Gene Finding in *Homo Sapiens*

(MBB452 Genomics and Bioinformatics Term Project 2005)

Yuk-Lap Yip (Kevin)
Department of Computer Science

This article reviews the existing computational techniques for gene finding, and proposes appropriate approaches to finding human (*Homo Sapiens*) genes. Traditionally, genes are identified by experimental methods that are accurate but labor intensive and time consuming. In recent years, with large amount of sequence data being generated by high-throughput sequencing methods, it has become popular to find genes directly from the sequences using computational methods. Given a DNA sequence as input, the methods try to predict the locations of all the genes and their substructures (exons, introns, etc.) containing in it. These methods are less reliable than the experimental methods, but are much more efficient and scalable. A common protocol for gene finding nowadays is to use computational methods to quickly identify putative genes from the target sequences, and then verify and probably make corrections on them by experimental methods. Our approach also follows this protocol.

Finding genes computationally is especially difficult for predicting eukaryotic (including human) genes, due to their complicated gene structures. In the past 20 years, many computational methods for gene finding have been proposed. In the following we review some of the most important techniques involved, and explain how we are going to use them to identify human genes.

The first important technique is database searching. This includes several methods. The first is to query the input DNA sequence against a protein database using tools such as BLASTX. The presence of proteins with significant matching scores would be a strong indicator that the sequence may contain a gene that codes for a similar protein. The exact location of the gene (more specifically, the CDS) can then be derived from the protein sequences. Similarly, the input DNA sequence can be used to search an EST library. The method is simple, and reliable in cases with very significant scores. An obvious shortcoming is the need to have the proteins or ESTs available in the database. Also, alternate splicing in human genes makes the search more complicated and error-prone.

Another database searching method is to assume that the sequence contains a gene, and search for a homolog whose sequence has been identified. Again, this requires the homologous sequences be already in the database. A related method is to perform sequence alignment between the input sequence and some sequences of some close species, such as mouse in our case, and look for highly conserved regions. Using the assumption that coding regions are much better conserved than non-coding regions, the regions with significant alignment scores can be treated as putative genes for further analysis. Of course, a fundamental weakness of these methods is that they can only identify genes with homologs, which has been shown in some studies to be unlikely for some newly identified genes.

The accuracy of the methods just described can be affected by uninformative repetitive patterns in the human genome. We will therefore need to filter out such patterns before running any gene finding programs. Many algorithms have been developed for this purpose.

As discussed, it is likely that many genes cannot be identified correctly by database searches. Therefore we will also use *ab initio* methods, methods that predict gene locations from the input sequence alone. All such methods look for some gene signals and content statistics in the sequence, and make predictions based on them. A signal is a consensus sequence that is highly conserved in some substructures, including the start codon ATG, the stop codons TAA, TAG and TGA, the donor site GT junction, the acceptor site AG junction, the poly A tail, and the TATA-box in some promoter sites. The programs look for such signals in the input sequence to determine the potential locations of different parts of the gene in the sequence.

A content statistic is a measure that can be used to determine the chance that a certain part of a sequence being belonged to a certain substructure. For example, human exhibits a biased usage of codons in exons. This bias can be captured by some statistics based on the data from known human genes. By counting the number of times that different codons appear in a given region, the chance that the region is within an exon can be estimated. Many other content statistics have been considered in the literature, including measures on 3-periodicity of bases, dinucleotide occurrence, and length distributions.

Gene-finding algorithms use the feature values (signals and content statistics) to predict the location of genes and their substructures in an input sequence. The predictions can be done by various means. Some early systems use some predefined rules to combine the feature values. The programs are relatively simple, but the prediction results are not always satisfactory. This is probably due to our incomplete understanding of the gene properties.

A natural idea is then not to define the rules explicitly, but to let the programs learn them from some sample data. This corresponds to the well-posed classification problem in machine learning. The task is to predict which “class” (intron/exon, etc.) each nucleotide belongs to. The problem has been studied for a long time, and many algorithms have been proposed. In one of the earliest attempts, the GRAIL system used a simple neural network to combine the feature values to produce classification rules. The results were surprisingly good. This led to the attempts of using other machine learning techniques for the task, including decision trees and discriminant analysis.

While the new algorithms kept improving the accuracy, there was a fundamental problem that hindered them to overcome a certain accuracy bottleneck. The problem is that each nucleotide is individually predicted, and so the global gene structure is not captured well. Some methods use dynamic programming to impose some rules (e.g. an internal exon must be preceded by an acceptor site and followed by a donor site) on the overall gene structure. A breakthrough was then brought by the use of hidden Markov models (HMMs) in modeling gene structures. Since the introduction of HMMs to gene finding, the prediction accuracy of whole-genes has been improved significantly. Some of the methods make use of generalized hidden Markov models (GHMMs) in which each state in the HMM can be yet another HMM or even a separate program. Their high modeling power also makes it possible to make quite accurate predictions when a testing sequence contains a partial gene or multiple genes.

Although modern gene finding algorithms have a much higher accuracy than their ancestors 20 years ago, there is still quite some distance from achieving 100% accuracy. In order to make a further boost of the accuracy, some researchers have proposed to utilize more kinds of data in training the models, such as constructing auxiliary phylogenetic trees in evolutionary hidden Markov models (EHMMs) from multiple aligned sequences. Some other researchers have proposed combining the prediction results of multiple algorithms, such as taking the union of all predictions with high likelihood values, and intersecting those with low likelihood values. Both approaches have received some success.

Our proposal goes in line with the idea of the last approach: to combine multiple methods. We will use both database searching and machine learning methods. For the latter, we will include methods that have been shown to perform well, and of which the prediction results complement each other. Methods that do not perform well will not be included, since they are unlikely to contribute to the overall prediction accuracy. Likewise, methods that are too similar or give too similar predictions will not be included simultaneously. For instance, it has been shown in a survey that the methods HMMGene and Genescan both perform well, and they are good complements to each other. Predictions that are consistently made by most methods that we will use are likely to be reliable. For those that are predicted by only one or a few methods, some prediction scores obtained from the methods (e.g. the E-value of BLAST searches and the likelihood of HMM-based methods) can be used to estimate the prediction reliability and the predictions from various methods can be combined as described. The confidence scores of some algorithms, including HMMGene and Genescan, have been shown to be highly correlated to the prediction accuracy.

There is a tradeoff between the sensitivity and specificity of the algorithms. Using tighter quality thresholds make the results more specific, but less sensitive, and vice versa. Our approach is to use some loose thresholds at the beginning in order to have the results covering as many real genes as possible. Within the results, those receiving high confidence scores have a better chance of being correctly predicted. They will be given higher priority to be verified by experimental methods.

The predictions can be verified by RT-PCR, cDNA-library screening, exon trapping and Northern blot analysis. Since there will be many predicted genes to be verified, high-throughput microarrays can also be used. For example, the prediction results can be probed to a microarray, and be hybridized against randomly primed cDNA.