University of Alberta

Statistical models for exon and intron content, with an application to genetic structure prediction

by

Fiona Lusby ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science

in

Statistics

Department of Mathematical Sciences

Edmonton, Alberta

Fall 2000

0-612-59836-5

University of Alberta

Library Release Form

Name of Author: Fiona Lusby

Title of Thesis: Statistical models for exon and intron content, with an application to genetic structure prediction

Degree: Master of Science

Year this Degree Granted: 2000

Permission is hereby granted to the University of Alberta to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as herein before provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Fiona Lusby

8th August 2000

27 Beechgrove,
Oranmore,
Co. Galway,
Ireland.

UNIVERSITY OF ALBERTA

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Statistical models for exon and intron content, with an application to genetic structure prediction** submitted by **Fiona Lusby** in partial fulfillment of the requirements for the degree of **Master of Science** in Statistics.

Peter Hooper

Dr. P.M. Hooper (Supervisor)

Dr. N.G.N. Prasad

David Wishart

Dr. D. Wishart

Date: 4th August 2000

Abstract

In recent years a number of computer programs for eukaryotic gene identification have been developed. One such program, GRPL (Hooper *et al.*, 2000), is discussed in detail in Chapter 1. GRPL has four distinct components, namely, the state score, the length score, the functional site score, and the content score. The content score consists of a log-likelihood ratio for nucleotide content, determined by models for coding and non-coding regions.

Alternative models for content score are considered in Chapter 3. These models make use of a regression technique called reference point logistic regression (Hooper, 2000). Reference point logistic regression is a generalization of logistic regression that can be used in complex classification problems to model the conditional probability that an item belongs to a specified class given the features observed for the item; details are given in Chapter 2.

I would like to thank Dr. P.M. Hooper for suggesting this problem and for his help throughout.

I would also like to thank my fiance, Connell, and my friends for their endless support.

Table of Contents

List of Tables

List of Figures

Introduction

The cells of all organisms contain DNA sequences. A DNA sequence can be represented as a sequence from {A,C,G,T}. These DNA sequences contain information which is used to build proteins. Only small parts of the DNA sequence actually codes for protein.

The information in the DNA sequence is contained in genes. A gene consists of one or more exons, where an exon is a continuous segment of the DNA sequence. For example, a gene which is made up of two exons could be represented as follows:

$$\ldots\text{ATGAAACTGC}\underbrace{\text{ATGGTCCG}}_{\text{exon 1}}\text{GTAAAGTACGTAG}\underbrace{\text{GCTCTCGTAA}}_{\text{exon 2}}\text{GTCGATG}\ldots$$

The gene is therefore, ATGGTCCGGCTCTCGTAA.

Given a DNA sequence, we are interested in locating all exons in the sequence, if any, and in determining the formation of consecutive exons into genes. In recent years a number of computer programs have been developed which address this problem by computational methods. Such programs are known as gene prediction programs.

We are interested in eukaryotic gene prediction, where a gene consists of one or more exons, as opposed to prokaryotic gene prediction, where a gene consists of a single exon. We restrict attention to human and vertebrate gene prediction. Genetic structure varies considerably among organisms so better results will be obtained by focusing on a particular organism type.

We will discuss two computer programs for gene prediction in detail: GENSCAN, which was developed by Burge & Karlin (1997), and GRPL, which was developed by Hooper *et al.* (2000), and which employs an approach similar to that of GENSCAN.

The model used by GRPL has four distinct components, the state score, the length score, the functional site score, and the content score.

The content score attempts to distinguish between the patterns which occur in exons and the patterns which occur outside exons. It involves the estimation of a large number of parameters. Two alternative models will be considered in an attempt to improve on the high level of accuracy shown by GRPL.

We will now present a brief overview of the thesis.

Chapter 1 contains the biological background for the problem. It also outlines the various measures that are used to assess the accuracy of a gene prediction program. GENSCAN and GRPL are discussed in some detail.

The alternative models for content score use a regression technique called Reference Point Logistic (RPL) regression. It is a generalization of logistic regression which was introduced by Hooper (2000). RPL regression will be discussed in Chapter 2.

Chapter 3 contains the new models for content score. These new models are compared to GRPL, GENSCAN and various other gene prediction programs.

Chapter 1

Prediction of Genetic Structure

1.1 Biological Background and Terminology

A molecule of DNA is formed from two strands of nucleotides. A nucleotide consists of a sugar-phosphate backbone and one of four bases. Two possible types of base are present, pyrimidine bases and purine bases. The pyrimidine bases are cytosine (C) and thymine (T), the purine bases are adenine (A) and guanine (G). The two strands are parallel, forming a double helix. They are complementary in that, opposite A there is a T and opposite C there is a G. A sequence of nucleotides, that is, a strand of DNA, can be represented by a sequence from {A,C,G,T}.

A gene is a sequence of nucleotides that codes for a protein. A protein is a polymer made up of amino acids. There are twenty common amino acids, each determined by one or more nucleotide triplets called codons. There are three codons, namely TAA, TAG and TGA, which do not code for amino acids. These are called stop codons and they signal the end of a gene. The portion of a DNA sequence between genes is known as intergenic. In prokaryotic species (bacteria) each gene is a continuous segment of a DNA strand. In eukaryotic species (multicellular organisms) a gene usually consists of several coding regions called exons which are separated by noncoding regions called introns.

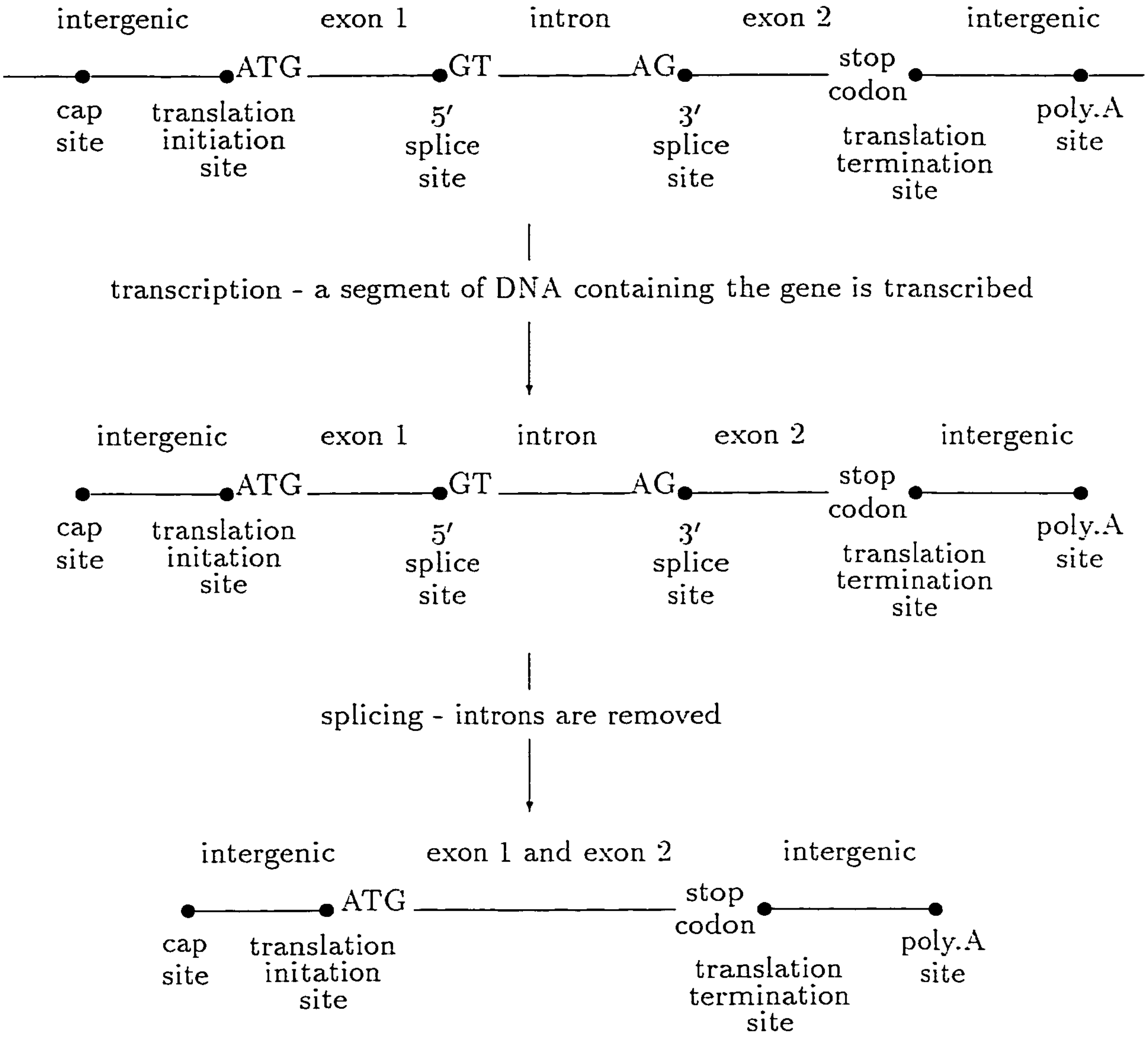
The transfer of genetic information takes place in three stages. First, a segment of DNA containing a gene is transcribed, producing a precursor of mRNA. The introns are then removed in a process called splicing, leaving a single coding region with an intergenic region at either end. This is the mRNA which is transported to the cytoplasm for translation into protein.

A number of special sites play a role in the transcription, splicing and translation. A cap site and related promoter sites signal the start of transcription. A polyadenylation site signals the end of transcription. A translation initiation site is at the start of every gene and is always followed by a methionine codon ATG. A translation termination site is at the end of every gene and is always preceeded by a stop codon. The transfer point from an exon to an intron is called a $5'$ splice site, and from an intron to an exon is called a $3'$ splice site. About 99% of introns begin with the dinucleotide GT and end with the dinucleotide AG (Senapathy *et al.*, 1990). A branch point site, located in an intron upstream from the $3'$ splice site, plays a role during splicing. A two-exon gene and its conversion to mRNA can be represented as in Figure 1.

A gene prediction program tries to identify the translation initiation site, all $5'$ and $3'$ splice sites, and the translation termination site of a gene. These sites are known as the functional sites. A complete specification of functional sites can be usefully represented in terms of a parse of a DNA sequence. A parse is a possible partition of the sequence into non overlapping intervals, with each interval assigned to one of the following states: intergenic, intron of phase 0, 1 or 2, initial exon, internal exon of phase 0, 1 or 2, single exon gene and terminal exon. The phase of an intron is 0 if the the last codon in the previous exon is complete, the phase is 1 if the last codon in the previous exon needs 2 nucleotides for completion and the phase is 2 if the last codon in the previous exon needs 1 nucleotide for completion. The phase of an internal exon is the phase of the preceeding intron. The parse must respect the permissible state transitions which are

- from intergenic to single exon gene or to initial exon
- from initial exon to intron of phase 0, 1 or 2

Figure 1: The conversion of a two-exon gene to mRNA



- from intron of phase 0, 1 or 2 to internal exon of the same phase or to terminal exon
- from single exon gene or terminal exon to intergenic

Some authors (Burge & Karlin, 1997, Synder & Stormo, 1995) include additional states in the parse, for example, the $5'$ untranslated region (extending from the start of transcription to the translation initiation site) and the $3'$ untranslated region (extending from just after the stop codon to the polyadenylation site). Given a DNA sequence a gene prediction program selects a parse based on some optimality criterion.

1.2 Measures of Predictive Accuracy

The various models on which gene prediction programs are based, all have a number of parameters which need to be estimated. Parameter estimates are obtained from a training set, that is, a collection of DNA sequences with known structure. As noted by Burset & Guigó (1996), it is important that a program be tested on a set of DNA sequences distinct from the training set, so that it can be evaluated objectively rather than giving possibly inflated results due to over-fitting of the training set.

The coding proportion of a DNA sequence is the proportion of nucleotides in the sequence that are in coding regions, that is, in exons. The coding proportion of the sequences in the training and test sets are of interest because it is likely that accuracy results will decrease as the coding proportion decreases substantially below that of the training set (Hooper *et al.*, 2000).

We employ three standard test sets for evaluating gene prediction programs, the Burset/Guigó test set (Burset & Guigó, 1996), which contains 570 vertebrate genes, with an average coding proportion of 0.21; the GeneParser test set I (Synder &

Stormo, 1995), which contains 28 human genes, with a coding proportion of 0.14; and the GeneParser test set II, which contains 34 human genes, with a coding proportion of 0.17. The predictive accuracy is calculated for each sequence in the test set, the results are then averaged over all sequences, with each sequence given equal weight.

The accuracy of the predictions is measured at two levels: the nucleotide level and the exon level. Consider first the nucleotide accuracy. For a given sequence, let TP (true positive) be the number of nucleotides correctly predicted to be in coding regions, let TN (true negative) be the number of nucleotides correctly predicted to be in noncoding regions, let FP (false positive) be the number of nucleotides incorrectly predicted to be in coding regions and let FN (false negative) be the number of nucleotides incorrectly predicted to be in noncoding regions. Then PP=TP+FP (predicted positive) is the number of nucleotides predicted to be in coding regions, PN=TN+FN (predicted negative) is the number of nucleotides predicted to be in noncoding regions, AP=TP+FN (actual positive) is the number of nucleotides in coding regions and AN=TN+FP (actual negative) is the number of nucleotides in noncoding regions.

Sensitivity is defined as Sn=TP/AP, the proportion of coding nucleotides that are correctly classified. Specificity is defined as Sp=TP/PP, the proportion of nucleotides predicted to be in coding regions that are correctly classified. We set Sq=TN/AN and Sr=TN/PN. If, for a particular sequence, any of the denominators is zero, then the corresponding numerator is also zero. The ratio is now undefined and is ignored when averaging across sequences. Two additional measures reported are the Correlation coefficient

$$CC = \frac{(TP)(TN) - (FP)(FN)}{\sqrt{(PP)(PN)(AP)(AN)}}$$

and the approximate correlation AC=-1+2(Sn+Sp+Sq+Sr)/4. Usually CC and AC are close in value but, if one of the ratios Sn, Sp, Sq, Sr has a zero denominator, then CC is undefined while AC is redefined using the average of the other three ratios.

At the exon level two accuracy measures are common, exon sensitivity and exon specificity. Let XTP (exon true positive) be the number of actual exons that exactly match predicted exons. Exon sensitivity, XSn, is XTP divided by the number of actual exons. Exon specificity, XSp, is XTP divided by the number of predicted exons.

1.3 Gene Prediction Programs

Since the early 1990's a number of computer programs designed to predict the structure of protein coding genes in genomic DNA sequences have been developed. These include GRPL and GRPL+ (Hooper *et al.*, 2000), GENSCAN (Burge & Karlin, 1997), Genie (Kulp *et al.*, 1996), Xpound (Thomas & Skolnick, 1994), GeneID (Guigó *et al.*, 1992), GRAIL2 (Xu *et al.*, 1994) and GeneParser (Synder & Stormo, 1995).

Many of these programs have an option for organism type. Genetic structure varies considerably among organisms, and prediction can be improved by using parameters specific to the organism type. Where there was such an option, human or vertebrate was chosen for the following comparisons.

Some of the programs make use of database sequence alignment methods such as BLAST or XBLAST (Altschul *et al.*, 1990). Several authors (Burset & Guigó, Synder & Stormo, 1995) have commented on the difficulty in comparing programs that use sequence alignment methods. It is conceivable that most programs will be improved by incorporating sequence alignment techniques, though the extent of the improvement may depend on the initial accuracy. The increase in accuracy will also depend on the size of the database being searched. Among the programs compared below, those using sequence alignment methods are GRPL+, GeneID+ and GeneParser3.

Burset & Guigó (1996) compared a number of the above programs using the Burset/Guigó test set described previously. Their results appear in Table 1. Those not analysed by Burset & Guigó are GRPL, GENSCAN and Genie. The results for GRPL and GRPL+ were obtained from Hooper *et al.* (2000). The results for GENSCAN are from Burge & Karlin (1997). The results for Genie appear in Kulp *et al.* (1996).

Table 1: Performance comparisons for the Burset/Guigó test set.

Program	Sn	Sp	Sq	CC	AC	XSn	XSp
GRPL(Hu)	0.93	0.93	0.984	0.91	0.91	0.76	0.79
GRPL(Hu)+	0.97	0.97	0.990	0.96	0.96	0.81	0.85
GENSCAN	0.93	0.93	n/a	0.92	0.91	0.78	0.81
Genie	0.76	0.77	n/a	n/a	0.72	0.55	0.48
GRAIL2	0.72	0.87	n/a	0.76	0.75	0.36	0.43
GeneID	0.63	0.81	n/a	0.65	0.67	0.44	0.46
Xpound	0.61	0.87	n/a	0.69	0.68	0.15	0.18
GeneID+	0.91	0.91	n/a	0.88	0.88	0.73	0.70
GeneParser3	0.86	0.91	n/a	0.85	0.86	0.56	0.58

Tables 2 and 3 contain the performance results for the GeneParser test sets. The results were obtained from Hooper *et al.* (2000), where the test sequences were submitted to the respective program's webserver, except in the case of Xpound which was run locally.

Table 2: Performance comparisons for the GeneParser I test set.

Program	Sn	Sp	Sq	CC	AC	XSn	XSp
GRPL(Hu)	0.96	0.88	0.978	0.90	0.91	0.72	0.69
GRPL(Hu)+	0.98	0.95	0.994	0.96	0.96	0.73	0.80
GENSCAN	0.97	0.88	0.982	0.91	0.91	0.74	0.73
Genie	0.86	0.81	0.964	0.80	0.80	0.68	0.64
GRAIL2	0.91	0.86	0.980	0.86	0.87	0.48	0.44
GeneID	0.80	0.84	0.979	0.79	0.79	0.60	0.51
Xpound	0.76	0.87	0.984	0.78	0.79	0.21	0.24

Table 3: Performance comparisons for the GeneParser II test set.

Program	Sn	Sp	Sq	CC	AC	XSn	XSp
GRPL(Hu)	0.89	0.93	0.989	0.89	0.90	0.69	0.77
GRPL(Hu)+	0.93	0.96	0.995	0.94	0.94	0.71	0.76
GENSCAN	0.89	0.92	0.986	0.90	0.89	0.68	0.69
Genie	0.75	0.76	0.967	0.71	0.72	0.54	0.51
GRAIL2	0.70	0.84	0.978	0.73	0.72	0.36	0.32
GeneID	0.70	0.75	0.961	0.68	0.71	0.49	0.48
Xpound	0.71	0.91	0.987	0.76	0.78	0.26	0.27

The programs with the highest accuracy on all three test sets are GRPL, GRPL+ and GENSCAN. GRPL and GENSCAN have similar results at the nucleotide level, with GENSCAN more accurate at the exon level. GRPL+, which refers to GRPL with protein sequence alignment techniques included, is more accurate at both levels.

This thesis will concentrate on a model for the content score of exons as defined in

GRPL. GRPL is based on a Generalized Hidden Markov Model, similar in structure to the model used by GENSCAN. GRPL and GENSCAN differ primarily in their models for content score and functional sites. We will continue by describing the underlying models in both of these programs.

GRPL and GENSCAN treat the general case where the input sequence may contain a partial gene, a complete gene, multiple complete (or partial) genes or no gene at all.

GRPL and GENSCAN use essentially the same training set, a set of 380 human DNA sequences compiled by Burge & Karlin (1997), containing 238 multi-exon genes and 142 single-exon genes. This set will be denoted by $\mathcal{L}$. A small number of the sequences were removed by Hooper *et al.* (2000) for the training of GRPL. A set of 1619 human cDNA sequences is combined with $\mathcal{L}$ to produce the set $\mathcal{C}$. This set was used by both Burge & Karlin (1997) and Hooper *et al.* (2000) to model content scores.

1.4 GENSCAN

GENSCAN predicts genetic structure simultaneously on both DNA strands. In the model used, the possible states for a parse are intergenic, $5'$ untranslated region, single exon gene, initial exon, intron of phase 0, 1 or 2, internal exon of phase 0, 1 or 2, terminal exon and $3'$ untranslated region. All of these can occur on either the forward or the reverse strand. A parse ϕ can be represented by an ordered set of states $\{q_1,q_2,...,q_n\}$ with an associated set of lengths $\{d_1,d_2,...,d_n\}$. The meeting point between consecutive states is called a site. An illustration of a parse is given in Figure 2.

For a fixed sequence length L, let Φ_L be the set of all possible parses of length

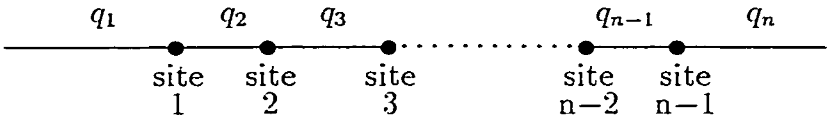


Figure 2: A parse ϕ

L and let $\mathcal{S}_L$ be the set of all possible DNA sequences of length L. Given L, GENSCAN assigns a joint probability to each parse/sequence pair, that is, to each (ϕ, S) $\in \Phi_L \times \mathcal{S}_L$. Thus, for a given DNA sequence S of length L, the conditional probability of a particular parse $\phi \in \Phi_L$ is calculated using Bayes rule,

$$P(\phi|S) = \frac{P(\phi, S)}{\sum_{\psi \in \Phi_L} P(\psi, S)}.$$

Given a DNA sequence S, let z be a sequence of proportions obtained from S as follows: The ith element of z records the combined proportion of cytosine (C) and guanine (G) in the general vicinity of the ith nucleotide in S, that is, in a window of up to 10,000 nucleotides on either side of the ith nucleotide in S. Shorter windows are used near the ends of sequences and for shorter sequences. z is called the sequence of C+G content of S. Gene density and certain aspects of gene structure are known to vary quite dramatically in regions of differing C+G content. Burge & Karlin (1997) attempt to compensate for this by using parameters conditioned on a C+G grouping. They use four groups of C+G proportions with boundaries 0.43, 0.51 and 0.57. Since z is a function of S, we can write

$$P(\phi|S) = \frac{P(\phi, S)}{\sum_{\psi \in \Phi_L} P(\psi, S)} = \frac{P(\phi, S, z)}{\sum_{\psi \in \Phi_L} P(\psi, S, z)} = \frac{P(\phi, S|z)}{\sum_{\psi \in \Phi_L} P(\psi, S|z)}.$$

Given a sequence S of length L, an optimal parse is defined to be a parse ϕ, maximizing the conditional probability $P(\phi|S)$. To find an optimal parse it is thus sufficient to find a parse maximizing the joint conditional probability $P(\phi, S|z)$.The joint conditional probability can be rewritten as

$$P(\phi, S|z) = P(\phi|z)P(S|\phi, z).$$

The conditional probability, $P(\phi|z)$, is modeled using an explicit state duration Hidden Markov Model. It has three main components, a vector of initial probabilities π for the state of the first interval, a matrix of state transition probabilities T for the state of the following intervals, a set of state length (state duration) distributions f. The conditional probability can thus be written as

$$P(\phi|z) = \pi_{q_1|z} f_{q_1}(d_1|z) \times \prod_{k=2}^{n} T_{(q_{k-1},q_k|z)} f_{q_k}(d_k|z)$$

where the states of ϕ are $q_1, q_2, ..., q_n$ with associated state lengths $d_1, d_2, ..., d_n$.

Separate initial and transition probabilities are estimated for the four C+G categories. The initial probability of each state is chosen to be proportional to its estimated frequency in bulk genomic DNA. Since exon lengths are generally small compared with intron and intergenic lengths, exons are assigned zero initial probability.

Empirically derived length distribution functions are estimated for initial, internal and terminal exons and for single exon genes. Exon length distributions are considered independent of the C+G proportion. Intron and intergenic length distributions are modeled as geometric with the parameter p dependent on the C+G group. The length distributions for the $5'$ and $3'$ untranslated regions are also modeled as geometric but with the mean independent of z.

The model for the remaining term, $P(S|\phi, z)$, is based upon a partition of the sequence S as follows: for each site i in ϕ, a subsequence w_i of S is formed consisting of nucleotides in the immediate vicinity of the site, the number of nucleotides depending on the states at either side of the site. The w_i are called signal sequences. By removing the signal sequences from S there is a natural set of subsequences $\{v_1,v_2,....v_n\}$ remaining, so that S is simply the concatenation of $v_1,w_1,v_2,w_2,.....v_{n-1},w_{n-1},v_n$, where n is the number of states in the parse ϕ. The v_i are called the content sequences.

All signal and content sequences are assumed conditionally independent given ϕ and z, therefore

$$P(S|\phi, z) = P(v_1|\phi, z)P(w_1|\phi, z).........P(v_{n-1}|\phi, z)P(w_{n-1}|\phi, z)P(v_n|\phi, z).$$

The conditional distributions of the signal sequences given ϕ and z are modeled as dependent only on the site type which is determined by the states either side of the site. The conditional distributions of the content sequences given ϕ and z are modeled as dependent only on the state of the interval in which they occur and the C+G proportion. The conditional distribution of S given ϕ and z can thus be written as

$$P(S|\phi, z) = P(v_1|q_1, z)P(w_1|q_1, q_2)......P(v_{n-1}|q_{n-1}, z)P(w_{n-1}|q_{n-1}, q_n)P(v_n|q_n, z).$$

Burge & Karlin use three models for the conditional distribution of the signal sequences given the site type, the weight matrix method (WMM), the weight array method (WAM), and the maximal dependence decomposition method (MDD). The WMM is used for polyadenylation signals and translation initiation signals, it is also used for translation termination and cap site signals, but with some adjustments. A modified WAM model is used for $3'$ splice signals. A MDD model is used for $5'$ splice signals.

The weight matrix method was introduced by Staden (1984). It derives the frequency of each nucleotide j at each position i of a signal of length r from a collection of aligned signal sequences. The nucleotides at each position are considered independent. The weight array method, a generalization of the weight matrix method, was applied by Zhang & Marr (1993). It allows for dependencies between adjacent positions. The maximal dependence decomposition method was introduced by Burge & Karlin (1997). It generates a model from an aligned set of signal sequences, which captures the most significant dependencies between positions (allowing for non-adjacent as well as adjacent dependencies) by replacing WMM probabilities by appropriate conditional probabilites provided sufficient data is available to do so reliably.

The content sequences occuring in coding regions are modeled using an inhomogeneous 3-periodic fifth-order Markov model. Separate fifth-order transition matrices are determined for hexamers ending at each of the three codon positions, in each of two C+G compositional groups with boundary 0.43. The content sequences occuring in non-coding regions are modeled using two fifth-order Markov models, again one for each of the two C+G compositional groups mentioned above.

1.5 GRPL

In the model used by GRPL, the possible states for a parse are intergenic, single exon gene, initial exon, intron of phase 0, 1 or 2, internal exon of phase 0, 1 or 2 and terminal exon. Unlike GENSCAN, GRPL does not incorporate the $5'$ untranslated region and the $3'$ untranslated region directly in the parse.

The program has two stages. First, all consensus sites are investigated. A consensus site is a site exhibiting the common nucleotide patterns observed at either a translation initiation site, a translation termination site, a $5'$ splice site or a $3'$ splice site. RPL regression models are used to estimate the conditional probability that a consensus site is a functional site given certain features observed for the site. A consensus site that is not a functional site is called a pseudo site. Different RPL models and feature vectors are used for each of the four site types. The feature vectors are mostly indicators of nucleotide usage in the immediate vicinity of the consensus site. For example, features for the translation initiation site include indicators of nucleotide usage at positions -1 to -7 (preceeding the site), a statistic measuring evidence of a cap site some distance upstream, and content statistics measuring evidence of a coding region immediately downstream.

In each RPL model equal prior probabilities for functional sites and pseudo sites are used. Pseudo sites actually outnumber functional sites by more than 100 to 1 but only the likelihood ratio

$$\frac{f(x|\text{functional})}{f(x|\text{pseudo})}$$

is used in GRPL, and the likelihood is unaffected by the choice of prior probabilities. A consensus site is classified as a pseudo site if the estimated log likelihood ratio is less than a specified value. A cutoff value of -3 was chosen in order to screen out the majority of pseudo sites while keeping almost all functional sites.

The second stage of GRPL is concerned with the set of all possible parses formed from the collection of admitted consensus sites. A score is assigned to each parse via a Generalized Hidden Markov Model, and a dynamic programming algorithm is used to find the parse with the highest score. Let x be the vector of observed nucleotides and let y be a vector consisting of functional site locations and types representing a possible parse for x. Let z be the vector of C+G content of the sequence x. The score for a parse y is defined to be an estimate of

$$\log\frac{P(y|x)}{P(y_0|x)},$$

where y_0 is the null parse consisting of a single intergenic region. Using the fact that z is a function of x, this can be rewritten as

$$\begin{aligned}
\log\frac{P(y|x)}{P(y_0|x)} &= \log\frac{P(y|x,z)}{P(y_0|x,z)} \\
&= \log\frac{P(y,x|z)/P(x|z)}{P(y_0,x|z)/P(x|z)} \\
&= \log\frac{P(y,x|z)}{P(y_0,x|z)} \\
&= \log\frac{P(y|z)}{P(y_0|z)}\frac{P(x|y,z)}{P(x|y_0,z)} \\
&= \log\frac{P(y|z)}{P(y_0|z)} + \log\frac{P(x|y,z)}{P(x|y_0,z)}.
\end{aligned}$$

Consider the first term in the final sum. The conditional probability, $P(y_0|z)$, can be ignored when searching for an optimal parse. The conditional probability, $P(y|z)$, is evaluated in a manner similar to that in GENSCAN using a vector of initial probabilities π for the state of the first interval, a matrix of state transition probabilities T for the state of the following intervals, and a set of state length distributions f, all conditioned on the C+G content.

The remaining term

$$\log\frac{P(x|y,z)}{P(x|y_0,z)},$$

can be rewritten by expressing x in terms of two vectors v and w, which depend on the parse y, as follows: The vector v consists of all of x except nucleotides in the immediate vicinity of functional sites. The vector w consists of features extracted from x for each functional site. The features extracted depend on the type of site but always include indicators of nucleotide usage close to the site. This decomposition of x can be illustrated as follows

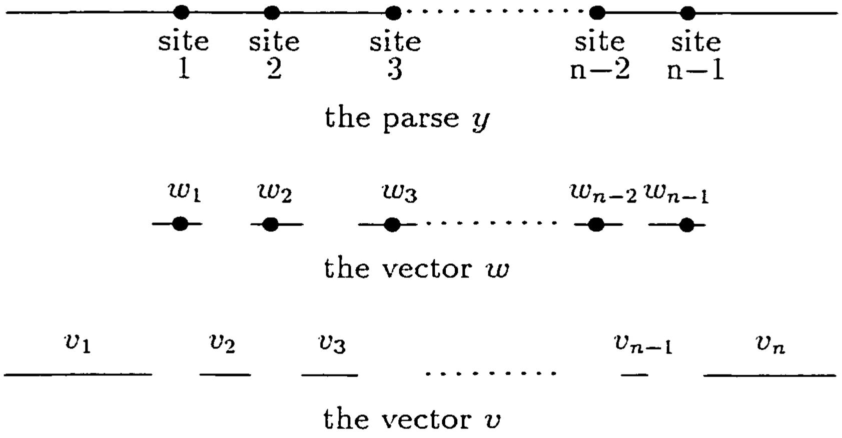


Figure 3: The partition of x into the vectors v and w.

There is some overlap between v and w but this is ignored and they are considered conditionally independent given y. Hence

$$\begin{aligned} \log\frac{P(x|y,z)}{P(x|y_0,z)} &= \log\frac{P(v,w|y,z)}{P(v,w|y_0,z)} \\ &\approx \log\frac{P(v|y,z)}{P(v|y_0,z)} + \log\frac{P(w|y)}{P(w|y_0)}. \end{aligned}$$

The distribution of w is assumed independent of z. The first term in the sum is called the content score, the second is called the functional site score.

1.5.1 Functional site scores

Feature vectors for different functional sites are assumed to be independent, so the functional site score is a sum of scores, one for each site. Let w_i be the feature vector for the i^{th} functional site in the parse y. The i^{th} functional site score is then

$$\log\frac{P(w_i|y)}{P(w_i|y_0)} = \log\frac{P(w_i|\text{ the site is a functional site})}{P(w_i|\text{ the site is a pseudo site})}.$$

As in the preliminary screening, RPL models are used to estimate this likelihood ratio with the following expression

$$\log\frac{\hat{P}(\text{functional site }|w_i)}{\hat{P}(\text{pseudo site }|w_i)},$$

where $\hat{P}$ is an estimate of the conditional distribution of the i^{th} site given the observed feature vector, obtained from RPL models using equal prior probabilities for functional sites and pseudo sites.

It was found that direct use of these estimates resulted in low sensitivity relative to specificity. To improve accuracy the log likelihood ratio was adjusted as follows

$$\text{Functional site score} = 2.8 + 1.2\log\frac{\hat{P}(\text{functional site }|w_i)}{\hat{P}(\text{pseudo site }|w_i)}.$$

These values were selected by tuning GRPL on the training set $\mathcal{L}$.

1.5.2 Content score

The content score is defined to be an estimate of

$$\log\frac{P(v|y,z)}{P(v|y_0,z)}.$$

Conditional independence of content for different intervals is assumed, so the ratio $P(v|y,z)/P(v|y_0,z)$, can be expressed as a product of terms, one for each interval.

The content score for an interval is the logarithm of the likelihood ratio for that interval. The content score for introns and intergenic regions is zero.

Let $h = (b_1, b_2, b_3, b_4, b_5, b_6)$ denote a hexamer observed at some point in the sequence. Let $t \in \{0, 1, 2, 3\}$ be a value, called the hexamer phase, determined by a sequence parse and the position of b_1 in the sequence defined as follows: If b_1 is in a coding region determined by the parse then let $t + 1$ be the position of b_1 in the codon containing b_1. If b_1 is a non-coding region then t=3.

Under this model the conditional distribution of b_1, given the sequence parse and the sequence downstream from b_1, depends only on the hexamer phase t and the following five nucleotides. Similarily, the conditional distribution of b_6 given the sequence parse and the sequence upstream from b_6, depends only on the hexamer phase t and the previous five nucleotides. Let

$$s(t, h, z) = \frac{1}{2}\log\frac{P_1(b_1|b_2...b_6, t, z)}{P_1(b_1|b_2...b_6, 3, z)} + \frac{1}{2}\log\frac{P_6(b_6|b_1...b_5, t, z)}{P_6(b_6|b_1...b_5, 3, z)}$$

where P_1 and P_6 denote the upstream and downstream conditional probability functions, and $t \in \{0, 1, 2\}$.

The content score for an exon is then defined to be

$$\text{content score} = \sum s(t, h, z),$$

where the summation is over all hexamers h covering most of the exon.

The functions $s(t, h, z)$ are modeled as linear functions of z, with coefficients depending on t and h, as follows

$$s(t, h, z) = \beta_0(t, h) + \beta_1(t, h)(z - 0.5).$$

Estimates of the coefficients are obtained from the training set $\mathcal{C}$. Let g_z be a grouping variable for C+G content z, having 6 groups with boundaries 0.4, 0.45, 0.5, 0.55 and 0.6. The conditional probabilities $P_1(b_1|b_2,...b_6,t,g_z)$ and $P_6(b_6|b_1,...b_5,t,g_z)$, are first estimated from the training set for each $t \in \{0,1,2,3\}$, for each hexamer $h = (b_1,b_2,b_3,b_4,b_5,b_6)$ and for each g_z.

Consider, for example, estimating $p = P_1(b_1|b_2,...b_6,t,g_z)$. Let m be the number of occurrences of the hexamer $(b_1,b_2,b_3,b_4,b_5,b_6)$ with hexamer phase t and C+G proportion in the group g_z. Let n be the number of occurrences of hexamers of the form $(*,b_2,b_3,b_4,b_5,b_6)$ with hexamer phase t and C+G proportion in the group g_z, where $*$ can be any base.

The conditional distribution of m given n is Binomial(n,p) and the maximum likelihood estimator for p is m/n. The MLE is undefined when n is zero and highly variable when n is small so, a Bayes estimator, which is biased but has smaller variance, is prefered. Let $\tilde{p}$ be the MLE for the conditional probability $P(b_1|b_2,b_3,t,g_z)$. The Bayes estimator used by GRPL, for p is

$$\hat{p} = \frac{m + n_0\tilde{p}}{n + n_0}.$$

A value of n_0=10 was chosen.

In this way, estimates of $s(t,h,z)$ are obtained for each hexamer h, for each $t \in \{0,1,2\}$ and for each $z \in \{.355,.425,.474,.525,.575,.635\}$, representatives of the 6 groups defined by g_z. For each (t,h), the estimates of $s(t,h,z)$ are regressed on z using weighted least squares to estimate the coefficients $\beta_0(t,h)$ and $\beta_1(t,h)$. The weights are inverse variance estimates for the $s(t,h,z)$ estimates. The weighted least squares slope estimates were multiplied by 0.7, shrinking the slopes towards zero. In all 3×4^6 pairs of coefficients are estimated for the content scores.

1.6 A Comparison Between GENSCAN and GRPL

GENSCAN and GRPL are similar in their overall architecture while differing in a number of significant ways. Some of these differences are

- GENSCAN searches for genes simultaneously on both DNA strands while GRPL analyses each strand separately.

- GRPL uses a reduced set of consensus sites in constructing possible parses, but this is in order to increase the speed of the program and does not alter the overall model.

- GENSCAN includes cap sites and related promoter sites in the possible site types in a parse and attempts to model them flexibly. GRPL incorporates this information as features for translation initiation and translation termination sites.

- Both condition on C+G content, GENSCAN by grouping sequences into C+G "rich" and "poor" categories, and GRPL by using a continuous model of C+G content.

- Each program defines an optimal parse in terms of the conditional probability of a parse given the sequence. However, GRPL incorporates the conditional probability of the null parse. This allows GRPL to easily use RPL models for the likelihood ratio of a feature vector given that the site is a functional site or a pseudo site, without having to worry about prior probabilities. GENSCAN uses a variety of models for the feature vectors, depending on the site type. The MDD model used for $5'$ splice sites is closest to the RPL models used in GRPL, in that it attempts to capture the most significant dependencies between positions.

- In GENSCAN the content score is a sum of log likelihoods of content, one for each interval. In GRPL, because a log likelihood ratio is used, only the content of exons is considered. The content score for introns and intergenic regions is zero.

Chapter 2

Reference Point Logistic Regression

Reference point logistic (RPL) regression, introduced by Hooper (2000), is a generalization of logistic regression which is highly flexible. It is closely related to a method, also developed by Hooper (1999), for constructing classification rules.

Consider the following problem. An item belongs to one of J possible classes. The particular class is unknown, but a number of features associated with the item can be measured. Let Y be the unknown class and let X be the d-dimensional vector of features measured. We are interested in estimating the conditional distribution of Y given X.

RPL regression models $p(y|x)$ in terms of the proximity between x and reference points for each class. The parameters for the model have a two-level hierarchial structure. The first-level parameters (including the number of reference points) are fixed, then optimal second-level parameters are chosen. For this complete set of parameters the risk is estimated. Based on these risk estimates first-level parameters are chosen empirically.

Given the first-level parameters which are, a smoothing parameter λ, which is non-negative, the total number of reference points K, and the number of reference points K_j assigned to the class j, where

$$K = K_1 + K_2 + \ldots.. + K_J,$$

the model is further parameterized by a positive scale parameter τ, vectors $\xi_k \in \mathbb{R}^d$ and scalars $\gamma_k \in \mathbb{R}$, $k = 1, 2, ..., K$. The pair (ξ_k, γ_k) are associated with the class cls(k) given by the class assignment function

$$\text{cls}(k) = j, \; if \; K_1 + ... + K_{j-1} < k \leq K_1 + ... + K_j.$$

Assume the first-level RPL parameters are fixed. Reference point logistic functions w_k are defined as follows

$$w_k(x) = \frac{\exp(\gamma_k - \tau^{-2} \parallel x - \xi_k \parallel^2)}{\sum_{m=1}^{K} \exp(\gamma_m - \tau^{-2} \parallel x - \xi_m \parallel^2)}. \tag{1}$$

The RPL model assumes that

$$p(y|x) = \sum_{k=1}^{K} I(\text{cls}(k) = y) w_k(x).$$

Some algebraic manipulation of the reference point logistic functions gives

$$w_k(x) = \frac{\exp(\alpha_k + \beta_k^T x)}{\sum \exp(\alpha_m + \beta_m^T x)} \tag{2}$$

where

$$\alpha_k = \gamma_k - \tau^{-2} \parallel \xi_k \parallel^2 - (\gamma_K - \tau^{-2} \parallel \xi_K \parallel^2)$$

and

$$\beta_k = 2\tau^{-2}(\xi_k - \xi_K).$$

So, with one reference point per class, RPL regression is simply logistic regression. Note that $\alpha_K = 0$ and $\beta_K = \mathbf{0}$ so in the reference point parameterization (1) the scale parameter τ and one pair (ξ_k, γ_k) are redundant. The redundancies are not removed however, because the over-parameterized model simplifies the selection of initial parameter estimates in the likelihood maximization.

We define the loss incurred by the model given that $(X, Y) = (x, y)$ as

$$L(x, y) = -\log p(y|x).$$

The goal is to find parameters that minimize the risk $E_P L(X, Y)$, where P is the joint distribution of (X, Y). To this end, an estimate $\hat{P}$ of P is obtained from a training set, $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$, that is, a set of items for which the feature vector and class membership are known, and second-level parameters are chosen to minimize the training risk $E_{\hat{P}} L(X, Y)$.

Let $\hat{p}(y)$ be an estimate of the prior probability of the class y. If the training set is a simple random sample from the population, then $\hat{p}(y)$ is the proportion of the class y in the sample. If not, the prior probabilities have to be estimated in some other way. Given the estimates $\hat{p}(y)$, let U be a random variable taking values in $\{1, 2, ..., n\}$, so that (i) $Pr\{y_U = j\} = \hat{p}(j)$ and (ii) the conditional probability of U given $y_U = j$ is uniform on $\{i \,|\, y_i = j\}$. Let Z be a d-dimensional vector of independent standard normal random variables, with Z and U independent. The estimate $\hat{P}$ is defined to be the distribution of $(x_U + \lambda Z, y_U)$, where λ is the first-level smoothing parameter.

Initial parameter values ξ_k^0 are selected by using the K-means clustering criterion (MacQueen 1967). Initially the γ_k^0 are set to zero. The scale parameter τ controls the smoothness of the functions w_k, it is set to $\tau = s_{nn}$, where s_{nn} is the average distance between nearest neighbours among the K initial reference points ξ_k^0.

Stochastic approximation is used to minimize the training risk over (ξ_k, γ_k), it uses the following iterative step. An observation (x, y) is selected from $\hat{P}$ and the gradient of the loss incurred by the model at parameter values ξ_k^r and γ_k^r calculated. New parameters values ξ_k^{r+1} and γ_k^{r+1} are then selected by moving ξ_k^r and γ_k^r in the directions of maximal decrease. The step sizes are initially large but approach zero as the number of iterations increase.

Given the reference point numbers K_j and the smoothing parameter λ, we obtain a model estimate. The actual risk of the model can be estimated using a test set or cross validation. The reference point numbers and the smoothing parameter are selected empirically based on these risk estimates.

The RPL regression model has two invariance properties. First, RPL regression is invariant under affine transformation, provided the unsmoothed training risk is used (i.e. $\lambda = 0$) and one is able to minimize this training risk. This invariance property is invoked to justify standardizing the variance of the features, which helps in selecting good starting values for the reference points.

The second invariance property concerns specification of prior probabilities. Given conditional densities $f(x|y)$ and two sets of prior probabilities $p^*(y)$ and $p^\dagger(y)$, the respective conditional probability models $p^*(y|x)$ and $p^\dagger(y|x)$, obtained from Bayes theorem, are related as follows

$$p^\dagger(y|x) = \frac{p^\dagger(y)p^*(y|x)/p^*(y)}{\sum_j p^\dagger(j)p^*(j|x)/p^*(j)} \tag{3}$$

If $p^*(y|x)$ is an RPL model with $\gamma_k = \gamma_k^*$, then $p^\dagger(y|x)$ can also be expressed as an RPL model with $\gamma_k^\dagger = \gamma_k^* + \log\{p^\dagger(y)/p^*(y)\}$. The other parameters ξ_k and τ are the same in both models.

Consider estimates $\hat{p}^*(y|x)$ and $\hat{p}^\dagger(y|x)$ obtained from the same training set but based on different specified priors $p^*(y)$ and $p^\dagger(y)$. Such estimates are typically not related as in equation (3). However, a consistency argument shows that (3) is approximated by estimates when the sample size is large and an RPL model holds.

Numerical problems can arise when highly unbalanced prior probabilities are used in an RPL model. Parameter estimates may diverge during training, causing $p(y|x)$ to be underestimated for classes y assigned small prior probability. It can be useful to use more balanced priors in the RPL training, and then adjust the estimates obtained using equation (3).

Chapter 3

Content Score

3.1 Introduction

Let x denote a DNA sequence, and y a parse of x, consisting of functional site locations and types. We denote by v the vector consisting of all of x except nucleotides in the immediate vicinity of functional sites.

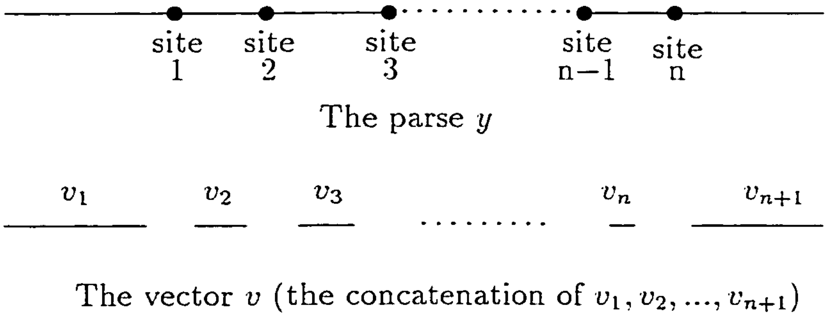


Figure 4: The vector v given the parse y

The number of nucleotides on either side of the functional site that are removed in the formation of v, depend on the type of site and are as shown in Figure 5.

Recall that y_0 denotes the null parse consisting of a single intergenic region, and that z denotes the vector of C+G content of the sequence x, that is, the ith element

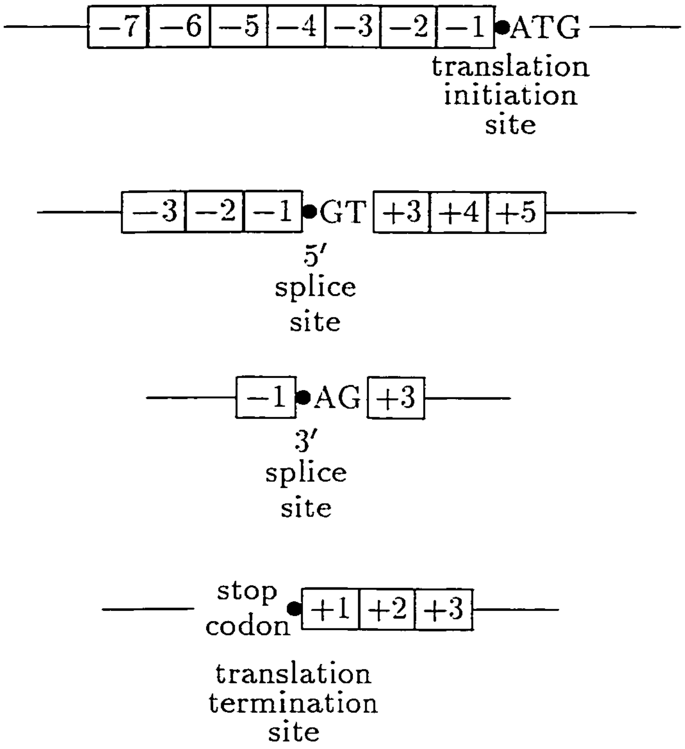


Figure 5: The nucleotides that are removed are in boxes

of z is the combined proportion of C and G in the general vicinity of the ith element of x.

In GRPL, the content score is defined as follows

$$\begin{aligned}\text{content score} &= \log\frac{P(v|y,z)}{P(v|y_0,z)} \\ &= \sum_{i=1}^{n+1}\log\frac{P(v_i|y,z)}{P(v_i|y_0,z)},\end{aligned}$$

where n is the number of functional sites in the parse y.

The content score for introns and intergenic regions is zero. The content score can thus be found by summing the content score of all the exons in the parse.

The model used by GRPL for the content score of an exon, involves the estimation of a large number of parameters (6×4^6). Two alternative models are now investigated in an attempt to improve on the high level of accuracy demonstrated by GRPL. Both models use RPL regression and use the training set $\mathcal{L}$ for parameter estimation.

3.2 Model 1

Let $v_i = (b_1, b_2, \ldots\ldots, b_m)$ represent the content of an exon with up to three nucleotides removed from either end. Then

$$\text{content score}\,(v_i) = \log \frac{P(b_1, b_2, \ldots\ldots, b_m | y, z)}{P(b_1, b_2, \ldots\ldots, b_m | y_0, z)}.$$

We define the phase t of a nucleotide b observed at some point in the sequence in a manner similar to hexamer phase. Let $t \in \{0, 1, 2, 3\}$ be a value, determined by y and the position of b in the sequence, defined as follows: If b is in a coding region determined by y then let $t+1$ be the position of b in the codon containing b. If b is in a non-coding region then set $t = 3$.

We model the conditional distribution of a nucleotide b, given the sequence parse, the sequence z of C+G content, and the sequence downstream from b, as dependent only on the phase t, the C+G proportion in the vicinity of b, and the n_t nucleotides immediately downstream, where n_t is to be determined. Similarly, the conditional distribution of a nucleotide b, given the sequence parse, the sequence z of C+G content, and the sequence upstream from b, depends only on the phase t, the C+G proportion in the vicinity of b, and the r_t nucleotides immediately upstream, where r_t is to be determined.

We work with symmetrized log likelihood ratios

$$s(t,b,z) = \frac{1}{2}\log\frac{P(b|b_{+1},b_{+2},....b_{+n_t},t,z)}{P(b|b_{+1},b_{+2},....b_{+n_3},3,z)} + \frac{1}{2}\log\frac{P(b|b_{-1},b_{-2},....b_{-r_t},t,z)}{P(b|b_{-1},b_{-2},....b_{-r_3},3,z)},$$

where b_{+i} refers to the i^{th} nucleotide following b, and b_{-i} refers to the i^{th} nucleotide preceeding b.

Hence

$$\text{content score}\,(v_i) = \sum_{i=1}^{m} s(t,b_i,z).$$

This model has therefore the same basic structure as the model used by GRPL, it differs primarily in the way in which the function s is estimated. We use eight RPL regression models to estimate $s(t,b,z)$, two for each $t \in \{0,1,2,3\}$. Given t, we need to estimate the conditional distribution of b given z and the upstream nucleotides, and the conditional distribution of b given z and the downstream nucleotides. We use a different RPL regression model for each.

Consider, for example, position one of a codon, that is, $t = 0$. We want to estimate the conditional distribution of the nucleotides A, C, G, and T, given the C+G proportion and the r_0 nucleotides immediately upstream. We use RPL regression to estimate $P(\text{class}\,|\,\text{feature vector})$, where the class is either A, C, G, or T, and the feature vector consists of the C+G proportion and the r_0 upstream nucleotides.

How the upstream nucleotides are included in the feature vector is important. Individual nucleotides may not be as informative as pairs of nucleotides, which may not in turn be as informative as triples of nucleotides and so on. Suppose, for example, $r_0 = 5$ and the 5 upstream nucleotides are ATCAG. If we consider nucleotides individually, the feature vector can represented by $(z, \mathrm{A}, \mathrm{T}, \mathrm{C}, \mathrm{A}, \mathrm{G})$, whereas if we use pairs of nucleotides, the feature vector can represented by $(z, \mathrm{AT}, \mathrm{TC}, \mathrm{CA}, \mathrm{AG})$, and if we use triples of nucleotides, the feature vector can represented by (z, ATC, TCA, CAG). We would like to use the grouping system, that is, individuals, pairs, triples, etc. which will best estimate P(class | feature vector).

The feature vector is a vector of numbers so indicator variables are used to represent nucleotide singletons, pairs, triples, etc. For example, if the nucleotides are included individually, 4 indicator variables are used for each nucleotide. A is represented by $(1,0,0,0)$, C is represented by $(0,1,0,0)$, G is represented by $(0,0,1,0)$, and T is represented by $(0,0,0,1)$. If the nucleotides are included in pairs then 16 indicators variables are used to represent each pair, for example, AA would be represented by $(1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0)$. If the nucleotides are included in triples, then 64 indicators variables are used to represent each triple, in which case AAA would have the first indicator equal to 1 and the rest equal to 0.

Grouping the nucleotides in groups of four or more will lead to a model with a large number of parameters. In order to keep the number of parameters low we will only consider grouping nucleotides in groups of three or less.

We choose the grouping system as well as n_t and r_t to minimize risk, where risk is defined to be $E_D\{-\log P(\text{class} \mid \text{feature vector})\}$, and D is the joint distribution of (class, feature vector).

The values n_t, r_t, and the grouping system can be considered additional first-level parameters in the RPL regression model. Consider again the conditional distribution of nucleotides in position one of a codon ($t = 0$), given the C+G proportion and the upstream nucleotides. A training set is randomly selected from $\mathcal{L}$. Each item in the training set consists of the nucleotide in position one as well as the feature vector consisting of the C+G proportion and the upstream nucleotides. A test set is also selected from $\mathcal{L}$

All first-level RPL parameters are fixed, that is, the grouping system, n_0, r_0, the number of reference points, and the smoothing parameter are fixed. The second-level parameters in the RPL model are chosen to minimize the risk over the training set. The first-level parameters can then be selected empirically by using the test set to estimate the risk.

This is the usual way of fitting an RPL model when we are interested in estimating conditional distributions. In this case, however, we are estimating the ratios

$$\frac{P(b|b_{-1}, b_{-2}, \dots b_{-r_0}, 0, z)}{P(b|b_{-1}, b_{-2}, \dots b_{-r_3}, 3, z)} \text{ and } \frac{P(b|b_{+1}, b_{+2}, \dots b_{+n_0}, 0, z)}{P(b|b_{+1}, b_{+2}, \dots b_{+n_3}, 3, z)},$$

and so we will adopt a slightly different approach. If b has phase 0 we want the above ratios to be large, this can be achieved not only by having a large numerator but also by having a small denominator. Whereas if b has phase 3 we want the ratios to be small, which can be achieved by a small numerator as well as a large denominator. This amounts to a model for phase $t \in \{0, 1, 2\}$ which has low risk on a test set with phase t but high risk on a test set with phase 3, and a model for phase 3 which has low risk on a phase 3 test set and high risk on a coding test set. An empirical attempt was made to choose such models, that is, to choose the first-level RPL parameters which appear to give the above results.

In each RPL model, twelve reference points are selected as optimal, three for each of the classes A, C, G, and T, and the smoothing parameter is set to zero. The number of upstream and downstream nucleotides used for each $t \in \{0, 1, 2, 3\}$ are given in Table 4.

In all of the above models the nucleotides are included individually in the feature vector except in the case of position three where the upstream nucleotides are included in pairs.

We assumed equal prior probabilities for A, C, G, and T. All nucleotides triples code for amino acids except the stop codons. Two or more triples can code for the same amino acid and the amino acids occur with varying frequencies, hence equal

Table 4: The number of upstream and downstream nucleotides used in the feature vectors of the RPL models.

Phase	upstream	downstream
$t = 0$	8	5
$t = 1$	6	7
$t = 2$	2	7
$t = 3$	8	8

prior probabilities for the coding regions do not seem justified. The prior probabilities of A, C, G, and T, for the three codon positions, are estimated from the larger training set $\mathcal{C}$ and are as in Figure 6. This set contains 1,061,147 occurrences of each codon position.

The estimates of the conditional distribution of nucleotides, in positions one, two, and three of a codon, obtained using RPL regression with equal prior probabilities, are modified using equation (3) to reflect the true prior probabilities.

To evaluate the performance of the RPL regression models, eight test sets are selected from $\mathcal{L}$. For each $t \in \{0, 1, 2, 3\}$, two test sets are selected from all nucleotides with phase t in $\mathcal{L}$, one with the feature vector containing the upstream nucleotides and the second with the feature vector containing the downstream nucleotides. Each test set contains approximately 10,000 entries. Each item in the test set has the form (class, feature vector). We use the RPL models to estimate $P(\text{class} \mid \text{feature vector})$, we then adjust these estimates for the true priors to obtain estimates $\hat{P}(\text{class} \mid \text{feature vector})$. An estimate of the risk is then an average, over all items in the test set, of $-\log\hat{P}(\text{class} \mid \text{feature vector})$. Risk estimates for the eight models are shown in Figure 7.

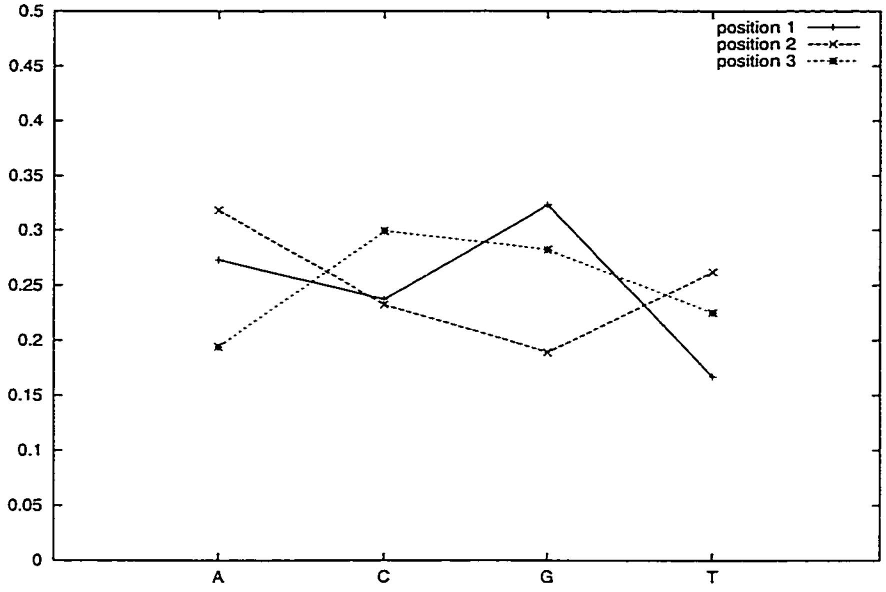


Figure 6: Prior probabilities obtained from $\mathcal{C}$

We are interested in the ability of the model to differentiate between an exon and a non-coding region. That is, if b is in an exon, with phase $t \in \{0, 1, 2\}$, we would like

$$\log \frac{P(b|b_{-1}, b_{-2}, ..., b_{-r_t}, t, z)}{P(b|b_{-1}, b_{-2}, ..., b_{-r_3}, 3, z)} > 0$$

and

$$\log \frac{P(b|b_{+1}, b_{+2}, ..., b_{+n_t}, t, z)}{P(b|b_{+1}, b_{+2}, ..., b_{+n_3}, 3, z)} > 0.$$

Whereas if b has phase 3 we would like

$$\log \frac{P(b|b_{-1}, b_{-2}, ..., b_{-r_t}, t, z)}{P(b|b_{-1}, b_{-2}, ..., b_{-r_3}, 3, z)} < 0$$

and

$$\log \frac{P(b|b_{+1}, b_{+2}, ..., b_{+n_t}, t, z)}{P(b|b_{+1}, b_{+2}, ..., b_{+n_3}, 3, z)} < 0.$$

Consider position one of a codon and the feature vector containing the upstream

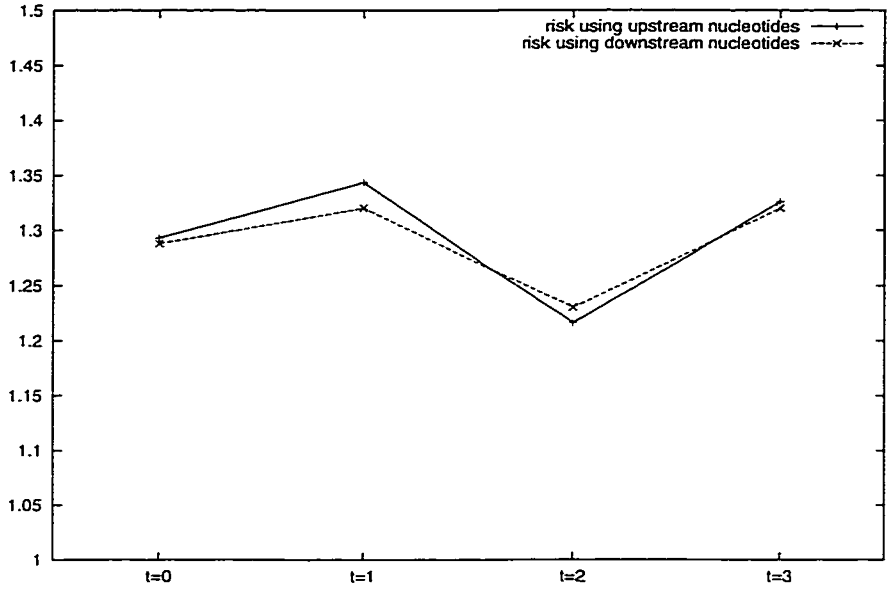


Figure 7: Risk estimates for the RPL models

nucleotides. For each item in the test set we use the RPL regression models to estimate $P(b|b_{-1}, b_{-2}, ..., b_{-r_0}, 0, z)$ and $P(b|b_{-1}, b_{-2}, ..., b_{-r_3}, 3, z)$. We adjust the estimates to reflect the true priors and obtain estimates $\hat{P}(b|b_{-1}, b_{-2}, ..., b_{-r_0}, 0, z)$ and $\hat{P}(b|b_{-1}, b_{-2}, ..., b_{-r_3}, 3, z)$. We calculate the average log likelihood ratio

$$\log \frac{\hat{P}(b|b_{-1}, b_{-2}, ..., b_{-r_0}, 0, z)}{\hat{P}(b|b_{-1}, b_{-2}, ..., b_{-r_3}, 3, z)},$$

over all items in the test set. This gives an idea of how well the model is performing for our specific purpose. These log likelihood ratios for each $t \in \{0, 1, 2\}$ are given in Figure 8.

For the non-coding test sets we use the RPL models to estimate $P(b|b_{-1}, b_{-2}, ..., b_{-r_t}, t, z)$ and $P(b|b_{-1}, b_{-2}, ..., b_{-r_t}, t, z)$ for each $t \in \{0, 1, 2, 3\}$. We obtain estimates $\hat{P}(b|b_{-1}, b_{-2}, ..., b_{-r_t}, t, z)$ and $\hat{P}(b|b_{-1}, b_{-2}, ..., b_{-r_t}, t, z)$ by adjusting

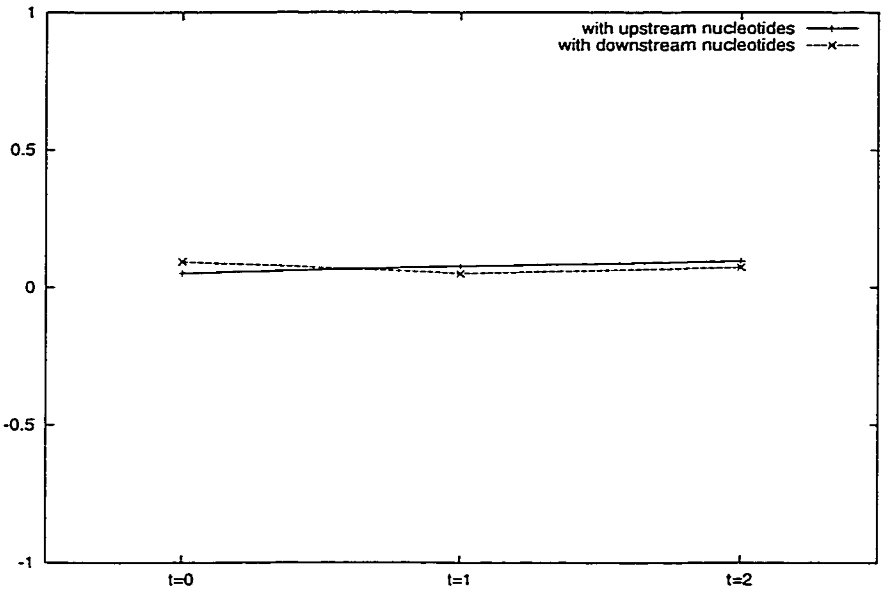


Figure 8: The log likelihood ratio estimates for phase $t \in \{0, 1, 2\}$

for the true priors. We report the average log likelihood ratios,

$$\log \frac{\hat{P}(b|b_{-1}, b_{-2}, ..., b_{-r_t}, t, z)}{\hat{P}(b|b_{-1}, b_{-2}, ..., b_{-r_3}, 3, z)}$$

and

$$\log \frac{\hat{P}(b|b_{+1}, b_{+2}, ..., b_{+n_t}, t, z)}{\hat{P}(b|b_{+1}, b_{+2}, ..., b_{+n_3}, 3, z)},$$

for each coding nucleotide phase, in Figure 9.

The log likelihood ratio estimate, obtained from the non-coding test set, for position three of a codon using the upstream nucleotides is substantially less than the other estimates. Recall that in the RPL model for $t = 2$ and the upstream nucleotides, the feature vector consisted of the C+G content and the two nucleotides immediately upstream, included in the feature vector as a pair.

The training set for this model was selected from exons (with three nucleotides removed from the end) in $\mathcal{L}$, so no stop codons were included in the training set.

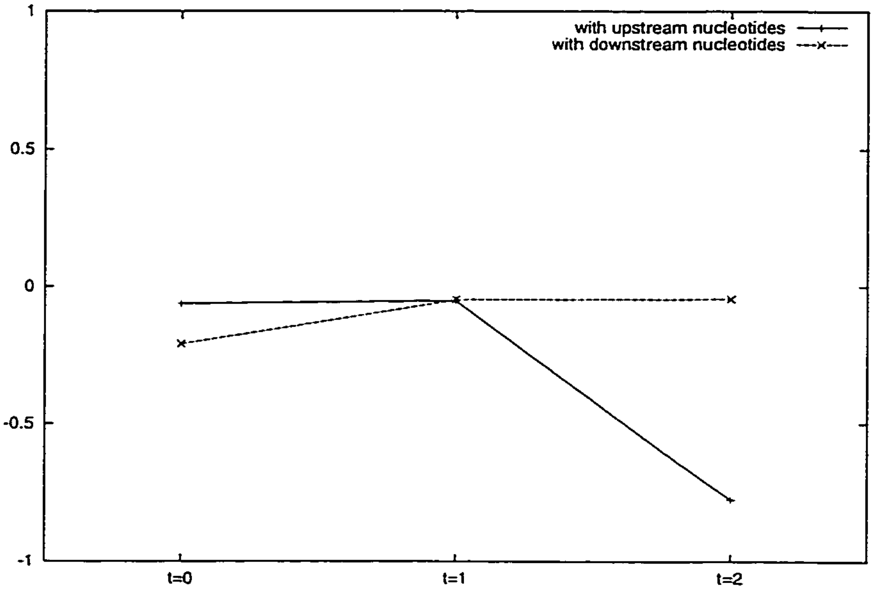


Figure 9: The likelihood ratio estimates for the non-coding test set

Therefore, if the feature vector contains the pair TA, the estimated conditional probability of A and G will be very small. Similarly, if the feature vector contains the pair TG, the estimated conditional probability of A will be very small. In the non-coding test set, the classes A and G can occur with the feature vector TA and the class A can occur with the feature vector TG. When this happens the estimate of

$$\log \frac{P(b|b_{-1}, b_{-2}, 2, z)}{P(b|b_{-1}, b_{-2}, ..., b_{-r_3}, 3, z)}$$

will be a large negative, since $\hat{P}(b|b_{-1}, b_{-2}, 2, z)$ will be very small. Figure 10 contains the proportions of A, C, G, and T, in position three of a codon, given the upstream pair, for all exons in $\mathcal{L}$.

It should be noted that in numerous cases the triples coding for the same amino acid differ only at the third nucleotide. For example, TCA, TCC, TCG, and TCT, all code for the same amino acid, serine. In fact, for all codons with C as the second nucleotide, the amino acid is determined by the first two nucleotides in the codon.

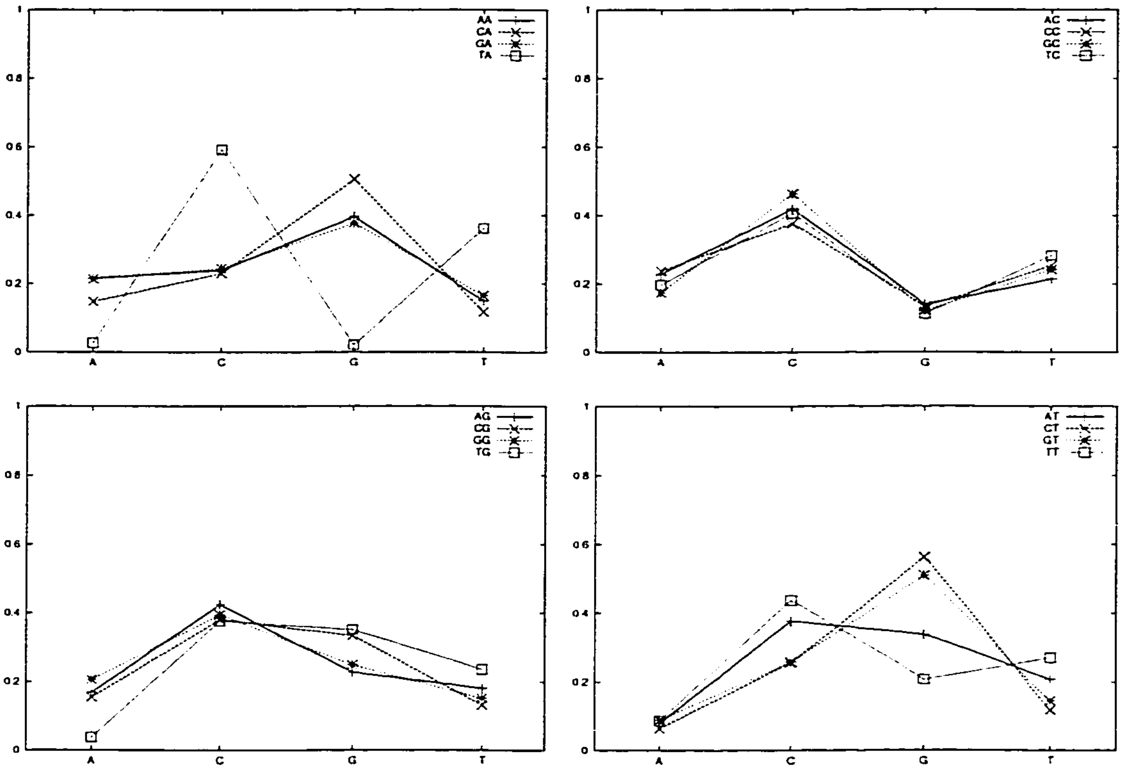


Figure 10: The proportions of A, C, G, and T, in position three of a codon, given the upstream pair

In this case, however, there seems to be a definite preference for C as the third nucleotide. These patterns will be picked up by the RPL model for phase 2. This model will then have high risk on a test set which doesn't observe these patterns as is the case with the non-coding test set.

In the case of phase 2, it was possible to choose an RPL model with high risk on the non-coding test set, without compromising the risk on the phase 2 test set. A similar model for phase 0 gave equally high risk on the non-coding test set but the risk on the phase 0 test set was higher than for other models, and therefore, this model was not chosen as the optimal model.

The gene prediction program using this newly defined content score will be referred to as GRPL1. As in GRPL, direct use of the estimates of the functional site scores results in low sensitivity relative to specificity. To improve accuracy we adjust the functional site score as follows

$$\text{Functional site score} = 2.8 + 1.3 \log \frac{\hat{P}(\text{functional site } |w_i)}{\hat{P}(\text{pseudo site } |w_i)}.$$

These values were selected by tuning GRPL1 on the training set $\mathcal{L}$.

In all Model 1 estimates a total of 2517 parameters. We estimated nine prior probabilities, three for each codon position. For each RPL model, we estimated 11× (the length of the feature vector +1) parameters. The length of the feature vector was $1 + 4\times$ (the number of previous or following nucleotides), except in the case of codon position three where the feature vector had length 17.

3.2 Model 2

Let $h = (b_1, b_2, b_3, b_4, b_5, b_6)$ denote a hexamer observed at some point in x. We define hexamer phase t in a slightly different way to that used in GRPL. Let $t \in \{0, 1, 2, 3\}$ be a value, determined by y and the position of b_6 in the sequence defined as follows: If b_6 is in a coding region determined by the parse then let $t + 1$ be the position of b_6 in the codon containing b_6. If b_6 is a non-coding region then t=3. In GRPL, hexamer phase was defined on the first nucleotide in the hexamer.

For a hexamer $h = (b_1, b_2, b_3, b_4, b_5, b_6)$, we denote by q, the embedded pentamer $(b_2, b_3, b_4, b_5, b_6)$. We define pentamer phase in the same way as hexamer phase so that a hexamer and its embedded pentamer have the same phase.

Under this model the conditional distribution of a hexamer h given y and z, depends only on the hexamer phase t and the C+G proportion in the vicinity of b_6. Similarly, the conditional distribution of a pentamer q given y and z, depends only on the pentamer phase t and the C+G proportion in the vicinity of b_6. For the remainder of this page we will suppress the explicit mention of the C+G content z and assume that all probabilities are conditioned on z.

Using a fifth-order Markov model we can express $P(v_i|y)$ as

$$\begin{aligned}
P(v_i|y) &= P(b_1, b_2, ..., b_m|y) \\
&= P(b_1, .., b_6|y)P(b_7|b_2, .., b_6, y)P(b_8|b_3, .., b_7, y).....P(b_m|b_{m-5}, .., b_{m-1}, y) \\
&= P(b_1, .., b_6|y)\frac{P(b_2, .., b_7|y)}{P(b_2, .., b_6|y)}\frac{P(b_3, .., b_8|y)}{P(b_3, .., b_7|y)}.....\frac{P(b_{m-5}, .., b_m|y)}{P(b_{m-5}, .., b_{m-1}|y)} \\
&= P(h_1|t_1)\frac{P(h_2|t_2)}{P(q_1|t_1)}.......\frac{P(h_{m-5}|t_{m-5})}{P(q_{m-6}|t_{m-6})}
\end{aligned}$$

Therefore

$$\log P(v_i|y) = \sum_{i=1}^{m-5} \log P(h_i|t_i) - \sum_{i=1}^{m-6} \log P(q_i|t_i),$$

and so

$$\text{content score}(v_i) = \log\frac{P(v_i|y)}{P(v_i|y_0)} = \sum_{i=1}^{m-5} \log\frac{P(h_i|t_i)}{P(h_i|3)} - \sum_{i=1}^{m-6} \log\frac{P(q_i|t_i)}{P(q_i|3)}. \qquad (4)$$

For a given hexamer h, with hexamer phase t and C+G proportion z, the likelihood ratio of interest can be expressed as

$$\frac{P(h|t,z)}{P(h|3,z)} = \frac{P(h,t|z)/P(t|z)}{P(h,3|z)/P(3|z)}$$

$$
\begin{aligned}
&= \frac{P(h,t|z)}{P(h,3|z)}\frac{P(3|z)}{P(t|z)} \\
&= \frac{P(h|z)P(t|h,z)}{P(h|z)P(3|h,z)}\frac{P(3|z)}{P(t|z)} \\
&= \frac{P(t|h,z)}{P(3|h,z)}\frac{P(3|z)}{P(t|z)}.
\end{aligned}
$$

Similarly, for a pentamer q with pentamer phase t and C+G proportion z, we have

$$\frac{P(q|t,z)}{P(q|3,z)} = \frac{P(t|q,z)}{P(3|q,z)}\frac{P(3|z)}{P(t|z)}.$$

Therefore, the content score of v_i can be re-expressed as

$$
\begin{aligned}
\text{content score}(v_i) &= \sum_{i=1}^{m-6} \log \frac{P(h_i|t_i,z)}{P(h_i|3,z)}\frac{P(q_i|3,z)}{P(q_i|t_i,z)} + \log \frac{P(h_{m-5}|t_{m-5},z)}{P(h_{m-5}|3,z)} \\
&= \sum_{i=1}^{m-6} \log \frac{P(t_i|h_i,z)}{P(3|h_i,z)}\frac{P(3|q_i,z)}{P(t_i|q_i,z)} + \log \frac{P(t_{m-5}|h_{m-5},z)}{P(3|h_{m-5},z)}\frac{P(3|z)}{P(t_{m-5}|z)}.
\end{aligned}
$$

Consider first the likelihood ratio

$$\frac{P(t|h,z)}{P(3|h,z)}.$$

We employ an RPL model to estimate the conditional distribution of hexamer phase $t \in \{0,1,2,3\}$, given the hexamer h and the C+G content z, that is, to estimate $P(\text{class} \mid \text{feature vector})$, where the class is the hexamer phase and the feature vector consists of the hexamer and the C+G content.

We select mutually exclusive training and test sets from $\mathcal{L}$. As in Model 1, the nucleotides in a hexamer can be included in the feature vector in a number of ways, as six individual nucleotides, as five pairs of nucleotides, or as four triples of nucleotides. Again we consider the grouping system as an additional first-level parameter in the RPL model. The optimal model uses the feature vector which groups the nucleotides in pairs. It has twelve reference points, and the smoothing parameter is set to zero.

In the RPL model the prior probabilities were defined as follows:

$$P^*(3) = \frac{1}{2},$$

$$P^*(t) = \frac{1}{6}, \text{ for } t \in \{0, 1, 2\},$$

that is, equal prior probabilities were used for coding and non-coding hexamer phase, and within coding, equal prior probabilities were used for each $t \in \{0, 1, 2\}$.

Let $p(\text{coding})$ be the true prior probability of coding hexamer phase, and let $p(3)$ be the true prior probability of non-coding hexamer phase. Clearly $p(\text{coding}) = 1 - p(3)$, and since each coding hexamer phase must occur equally often, we see that the true prior probability of each $t \in \{0, 1, 2\}$ is given by

$$p(t) = \frac{1 - p(3)}{3}.$$

We use the invariance property of RPL regression to modify the estimates obtained from the RPL model to account for the true priors. Let $P^*(t|h, z)$ be the value of the conditional distribution of hexamer phase given by the RPL model. We modify these estimates using equation (3) to obtain estimates $\hat{P}(t|h, z)$ as follows:

$$\begin{aligned} \frac{\hat{P}(t|h, z)}{\hat{P}(3|h, z)} &= \frac{p(t)P^*(t|h, z)/\frac{1}{6}}{p(3)P^*(3|h, z)/\frac{1}{2}} \\ &= \frac{3p(t)P^*(t|h, z)}{p(3)P^*(3|h, z)} \\ &= \frac{p(\text{coding})P^*(t|h, z)}{p(3)P^*(3|h, z)}. \end{aligned}$$

Let q be a pentamer with pentamer phase t and C+G proportion z, we can repeat the above procedure to obtain an RPL regression model which estimates the conditional distribution of pentamer phase t given the pentamer q and the C+G content z, that is, which estimates $P(\text{class}|\text{feature vector})$, where the class is pentamer phase and the feature vector consists of q and z. We include the pentamer in the feature vector as four pairs. The chosen RPL model has twelve reference points and the smoothing parameter is set to zero.

Consider now the remaining ratio to be estimated,

$$\frac{P(3|z)}{P(t|z)}, \text{ where } t \in \{0, 1, 2\}.$$

This can be rewritten as

$$\frac{P(3|z)}{P(t|z)} = \frac{3P(3|z)}{P(\text{coding}|z)}. \tag{5}$$

We use RPL regression to model the conditional distribution of coding hexamer phase versus non-coding hexamer phase given z. Equal prior probabilities for coding and non-coding are used. Two reference points and a zero smoothing parameter are chosen as optimal. Let $P^{\star}(3|z)$ and $P^{\star}(\text{coding}|z)$ be the values given by the RPL model for a particular z. Let $\hat{P}(3|z)$ and $\hat{P}(\text{coding}|z)$ be the values given by equation (3) on adjusting for the correct prior probabilities. Hence

$$\frac{\hat{P}(3|z)}{\hat{P}(\text{coding}|z)} = \frac{p(3)P^{\star}(3|z)}{p(\text{coding})P^{\star}(\text{coding}|z)}.$$

Therefore, by equation (5) we see that the ratio of interest is estimated as follows:

$$\begin{aligned} \frac{\hat{P}(3|z)}{\hat{P}(t|z)} &= \frac{3\hat{P}(3|z)}{\hat{P}(\text{coding}|z)} \\ &= \frac{3p(3)P^{\star}(3|z)}{p(\text{coding})P^{\star}(\text{coding}|z)}. \end{aligned}$$

We combine the estimates obtained from the three RPL models to produce an estimate of the content score

$$\text{content score}(v_i) = \sum_{i=1}^{m-6} \log \frac{P^*(t_i|h_i,z)}{P^*(3|h_i,z)} \frac{P^*(3|q_i,z)}{P^*(t_i|q_i,z)} + \log \frac{3P^*(t_{m-5}|h_{m-5},z)}{P^*(3|h_{m-5},z)} \frac{P^*(3|z)}{P^*(\text{coding}|z)}.$$

This model gave poor results, however, when incorporated into GRPL. The reason for this may be that when z is small, both $P^*(3|h,z)$ and $P^*(3|q,z)$ tend to be large, so we end up with a ratio of two large numbers and a ratio of two small numbers and any information tends to be lost. Also, the risk for the RPL model used to estimate $P(t|q,z)$ is higher than the risk for the model used to estimate $P(t|h,z)$, so more noise is being introduced and true information is harder to find.

An alternative approach is now considered in an attempt to overcome this problem. Recall equation (4),

$$\text{content score}(v_i) = \sum_{i=1}^{m-5} \log \frac{P(h_i|t_i,z)}{P(h_i|3,z)} - \sum_{i=1}^{m-6} \log \frac{P(q_i|t_i,z)}{P(q_i|3,z)}.$$

We estimate the content score as follows:

$$\text{content score}(v_i) = \sum_{i=1}^{m-5} \log \frac{P(h_i|t_i,z)}{P(h_i|3,z)} - r \sum_{i=1}^{m-6} \log \frac{P(h_i|t_i,z)}{P(h_i|3,z)}.$$

where r is a constant which is chosen by tuning the gene prediction program on $\mathcal{L}$. In order to see why this approximation is plausible, note that for a true exon we assume

$$\sum_{i=1}^{m-5} \log \frac{P(h_i|t_i,z)}{P(h_i|3,z)} > 0,$$

and

$$\sum_{i=1}^{m-6} \log \frac{P(q_i|t_i,z)}{P(q_i|3,z)} > 0,$$

and that the first of these will be greater than the second. We also assume that there is a large association between these quantities. For a falsely predicted exon we assume

$$\sum_{i=1}^{m-5} \log \frac{P(h_i|t_i, z)}{P(h_i|3, z)} < 0,$$

and

$$\sum_{i=1}^{m-6} \log \frac{P(q_i|t_i, z)}{P(q_i|3, z)} < 0,$$

and that the absolute value of the first of these will be greater than the absolute value of the second. Again we assume a large association between them. Under these circumstances it is possible that

$$\sum_{i=1}^{m-6} \log \frac{P(q_i|t_i, z)}{P(q_i|3, z)} \approx r \sum_{i=1}^{m-6} \log \frac{P(h_i|t_i, z)}{P(h_i|3, z)}.$$

For a given hexamer h, with hexamer phase $t \in \{0, 1, 2\}$ and C+G content z, we use the RPL models to estimate the log likelihood ratio of interest as follows:

$$\begin{aligned}
\frac{\hat{P}(h|t, z)}{\hat{P}(h|3, z)} &= \frac{\hat{P}(t|h, z)}{\hat{P}(3|h, z)} \frac{\hat{P}(3|z)}{\hat{P}(t|z)} \\
&= \frac{p(\text{coding}) P^*(t|h, z)}{p(3) P^*(3|h, z)} \frac{p(3) P^*(3|z)}{p(\text{coding}) P^*(\text{coding}|z)} \\
&= \frac{3P^*(t|h, z)}{P^*(3|h, z)} \frac{P^*(3|z)}{P^*(\text{coding}|z)}.
\end{aligned}$$

We evaluate the performance of this estimator using four test sets, one for each hexamer phase $t \in \{0, 1, 2, 3\}$. Each test set has approximately 10,000 entries. For a hexamer in a coding region, that is, with hexamer phase $t \in \{0, 1, 2\}$ we would like

$$\log \frac{\hat{P}(h|t, z)}{\hat{P}(h|3, z)} > 0.$$

Consider hexamer phase $t = 0$, for each item in the test set we estimate

$$\log \frac{P(h|0,z)}{P(h|3,z)}.$$

We then calculate the average log likelihood ratio over all items in the test set. These log likelihood ratios for each $t \in \{0,1,2\}$ are given in Figure 11.

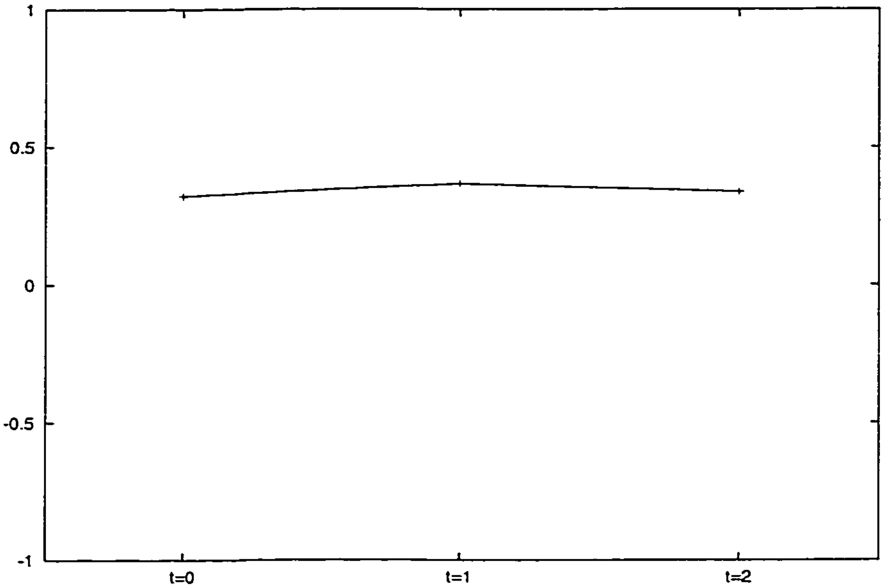


Figure 11: The log likelihood ratio estimates for hexamer phase $t \in \{0,1,2\}$

For a hexamer in a non-coding region we would like

$$\log \frac{\hat{P}(h|t,z)}{\hat{P}(h|3,z)} < 0, \text{ for each } t \in \{0,1,2\}.$$

For each item in the non-coding test set, we calculate this log likelihood ratio estimate for each $t \in \{0,1,2\}$. The average log likelihood ratios, for $t \in \{0,1,2\}$, are given in Figure 12.

These newly defined content scores are incorporated into a program called GRPL2. The value of r and the functional site constants are found by tuning GRPL2 on the training set $\mathcal{L}$. A value $r = 0.88$ is chosen and the functional site score is adjusted as

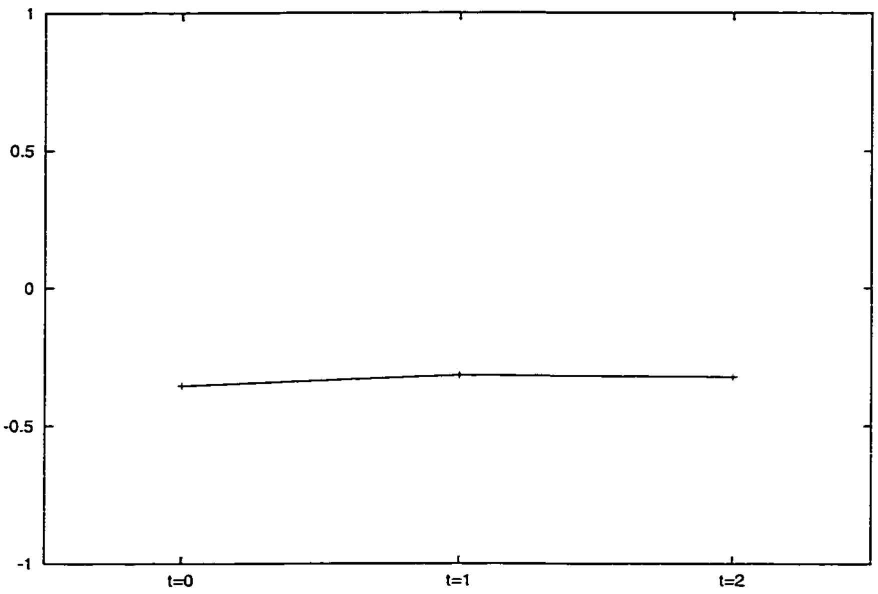


Figure 12: The likelihood ratio estimates for the non-coding test set

follows

$$\text{Functional site score} = 3 + 1.2 \log \frac{\hat{P}(\text{functional site } |w_i)}{\hat{P}(\text{pseudo site } |w_i)}.$$

Model 2 estimates a total of 904 parameters. The first RPL model involves 902 parameters and the second RPL model involves 2 parameters.

3.3 Results and Conclusions

GRPL1 and GRPL2 are now compared to the best performers out of the gene prediction programs discussed in Chapter 1, which do not use sequence alignment techniques. Table 5 contains the performance results on the Burset/Guigó test set, Tables 6 and 7 contain the performance results on the GeneParser test sets.

Table 5: Performance comparisons for the Burset/Guigó test set.

Program	Sn	Sp	Sq	CC	AC	XSn	XSp
GRPL(Hu)	0.93	0.93	0.984	0.91	0.91	0.76	0.79
GENSCAN	0.93	0.93	n/a	0.92	0.91	0.78	0.81
GeneParser3	0.86	0.91	n/a	0.85	0.86	0.56	0.58
Genie	0.86	0.81	0.964	0.80	0.80	0.68	0.64
GRPL1	0.89	0.91	0.981	0.88	0.87	0.71	0.74
GRPL2	0.86	0.89	0.975	0.84	0.83	0.66	0.71

Table 6: Performance comparisons for the GeneParser I test set.

Program	Sn	Sp	Sq	CC	AC	XSn	XSp
GRPL(Hu)	0.96	0.88	0.978	0.90	0.91	0.72	0.69
GENSCAN	0.97	0.88	0.982	0.91	0.91	0.74	0.73
Genie	0.86	0.81	0.964	0.80	0.80	0.68	0.64
GRPL1	0.97	0.90	0.984	0.93	0.92	0.76	0.75
GRPL2	0.94	0.85	0.962	0.87	0.87	0.68	0.66

Table 7: Performance comparisons for the GeneParser II test set.

Program	Sn	Sp	Sq	CC	AC	XSn	XSp
GRPL(Hu)	0.89	0.93	0.989	0.89	0.90	0.69	0.77
GENSCAN	0.89	0.92	0.986	0.90	0.89	0.68	0.69
Genie	0.75	0.76	0.967	0.71	0.72	0.54	0.51
GRPL1	0.78	0.85	0.974	0.78	0.77	0.59	0.68
GRPL2	0.78	0.87	0.974	0.79	0.76	0.58	0.69

Neither GRPL1 or GRPL2 match the performance of GRPL or GENSCAN on the Burset/Guigó test set or the GeneParser II test set. GRPL1 does a little better than GRPL and GENSCAN on the GeneParser I test set. GRPL1 performs better than GRPL2 on all three test sets, particularly at the exon level.

GRPL2 appeared promising judging by Figures 11 and 12. We believe that the poor performance of GRPL2 has to do with the choice of the constant r used to adjust the log likelihoods. Alternative adjustment choices might be more effective. It is conceivable that both GRPL1 and GRPL2 could be improved with some tuning.

Bibliography

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *J. Mol. Biol.* **273**, 355-368.

Benveniste, A., Métivier, M., & Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximations*, New York: Springer-Verlag.

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

Burset, M. & Guigó, R. (1996). Evaluation of gene structure prediction programs. *Genomics* **34**, 353-367.

Etienne-Decant, J. (1988). *Genetic Biochemistry: from gene to protein*, Ellis Horwood limited.

Guigó, R., Knudsen, S., Drake, N., & Smith, T. F. (1992). Prediction of gene structure. *J. Mol. Biol.* **226**, 141-157.

Hooper, P. M. (1999). Reference point logistic classification. *Journal of Classification.* **16**, 91-116.

Hooper, P. M. (2000). Reference point logistic regression and the identification of DNA functional sites. Submitted.

Hooper, P. M., Zhang, H., & Wishart, D. S. (2000). Prediction of genetic structure in eukaryotic DNA using reference point logistic regression and sequence alignment. *Bioinformatics* **16**, 425-438.

Kulp, D., Haussier, D., Reese, M. G. & Eeckman, F. H. (1196). A generalized Hidden Markov Model for the recognition of human genes in DNA. In *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology.* AAAI Press, Menlo Park, CA.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, **1**, 281-297.

Senapathy, P., Shapiro, M. B., & Harris, N. L. (1990). Splice junctions, branch point sites, and exons: sequence statistics, identification, and application to genome project. *Methods Enzymol.* **183**, 252-278.

Snyder, E. E. & Stormo, G. D. (1995). Identification of protein coding regions in genomic DNA. *J. Mol. Biol.* **248**, 1-18.

Staden, R. (1984). Computer methods to locate signals in nucleic acid sequences. *Nucl. Acids Res.* **12**, 505-519.

Thomas, A., & Skolnick, M. H. (1994). A probabilistic model for detecting coding regions in DNA sequences. *IMA J. Math. Appl. Med. Biol.* **11**, 149-160.

Xu, Y., Einstein, J. R., Mural, R. J., Shah, M. & Uberbacher, E. C. (1994). An improved system for exon recognition and gene modeling in human DNA sequences. In *Proceedings of the second International Conference on Intelligent Systems for Molecular Biology*, pp. 376-384, AAAI Press, Menlo Park, CA.

Zhang, M. Q. & Marr, T. G. (1993). A weight array method for splicing signal analysis. *Comp. Appl. Biol. Sci.* **9(5)**, 499-509.