12-1-2003

On Gene Prediction by Cross-Species Comparative Sequenced Analysis.

Rong Chen

On Gene Prediction by Cross-Species Comparative Sequence Analysis

A Thesis

Presented to the

Department of Computer Science

and the

Faculty of the Graduate College

University of Nebraska

in Partial Fulfillment

of the Requirements for the Degree

Master of Science

University of Nebraska at Omaha

by

Rong Chen

December 2003

THESIS ACCEPTANCE

Acceptance for the faculty of the Graduate College,
University of Nebraska, in partial fulfillment of the
requirements for the degree of Master of Science,
University of Nebraska at Omaha.

Committee

Name	Signature
Hesham Ali	
Bruce Chase	Bruce A. Chase
Zhengxin Chen	

Chairperson (signature) ______________ Date 10/31/03

Co-Chairperson (signature) ______________ Date __________

(if applicable)

On Gene Prediction by Cross-Species Comparative Sequence Analysis

Rong Chen, MS

University of Nebraska, 2003

Advisor: Dr. Hesham Ali

Abstract

The availability of large fragments of genomic DNA makes it possible to apply comparative genomics for identification of protein-coding regions. We have conducted a comparative analysis of homologous genomic sequences of organisms with different evolutionary distances and found the conservation of the non-coding regions between closely related organisms. In contrast, more distance shows much less intron similarity but less conservation on the exon structures. We sought to illuminate the impact of evolutionary distances on the performance of our gene-finding program based on the cross-species sequence comparison. Base on our finding and training of data sets, we proposed a model by which coding sequence could be identified by comparing sequences of multiple species, both close and approximately distant. The reliability of the proposed method is evaluated in terms of sensitivity and specificity, and results are compared to those obtained by other popular gene prediction programs. Provided sequences can be found from other species at appropriate evolutionary distances, this approach could be applied in newly sequenced organisms where no species-dependent statistical models are available.

Keywords: Gene Prediction, Genomic Annotation, Sequence Comparison, Multiple-species, Genomic DNA, Exon, Intron, Dynamic Programming, Alignment, LZ Complexity, Evolutionary Distance

ACKNOWLEDGEMENTS

My deep gratitude goes to my advisor, Dr. Hesham Ali, for his continued support, encouragement, inspiration and guidance which made this thesis a reality.

My sincere appreciation also goes to Dr. Zhengxin Chen and Dr. Bruce Chase, for graciously serving on my committee and providing valuable suggestions that helped me improve this thesis.

I would like to thank my parents, Longquan Chen and Jingzhao Dai, and my sister, Guang Chen. Without their support and encouragement, I would not have the opportunity to pursue graduate study in UNO. I am also grateful to my husband, Feng Cui, and my son, Richard Cui, for their love and cooperation which are very important for the accomplishment of this thesis.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1 Biological Overview – The Flow of Genetic Information

1.1 Central Dogma of Molecular Biology

The flow of genetic information in normal cells is from DNA to RNA to protein. The synthesis of RNA from a DNA template is called *transcription*, whereas the synthesis of a protein from an RNA template is termed *translation.* The genetic code is the relation between the sequence of bases in DNA (or its RNA transcript) and the sequence of amino acids in proteins. The flow of genetic information is summarized in the *central dogma* in molecular biology, as shown in Figure 1. This rule was dubbed the "central dogma", because it was thought that the same principle would apply to all organisms [1].

DNA -transcription→ RNA -translation→ Protein

Figure 1. The central dogma of molecular biology, showing the path of flow of genetic information

1.2 Gene, Genome and DNA Library

1.2.1 Genome

Genome is the total genetic material of an organism, contained in a set of chromosomes (in eukaryotes), in a single chromosome (in bacterial), or in a DNA or RNA molecule (in viruses) [2]. Unless specified otherwise, the *genome size* in eukaryotes usually means the *haploid* genome size. That is, only one set of chromosomes is counted. Table 1 shows the genome size and gene number of some organisms.

Table 1. The genomes of prominent organisms.

Organism	**Genome Size** (Mb)	**Gene Number**
Hepatitis D virus	0.0017	1
Hepatitis B virus	0.0032	4
HIV-1	0.0092	9
Bacteriophage l	0.0485	80
Escherichia coli	4.6392	4400
S. cerevisiae (yeast)	12.155	6300
C. elegans (nematode)	97	19000
D. melanogaster (fruit fly)	137	13600
Mus musculus (mouse)	3000	?
Homo sapiens (human)	3000	30000(?)**

* 1 Mb = 1 million base pairs (for double-stranded DNA or RNA) or 1 million bases (for single-stranded DNA or RNA).

** The total number of human genes is still quite controversial. It could be as high as 75,000.

1.2.2 The structure of eukaryotic genes

As shown in figure 2, a typical DNA molecule consists of *genes, pseudogenes* and *extragenic regions.* A gene is a segment of DNA that encodes protein or RNA molecules. Pseudogenes are nonfunctional genes. They often originate from mutation of duplicated genes. Because duplicated genes have many copies, the organism can still survive even if a couple of them become nonfunctional [3].

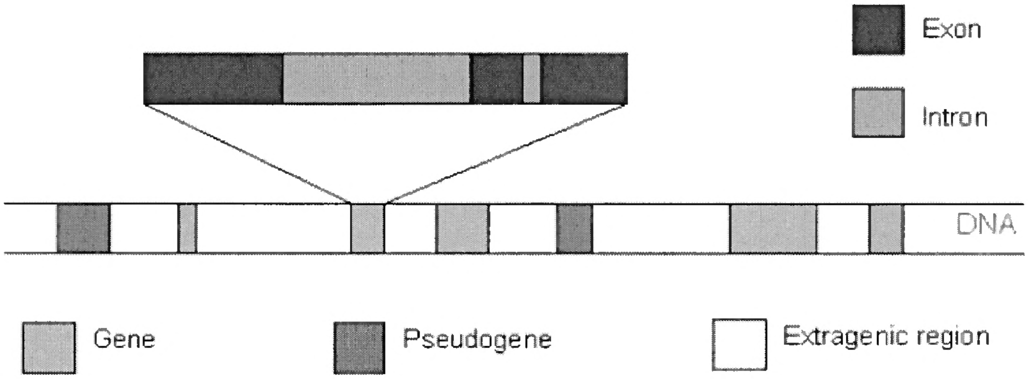


Figure 2. General organization of the DNA sequence. Only the exons encode a functional peptide or RNA. The coding region accounts for about 3% of the total DNA in a human cell.

The genomes of most eukaryotes are larger and more complex than those of prokaryotes. This larger size of eukaryotic genomes is not inherently surprising, since one would expect to find more genes in organisms that are more complex. However, the genome size of many eukaryotes does not appear to be related to genetic complexity [4]. This apparent paradox was resolved by the discovery that the genomes of most eukaryotic cells contain not only functional genes but also large amounts of DNA sequences that do not code for proteins. The presence of large amounts of noncoding sequences is a general property of the genomes of complex eukaryotes [4].

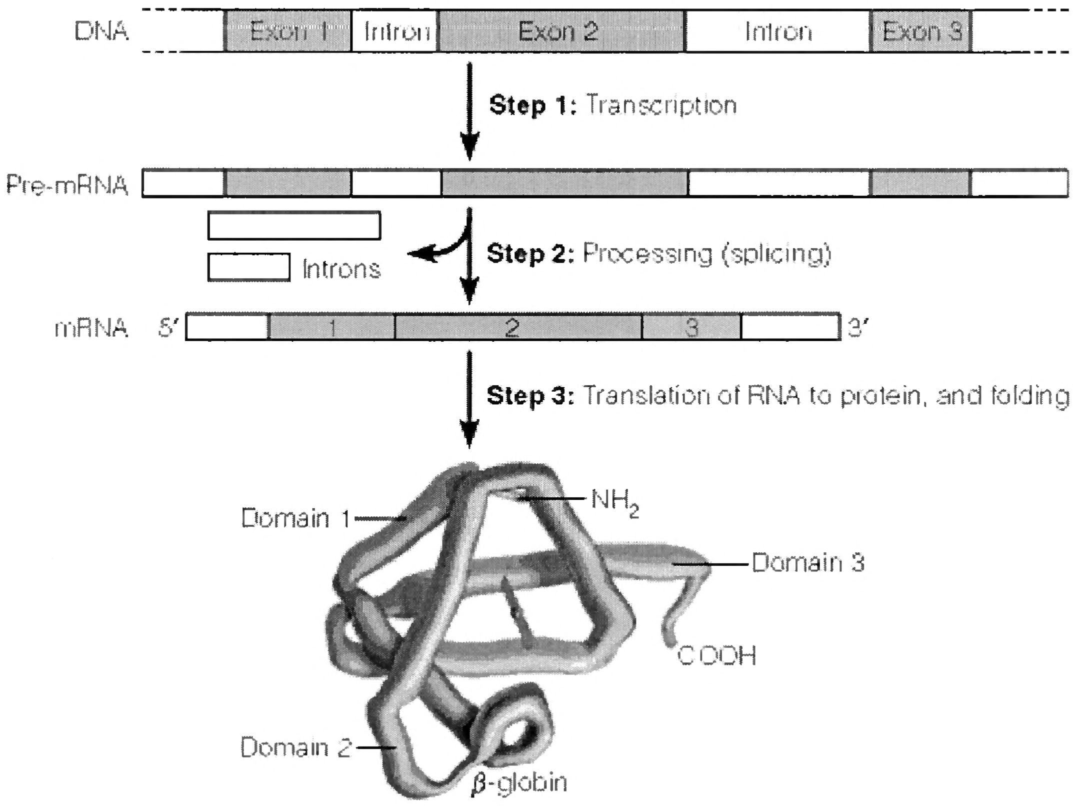


Figure 3. Coding and noncoding regions of the mouse β hemoglobin gene

In molecular terms, a gene can be defined as a segment of DNA that is expressed to yield a functional product, which may be either an RNA (e.g., ribosomal and transfer RNAs) or a polypeptide. Some of the noncoding DNA in eukaryotes is accounted for by long DNA sequences that lie between genes (*spacer sequences* or *extragenic region*, shown in Figure 2). However, large amounts of noncoding DNA are also found within most eukaryotic genes. Such genes have a split structure in which segments of coding sequence (called *exons*) are separated by noncoding sequences (*intervening sequences*, or *introns*). Figure 3 shows the coding and noncoding regions of the mouse β hemoglobin gene. The entire gene is transcribed to yield a long RNA molecule and the introns are then removed by splicing, so only exons are included in the mRNA. Although most introns have no known function, they account for a substantial fraction of DNA in the genomes of higher eukaryotes [3].

1.2.3 DNA Library

DNA library is a collection of cloned DNA fragments. There are two types of DNA library. The *genomic library* contains DNA fragments representing the entire genome of an organism. The *cDNA library* contains only complementary DNA molecules synthesized from mRNA molecules in a cell. The advantage of the cDNA library is that it contains only the coding region of a genome [1].

1.3 Overview of Gene Expression

An organism may contain many types of somatic cells, each with distinct shape and function. However, they all have the same genome. The genes in a genome do not have any effect on cellular functions until they are "*expressed*". Different types of cells express different sets of genes, thereby exhibiting various shapes and functions [4].

Gene expression means the production of a protein or a functional RNA from its gene. Several steps are required (Figure 4) [3]:

- *Transcription*: A DNA strand is used as the *template* to synthesize an RNA strand, which is called the *primary transcript*.

- *RNA processing*: This step involves modifications of the primary transcript to generate a mature mRNA (for protein genes) or a functional tRNA or rRNA.

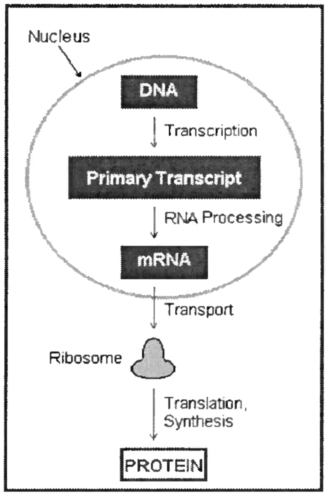


Figure 4. Essential steps involved in the expression of protein genes

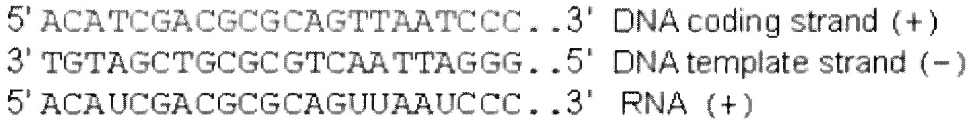
For RNA genes (tRNA and rRNA), the expression is complete after a functional tRNA or rRNA is generated. However, protein genes require additional steps:

- *Nuclear transport*: mRNA has to be transported from the nucleus to the cytoplasm for protein synthesis.

- Protein synthesis: In the cytoplasm, mRNA binds to ribosomes, which can synthesize a polypeptide based on the sequence of mRNA. Amino acids are coded by groups of three bases (called codons) starting from a fixed point. Sixty-one of the 64 codons specify particular amino acids, whereas the other three codons (UAA, UAG, and UGA) are signals for chain termination. Thus, for most amino acids there is more than one code word. In other words, the code is degenerate. Codons specifying the same amino acid are called *synonyms*. Most synonyms differ only in the last base of the triplet. The genetic code is nearly the same in all organisms [1].

1.3.1 Overview of Transcription

Transcription is a process in which one DNA strand is used as a template to synthesize a complementary RNA [3]. The following is an example:



```
5' ACATCGACGCGCAGTTAATCCC..3'  DNA coding strand (+)
3' TGTAGCTGCGCGTCAATTAGGG..5'  DNA template strand (-)
5' ACAUCGACGCGCAGUUAAUCCC..3'  RNA (+)
```

The DNA strand which serves as the template may be called "*template strand*", "*minus strand*", or "*antisense strand*". The other DNA strand may be termed "*non-*

template strand", "*coding strand*", "*plus strand*", or "*sense strand*". Since both DNA coding strand and RNA strand are complementary to the template strand, they have the same sequences except that T in the DNA coding strand is replaced by U in the RNA strand.

Growth of a nucleic acid strand is always in the 5' to 3' direction. This is true not only for the synthesis of RNA during transcription, but also for the synthesis of DNA during replication. *Downstream* refers to the direction of transcription and *upstream* is opposite to the transcription direction. The numbering of base pairs in the transcription initiation region is as follows. The number increases along the direction of transcription, with "+1" assigned for the initiation site. There is no "0" position. The base pair just upstream of +1 is numbered "-1", not "0". The enzymes, called *polymerases*, are used to catalyze the synthesis of nucleic acid strands. RNA strands are synthesized by RNA polymerases. DNA strands are synthesized by DNA polymerases. The entire transcription process should involve the following essential steps [1, 3]:

1) Binding of polymerases to the initiation site. The DNA sequence which signals the initiation of transcription is called the *promoter***.** Prokaryotic polymerases can recognize the promoter and bind to it directly, but eukaryotic polymerases have to rely on other proteins called *transcription factors*.

2) Unwinding (melting) of the DNA double helix. The enzyme which can unwind the double helix is called *helicase.* Prokaryotic polymerases have the helicase activity, but eukaryotic polymerases do not. Unwinding of eukaryotic DNA is carried out by a specific transcription factor.

3) Synthesis of RNA based on the sequence of the DNA template strand. RNA polymerases use nucleoside triphosphates (NTPs) to construct an RNA strand.

4) Termination of synthesis. Prokaryotes and eukaryotes use different signals to terminate transcription.

Transcription in eukaryotes is much more complicated than in prokaryotes, partly because eukaryotic DNA is associated with histones, which could hinder the access of polymerases to the promoter.

Transcriptional regulation is mediated by the interaction between transcription factors and their DNA binding sites which are the *cis-acting* elements, whereas the sequences encoding transcription factors are *trans-acting* elements. The regulatory elements include promoter, response element, enhancer and silencer [3].

1.3.2 RNA Processing

RNA processing is to generate a mature mRNA (for protein genes) or a functional tRNA or rRNA from the *primary transcript.* Processing of pre-mRNA involves the following steps [3]:

1. *Capping* - adding 7-methylguanylate (m^7G) to the 5' end. Capping occurs shortly after transcription begins. m^7G is linked to the first nucleotide by a special 5'-5' triphosphate linkage. In most organisms, the first nucleotide is methylated at the 2'-hydroxyl of the ribose. In vertebrates, the second nucleotide is also methylated.

2. *Polyadenylation* - adding a poly-A tail to the 3' end. A stretch of adenylate residues is added to the 3' end. The poly-A tail contains ~ 250 A residues in mammals, and ~ 100 in yeasts.

3. *Splicing* - removing introns and joining exons. RNA splicing is a process that removes introns and joins exons in a primary transcript. An intron usually contains a clear signal for splicing. In some cases, a splicing signal may be masked by a regulatory protein, resulting in *alternative splicing*. In rare cases (e.g., HIV genes), a pre-mRNA may contain several ambiguous splicing signals, resulting in a few alternatively spliced mRNAs.

   Most introns start from the sequence *GU* and end with the sequence *AG* (in the 5' to 3' direction). They are referred to as the *splice donor* and *splice acceptor* site, respectively. In over 60% of cases, the exon sequence is (A/C)AG at the donor site, and G at the acceptor site (Figure 6).

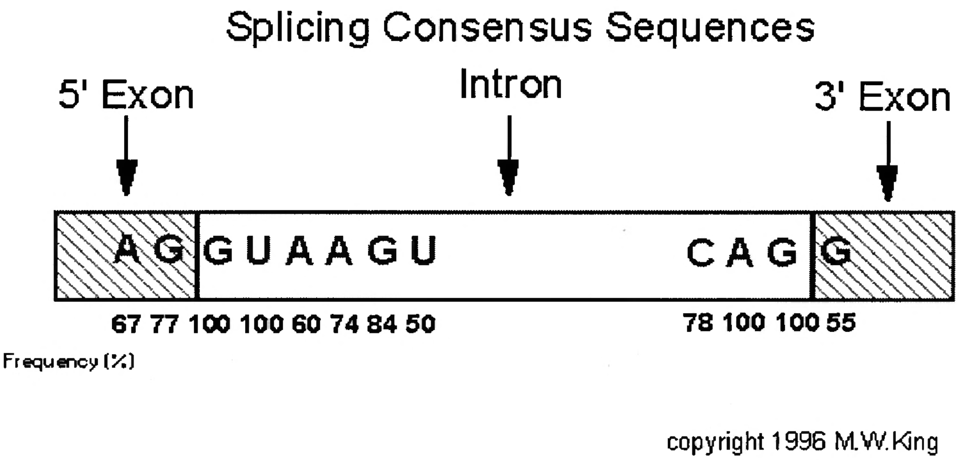


Figure 5. The consensus sequence for splicing

Chapter 2 Gene Prediction

The sequencing of the entire genomes of many organisms of both plant and animal is complete now. The draft version of the human genome sequence is also produced [5]. However, the annotation of the sequences is not meeting the needs for studying the function of a newly sequenced genome. Unfortunately, finding genes in a genomic sequence is far from being a trivial problem [6].

This chapter will provide a general overview of the classical gene-finding approaches, and an analysis of the strength and problem of the currently available gene-prediction methods.

2.1 Problem Description – Genome Annotation in Eukaryote

Genome annotations are features on the genome derived through the transformation of raw genomic sequences into information by integrating computational tools, auxiliary biological data, and biological knowledge.

In this work, we consider the problem of finding genes coding for a protein sequence in eukaryotes only. The problem of finding genes in prokaryotes presents different types of difficulties (there are no introns and the intergenic regions are small, but genes may often overlap each other and the translation starts are difficult to predict correctly) [7].

Functionally, a eukaryotic gene can be defined as being composed of a transcribed region and of regions that cis-regulate the gene expression, such as the promoter region which is mostly found in the 5' part of the gene and controls transcription. The currently existing gene-prediction software looks only for the transcribed region of genes, which is then called '*the gene*'. This definition of a gene is adopted in this work; the region between two transcribed regions will be called *intergenic*. In the current practice, the promoter is seen as sitting in the intergenic region, immediately upstream of the gene and not overlapping with it. This is also a simplification of reality [7].

A gene is further divided into exons and introns, the latter being removed during the splicing mechanism that leads to the mature mRNA. Although some exons (or parts of them) may be non-coding, most gene-finding software uses the term exon to denote the coding part of the exons only. In this work will adopt this definition and regard the gene finding as the identification of CDs only. Indeed, in the mature mRNA, the untranslated terminal regions (*UTRs*) are the non-coding transcribed regions, which are located upstream of the translation initiation (*5'-UTR*) and downstream (*3'-UTR*) of the translation stop (Figure 6) [3]. They are known to play a role in the post-transcriptional regulation of gene expression, e.g. the regulation of translation and the control of mRNA decay [7]. Inside or at the boundaries of the various genomic regions, specific functional sites (or signals) are documented to be involved in the various levels of protein encoding gene expression, e.g. transcription (transcription factor binding sites and TATA boxes), splicing (donor and acceptor sites and branch points), polyadenylation [poly(A) site], translation (initiation site, generally ATG with exceptions, and stop codons) [1, 3, 7].

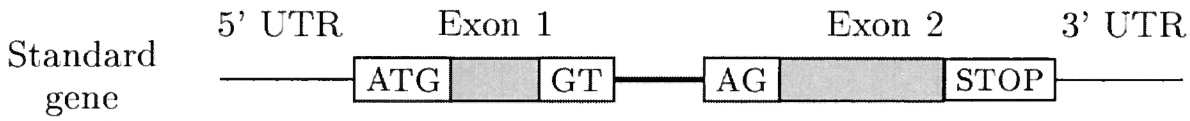


Figure 6. A simplified gene structure for prediction

2.2 Classic Gene-Finding Approaches

There are two important aspects to any program for gene identification: one is finding evidence for a gene, and the other is the algorithm that is employed to combine the evidence into a coherent prediction.

2.2.1 Types of information used – Finding the evidence

Three types of information are used in predicting gene structures: "signals" in the sequence, such as splice sites; "content" statistics, such as codon bias; and “similarity” to known genes. The first two types have been used since the early days of gene prediction, whereas similarity information has been used routinely only in recent years. One of the reasons that the accuracy of gene-prediction programs has improved in the last few years is the enormous increase in the number of examples of known coding sequences. This much larger sample size allows for more reliable statistical measures to be developed, as well as a much greater likelihood of encountering a gene that is related to one that has been identified previously [8].

2.2.1.1 Signals

The basic and natural approach to finding a signal that may represent the presence of a functional site is to search for a match with a consensus sequence, the consensus being determined from a multiple alignment of functionally related documented sequences. This type of method is used, for instance, for splice sites prediction in some gene-finding programs [7].

The splice junctions—the donor and acceptor sites—are the most important features to identify. If these could be reliably detected from the genomic DNA, the difficulty in identifying the coding regions would be greatly reduced because most genes could be recognized simply by finding the long open reading frames (*ORFs*). It would still be somewhat more difficult than for prokaryotes simply because genes are much less dense in eukaryotes, but a high degree of accuracy could be obtained easily. Unfortunately, splice junctions are not reliably detectable in the genomic sequence. The most common method for predicting them has been the PWMs, which are separate weight matrices for acceptor and donor sites, and the scores for each base depend on the frequencies of each base at each position in the known sites, which employ various methods including simply using the logarithm of the frequency, or using a log-odds ratio between the frequency of each base in the collection of sites and the expected frequency of that base in the genome [8].

Using weight matrices to identify donor and acceptor sites is much more reliable than a consensus sequence but still predicts a large excess over the correct sites; that is, there are many false positives for every true positive prediction. More complicated site descriptors have also been tried. For example, "weight array matrix," which has a score for each dinucleotide and thereby takes into account the non-independence of adjacent positions in the sites, is a maximal dependence decomposition (MDD) method by Burge and Karlin [9] for representing splice sites. In addition, neural networks have been employed to detect splice sites. Neural networks are a pattern recognition technique that takes as input positive and negative examples (i.e., true splice sites and similar sites that are not functional splice sites) and discover the features that distinguish the two sets. The essential distinguishing features may include correlations in the positions of the sites [8]. Study shows that if one can narrow the region in which a splice site is expected to occur, then the accuracy increases significantly. This means that the unreliable splice site prediction methods can be combined with methods for identifying exons based on content measures and increase the exon prediction accuracies significantly [9].

Other signals can also be useful in predicting exons. The start and stop codons are essential in predicting the correct gene. Unfortunately, they are fairly uninformative without knowledge of the reading frame. But they are essential in categorizing exons into four classes: single exon genes that begin with a start codon and end with a stop codon; initial exons that begin with a start codon and end with a donor site; terminal exons that begin with an acceptor site and end with a termination codon; and internal exons that begin with an acceptor site and end with a donor site [9]. Initial and terminal exons tend to be the most difficult to identify, both because the signals are less informative and because they are often much shorter than internal exons and therefore harder to identify by content measures [8].

Some programs also look for sites associated with promoters, such as TATA boxes, transcription factor binding sites, and CpG islands. Identifying promoters can sometimes add information that is useful for predicting genes. Poly A addition signals are also used sometimes to aid in identifying the proper carboxyl terminus of the gene. In general, the use of these other types of signals provides a marginal improvement over methods that do not use them [8].

2.2.1.2 Content Measures

Coding regions have statistical properties that can help to distinguish them from non-coding regions. In prokaryotes, the length of most coding ORFs is statistically significant. In eukaryotes, the lengths of typical exons are not especially significant, but they have other properties that are useful. For example, every species employs a bias in its choice of codons, such that synonymous codons are not used with the same frequency. So knowing the codon bias for a species can help to identify the genes from the DNA sequence. Other statistical tests have also been applied to the problem of distinguishing coding from noncoding sequences based on their sequences, such as nucleotide composition and especially (G+C) content (introns being more A/T-rich than exons, especially in plants), codon composition, hexamer frequency, base occurrence periodicity, etc [7]. Neural networks have also been used to distinguish coding from noncoding sequences. The network was trained to classify whether a particular nucleotide was coding or not based on the surrounding nucleotides, using regions of 100-400 bases [8]. In general, the strengths of these content measures increases with the length of the exons, so that long exons are fairly easy to identify whereas short ones remain difficult even after applying these tests [7].

2.2.1.3 Similarity Measures

A region of genomic DNA that is significantly similar to a known sequence will usually have the same, or very similar, function [8]. This can be used as both positive and negative evidence about the coding likelihood of the region. For example, if the region matches well to a known repetitive sequence it is unlikely to be protein coding.

If a region of DNA is similar, after translation, to a known protein or protein family, that is strong evidence that the region codes for a protein, and even gives you information about its likely function. This information has been used for a long time to compare the predicted genes with protein databases and provide added confidence for predictions with matches [7, 8]. When a region of genomic DNA matches a sequenced cDNA, that is very strong evidence that it is transcribed and likely to be part of a coding region. The same can be said of expressed sequence tags (ESTs) databases, which are one-shot sequences from a whole cDNA, although they tend to contain more artifacts that can be misleading. In general, similarities between genomic DNA and sequences that correspond to genes, whether from protein, cDNA, or EST databases, can provide useful evidence for the occurrence of protein coding regions [8].

2.2.2 Combing the evidence to predict gene structures

Given a sequence and using signal sensors, one can accumulate evidence on the occurrence of signals: translation starts and stops and splice sites are the most important ones since they define the boundaries of coding regions. In theory, each consistent pair of detected signals defines a potential gene region (intron, exon or coding part of an exon). If one considers that all these potential gene regions can be used to build a gene model, the number of potential gene models grows exponentially with the number of predicted exons. In practice, this is slightly reduced by the fact that 'correct' gene structures must satisfy a set of properties [7]:

(i) there are no overlapping exons;

(ii) coding exons must be frame compatible;

(iii) Merging two successive coding exons will not generate an in-frame stop at the junction.

The number of candidates remains, however, exponential. In almost all existing approaches, such an exponential number is coped with in reasonable time by using dynamic programming techniques.

Until recently, prediction methods that try to determine the whole gene structure, i.e. to assemble all the pieces, could be separated into two classes depending on whether the content of exon/intron regions was assessed using extrinsic or intrinsic content sensors.

2.2.2.1 Extrinsic approaches

Much software based on similarity searches has emerged during the last few years. One of the main weaknesses of the pure similarity-based content sensors is that the limits of similarities are never accurately defined. The principle of most of these programs is to combine similarity information with signal information obtained by signal sensors. This information will be used to refine the region boundaries [8]. These programs inherit all the strengths and weaknesses of the sensors used.

All the programs in this class may be seen as sophistications of the traditional Smith–Waterman local alignment algorithm where the existence of a signal allows for the opening (donor) or closure (acceptor) of a gap with an essentially free extension cost. They are often referred to as 'spliced alignment' programs [8]. Existing software may be further divided according to the type of similarity exploited: genomic DNA/protein, genomic DNA/cDNA or genomic DNA/genomic DNA [7]. Some of these methods are able to deal with more than one type and to take into account possible frame shifts in the genomic DNA or cDNA sequences.

In the methods that align a genomic sequence with a protein such as Procrustes, the selection of the target protein may be retrieved from a BLASTX search. One can consider all potential exons from the query DNA sequence, initially with the only constraint that they must be bordered by donor and acceptor sites. All possible exon assemblies are explored by translating the exons and aligning them with the target protein, using the PAM120 matrix for scoring mismatches. This is done in a time proportional to the product of the lengths of the query and target sequences. As a result, it produces an assembly with the highest similarity score to the target protein [6, 8].

Other available programs including AAT, GeneSeqer, SIM4 and Spidey, perform an alignment of the genomic DNA sequence against a cDNA database. This is a very reliable way of identifying exons, independently of their coding status, especially when the genomic sequence is aligned against a cDNA from the same (or a close) organism [7]. Difficulties may, however, be encountered when trying to delineate the UTR part of the genes, and thus the correct translation initiator and stop codons [7, 8].

The approach adopted is rather different for programs which try to elucidate the gene structure from EST matches, like EbEST, Est2genome, TAP and PAGAN. The reason for this comes from the specific nature of an EST. A first characteristic of ESTs is that they are very redundant and a large number of them may be retrieved when performing a BLAST search against dbEST. EbEST faces this problem in its first step by clustering ESTs into non-overlapping groups and then by selecting the most informative ESTs within each group. A second characteristic of ESTs is that they are naturally error prone since they are generated from single-read sequences. The Smith-Waterman algorithm used in EbEST tolerates the presence of such errors. Another characteristic of ESTs is that most of them are 3' ESTs generated from oligo(dT)-primed cDNA libraries and are therefore useful for detecting the 3'-UTRs in long sequences. This last point is an important added value for the EST-driven gene modeling approaches, as it leads to a rather confident prediction of a gene 3'-end. This importance may be somewhat weakened by the fact that ESTs represent only partial mRNA sequences and even clusters of ESTs may not lead to the complete identification of the gene structure [6, 7, 8].

In all cases, an important strength of similarity-based approaches is that predictions rely on accumulated pre-existing biological data (with the caveat, mentioned later, of possible poor database quality). They should thus produce biologically relevant predictions (even if only partial) [7].

2.2.2.2 Intrinsic approaches

Unlike most of the "spliced alignment" approaches described in the previous section, which aim at producing a gene structure based on similarities to known sequences, intrinsic gene finders aim at locating all the gene elements that occur in a genomic sequence, including possible partial gene structures at the border of the sequence.

To efficiently deal with the exponential number of possible gene structures defined by potential signals, almost all intrinsic gene finders use dynamic programming to identify the most likely gene structures according to the evidence defined by both content and signal sensors [8].

The problem of having too many models to exhaustively enumerate and analyze was solved by the application of dynamic programming methods. These are recursive optimization techniques that are guaranteed to find the highest scoring prediction without examining all possible ones. In the context of gene prediction it uses the grammatical rules of gene structure. These are the constraints on the order in which different segments can occur. For example, an initial exon must occur before any introns, an internal exon is bounded on both sides by introns, and a terminal exon is preceded by an intron but not followed by one. Given these constraints and a collection of potential exons and introns, each with an associated score, it is possible to scan across the sequence once and determine the highest scoring predicted gene structure. For example, when considering some particular internal exon candidate, one has to account only for the highest scoring solution that ends with an intron preceding it, and not all possible solutions. So by starting at one end and keeping track of the best solutions ending with each potential exon or intron, the most preferable solution is guaranteed to be found quickly. Most of the previously mentioned gene-finding methods added a dynamic programming step to combine features and determine the optimum predictions based on their own scoring systems. While these methods are guaranteed to find the highest scoring predictions, they do not always find the correct predictions. In the earliest adaptation of dynamic programming methods the overall prediction accuracy was not much improved over the previous methods [9]. But because that approach was guaranteed to efficiently find the highest scoring prediction given the scoring system and the information used, researchers could focus their efforts on identifying more useful types of information and improving the scores assigned to them [8].

In the signal-based methods, the gene assembly is produced directly from the set of detected signals. In the simplest signal-based methods, there is an implicit assumption that the 'content' score of a segment is defined as the sum of the local (nucleotide-based) content scores and therefore does not depend on the global characteristics of the segment. This is, for instance, the case in basic Hidden Markov Models (HMMs) [7].

An HMM has several "states," and in gene prediction these correspond to exons, introns, and any other classes of sequences desired (such as 5' and 3' UTRs, promoter regions, intergenic regions, repetitive DNA, etc.). There are probabilities for transitions between the different states that correspond to the allowed changes in state, for example, an intron can only be followed by an internal exon or a terminal exon. The probability of changing from an intron to an exon depends on the local sequence such that it is high only at plausible splice junctions. While the HMM is in any particular state, it "emits" a DNA sequence, which is visible. The "hidden" in HMMs denotes the fact that we see only the DNA sequence directly, and the state that generated the sequence (exon, intron, etc) is not visible. But the different states emit DNA with different characteristics. For example, exons emit DNA that must have an ORF, tends to have a certain codon bias, tends to have a certain length distribution, etc. DNA emitted by intron states has different characteristics.

All parameters of the model are probabilities. There are transition probabilities between the states, and emission probabilities from the states. Any "parse" of a sequence (i.e., assignment of its bases to specific states) has an associated probability. The probabilities of any two, or more, parses can be compared directly. Most importantly, there are efficient methods, usually dynamic programming, to perform all of the essential tasks. Given a collection of known correct parses, that is, examples where we know both the genomic DNA and the correct assignment of each nucleotide to its proper class (state), we can find the set of parameters (the probabilities of the model) that maximize the probability of those example sequences. So a "training set" of correct examples is sufficient (but the more examples the better) to find the optimal values of the parameters. Then those parameters are sufficient to determine the optimum (highest probability) parse of any new sequence [7].

Chapter 3 Comparative Genomics and Its Application in Gene Prediction

Comparative genomics is the analysis and comparison of genomes from different species. The purpose is to gain a better understanding of how species have evolved and to determine the function of genes and noncoding regions of the genome. For instance, researchers have learned a great deal about the function of human genes by examining their counterparts in simpler model organisms such as the mouse. Genome researchers look at many different features when comparing genomes: sequence similarity, gene location, the length and number of coding regions (called exons) within genes, the amount of noncoding DNA in each genome, and highly conserved regions maintained in organisms as simple as bacteria and as complex as humans [10].

Comparative genomics involves the use of computer programs that can line up multiple genomes and look for regions of similarity among them. This research will focus on the application of comparative genomics in the identification of protein-coding regions.

3.1 Idea of gene reorganization by comparative genomics

The similarity-based and statistical gene-finding methods are useful, but each of them has its limitations. Although the specificity and sensitivity of individual predictions by statistical methods has increased, especially due to the appearance of hidden Markov model algorithms such as GenScan [8], their reliability is yet insufficient, especially in the context of genome projects generating long multi-gene fragments. Similarity-based methods are the most reliable if a sufficiently close protein is available, but they cannot be used for genes encoding new proteins. EST-based algorithms have problems with artifactual ESTs and they cannot help in analysis of genes not represented in clone libraries because of narrow stage and tissue specificity of these genes [11].

Sequencing of large fragments of genomic DNA, and even complete eukaryotic chromosomes, makes it possible to apply comparison of genomic sequences for identification of protein-coding regions [11]. This approach is based on the assumption that coding sequences are more conserved than non-coding ones, similarity with genomic DNA can also be a valuable source of information on exon/intron location. Two approaches are possible: intra-genomic comparisons can provide data for multi-genic families, apparently representing a large percentage of the existing genes (e.g. 80% for Arabidopsis); inter-genomic (cross-species) comparisons can allow the identification of orthologous genes, even without any preliminary knowledge of them. In this case, candidate exons are seen as islands of similarity in alignment of genomic sequences harboring homologous genes [7].

3.2 Related work

Several programs tried to retrieve information on conservation or synteny between organisms from genomic alignments, as, for example, MUMmer, WABA, PipMaker and DIALIGN [7]. Recently, a fcw algorithms havc appcared that focus more specifically on the gene recognition problem by comparison of two genomic sequences based on the hypothesis that coding DNA sequences are more conserved than non-coding sequences (intronic and intergenic). Comparing two homologous genomic sequences (cross- or intra-species) should thus help to reveal conserved exons and allow the prediction of genes simultaneously on both sequences.

3.2.1 ROSETTA [12]

ROSETTA is the first program for gene recognition based on cross-species comparison of genomic DNA from two organisms. It developed algorithms for cross-species gene recognition, consisting of GLASS, a new alignment program designed to provide good global alignments of large genomic regions by using a hierarchical alignment approach, and ROSETTA, a program that identifies coding exons in both species based on coincidence of genomic structure (splice sites, exon number, exon length, coding frame, and sequence similarity). ROSETTA had 95% sensitivity and 97% specificity at the nucleotide level.

ROSETTA is specifically designed for the comparison of closely related species. In particular, it make the hypothesis of conserved exon-intron structure in the two sequences and the further (very strong) hypothesis that the corresponding exons in the two genes have roughly the same length.

3.2.2 SGP (Syntenic Gene Prediction) [14]

SGP-1 is a similarity-based gene-prediction program. Given two genomic DNA sequences it post-processes the pair-wise local alignment to predict single or multiple gene models of protein coding genes in forward and reverse strands. In contrast to ROSETTA, the accuracy of SGP-1 depends little on species-specific properties such as codon usage or the nucleotide distribution. SGP-1 may therefore be applied to nonstandard model organisms in vertebrates as well as in plants, without the need for extensive parameter training.

3.2.3 Pro-Gen [13]

Pro-gen finds in each sequence an exon chain with the maximum similarity on the protein level. Unlike other algorithms, Pro-Gen does not assume conservation of the exon-intron structure and thus can be applied to analysis of relatively distant homologs. Amino acid sequences obtained by the formal translation of candidate exons are aligned instead of nucleotide sequences, which allows for distant comparisons.

When the algorithm was tested on a sample of human-mammal (mouse), human-vertebrate (Xenopus) and human-invertebrate (Drosophila) gene pairs, the best results, 97-98% correlation between the actual and predicted genes, were obtained for more distant comparisons, whereas the correlation on the human-mouse sample was only 93%. The latter value increases to 95% if conservation of the exon-intron structure is assumed. This is caused by a large amount of sequence conservation in non-coding regions of the human and mouse genes probably due to regulatory elements.

3.2.4 AgenDA [11]

They use a new version of the DIALIGN alignment program to get a pair of syntenic sequences from evolutionary related organisms. This program integrates local and global features by assembling pair-wise and multiple alignments from gap-free local segment alignments (so-called fragments). They compare these and identify a chain of local sequence similarities. Then, it searches for conserved splice signals and start or stop codons near the boundaries of the identified sequence similarities. Finally, local homologies that are bounded by conserved splice signals are chained together in a biologically consistent way.

3.3 Challenging problems in this gene-finding approach

All the comparative genomic methods have, theoretically, the advantage of not being species-specific. In practice, performance will depend on the evolutionary distance between the compared sequences. Previous results show that the relationship is not straightforward. Indeed, a greater evolutionary distance allows some algorithms to more accurately discriminate between coding and non-coding sequence conservation. Such programs are often computer intensive and consequently much work remains to be done. In particular, a major challenge that could considerably improve the performance of gene-finding programs would be to introduce multiple comparisons into these methods [7].

Moreover, the similarity might not cover entire coding exons but might be limited to the most conserved part of them. Alternatively, it may sometimes extend to introns and/or to the UTRs and promoter elements. This will be the case when genomes are evolutionarily close or when genome duplications are recent events. In both cases, exactly discriminating between coding and non-coding sequences is not an obvious task [7]. It is therefore necessary to take more information into account to identify conserved gene structures in syntenic genome sequences [8].

In this work, the following research is done on the gene prediction by cross-species comparative sequence analysis:

1) To conduct a comparative analysis of homologous genomic sequences of organisms with different evolutionary distances to find the conservation of the

coding regions as well as the non-coding regions between organisms with different distances.

2) To create a model by which protein-coding regions could be identified by comparison of genomic sequences of two species, using new sequence comparison method combining local alignment and LZ complexity.

3) To find the impact of evolutionary distance on the performance of the gene-finding program based on the two-species sequence comparison.

4) To propose a model by introducing the third species, with less close distance, and by which coding sequence could be identified by comparing sequences of multiple species, leading to better performance of prediction.

Chapter 4 Comparison of Genomic Structures with Different Evolutionary Distances

4.1 Data Resource

4.1.1 The selection of organisms

It is critical to find out the extent of conservation of genomic structures of different species before implementation any gene-finding methods based on sequence comparison of organisms. *mus musculus* (mouse), *gallus gallus* (chicken), *xenopus laevis* (frog) and *drosophila melanogaster* (fruit fly) are chosen for analysis of relationship with homo sapiens (human). The selection of species is based on the consideration of each organism's evolutionary distance with *homo sapiens*, the genome sequence availability and the results of previous work.

mus musculus is most frequently used as a "model organism" in functional genomics research. Mice and humans (indeed, most or all mammals including dogs, cats, rabbits, monkeys, and apes) have roughly the same number of nucleotides in their genomes – about 3 billion base pairs. This comparable DNA content implies that all mammals contain more or less the same number of genes, and previous work has provided evidence to confirm that notion [10]. Mouse is also the organism many authors used to develop their approaches to gene-finding by comparative sequence analysis, as described in the last chapter.

Fruit fly *drosophila melanogaster* was a crucial model organism in research in developmental biology and genetics in the early twentieth century. The fruit fly genome was completed in 2000. As an invertebrate, the genome of *drosophila* is different from that of the human in size, number of genes, and gene intensity. However, research shows that *drosophila* contains similar kind of genes and gene expression mechanisms existed in vertebrate [4]. Thus, in this work, *drosophila* is selected as an example of long-distance with human.

Unlike *drosophila* and *mus musculus*, the genomes of chicken *gallus gallus* and frog *xenopus laevis* are still incomplete. However, many sequences homologous to genes of other organisms including human and mouse genes have been identified. These two vertebrates, whose evolutionary distances with human genes are closer than *drosophila* but more distance than *mus musculus*, are analyzed to illuminate the relationship between conservation of gene structure among organisms and evolutionary distances.

4.1.2 Compiling of data set

In the work of Batzoglou *et al.* [12], they compiled 117 orthologous human-mouse gene pairs. According to the authors, these sequences are carefully annotated so they can be considered as a standard of truth [11, 12]. The gene pairs are listed in Table 8 of Appendix. This set of data was also used by Rinner *et al.* in testing their AgenDA program [11]. This data set will be used in this work, because it will be helpful in comparing the results to other authors'. Furthermore, this set of data is expanded to include the homologous sequences from the other species (*gallus*, *xenopus* and *drosophila)*, by TBLASTX, which compares the six-frame translations of a nucleotide query sequence against the six-frame translations of a nucleotide sequence database. GenBank release 137.0 was searched against to get the complete data set.

GenBank is the NIH genetic sequence database, an annotated collection of all publicly available DNA sequences. There are approximately 28,507,990,166 bases in 22,318,883 sequence records as of January 2003. A new release is made every two months. GenBank is part of the *International Nucleotide Sequence Database Collaboration*, which comprises the DNA DataBank of Japan (*DDBJ*), the European Molecular Biology Laboratory (*EMBL*), and GenBank at *NCBI.* These three organizations exchange data on a daily basis. TBLASTX searches data from all the non-redundant databases of GenBank, RefSeq Nucleotides, EMBL, DDBJ and PDB (sequences derived from the 3-dimensional structure from *Brookhaven Protein Data Bank*), but no EST, STS, GSS, or phase 0, 1 or 2 HTGS sequences [15].

The original BLAST program was developed at NCBI. The BLAST (*Basic Local Alignment Search Tool*) program uses a strategy based on matching sequence fragments by employing a powerful statistical model, developed by Samuel Karlin and Stephen Altschul [16], to find the local alignments. BLAST does not necessarily find the best local alignment, but the algorithm does find reasonable answers quickly.

In summary, the following arc the steps of the BLAST algorithm [15]:

- Split the query sequence into all possible words of length *w*. Compile a list of each word that, relative to a *w*-mer from the query sequence, has a score greater than or equal to a certain threshold parameter *T* according to some PAM (Point Accepted Mutations) scoring matrix.

- Scan the database for seeds, or matches with words from the list of the previous step.

- Extend each of the seeds in both directions until reaching a maximum score according to a PAM matrix. Report all segment pairs with scores above some threshold *S*.

TBLASTX is the NCBI BLAST program for comparing a 6-frame translated nucleotide query sequence to a 6-frame translated nucleotide database. BLOSUM62 matrix is used in search against *gallus*, *xenopus* and *drosophila* databases (The BLOSUM matrix assigns a probability score for each position in an alignment that is based on the frequency with which that substitution is known to occur among consensus blocks within related proteins). The word size is set to 3, cost for gap existence is 11, for gap extension is 1.

4.2 Comparative studies of gene structures

To compare the sequences of human vs. its homologs in each of the other species, dynamic programming is implemented to find the global alignment of each pair of homologous genes. The scores for column scores as alignment are defined as following: +1 for a match, -1 for a mismatch, -2 for a space in one of the corresponding positions. The similarity matrix is computed as:

$$\begin{aligned} a[i][j] = \max \{ & a[i-1][j-1] + p(s[i], t[j]), \\ & a[i-1][j] + (-2), \\ & a[i][j-1] + (-2) \} \end{aligned}$$

where

$$p(s[i], t[j]) = \begin{cases} +1, & \text{if } s[i] = t[j] \\ -1, & \text{if } s[i] \neq t[j] \end{cases}$$

The method and all remaining work are implemented in Visual C++ 6.0.

4.2.1 The comparison of exon structures

4.2.1.1 Exon number

The number of exons in human and mouse are well conserved. Results are not totally the same as reported by Batzoglou *et al.* on the same data. Among all of the human-mouse gene pairs, 96% have the same number of exons. Only five gene pairs have different exon numbers, instead of six as Batzoglou *et al.* reported [12].

In two cases (genes 30 and 85), a single internal coding exon in mouse corresponds to two internal coding exons in human, with the total exonic lengths agreeing perfectly or differing in a multiply of 3. In the other three cases, the correspondence broke down for terminal exons. In genes 40 and 67, at the 3'-end one organism has an extra exon. In gene 46, the extra human exon shows striking sequence similarity to a portion of the 3'-untranslated region (UTR) in the mouse.

For human-chicken gene pairs, 77% are identical in exon number. Only 30% of human-frog gene pairs and 28% of human-fruit fly gene pairs have identical exon numbers. Among the gene pairs with different exon numbers, about half (56%) have difference of length greater than 1.

4.2.1.2 Exon length

The length of corresponding exons was strongly conserved in human and mouse genes. The lengths were identical in 74% of pairs. Those differences that did occur were quite small: the mean ratio of the larger to smaller length was 1.05. Moreover, the differences were a multiple of three in most exons (except three cases) with difference in length. The biological meaning under this phenomenon is that length differences other than divisible by three would alter the translational reading frame and would thus be less likely under the effects of evolutionary selection.

The number and length of exons are more divergent in human-frog and human-fruit fly gene pairs. 45% of human exons cannot find the aligned regions in frog sequences, for another 20% exons, only a small portion (<50%) align with part or whole frog exon.

Most human and fruit fly exons are totally differently organized, with many cases of multiple human exons corresponding to single fruit fly exons.

4.2.1.3 Exon sequence similarity

In human and mouse gene pairs, coding regions showed strong sequence similarity, with approximately 85% identity. For chicken, frog and fruit fly, although the number and length of exons are much divergent, among the conserved exon regions, no explicit difference exists in the sequence analysis. The nucleotide level similarity is ranged from 80-85%, while the protein level similarity is above 90%. This provides the evidence that sequence conservation indicates the functional inheritance.

4.2.2 The comparison of intron structures

While exon lengths tended to be preserved in different degree among sequences of various distance, intron lengths varied considerably. Even in human and mouse gene pairs, the mean ratio of the larger to the small length was 1.5. Moreover, there is no tendency for intron lengths to differ by a multiple of three.

Human and mouse introns overall showed only weak sequence similarity with approximately 35% sequence identity, which is not much higher than the background rate of sequence identity in gapped alignments of random sequences. However, some cases exhibited a striking degree of similarity (up to 75%) in conserved intron areas of human and mouse sequences, in non-coding exons (UTR), intergenic intron and internal intron regions. In section 4.3, further global view of conservation of gene structure will be illustrated through a multiple alignment.

The non-coding regions of chicken, frog and fruit fly genes show little conservation compared to human non-coding regions. The total intron similarity is around 30% (Table 2), which is just a background rate of sequence identity.

4.2.3 Summary

Human-mouse gene pairs have the strongest conservation in exon structure. However, also non-coding regions are found conserved in many cases. Therefore, further species could be used together to discriminate coding and non-coding regions.

Table 2. The comparison of gene structures of species with different evolutionary distances

Species compared	Percentage of pairs identical in exon numbers	Percentage of pairs identical in exon length	Conserved Exon similarity (%)	Intron similarity(%)
homo-mus	96	75	85	Up to 75
homo-gallus	77	52	81	33
homo-xenopus	30	45	79	29
homo- drosophila	28	34	79	32

Human and chicken still keep relatively strong conservation in exon structures, although less than that of human and mouse. Compared to the strong conservation of exon-intron structure in human-mouse and chicken, evolutionary distances greater than chicken and human show much less conservation in exon structure.

4.3 The conservation of gene structures among species

Multiple alignment of the data set was done with VISTA, a global alignment tool which is based on moving a user-specified window over the entire alignment, thus allowing for easy identification of conserved regions in the global perspective. The VISTA plot (Figure 7) is based on moving a user-specified window over the entire alignment and calculating the percent identity over the window at each base pair. The x-axis represents the base sequence; the y-axis represents the percent identity [17].

It is found that most conserved regions (85%) correspond to the coding regions. With the evolutionary distance increasing, especially greater than homo sapiens and gallus,

Table 3. VISTA output for alignment of homologous gene in five species, Criteria: 75% identity over 100 bp

```
 exon: homo (mus)
 1680  (1785) to   1808  (1913) =   129bp  exon
 1915  (2012) to   2239  (2336) =   325bp  exon
 2367  (2477) to   2528  (2638) =   162bp  exon
 2608  (2732) to   2799  (2923) =   192bp  exon
 2886  (3056) to   3067  (3237) =   182bp  exon
 3146  (3312) to   3289  (3455) =   144bp   exon

*************** Conserved Regions - homo (mus) ***************

  473  (518)  to    692  (738)  =   227bp at 77.1% noncoding
 1680  (1785) to   1808  (1913) =   129bp at 89.9% exon
 1915  (2012) to   2239  (2336) =   325bp at 89.2% exon
 2367  (2477) to   2528  (2638) =   162bp at 90.1% exon
 2608  (2732) to   2799  (2923) =   192bp at 90.6% exon
 2886  (3056) to   3067  (3237) =   182bp at 92.9% exon
 3146  (3312) to   3289  (3455) =   144bp at 93.8% exon
 3423  (3575) to   3549  (3704) =   134bp at 76.9% noncoding

Total 1495bp at 87.5%

*************** Conserved Regions - homo (gallus) ***************

 1680   (841) to   1808   (969) =  129bp at 84.5% exon
 1915  (1802) to   2237  (2124) =   323bp at 83.6% exon
 2369  (2754) to   2528  (2913) =   160bp at 84.4% exon
 2608  (3026) to   2797  (3215) =   190bp at 77.9% exon
 2886  (4216) to   3067  (4397) =   182bp at 82.4% exon
 3146  (4757) to   3288  (4899) =   143bp at 82.5% exon

Total 1127bp at 82.5%

*************** Conserved Regions - homo (xenopus) ***************

 1680  (2462) to   1808  (2590) =   129bp at 77.5% exon
 1915  (3543) to   2237  (3865) =   323bp at 84.2% exon
 2369  (3978) to   2528  (4137) =   160bp at 80.0% exon
 2608  (4653) to   2797  (4842) =   190bp at 81.1% exon
 2886  (4946) to   3067  (5127) =   182bp at 82.4% exon
 3146  (5531) to   3288  (5673) =   143bp at 79.7% exon

Total 1127bp at 81.5%

*************** Conserved Regions - homo (drosophila) ***************

 1915  (2879) to   2228  (3192) =   314bp at 88.2% exon
 2369  (3206) to   2526  (3363) =   158bp at 84.2% exon
 2610  (3368) to   2797  (3555) =   188bp at 86.7% exon
 2888  (3560) to   3001  (3673) =   114bp at 79.8% exon
 3146  (3801) to   3288  (3943) =   143bp at 86.7% exon

Total 917bp at 85.9%
```

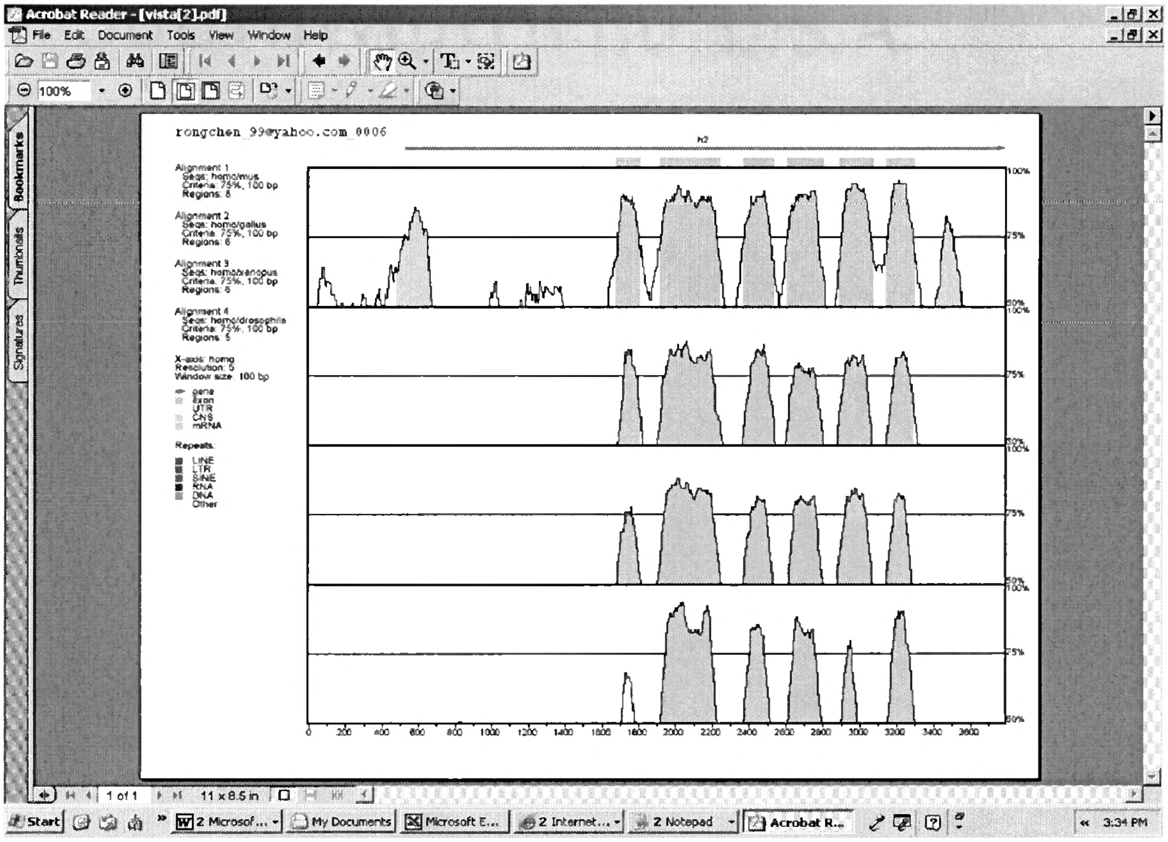


Figure 7. VISTA plot for alignment of homologous gene (gene#2) in five species

the number, length and similarities of exons become less similar. Most (93%) of the intron conservations happened between human and mouse, the conservations are not limited to UTRs, and about 30% of intron conservative regions are in internal intron areas, 20% are in intergenic regions.

Table 3 and Figure 7 illustrate the conserved regions of genes homologous to the human cardiac muscle a-actin gene (gene #2). There are 6 CDs in human gene, all of them are well conserved in mouse, chicken and frog. The initial CDs is missed in fruit fly. In mouse, there are two more conserved regions other than CDs. One is in 3'UTR, the other is upstream to the initial exon (there is no evidence it is in 5'UTR).

4.4 Sequence similarity vs. sequence distance

Furthermore, a study on the phylogeny is done on the organisms data on hand. Phylogeny is the study of the evolution of life forms. Distances between species are essential concepts in phylogeny. If two species have a small distance between them (as measured by the number of differences in their character sequences), then they have a recent common ancestor; but if they are far apart, then their common ancestor is in the remote past. The distance between the species can be used as a measure of the distance in time since the species diverged. These two distances, the number of character differences and the time since divergence, will be approximately proportional when they're relatively small.

A phylogenetic tree, also called a cladogram or a dendrogram, is a tree of several life forms and their relations. When two lines converge to a point, that should be interpreted as the point when the two species diverged from a common ancestral species, the point being the common ancestral species.

Clustal W [18], a multiple alignment algorithm based on hierarchical clustering, is used to get an alignment of each group of homologous sequences from five species, and nearest-neighbor joining algorithm is used to draw the dendrogram [28]. This is a form of cluster analysis and the end result produces something that looks like a tree. It represents the similarity of the sequences as a hierarchy.

Figure 8 is the dendrogram obtained from the sequences of homologous muscle action genes. Eighty-five percent of gene data sets in this work have the similar dendrogram, with homo-mus having closest relation, drosophila most distant with homo. This study verifies that the similarities between species in the global level decreases with the increase of the sequence distances.

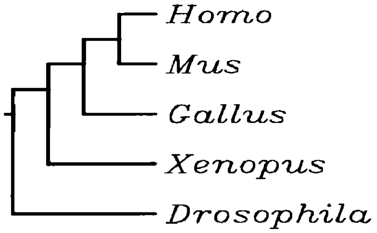


Figure 8. The dendrogram based on the multiple alignment of the homologous muscle a-actin genes of five species

Chapter 5 A Gene Recognition Approach Based on Cross-Species Sequence Comparison and the Impact of Evolutionary Distance on the Performance

5.1 Methods

5.1.1 Overall Approach

A gene-finding model is proposed by which the protein-coding regions can be identified by comparing genomic sequences of two species.

Firstly, given the sequences containing orthologous genes from two related organisms (obtained from TBLASTX search against GenBank), a new method for measuring the similarities among sequences is employed to get the most conserved fragments of two species, termed high-scoring fragments (*HSFs*).

HSFs are further processed to get the *candidate exons*. The conserved splice junctions and start/stop codons are identified using Salzberg's detecting procedures [20]. A candidate exon starts with a start codon or an acceptor signal and ends with a donor signal or a stop codon.

Finally, dynamic programming is implemented to assemble an optimal chain of candidate genes with consistent feature, that is, a single gene (or each of the multiple genes) beginning with a start codon, and ending with a stop codon, each exon ending with a donor signal is followed by an exon starting with an acceptor signal.

The model can be presented in Figure 9.

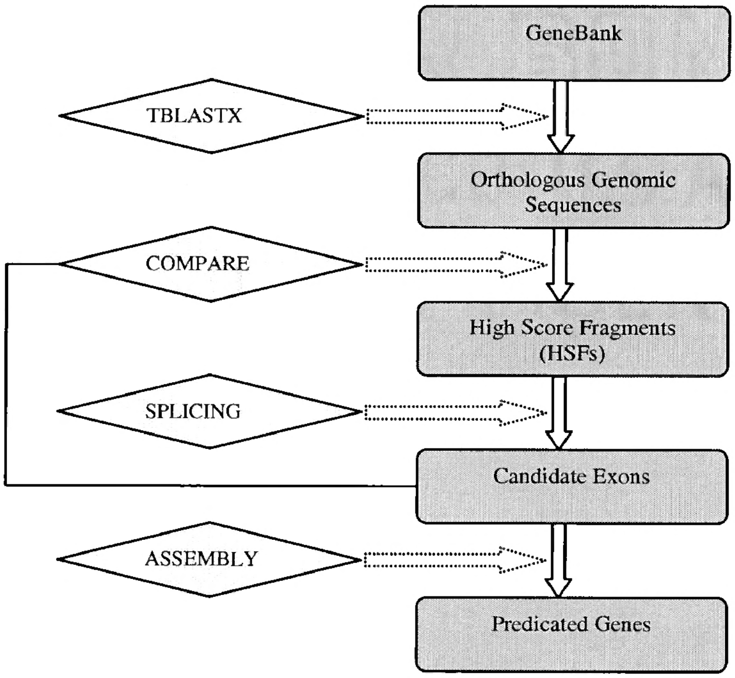


Figure 9. The flow-chart of the gene-finding method by cross-species comparison.

5.1.2 Sequence Comparison

The most important step in comparative genomics is sequence comparison. Alignment is the most frequently used method in comparative analysis. Most standard alignment methods are either global methods that try to align scquences over their entire length or local methods that return only the most highly conserved region of local similarity. These methods are not appropriate for alignment of large genomic sequences where local homologies may be separated by large stretches of unrelated 'junk DNA' [11]. For example, standard dynamic programming methods (i.e. Needleman-Wunsch (1970) or Smith-Waterman (1981) algorithms) are not sensitive to finding short regions of good alignment (such as a 50-base exon) flanked by much longer regions of poor alignment (such as long introns) [10]. Global alignment methods would try to align even completely unrelated parts of the sequences. Local methods, especially the faster heuristic local alignment methods (such as BLAST) are better suited, but still insufficient. They would identify high-scoring local similarities, but would not give an overall picture of the homologies among the input sequences [11]. Moreover, BLAST detects alignments by looking for perfect matches of a predetermined length (e.g., 11 bases) and thereby may miss important conserved regions.

Thus, in this work, both BLAST engine and Limpel-Ziv (LZ) complexity [21, 22, 24], an original distance-measurement method are utilized to define sequence similarity. In the following section, some basic concepts of LZ complexity will be introduced.

5.1.2.1 LZ complexity [19]

LZ complexity of a finite sequence S is related to the number of steps required by a production process that builds S.

Reproduction and Production

Let S, Q and R be sequences defined over an alphabet A, $l(S)$ be the length of S, $S(i)$ denote the i^{th} element of S and $S(i,j)$ define the substring of S composed of the elements of S between positions i and j (inclusive). An extension $R = SQ$ of S is *reproducible* from S (denoted $S \rightarrow R$) if there exists an integer $p \leq l(S)$ such that $Q(k) = R(p+k-1)$ for $k=1, \ldots, l(Q)$. For example $AACGT \rightarrow AACGTCGTCG$ with $p = 3$ and $AACGT \rightarrow AACGTAC$ with $p = 2$.

Another way of looking at this is to say that R can be obtained from S by copying elements from the p^{th} location on in S to the end of S. As each copy extends the length of the new sequence beyond $l(S)$, the number of elements copied can be greater than $l(S) - p$. Thus, this is a simple copying procedure of S starting from position p, which can carry over the added part, Q.

A sequence S is *producible* from its prefix $S\ (1,j)$ (denoted $S(1,j) \Rightarrow S$), if $S(1,j) \rightarrow S(1,\ l(S\ )\text{-}\ 1)$. For example $AACGT \Rightarrow AACGTAC$ and $AACGT \Rightarrow AACGTACC$ both with pointers $p = 2$. Note that production allows for an extra "different" symbol at the end of the copying process which is not permitted in reproduction. Therefore an extension which is reproducible is always producible but the reverse may not always be true.

Exhaustive History

Any sequence S can be built using a *production* process where at its i^{th} step $S(1,\ h_{i-1}) \Rightarrow S(1,h_i)$ (note that $\varepsilon = S(1,0) \Rightarrow S(1,1)$). An m-step production process of S results in a parsing of S in which $H(S) = S(1,h_1)/S(h_1+1,h_2),\ \ldots,\ S(h_{m-1}+1,\ h_m)$ is called the *history* of S and $H_i(S)=S(h_{i-1}+1,\ h_i)$ is called the i^{th} component of $H(S)$. For example for $S = AACGTACC$, $A/A/C/G/T/A/C/C$, $A/AC/G/T/A/C/C$, $A/AC/G/T/ACC$ are three different (production) histories of S.

If we *cannot* get $S(1,\ h_{i-1}) \rightarrow S(1,h_i)$, then $H_i(S)$ is called *exhaustive*. In other words, for $H_i(S)$ to be exhaustive the i^{th} step in the production process must be a production only meaning that the copying process cannot be continued and the component should be halted with a single letter innovation. A history is called exhaustive if each of its components (except maybe the last one) is exhaustive. For example, the third history given in the preceding paragraph is an exhaustive history of $S = AACGTACC$. Moreover, every sequence S has a unique exhaustive history.

Let $c_H(S)$ be the number of components in a history of S. Then the LZ complexity of S is $c(S) = min\{c_H(S)\}$ over all histories of S. It can be shown that $c(S) = c_E(S)$ where $c_E(S)$ is the number of components in the exhaustive history of S. This is quite intuitive as an exhaustive component is the longest possible at a given step of a production process.

Distance measurement by LZ complexity

Given two sequences Q and S, consider the sequence SQ, and its exhaustive history. By definition, the number of components needed to build Q when appended to S is $c(SQ) - c(S)$. This number will be less than or equal to $c(Q)$ because at any given step of the production process of Q (in building the sequence SQ) we will be using a larger search space due to the existence of S. Therefore the copying process can only be longer which in turn would reduce the number of exhaustive components. This can also be seen from the subadditivity of LZ complexity [100]: $c(SQ) \leq c(S) + c(Q)$. How much $c(SQ) - c(S)$ is less than $c(Q)$ will depend on the degree of similarity between S and Q.

For example, let $S = AACGTACCATTG$, $R = CTAGGGACTTAT$ and $Q = ACGGTCACCAA$. The exhaustive histories of these sequences would be:

$$H_E(S) = A/AC/G/T/ACC/AT/TG$$

$$H_E(R) = C/T/A/G/GGA/CTT/AT$$

$$H_E(Q) = A/C/G/GT/CA/CC/AA$$

Yielding $c(S) = c(R) = c(Q) = 7$. The exhaustive histories of the sequences *SQ*, and *RQ* would be:

$$H_E(SQ) = A/AC/G/T/ACC/AT/TG/ACGG/TC/ACCAA$$

$$H_E(RQ) = C/T/A/G/GGA/CTT/AT/ACG/GT/CA/CC/AA$$

Note that it took 3 steps to build *Q* in the production process of *SQ*. On the other hand, we used 5 steps to generate *Q* in the production process of *RQ*. The reason it took more steps in the second case is because *Q* is "closer" to *S* than *R*. In this example we can observe this by looking at the patterns *ACG* and *ACC* which *Q* and *S* share. We can formulate the number of steps it takes to generate a sequence *Q* from a sequence *S* by $c(SQ) - c(S)$. Thus, if *S* is closer to *Q* than *R* then we would expect $c(SQ)-c(S)$ to be smaller than $c(RQ) - c(R)$ as is the case in the above example.

Given two sequences *S* and *Q*, the distance function *distance (S, Q)* is defined as the following formulas:

$$d(S,Q) = max\{ c(SQ) - c(S), c(QS) - c(Q) \} \quad (1)$$

$$d^*(S,Q) = max\{ c(SQ)-c(S), c(QS)-c(Q) \} / max\{ c(S), c(Q) \} \quad (2)$$

$$d_1(S,Q) = c(SQ) - c(S) + c(QS) - c(Q) \quad (3)$$

$$d_1^*(S,Q) = [\, c(SQ)-c(S) + c(QS)-c(Q) \,] / c(SQ) \quad (4)$$

$$d_1^{**}(S,Q) = [\, c(SQ)-c(S) + c(QS)-c(Q)] / ½[\, c(SQ) + c(QS)] \quad (5)$$

All of the above functions have been verified as distance metric. That is, a distance metric, $D(S, Q)$ should satisfy the following conditions:

1. $D(S, Q) \geq 0$ where the equality is satisfied iff $S = Q$ (identity).
2. $D(S, Q) = D(Q, S)$ (symmetry).
3. $D(S, Q) \leq D(S, R) + D(R, Q)$ (triangle inequality).

In this work, function (3) is used to determine the distance between two sequences and the value of one parameter in the scoring function in the model (section 5.1.4).

5.1.2.2 Sequence comparison by BLAST and LZ complexity

The algorithm of LZ complexity [19] is implemented to get the exhaustive history. Although LZ complexity is an efficient approach to determine the distance between long genomic sequences, it is not enough to get a view of conservation feature of gene structure. One reason is that there exist many short components that come from "randomly repeated" and no gap extension; the other reason is that no corresponding similarity structure is established as in alignment. However, these feature could just be complementary with alignment, i.e. they could find some short regions that BLAST misses due to the fixed word size; Moreover, the LZ complexity gives the overall features of similarities, while the local alignment would not give an overall picture of the homologies among the input sequences.

Thus, in this model, BLAST is used to get local high-score alignment regions, and LZ complexity is used to find regions with intense similarity above some threshold (the number of exhaustive history components smaller than 4 over a 50bp window). The results from two methods are united to get total HSFs. The method includes the following parts:

1) Given two sequences *Q* and *S*, build the exhaustive history of *QS* and *SQ*, represented as H_E *(QS)* and H_E *(SQ)*. Then, put all the "halting points" in H_E *(QS)* and only in S to H_E*List (S)*. H_E*List (Q)* is got in similar way.

2) Find regions in Q and S, with "more intensive" repeated sequences, in the term of number of exhaustive history components smaller than some threshold T over a specified length bp window (4 components over 50 bp in our program), update the H_E*List (S)*, H_E*List (Q)* to include only high intensity groups.

3) Align Q and S with BLAST tool (Blast 2 Sequences, reward for a match is 1, penalty for a mismatch, open gap and extension gap is 2, 5, 2, respectively, word size is 11). Put the boundary points of each high score pair to *BlastList* (include the information of both sequences)

4) Parse and combine *BlastList,* H_E*List (S)* and H_E*List (Q)*. If the region in H_E*List (S)* and H_E*List (Q)* is overlapping with BLAST results, extend the high score pairs of BLAST to include them. If the region in $H_E List(S)$ and $H_E List(Q)$ is outside BLAST results, treat it as an additional HSF.

The implementation of the proposed method demonstrates that most high-intensity regions by LZ complexity overlapped with BLAST alignment regions. But the calculation of exhaustive history provides another way to view the sequence similarity in somewhat global aspect, and it is helpful to combine some short local alignments of BLAST together. There are three cases in the 117 human-mouse sequences in which short weakly conserved regions are missed by BLAST but detected by LZ complexity.

5.1.3 Splicing Signals

5.1.3.1 The Consensus Sequences around Splicing Sites and Identification

Many studies have attempted to characterize the sequences around the start, donor, and acceptor sites in eukaryotic organisms. The consensus sequences have been found as matrices containing the probabilities of the four bases in the positions immediately surrounding the sites [20].

Proper identification of sequence signals (both splice junctions and translation start sites) is a critical component in gene identification systems. Specific computational systems for identifying splice junctions have been developed by many previous researchers. The approaches include neural network for positional frequencies, dinuclotide frequencies, triplet counts, octamer frequencies, etc. [25]

To identify the consensus sequences that signal the start of translation and the boundaries between exons and introns (donor and acceptor sites), the conditional probability (CP) matrices proposed by Salzberg [20] are used. This method takes into account the dependencies between adjacent bases, in contrast to the usual technique of considering each position independently. The consensus sequence information is summarized in conditional probability matrices which, in Salzberg's study, when used to locate signals in uncharacterized genomic DNA, have greater sensitivity and specificity than conventional matrices.

5.1.3.2 Conditional Probability (CP) Matrices for Splice Sites and Translational Start Sites by Salzberg [20]

The basis of the method is the computation of conditional probabilities for each of the four bases that comprise DNA in a fixed set of positions around each site. The standard method, by contrast, computes the probabilities of the bases in each position as if they were independent of adjacent bases. Instead, the new method by Salzberg is to compute, for each position, the probability of each base given the base in the previous position, where "previous" is defined as the adjacent base in the 5' direction. In the consensus pattern that emerges, the identity of each base is dependent on its neighbors.

The resulting conditional probability (CP) matrices indicate that for several positions in all three types of sites (start of translation, donor, and acceptor sites) the probability of a base occurring in a given position is sometimes strongly dependent on the previous base. This has a natural biological explanation, in that the mechanisms responsible for translation and splicing involve molecules that recognize and bind to sets of adjacent bases in the mRNA [20].

5.1.3.3 Detecting Signals with CP Matrices

Consensus matrices can be used for signal detection in the following manner. For any pattern of anonymous DNA, one must compute a score based on its probability of being a true instance of a start, donor, or acceptor site. This score can be compared to the scores of known true sites to determine if the anonymous pattern is also a true site.

Salzberg proposed three conditional probability matrix for start sites, donor sites and acceptor sites, respectively (Table 4, 5, and 6).

For conditional probability matrices, the score in the CP matrix is really a 1-state Markov chain model, computed by multiplying the conditional probabilities of each successive base, given the previous base in the sequence. Thus the CP matrix takes into account the dependencies between adjacent bases in the sequence.

These CP matrices are based to compute *P (S | T)*; i.e., the probability of a sequence *S* = *(s₁ , s₂ , ..., sₙ)* given that it is a true site. Then compute *P (S | T)* as

$$\boldsymbol{P}\,(s_1)\prod \boldsymbol{P}\,(s_i\,|\,s_{i\text{-}1}) \qquad (i=2,3,\ldots,n) \tag{6}$$

The probability can be obtained from the CP matrix. This score (called splicing weight score after modification; see section 5.1.4) is used as one parameter in the scoring function for identifying candidate exons.

Table 4. Conditional probability matrix for vertebrate start sites. Each column after the first contains the probability of a base in that position given the base in the previous position, as indicated at the end of each row.

-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	+1	+2	+3	+4	+5	+6	
.23	.24	.42	.27	.16	.30	.16	.20	.16	.44	.28	.29	1.0	0.0	0.0	0.0	.38	.11	.37	$P\,(a_i \mid a_{i\text{-}1})$
.23	.27	.24	.32	.57	.29	.08	.22	.67	.06	.45	.17	0.0	0.0	0.0	0.0	.14	.19	.27	$P\,(c_i \mid a_{i\text{-}1})$
.23	.45	.24	.28	.23	.30	.68	.45	.14	.47	.15	.50	0.0	0.0	0.0	0.0	.40	.59	.27	$P\,(g_i \mid a_{i\text{-}1})$
.23	.05	.10	.14	.04	.11	.08	.13	.03	.03	.13	.05	0.0	1.0	0.0	0.0	.08	.11	.08	$P\,(t_i \mid a_{i\text{-}1})$
.40	.35	.30	.25	.15	.33	.29	.08	.32	.78	.48	.08	1.0	0.0	0.0	0.0	.32	.18	.38	$P\,(a_i \mid c_{i\text{-}1})$
.40	.26	.33	.26	.47	.29	.28	.47	.46	.04	.41	.80	0.0	0.0	0.0	0.0	.29	.29	.28	$P\,(c_i \mid c_{i\text{-}1})$
.40	.09	.11	.20	.10	.07	.21	.05	.13	.17	.10	.05	0.0	0.0	0.0	0.0	.01	.17	.13	$P\,(g_i \mid c_{i\text{-}1})$
.40	.30	.26	.29	.28	.31	.21	.40	.10	.01	0.0	.07	0.0	0.0	0.0	0.0	.38	.37	.22	$P\,(t_i \mid c_{i\text{-}1})$
.17	.17	.45	.22	.24	.29	.29	.41	.21	.59	.19	.19	1.0	0.0	0.0	.28	.17	.09	.22	$P\,(a_i \mid g_{i\text{-}1})$
.17	.35	.19	.37	.40	.36	.33	.30	.55	.03	.67	.35	0.0	0.0	0.0	.15	.35	.28	.47	$P\,(c_i \mid g_{i\text{-}1})$
.17	.33	.15	.30	.21	.17	.29	.16	.21	.34	.06	.44	0.0	0.0	0.0	.48	.14	.39	.23	$P\,(g_i \mid g_{i\text{-}1})$
.17	.15	.21	.11	.16	.17	.09	.14	.03	.03	.07	.01	0.0	0.0	0.0	.09	.34	.21	.07	$P\,(t_i \mid g_{i\text{-}1})$
.19	.10	.11	.11	.07	.20	.05	.06	.14	.47	.30	.11	1.0	0.0	0.0	0.0	.04	.03	.13	$P\,(a_i \mid t_{i\text{-}1})$
.19	.47	.37	.51	.48	.32	.20	.40	.59	.12	.20	.82	0.0	0.0	0.0	0.0	.44	.17	.46	$P\,(c_i \mid t_{i\text{-}1})$
.19	.26	.24	.22	.23	.24	.60	.27	.20	.38	.10	.03	0.0	0.0	1.0	0.0	.30	.69	.25	$P\,(g_i \mid ta_{i\text{-}1})$
.19	.17	.28	.16	.23	.25	.15	.27	.06	.03	.40	.03	0.0	0.0	0.0	0.0	.22	.12	.16	$P\,(t_i \mid t_{i\text{-}1})$

Table 5. Conditional probability matrix for vertebrate donor sites.

-3	-2	-1	0	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	+16	+17	
.35	.60	.07	0.0	0.0	0.0	.64	.06	.20	.24	.19	.26	.16	.29	.23	.25	.28	.24	.26	.25	.24	$P(a_i \mid a_{i-1})$
.35	.09	.02	0.0	0.0	0.0	.10	.03	.11	.22	.28	.24	.19	.18	.20	.20	.25	.24	.21	.24	.18	$P(c_i \mid a_{i-1})$
.35	.18	.86	1.0	0.0	0.0	.13	.89	.39	.37	.23	.30	.38	.33	.37	.30	.31	.31	.30	.27	.34	$P(g_i \mid a_{i-1})$
.35	.14	.06	0.0	0.0	0.0	.13	.03	.30	.17	.29	.20	.27	.19	.20	.25	.16	.21	.23	.24	.23	$P(t_i \mid a_{i-1})$
.35	.69	.17	0.0	0.0	0.0	.70	.19	.25	.35	.20	.27	.27	.30	.22	.26	.24	.22	.21	.29	.27	$P(a_i \mid c_{i-1})$
.35	.11	.06	0.0	0.0	0.0	.05	.21	.27	.26	.38	.31	.33	.33	.34	.35	.33	.36	.37	.30	.31	$P(c_i \mid c_{i-1})$
.35	.07	.61	1.0	0.0	0.0	.07	.41	.09	.13	.06	.11	.11	.11	.10	.11	.10	.09	.10	.11	.09	$P(g_i \mid c_{i-1})$
.35	.13	.16	0.0	0.0	0.0	.18	.20	.39	.26	.37	.31	.29	.27	.33	.28	.33	.33	.32	.31	.34	$P(t_i \mid c_{i-1})$
.19	.65	.11	0.0	0.0	0.0	.83	.05	.15	.28	.21	.18	.21	.20	.24	.19	.24	.17	.20	.19	.20	$P(a_i \mid g_{i-1})$
.19	.15	.01	0.0	0.0	0.0	.06	.05	.15	.29	.26	.30	.24	.23	.26	.25	.21	.30	.25	.22	.20	$P(c_i \mid g_{i-1})$
.19	.11	.80	1.0	0.0	0.0	.09	.87	.15	.28	.34	.37	.39	.43	.32	.42	.36	.35	.39	.43	.39	$P(g_i \mid g_{i-1})$
.19	.09	.08	0.0	1.0	0.0	.03	.03	.55	.15	.20	.14	.15	.14	.18	.15	.19	.17	.15	.16	.21	$P(t_i \mid g_{i-1})$
.11	.16	.02	0.0	0.0	.51	.19	.05	.11	.24	.15	.15	.15	.18	.16	.10	.13	.15	.16	.15	.15	$P(a_i \mid t_{i-1})$
.11	.24	.03	0.0	0.0	.03	.08	.11	.12	.19	.30	.28	.21	.18	.25	.24	.25	.22	.26	.21	.23	$P(c_i \mid t_{i-1})$
.11	.31	.86	1.0	0.0	.43	.63	.77	.43	.36	.28	.31	.37	.40	.25	.32	.27	.30	.31	.32	.34	$P(g_i \mid ta_{i-1})$
.11	.29	.08	0.0	0.0	.03	.10	.06	.33	.20	.27	.25	.26	.24	.34	.35	.35	.33	.26	.32	.29	$P(t_i \mid t_{i-1})$

Table 6. Conditional probability matrix for vertebrate acceptor sites.

-14	-13	-12	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1	0	
.09	.18	.22	.10	.13	.14	.14	.11	.09	.18	.58	.06	1.0	0.0	0.0	$P(a_i \mid a_{i-1})$
.09	.27	.33	.29	.36	.40	.35	.41	.40	.34	.28	.76	0.0	0.0	0.0	$P(c_i \mid a_{i-1})$
.09	.03	.02	.04	.02	.02	.01	.02	0.0	0.0	.04	.01	0.0	1.0	0.0	$P(g_i \mid a_{i-1})$
.09	.52	.43	.56	.50	.44	.50	.47	.51	.48	.09	.18	0.0	0.0	0.0	$P(t_i \mid a_{i-1})$
.34	.09	.09	.09	.10	.12	.13	.09	.08	.10	.36	.04	1.0	0.0	0.0	$P(a_i \mid c_{i-1})$
.34	.32	.32	.32	.37	.41	.38	.42	.51	.48	.31	.68	0.0	0.0	0.0	$P(c_i \mid c_{i-1})$
.34	.06	.03	.03	.04	.02	.07	.03	.02	.02	.09	0.0	0.0	0.0	0.0	$P(g_i \mid c_{i-1})$
.34	.52	.57	.56	.50	.45	.42	.47	.40	.41	.23	.27	0.0	0.0	0.0	$P(t_i \mid c_{i-1})$
.13	.08	.09	.03	.10	.06	.06	.04	.06	.07	.24	.02	1.0	0.0	.25	$P(a_i \mid g_{i-1})$
.13	.27	.33	.24	.32	.33	.38	.42	.30	.27	.21	.85	0.0	0.0	.16	$P(c_i \mid g_{i-1})$
.13	.11	.12	.15	.16	.10	.12	.14	.08	.19	.45	0.0	0.0	0.0	.50	$P(g_i \mid g_{i-1})$
.13	.54	.46	.58	.42	.50	.43	.40	.55	.47	.10	.13	0.0	0.0	.09	$P(t_i \mid g_{i-1})$
.44	.06	.06	.03	.05	.07	.06	.06	.05	.06	.16	.04	1.0	0.0	0.0	$P(a_i \mid t_{i-1})$
.44	.32	.30	.32	.40	.32	.41	.40	.41	.28	.26	.68	0.0	0.0	0.0	$P(c_i \mid t_{i-1})$
.44	.18	.18	.15	.14	.20	.17	.12	.09	.08	.31	0.0	0.0	0.0	0.0	$P(g_i \mid t_{i-1})$
.44	.44	.46	.50	.42	.41	.36	.42	.45	.58	.26	.28	0.0	0.0	0.0	$P(t_i \mid t_{i-1})$

To reduce the noise generated by false positive splice signals, only those splice signals and start/stop codons that occur in both respective segments at the same relative position are accepted.

Candidate exons (CEs) are obtained by elongating or shortening HSFs such that they start with a conserved start codon or acceptor site and end with a conserved stop codon or donor site.

5.1.4 A dynamic-programming procedure for finding optimal chains of candidate genes

5.1.4.1 Necessary definitions, functions and conditions

A dynamic programming procedure is implemented to find a chain of candidate genes that is most likely to correspond to the real genes in the input sequences. Some definitions:

1) *Scoring Function (sc)*:

Given a set of candidate exons, for every possible chain of candidate genes, a quality score is calculated by the *scoring function.* The scoring function is defined based on:

i) the degree of similarity among the respective segments of the candidate exons sequences, as measured by the alignment weight scores, reward for a match is 1, penalty for a mismatch, open gap and extension gap is 2, 5, 2, respectively, and the LZ distance of two segments calculated by function (2) in section 5.1.2;

ii) the quality of the splice signals, obtained based on Salzberg's consensus matrices, the scores are justified by multiplication of 10^9 , 10^{10} , 10^7 for start, donor and acceptor sites.

Consider a *candidate exon E*, *w*(*E*) is defined as the *weight score* from alignment, *lz*(*E*) as the LZ distance between two segments, *sc(splice)* as the score of the splice signals by which *E* is bounded. *k* is constant. The score *sc(E)* of *E* is then defined as

$$sc(E) = w(E) + k / lz(E) + sc(splice) \qquad (7)$$

2) *open reading frame index (orfi):* it is requested that all candidate exons are open reading frames (ORFs), i.e. they are not allowed to contain internal stop codons. Each candidate exon CE is associated with exactly one reading frame *orfi (E)* $\in$ $\{0,1,2\}$, thus, a region of local sequence similarity flanked by conserved splice signals or start/stop codons can respond to up to three distinct CEs, depending on how many ORFs it comprises.

3) *remainder* of a candidate exon *CE* (*re(E)*): the length of the last (possibly truncated) codon of *E*, so we have $re(E) \in \{1, 2, 3\}$ for all candidate exons *E*.

With these definitions, the requirement that a chain $C = (E_1, \ldots , E_k)$ of candidate exons represents a biologically consistent chain of candidate genes is equivalent to the following conditions

```
1)    if        ($E_1$ is on the plus strand)
      then      $E_1$ starts with a start codon;
      else if   ($E_1$ is on the mius strand)
      then      $E_1$ starts with a stop codon;
2)    if        ($E_k$ is on the plus strand)
      then      $E_k$ starts with a start codon;
      else if   ($E_k$ is on the mius strand)
      then      $E_k$ starts with a stop codon;
3)    if        (($Ei$ is on the plus strand) and
                ($Ei$ ends with a donor splice site))
      then      $E_{i+1}$ is on the plus strand;
                $E_{i+1}$ starts with an acceptor splice site;
                $re(E_i) + orfi(E_{i+1}) = 3$;
      else if   (($Ei$ is on the minus strand) and
                ($Ei$ ends with an acceptor splice site))
      then      $E_{i+1}$ is on the minus strand;
                $E_{i+1}$ starts with a donor splice site;
                $re(E_i) + orfi(E_{i+1}) = 3$;
      else if   (($Ei$ is on the plus strand) and
                ($Ei$ ends with a stop codon))
                        OR
```

((*Ei* is on the minus strand) and
(*Ei* ends with an start codon))
then E_{i+1} is on the plus strand;
E_{i+1} starts with a start codon;
OR
E_{i+1} is on the minus strand;
E_{i+1} starts with a stop codon;

Given a set of candidate exons $EX= \{E_1, \dots , E_N\}$, and a scoring function *sc* (as function (7)) assigning a score *sc*(*E*) to every $E \in EX$, a standard one-dimensional interval chaining dynamic algorithm with modification [11, 23] is implemented. The algorithm returns a chain of candidate exons ($\tilde{E}_1, \dots ,\tilde{E}_k$), $\tilde{E}_i \in EX$, satisfying conditions described above with maximal total score.

5.1.4.2 Pseudo code

Input data are a set $EX = \{E_1, \dots , E_N\}$ and a score *sc* (E_j) for each $E_j \in EX$. *BList* is a list containing the starting and end points of all $E_j \in EX$. After sorting the list *BList* the algorithm calculates for each $E_j \in EX$ the total score $SC(E_j\ )$ of an optimal chain *ending* in E_j together with the *predecessor* $PR(E_j\ )$ of *E* in this chain. max_i^+ is the maximal total score so far of a chain ending with a CE E_j on the plus strand with a remainder $re(E_j\ ) = i$, pr_i^- is the last CE E_j in this chain; max_i^- and $pr_i^-$are defined accordingly. *max** is the maximal total score so far of an optimal chain ending with a start or stop codon, *pr** is the last CE in this chain.

```
Step 1: Initialization:
    Sort the list BList of end points of candidate exons in EX
    $max^- = 0$; $pr^- =$ NIL
    for i = 1 to 3 do
        $max_i^+ = -\infty$;
        $max_i^- = -\infty$;
        $pr_i^+ =$ NIL;
        $pr_i^- =$ NIL;
    for i=1 to N do
        $SC(E) = -\infty$

Step 2: Recursion:
    for i = 1 to 2 x N (number of candidate exons) do
        if      (BList[i] is the left end point of some $Ej \in$ EX )
        then
                if      ((Ej is on the plus strand) and
                         (Ej starts with an acceptor site))
                then    $SC(Ej) \leftarrow sc(Ej) + max_{3-orfi(Ej)}^{+}$;
                        $PR(Ej) \leftarrow pr_{3-orfi(Ej)}^{+}$;
                if      ((Ej is on the minus strand) and
                        (Ej starts with a donor site))
                then    $SC(Ej) \leftarrow sc(Ej) + max_{3-orfi(Ej)}^{-}$;
                        $PR(Ej) \leftarrow pr_{3-orfi(Ej)}^{-}$;
                if      (Ej starts with a start or stop codon)
                then    $SC(Ej) \leftarrow sc(Ej) + max^-$;
                        $PR(Ej) \leftarrow pr^-$;
        if      (BList [i] is the right end point of some $Ej \in EX$ )
        then
                if      ((Ej is on the plus strand) and
            ($Ej$ starts with a donor site))
    then
            if      $(SC(Ej) > max_{re(Ej)}^{+})$
            then    $max_{re(Ej)}^{+} \leftarrow SC(Ej)$;
                    $pr_{re(Ej)}^{+} \leftarrow Ej$;
    if      (($Ej$ is on the minus strand) and
            ($Ej$ starts with an acceptor site))
    then
            if      $(SC(Ej) > max_{re(Ej)}^{-})$
            then    $max_{re(Ej)}^{-} \leftarrow SC(Ej)$;
                    $pr_{re(Ej)}^{-} \leftarrow Ej$;
    if      ($Ej$ ends with a start or stop codon)
    then
            if      $(SC(Ej) > max^{-})$
            then    $max^{-} \leftarrow SC(Ej)$;

                    $pr^{-} \leftarrow Ej$;
```

Step 3: Trace Back:

```
    $\tilde{E}_1 = pr^{-}$, i=1
    while $\tilde{E}_i \neq$ NIL do
        $\tilde{E}_{i+1} = PR(\tilde{E}_i)$;
        i = i+1;
```

5.2 Testing and Evaluation

The current program is tested with Batzoglous's 117 human and mouse DNA sequences; eight pairs of genes are selected as training set (gene 12, 23, 29, 45, 51, 72, 82 and 96), and the other 109 gene pairs are tested.

The accuracy of predictions is measured by comparing predicted and true exons (because the data set is well annotated according to Batzoglou's [10], the annotation can be regarded as the right answer). The set of correct and predicted exons can be classified into the following four types: Exons predicted correctly (*true positives (TP)),* exons not predicted (*Totally Missing Exons (ME)*), predicted exons with overlapping true exons (OV), and incorrectly predicted exons (*false positive(FP)*) [26]. The performance of prediction is evaluated at exon level; thus, an overlapping prediction is considered incorrect. The performance is evaluated in two measurements, *Sensitivity* and *Specificity.*

$$Sensitivity\ (Sn) = TP\,/\,(TP + OV + ME)$$

$$Specificity\ (Sp) = TP\,/\,(TP + FP + OV)$$

The 109 pairs of human and mouse genes contain a total of 924 exons. This program returns a prediction of 953 exons as a result, among which, 829 are true positives, 75 are overlapping with true exons, and 43 are false positive. 25 true exons are totally missing. The prediction results of each gene are listed in Table 9, 10 in Appendix. Figure 10 shows the fraction of TP, OV and FP in the predicted exons. OV and FP represents 13% of predicted exons. All of the FP exons and the parts of OV are obtained due to the conservation of non-coding regions. 78% of OV and 72% of FP are from internal intron area.

The sensitivity and specificity of this program are 90% and 87% respectively. There are two other studies which used the same data set to do gene-finding by sequence comparison. One is ROSETTA by Batzoglou *et al.* They obtained a sensitivity of 95% and a specificity of 97%; however, this result is on the nucleotide level, and no result at the exon level was reported. In addition, their program was specifically designed for human and mouse genes, and it involved many species-specific features such as codon usage, species-specific splicing sites detection, etc. Moreover, ROSETTA assumed the identical exon number and length in human and mouse genes. In contrast, the method proposed in this work excludes the assumption and relies little on statistical features from species-specific study, except a splicing-signal identification approach, which is applied to vertebrate and not very specific. Therefore, this method is much more applicable to other species with various evolutionary distances.

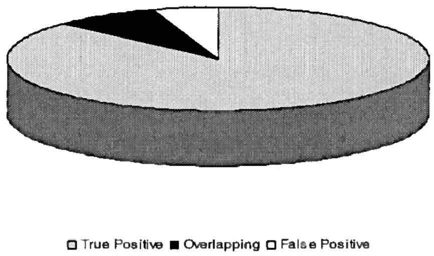


Figure 10. Fraction of the predicted exons by two-species method

AGenDA, the other method which used the same data, is quite similar to this method, except that they developed their own sequence alignment approach, which is a gap-free DIALIGN alignment. The results are compared, and better sensitivity and specificity are obtained by the method proposed in this work (Figure 11). This result demonstrates that the approach based on the sequence comparison by BLAST and LZ complexity succeeds in identification of coding regions, and would be useful for further study of more various organisms.

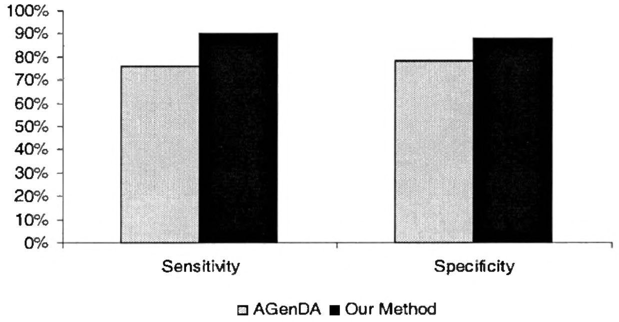


Figure 11. Comparison of results by two-species model and AGenDA

5.3 The Impact of Evolutionary Distance on the Performance of Gene-finding by pairwise comparison

This program is tested with pairs of sequences with different evolutionary distances, including *Homo-Mus, Homo-Gallus, Homo-Xenopus* and *Homo-Drosophila.* It showed that, in the model based on two-species comparison, human-mouse gene pairs are still the best for comparative study.

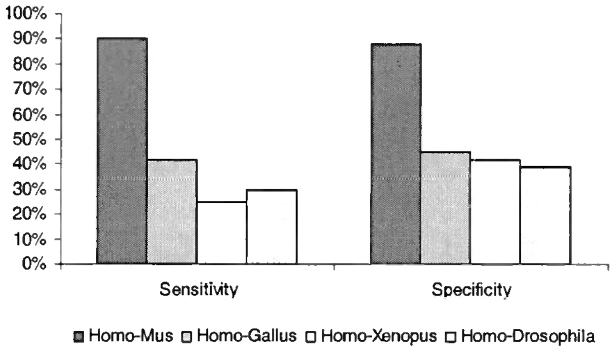


Figure 12. Results of prediction based on comparison with different evolutionary distances.

The evolutionary distance between sequences compared is a critical factor in the comparative gene-finding methods. Organisms whose sequence has not drifted sufficiently far from that of humans will not increase the signal-to-noise ratio sufficiently, while organisms that are too distant may make it difficult to recognize important signals [10]. Unfortunately, there is little study in this area. As described in Chapter 3, most current methods use human and mouse genes as optimal pairs for comparison. At evolutionary distances approximately 50 - 100 Myrs (human-mouse), the conservation extends not only to the exon but also to other functional regions maintaining gene expression, maintaining genome structure, etc [13, 26].

The decrease in sensitivity and specificity in this program with comparison of human and chicken, frog and fruit fly comparison is due to less conservation in the number and length of exon. Although coding regions reportedly are generally well conserved in species as far back as 450Myrs, and Novichkov [13] obtained the best result by comparison of human and fruit fly sequences, in this work, sequence analysis (Chapter 3) and performance study with the proposed program support that for pairwise comparison, human-mouse pairs are still the best for gene identification.

Chapter 6 Gene Recognition by Multiple Comparisons

The next question is: although comparison of human and mouse sequence leads to good sensitivity and specificity in coding-region identification, is it possible to get improved performance further by introducing one more species sequence? In multiple-species model, the homolog of the third sequence is introduced for comparison, to filter out the conserved non-coding regions. The homologous sequence in Gallus Gallus is obtained by TBLASTX. The scoring function (7) is modified to:

$$sc(E) = \sum [w_i(E) + k / lz_i(E)] + sc(splice) \tag{8}$$

$\sum w_i(E)$ means the sum of the weight scores of alignments of one candidate exon to other two species. If the alignment cannot be found in either one, the value is zero. $\sum k / lz_i(E)$ is similarly defined. The candidate exons with new *sc(E)* below some threshold T enter a filter system, which is essentially the dynamic programming procedure in Chapter 5 finding the optimal chain of candidate exons without or, with one or more low *sc(E)* candidate exons excluded.

While the current aim is to exclude the conserved non-coding regions, we are taking a risk as it seems possible that the less conserved exon region in human and chicken would be missed. The weight score of human/mouse and chicken is amplified by 3 to keep these regions. Furthermore, the program requires that the assembly of exons must keep the biological consistence, as the conditions described in section 5.1.4.1. In practice, the exclusion of a TP exon always leads to the change of reading frame and thus the consistence is broken. Thus, this method is feasible in improving the specificity while keeping the sensitivity.

The training set used in two-species model is also used in multiple comparison. The rest of genes which can find orthologous genes in chicken are tested. The results are listed in Table 11 and 12 in Appendix. To evaluate the accuracy of our multiple species model, the same set of data of human and mouse are also tested on two-species model.

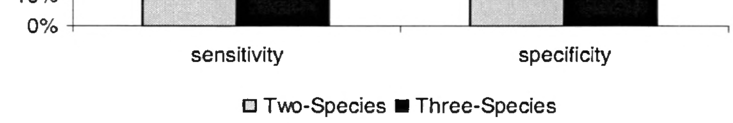


Figure 13. Comparison of results based on sequence comparison with two species and three species.

Figure 14 shows the comparison of results of two-species and three-species models, on predication of 246 exons of human and mouse. The three-species method increases the specificity from 85% to 95%. Interestingly, the sensitivity is improved very slightly.

Table 7. Comparison of predicted exons by two-species and three-species

	total # of predicated	true positive	overlapping	totally missing	false positive
Two-Species	256	218	24	6	12
Three-Species	231	219	6	20	6

Table 7 and Figure 14 list the difference in the results of two models. The overlapping and false positive exons were decreased by 75% and 50% respectively by the three-species method. Interestingly, the three-species method increases one true positive. This case is in gene 57, where, human found one homologous exon in chicken, while not in mouse.

The result obtained from human-mouse-chicken model shows greater specificity than from human-mouse. Thus, this method demonstrated improved accuracy in gene prediction by applying multiple species comparison of different distances, taking advantage both of the strong conservation of coding region between close species, and divergence of non-coding region between farther species.

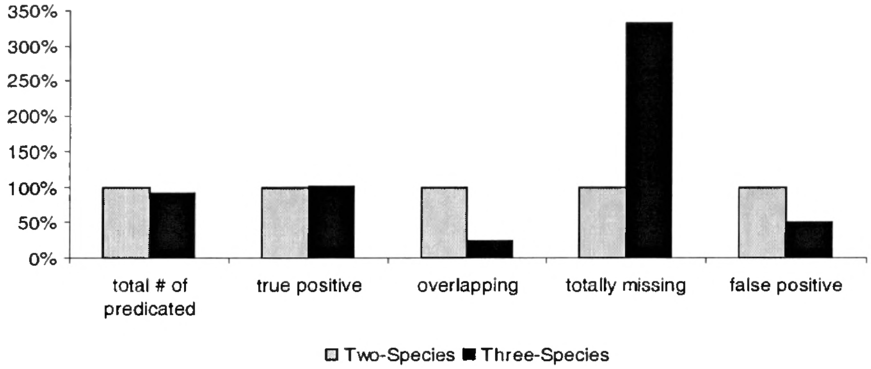


Figure 14. Comparison of exons by two-species method and three-species method

The results are compared with those by GenScan [8], the most successful software tool for gene prediction currently available [9]. This program produces better sensitivity and specificity than GenScan (Figure 15) on the same set of data.

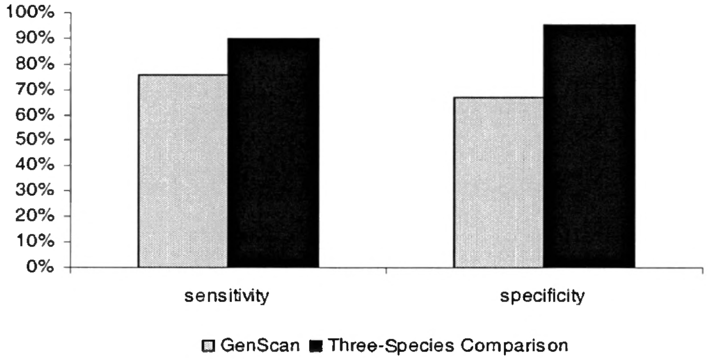


Figure 15. Comparison of results by GenScan and our three species comparison method

Figure 16 and 17 show the difference in the exons predicted by GenScan and three-species method. GenScan, which uses statistic method to get an optimal parse of genome sequence and distinguish coding-region from non-coding region, seems to have more overlapping and false positive predictions.

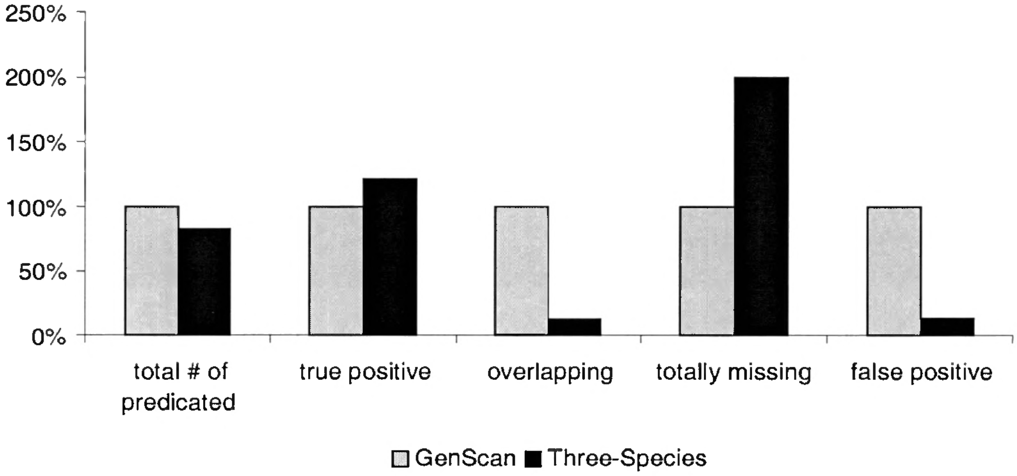


Figure 16. Comparison of exons predicted by GenScan and our three-species method

69% of the exons are correctly predicted by both methods, 21% are only detected by our method, 7% are detected by GenScan only. Since one method could identify exons that were missed by the other, in the real case, the methods from different approaches could complement each other.

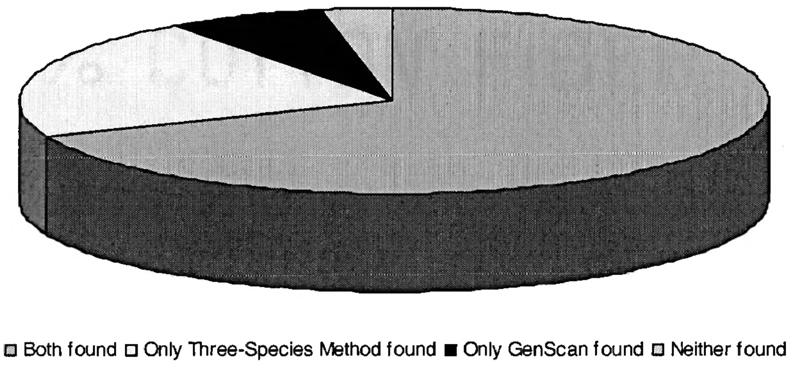


Figure 17. Percentage of exons correctly predicted by our three-species method and by GenScan.

Chapter 7 Summary and Conclusion

This work is the first one that considers comparisons of sequences from multiple organisms for identification of protein-coding regions. The sequence analysis shows that the conservation of gene structure decreases with the increase in the evolutionary distance. Conservation in non-coding regions is found in close species, i.e., in human-mouse gene pairs. In contrast, distant species have far fewer conservative features in non-coding regions.

A new sequence comparison method is developed, employing both local alignment and LZ complexity. Based on this method, together with a dynamic programming procedure, a program for identifying coding regions by pair-wise comparison is designed. Unlike the work of other authors, this program does not rely on the species-specific features, thus it is applicable in study with a wide range of species.

In the study of the impact of evolutionary distance on the performance of gene-finding by this approach, it is found that among the species in this work, the human-mouse is the best.

For the first time, a system for gene-finding with comparison of sequences of three species is proposed. The results demonstrate that introducing an appropriate distant species can lead to improved specificity while retaining the sensitivity of a system with two close species.

This method could be a valuable addition to current gene-prediction tools. Since it not species-specific, it could be applied to predict newly sequenced organisms which have no much known knowledge, provided the syntenic sequence group from organisms with appropriate evolutionary distances can be found. And this method will be more applicable with the increasing number of completed genome sequencing projects.

Appendix

Table 8. 117 pairs of human and mouse homologous gene.

Group #	Name of gene in human homolog	Homo Loci	Mus Loci
1	casein kinase II subunit beta	HSCKIIBE	MMGMCK2B
2	skeletal alpha-actin gene	HUMSAACT	MUSACASA
3	H4/e gene for H4 histone	HSH4EHIS	MMHIS412
4	ribosomal protein S24 (rps24) gene	HSU12202	MMMRPS24
5	histone H4 gene	HUMHIS4	MUSHIST4
6	histone H3 gene	HSHISH3	MMHIST31
7	hsc70 gene for 71 kd heat shock cognate protein	HSHSC70	MMU73744
8	POU-domain transcription factor	HUMNOCT	MUSPOUDOMB
9	slow twitch skeletal muscle/cardiac muscle troponin C	HUMTROC	MUSCTNC
10	int-1 mammary oncogene	HSINT1G	MUSINT1A
11	somatostatin receptor isoform 1 gene	HUMSRI1A	MUSSRI1A
12	M1 gene for muscarinic acetylcholine receptor	HSMIMAR	MUSACHRM1
13	fau 1 gene	HSFAU1	MUSFAUA
14	alpha-B-crystallin gene, 5' end	HUMCRYABA	MUSALPBCRY
15	ENO3 gene for muscle specific enolase	HSENO3	MMENO3G
16	21 kDa protein gene	HUMPPIB	MUSPPIA
17	voltage-gated potassium channel (HGK5) gene	HUMKCHN	MUSMK3A
18	proliferating cell nuclear antigen (PCNA) gene	HUMPCNA	MMPCNAG
19	CB1 cannabinoid receptor (CNR1) gene	HSU73304	MMU22948
20	Na,K-ATPase beta 2 subunit gene	AF007876	MMATPB2
21	neurotrophin-3 gene	HUMNT3A	MMNT3
22	gene for creatine kinase B	HSCKBG	MUSCRKNB
23	m4 muscarinic acetylcholine receptor gene	HUMACHRM4	MMM4ACHR
24	MHC class III HSP70-2 gene (HLA)	HUMMHHSP2	MUSHSP7A2
25	APX gene encoding APEX nuclease	HUMAPEXN	MUSAPEX
26	Human gadd45 gene	HUMGAD45A	MUSGAD45
27	MHC class III HSP70-HOM gene (HLA)	HUMMHHSPHO	MUSHSC70T
28	HOX 5.1 gene for HOX 5.1 protein	HSHOX51	MMU77364
29	histone H1 (H1F4) gene	HUMHISAC	MUSH1EH2B
30	spermidine synthase gene	HUMSPERSYN	MMSPERSYN
31	gene for histone H1(0)	HSHIS10G	MMU18295
32	cellular oncogene c-fos	HSCFOS	MMCFOS
33	gene for serotonin 1B receptor	HUMHGCR	MUS5HT1B
34	MGAT gene	HUMUDPCNA	MUSGLCNACT
35	gene for ornithine decarboxylase ODC	HSODCG	MUSODCG
36	galactose-1-phosphate uridyl transferase (GALT) gene	HUMGALTB	MMU41282

Group #	Name of gene in human homolog	Homo Loci	Mus Loci
37	prion protein (PrP) gene	HSU29185	MUSPRNPA
38	olfactory marker protein (OMP) gene	HSU01212	MMU01213
39	macrophage migration inhibitory factor (MIF) gene	HUMMIF	MMU20156
40	keratin 13 gene	AF049259	MMU13921
41	H1.2 gene for histone H1	HSH12	MUSHIS1A
42	ACTH-R gene for adrenocorticotropic hormone receptor	HSACTHR	MUSACTHR
43	myogenic determining factor 3 (MYOD1) gene	AF027148	MMMYOD1
44	keratin 18 (K18) gene	HUMKER18	MUSENDOBA
45	platelet alpha-2-adrenergic receptor gene	HUMADRA	MUSALP2ADB
46	gene for MHC encoded proteasome subunit LMP2	HSMHCPU15	MUSLMP2A
47	alpha2-C4-adrenergic receptor gene	HSU72648	MUSADRA
48	midkine gene	HUMMK	MUSMKPG
49	myf4 for skeletal muscle-specific transcription factor	HSMYF4G	MUSMYOGEN
50	histone H1 (H1F3) gene	HUMHISAB	MMHISTH1
51	prepro-oxytocin-neurophysin I (OXT) gene	HUMOTNPI	MUSOXYNEUI
52	thymidine kinase gene	HUMTKRA	MUSTKM
53	beta-2-adrenergic receptor gene	HUMADRBRA	MMB2ARG
54	metallothionein-III gene	HUMMETIII	MUSMETIII
55	XRCC1 DNA repair gene	HUMXRCC1G	MUSXRCC1G
56	zinc finger transcriptional regulator (GOS24) gene	HUMG0S24B	MUSZPF36G
57	alpha-globin germ line gene	HSAGL1	MUSHBA
58	testis-specific PGK-2 gene for phosphoglycerate kinase	HSPGK2G	MUSPGK2
59	fatty acid binding protein FABP gene	HSU57623	MMU02884
60	DNA for arylsulphatase A	HSARYLA	MMDNAASFA
61	CCAAT/enhancer-binding protein delta	S63168	MUSCRP3A
62	gene for 27kDa heat shock protein (hsp 27)	HSHSP27	MUSHSP25A
63	rod outer segment membrane protein 1 (ROM1) gene	HUMROD1X	MUSROM1X
64	somatostatin receptor subtype 3 (SSTR3) gene	HUMSSTR3X	MUSSSTR3A
65	gene for phenylethanolamine N-methylase (PNMT)	HSPNMTB	MUSPNMT
66	gene for insulin-like growth factor II	HSIGF2G	MMU71085
67	adenosine deaminase (ADA) gene	HUMADAG	MMU73107
68	acid sphingomyelinase (SMPD1) gene	HUMSMPD1G	MMASM1G
69	cytochrome c oxidase subunit Vb (COX5B) gene	HUMCOX5B	MUSCYTCOVB
70	nucleolin gene	HUMNUCLEO	MMNUCLEO
71	erythropoietin receptor	S45332	MMERYPR
72	alpha-type insulin and 5' flanking polymorphic region	HUMINSPR	MMINSIIG
73	transforming protein (hst) gene	HUMHST	MMKFGF
74	pulmonary surfactant protein C (SP-C) and SP-C1 genes	HUMPLPSPC	MUSPSPC
75	coseg gene for vasopressin-neurophysin precursor	HSCOSEG	MUSVASNEU
76	gene for beta-3-adrenergic receptor	HSB3A	MMB3A
77	loricrin gene exons 1 and 2	HUMLORI	MMU09189

Group #	Name of gene in human homolog	Homo Loci	Mus Loci
78	hepatocyte growth factor-like protein gene	HSU37055	MUSHEPGFA
79	H1.1 gene for histone H1	HSH11	MUSH1X
80	cytochrome oxidase subunit VIa heart isoformrecursor(COX6AH) gene	HSU66875	MMU63716
81	int-2 proto-oncogene	HSINT2	MMINT2
82	germ line gene for beta-globin	HSBGL3	MUSHBBMAJ
83	leukemia inhibitory factor (LIF) gene	HUMALIFA	MUSALIFA
84	intestinal fatty acid binding protein gene	HUMFABP	MUSFABPI
85	lymphotoxin-beta gene	HUMLYTOXBB	MMU16984
86	LYL-1 protein gene	HUMLYL1B	MMLYL1
87	atrial natriuretic factor (PND) gene	HUMANFA	MUSANF
88	encoding alpha subunit of murine cytokine(MIP1/SCI)	HUMG0S19A	MMSCIMIP
89	intestinal alkaline phosphatase (ALPI) gene	HUMALPI	MUSIAP
90	N-formyl peptide receptor (FPR1) gene	HUMFPR1A	MUSNFORREC
91	S-protein gene	HSSPRO	MMVITRO
92	gene for granulocyte colony-stimulating factor (G-CSF)	HSGCSFG	MMGCSFG
93	tumor necrosis factor-beta (TNFB) gene	HUMTNFBA	MMTNFBG
94	interleukin 10 (IL10) gene	HSU16720	MUSIL10Z
95	21-hydroxylase B gene	HUMCP21OH	MUS21OHA1
96	Mullerian inhibiting substance gene	HUMMIS	MMAMH
97	apolipoprotein E (epsilon-4 allele) gene	HUMAPOE4	MUSAPE
98	regenerating protein (reg) gene	HUMREGB	MUSREGI
99	cathepsin L gene	HUMPROLA	MUSPROL
100	Flt3 ligand and Flt3 ligand alternatively spliced isoform	HSU29874	MMU44024
101	UHS KerA gene	HSA6693	MUSSER1
102	IL2RG gene	HUMIL2RGA	MMU21795
103	C-reactive protein gene	HUMCRPGA	MMCRPG
104	gene for B cell differentiation factor I	HSBCDIFFI	MMIL5G
105	Thy-1 glycoprotein gene	HUMTHY1A	MUSTHY1GC
106	uPA gene	HSUPA	MUSUPAA
107	gene for serum amyloid P component	HUMSAP01	MUSSAPRB
108	pancreatits-associated protein (PAP) gene	HUMPAP	D63360
109	interleukin 1-beta (IL1B)	HUMIL1B	MMIL1BG
110	cathepsin G gene	HUMCAPG	MUSCATHG
111	cytotoxic T-lymphocyte-associated serine esterase 1	HUMCTLA1	MUSSPCTLS
112	alpha-lactalbumin gene	HSLACTG	MUSALCALB
113	DNA for osteopontin	HUMOSTP	MMOESTEOP
114	gene for CD14 differentiation antigen	HSCD14G	MMCD14
115	gap-I gene	HSGAPIGNA	MMU60528
116	gene for bone gla protein (BGP)	HSBGPG	MUSOGC
117	fetal gene for apolipoprotein AI precursor	HSAPOAIA	MUSAICIIIA

Table 9. The results of prediction of human genes by two-species method

	correct # of exons	predicated # of exons	true positive	overlapping	totally missing	false positive
1	6	6	6	0	0	0
2	6	6	6	0	0	0
3	1	1	1	0	0	0
4	5	5	4	1	0	0
5	1	1	1	0	0	0
6	1	1	1	0	0	0
7	8	8	8	0	0	0
8	1	1	1	0	0	0
9	6	6	6	0	0	0
10	4	4	4	0	0	0
11	1	1	0	1	0	0
13	4	4	4	0	0	0
14	3	3	3	0	0	0
15	11	12	10	2	0	0
16	1	1	1	0	0	0
17	1	1	1	0	0	0
18	6	6	4	2	0	0
19	1	1	1	0	0	0
20	7	7	7	0	0	0
21	1	1	1	0	0	0
22	7	7	7	0	0	0
24	1	1	1	0	0	0
25	4	4	4	0	0	0
26	4	4	3	1	0	0
27	1	1	1	0	0	0
28	2	2	2	0	0	0
30	8	8	8	0	0	0
31	1	1	1	0	0	0
32	4	4	4	0	0	0
33	1	2	1	0	0	1
34	1	1	1	0	0	0
35	10	10	10	0	0	0
36	11	11	11	0	0	0
37	1	1	1	0	0	0
38	1	1	1	0	0	0
39	3	3	3	0	0	0
40	7	7	6	1	0	0
41	1	1	1	0	0	0
42	1	1	1	0	0	0

	correct # of exons	predicated # of exons	true positive	overlapping	totally missing	false positive
43	3	3	3	0	0	0
44	7	7	7	0	0	0
46	6	7	5	1	0	1
47	1	1	1	0	0	0
48	4	5	4	0	0	1
49	3	3	3	0	0	0
50	1	1	1	0	0	0
52	7	7	7	0	0	0
53	1	1	1	0	0	0
54	3	3	3	0	0	0
55	17	19	12	3	2	1
56	2	2	2	0	0	0
57	3	2	2	0	1	0
58	1	1	1	0	0	0
59	4	4	4	0	0	0
60	8	7	7	0	1	0
61	1	1	1	0	0	0
62	3	3	3	0	0	0
63	3	3	3	0	0	0
64	1	1	1	0	0	0
65	3	3	3	0	0	0
66	3	4	2	1	0	1
67	12	11	10	1	1	0
68	6	6	6	0	0	0
69	4	4	4	0	0	0
70	14	14	12	2	0	0
71	8	8	8	0	0	0
73	3	3	3	0	0	0
74	5	7	3	2	0	2
75	3	3	3	0	0	0
76	2	2	1	0	1	1
77	1	1	1	0	0	0
78	18	20	17	1	0	2
79	1	2	1	1	0	0
80	3	3	1	0	2	2
81	3	3	3	0	0	0
83	3	3	3	0	0	0
84	4	5	4	0	0	1
85	4	4	4	0	0	0
86	3	4	3	1	0	0

	correct # of exons	predicated # of exons	true positive	overlapping	totally missing	false positive
88	3	4	3	0	0	1
89	11	12	10	1	0	1
90	1	1	1	0	0	0
91	8	8	8	0	0	0
92	5	5	4	1	0	0
93	3	3	3	0	0	0
94	5	5	5	0	0	0
95	10	11	9	1	1	1
97	3	3	3	0	0	0
98	5	5	5	0	0	0
99	1	1	1	0	0	0
100	7	7	6	1	0	0
101	1	1	1	0	0	0
102	8	8	5	3	0	0
103	2	2	2	0	0	0
104	4	4	2	0	2	2
105	3	3	3	0	0	0
106	10	10	7	3	1	0
107	2	2	2	0	0	0
108	5	5	5	0	0	0
109	6	4	2	2	0	0
110	5	5	4	1	0	0
111	5	5	3	1	1	1
112	4	5	4	0	0	1
113	6	6	6	0	0	0
114	2	2	0	2	0	0
115	3	3	3	0	0	0
116	4	4	4	0	0	0
117	3	3	2	1	0	0

Table 10. The results of prediction of mouse genes by two-species method

	correct # of exons	predicated # of exons	true positive	overlapping	totally missing	false positive
1	6	6	6	0	0	0
2	6	6	6	0	0	0
3	1	1	1	0	0	0
4	5	5	4	1	0	0
5	1	1	1	0	0	0
6	1	1	1	0	0	0
7	8	8	8	0	0	0
8	1	1	1	0	0	0
9	6	6	6	0	0	0
10	4	4	4	0	0	0
11	1	1	0	1	0	0
13	4	4	4	0	0	0
14	3	3	3	0	0	0
15	11	12	10	2	0	0
16	1	1	1	0	0	0
17	1	1	1	0	0	0
18	6	6	4	2	0	0
19	1	1	1	0	0	0
20	7	7	7	0	0	0
21	1	1	1	0	0	0
22	7	7	7	0	0	0
24	1	1	1	0	0	0
25	4	4	4	0	0	0
26	4	4	3	1	0	0
27	1	1	1	0	0	0
28	2	2	2	0	0	0
30	7	7	7	0	0	0
31	1	1	1	0	0	0
32	4	4	4	0	0	0
33	1	2	1	0	0	1
34	1	1	1	0	0	0
35	10	10	10	0	0	0
36	11	11	11	0	0	0
37	1	1	1	0	0	0
38	1	1	1	0	0	0
39	3	3	3	0	0	0
40	8	8	6	1	0	1
41	1	1	1	0	0	0
42	1	1	1	0	0	0

	correct # of exons	predicated # of exons	true positive	overlapping	totally missing	false positive
44	7	7	7	0	0	0
46	4	7	4	0	0	3
47	1	1	1	0	0	0
48	4	5	4	0	0	1
49	3	3	3	0	0	0
50	1	1	1	0	0	0
52	7	7	7	0	0	0
53	1	1	1	0	0	0
54	3	3	3	0	0	0
55	17	19	12	3	2	1
56	2	2	2	0	0	0
57	3	2	2	0	1	0
58	1	1	1	0	0	0
59	4	4	4	0	0	0
60	8	7	7	0	1	0
61	1	1	1	0	0	0
62	3	3	3	0	0	0
63	3	3	3	0	0	0
64	1	1	1	0	0	0
65	3	3	3	0	0	0
66	3	4	2	1	0	1
67	11	11	10	1	0	0
68	6	6	6	0	0	0
69	4	4	4	0	0	0
70	14	14	12	2	0	0
71	8	8	8	0	0	0
73	3	3	3	0	0	0
74	5	7	3	2	0	2
75	3	3	3	0	0	0
76	2	2	1	0	1	1
77	1	1	1	0	0	0
78	18	20	17	1	0	2
79	1	2	1	1	0	0
80	3	3	1	0	2	2
81	3	3	3	0	0	0
83	3	3	3	0	0	0
84	4	5	4	0	0	1
85	3	3	3	0	0	0
86	3	4	3	1	0	0
87	3	3	3	0	0	0
88	3	4	3	0	0	1
89	11	12	10	1	0	1
90	1	1	1	0	0	0

	correct # of exon	predicated # of exon	true positive	overlapping	totally missing	false positive
92	5	5	4	1	0	0
93	3	3	3	0	0	0
94	5	5	5	0	0	0
95	10	11	9	1	1	1
97	3	3	3	0	0	0
98	5	5	5	0	0	0
99	1	1	1	0	0	0
100	7	7	6	1	0	0
101	1	1	1	0	0	0
102	8	8	5	3	0	0
103	2	2	2	0	0	0
104	4	4	2	0	2	2
105	3	3	3	0	0	0
106	10	10	7	3	1	0
107	2	2	2	0	0	0
108	5	5	5	0	0	0
109	6	4	2	2	0	0
110	5	5	4	1	0	0
111	5	5	3	1	1	1
112	4	5	4	0	0	1
113	6	6	6	0	0	0
114	2	2	0	2	0	0
115	3	3	3	0	0	0
116	4	4	4	0	0	0
117	3	3	2	1	0	0

Table 11 The identified exons of human and mouse with the method based on two-species comparison (for those gene pairs which have homologs in chicken's sequence)

gene#	correct # of exons	predicated # of exons	true positive	overlapping	totally missing	false positive
h2	6	6	6	0	0	0
3	1	1	1	0	0	0
5	1	1	1	0	0	0
6	1	1	1	0	0	0
7	8	8	8	0	0	0
8	1	1	1	0	0	0
10	4	4	4	0	0	0
13	4	4	4	0	0	0
14	3	3	3	0	0	0
21	1	1	1	0	0	0
22	7	7	7	0	0	0
24	1	1	1	0	0	0
27	1	1	1	0	0	0
28	2	2	2	0	0	0
31	1	1	1	0	0	0
32	4	4	4	0	0	0
33	1	2	1	0	0	1
41	1	1	1	0	0	0
42	1	1	1	0	0	0
43	3	3	3	0	0	0
47	1	1	1	0	0	0
48	4	5	4	0	0	1
49	3	3	3	0	0	0
50	1	1	1	0	0	0
52	7	7	7	0	0	0
53	1	1	1	0	0	0
57	3	2	2	0	1	0
59	4	4	4	0	0	0
61	1	1	1	0	0	0
66	3	4	2	1	0	1
73	3	3	3	0	0	0
76	2	2	1	0	1	1
79	1	2	1	1	0	0
84	4	5	4	0	0	1
86	3	4	3	1	0	0
88	3	4	3	0	0	1
102	8	8	5	3	0	0
106	10	11	7	3	1	0
109	6	4	2	2	0	0
117	3	3	2	1	0	0
m2	6	6	6	0	0	0
3	1	1	1	0	0	0

gene#	correct # of exons	predicated # of exons	true positive	overlapping	totally missing	false positive
5	1	1	1	0	0	0
6	1	1	1	0	0	0
7	8	8	8	0	0	0
8	1	1	1	0	0	0
10	4	4	4	0	0	0
13	4	4	4	0	0	0
14	3	3	3	0	0	0
21	1	1	1	0	0	0
22	7	7	7	0	0	0
24	1	1	1	0	0	0
27	1	1	1	0	0	0
28	2	2	2	0	0	0
31	1	1	1	0	0	0
32	4	4	4	0	0	0
33	1	2	1	0	0	1
41	1	1	1	0	0	0
42	1	1	1	0	0	0
43	3	3	3	0	0	0
47	1	1	1	0	0	0
48	4	5	4	0	0	1
49	3	3	3	0	0	0
50	1	1	1	0	0	0
52	7	7	7	0	0	0
53	1	1	1	0	0	0
57	3	2	2	0	1	0
59	4	4	4	0	0	0
61	1	1	1	0	0	0
66	3	4	2	1	0	1
73	3	3	3	0	0	0
76	2	2	1	0	1	1
79	1	2	1	1	0	0
84	4	5	4	0	0	1
86	3	4	3	1	0	0
88	3	4	3	0	0	1
102	8	8	5	3	0	0
106	10	11	7	3	1	0
109	6	4	2	2	0	0
117	3	3	2	1	0	0

Table 12. The identified exons of human and mouse with the method based on three-species comparsion

gene#	correct # of exons	predicated # of exons	true positive	overlapping	totally missing	false positive
h2	6	6	6	0	0	0
3	1	1	1	0	0	0
5	1	1	1	0	0	0
6	1	1	1	0	0	0
7	8	8	8	0	0	0
8	1	1	1	0	0	0
10	4	4	4	0	0	0
13	4	4	4	0	0	0
14	3	3	3	0	0	0
21	1	1	1	0	0	0
22	7	7	7	0	0	0
24	1	1	1	0	0	0
27	1	1	1	0	0	0
28	2	2	2	0	0	0
31	1	1	1	0	0	0
32	4	4	4	0	0	0
33	1	2	1	0	0	1
41	1	1	1	0	0	0
42	1	1	1	0	0	0
43	3	3	3	0	0	0
47	1	1	1	0	0	0
48	4	5	4	0	0	1
49	3	3	3	0	0	0
50	1	1	1	0	0	0
52	7	7	7	0	0	0
53	1	1	1	0	0	0
57	3	3	3	0	0	0
59	4	4	4	0	0	0
61	1	1	1	0	0	0
66	3	3	2	0	0	1
73	3	3	3	0	0	0
76	2	1	1	0	0	0
79	1	1	1	0	1	0
84	4	4	4	0	0	0
86	3	4	3	1	0	0
88	3	3	3	0	0	0
102	8	5	5	0	2	0
106	10	8	7	1	3	0
109	6	2	2	0	4	0
117	3	3	2	1	0	0
m2	6	6	6	0	0	0
3	1	1	1	0	0	0

gene#	correct # of exons	predicated # of exons	true positive	overlapping	totally missing	false positive
6	1	1	1	0	0	0
7	8	8	8	0	0	0
8	1	1	1	0	0	0
10	4	4	4	0	0	0
13	4	4	4	0	0	0
14	3	3	3	0	0	0
21	1	1	1	0	0	0
22	7	7	7	0	0	0
24	1	1	1	0	0	0
27	1	1	1	0	0	0
28	2	2	2	0	0	0
31	1	1	1	0	0	0
32	4	4	4	0	0	0
33	1	2	1	0	0	1
41	1	1	1	0	0	0
42	1	1	1	0	0	0
43	3	3	3	0	0	0
47	1	1	1	0	0	0
48	4	5	4	0	0	1
49	3	3	3	0	0	0
50	1	1	1	0	0	0
52	7	7	7	0	0	0
53	1	1	1	0	0	0
57	2	2	2	0	0	0
59	4	4	4	0	0	0
61	1	1	1	0	0	0
66	3	3	2	0	0	1
73	3	3	3	0	0	0
76	2	1	1	0	0	0
79	1	1	1	0	1	0
84	4	4	4	0	0	0
86	3	4	3	1	0	0
88	3	3	3	0	0	0
102	8	5	5	0	2	0
106	10	8	7	1	3	0
109	6	2	2	0	4	0
117	3	3	2	1	0	0

References

[1] Stryer L. Biochemistry. 4th Edition. Freeman Press. 1995

[2] Klopfenstein, N. B. et al. (ed.) 1997. USDA Forest Service General Technical Report RM-GTR-297

[3] Introduction to Molecular Biology, http://www.web-books.com

[4]Cooper, G. The Cell – A Molecular Approach. ASM Press, Washington, DC. 1997

[5] The International Human Genome Sequencing Consortium (2001). Initial sequencing and analysis of the human genome. Nature, 409, 860-921

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

[7] Mathé, C *et. al.* Current Methods of Gene Prediction, Their Strengths and Weakness. Nucleic Acids Research, 2002, Vol. 30 No. 19, 44103-4117.

[8] Stormo G. Gene-Finding approaches for Eukaryotes. Genome Research Vol. 10, Issue 4, 394-397, April 2000.

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

[10] Functional and Comparative Genomics Fact Sheet, Human Genome Project Information.

[11] Morgenstern, B., Rinner, O., Abdeddaïm, S., Haase, D., Mayer, K., Dress, A. and Mewes, H.-W. (2001). Exon prediction by comparative sequence analysis. In: The Human Genome Meeting 2001, Edinburgh, Programme and Abstract Book pp. 146-147.

[12] Batzoglou, S., Pachter, L., Mesirovi, J. P., Berger, B. and Lander, E. S. (2000). Human and mouse gene structure: comparative analysis and application to exon prediction. Genome Res. 7, 950-958.

[13] Novichkov, P. S., Gelfand, M. S. and Mironov, A. A. (2001). Gene recognition in eukaryotic DNA by comparison of genomic sequences. Bioinformatics 17, 1011-1018.

[14] Wiehe, T., Gebauer-Jung, S., Mitchell-Olds, T., and Guigo, R. SGP-1: Prediction and validation of homologous genes based on sequence alignments. Genome Research 11, 1574-1583.

[15] NCBI websit, http://www.ncbi.nlm.nih.gov.

[16] Altschul, SF (1990) Methods for assessing the statistical significance of molecular sequence features using general scoring schemes. PNAS (USA) 87: 2264-2268.

[17] Mayor C., Brudno M., Schwartz J. R., Poliakov A., Rubin E. M., Frazer K. A., Pachter L. S. and Dubchak I. (2000) *VISTA: Visualizing Global DNA Sequence Alignments of Arbitrary Length.* Bioinformatics, 16:1046.

[18] Thompson, J.D., Higgins, D.G. and T.J. Gibson CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions- specific gap penalties and weight matrix I choice. Nucleic Acids Research, 22:4673-4680, 1994.

[19] Otu, H. and Sayood, K. (2002). A New Sequence Distance Measure for Phylogenetic Tree Construction.

[20] Salzberg, S.L., A method for identifying splice sites and translational start sites in eukaryotic mRNA. Comput. Appl. Biosci. 13, 365-376, 1997.

[21] Ziv, J., Lempel, A. (1978). Compression of individual sequences via variable-rate coding. IEEE Tran. Inf. The., 24:530-536.

[22] Ziv, J., Merhav, N. (1978). A measure of relative entropy between individual sequences with application to universal classification. . IEEE Tran. Inf. The., 39: 1270-1279.

[23] Gusfield, D. Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, Cambridge, UK, 1997.

[24] Ziv, J., Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Tran. Inf. The.,* 23:337-343.

[25] Fickett, J., The gene identification problem: an overview for developers. Computers and Chemistry, 20: 103-118, 1996.

[26] Burset, M. and Guigó, R. (1996). Evaluation of gene structure prediction programs. Genomics 34, 353-367

[27] Kececioglu, J.and Ravi, R. (1995). Of mice and men. Evolutionary distances. In Proceedings of the 6th ACM-SIAM Symposium on Disctrete Algorithms, pages 604-613.

[28] Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.,* 4:406-425.