCTTATCATGCTAATAAAATTCCTGATAATGTTAAACCACCAACTATGGATGAAGT
GCTTGTACCATGTGGCTCGTGGAAGGAAGCAAATGCTAAAGCTCGAACTAAATATAATCTACAATTTGTA
GCTGGTGTTGTTATACTTGCTGCAACTATAGCATATGGTAGAATAACTGGCGTACTTTGGCTTAACTTTC
TTCCACCAACTCCAAAAGATAAAGATTCTGAATAAAATGAATTCAAATATAGTATTATAGTACATATAGC
AAATATAATTATTGTATAATTATTTTATTGTAAAAATAATAAAATATTTCTATTTATAGTATATTTAATT
TTTATTATATATATATATATAAGAAAAAGGGAAATGTTTTTTGATGATTTAAAATATATATTTTATTTTG
TTTACATTTTTGTATTTAAAATTGTTTTAATGCGTTTTGAAAAATATTGAAAAATTATTAAAAAATAAAA
ATACATAAAATATTTGAATTAAATTTCATTATAATTTTATTAATATTAATTTCATTTCATAAATAAATTC
ATTATGTTCTTATTTATCAAAAAGATTCAAATTTAACAAAAGTTCAAATGATATAAAAATTAAATATTTA
CATTTTTATTTACAGAAAAATTATTATTCTGCTAATATGTTATTTAAATTTAATAGATATAATTGTTTAT
TTCACATTTTTTAAATTATCAAATTTTTTAATGAGATCGAGCTATTTCTGAGAAAAATTCTAAAATATCT
TGTATGGTAAATTCAAGAGTTACCATACTATCTGGATAGCAT
```

Figure 4.4: A sample snapshot of two different mRNA sequences during the refining process. The upper one shows a well-established mRNA without the 'PREDICTED' keyword and the lower one shows a mRNA with the 'PREDICTED' keyword.

file, which saves both time and complexity. Note that, the feature "CDS", below the qualifier "join" in Figure 4.5 shows the exon positions in the mRNA, but there are many cases where there is no information available on the exons in NCBI database. For these reasons, exon information from the gene features is not considered in the pipeline.

I use a local tblastx search using the CDS to find the direction of the transcription (forward or reverse complement) of the chromosome. To make a local search, BLAST must be installed on a local machine and a database must be created on the machine from the fasta files we want to compare to the CDSs. In this local BLAST search, the database is the chromosomes (a set of fasta files) of the reference genome sequence (e.g., *Apis mellifera*). As searching using BLAST can be computationally intensive and huge traffic frequently occurs on the NCBI online search, I prefer the local search to reduce the run time in this step. The output from the BLAST search gives the

```
FEATURES             Location/Qualifiers
     source          1..1097
                     /organism="Apis mellifera"
                     /mol_type="genomic DNA"
                     /strain="DH4"
                     /db_xref="taxon:7460"
                     /linkage_group="LG12"
     gene            <1..1097
                     /gene="TpnCIIIa"
                     /note="Derived by automated computational analysis using
                     gene prediction method: BestRefseq."
                     /db_xref="BEEBASE:GB13594"
                     /db_xref="GeneID:408379"
     mRNA            join(<1..29,105..248,441..692,1063..1097)
                     /gene="TpnCIIIa"
                     /product="troponin C type IIIa"
                     /exception="unclassified transcription discrepancy"
                     /note="Derived by automated computational analysis using
                     gene prediction method: BestRefseq."
                     /transcript_id="NM_001011651.1"
                     /db_xref="GI:58585245"
                     /db_xref="BEEBASE:GB13594"
                     /db_xref="GeneID:408379"
     CDS             join(<1..29,105..248,441..692,1063..1097)
                     /gene="TpnCIIIa"
```

Figure 4.5: Different features for a gene ID. The feature "source" shows the length, name of the organism it is coming from, and chromosome number of the reference genome (e.g., linkage_group). Feature "gene" provides the reference database and gene ID. Next, the feature "mRNA" provides information on the location of the exons, the transcript id and some other attributes. Lastly, "CDS" shows the length of the exons and gene name.

value of the reading frame which tells whether the pipeline should use forward or reverse complement of the chromosome with est2genome in the next stage of the pipeline.

4.3 Using est2genome to find homologous regions

In this step, the inputs are the chromosomes of the reference genome sequence and the CDSs from the previous step. Initially, I collect the CDSs from the NCBI database using the gene ID found in the header of the mRNA sequence from the tblastx search in the previous step. To get the CDS, I use Entrez efetch tool from Biopython [Cock et al., 2009]. The CDS is the source where all the exons reside side by side. Also, the

CDS of the model organism helps to identify the homologous region in the reference chromosome sequence. So, the CDSs along with the extracted chromosome sequences facilitate identification of the intron positions and the intron-exon boundaries in the EST contigs.

I tried several tools to align the CDS to the homologous genome sequence but those tools generated unacceptable results. In most of the cases, they produced poor alignments, matching exon to intron and vice versa. After a few unsuccessful attempts with the tools that are discussed in Chapter 3, I found est2genome [Rice et al., 2000], which produced better alignments compared to the other tools. est2genome [Rice et al., 2000] is a tool that can align EST contigs to DNA sequences using sequence similarity. The program aligns transcripts (EST contigs and mRNAs from similarity searches with EST contigs) to a genomic DNA sequence. It also inserts gaps of variable length into the spliced mRNA when needed. The insertion of variable length gaps in the alignment prevents est2genome from predicting accurate intron-exon boundaries of the EST contig.

Figure 4.6 shows the output from est2genome [Rice et al., 2000]. In Figure 4.6, the identified introns and exons do not help to identify the intron-exon boundary properly because they are the approximate region and are incorrect compared to the true intron-exon boundaries that are known. The information from the line labeled "Span" provides the region of the chromosome that has the highest similarity to the CDS. This is the predicted gene region, including introns and exons. I extract this sequence region from the reference genome sequence.

```
1  Note Best alignment is between forward est and forward genome,
2  and splice sites imply forward gene
3
4  Exon        48 100.0 1453466 1453513 CM000066.5       1    48
5  +Intron    -20   0.0 1453514 1454459 CM000066.5
6  Exon        76 100.0 1454460 1454535 CM000066.5      49   124
7  +Intron    -20   0.0 1454536 1454618 CM000066.5
8  Exon        92 100.0 1454619 1454710 CM000066.5     125   216
9  +Intron    -20   0.0 1454711 1454816 CM000066.5
10 Exon       103 100.0 1454817 1454919 CM000066.5     217   319
11 +Intron    -20   0.0 1454920 1455058 CM000066.5
12 Exon        80 100.0 1455059 1455138 CM000066.5     320   399
13
14 Span       319 100.0 1453466 1455138 CM000066.5       1   399
15
16 Segment     48 100.0 1453466 1453513 CM000066.5       1    48
17 Segment     76 100.0 1454460 1454535 CM000066.5      49   124
18 Segment     92 100.0 1454619 1454710 CM000066.5     125   216
19 Segment    103 100.0 1454817 1454919 CM000066.5     217   319
20 Segment     80 100.0 1455059 1455138 CM000066.5     320   399
21
```

Figure 4.6: An output file generated from est2genome [Rice et al., 2000] that shows possible exon and intron positions of the *Apis mellifera* mRNA and *Apis mellifera* genome in between the line numbers 4 to 12. Line 14 shows the whole region where all the exon and intron reside. From line 16 to 20, the file shows only the exons in both of the sequences.

When both the CDS and the exact chromosome number are identified, both are provided as inputs to est2genome [Rice et al., 2000]. After running est2genome, I discard the EST contig if the length of the extracted chromosome from est2genome output is more than 10,000bp, as most of the EST contigs in the datasets I use are typically 300-600bp in length. When a CDS whose length is several times shorter than the extracted chromosome region, est2genome can not produce a probable region from the chromosome sequence that can be used in the pipeline. Also, too many gaps in the alignment that would be done in the next step of the pipeline decrease the similarity score of the alignment. So, to avoid excessively reducing the gap extension cost in next step of the pipeline, the extracted reference genome sequences having a length of more than 10K bp can be discarded prior to mapping the CDS to the reference genome sequence. From the est2genome output, I obtain the region of the reference

genome sequence that contains all introns and exons to work with in the next step of the pipeline.

4.4 Aligning the CDS with the extracted chromosome sequence

To find the intron-exon boundary in the EST contigs, I used the CDS as an auxiliary sequence to map with the chromosome region obtained in the previous step. Each CDS was mapped against the reference genome sequence. Figure 4.7 shows how exons from the CDS should be mapped to the exons of the reference genome sequence.

In my thesis, I had to design a new algorithm that can identify the exons in a nucleotide sequence of a chromosome. Figure 4.8 shows a small portion of a CDS and a reference genome sequence to demonstrate how Algorithm 1 (described below) works.

I consider the first few amino acids from the translated CDS and try to find them in the translated chromosome sequence (Algorithm 1). As there might be some exceptions at the starting position of an exon in the extracted reference genome sequence due to phase variation, I avoid searching for the first character from the translated sequence. For example, in a sequence like Figure 4.8, I start from the translated CDS position 2-4 (VTT) instead of positions 1-3 (VVT) as phase variation may affect the first amino acid position in downstream exons in the extracted chromosome sequence. If I find the sequence in the translated version of the extracted reference genome se-

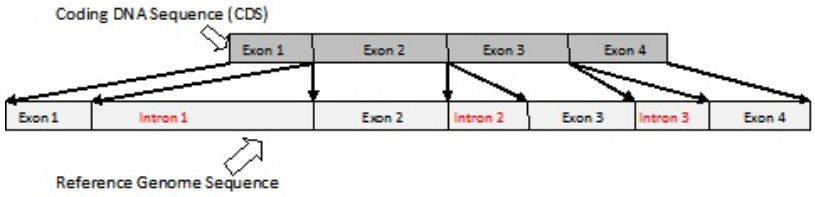

Figure 4.7: Figure showing a small portion of a CDS and a reference genome sequence. The figure also demonstrates how the exons from CDS map to the exons in reference genome sequence.

quence, then I use the common prefix algorithm. The common prefix algorithm finds the largest common prefixes from both sequences. The combination of search and common prefix algorithm makes it possible to find the exon boundaries in the CDS and intron-exon boundaries in the chromosome sequence.

In Algorithm 1, my hypothesis is that a search for 4 amino acids, which is equivalent to 12 nucleotides, is sufficient. This length of nucleotides is a good length to find the starting position of an exon in the reference genome sequence as more than that could miss a smaller exon while searching. In Algorithm 1, the find method in line 13 (see below) returns the start position where the match is found. If four amino acids from the prefix of the CDS are found in the reference genome sequence, then I use the longest common prefix algorithm on the CDS and the reference genome, starting from the position where the match was found. The common prefix algorithm returns the longest match between the two sequences. On the other hand, if it is not found, then the same process continues for all three forward reading frames in the reference

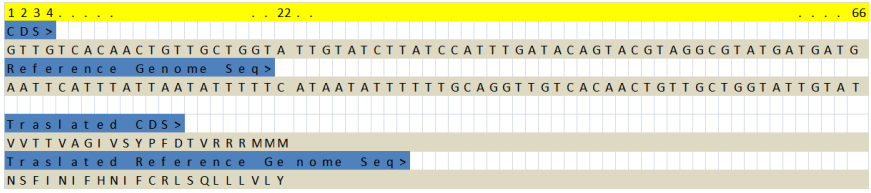

Figure 4.8: Figure showing a small portion of a CDS and a reference genome sequence (above) and their translated sequences (below).

genome sequence. For instance, in Figure 4.8, the first few amino acids in the CDS are not found in the translated reference genome sequence in the current reading frame. So, we need a shift in the reading frame of the reference genome sequence. If we look at Figure 4.9, the nucleotide sequence of reference genome does not start at the correct codon position for the exon. When the shifted nucleotide sequence is translated, four amino acids from the prefix of the CDS are found in the translated reference genome sequence. Next, the longest prefix algorithm is applied to both of the sequences and they are translated back into nucleotide sequences. Recursive calling continues until the last exon from the CDS is found in the reference genome sequence. The portions in between the exons in the chromosome are the introns and we can skip the introns to align the exons from the reference genome sequence to the CDS.

I append all the matches in the chromosome together, starting from the position of the start of the match in the extracted chromosome sequence. If there is a single reading frame shift (+1) in subsequent exons, then I collect the sequence starting from the position of the frame shift variable to ((start position of the match in chromosome sequence + length(result from common prefix)) * 3) as a nucleotide sequence. In Algorithm 1, I also track the chromosome and CDS sequence using two separate

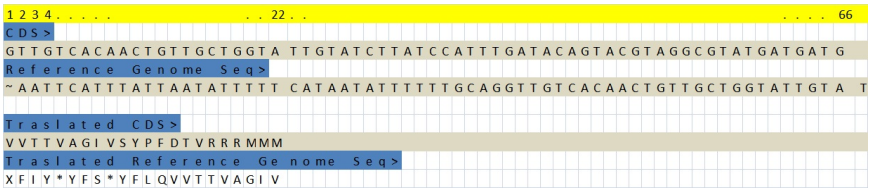

Figure 4.9: Figure showing a single shift in the reading frame of the CDS.

counters. The counters follow the rules below:

$$tracker\_cds+ = ((\text{length of result from common prefix algorithm in the translated CDS and translated the chromosome sequence}) * 3) + 3 \quad (4.1)$$

$$tracker\_chromosome+ = ((\text{starting position of match in the chromosome sequence}+ (\text{length of result from common prefix algorithm among the translated CDS and the translated chromosome sequence} * 3))) + \text{value of reading frame shift variable} \quad (4.2)$$

I continue until *tracker_cds has* completely traversed the length of CDS.

Then, I need to make sure that the EST contigs and the chromosomes do not contain any stop codons at the very beginning of the sequence. To make sure, in Algorithm 2, I translate the EST contig and I search for a stop codon in the first few amino acids from the beginning as the EST contigs sometimes contain stop codons if the reading frame is not correct. In particular, I use the first 20 amino acids in this case. I have to do the search for stop codons in all the three different reading frames. If there is a stop codon, then I change the reading frame with a shift and consider the EST contig from the next reading frame.

Algorithm 1 Exon finder from CDS and Reference genome sequence

```
procedure EXON_MAPPER(newCDS, newChromosome)
    temp_cds ← newCDS.translate()
    temp_protein ← newChromosome.translate()
    frameshift ← 0
    while frameshift<3 do
        temp_protein ← newChromosome[frameshift :].translate()
        max_match ← temp_protein.find(temp_cds[1 : 5]) // skip first amino acid, look for match of length 4
        if max_match >= 0 then
            result ← commonprefix([temp_cds[1 :], temp_protein[max_match :]])
            chromosome+ = temp_protein[frameshift : (max_match + len(result)) * 3 + frameshift]
        else
            //we did not find a match
            chromosome+ = newChromosome[frameshift :]
        CDS+ = newCDS[0 : (len(result) + 1) * 3]
        tracker_cds+ = (len(result) * 3) + 3
        tracker_chromosome+ = ((max_match + len(result)) * 3) + frameshift
        frameshift+ = 1
    if tracker_cds < len(temp_cds) − 9 then:
        //if the condition is met above, the procedure will be called recursively
        EXON_MAPPER(inputCDS[tracker_cds :], inputChromosome[tracker_chromosome :]) // recursively
find next exon and append it as string
    else
        CDS+ = inputCDS[tracker_cds :]
```

Algorithm 2 Defined boundary aligner

```
procedure PROFILE_ALIGNER(EST)
    workable_EST ← ""
    temp_ESTs ← []
    frameshift ← 0
    while frameshift<3 do
        EST_prot = EST[frameshift :].translate()
        temp_ESTs.append(max(EST_prot.split('*')), key = len)
        frameshift+ = 1
    i ← 0
    for i < 3 do
        if max(temp_ESTs, key = len) == temp_ESTs[i] then
            workable_EST = temp_ESTs[i :]
    return workable_EST
```

Algorithm 3 Top level part of the pipeline

```
//inputCDS = input Coding DNA Sequence (CDS)
//inputChromosome = input extracted homologous Chromosome sequence
tracker_cds ← 0 //tracks the CDS position
tracker_chromosome ← 0 //tracks the chromosome sequence position
CDS ← "" //empty string to hold the complete CDS after all the iterations
chromosome ← "" //empty string to hold the chromosome region after all the iterations
newCDS ← inputCDS //assigning the input CDS to a temporary variable
newChromosome ← inputChromosome //assigning the input chromosome to a temporary variable
//EST = input EST
EXON_MAPPER(inputCDS, inputChromosome)
workable_EST ←PROFILE_ALIGNER(EST)
Call MAFFT with CDS, chromosome, and workable_EST
```

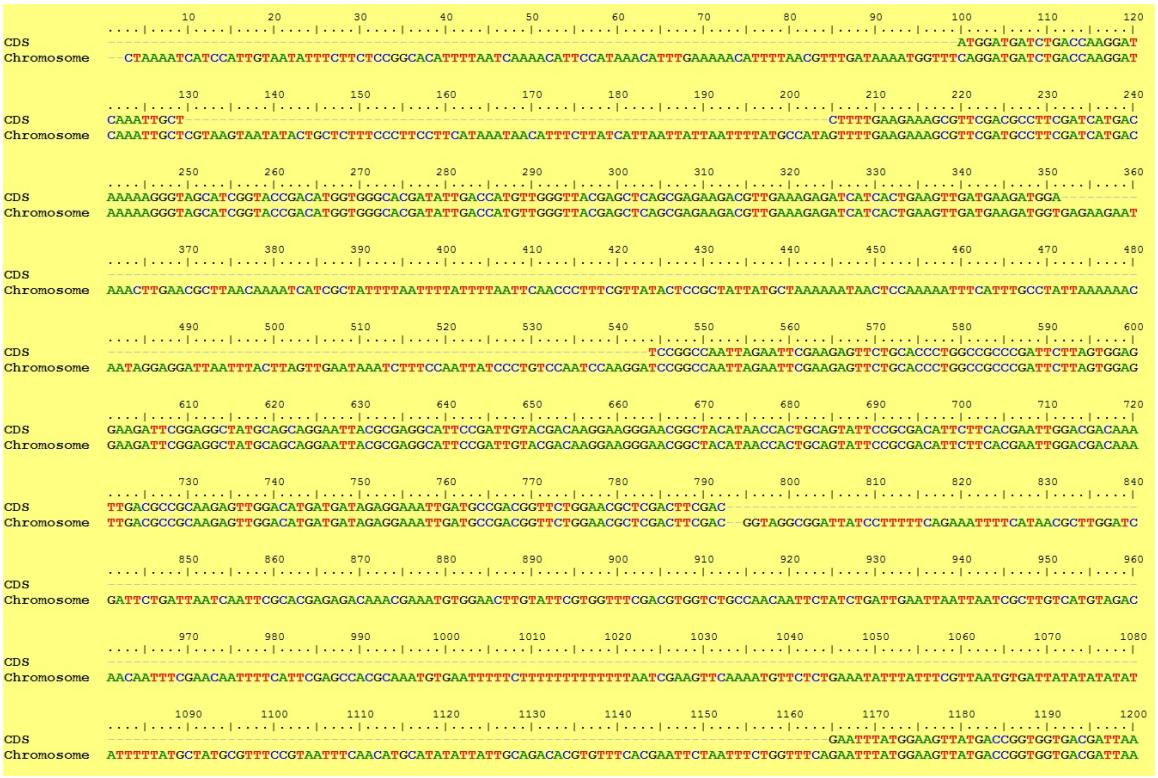

Figure 4.10: An example result showing an alignment between a CDS and the reference genome sequence in nucleotide format using the pipeline (alignment depicted using BioEdit [Hall, 1999]). In the first intron, phase is not adjusted, hence why the C does not exactly match here. It would be on the other side, so the intron starts with GT and ends with AG.

Algorithm 3 is a top level part of the implementation that calls Algorithm 1 and 2. This step results in an alignment of a CDS and the partial chromosome sequence. Figure 4.10 shows an arbitrary output of this step. This step also provides the EST contig with the correct reading frame to be used in the next step of the pipeline.

4.5 Aligning the EST contig with defined boundaries

Finally, to get the correct intron-exon mapping, I need to align all the EST contigs, their associated CDSs, and the extracted chromosome sequences from the reference genome sequence. To make the alignment, I use the add alignment feature from MAFFT [Katoh et al., 2009], which is a multiple sequence aligner.

A profile is a multiple sequence alignment where sequences are more sensitive than a single sequence because they influence the information of all the sequences in the alignment. Rychlewski et al. [2000] also showed that profile alignments are more accurate when trying to establish relationships between distantly related sequences. Both the speed and accuracy of the alignment are better in MAFFT than the widely used alignment tools CLUSTALW [Thompson et al., 1994] and T-Coffee [Notredame et al., 2000]. The first profile is comprised of the CDSs and chromosome sequences obtained from Algorithm 2 as in Figure 4.10 and then the EST contig is added to the the existing profile. The final alignment (shown in Figure 4.11 and Figure 4.12) of adding the EST contig is highly influenced by the existing alignment of the CDS and chromosome sequence. Algorithm 2 puts gaps in the CDS where the introns were predicted based on the alignment with the genome sequence. Thus, the number of gaps in the CDS for each intron equates to the length of the intron in the genome sequence. As the intron-exon boundary has been identified exactly in the CDSs, the aligned CDSs and reference genome sequences should give a more accurate multiple sequence alignment when combined with the EST contigs.

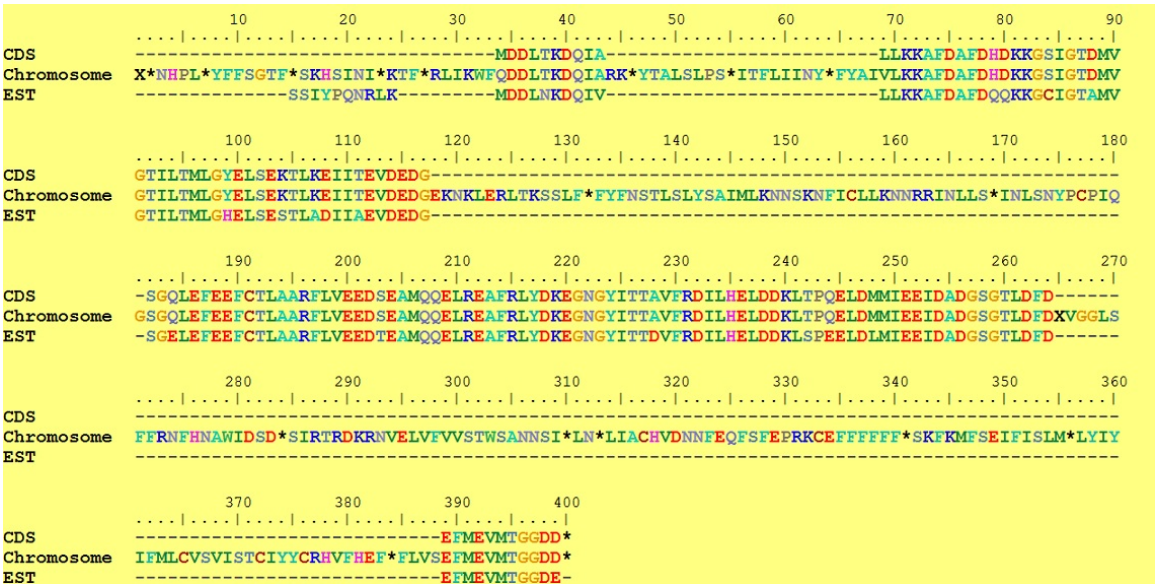

Figure 4.11: An example result showing the alignment among the EST contig, the CDS, and the reference genome sequence in amino acid format using the pipeline (alignment depicted using BioEdit [Hall, 1999]). Stop codons are depicted by an asterisk and ambiguous codons are depicted by an X. In this amino acid multiple sequence alignment, the first exon is depicted in position 34. However, the CDS and chromosome are not exact matches at position 34 as the start codon is actually a short (one codon) first exon that should be positioned upstream. As the first exon is misaligned, the first intron is also missing. The second should begin at position 35 and correctly ends at position 43. The remainder of the alignment accurately predicts all further exons and introns.

Finally, the MAFFT alignment gives the final output of the pipeline, which is a multiple sequence alignment of the chromosome of the reference genome, the most closely related CDS to the given EST contig from the model organism, and the EST contig from a non-model organism in the form of amino acids. Then, I have to trace back the amino acids and gaps in the alignment using a tracker that is similar to the tracker_cds and tracker_chromosome (described in previous section) to show the alignment in nucleotide format. Figure 4.11 shows the output of the pipeline in amino acid format and Figure 4.12 shows the final alignment in nucleotide format.

Figure 4.12: An example result showing the alignment among the EST contig, the CDS, and the reference genome sequence in nucleotide acid (the same gene alignment as depicted in Figure 4.11) format using the pipeline (alignment depicted using BioEdit [Hall, 1999]). In this nucleotide multiple sequence alignment, the first exon is located from position 100 to 103, but it should be placed upstream. The CAG in position 100 to 103 of the chromosome sequence is actually the end of the first intron. Thus, the first exon was not accurately predicted due to its short length and the first intron is missing. As it is a short starting exon, the first intron is also missing here. The next exon starts immediately from position 104 and ends at position 129. The following intron starts from position 131 and continues till position 205 including the phase variation. Similarly, the last intron of this alignment starts from position 796 with GT and ends at position 1165 including the phase variation. Finally, the alignment ends with the exon that spans from position 1168 to 1200.

In Figure 4.11, the EST contig has been added to the already aligned CDS and chromosome from the reference genome by the profile alignment. The first exon

starts at position 34 in Figure 4.11 and continues till position 43. Next, from position 44, the first intron starts and it ends at position 68. Continuing in this way, the alignment shows that all the exons are aligned with the chromosome. The amino acid fragments in the CDS and EST contig are the exons and after the first exon, the gaps until the next exon show the intron regions in the CDS and predicted intron positions in the EST contig. The X symbol can be seen at positions 0 and 265, which represents the ambiguous amino acid. These symbols are shown due to corrections for changes to the reading frame. The initial alignment of the CDS and the chromosome helps to constrain the alignment of the EST contig to predict the intron and exon regions for that EST contig. Next, using three different trackers the same alignment can be tracked back to a nucleotide sequence alignment that is shown in Figure 4.12. In Figure 4.12, we can see that there are 2 gaps at positions 0-1 in the reading frame of the chromosome, which was the reason for representing the first codon as an X in Figure 4.11. The alignment shown in nucleotide format can also easily be translated back to amino acid format using any popular sequence alignment tool, such as BioEdit.

Chapter 5

Evaluation

I evaluated my pipeline using three different datasets. Dataset 1 includes a collection of 52 aligned genes (including introns and exons) of *Apis mellifera* and *Bombus terrestris* from Sharanowski and Domaratzki [in prep]; gene selection was based on the phylogenetic dataset of Sharanowski et al. [2010]. Dataset 2 and 3 were two different collections of EST contigs of non-model organisms with different evolutionary distances from the model organism chosen in this study, the honey bee, *Apis mellifera* (Insecta: Hymenoptera: Apidae). Dataset 2 was a small set of 767 EST contigs from *Campoletis sonorensis* (Insecta: Hymenoptera: Ichneumonidae) from Sharanowski et al. [2010]. *Campoletis sonorensis* is in the family Ichneumonidae, which likely diverged from Apidae, the family of *Apis mellifera*, between 150-200mya [Grimaldi and Engel, 2005]. Dataset 3 was a larger collection of 13,333 ESTs from the bumble bee *Bombus terrestris* (Insecta: Hymenoptera: Apidae) from Sadd et al. [2010]. *Bombus* and *Apis* are both members of the family Apidae, but likely diverged from each other

80-100mya [Hines et al., 2007].

Analysis of the results from the pipeline for Dataset 1 demonstrates the accuracy and usability of the pipeline, as the pipeline output is directly compared to previously aligned genomic sequences. Dataset 2 and 3 are used to further demonstrate the usability of the pipeline to predict the intron-exon boundaries for distantly related species compared to the reference genome *Apis mellifera*. For these two highly divergent non-model datasets (*Campoletis sonorensis* and *Bombus terrestris*), previously aligned genomic sequences are not available. As a result, direct gene by gene comparisons for determining accuracy are not possible. However, different filtering steps in the pipeline (described in Section 5.1) show both Dataset 2 and 3 have similar characteristics.

In Section 5.1, I will describe the evaluation of the pipeline for the dataset of *Apis mellifera* and *Bombus terrestris* EST conigs from Sharanowski and Domaratzki [in prep]. In Section 5.1, I will also describe the result produced from the pipeline using the dataset of *Campoletis sonorensis* EST contigs from Sharanowski et al. [2010] and the dataset of ESTs of bumble bee *Bombus terrestris* from Sadd et al. [2010]

5.1 Results

Dataset 1 includes alignments of 52 genes (including introns and exons) from *Apis mellifera* and *Bombus terrestris* species. The intron and exon positions of those genes were located by Sharanowski and Domaratzki [in prep] by aligning genome scaffolds

for specific genes across species and directly comparing the alignments with gene maps in Entrez Gene. Thus, dataset 1 is a set of 52 genes with the exact exon and intron locations mapped between *Bombus terrestris* and *Apis mellifera*. In this case, I acquired the transcripts for those 52 genes from *Bombus terrestris* by blasting the *Bombus* gene sequence (including introns) to obtain the *Bombus* CDS and passed those transcripts to the pipeline. I am using the transcripts as input to mimic how EST contig datasets will function in the pipeline. Note that these *Bombus* sequences are not EST contigs, but are downloaded CDS sequences from NCBI. To mimic how EST contigs function in the pipeline, I am using *Apis mellifera* as the reference genome for mapping with the transcripts from *Bombus terrestris* as *Apis mellifera* is a closely related genome to *Bombus terrestris*.

Figure 5.1 shows a diagram of the number of correctly predicted exons and the total number of actual exons present in these 52 genes for *Bombus terrestris*. On the other hand, Figure 5.2 shows a diagram of the number of correctly predicted introns and the total number of actual introns present in these 52 genes for *Bombus terrestris*. Correctly predicted introns or exons mean that their positions (both start and end) are correctly identified by the pipeline.

There were a total of 203 exons and 147 introns in dataset 1 for *Bombus terrestris*. The pipeline was capable of predicting 187 exons (accuracy of 92.12%) and 132 introns (accuracy of 89.79%) correctly. See Section 2.3.1 for further explanation about accuracy. The accuracy would be higher if the starting exons were all more than 3 amino acids long. If we analyze the results, among the 16 exons that the pipeline could not predict, 8 genes (CG3997, CG7424, CG7434, CG11981, CG15442, CG5502,

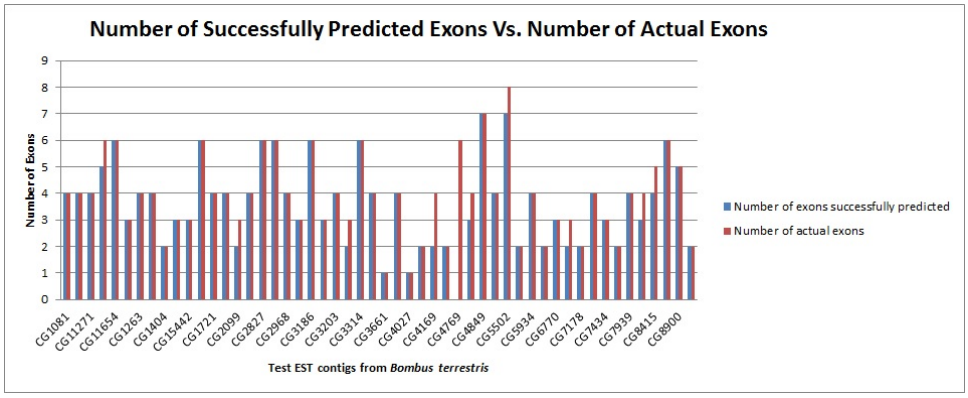

Figure 5.1: The data used in this graph was based on 52 transcripts from *Bombus terrestris*. It shows the number of correctly predicted exons and the number of true exons in *Bombus terrestris* gene.

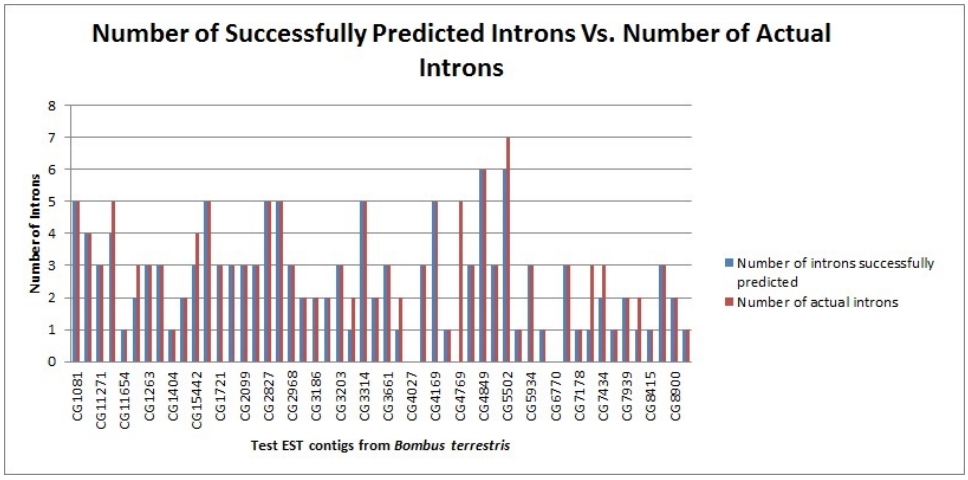

Figure 5.2: The data used in this graph was based on 52 transcripts from *Bombus terrestris*. It shows the number of correctly predicted introns and the number of true introns in *Bombus terrestris* gene.

CG11276, CG32662) had a short first exon (1-3 amino acids in length). At the same time, those short exons all have introns following them and the pipeline could not predict any of them successfully. If we do not consider the shorter exons (less than 3 amino acids), the accuracy would be 96.06% for exon identification and 95.23% accuracy for intron identification.

The sensitivity of the pipeline is 92.12% for predicting exons. The pipeline does not predict any false negatives and thus the specificity of the pipeline is 100% for predicting exons. The specificity of the pipeline for predicting introns is also 100%.

Dataset 2 consisted of a total of 767 *Campoletis sonorensis* EST contigs that were taken from Sharanowski et al. [2010]. Figure 5.3 shows the filtering results for the dataset in the pipeline. At the very beginning, there are total of 767 *Campoletis sonorensis* EST contigs. Among those 767 EST contigs, only 234 ESTs have a match with *Apis mellifera* based on the tblastx search described in Section 4.1. Those 234 matches were further filtered down to 201 (26.21%) contigs based on those matches with an expect value below 1e-25, as described in Section 4.2.

Those 201 EST contigs were analyzed with the pipeline to predict intron-exon boundaries. When I passed all of those 201 EST contigs to the pipeline, it filtered out 31 based on the minimum length criteria of the extracted chromosome from the est2genome output (see Section 4.3). Thus, after filtering, 170 *Campoletis sonorensis* EST contigs were left for intron-exon boundary prediction.

Next, I obtained the ESTs of dataset 3 from supplementary files provided by Sadd et al. [2010]. All of the ESTs are from the bumble bee *Bombus terrestris*. A total of 13,333 unique EST sequences were obtained and used as input for the pipeline. I am only considering the *Apis mellifera* genome to map the ESTs. As both *Bombus terrestris* and *Apis mellifera* are from the same family (Apidae), *Apis mellifera* has higher similarity with *Bombus terrestris* than *Campoletis sonorensis*. Among the 13,333 unique EST sequences, only 6,275 ESTs had a match in *Apis mellifera*. After

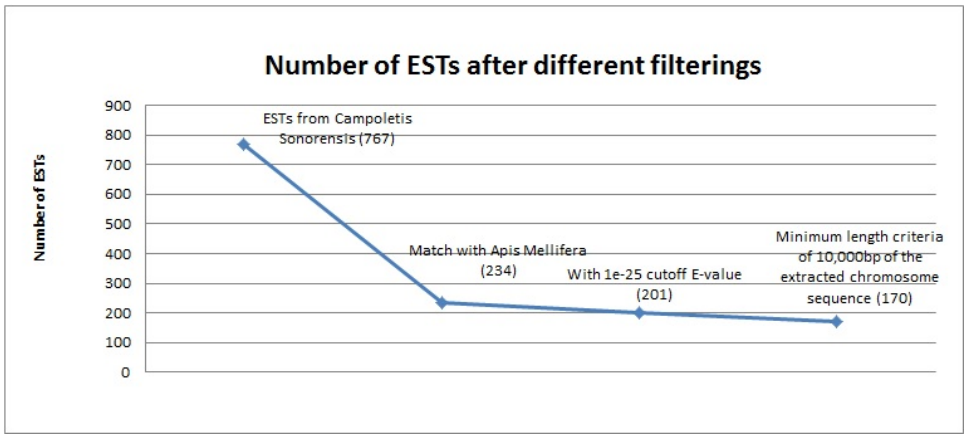

Figure 5.3: The graph shows how *Campoletis sonorensis* EST contigs are filtered in different stages before passed to the next stage in the pipeline. Based on the tblastx search, there are only 234 EST contigs that have a match with *Apis mellifera*. Of those 234 blast hits, only 201 had e-values at e-25 or lower and only 170 of those met the minimum length criteria when compared to the extracted genome sequence.

filtering those 6,275 ESTs using an e value cutoff of 1e-25, only 3,366 (25.29% of the total 13,333 ESTs) were considered suitable for the next step of analysis in the pipeline. After filtering for length, 2,714 ESTs remained for intron-exon prediction (Figure 5.4).

Both datasets 2 and 3 were similar in the number of ESTs that were filtered out using each of the three filtering criteria (matches to *Apis mellifera*, e-value cut-off, and minimum length of the extracted chromosome) (Figures 5.3 and 5.4). In both datasets, approximately 50% or less of the ESTs had matches in *Apis mellifera* and approximately 50% of the ESTs with matches had appropriate e-values. Overall, 74.71% of the total 13,333 *Bombus terretris* ESTs and 73.79% of the total 767 *Campoletis sonorensis* ESTs were discarded due to a lack of gene match with *Apis mellifera*, the e value cutoff of 1e-25 for those matches. Similarly, about 4-5% of the total ESTs were further discarded after the minimum length filtering criterion.

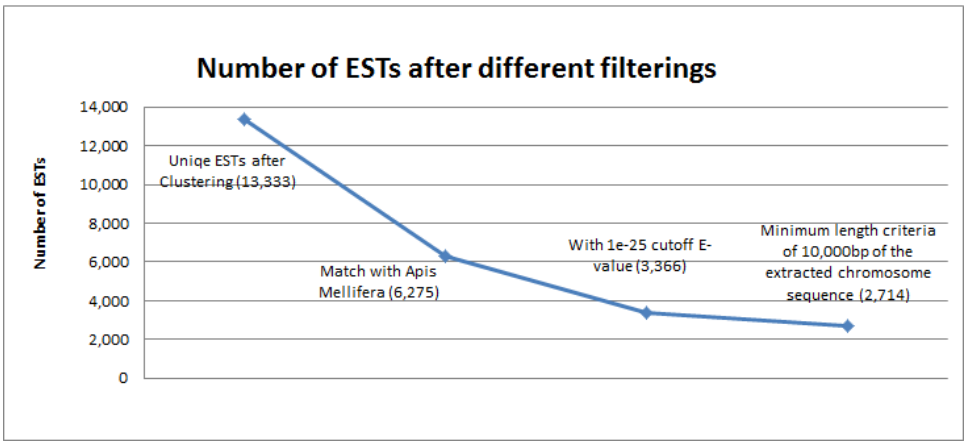

Figure 5.4: The graph shows how *Bombus terrestris* Sadd et al. [2010] ESTs are filtered in different stages before passed to the next stage in the pipeline. Based on the tblastx search, there are only 6,275 ESTs out of 13,333 ESTs with a match to *Apis mellifera*. Of those 6,275 blast hits, only 3,366 had e-values at e-25 or lower and only 2,714 of those met the minimum length criteria when compared to the extracted genome sequence.

While we can not determine whether the splice sites in the *Campoletis sonorensis* EST contigs are accurately predicted, we can determine whether the alignments indicate that the EST contigs are split along sites that begin with GT and end with AG in the reference genome chromosome. In the 170 *Campoletis sonorensis* EST contigs, there are in total 453 predicted intron junctions. Of these 453 predicted junctions, 438 start with GT and end with AG, accounting for phase variation. Thus, the accuracy of predicting intron junctions with donor and acceptor sites for dataset 2 is 96.70%. This accuracy is similar to the $> 90\%$ results for dataset 1, which suggests that the pipeline works similarly in dataset 2 compared to the results on known genes.

Similar to dataset 2, we again examined the predicted splice site junctions in dataset 3. In the 2,714 *Bombus terrestris* EST contigs, there are a total of 6,409 predicted intron junctions. Of these 6,409 predicted intron junctions, 5,851 introns

start with GT and end with AG, accounting for phase variation. So, the accuracy of predicting intron junctions with donor and acceptor sites for dataset 3 is 91.29%. This accuracy is also similar to the $> 90\%$ results for dataset 1, which also suggests that the pipeline works similarly in this dataset 3 compared to the results of known genes.

5.2 Discussion

In addition to short starting exons, the pipeline could not handle the issue of paralogs. If the algorithm finds a paralog as the working mRNA, the EST will not be aligned properly to the CDS and genome sequence. This incorrect alignment between a paralgous CDS and thus genome sequence with the EST degrades the performance of the pipeline. That's why to avoid paralogs in the pipeline, I am using a comparatively low e value of 1e-25. If I increase the e value (e.g., 1e-5), then more ESTs are available to pass to the pipeline but most of the ESTs can not be aligned as those ESTs may match to paralogous genes. If a paralogous CDS was chosen as a match for the EST contig, the CDS and the chromosome will still be aligned but the EST will be incorrectly aligned to the wrong gene. Thus, the accuracy calculations for datasets 2 and 3 based on the prediction of GT-AG splice site junctions does not account for incorect alignments of paralogous ESTs, and therefore is not a conservative measure of accuracy.

At this moment, the run time of the pipeline is five to fifteen minutes to process a single EST depending on the size of the extracted chromosome sequence. The

mapping between an EST and a genome using est2genome is the most time consuming part of the pipeline. If that part can be improved or can be replaced by a better approach, then the pipeline would be able to predict intron-exon junctions much faster than now. To reduce the run time, a parallel version of the pipeline can be developed that will increase the performance of the pipeline significantly.

Chapter 6

Conclusion

In my thesis, I showed that intron-exon junctions can be successfully predicted in non-model ESTs. To accomplish this, I used a closely related complete genome (e.g., *Apis mellifera*) and the most similar transcript from the same genome using the NCBI nucleotide database. In my thesis, I also designed two new algorithms to develop the pipeline for my thesis. The output of the thesis is a multiple sequence alignment (an example is shown in Figure 4.12) that shows the predicted intron-exon junctions of the non-model EST.

I evaluated my thesis by comparing the results from the pipeline to previously predicted alignments, as there are no other implementations to compare to that can identify intron positions for those non-model ESTs. Based on a comparison with 52 previously aligned genes (with introns and exons) of *Apis mellifera* and *Bombus terrestris*, the sensitivity of the pipeline is 92.12% and the specificity is 100% for the total number of exons in the aligned test genes. For introns, the sensitivity of the

pipeline is 89.79% and the specificity is 100% for the total number of introns in the test genes.

I also ran two other datasets (767 *Campoletis sonorensis* EST contigs and 13,333 *Bombus terrestris* ESTs) through the pipeline and the pipeline was able to predict intron-exon boundaries for 84.58% for the filtered (described in Section 4.2) *Campoletis sorensis* dataset and 80.63% for the filtered (described in Section 4.2) *Bombus terrestris* dataset. The accuracy of predicting intron junctions with donor and acceptor sites for dataset 2 is 96.70% and for dataset 3 is 91.29%.

The pipeline can be easily run from the command prompt in any Unix/Linux system. My thesis will be beneficial for biologists who work with non-model ESTs. Specifically, researchers who want to design primers for the exons of the non-model genes need to know the specific intron-exon junctions to determine if the length of the amplicon is suitable for Sanger sequencing. The pipeline will also save time and would increase the productivity and the accuracy compared to the manual alignment process. My thesis will also be beneficial for research on non-model organisms that are evolutionary distant from sequenced genomes.

6.1 Future Work

The pipeline is able to handle the aligned genes (including introns and exons) of *Apis mellifera* and *Bombus terrestris* from Sharanowski and Domaratzki [in prep]. For the two other datasets, the prediction can handle 84.58% and 80.63% of the

filtered datasets.

At this moment, the mapping between an EST and a genome using est2genome is the most time consuming part of the pipeline. Moreover, to reduce the run time (5-15 minutes per EST), a parallel version of the pipeline can be developed that will increase the performance of the pipeline significantly.

One more additional issue is the accuracy of the pipeline. It would be interesting to see if the accuracy could be increased. To increase the accuracy, a machine learning technique can be applied so that the starting short exons having length of 1-3 amino acids can be predicted. Also, a non machine learning approach like probabilistic statistical model can be built to solve the problem and improve the accuracy. Then, the sensitivity of the pipeline could reach 100%.

Bibliography

The maize full length cDNA project. `http://www.maizecdna.org/outreach/flcdna.html`, 2013.

M. D. Adams, M. Dubnick, J. D. Gocayne, J. M. Kelley, A. R. Kerlavage, and W. R. McCombie. Complementary dna sequencing: expressed sequence tags and human genome project. *Science*, pages 1651–1656, 1991.

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.

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

F. D. Bona, S. Ossowski, K. Schneeberger, and G. Ratsch. Optimal spliced alignments of short sequence reads. *Nucleic Acids Research*, 24:i174–i180, 2008. doi: 10.1093/bioinformatics/btn300.

N. Bray, I. Dubchak, and L. Pachter. AVID: a global alignment program. *Genome Research*, 13:97–102, 2003. doi: 10.1101/gr.789803.

M. Brudno, C. Do, G. Cooper, M. F. Kim, E. Davydov, E. D. Green, A. Sidow, and S. Batzoglou. LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic dna. *Genome Research*, 13(4):721–731, 2003.

P. J. A. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. L. de Hoon. Biopython: freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

L. J. Collins, P. J. Biggs, C. Voelckel, and S. Joly. An approach to transcriptome analysis of non-model organisms using short-read sequences. In *Proceedings of the 19th International Conference on Genome Informatics (GIW'08), Gold Coast, Australia*, pages 3–14, December 2008.

J. M. Comeron. What controls the length of noncoding dna? *Current Opinion in Genetics & Development*, 11(6):652–659, 2001.

G. M. Cooper. *The Cell A Molecular Approach*. Sinauer Associates, Sunderland (MA), 2nd edition, 2000. ISBN 10: 0-87893-106-6.

M. Deutsch and M. Long. Intron-exon structures of eukaryotic model organisms. *Nucleic Acids Research*, 27(15):3219–3228, 1999.

D. Grimaldi and M. S. Engel. *Evolution of the Insects*. Cambridge University Press, 2005.

P. Haddrill, B. Charlesworth, D. Halligan, and P. Andolfatto. Patterns of intron se-

quence evolution in drosophila are dependent upon length and gc content. *Genome Biology*, 6(8):R67, 2005. ISSN 1465-6906. doi: 10.1186/gb-2005-6-8-r67.

T. Hall. BioEdit: a user-friendly biological sequence alignment editor and analysis program for windows 95/98/nt. *Nucleic Acids Symposium Series*, 41:95–98, 1999.

H. M. Hines, S. A. Cameron, and A. R. Deans. Nest architecture and foraging behavior in bombus pullatus (Hymenoptera: Apidae), with comparisons to other tropical bumble bees. *Journal of The Kansas Entomological Society*, 80(1):1–15, 2007.

Y. Jia, Y. Zhang, P. R. Kolatkar, C. K. Kwoh, and M. K. Sundaram. Intron/exon: Which one tell us more about coding of life - evidence from statistical analysis of length distribution. In *Workshop on Genomic Signal Processing and Statistics (GENSIPS'04)*, 2004.

K. Katoh, G. Asimenos, and H. Toh. Multiple alignment of dna sequences with mafft. *Bioinformatics for DNA Sequence Analysis, Methods in Molecular Biology*, 537:39–64, 2009.

W. J. Kent. BLAT-the BLAST-Like alignment tool. *Genome Research*, 12:656–664, 2002.

S. Kuersten and E. B. Goodwin. The power of the 3′ utr: translational control and development. *Nature Reviews Genetics*, 4:626–637, 2003. doi: 10.1038/nrg1125.

S. Kurtz, A. Phillippy, A. L. Delcher, M. Smoot, M. Shumway, C. Antonescu, and

S. L. Salzberg. Versatile and open software for comparing large genomes. *Genome Biology*, 5(2):R12, 2004.

C. Li, J.-J. M. Riethoven, and L. Ma. Exon-primed intron-crossing (EPIC) markers for non-model teleost fishes. *BMC Evolutionary Biology*, 10(90), 2010.

D. Maglott, J. Ostell, K. D. Pruitt, and T. Tatusova. Entrez Gene: gene-centered information at ncbi. *Nucleic Acids Research*, 39:D52–D57, 2010.

J. S. Mattick. The genetic signatures of noncoding rnas. *PLoS Genetics*, 5(4): e1000459, 2009. doi: 10.1371/journal.pgen.1000459.

J. Moran, R. DeBerardinis, and H. Kazazian Jr. Exon shuffling by l1 retrotransposition. *Science*, 283(5407):1530–1534, 1999.

B. Morgenstern, K. Frech, A. Dress, and T. Werner. DIALIGN: finding local similarities by multiple sequence alignment. *Bioinformatics*, 14(3):290–294, 1998. doi: 10.1093/bioinformatics/14.3.290.

R. Mott. Est_genome: a program to align spliced dna sequences to unspliced genomic dna. *CABIOS*, 13(4):477–478, 1997.

M. C. Munoz-Torres, J. T. Reese, C. P. Childers, A. K. Bennett, J. P. Sundaram, K. L. Childs, J. M. Anzola, N. Milshina, and C. G. Elsik. Hymenoptera Genome Database: integrated community resources for insect species of the order hymenoptera. *Nucleic Acids Research*, pages D658–D662, 2010.

T. Nawy. Non-model organisms. *Nature Methods*, 9(1):37, 2012.

NCBI. A science primer, Mar 2013a. URL `http://www.ncbi.nlm.nih.gov/About/primer/est.html`.

NCBI. Blast search parameters, Mar 2013b. URL `http://www.ncbi.nlm.nih.gov/BLAST/blastcgihelp.shtml`.

C. Notredame, D. G. Higgins, and J. Heringa. T-Coffee: a novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302:205–217, 2000.

S. Ohno. So much "junk" DNA in our genome. *Brookhaven Symp Biol*, 23:366–370, 1972.

I. Pagani, K. Liolios, J. Jansson, I.-M. A. Chen, T. Smirnova, B. Nosrat, V. M. Markowitz, and N. C. Kyrpides. The Genomes OnLine Database (GOLD) v.4: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Research*, 40(D1):571–579, February 2013.

N. B. Pandey, N. Chodchoy, T. J. Liu, and W. F. Marzluff. Introns in histone genes alter the distribution of 3′ ends. *Nucleic Acids Research*, 18(11):3161–3170, 1990.

P. Rice, I. Longden, and A. Bleasby. EMBOSS: the european molecular biology open software suite. *Trends in Genetics*, 16(6):276–277, 2000.

I. T. Rombel, K. F. Sykes, S. Rayner, and S. A. Johnston. ORF-FINDER: a vector for high-throughput gene identification. *Gene*, 282:33–41, 2002.

A. Rose. Intron-mediated regulation of gene expression. *Nuclear Pre-mRNA Processing in Plants*, pages 277–290, 2008.

A. B. Rose. The effect of intron location on intron-mediated enhancement of gene expression in arabidopsis. *The Plant Journal*, 40:744–751, 2004.

L. Rychlewski, L. Jaroszewski, W. Li, and A. Godzik. Comparison of sequence profile-strategies for structural predictions using sequence information. *Protein Science*, 9:232–241, 2000.

B. M. Sadd, M. Kube, S. Klages, R. Reinhardt, and P. Schmid-Hempel. Analysis of a normalised expressed sequence tag (EST) library from a key pollinator, the bumblebee *Bombus terrestris*. *BMC Genomics*, 11:110, 2010.

B. J. Sharanowski, B. Robbertse, J. Walker, S. R. Voss, R. Yoder, J. Spatafora, and M. J. Sharkey. Expressed sequence tags reveal proctotrupomorpha (minus chalcidoidea) as sister to aculeata (hymenoptera: Insecta). *Molecular Phylogenetics and Evolution*, 57:101–112, 2010.

H.-B. Shen and K.-C. Chou. Ensemble classifier for protein fold pattern recognition. *Bioinformatics*, 22(14):1717–1722, 2006.

T. Smith and M. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.

J. Thompson, D. Higgins, and T. Gibson. CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22): 4673–4680, 1994.

C. Trapnell, L. Pachter, and S. L. Salzberg. TopHat: discovering splice junctions

with RNA-Seq. *BMC Bioinformatics*, 25(9):1105–1111, 2009. doi: 10.1093/bioinformatics/btp120.

J. D. Watson and F. Crick. Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 171:737–738, 1953.

S. J. Wheelan, D. M. Church, and J. M. Ostell. Spidey: A tool for mrna-to-genomic alignments. *Genome Research*, 11:1952–1957, 2001.

D. Wheeler and M. Bhagwat. *BLAST QuickStart: Example-Driven Web-Based BLAST Tutorial. Comparative Genomics*, volume 1 and 2. Humana Press, 2007.

S. Wilkeninga and A. Baderb. Quantitative real-time polymerase chain reaction: Methodical analysis and mathematical model. *Journal of Biomolecular Techniques*, 15(2):107–111, 2004.

E. Wolf, B. Kastner, J. Deckert, C. Merz, H. Stark, and R. Luhrmann. Exon, intron and splice site locations in the spliceosomal b complex. *The EMBO Journal*, 28: 2283–2292, 2009.

A. Woolfe, M. Goodson, D. K. Goode, P. Snell, G. K. McEwen, T. Vavouri, S. F. Smith, P. North, H. Callaway, K. Kelly, K. Walter, I. Abnizova, W. Gilks, Y. J. K. Edwards, J. E. Cooke, and G. Elgar. Highly conserved non-coding sequences are associated with vertebrate development. *PLoS Biol*, 3(1):e7, 11 2004. doi: 10.1371/journal.pbio.0030007.

M. Yandell, C. J. Mungall, C. Smith, S. Prochnik, J. Kaminker, G. Hartzell, S. Lewis, and G. M. Rubin. Large-scale trends in the evolution of gene structures within 11

animal genomes. *PLoS Comput Biol*, 2:e15, 03 2006. doi: 10.1371/journal.pcbi.0020015.

M. Zhang. Statistical features of human exons and their flanking regions. *Human Molecular Genetics*, 7(5):919–932, 1998.

L. Zhu, Y. Zhang, W. Zhang, S. Yang, J.-Q. Chen, and D. Tian. Patterns of exon-intron architecture variation of genes in eukaryotic genomes. *BMC Genomics*, 10 (1):47, 2009. ISSN 1471-2164. doi: 10.1186/1471-2164-10-47.