

Fast and Accurate Gene Prediction by Decision Tree Classification

Rong She¹, Jeffrey Shih-Chieh Chu², Ke Wang¹, and Nansheng Chen²

¹School of Computing Science, Simon Fraser University, Canada.

²Department of Molecular Biology and Biochemistry, Simon Fraser University, Canada.

Abstract

Gene prediction is one of the most challenging tasks in genome analysis, for which many tools have been developed and are still evolving. In this paper, we present a novel gene prediction method that is both fast and accurate, by making use of protein homology and decision tree classification. Specifically, we apply the principled entropy and decision tree concepts to assist in such gene prediction process. Our goal is to resolve the exact gene structures in terms of finding “coding” regions (*exons*) and “non-coding” regions (*introns*). Unlike traditional classification tasks, however, we do not have explicit class labels for such structures in the genes. We use protein sequence (the product of gene) as a query to help in finding genes that are homologous to the query protein and deduce class labels based on homology. Our experiments on the genomes of two nematodes *C. elegans* and *C. briggsae* show that in addition to achieving prediction accuracy comparable with that of the state of the art methods, it is several orders of magnitude faster, especially for genes that encode longer proteins.

1 Introduction.

With the fast development of genome sequencing technologies, the amount of genome data accumulated has been increasing exponentially. To date, biologists have sequenced genomes of more than a thousand species, while thousands of more species are currently being sequenced (<http://genomesonline.org/>). The genome sequence for any organism is composed of DNA sequences for each of the chromosomes in that organism. In the human genome, a DNA sequence (chromosome) can be hundreds of millions

of base pairs (bp) long, where usually only less than 5% contains instructions for protein-coding and non-coding genes. The DNA segments that carry this genetic information are called *genes* (Figure 1). In order to understand the sequenced genome of a species and exploit such resources for biological and medical purposes, one of the first and most important steps in genome study is gene prediction, i.e. determining the positions of genes and their structures on the DNA sequences [9]. In this paper, we will focus on the prediction of protein-coding genes.

A typical protein-coding gene contains “coding” regions (*exons*) as well as “non-coding” regions (*introns*) (Figure 1). When a gene is expressed, it is first transcribed as pre-mRNA, which then undergoes a process called *splicing*, in which non-coding regions are removed. A mature mRNA, which does not contain introns, serves as a template for the synthesis of a protein in *translation*. In translation, each codon, a group of three adjacent base pairs in mRNA, directs the addition of one amino acid (according to the genetic code [19]) to a peptide being synthesized. Thus, a protein is a sequence of amino acid residues corresponding to the mRNA sequence of a gene.

The junction between an exon and an intron is called a *splice site*, which is either a *donor* (the start site of an intron) or an *acceptor* (the end site of an intron), as illustrated in Figure 1. The DNA sequence that is formed by removing the introns and joining the exons is known as a *spliced sequence*.

The tasks of gene prediction include finding the positions of gene start, gene end, and splice sites on the DNA sequence. Since introns and exons are complementary to each other (i.e. introns can

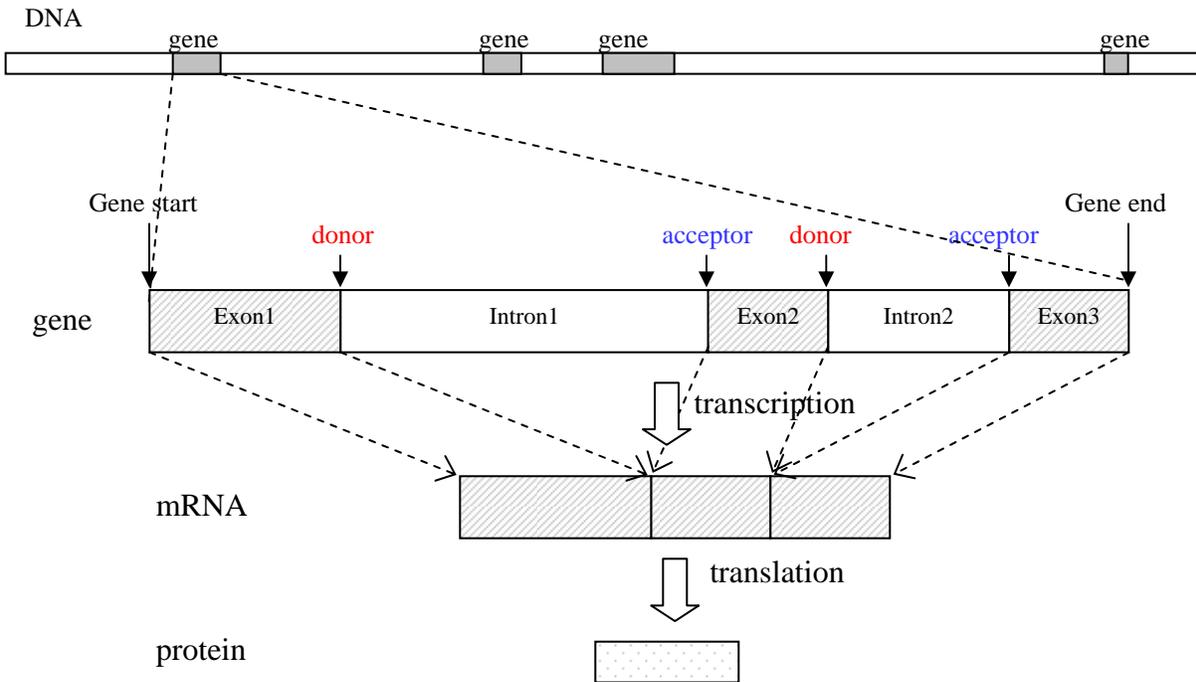


Figure 1: The structure of a protein-coding gene on a DNA sequence and the process of gene expression. A DNA sequence may contain thousands of genes. A gene consists of exons that are coding regions (shaded regions in gene structure) and introns that are non-coding (non-shaded regions in gene structure). During transcription, introns are removed and exons are glued together to form mRNA. Finally, mRNA is translated into a protein (gene product).

be identified as genomic regions flanked by adjacent exons) (Figure 1), the gene structure is determined once all its introns are identified. If a protein sequence (gene product) is used as a query to align with the gene on the DNA sequence, introns should be DNA regions that have no query alignment, because the query is translated from exons only. The spliced sequence of the gene should be homologous to the query protein.

1.1 Gene prediction: current status. Given a newly-sequenced genome, there are currently many computational tools to predict possible gene structures on the DNA sequences [5, 6, 7, 8, 9]. These tools can be divided into two major categories: *ab initio* and homology-based methods. *ab initio* methods find genes by systematically examining the DNA sequences for certain signals including start codon,

stop codon and donor and acceptor signals, such as GENSCAN [20], and AUGUSTUS [21]. On the other hand, homology-based programs make use of extrinsic evidence such as protein sequences, mRNAs, ESTs or other genomic sequences in finding the genes on the DNA sequence. The availability of genome sequences of related species has created growing demand for better and faster homology-based gene prediction programs, which gives rise to many developments in such area, such as GeneWise [18], Projector [22], TwinScan [23], exonerate [24], SGP2 [25], SLAM [26]. In general, it has been shown that homology-based gene prediction methods generally outperform the *ab initio* methods in terms of accuracy when there are extrinsic evidences available [9, 11], with GeneWise [18] being one of the most widely used homology-based gene prediction programs.

Most current gene prediction programs such as GeneWise are based on hidden Markov models (HMMs) and their computational complexity is intrinsically high due to the high running cost of Viterbi algorithm [1] that is used in HMM solutions. This makes them very slow when annotating large scale of genome sequences. For whole-genome scale analysis, genome sequences are usually preprocessed to refine the input sequence for these programs, including GeneWise [2], so that they can be executed within a reasonable time frame. Nevertheless, for longer query proteins, GeneWise takes more than one hour to finish prediction on one gene even with such preprocessing.

1.2 Our contribution. We have developed a novel homology-based gene prediction program, genBlastDT, which is comparable on accuracy with GeneWise but is many orders of magnitude faster than GeneWise. Similar to GeneWise, genBlastDT takes two biological sequences as input: a query protein sequence (i.e., gene product), Q , and a target DNA sequence, T . genBlastDT is able to quickly and accurately find genes on the target DNA sequence T that are homologous to the query protein Q . The unique contribution of genBlastDT is a novel application of decision tree classification in identifying intron regions. More details are described in the later discussion of our approach.

genBlastDT is built on top of a recently developed program, genBlastA [17], which can quickly identify homologous gene regions on the DNA sequence for a given protein and thus can be used as an independent preprocessing tool for GeneWise-like programs. genBlastA utilizes a fast and sensitive local alignment tool such as BLAST [16] that finds all sequence homologies between the given query protein Q and the target DNA sequence T . The result is a set of unorganized local alignments called *HSPs* (high-scoring segment pairs), where each HSP is a pair of segments, one from the query sequence (called *query segment*), and the other from a target sequence (called *target segment*). genBlastA then organizes the HSPs into groups on the DNA sequence, so that each group of HSPs corresponds to a potential gene that is homol-

ogous to the query protein. These groups indicate the candidate and approximate regions where homologous genes are present (*gene regions*). Each group also identifies the relevant HSPs with alignment information for each gene region.

Although each HSP group indicates a candidate region for a gene on the target DNA sequence, it does not provide details of the exact structure of the potential gene. To resolve the structure of a gene, we need to identify the positions of its exons and introns. This is exactly the purpose of genBlastDT. In this paper, we will focus on the problem of resolving gene structures using the HSP groups returned by genBlastA.

Challenges in resolving gene structures. In each gene region, the HSPs reported by genBlastA suggest the presence of exons of a gene. Introns are not represented by HSPs because the query is a protein sequence that is coded only by the exons in a gene. However, mapping the HSPs to exons is a challenging task due to several reasons. First, HSPs often contain gaps and mismatches in their alignments and the boundaries of exons usually do not coincide with boundaries of HSPs. Second, because BLAST always tries to extend HSPs as long as its score is above a threshold, it is possible for one HSP to correspond to the region that contains multiple exons. On the other hand, due to evolutionary divergence, exons may not have precise correspondences with query protein fragments and it is possible for the region of one exon to be represented by multiple HSPs with small gaps between them. Third, splice sites are usually signaled by reserved sequences (i.e. splice site signals) which must be taken into account when resolving the gene structure. However, random sequences in the target DNA sequence may resemble splicing signals, which makes it difficult to identify the real splice sites.

Our approach. Since the gene region contains only introns and exons, the problem of determining intron regions can be seen as a two-class classification problem: every base pair belongs to either the exon

class or the intron class. Therefore, in principle, classification techniques can be used to separate intron regions from exon regions. However, unlike traditional classification, we do not have labeled training data. Instead, we have the alignment information returned by genBlastA. One of our innovations is a novel classification of intron and exon regions by incorporating the decision tree technique and representing the local alignments by a sequence of alignment scores at each base pair. Our experiments on both *C. elegans* and *C. briggsae* genomes show that while this approach achieves prediction accuracy comparable with that of the state of the art methods, it is several orders of magnitude faster, especially for genes that encode longer proteins.

The rest of this paper is organized as follows. Section 2 describes the method used in genBlastDT that includes a novel application of decision tree construction algorithm as its first component. Section 3 gives empirical evaluations. Section 4 concludes the paper.

2 Methods.

Our overall gene prediction process is as follows.

1. *Local alignment*: given a query protein sequence and a target DNA sequence, a local similarity search program such as BLAST [16] is first used to locate local similarities between the query and the target sequence. The output of is a list of HSPs, where each HSP is a pair of segments (query segment and target segment).
2. *Gene region identification*: from the usually large number of BLAST HSPs, genBlastA [17] is used to produce a ranked list of HSP groups where each group corresponds to a candidate target gene.
3. *Gene structure resolution*: the program presented in this paper, genBlastDT, takes the output from genBlastA and the target DNA sequence as well as the query protein sequence as its input, and output the exact gene structure. Because each group of HSPs is independent of

each other, genBlastDT returns a gene structure for each group independently. In the rest of our discussions, we will only focus on resolving gene structure for a single group of HSP.

The focus of this project is the third step. For illustration, Figure 2 shows an example of an HSP group found by genBlastA. Each HSP is a local sequence alignment between the query sequence and the target sequence, containing a query segment and a target segment. An example of HSP is shown as follows:

```

query segment:  SGLKLESLDLSHKNLTEV
                | | | | | | | | | | | | | |
target segment:  SSFFLESLDLSHKNLTEV

```

Each letter in the query or target segment denotes an amino acid residue. The middle line indicates the matching positions by “|”s.

Figure 2 also shows the actual gene model (exons are the purple boxes and introns are the lines connecting exons) as a reference, which is the desired output of genBlastDT. It can be seen that HSPs generally have some correspondences with exons in the gene, however, such correspondences are only approximate and the exact structure of the gene still needs to be carefully identified. For example, in Figure 2, the circled HSP represents two adjacent exons that are separated by a short intron.

2.1 genBlastDT overview. Given a group of HSPs (returned by genBlastA) that corresponds to a candidate gene, genBlastDT needs to determine the positions of gene start, gene end and all splice sites. The gene is considered to fall in the region between the start position of the first HSP and the end position of the last HSP in the group, with some allowance for extension at both ends (up to 1000 base pairs extension).

Gene start is almost always signaled by a “ATG” sequence (called *start codon*) and gene end is signaled by either “TAG”, “TGA” or “TAA” (called *stop codon*). Thus gene start and gene end can be found by looking for such signals on the target DNA sequence. The position of gene start is given by the closest start

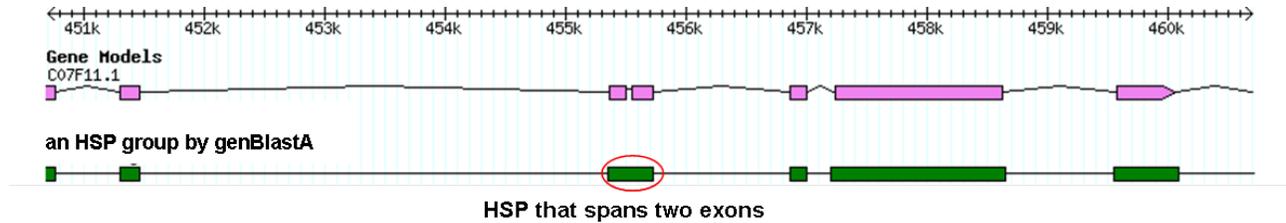


Figure 2: An HSP group for gene C07F11.1 (a gene in the *C. elegans* species) in its corresponding genomic region.

codon before the first HSP. Similarly, gene end is given by the closest stop codon after the last HSP.

Splice sites also usually have well-conserved signals. Most introns start with the base pairs “GT” (donor signal) and end with the base pairs “AG” (acceptor signal) [27]. However, the presence of these signals is not sufficient to detect the splice sites because such signals are very general and identifying the true splice sites is non-trivial. We tackle this problem by dividing it into three smaller tasks: first, we determine the approximate regions for each intron, called “intron regions”; next, for each approximate intron region, we find the candidate donor sites and acceptor sites in that region; finally, we determine the best combination of donor and acceptor sites among these candidates for each intron region.

2.2 Classifying intron regions. An intron region in this step is an approximate region where a possible intron is located. The exact positions of the splice sites will be defined in the following steps. Thus, this intron-region identification step serves as a critical starting point to search for actual splice sites.

We consider two types of DNA regions separately: regions that fall outside HSPs and inside HSPs. As discussed earlier, introns are DNA regions that have no correspondence with the query protein. Since HSPs are local alignments between the DNA sequence and the query protein, we can consider those DNA regions outside HSPs to have no query correspondence and thus likely belong to intron regions. On the other hand, for DNA regions within HSPs, although each HSP usually corresponds to a single exon, a single HSP sometimes corresponds to multiple exons (e.g. the circled HSP in Figure 2), which makes intron

identification within a HSP difficult. Thus, we need a more systematic way to split the region inside a HSP into possible intron and exon regions.

The general idea of classifying exon and intron regions is that exons are DNA regions that align with the query in high similarity, and introns are DNA regions that are dissimilar from the query (i.e., have little or no query correspondence). Such similarity can be measured by the alignment scores for each aligned pair of amino acid residues. In this project, we use the BLOSUM62 score matrix [3], which provides a score for each residue in an alignment based on statistics from conserved blocks of known related residues. It is well used in similarity search programs such as BLAST. With this score matrix, each HSP alignment can be seen as a sequence of numeric scores, one for each alignment residue. Figure 3 shows the scores for an example HSP.

The problem of splitting residues in the HSP region into intron and exon regions can be modeled by the decision tree classification problem [4]. In our context, each data item to be classified is a residue in the HSP alignment and the ordering of the residues are fixed. Each residue has only one feature - its alignment score according to the BLOSUM62 matrix. However, we do not have the explicit class attribute assumed by a traditional classification algorithm. Instead, the class labels need to be determined through computation.

We assign a class label (either “intron region” or “exon region”) to each residue in a HSP based on its score. A user defined “*class threshold*” is used to determine the class label, i.e. residues with scores higher than the threshold belong to exon regions and residues with lower scores are intron regions. The

	Split: S		R																				
Position:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
Query:	K	R	I	A	H	L	H	L	E	H	N	S	I	V	L	A	N	A	T	N	L	L	
Target:		+											+						+		+		
Score:	5	2	-3	-2	0	4	-1	4	0	-1	6	0	2	4	4	-1	0	-2	1	0	2	4	
Class:	E	E	I	I	I	E	I	E	I	I	E	I	E	E	E	E	I	I	I	E	I	E	E
	R_1		R_2																				

Figure 3: A sequence of scores for an HSP.

class threshold can be adjusted for different genomes.

Our algorithm works as follows. Starting from the region of the entire HSP, a decision tree is constructed to recursively find the best binary split on the residues in this HSP region. At each iteration, we use the entropy-based measure that is commonly used in decision tree algorithms to evaluate all possible split points and find the best split that results in maximal information gain ratio [4]. Because the absolute value of a BLOSUM62 score indicates significance in protein similarity, we consider such value as the weight of each residue and use it to adjust the frequency count during the evaluation of each possible split.

As an example, in Figure 3, an HSP region, R , consists of 22 residues. If the class threshold is 0, each residue can be labeled as shown in the figure ('E' stands for "exon", 'I' stands for "intron"). The total weight of all intron residues is 10, while the total weight of all exon residues is 38. Thus the total weight of the entire region is 48. The information gain ratio that corresponds to the split between residue 2 and 3, denoted as S , is computed as follows:

The entropy for the entire region R is:

$$Entropy(R) = -\frac{38}{48} \times \log\left(\frac{38}{48}\right) - \frac{10}{48} \times \log\left(\frac{10}{48}\right) = 0.73$$

For the left region R_1 of the split, which contains only exon residues of total weight 7, its entropy is:

$$Entropy(R_1) = 0$$

For the right region R_2 , which contains intron residues of total weight 10 and exon residues of total weight 31, its entropy is:

$$Entropy(R_2) = -\frac{31}{41} \times \log\left(\frac{31}{41}\right) - \frac{10}{41} \times \log\left(\frac{10}{41}\right) = 0.8$$

Thus the information gain w.g.t the split is:

$$\begin{aligned} Gain &= Entropy(R) - \sum_{i=1}^2 \left(\frac{R_i}{R} \times Entropy(R_i)\right) \\ &= 0.73 - \frac{7}{48} \times 0 - \frac{41}{48} \times 0.8 = 0.05 \end{aligned}$$

The information gain ratio is:

$$\begin{aligned} GainRatio &= \frac{Gain}{Entropy(R|S)} \\ &= \frac{0.05}{\left(-\left(\frac{7}{48}\right) \times \log\left(\frac{7}{48}\right) - \frac{41}{48} \times \log\left(\frac{41}{48}\right)\right)} \\ &= 0.08. \end{aligned}$$

Each iteration determines a best binary split with maximal gain ratio and splits the current region into two sub-regions. This is the same as generating two decision tree nodes that split the data from the parent node. Each node corresponds to a continuous region. Such procedure is recursively done until the region reaches minimum weight or all residues in the region belong to the same class. Pruning is then performed to avoid overfitting the individual residues, similar to the process in the traditional decision tree algorithm, based on error estimation on nodes and subtrees. Usually the errors may be estimated using external test data with pre-defined class labels. However, due to the lack of such test data in our context, pruning needs to be done with only the data from which the tree was built. We follow the error estimation method used in C4.5 [4].

To estimate the error in each decision tree node, a class label needs to be determined for each node. Unlike a pre-determined class attribute, however, the

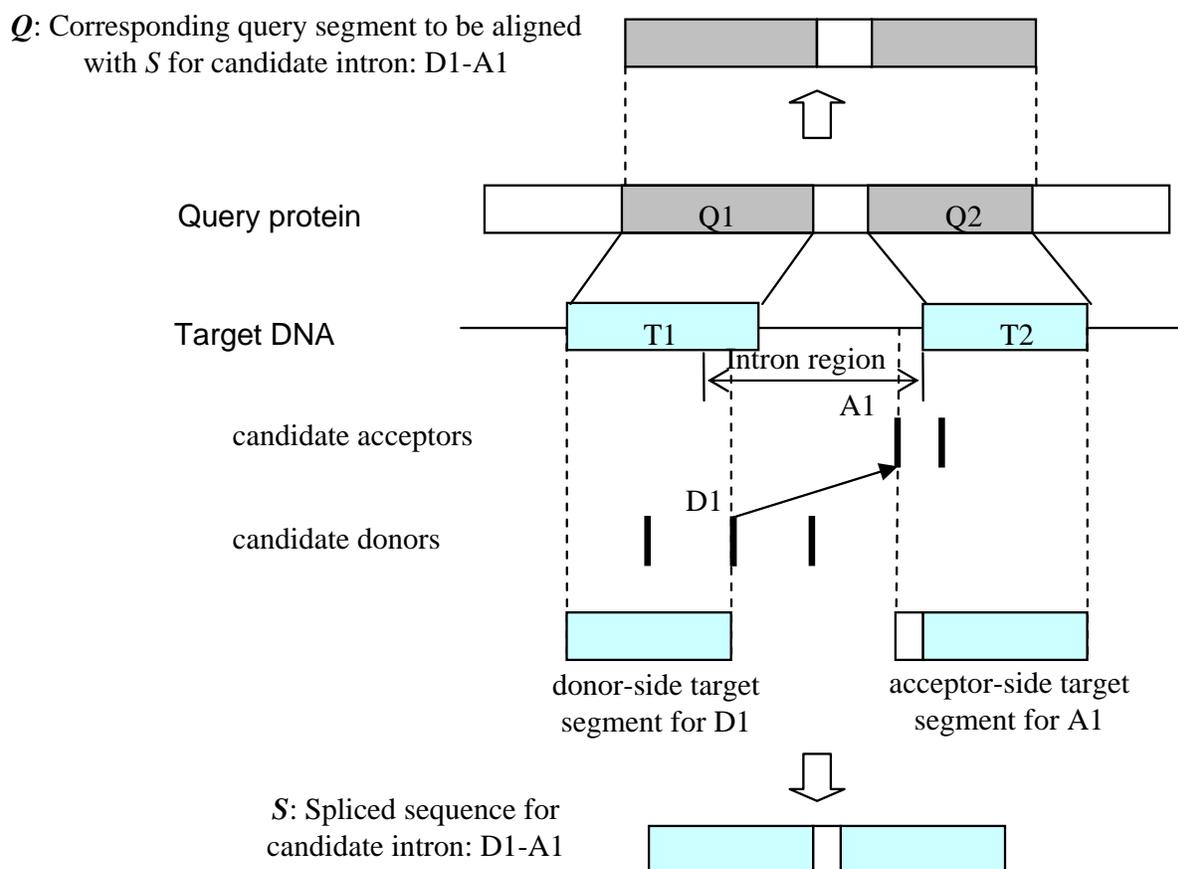


Figure 4: Finding the best pair of donor and acceptor for each intron region. Two HSPs ($\langle T1, Q1 \rangle$, $\langle T2, Q2 \rangle$) are shown in the current intron region. For donor *D1* and acceptor *A1*, its spliced sequence is deduced from the positions of the two HSP target segments. On the other hand, the corresponding query segment is deduced from the positions of the two HSP query segments.

class label assigned to each node is determined on the fly by the aggregated alignment score of all residues within that region. A node (region) is labelled as intron region if its score is lower than the class threshold. Errors are estimated for each node and the tree is pruned in a bottom-up fashion, where children nodes are pruned if their parent node contains less error than the subtree rooted at that parent node.

Finally, the splits encoded by the pruned decision tree will represent the way of splitting the HSP region into exon and intron regions.

2.3 Selecting candidate splice sites. With the approximate intron regions found, we can now look

for splice sites and exact intron coordinates. An intron region defines the approximate boundaries of a possible intron. From these intron regions, we search for splicing signals that are close to the borders of such regions. More specifically, for each intron region, we search for donor and acceptor signals (GT/AG) independently. We use a user defined threshold (`MAX_NUM_SPLICE_SITES`) to control the number of candidates selected around each border. For each intron region, we look for at most `MAX_NUM_SPLICE_SITES` donor signals (“GT”) that are closest to the upstream border of the region. Similarly, we also look for at most `MAX_NUM_SPLICE_SITES` acceptor signals (“AG”)

that are closest to the downstream border of each intron region. These sites are the candidate splice sites that will be examined later to find the most possible splice sites.

2.4 Finding the best splice sites. For each intron region, we now determine the *best pair* of donor and acceptor sites among the candidate sites. Since we are searching for genes that are homologous to the query protein, we should look for splice sites so that the spliced sequence (obtained by removing introns and joining exons) has maximized homology to the query sequence, which indicates maximized similarity between the query and the target genes. The similarity between two sequences is measured by the PID (percentage of identity) in their alignment. Since HSPs have already contained the alignment information, we can use HSPs to quickly compute the similarity between a spliced sequence and the query.

Consider an intron region I . I is associated with its own set of donors D_1, \dots, D_p and acceptors A_1, \dots, A_q , as given in previous step. For each pair $\langle D_i, A_j \rangle$ in I , it induces the spliced sequence S and the corresponding query segment Q . S is formed by connecting the target segment of the HSP at the upstream side of a donor site (called “*donor-side target segment*”) with the target segment of the HSP at the downstream side of an acceptor site (called “*acceptor-side target segment*”). Q is induced from the two HSPs that defined the region of S . The alignment between S and Q is computed based on the HSPs that form S . Figure 4 shows an example of S and Q for donor D_1 and acceptor A_1 . The pair that results in the highest alignment PID between S and Q is the best pair of donor and acceptor in region I .

Once the splice sites are selected for all intron regions, the predicted gene structure is thus revealed.

3 Empirical Evaluation.

We tested the performance of genBlastDT on the genomes of the popular model organism *C. elegans* [15] and its sister species *C. briggsae* [28] (Wormbase release WS200 [12]). Using the entire *C. elegans*

protein sequences (23,973 peptides) as queries, we evaluated genBlastDT for its performance in finding homologous genes on both *C. elegans* genome and *C. briggsae* genome. All experiments are done on a computer with Intel Xeon 3.06GHz CPU and 2G memory, running Linux version 2.6.18.

For comparison, the predictions made by genBlastDT are compared with GeneWise, which is also based on protein homology. Because the running time of GeneWise depends heavily on the length of DNA sequence to be examined, we first use genBlastA to narrow down the gene regions so GeneWise only needs to scan much smaller DNA sequences instead of the entire genome. This means both genBlastDT and GeneWise use the same gene regions as input for gene prediction. The running time of both genBlastDT and GeneWise does not include the time spent by their preprocessing tools (BLAST and genBlastA).

In addition, we also compared genBlastDT results with nGASP predictions [11]. The nGASP predictions were produced by combining results from a number of different gene prediction programs that are among the best performers in the nGASP evaluation. The nGASP results are obtained from the WormBase ftp site [13].

3.1 Results on *C. elegans* genome. We compared genBlastDT with GeneWise on both accuracy and running time, using all *C. elegans* genes on Chromosome I (ChrI) with totally 3,523 peptides as the queries. We also compared genBlastDT predictions on the entire *C. elegans* genome (6 chromosomes in total) with the existing predictions of nGASP on prediction accuracy.

Accuracy comparison. As *C. elegans* is a well-studied species with most of its gene models well-defined and confirmed, we used the curated gene models in Wormbase [12] as the ground truth for accuracy evaluation. We compared genBlastDT, GeneWise and nGASP gene models in terms of their *specificity* and *sensitivity* at transcript, exon and nucleotide levels [14]. Specificity ($Sp.$) is the percentage of predictions that are correct; sensitivity ($Sn.$) is the per-

Table 1: Accuracy comparisons of genBlastDT, GeneWise, nGASP on *C. elegans* genome

	Transcript level			Exon level			Nucleotide level		
	<i>Sp.</i> (%)	<i>Sn.</i> (%)	<i>F</i>	<i>Sp.</i> (%)	<i>Sn.</i> (%)	<i>F</i>	<i>Sp.</i> (%)	<i>Sn.</i> (%)	<i>F</i>
genBlastDT (ChrI)	85.02	85.02	0.85	97.61	96.56	0.97	99.45	99.39	0.99
GeneWise (ChrI)	89.02	88.66	0.89	98.02	96.44	0.97	99.89	99.64	1.00
nGASP (ChrI)	74.31	58.38	0.65	93.12	70.17	0.80	97.37	94.74	0.96
genBlastDT (entire genome)	84.92	84.90	0.85	97.46	96.37	0.97	99.38	99.39	0.99
nGASP (entire genome)	72.68	58.97	0.65	92.17	71.43	0.80	97.03	94.02	0.96

Table 2: Length distribution of genes in *C. elegans* Chromosome I

Gene length	[1, 500)	[500, 1000)	[1000, 2500)	[2500, 5000)	Over 5000
number of genes	2402	864	220	31	6

centage of actual coding genes/exons/nucleotides that are predicted as such. We also computed *F-measure* (the harmonic mean of *Sp.* and *Sn.*) to evaluate the trade-off between specificity and sensitivity.

Table 1 shows the accuracy results of genBlastDT, GeneWise and nGASP. genBlastDT shows very comparable performance as GeneWise with slightly lower accuracy at the transcript level. On the exon level, both specificity and sensitivity of genBlastDT are very close to those of GeneWise, with close to 98% specificity and 96% sensitivity. At the nucleotide level, the accuracy of genBlastDT and GeneWise are even closer, with both specificity and sensitivity over 99%. On the other hand, the accuracy of nGASP is much lower than either genBlastDT or GeneWise, at all three levels. In particular, the sensitivity of nGASP on the transcript level is almost 30 percentage points lower than either genBlastDT or GeneWise. The genome-wide results of genBlastDT and nGASP show very close performances to Chromosome I experiments. This shows that genBlastDT is comparable with state-of-the-art gene predictors in terms of overall accuracy.

Speed comparison. In order to examine the effect of gene length on the running time of genBlastDT and GeneWise, we divided genes into five categories depending on their lengths, as shown in Table 2.

While genBlastDT achieves the similar accuracy performance as GeneWise, it runs much faster than

GeneWise. Figure 5(a) shows the running time of genBlastDT and GeneWise on *C. elegans* for Chromosome I genes. The running time for each category is the average running time for all genes in that category. Note that this figure is on logarithmic scale, which shows the great difference between the speed of genBlastDT and GeneWise. In particular, for genes of longer lengths, genBlastDT is hundreds of times faster. This is because the running time of GeneWise depends heavily on the query gene length and grows rapidly as the gene length increases. On the other hand, the running time of genBlastDT remains rather stable. This shows the drastic speed advantage of genBlastDT over GeneWise and makes genBlastDT very practical for large-scale genome analysis.

3.2 Results on *C. briggsae* genome. *C. briggsae* [28] is a sister species of *C. elegans* and most *C. briggsae* genes have homologous counterparts on the *C. elegans* genome. To compare genBlastDT with GeneWise on the *C. briggsae* genome, we used the same set of *C. elegans* ChrI genes as queries. We also used genBlastDT to find all homologous *C. briggsae* genes by using the entire set of *C. elegans* genes as queries and compare the results with nGASP and WormBase gene models.

Accuracy comparison. The accuracy measure used in *C. briggsae* experiments is different from those in *C. elegans* experiments. For the *C. briggsae*

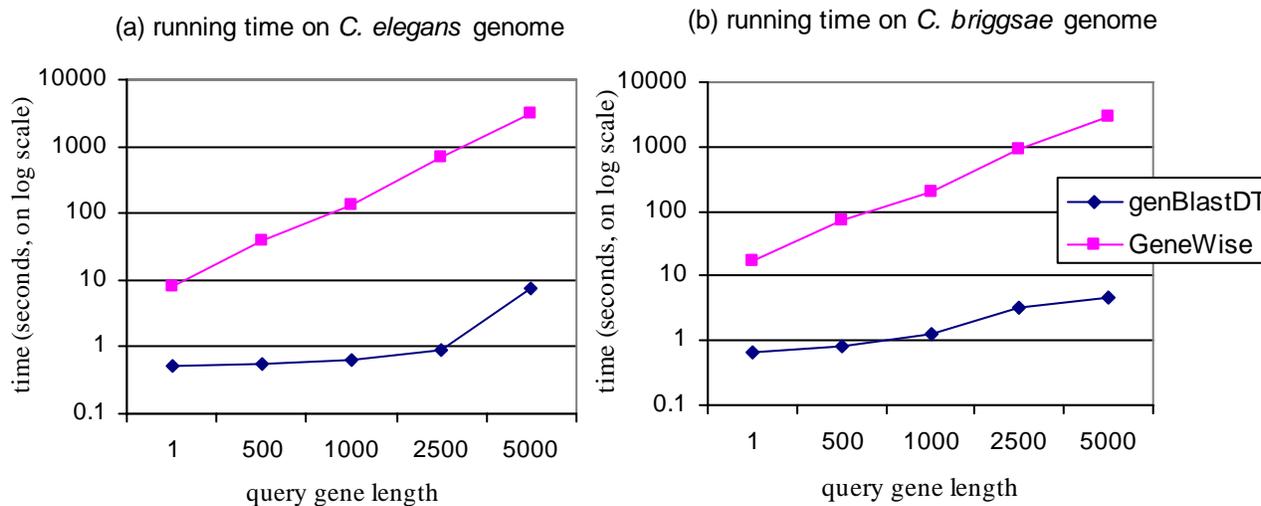


Figure 5: Running time comparison of genBlastDT vs GeneWise, on logarithmic scale.

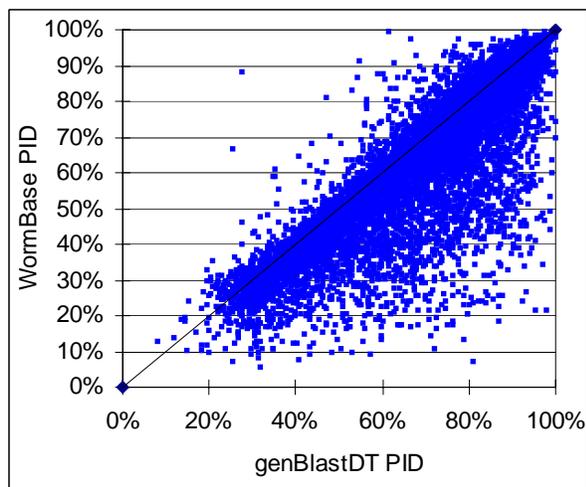


Figure 6: Accuracy comparison of genBlastDT vs Wormbase predictions on *C. briggsae* genome.

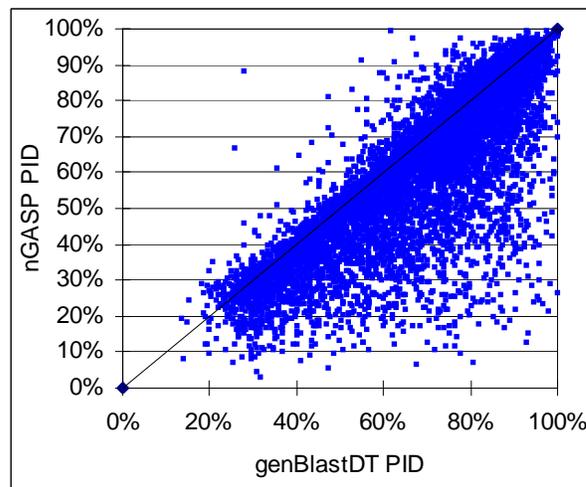


Figure 7: Accuracy comparison of genBlastDT vs nGASP predictions on *C. briggsae* genome.

genome, because its current Wormbase annotations [12] are mostly based on other gene prediction programs and cannot be considered as the ground truth, we evaluated all predictions (genBlastDT, GeneWise, nGASP [13], and Wormbase [12]) based on the alignment identity (PID, percentage of identity in the alignment) between the predicted gene product and the query protein. This measure is appropriate since

it provides a good indication on homology between the query and predicted gene model, as our ultimate goal is to predict genes that are homologous to the query. The higher the PID is, the more confident we are in the prediction.

Figure 6 shows the PID comparisons between genBlastDT and Wormbase. Each data point shows the correspondence between genBlastDT PID and

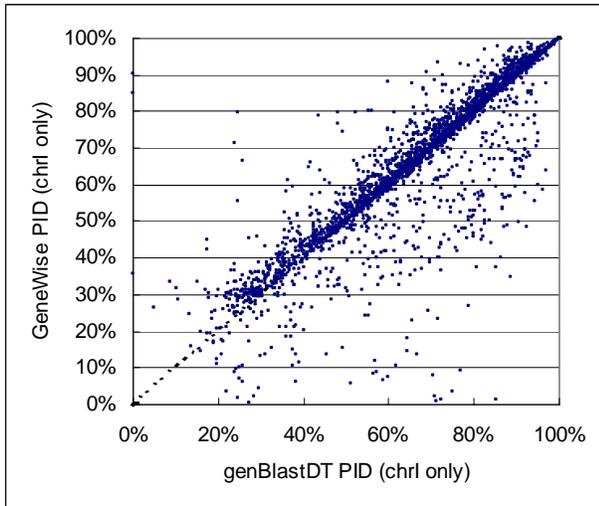


Figure 8: Accuracy comparison of genBlastDT vs GeneWise predictions on *C. briggsae* genome, with only *C. elegans* ChrI genes as query.

Wormbase PID for the same query gene. The area below the diagonal line in the figure shows the cases (dots) where genBlastDT gives better PID than Wormbase predictions. It can be seen that lots of cases fall in this area. Similarly, Figure 7 shows the PID comparisons between genBlastDT and nGASP. It demonstrates the same trend that genBlastDT frequently gives much higher PIDs than nGASP predictions. The average PID of the models predicted by Wormbase is 68.6%, which is very close to that of nGASP predictions (average PID of 68.3%). In contrast, genBlastDT has average PID of 73%, which is a solid improvement, considering there are over 20,000 cases.

Figure 8 compares genBlastDT with GeneWise predictions based on ChrI query genes only. genBlastDT generally gives higher PIDs. genBlastDT produced much higher PID than that of GeneWise in 8.6% of cases (with PID difference larger than 10%), compared with 3.1% of cases where GeneWise PIDs are more than 10% higher. There are certain cases that GeneWise gives higher PIDs, however, it should be noted that GeneWise does not always predict a complete gene, especially when the target gene is some distance away from the query gene, due to the

fact that *C. briggsae* genes are often different from their *C. elegans* counterparts. On the other hand, genBlastDT always follows the gene signals (start codon, stop codon, splice site signals) rigorously and predicts the complete gene structure. On average, GeneWise achieved PID of about 72%, almost the same as the average PID of genBlastDT. This shows that both genBlastDT and GeneWise have comparable accuracy performance.

Speed comparison. The running time comparison is similarly done as in the *C. elegans* experiments. Figure 5(b) shows the running time of genBlastDT and GeneWise on the *C. briggsae* genome, using the same ChrI genes as queries. It shows very similar trends as in Figure 5(a).

4 Conclusion.

Gene prediction is an important and challenging problem. Our contribution in this paper is applying the principled data mining concept, decision tree classification, to resolve the approximate gene regions, i.e. intron regions and exon regions. Importantly, this simple method produces highly competitive results when compared with the best performers such as GeneWise and nGASP that employed heavy biological heuristics and were complex. An outstanding advantage of our method is the speed compared with these popular predication tools, making it especially efficient for large scale genomes and complex genes.

Acknowledgement. K.W. and N.C. are supported by Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants. J.S.-C.C. is supported by a NSERC Postgraduate Scholarship. Some support was also provided by the SFU Community Trust funded BCID Project. N.C. is also a Michael Smith Foundation for Health Research (MSFHR) Scholar.

References

- [1] A. J. Viterbi, *Error bounds for convolutional codes and an asymptotically optimum decoding algorithm*, IEEE Transactions on Information Theory, 13:2 (1967), pp. 260-269.

- [2] V. Curwen, E. Eyras, T. D. Andrews, L. Clarke, E. Mongin, S. M.J. Searle, and M. Clamp, *The Ensembl Automatic Gene Annotation System*, Genome Research, 14:5 (2004), pp. 942–950.
- [3] S. Henikoff and J. G. Henikoff, *Amino acid substitution matrices from protein blocks*, Proc. Natl. Acad. Sci., 89 (1992), pp. 10915–10919.
- [4] J. R. Quinlan, *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993.
- [5] J. H. Do. and D. K. Choi, *Computational approaches to gene prediction*, J. of Microbiology, 44:2 (2006), pp. 137–144.
- [6] W. H. Majoros and U. Ohler, *Advancing the state of the art in computational gene prediction*, K. Tuyls et al. (Eds.): KDECB 2006, LNBI 4366, pp. 81–106, Springer-Verlag Berlin Heidelberg, 2007.
- [7] C. Mathe, M. F. Sagot, T. Schiex and P. Rouze, *Current methods of gene prediction, their strengths and weaknesses*, Nucleic Acids Res., 30:19 (2002), pp. 4103–4117.
- [8] Z. Wang, Y. Chen and Y. Li, *A brief review of computational gene prediction methods*, Geno. Prot. Bioinfo., 2:4 (2004), pp. 216–221.
- [9] M. Zhang, *Computational prediction of eukaryotic protein-coding genes*, Nature Reviews Genetics, 3 (2002), pp. 698–709.
- [10] R. Guigo, P. Flicek, J. F. Abril, A. Reymond, J. Lagarde, F. Denoeud, S. Antonarakis, M. Ashburner, V. B. Bajic, E. Birney, R. Castelo, E. Eyras, C. Ucla, T. R. Gingeras, J. Harrow, T. Hubbard, S. E. Lewis and M. G. Reese, *EGASP: the human ENCODE Genome Annotation Assessment Project*. Genome Biology, 7(Suppl 1) (2006), S2.
- [11] A. Coghlan, T. J. Fiedler, S. J. Mckay, P. Flicek, T. W. Harris and D. Blasiar, *nGASP the nematode genome annotation assessment project*, BMC Bioinformatics, 9:549, 2008.
- [12] *WormBase web site*, <http://www.wormbase.org>, release WS200, Mar. 2009.
- [13] *WormBase ftp site for nGASP results*, http://ftp.wormbase.org/nGASP/final_gene_predictions/predictions/, 2009.
- [14] M. Burset and R. Guig, *Evaluation of gene structure prediction programs*, Genomics, 34 (1996), pp. 353–367.
- [15] C. elegans Sequencing Consortium, *Genome sequence of the nematode C. elegans: a platform for investigating biology*, Science, 282:5396 (1998), pp. 2012–2018.
- [16] S. F. Altschul, W. Gish, W. Miller, E. W. Myers and D. J. Lipman, *Basic local alignment search tool*, J. Mol. Biol., 215:3(1990), pp. 403–410.
- [17] R. She, J. S.C. Chu, K. Wang, J. Pei and N. Chen, *genBlastA: Enabling BLAST to identify homologous gene sequences*, Genome Res., 19 (2009), pp. 143–149.
- [18] E. Birney, M. Clamp and R. Durbin, *GeneWise and Genomewise*, Genome Res, 14 (2004), pp. 988–995.
- [19] *NCBI: The Genetic Codes*, compiled by A. Elzanowski and J. Ostell, NCBI, Maryland, U.S.A., <http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi>.
- [20] C. Burge and S. Karlin, *Prediction of complete gene structures in human genomic DNA*, J. Mol. Biol., 268 (1997), pp. 78–94.
- [21] M. Stanke and S. Waack, *Gene prediction with a hidden Markov model and new intron submodel*, Bioinformatics, 19(Suppl.2) (2003), pp. ii215–ii225.
- [22] I. M. Meyer and R. Durbin, *Gene structure conservation aids similarity based gene prediction*, Nucleic Acids Res., 32:2 (2004), pp. 776–783.
- [23] I. Korf, P. Flicek, D. Duan and M. R. Brent, *Integrating genomic homology into gene structure prediction*, Bioinformatics, 17(Suppl.) (2001), pp. S140–S148.
- [24] G. S. Slater and E. Birney, *Automated generation of heuristics for biological sequence comparison*, BMC Bioinformatics, 6:31, 2005.
- [25] G. Parra, P. Agarwal, J. F. Abril, T. Wiehe, J. W. Fickett and R. Guigo, *Comparative gene prediction in human and mouse*, Genome Res., 13 (2003), pp. 108–117.
- [26] L. Pachter, M. Alexandersson and S. Cawley, *Applications of generalized pair hidden Markov models to alignment and gene finding problems*, J. Comput. Biol., 9 (2002), pp. 389–399.
- [27] M. Burset, I. A. Seledtsov and V. V. Solovyev, *Analysis of canonical and non-canonical splice sites in mammalian genomes*, Nucleic Acids Res., 28 (2000), pp. 4364–4375.
- [28] L. D. Stein, Z. Bao, D. Blasiar, T. Blumenthal, M. R. Brent, N. Chen, A. Chinwalla, L. Clarke, C. Clee, A. Coghlan, et al., *The Genome Sequence of Caenorhabditis briggsae: A Platform for Comparative Genomics*, PLoS Biol., 1:2 (2003), e45.