

Tuning Gene Prediction to Specific Genomes

Ian Korf, Washington University, St. Louis MO

Abstract

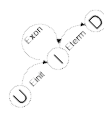
In general, eukaryotic protein coding genes have similar structures (eg. a promoter, exons, introns, a poly-A site, etc.). And in general, there are certain invariant features such as the genetic code and the GT-AG rule of introns. However, there are also particular features that distinguish a specific genome from the generalities. For example, mammals have CpG islands, many repetitive elements, and introns characterized by a branch-point consensus and poly-pyrimidine tract. This is quite different from *Caenorhabditis elegans*, which has trans-splicing, genes organized in operons, few repetitive elements, and A/T-rich introns without a poly-pyrimidine tract or branch point consensus. Even in a clade of closer relatives, such as the flowering plants, one can find large differences in such features as the distribution of repetitive elements. In order to maximize the accuracy of gene prediction algorithms, it is important to take genome-specific features into account. In practice, however, people often use the same gene prediction program with identical parameters on dissimilar genomes. I am developing an ANSI C software library that facilitates creating gene prediction algorithms that are tuned to specific genomes. The library is designed with flexibility in mind. One can choose among various kinds of sequence models and these can be connected in a way that resembles the target genome. A useful feature of the sequence models is that they support external definitions. This enables experts to control the behavior of the program in specific regions before the optimization algorithm decodes the sequence into gene structures. For example, given a collection of reliably aligned transcripts, one can create non-canonical splice sites or increase the scores of confirmed splice sites. Similarly, one might boost exon scores that overlap protein homologies or cloak exons in the vicinity of a pseudogene. I will present progress on the library and on applications for parameter estimation and gene prediction.

Credits

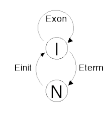
I'd like to thank my post-doc advisors Warren Gish and Michael Brent and the members of their labs for useful discussions, the Department of Genetics at the Washington University School of Medicine and the Department of Computer Science at Washington University for providing a stimulating research environment, www.arabidopsis.org, www.wormbase.org, and Roderic Guigo for providing easily accessible data, Chris Burge for providing Pictogram and Genscan, Apple Computer for making beautiful hardware and software (this poster, including software development and experiments, was "made on a Mac"), the open source community for providing operating systems and development tools for free, and especially the National Human Genome Research Institute who made this research possible through a Genome Scholar and Faculty Transition Award (HGR-00064-01).

Gene Prediction

How much is gene prediction accuracy affected by genome-specific parameters? It depends on the genome, but potentially quite a bit. In the tables below, sensitivity (SN) and specificity (SP) are calculated at the nucleotide level. Note: as expected, the single-gene model outperforms the multi-gene model by a little on these single-gene sequences.



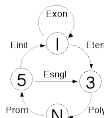
Training Set	Testing Set					
	Human		Plant		Worm	
	SN	SP	SN	SP	SN	SP
Human	86.4	84.5	36.4	96.4	51.8	96.4
Plant	83.4	27.1	96.8	98.0	88.7	88.0
Worm	81.7	42.3	88.4	94.9	90.1	97.3



Training Set	Testing Set					
	Human		Plant		Worm	
	SN	SP	SN	SP	SN	SP
Human	87.4	82.9	35.7	98.3	51.2	95.9
Plant	80.8	25.9	94.7	97.4	86.9	86.4
Worm	82.0	39.2	84.8	93.8	89.2	96.2

It looks like *A. thaliana* and *C. elegans* are related in some way. Is the effect here just different intron lengths? I've tried varying the expected intron length from 100 to 800 and repeated the experiments above. Longer intron lengths decrease sensitivity and increase specificity, but the effect is trivially small. The sequence content appears to be much more important than the length parameter.

How do these gene finders compare to Genscan? The simple models above do not work as well as Genscan on human sequences but perform as well or better on *A. thaliana* and *C. elegans* sequences.



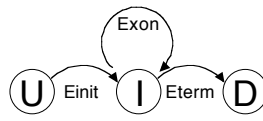
Genscan param file	Testing Set					
	Human		Plant		Worm	
	SN	SP	SN	SP	SN	SP
Human	96.2	87.5	71.8	96.6	84.9	92.8
Plant	97.4	76.2	85.8	96.7	92.0	92.3

Genome Models

The genome models my software supports are Hidden Markov Model (HMM) variants. Here, circles correspond to states with geometric length distributions and arrows correspond to states with defined distributions.

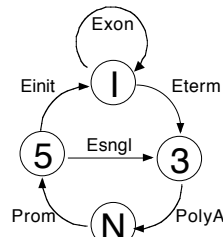
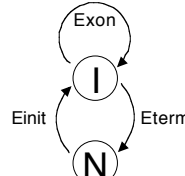
Simple Single-Gene Model

This model generates a single multi-exon gene. There are states for upstream (U), intron (I), downstream (D), and exons (Einit, Exon, Eterm). The complex network of states connecting exons and introns of different phases is not shown.



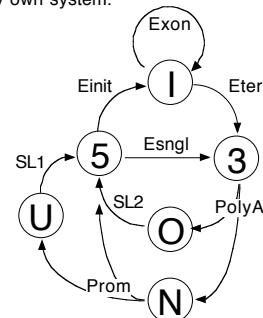
Simple Multi-gene Model

This model generates any number of multi-exon genes. There are states for intergenic (N), intron (I), and exons (Einit, Exon, Eterm).



Genscan Model

This is the model used by Genscan. Its structure is richer than those above, providing states for single exon genes (Esngl), promoters, poly-A sites, 5'UTRs and 3'UTRs. I have not yet tested this model in my own system.



C. elegans Genome Model

Here is an example of the kind of specific genome model I plan to implement in the future. Some *C. elegans* genes contain an SL1 trans-spliced leader at the 5' end of their transcripts. And some genes are organized in co-transcriptional operons. The downstream genes in operons are promoter-less and contain an SL2 trans-spliced leader at their 5' end.

External Definitions

My sequence models support user-defined values at any point in the sequence. I call these external definitions, and they are useful for three reasons.

(1) Runtime Replacement

External definitions allow one to swap out entire models at runtime. This means that if someone comes up with the world's best splice acceptor scanner, this can be used in place of the one defined in the parameter file.

(2) Experts Users & Expert Systems

Despite roughly 20 years of research, gene prediction algorithms are still not accurate enough to derive a proteome from a genome. Genome centers often use expert annotators to determine the true structure of the genes in the sequences they produce. And large-scale annotation providers like Ensembl and Celera have complex evidence-based expert systems to guide their gene assemblies. External definitions provide a mechanism for expert users and expert systems to communicate suggestions and constraints in a straightforward manner.

(3) Breaking Rules

There are several universal rules in biology, like the genetic code and the GT..AG rule of introns. But real biology sometimes violates these rules. Accounting for all the minor exceptions is a computational burden, so the generic rules are usually followed in computational gene prediction. External definitions allow one to break the rules when you want to. (A) shows the true structure of a *C. elegans* gene which is predicted correctly (B). Note that the initial exon ends at 220 with a score of 291. I mutated the canonical GT to a GC at 221 and ran the program again. The initial exon in (C) now ends at 178 and the score drops from 291 to 148 because a poor splice donor is in the maximum likelihood path. In (D) I ran the program again, this time defining the GC to have a score of 1000. This restores the correct structure and artificially raises the score to 1164.

A	~T14F9.4					
	Einit 101	220		T14F9.4		
	Exon 419	520		T14F9.4		
	Exon 576	749		T14F9.4		
	Exon 1573	1770		T14F9.4		
	Exon 1826	2413		T14F9.4		
B	Einit 101	220	+	291	0	0
	Exon 419	520	+	365	0	0
	Exon 576	749	+	377	0	1
	Exon 1573	1770	+	602	0	2
	Exon 1826	2413	+	790	0	0
	Eterm 2468	2569	+	207	0	0
C	Einit 101	178	+	148	0	0
	Exon 419	520	+	365	0	0
	Exon 576	749	+	377	0	1
	Exon 1573	1770	+	602	0	2
	Exon 1826	2413	+	790	0	0
	Eterm 2468	2569	+	207	0	0
D	Einit 101	220	+	1164	0	0
	Exon 419	520	+	365	0	0
	Exon 576	749	+	377	0	1
	Exon 1573	1770	+	602	0	2
	Exon 1826	2413	+	790	0	0
	Eterm 2468	2569	+	207	0	0

Sequence Models

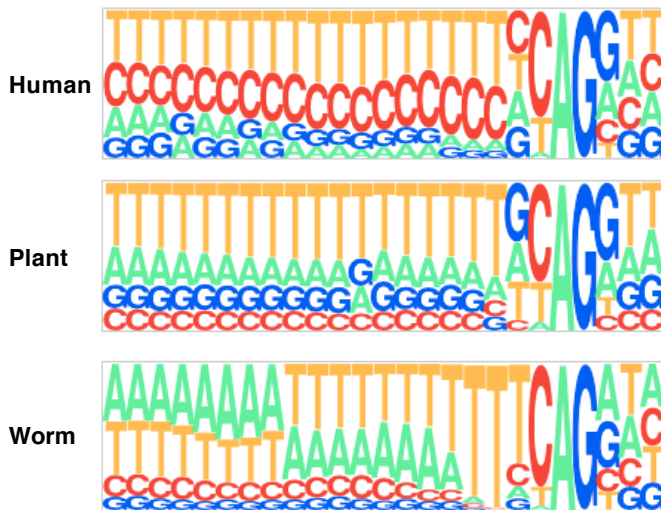
How different is the sequence content among different genomes?

To answer this questions I started by obtaining the genomic sequence and annotation for a few convenient organisms. For consistency, the *A. thaliana* and *C. elegans* sets contain DNA sequences with a single multi-exon gene just like the Burset & Guigo set.

- 590 Human genes (Burset & Guigo, 1996)
- 1375 cDNA-confirmed *Arabidopsis thaliana* genes (www.arabidopsis.org)
- 1083 cDNA-confirmed *Caenorhabditis elegans* genes (www.wormbase.org)

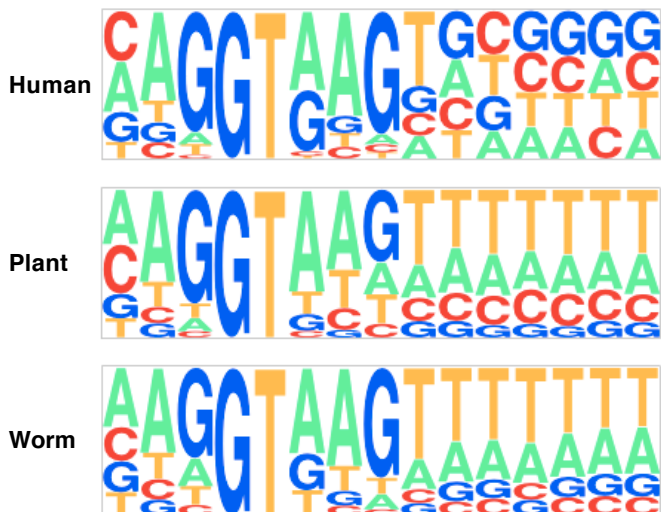
Splice acceptor sites

To compare the splice acceptor sites of the three genomes, I used the Pictogram program by Chris Burge (<http://genes.mit.edu/pictogram.html>). Qualitatively, the acceptor sites in the human, plant, and worm datasets are quite different. The YYYCAG consensus is much more highly conserved in *C. elegans* than in *A. thaliana* or in humans. In the region upstream of the consensus, human splice sites are pyrimidine-rich while *A. thaliana* are T-rich and *C. elegans* are AT-rich. This result mirrors those found in the comprehensive analysis of short introns in Lim & Burge (2001).



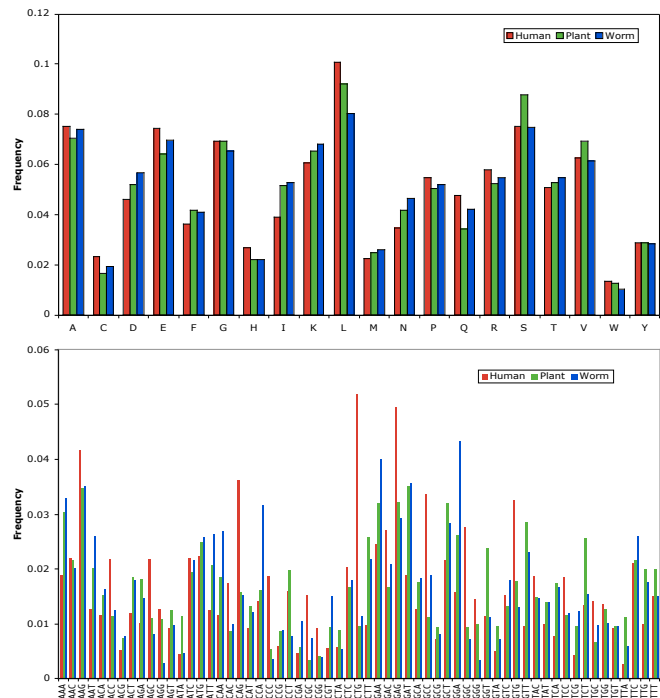
Splice donor sites

I compared the splice donor sites in the same manner as the splice acceptor sites. There is some preference for AT-richness downstream of the U1 binding site, but overall they are not very different.



Coding Sequence

To look at differences in coding sequence I plotted histograms for amino acid and codon usage. The amino acid usage in the human, plant, and worm datasets is nearly identical but their codon usage is very different.



How important is it to train sequence models for a specific genome?

In order to measure what happens when sequence models are trained on one genome and used to scan other genome, I first split the genomic data in half and used one set for training and the other set for testing. I then trained a splice acceptor model (25 bp weight matrix), a splice donor model (9 bp weight matrix), and a codon model (3 periodic, 2nd order Markov model) for each genome. These were used to score all the splice sites and open reading frames in each testing set.

The table below displays the average scores of true sites and pseudo-sites for splice acceptors, splice donors, and coding regions. The same-species scores are in colored blocks. To a first approximation, the ability of a sequence model to discriminate between true sites and pseudo-sites can be given by the difference between their average scores. Notice that the sequence models are less discriminative when the training and testing genomes are different.

Training Set	Seq Model	Testing Set					
		Human		Plant		Worm	
		True	Pseudo	True	Pseudo	True	Pseudo
Human	Accptr	+94	-41	+52	-41	+39	-46
	Donor	+83	-5	+57	-12	+6	-15
	Coding	+172	-63	-21	-97	-27	-103
Plant	Accptr	+50	-26	+98	-6	+90	-14
	Donor	+72	+9	+70	+11	+68	+8
	Coding	+12	-76	+135	-58	+149	-63
Worm	Accptr	-5	-102	+37	-72	+141	-72
	Donor	+76	+5	+64	+7	+74	+4
	Coding	-13	-97	+85	-70	+195	-58

Notes

Although my library supports many kinds of sequence models, only weight matrices and 2nd order Markov models have been tested in this way to date. It is likely that weight array matrices (an array of n th-order Markov models) and higher order Markov models should make the models more discriminative as well as make them more genome specific.