Last modified August 19, 2015

Elementary Sequence Analysis

Brian Golding, Dick Morton and Wilfried Haerty

Department of Biology
McMaster University
Hamilton, Ontario
L8S 4K1

These notes are in Adobe Acrobat format (they are available upon request in other formats) and they can be obtained from the website http://helix.biology.mcmaster.ca/courses.html. Some of the programs that you will be using in this course and which will be run locally can be found at http://evol.mcmaster.ca/p3S03.html.

The "blue text" should designate links within this document while the "red text" designate links outside of this document. Clicking on the latter should activate your web browser and load the appropriate page into your browser. If these do not work please check your Acrobat reader setup. The web links are accurate to the best of our knowledge but the web changes quickly and we cannot guarantee that they are still accurate. The links designated next to the JAVA logo,  , require that JAVA be installed on your computer.

These notes are used in Biology 3S03. The purpose of this course is to introduce students to the basics of bioinformatics and to give them the opportunity to learn to manipulate and analyze DNA/protein sequences. Of necessity only some of the more simple algorithms will be examined.

The course will hopefully cover . . .

- databases of relevance to molecular biology.
- some common network servers/sites that provide access to these databases.
- methods to obtain sequence analysis software and data.
- methods of sequence alignment.
- methods of calculating genetic distance.
- methods of phylogenetic reconstruction.
- methods for detecting patterns and codon usage.
- methods for detecting gene coding regions.

The formal part of the course will consist of two approximately one hour lectures each week. Weekly assignments will be be provided to practice and explore the lecture material. In addition there will be an optional tutorial to help students with these assignments or other problems. These assignments will be 40% of your grade and three, in class quizzes will make up the remainder.

We would appreciate any comments, corrections or updates regarding these notes.

Golding@McMaster.CA Morton@McMaster.CA HaertyW@McMaster.CA

This document was entirely constructed with pdfLaTeX. Enormous kudos to all those that make this great software free to everyone.

Table of Contents in Brief

In order to speed download, I place here links to the individual chapters in pdf format. The contents of these are shown on the following 'Contents' pages but note that the links will function only for the individual chapter included here.

Contents

Chapter 11

Exon Analysis

Locating protein-coding genes is an important goal of genomics. This, together with locating RNA genes and regulatory elements is the process of annotation. Annotating DNA is based on three tools; 1) aligning cDNA with genomic DNA, 2) similarity to previously identified genes and 3) theoretical prediction. Annotating the human genome is an ongoing process. The 3×10^9 bp of DNA is estimated to contain the order of 3×10^4 genes. Approximately 1×10^4 complete cDNA sequences have so far been identified. It is likely that complete cDNA sequences will never be obtained for all genes so that computational techniques will be necessary to obtain a complete understanding of its coding potential.

11.1 Open Reading Frames

Prediction of protein-coding genes is primarily based on identifying open reading frames (ORF). Many programs determine open reading frames, among them the software at http://www.ncbi.nlm.nih.gov/gorf/gorf.html and "Translate" on the ExPasy Molecular Biology Server (http://expasy.org/tools/dna.html). This however, is only the first step. There are a number of problems in determining if an ORF is actually used to code for protein.

1. Sequencing errors, internal "stop" codons that are removed by editing, and codons for selenomethionine.
2. Spurious ORFs that are not part of any protein-coding gene. The non-coding strand of exons often contains ORFs. That is, the reverse complements of stop codons (TTA [TAA], CTA [TAG], TCA [TGA]) are often statistically avoided, creating ORFs on the complementary strand.
3. Intron exon structure combines several ORFs into a single gene. The splice junction fusion may create the in-phase codon.
4. Splicing creates multiple transcripts and multiple proteins. Certain exons may only be used in a subset of transcripts. The *D. melanogaster Adh* gene, for example has different transcripts during larval and adult phases of growth.

11.2 Gene Recognition

Two general approaches are used to recognize genes within a DNA sequence.

Local alignment methods such as BLAST detect sequence similarity to ESTs or genes already in a database. These are very powerful at finding the approximate location of exons, but do not accurately determine their boundaries. Nor can they combine exons into a gene without additional information.

Global approaches calculate a vector that estimates the protein-coding capacity of a window within the sequence. This vector is simply a one-dimensional array of numbers that incorporate various features that the algorithm uses to determine protein-coding capacity. The measured vector is compared to one obtained from a set of standard genes.

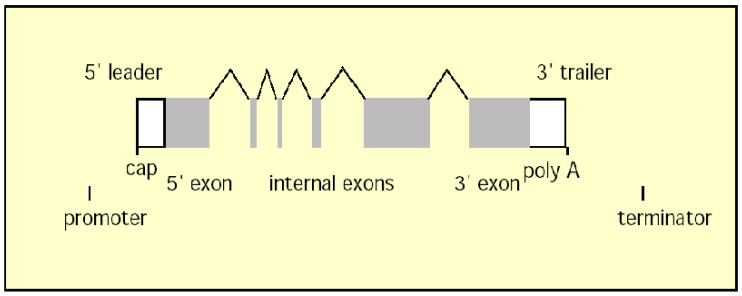

Figure 11.1: Eukaryotic gene with exon - intron structure, protein-coding is gray.

Both of these approaches are combined in annotation software that is used for complete genome annotation. An example is GenomeScan, used extensively to annotate the human genome sequence (Yeh *et al.* 2001. Genome Res. 11: 803-816). Gene prediction combines the location of ORFs with other sequence information to make a model of the entire gene. Data about possible promoters, transcription initiation (cap sites), translation signals (initiation, termination codons), splice signals, and transcription terminators are combined to make an inference that rejects unlikely ORFs and includes likely ORFs in a consistent gene model (Figure 11.1).

Gene prediction algorithms calculate an overall statistic and make a decision as to whether or not to present the model as a potential gene. Neural network methods are often used in which the algorithm is trained on a set of test genes and learns what weights should be assigned to the various measures in order give the best discrimination between valid and invalid test genes.

The ability of various approaches to predict protein-coding genes was assessed by Fickett and Tung (1992. Nucleic Acids Res. 20: 6441-6450). They identified several features that are particularly useful.

1. Codon usage. A codon usage vector (frequencies of the 64 possible codons) for a potential exon is compared to that of a reference sets of genes, preferably from the same or closely related organism. Methods differ in how the reference set is obtained and how the measure of fit is calculated. Reference sets that incorporate information about the amino acid composition of the potential gene are superior to those that do not.

2. In-phase words. A vector similar to the codon vector is calculated for longer words (oligonucleotides of length n). Hexamers have proven useful. These take into account tendencies of codon use to be correlated over short ranges (e.g., a codon ending in G tends not to be followed by one beginning in G).

3. The presence of STOP codons. Most methods only consider ORFs. However, it is possible to incorporate stop codons into a measure of amino acid content.

4. Amino acid content. Measures of protein function, such as vectors of amino acids, dipeptides and hydrophobicity, can be obtained for a potential exon. Like the codon usage vectors, these are compared to a reference set. This, however, may limit identification to particular types of protein-coding genes.

5. Nucleotide periodicity. Nucleotides do not appear at random in coding sequences (nor in non-coding ones). The statistical average codon is RNY, leading to a periodicity of 3 nucleotides. Periodicity vectors are calculated for potential exons (e.g., using Fourier transforms or autocorrelation functions).

11.2.1 Splice Sites

Gene prediction programs must locate splice sites in genes that have exon / intron organization. Information about the splice site is mainly contained within a few nucleotides of the boundary. The dinucleotides ...GT and AG... form the canonical splice sites of most exon-intron junctions (Figure 11.2). The method of sequence logos has been used to represent the contributions of various positions to the information content of human splice sites.
http://www.lecb.ncifcrf.gov/ toms/gallery/SequenceLogoSculpture.gif

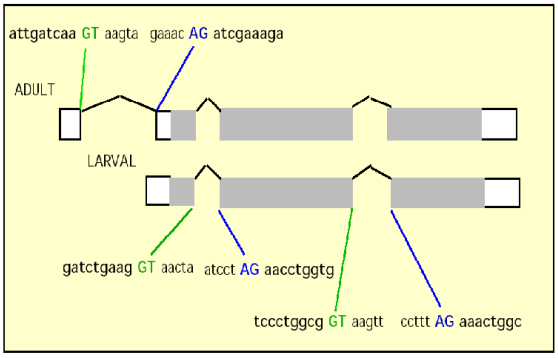

Figure 11.2: Exon - intron boundaries of the *D. melanogaster* Adh gene.

11.2.2 Codon Usage

Codon use and nucleotide periodicities are interdependent properties of protein-coding regions that influence exon prediction.

Base Composition. Base composition is a major factor influencing codon usage. Organisms, especially bacteria, have variable GC content. This alters both the types of amino acids and the codons used to code for these amino acids As an example, AAA and AAG both code for lysine. As the genome content of (A+T) increases, proteins tend to use more lysine and more AAA codons (Figure 11.3).

This trend across genomes is repeated within a genome across different genes, although with much more variability. To illustrate, *E. coli* genes that have greater (A+T) content tend to use more AAA codons (Figure 11.4).

Mutational bias is thought have a major effect in determining overall base composition. Other influences, such as selection for compact genome size, have also been suggested.

Mutational bias could reflect replication error, repair efficiency, nucleotide pools or other, unidentified factors. The causes of high, low or intermediate GC content among organisms are not known. Neither are the causes of variation among genes within a genome. Amino acid composition is an obvious possibility, but even with constant composition, GC content can vary because of synonymous codon choice. The problem of GC content and codon choice is a chicken-or-egg situation. They are correlated, but which is driving which and what are the underlying forces?

Codon Position. Codon choice is patterned differently at each of the three codon positions (c1, c2, c3). Figure 11.5 shows nucleotide choices for *E. coli*. The average nucleotide frequencies of all genes are to the right of histograms showing deviations from this average at each position.

In all organisms, G is preferred in the first position. T and, less obviously for *E. coli*, A are avoided. The second position is less consistent, but A is often preferred, especially at moderate or high GC content. The third (synonymous) position shows most clearly the effect of variable GC content. In organisms with high GC content, G and C are preferred in the third position, but are avoided in organisms with high AT content. In *E. coli*, which has an even distribution of nucleotides, G and C are slightly preferred and A slightly avoided.

The choice of codon at the second position is very dependent on the hydrophopicity of the protein because of a pattern in the universal genetic code. T (U in RNA) at c2 is confined to hydrophobic amino acids, while A at c2 is confined to hydrophilic ones.

The effect of this bias in the genetic code is clearly seen in the distribution of nucleotides at c2 in the *E. coli* genome (Figure 11.6). There is a peak of relatively hydrophobic proteins that prefer T (U in RNA) instead of A.

The patterns of codon use described above are complex, but they are not taken individually into account by gene prediction programs. Rather they create trends in protein-coding regions that are utilized by algorithms as frequency distributions of "words" (for example, hexamer frequencies).

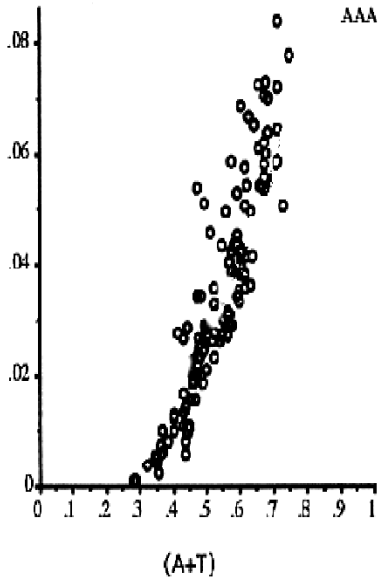

Figure 11.3: The fraction of all codons that are AAA across genomes with different AT contents.

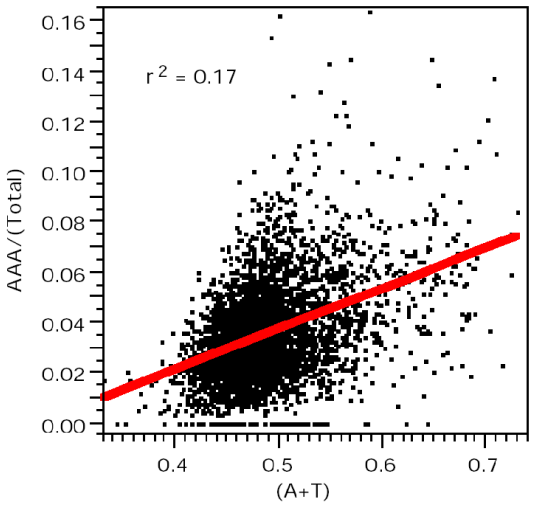

Figure 11.4: The fraction of codons that are AAA for genes of the *E. coli* genome as a function of the gene's (A+T) fraction.

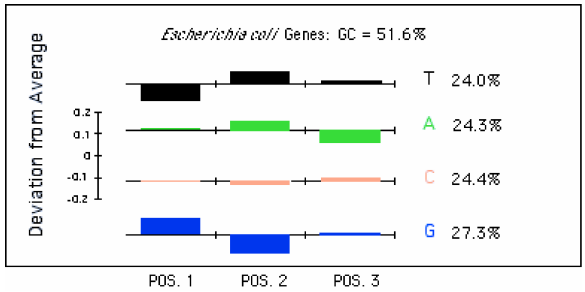

Figure 11.5: Nucleotide composition by codon position for *E. coli* genes.

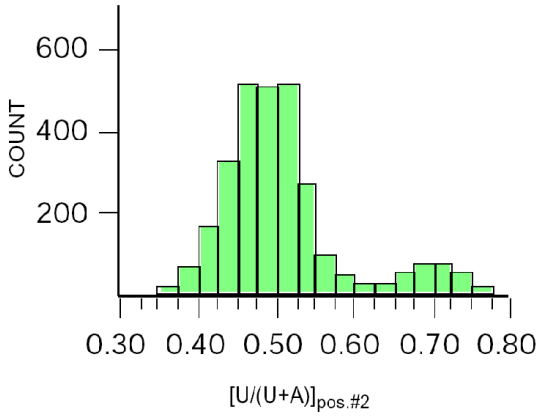

Figure 11.6: The relative content of T to (T+A) at the second position of 3180 *E. coli* genes.

11.2.3 Gene Prediction Software

Many programs are available to build gene models, such as FGENEH, GENMARK, GRAIL, GeneParser. Burset and Guigo (1996. Genomics 34: 353-367) and Guigó *et al*. (2000. Genome Res. 10: 1631-1642) compared many of them and found that their accuracy is often overrated because they have been evaluated on genes similar to the test set used to build the discrimination functions. Three of the most commonly used programs are summarized.

GeneFinder is a group of programs for gene identification written by Victor Solovyev's group of the Computational Genomics Group at the Sanger Centre (Solovyev V and Salamov A. 1997. Proc Int Conf Intell Syst Mol Biol 5:294-302). They were used to predict genes in the *Drosophila* genome (Solovyev V and Salamov A. 2000. Genome Res. 10: 516-22). The software can be accessed for testing at the commercial site http://www.softberry.com/berry.phtml. FGENES is designed to identify and piece exons together to predict multiple genes on both strands. There is a version, FGENES-M that predicts multiple models of a single gene, useful if there are alternate splice forms. FGENESH is a variant using a Hidden Markov Model (HMM, section 11.2.4). FGENESH+ is a program that uses a protein sequence similar to the predicted gene product (possibly obtained from BLAST) in conjunction with FGENESH to more accurately predict exon structure.

FGENES relies on identifying exon donor and acceptor splice sites as described by Solovyev *et al*. (1994. Nucleic Acids Res. 22: 5156-5163). Flanking (5′ and 3′) and internal exons are treated with separate algorithms. The program examines each ORF that terminates in a GT or begins with AG and calculates a linear discriminant function, $z = \sum \alpha_i x_i$, where x_i are measures of a splice site and α_i are weights. The discriminant function is used to classify an exon as valid if z is above a critical value determined from the analysis of test (learning) data. The measures in the discriminate function are triplet nucleotides frequencies at the exon-intron boundaries. Because these are organism dependent, discriminant function weights must be obtained for each species or from a closely-related relative.

GENIE is a program written by the Computational Biology Group at the University of California, Santa Cruz and the Genomic Informatics Group at LBNL (Kulp D, Haussler D, Reese MG, Eeckman FH. 1996. Proc. Int. Conf. Intell. Syst. Mol Biol. 4:134-42). It uses a Generalized Hidden Markov Model (HMM, section 11.2.4) to develop gene models. It has been extensively used to predict genes in the human and fruitfly genomes (Reese MG, Kulp D, Tammana H, Haussler D. 2000. Genome Res. 10:529-38). The web version of Genie is available through the Berkeley *Drosophila* Genome Project (http://www.fruitfly.org/seq_tools/genie.html).

GENSCAN is a program developed by Burge and Karlin (1997. J. Mol. Biol. 268: 78-94). Although designed for human genes, it has been tested successfully on other vertebrate sequences and plants. It also works for *Drosophila*. A large, non-redundant set of human genes (2.58×10^6 nucleotides containing 1492 exons and 1254 introns) was used to develop GENSCAN. GENSCAN is generally regarded as one of the best gene prediction programs and has been extensively used in the human genome project. It incorporates a number of features to build a model.

1. Transcriptional and translational signals are evaluated by weight matrices. Potential signals are: polyadenylation, cap site, promoter (both TATA and TATA-less promoters are allowed with variable distance to the cap site), translational start sites (6 nt prior to start codon) and stop sites (3 nt following stop codon).

2. Splice signals. A modified weight matrix method is used to examine potential splice sites (3 nt in exon, 6 nt in intron). The modified method takes into account correlations between positions.

3. Exon models. Potential coding portions of exons are evaluated using a Markov model. This computes transition probability matrices for hexamers ending at each codon position. Scores are dependent on similarity between the GC-content of the training sequences and the sequence to be evaluated. GENSCAN uses one of two sets of expected transition probabilities that were generated from training sets having either $GC < 43\%$ or $GC > 43\%$.

The internet site for GENSCAN is (http://genes.mit.edu/GENSCAN.html).

Each of the programs described above uses general features of genes (Fig. 11.1) to develop its model. They derive their parameters from analyzing a group of example genes and will perform best if the target gene is similar. Another, potentially more powerful, approach is based on homology to closely related genes. In fact, it is even better to combine this with gene prediction methods. GENOMESCAN is an outgrowth of GENSCAN that evaluates a gene model by making it's probability conditional on similarity results from a BLASTX search of a protein database. In this respect it is similar to FGENESH+,

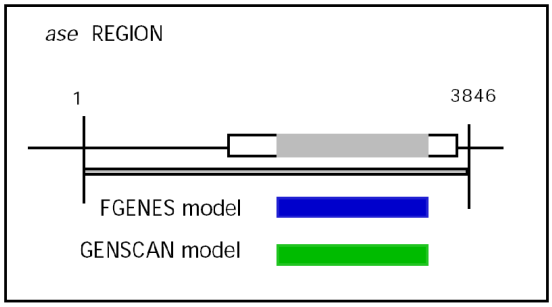

Figure 11.7: Gene models of the *D. melanogaster* ase region.

but more comprehensive and designed for genome annotation. It has been used in annotating the human genome project. It may be accessed at http://genes.mit.edu/genomescan.html. In the web version, you are required to input a similar protein sequence (rather than having the program obtain sequences from BLASTX).

11.2.4 Hidden Markov Models (HMM)

Hidden Markov Models are statistical methods for evaluating sequences labeled with biologically relevant information. Data may be promoter sites, exon positions and termination signals for a gene model. The gene model can be thought of as represented by an array of symbols (called a parse), To obtain the probability distribution for possible arrays of symbols, a "hidden" set of transition states and transition probabilities between these hidden states is assumed. This allows the parse of maximum likelihood to be obtained. HHM must be trained on a set of learning sequences in order to obtain the hidden transition probabilities.

11.2.5 Comparison of Programs

Figure 11.7 shows how FGENES and GENSCAN performed on the *D. melanogaster* asense gene (*ase*, accession: X52892), which does not have introns. The protein-coding portion of the exon was correctly defined and the 486 amino acid gene product returned. This is all that can be expected since none of the gene prediction programs can find non- translated regions of transcribed RNA.

The *Drosophila* Adh region has been extensively examined by genetically (Ashburner M. 1999. Genetics 153:179-219) and results compared with gene prediction programs. Figure 11.8 shows that both FGENES and GENSCAN precisely defined the three protein-coding Adh exons and combined them to give the Adh gene. The correct amino acid sequence of ADH was deduced. The adult promoter was not located, perhaps because it is too far from the first protein-coding exon, but the larval promoter was found. Neither the portion of the outspread exon at the beginning of the sequence nor the adh-dup exons at its end were located by FGENES. The polyA site was incorrectly located in the $3'$ mRNA trailer, however, it is possible that other, shorter transcripts exist that use this site. GENSCAN was unable to locate the outspread exon. Like FGENES, it performs poorly at the boundaries of sequences. It did, however, make a good attempt at the adh-dup exons, locating the beginning of the second exon correctly, but not the first. Interestingly, GENSCAN identified a potential exon at nucleotide position 1388-1566. This region of the DNA sequence has high complexity and GC content.

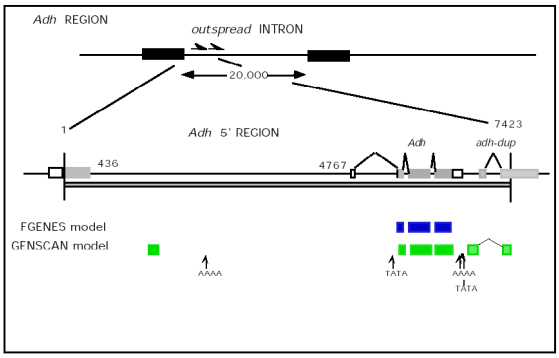

Figure 11.8: Gene models of the *D. melanogaster* Adh region.