Transcript prediction in eukaryotes using hidden Markov models

Kasper Munch
Bioinformatics Centre
Institute of Molecular Biology and Physiology
The Faculty of Science
University of Copenhagen
Denmark

November 30, 2005

Contents

Preface

This thesis is written in order to fulfil the requirements of the Ph.D. degree. The theme is transcript prediction in eukaryotes. As two alternative approaches towards this goal I have used hidden Markov models for *ab initio* transcript prediction as well as for detection of expression though analysis of tiling microarray experiments. Chapter one gives a short description of eukaryotic gene structure and the available information for modelling. Chapter two is a review of the use of hidden Markov models and their use for gene modelling. The contents of chapter three is a manuscript submitted to BMC Bioinformatics presenting Agene, an automatically generated HMM gene finder for use on newly sequenced genomes. Chapter four contains a manuscript to be submitted to Bioinformatics discussing correlations between suboptimal predictions of Agene and annotated alternative transcripts. Chapter five is a review of the accumulating experimental evidence for novel expression. Chapter six is a manuscript submitted to BMC Bioinformatics describing an analysis of tiling microarray data and the detection of expressed sequences. The last chapter lists some future directions.

I would like to thank my supervisor Anders Krogh for valuable discussions and advice. In addition Paul Gardner must be thanked for insipiration, brilliant comments, breathtaking enthusiasm, linguistic insight, and striking good looks. It goes without saying that my beloved Dorthe must be thanked for her support.

Chapter 1

The eukaryotic gene

1.1 Defining a gene

The definition of a gene has evolved from the notion that each gene produces a single protein. Recent estimates based on exon-junction array studies of 52 Human tissues and cell-lines suggest that as much as 74% of Human genes are alternatively spliced (Johnson et al., 2003). Defining a gene as a transcriptional unit - the collection of transcripts in the same orientation grouped by overlaps in transcribed sequence - now seems more appropriate. This also accounts for transcripts that are not protein coding. Based on recent large scale cDNA experiments even this broad definition seems outdated. A definition is called for that incorporates phenomena such as anti-sense transcription that has been shown for 72% transcriptional units using full-length cDNA (Katayama et al., 2005).

The work presented in this thesis aims in part to model and predict the structure of protein coding genes and in part to model expression data from tiling array experiments to infer expression. In the context of coding gene prediction I define a gene as the collection of transcripts that can be mapped to the same locus by pairwise overlaps of coding regions. In the context of expression detection analysis no strict gene definition is sensible since the analysis of expression data is an attempt to find not only coding and non-coding spliced transcripts but also miRNA, snRNA, snoRNA, tRNA, as well as RNA genes of unknown function. In addition, much of this novel expression is expected contain transcription in both sense and anti-sense directions.

1.2 Structure of protein coding genes

The structure of protein coding genes is naturally summarised by referring to the functional context that the different elements are part of. The first step in expression of genes is the assembly of the RNA polymerase II transcription-initiation

complex at the transcription start site. The positioning of the polymerase involves general transcription factors that are required for the transcription of most genes. These general factors bind to transcription factor binding sites, among which the most prevalent are the TATA-box and the BRE, INR, and DPE elements (Alberts et al., 2002). Combinations of special transcription factors with separate binding sites regulate the expression of individual genes. Transcription proceeds from the transcription start to the poly-adenylation (poly-A) site producing an RNA copy with thymine replaced with uracil. At the poly-A site transcription is terminated and a sequence of adenosines are is added to the end of the transcript. This pre-mRNA transcript contains a protein coding sequence (CDS) interrupted by introns. Each intron is bounded by an upstream donor splice site and a downstream acceptor splice site. The removal of introns from the transcript is performed by the spliceosome, a collection of small nuclear ribonucleoproteins (snRNPs), and proceeds as outlined in Figure 1.1 .

After splicing the mature mRNA contains an uninterrupted protein coding sequence that is translated into protein by the ribosome. The small ribosomal subunit, carrying a methionine tRNA that recognises the AUG start codon, binds to the 5' UTR and scans downstream until the first AUG triplet is found. The large ribosomal subunit then completes the ribosome and translation proceeds mediated by tRNAs specific for each nucleotide triplet. Each such triplet, or codon, encodes an amino acid in the growing peptide chain. When the ribosome encounters the first of three stop codons (either an UAG, UGA, or UAA) translation terminates. For this reason the number of nucleotides in the CDS is always a multiple of three. The sequence region from a start codon to the first occurrence of a stop codon is denoted an open reading frame (ORF).

Several pairs of donor and acceptor splice sites may exist for each intron. This allows for alternative ways to splice the transcript, resulting in alternative mature mRNA transcripts that in turn produce alternative proteins if the alternative splicing occurs in the coding region. The types of alternative splicing include elongation and truncation of both 5' and 3' ends of exons, exon skipping, use of alternative exons and intron retention. In addition, alternative transcription and translation start sites and truncation of the reading frame by frame shifting splicing adds to the plethora of splice forms. These are exemplified in Figure 1.2

1.3 Sequence level information

Modelling of gene structure amounts to capturing as much information as practically possible. In *ab initio* gene prediction the only information available is that contained in the nucleotide sequence. The sequence offers three kinds of information: Signals, sequence content, and sequence length information.

1.3.1 Signal Information

Sequence signals are motifs that serve as a cis-acting elements for either RNA or protein binding factors. The transcription start site does not constitute a strong signal. The CDS is always initiated by an ATG and terminated by one of the three stop codons. In addition, the immediately upstream and downstream positions of each signal also contains information. The fact that the two signals occur at a distance divisible by three adds to the contributed information. The branch point that binds the branch-point-binding protein early in the splice reaction does generally not contribute important information. The branch point is always an adenosine, but a significant consensus motif or neighbouring positions is only seen in some species. In contrast the splice sites recognised by splice factors constitute valuable signals due to highly conserved consensus motifs. In mammals the first two positions in the intron part of the donor site is GT in more than 98 percent of the cases. The second most frequent dinucleotide GC accounts for most of the remaining fraction (Burset et al., 2000). The last two positions in the intron part of the acceptor site are invariably AG. Upstream of the acceptor site is a region enriched in pyrimidine. This poly-pyrimidine tract binds the U2 auxiliary factor early in the splicing reaction. The information contained in the splice sites and especially the poly-pyrimidine tract vary between species. The poly-A site, that terminates transcription, also has a strong consensus sequence. The logos (Schneider and Stephens, 1990) in Figure 1.3 exemplify the important consensus motifs in *C. elegans*.

1.3.2 Sequence content Information

The higher order sequence content - the distribution of nucleotide words - also contributes information to gene models, especially within the coding region due to the over-representation of some nucleotide triplets in the reading frame compared to the two other frames. There is even a dependency between neighbouring codons that makes the distribution of six-nucleotide words yet more informative. The higher order word distribution of non-coding exons and introns is less informative.

1.3.3 Length Information

Valuable information is contained in the length distributions of gene structure blocks such as UTR exons, UTR introns, the UTR part of first exons, the coding part of lasts exons, introns in coding regions, internal coding exons, the coding part of last exons, the UTR part of last exons, and single coding exons. Most of these elements have a characteristic length distribution that can be captured by a gene model. The underlying mechanisms that impose the non-geometric features

on the length distributions are not well understood, but relate to exon recognition in splicing (Berget, 1995). Length distributions of gene structure blocks in *D. melanogaster* are shown in Figure 1.4.

1.4 Other information

Other available information for the identification of genes is homology to sequence regions of related species where gene structure is annotated.

The phylogenetic information available from genomic alignments of two or more species serves as an additional source of information. The molecular evolution of positions in different gene structure blocks varies and can be used in discrimination.

Experimental evidence is also used for gene prediction. Matching of full-length cDNA, mRNA, proteins, and ESTs to genomic sequence is used as evidence. This method is used extensively in the Ensembl automatic gene annotation system (Curwen et al., 2004).

Lately, data from tiling microarray experiments querying large genomic regions have become available. These data offer a unique possibility to make inferences of expression not only for coding genes but also for RNA genes of various sorts. This is the topic of chapter 5 and 6.

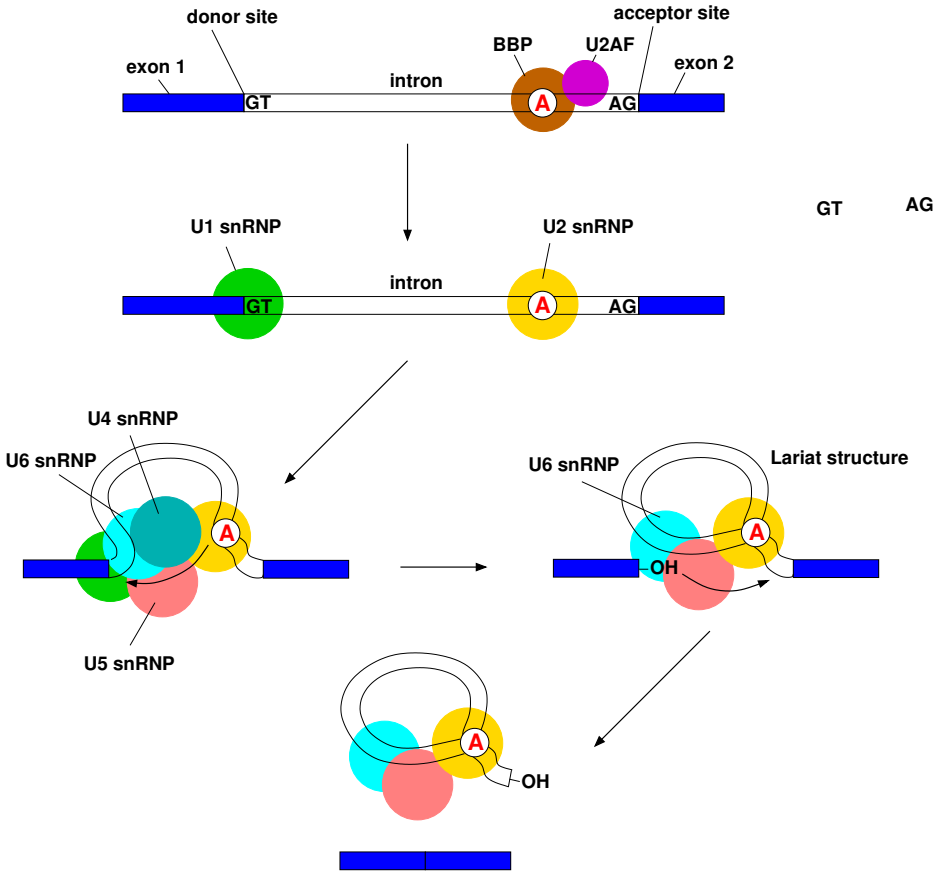


Figure 1.1. Splicing of the pre-mRNA transcript. The branch-point is first recognised by the branch point binding protein (BBP) and the U2 auxiliary factor (U2AF). Next, the U2 snRNP displaces BBP and U2AF and forms base pairs with the branch-point consensus sequence, and the U1 snRNP forms base pairs with the donor splice site. At this point the U4/U6/U5 snRNP complex enters the spliceosome. Though several RNA-RNA rearrangements two trans-estrifications take place that first turn the intron into a lariat structure and secondly frees this lariat by joining the donor and acceptor splice sites. The figure is adapted from Alberts *et. al* (2002).

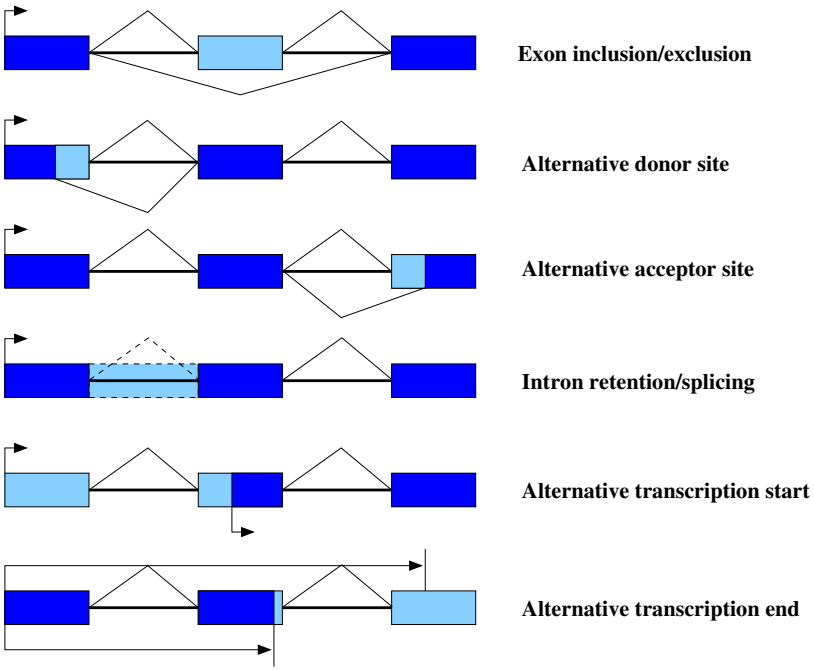
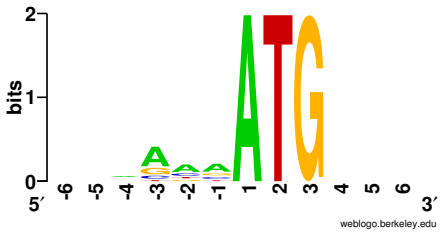
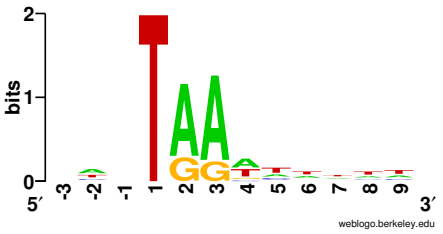


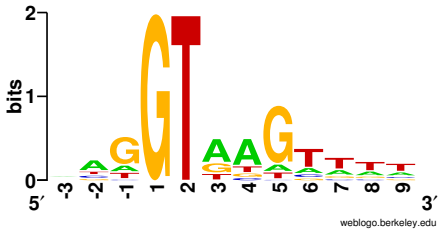
Figure 1.2. Alternative splicing. The types of alternative splicing include exclusion or retention of cassette exons, alternative donor and acceptor sites, and intron splicing or retention. In addition to alternative splicing, alternative transcription start sites and termination sites contribute to the variability of splice forms.



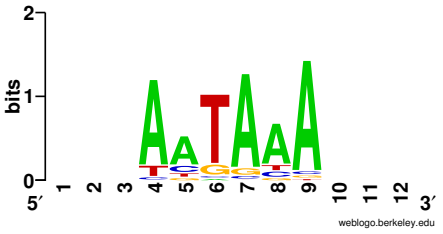
(a) Coding start



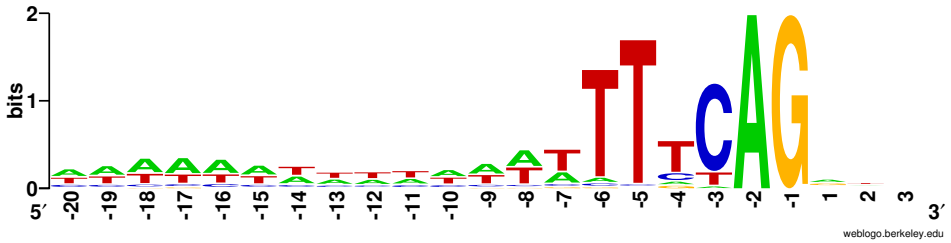
(b) Coding end



(c) Donor site



(d) Poly-adenylation site



(e) Poly-pyrimidine tract and acceptor site

Figure 1.3. Logos for the major sequence signals in *C. elegans* genes. Logos are a graphic representations of aligned instances of the same sequence signals. The relative heights of letters correspond to frequencies of bases at each position. The degree of sequence conservation is reflected in the total height of a stack of letters measured in bits of information. **(a)**: Coding start. **(b)**: Coding end. **(c)**: Donor site. **(d)**: Poly-adenylation site (as predicted by Agene). **(e)**: Acceptor site and poly-pyrimidine tract.

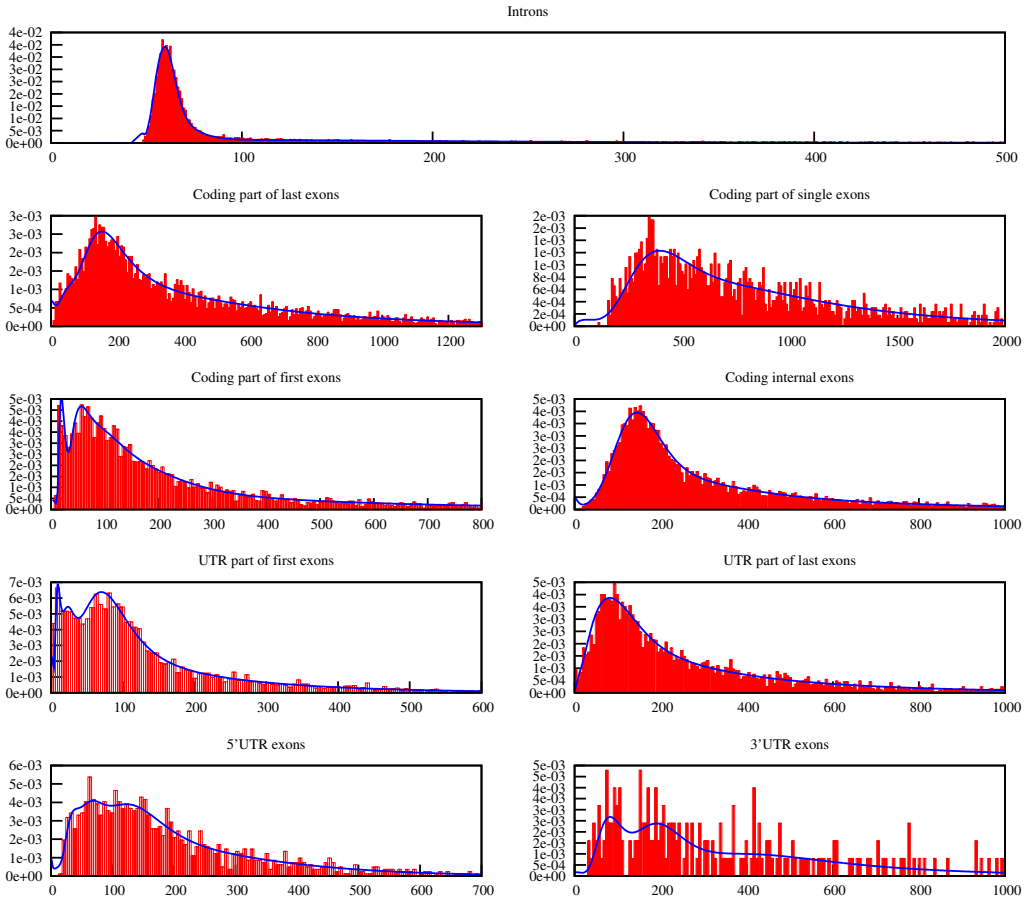


Figure 1.4. Length distributions of gene structure elements in *D. melanogaster*. A curve is fitted to to each histogram to emphasise the shapes.

Chapter 2

Gene prediction with hidden Markov models

The notation used in this chapter is adapted from Durbin *et. al* (1998) .

2.1 Hidden Markov models

A Markov chain is a sequence of states, in which the probability of new states, the transition probabilities, are dependent only on the current state. Given the transition probabilities the Markov chain generates a path, $\boldsymbol{\pi}$, of states. If the i'th state in such a path is called π_i, then the transition probability to this state from a previous state π_{i-1} can be expressed as:

$$a_{kl} = P(\pi_i = k | \pi_{i-1} = l). \tag{2.1}$$

The chain is initiated by a begin state at position 0 and terminated by an absorbing end state. The transition probabilities from the begin state and to the end state can be treated as normal transitions. Since each transition depends only on the current state the probability of a state path is:

$$P(\boldsymbol{\pi}) = P(\pi_L|\pi_{L-1})P(\pi_{L-1}|\pi_{L-2}) \cdots P(\pi_1|\pi_0)P(\pi_0) \tag{2.2}$$

$$= a_{0\pi_1} \prod_{i=1}^{L} a_{\pi_i \pi_{i+1}}, \tag{2.3}$$

since the probability of the begin state $P(\pi_0)$ is set to one.

A hidden Markov model (HMM) is different from a Markov model in that the path of states is not observed. Rather, each state, π_i, in the path emits an observable from an alphabet $\mathcal{A}$. These observables then constitute the observed sequence, $\boldsymbol{x}$. The correspondence between the path of states and the sequence of

observables is determined by the emission probabilities by which each state emits each observable. The probability of emitting observable b from state k is:

$$e_k(b) = P(x_i = b|\pi_i = k). \tag{2.4}$$

The application of HMMs used here is for modelling gene structure. Hence, for our purposes, states correspond to positions in gene structure elements such as exon and introns, and the observables are nucleotides observed at these positions. This is illustrated by the simple HMM in Figure 2.1.

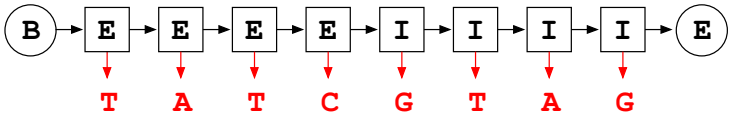


Figure 2.1. Example HMM. Exon and intron region is represented are by E, and I states that each emit nucleotides from characteristic distributions. Black boxes and arrows represent states and transitions. Red arrows and letters represent emissions and observables.

Given the transition and emission probabilities $\boldsymbol{e}$ and $\boldsymbol{a}$ the joint probability of an observed sequence, $\boldsymbol{x}$ and a state path $\boldsymbol{\pi}$ is given by:

$$P(\boldsymbol{x}, \boldsymbol{\pi}) = a_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) a_{\pi_i \pi_{i+1}}, \tag{2.5}$$

where $\pi_{L+1} = 0$. The state path, $\boldsymbol{\pi}$, is rarely known, but the correspondence between observables and states presented by (2.5) is used for both estimation of a best state path for the sequence of observables as well as for training of model parameters.

2.1.1 Higher order Markov chains

The Markov chain in the HMM described above is zeroth order. The order of the Markov chain indicates how far back the dependency of transition probabilities reaches. Hence, in a nth order Markov chain the probabilities of transition to a certain state are dependent on the n states preceding it. Formally:

$$P(\pi_i|\pi_{i-1}, \pi_{i-2}, \ldots, \pi_1) = P(\pi_i|\pi_{i-1}, \pi_{i-2}, \ldots, \pi_{i-n}). \tag{2.6}$$

From standard laws of conditional probabilities we have that

$$P(\pi_i|\pi_{i-1}, \pi_{i-2}, \ldots, \pi_{i-n}) = P(\pi_i, \pi_{i-1}, \ldots, \pi_{i-n+1}|\pi_{i-1}, \pi_{i-2}, \ldots \pi_{i-n}). \tag{2.7}$$

In other words, the probability of a state given an n-tuple of previous states is equal to the probability of an n-tuple ending at position i given the n-tuple of states ending at position $i-1$. This means that an nth order Markov chain with a state alphabet, $\mathcal{A}$, of single letter words is equivalent to a first order Markov chain with a state alphabet of n-letter words. This correspondence allows one to model higher order state dependencies in an way equivalent to first order models. For example, a second order chain with state alphabet $\mathcal{A} \in \{A, B\}$ corresponds to a first order chain with state alphabet $\mathcal{A} \in \{AA, BB, AB, BA\}$. Analogously, a first order model of n-letter words can be turned into a first order model of single-letter words by expanding the state alphabet.

2.1.2 Higher order emissions

Often the dependences we are interested in are the ones between observables and not between states. In this case higher order emission probabilities can be used. An nth order emission probability is dependent on the previous n observables:

$$e_k(b_0|b_1, b_2, \ldots, b_n) = P(x_i = b_0|\pi_i = k, x_{i-1} = b_1, x_{i-2} = b_2, \ldots, x_{i-n} = b_n) \tag{2.8}$$

Redefining emission probabilities this way allows for implantation of higher order emission probabilities using algorithms developed for standard HMMs.

2.2 Decoding

The motivation for using HMMs to model sequences, is that these allow one to estimate the underlying path of states most likely to produce an observed sequence. In the case of gene structures, states correspond to different types of sequence. E.g. exon, intron, or positions in signal sequence such as a splice site. Predicting gene structure from nucleotide sequence thus corresponds to finding a state paths that optimises the probability of the sequence. This step is called decoding.

2.2.1 The Viterbi algorithm

The Viterbi algorithm is used in situations where one wants to predict only the single most likely path $\boldsymbol{\pi}^*$:

$$\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi}} P(\boldsymbol{x}, \boldsymbol{\pi}) \tag{2.9}$$

Due to the Markov property of the state path the best sub-path ending in state k with observation i can be used to find the sub-best path ending in state $k+1$

with observation l. This is done recursively to find the best path for the entire sequence:

$$v_l(i+1) = e_l(x_{i+1}) \max_k (v_k(i) a_{kl}) \tag{2.10}$$

The optimal path is most efficiently obtained using dynamic programming. Using pointers to track which best sub-path each new sub-path was produced from, the full path can be found by backtracking.

$$\begin{array}{lll}
\text{Initialisation:} & \text{for } i = 0: & v_0(0) = 1, v_k(0) = 0 \text{ for } k > 0. \\
\text{Iteration:} & \text{for } i = 1, \ldots, L: & v_l(i) = e_l(x_i) \max_k v_k(i-1) a_{kl} \\
 & & \text{pointer}_i(l) = \arg\max_k v_k(i-1) a_{kl} \\
\text{Termination:} & & P(x, \pi^*) = \max_k v_k(L) a_{k0} \\
 & & \pi_L^* = \arg\max_k v_k(L) a_{k0} \\
\text{Traceback:} & \text{for } i = L, \ldots, 1: & \pi_{i-1}^* = \text{pointer}_i(\pi_i^*)
\end{array} \tag{2.11}$$

2.2.2 The forward algorithm

The forward algorithm is used to obtain the probability of a sequence of observables given transition and emission probabilities. The probability of a sequence, $P(\boldsymbol{x})$, is the sum of contributions from all possible paths and is often referred to as the forward probability:

$$P(\boldsymbol{x}) = \sum_{\boldsymbol{\pi}} P(\boldsymbol{x}, \boldsymbol{\pi}). \tag{2.12}$$

The forward algorithm is very similar to the Viterbi algorithm. Where Viterbi recursively finds the most probable sub-path up to a certain position in the sequence the forward algorithm recursively finds the sum of all paths up to a specific position.

$$f_l(i+1) = e_l(x_{i+1}) \sum_k f_k(i) a_{kl} \tag{2.13}$$

Note the correspondence to (2.10). The full algorithm is similar to the Viterbi:

$$\begin{array}{lll} \text{Initialisation:} & \text{for } i = 0: & f_0(0) = 1, f_k(0) = 0 \text{ for } k > 0. \\ \text{Iteration:} & \text{for } i = 1, \ldots, L: & f_l(i) = e_l(x_i) \sum_k f_k(i-1) a_{kl} \\ \text{Termination:} & & P(\boldsymbol{x}) = \sum_k f_k(L) a_{k0} \end{array} \tag{2.14}$$

2.2.3 Posterior probabilities of states

Knowing the probability that a particular state k is assigned to an observable in a sequence is often useful. This is the posterior probability of state k at position i given the observed sequence:

$$P(\pi_i = k|\boldsymbol{x}) = \frac{P(x_1, \ldots, x_i|\pi_i = k) P(x_{i+1}, \ldots, x_L|\pi_i = k)}{P(\boldsymbol{x})} \tag{2.15}$$

The denominator is the forward probability from (2.12). The first term in the numerator is the summed probability of all paths ending at state k at position i. This quantity, $f_k(i)$, is obtained in the computation of the forward probability. To calculate $P(\pi_i = k|\boldsymbol{x})$ we then need to calculate the second factor in the numerator. This is the summed probability of all paths initiated from state k at position i and ending at position L. This quantity, $b_k(i)$, is calculated the same way as the forward probabilities, but starting from the opposite end of the sequence. For obvious reasons this is called the backward algorithm:

$$\begin{array}{lll} \text{Initialisation:} & \text{for } i = L: & b_0(L) = a_{k0} \text{ for all } k. \\ \text{Iteration:} & \text{for } i = L-1, \ldots, 1: & b_l(i) = \sum_l a_{kl} e_l(x_{i+1}) b_l(i+1) \\ \text{Termination:} & & P(x) = \sum_l a_{0l} e_l(x_1) b_l(1) \end{array} \tag{2.16}$$

Using dynamic programming to calculate the forward and backward probabilities thus supplies us with everything we need to calculate the posterior probability of any state at any position using:

$$P(\pi_i = k|\boldsymbol{x}) = \frac{f_k(k) b_k(i)}{P(\boldsymbol{x})} \tag{2.17}$$

2.2.4 Posterior decoding

Often a single path does not meaningfully represent the features modelled on the sequence. This is typically the case when the best path is only marginally more probable than many other similar paths. In this case is is often preferable to create an alternative state path by assigning to each position the state with the maximal posterior probability,

$$\hat{\pi} = \arg\max_k P(\pi_i = k|\boldsymbol{x}). \tag{2.18}$$

This procedure is called posterior decoding. It should be noted that a state path obtained this way does not necessarily reflect the allowed transitions between states. Often, however, this is not required.

2.2.5 The class HMM

Often subsets of the HMM model will correspond to features on the sequence such as UTRs, coding sequence, introns, splice sites etc. Based on this the collection of states can be divided into classes, each designated by a label. By reading of the labels from the states in the state path obtained using Viterbi or posterior decoding the sequence features can be mapped to the sequence. In the case of posterior decoding the posterior probabilities of the states from each class can be summed to obtain a probability of each feature at each position in the sequence.

2.2.6 The N-best algorithm

More often than not, several states in a class HMM have the same label. If such states are not all present in all paths many different paths through the model will produce the same labelling. Since gene finding is about mapping features to the sequence, what we are interested is actually not the single most likely state path returned by Viterbi but the most likely labelling Y^* of the sequence x. This can be found by summing over all paths giving the same labelling of the sequence. An approximative N-best algorithm to do this was presented by Krogh (1997) :

Define a partial hypothesis h_i as a labelling of the sequence up to position i. Assume the probability of this hypothesis $\gamma_k(h_i)$ is known for every state k. The probability of a partial hypothesis with a given label at position i can only be non-zero for states k with that label. If the number of different labels in the class HMM is C, then the partial hypothesis $\gamma_k(h_i)$ can spawn C new partial hypotheses with each of the possible labels added to h_i: h_iY_l. The probability of these new hypotheses are found by propagating the probabilities $\gamma_k(h_i)$ forward as in the

forward algorithm,

$$\gamma_l(h_i Y_l) = \Big[\sum_k a_{kl}\gamma_k(h_i)\Big] e_l(x_{i+1}), \tag{2.19}$$

where the label of state l is called Y_l and where a_{kl} and $e_l(x_i)$ are transition and emission probabilities. In the 1-best algorithm the best partial hypothesis for each state is kept for position i in the sequence. These hypotheses are then propagated to the next position $i + 1$, and for each state the best is selected again. This continues till the end of the sequence, where the best overall hypothesis is the final answer. In summary the 1-best algorithm works like this:

1. Propagate the empty hypothesis forward to all states ($i = 1$). At this stage there is a hypotheses for each possible label. In state l the probability is $\gamma_l(h_i) = a_{0l}b_l(x_1)$, where h_i is one if the hypotheses and a_{0l} is the probability of starting in state l.

2. Propagate the hypothesis forward yielding three new hypotheses for every old one, and a probability $\gamma_l(h_i)$ given by (2.19) in each state for these hypotheses.

3. In each state, choose the hypothesis with the highest probability. Discard all hypotheses that were not chosen in any state. If you have not reached the end of the sequence go to step 2.

4. Sum the probabilities for each hypothesis over the states and save the one with the highest.

If there is a one-to-one correspondence between path and labelling 1-best is identical to Viterbi. The 1-best algorithm fails to find the correct result if the best hypothesis does not have the highest probability in any state of the model for a given position in the sequence. The complexity is much larger than for Viterbi. However, the number of active hypotheses can be reduced by discarding those with a probability below a threshold.

2.3 Parameter estimation

There are two basic tasks in constructing an HMM. The first is to build a state architecture that incorporates relevant biological knowledge by specifying states and allowed transitions between these. This is the topic of section 2.4. Given a state architecture the second task is to estimate transition and emission probabilities for the model from a set of training sequences. Depending on how restricted the architecture is different approaches are used.

2.3.1 Training sequences

Essential to any estimation approach is a set of training sequences. Typically these are nucleotide or protein sequences but may be sequences of any kind of discrete observables. Training the HMM amounts to optimising the likelihood for all training sequences $(x^1, x^2, \ldots, x^n)$ given the collection of transition and emission parameters, $\boldsymbol{\theta}$:

$$P(x^1, x^2, \ldots, x^n|\boldsymbol{\theta}) = \sum_{j=1}^{n} P(x^j|\boldsymbol{\theta}) \tag{2.20}$$

2.3.2 One path only

The most straight forward situation is when the state path for each sequence is known in advance. This is the case if only one state path through the model can account for the sequence as for very restricted architectures and/or in cases where the sequences are labelled in advance in such a way that only one state though the model is allowed. In these cases the parameters can be calculated directly by counting the number of times each transition and each emission is used. If A_{kl} is the number of times the transition from state k to state l is used, and $E_k(b)$ is the number of times state k emits observable b, then the maximum likelihood estimates of emission and transition probabilities are given by:

$$a_{kl} = \frac{A_{kl}}{\sum_i^n A_{ki}} \tag{2.21}$$

$$e_k(b) = \frac{E_k(b)}{\sum_i^n E_k(i)} \tag{2.22}$$

2.3.3 Unknown paths - Unsupervised training

More commonly there is not a unique path for each sequence. In this case the parameters cannot be calculated directly but must be optimised iteratively. The most common optimisation strategy is the Baum-Welch algorithm (Baum, 1972), a special case of the EM algorithm (Dempster et al., 1977).

Before starting the iterative procedure an initial guess of parameters can be made. Often probability is assigned uniformly to all transitions and all emissions. In some cases, however, biological prior knowledge may justify priming some parameters with higher or lower probabilities.

Each iteration is a two-step procedure. In the expectation step the current parameters are used to reconstruct all possible paths using the forward (and back-

ward) algorithm. In the maximisation step these paths are then used to reestimate transition and emission parameters.

To reestimate the transition probability a_{kl} we first calculate the cumulative posterior probability of the transition from k to l at position i:

$$P(\pi_i = k, \pi_{i+1} = l|\boldsymbol{x}, \theta) = \frac{f_k(i)a_{kl}e_l(x_{i+1})b_l(i+1)}{P(\boldsymbol{x})} \tag{2.23}$$

over all positions and all training sequences $\boldsymbol{x}^j$ to get

$$A_{kl} = \sum_j \frac{1}{P(\boldsymbol{x}^j)} \sum_i f_k^j(i)a_{kl}e_l(x_{i+1}^j)b_l^j(i+1). \tag{2.24}$$

Recall that $f_k(i)$, $b_l(i)$, and $P(\boldsymbol{x})$ are calculated by the forward and backward algorithms. The cumulative posterior probability of each emission at position, i, is summed over all positions and all training sequences j to get:

$$E_k(b) = \sum_j \frac{1}{P(\boldsymbol{x}^j)} \sum_i f_k^j(i)b_k^j(i). \tag{2.25}$$

The maximum likelihood estimate of each parameter is then obtained using (2.21) and (2.22). The procedure is halted when the improvement in likelihood falls below a predefined threshold.

2.3.4 Over-fitting and pseudo counts

Because the maximum likelihood estimate arises as relative occurrences of possible transitions and emissions they are prone to over-fitting if the number of training examples for each parameter is not sufficient. Situations may arise where rare transitions or emissions, known to exist, are assigned zero probability because they are not represented in the training examples. Such problems can be addressed by adding pseudo counts to A_{kl} and $E_k(b)$ in (2.21) and (2.22). A set of pseudo counts constitute a prior that allows for incorporation of biological information. A typical example is the addition of one pseudo count for each amino acid to ensure that rare ones not represented in the training set receive a positive probability.

Another approach to avoid over-fitting is to stop the iteration of the Baum-Welch algorithm before optimisation is complete. This is done by setting a lower threshold for the likelihood improvement of each iteration.

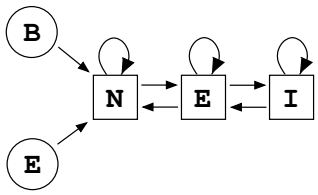


Figure 2.2. State architecture for a toy gene finder. Intergenic, exon, and intron regions are represented by N, E, and I states emitting nucleotides from characteristic distributions. The model begin and terminate in an intergenic region. The loop transitions allow any length of three types of sequence.

2.3.5 Training with labels

In a class HMM (Krogh, 1994, Krogh, 1997) the observables in the training sequences are each assigned a label corresponding to a class that represents a feature on the sequence. When training the HMM on such labelled sequences, observables with a given label can only be emitted by a state belonging to the class designated by that label. In practice this is done by assigning zero probability to the forward and backward probabilities, $f_k(i)$ and $b_k(i)$, where the label of state k does not agree with the label at position i in the training sequence. Hence, the labelling of states and observables in the training set determines which parts of the training data that are used to estimate which parts of the model. The class HMM has the advantage that it allows all model parameters to be estimated simultaneously. Alternatively each class of states would have to be trained independently. This way transitions between states of different classes that often play a crucial role in the model are not estimated.

2.4 Building a state architecture

We refer to the set of states and the allowed transitions between these as the architecture of the HMM. The major advantage of HMM modelling is that this architecture is decided on before parameter estimation. This allows for biological knowledge and constraints to be imposed on the model, so that only meaningful paths are considered in training and decoding. In addition, the number of parameters is reduced by removing unnecessary flexibility from the model. A toy architecture representing a gene is shown in Figure 2.2. As the model exemplifies, separate states for each observable are not necessary if states have transitions to themselfs or if cyclic paths are possible in the model.

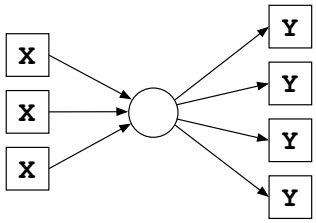


Figure 2.3. Example of the use of silent states. The silent state joins the three states on the left with the four to the right using seven transitions. Connecting all left and right states would take 12 transitions. In cases like this with many states this can save a lot transition parameters if individual connections are not necessary.

2.4.1 Tied, slave, and silent states

It is often useful to let states share transition or emission parameters. This may be done to limit the number of model parameters. The constraint thus imposed may also serve a specific modelling purpose. Sharing of parameters can be done in two ways. Tied states share parameters and contribute to estimation of these shared parameters. The second variety are states that share the parameters of another state but do not contribute the estimation of these parameters. These are states are called slave states.

In the Baum-Welch algorithm the posterior probabilities of each transition or emission from a state are summed over all observables in the training set and used for the calculation of new maximum likelihood estimates. Both tied and slave states are treated as one state in parameter estimation. For tied states the sums of posterior probabilities are added for all tied states before re-calculation of estimates. In effect, the estimated parameters becomes weighted averages over what would otherwise be individual estimates for each tied state. For slave states the sums of posterior probabilities are ignored and not included in the sums used for re-estimation.

Silent states are states that do not emit an observable. These are used to create combinatorial paths which can often diminish the number of parameters in otherwise highly connected architectures. Figure 2.3

2.4.2 Length distribution modelling

Modelling of length distributions of structure blocks is integral to modelling of genes. As outlined in Figure 1.4 most of these blocks have characteristic lengths.

The simplest length distribution is geometric. Though it often only crudely approximates the actual distribution it is widely used because of its low computational complexity. Because of its Markov property the geometric distribution can be modelled using one state with a self loop as shown in Figure 2.4. As is apparent

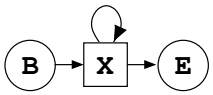


Figure 2.4. Architecture with one B state, one state with self-loop, and E state. The length of paths from the begin state to the end state will be geometrically distributed.

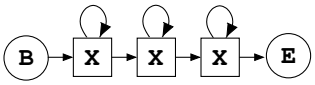


Figure 2.5. Architecture of negative binomial length distribution. If the loop probabilities are equal the shown architecture generates a negative binomial distribution.

from the simple architecture the complexity is $O(L)$ where L is sequence length.

Length modelling is most frequently done using generalised HMMs (GHMMs) that implement explicit duration (ED) modelling (Rabiner, 1989). Here variable length emissions are allowed. Each variable length gene structure block is then represented by a state that emits sequence segments with a length drawn from a parametric fitting to a the length distribution of observed lengths. GHMMs are the most widely used method for length modelling but has the drawback that the computational complexity is $O(L^2)$. Chapter 3 includes a further discussion of this topic.

As is the case with geometric distributions more complicated distributions can be modelled with state architecture. Serially connected states as shown in Figure 2.5, with identical probability self loops result in a negative binomial distribution (Durbin et al., 1998). Modelling length using state architecture requires that the model is decoded using the N-best algorithm, since the length distribution arises as a sum of probabilities of all paths through the architecture that has a given length. Distributions with more than one modality can be constructed by combining structures such as the one in Figure 2.5. These mixtures of negative binomials can be trained using Baum-Welch. Often, however, these mixtures do not offer the flexibility required for proper fitting of a distribution.

Explicit duration modelling can also be implemented as a state architecture by modelling the probability of each length explicitly. The architecture in Figure 2.6 can be trained using Baum-Welch to model any distribution between 1 and 8. As in the GHMM frame work the computational complexity is $O(L^2)$.

Acyclic discrete phase type (ADPH) distributions (Bobbio et al., 2003) offer a valuable alternative to explicit length modelling. An ADPH distribution describes the probability of moving through a directed acyclic graph with a number of phases (states) in a specified number of steps. For a special subset of sparse graphs there is a one to one correspondence between graph and ADPH distribution.

An example of such a graph and its distribution is shown in Figure 2.7. These

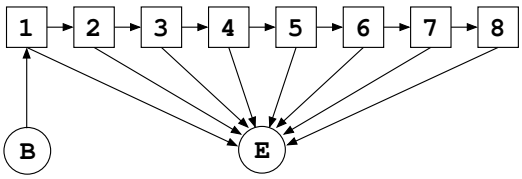


Figure 2.6. Architecture of explicit length distribution from 1 to 8.

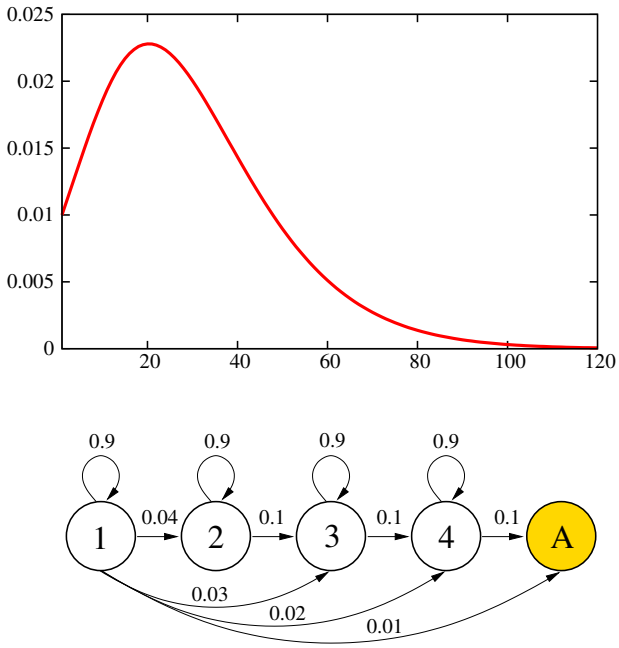


Figure 2.7. ADPH distribution with four phases and associated probability graph. The distribution describes the probability of passing from state one to the absorbing state, A, in a given number of steps. The example constitutes the special case of a mixture of an exponential and three negative binomial distributions. This arises when the loop probabilities are equal.

graphs conform to the following constraints: Phases are sequentially connected and only the first phase has edges to all other phases. All phases except the last absorbing phase has a loop edge to itself. If the probability associated with the loop edge of phase i is denoted q_i then the relation $q_1 \geq \ldots \geq q_i \geq \ldots \geq q_n$ must apply for all n phases. An ADPH distribution can be fitted to the length distribution of a sequence element and the graph underlying a fitted distribution can then be used as the HMM architecture for the gene structure element. The transition probabilities are fixed and not part of the subsequent training of the full model. As for the negative binomial the N-best algorithm must be used for decoding.

ADPH distributions allow for a very flexible modelling of length, and since the

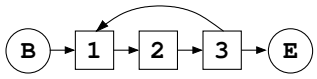


Figure 2.8. Cyclic architecture to model codons.

number of states in the corresponding architecture is a constant the computational complexity is linear in sequence length ($O(L)$). Chapter 3 includes a further discussion of length modelling using ADPH distributions. ADPH distributions are not easily fitted using Baum-Welch because this algorithm does not preserve the ordering of loop probabilities q_i. Instead separate fitting tools must be applied.

2.4.3 Inhomogeneous Markov chains

Inhomogeneous Markov chains can be implemented in HMMs using cyclic architectures. To capture codon structure in coding genes, separate modelling of the three codon positions is required. An example of an architecture to accomplish this is shown in Figure 2.8.

Chapter 3

Automatic generation of gene finders for eukaryotic species [Insert]

The contents of this chapter a manuscript submitted to BMC Bioinformatics.

Chapter 4

Ab initio prediction of alternatively spliced genes using suboptimal predictions from a hidden Markov model [Insert]

The contents of this chapter represents a paper to be submitted to Bioinformatics.

Chapter 5

Experimentally determined expression

As outlined in Chapter 1 there is an emerging picture that protein coding genes represent only a limited fraction of total transcription. The following is a summary of the recent evidence that transcription is far more widespread than previously assumed.

5.1 Tiling microarray and cDNA evidence

Tiling microarrays are a special variety of microarray that query consecutive positions on genomic sequence. The sequence of consecutive probes is called the tiling and the distance between the start of each probe is referred to as the step size or the resolution. The type of tiling arrays discussed here are the oligonucleotide arrays that use probe lengths from 25 to 60 nucleotides. These oligonucleotides are synthesised directly on slides in a density of up to 6.6 million spots on two square centimetres.

Tiling arrays have two major advantages for the discovery of novel transcription. Firstly, tiling arrays are not biased towards previous annotation or prediction. Secondly, the sensitivity of microarrays allow rare transcripts to be detected. Tiling arrays have been used for refinement of exon borders of gene predictions validated by exon arrays (Shoemaker et al., 2001). The unbiased nature of tiling arrays have made them increasingly popular in the search for non-coding transcription.

Tiling arrays of the non-repetitive parts of the Human chromosomes 21 and 22 has been done using 25-nucleotide probes for every 35 base pairs on average (Kapranov et al., 2002). Labelled double-stranded cDNA made from cytosolic poly-adenylated RNA from 11 different cell lines were hybridised to the arrays. Of probes that were positive in at least one cell line 94% did not overlap annotated exon positions. Further analysis of these data report that at least half the detected transcribed fragments do not overlap known ESTs of mRNAs (Kampa et al., 2004).

Tiling array experiments on the non-repetitive parts of chromosomes 20 and 22 with 60-nucleotide probes and 30 base pair steps reported that 47% of positive probes did not overlap known exons. Of these 22% were in introns and 25% were in intergenic regions (Schadt et al., 2004).

An analysis of the non-repetitive part of the human genome using strand-specific 36-nucleotide probes spaced 46 bases on average has yielded qualitatively similar results (Bertone et al., 2004). 41% of the novel transcribed sequences identified by this analysis overlaps with tiling array experiments by Kapranov *et al.*.

Recently, tiling arrays with 25-nucleotide probes for every five nucleotides has been constructed for Human chromosomes 6, 7, 13, 14, 19, 20, 21, 22, X, and Y (Cheng et al., 2005). This analysis used cytosolic poly-adenylated RNA from eight cell lines. Ten percent of interrogated nucleotides indicate transcription in at least one of the eight cell lines. Overall 32% of novel transcripts are found in intergenic regions, whereas 10% is found in introns. For one cell line, HepG2, poly-adenylated and non-poly-adenylated RNA fractions from both cytosol and nucleus were used. Of all transcribed sequence identified 19%, 44%, and 37% where observed to be poly-adenylated, non-poly-adenylated, or bimorphic respectively. Half of all transcribed sequences are only found in the nucleus. Of these the majority are un-annotated.

Tiling array studies have also been done for important model organisms such as *A. thaliana* and *D. malanogaster*. For the Fly, tiling arrays with 36-nucleotide probes have been combined with exon and exon-junction arrays to create a gene expression map of the euchromatic genome (Stolc et al., 2004). This study reports expression for 41% of probes in intronic and intergenic sequences. A combined cDNA and tiling array analysis of the *A. thaliana* genome also reports a large amount of un-annotated transcription from poly-adenylated RNA (Yamada et al., 2003). In addition, some amount of anti-sense transcription is reported for 30% of the annotated genes.

Recently, a large scale analysis of full-length Mouse cDNA matched to the genome has revealed 181,047 transcripts varying in promoter usage, splicing and poly-adenylation (Carninci et al., 2005). This data supports and multiplies the findings using tiling arrays. 16,247 novel protein coding transcripts has been identified among which are 5154 previously unknown genes. Based on this data the number of transcripts is at least one magnitude larger than the current estimate of 20,000-25,000 protein-coding genes. The definition of a gene as a transcriptional unit of overlapping mRNAs transcribed in the same direction seems inadequate in the light of these findings. The presented data shows that unrelated and differently annotated transcriptional units fuse into transcriptional frameworks. These are often transcriptional units joined end to end with transcription initiation in 3'UTRs. To further complicate matters these frameworks cluster into transcrip-

tional forests, ungapped regions of the genome transcribed on either strand. These forests comprise a staggering 62% of the Mouse genome. These results also indicate that anti-sense transcription is far more widespread than anticipated. More than 72% of all transcriptional units show evidence of anti-sense transcription (Katayama et al., 2005).

These tiling arrays as well as cDNA results support the consensus that extensive expression of intergenic and intronic sequences occurs in both the major evolutionary lineages of animals (deuterostomes and protostomes) and in plants.

5.2 The nature of novel transcription

The nature of the vast amount of novel transcription has been debated. In the following I will summarise some of the suggested explanations (See Johnson *et. al* (2005) for a more thorough review).

The varying estimates in the number of Human protein coding genes leaves some room for ascribing novel transcription to unknown genes. However, the number of annotated genes have followed a decreasing trend as the quality of the genome has improved. Hence, the fraction of unknown transcription attributable to unknown protein-coding genes is expected to be rather small.

One of the most interesting and most frequently proposed explanations is that much of the novel transcription is non-coding RNA (ncRNA). A large set of small ncRNA families are already known: E.g. miRNA, snRNA, snoRNA, tRNA, and siRNA. These may just be the beginning of a long list waiting to be discovered. Many of the known ncRNAs, however, are located in the nucleus and are thus not detected by poly-A fractions on tiling arrays. More likely, a large fraction of unknown transcription may be non-coding poly-adenylated RNAs. More than 40% of a non-redundant set of 33,000 mouse cDNAs do not have an open reading frame larger than 300 nucleotides (Okazaki et al., 2002). Similar results have been obtained using Human cDNA (Ota et al., 2004). These RNAs may be functional in their own right or may be vehicle structures for small separately functional ncRNAs.

The extensive anti-sense transcription shown in both Mouse and *Arabidopsis* may explain a large part of novel transcription. Unfortunately, the functional role of anti-sense transcription is still unclear.

Extension of known genes and alternative transcripts of known genes may contribute significantly to the novel transcription. In the tiling array study of Human chromosomes 20 and 22 roughly half of the positive probes were located in introns of known genes. This suggests that many, possibly rare, transcripts are yet to be found.

Only a limited fraction of the novel transcription may be functional. It is

expected that some degree of leaky transcription is taking place. In addition, aberrant transcription start and termination sites may be used. These explanations are consistent with the observation that many novel transcripts occur at low levels. In addition, selectively "neutral" nonsense transcription may exist because the cost of transcribing it may be outweighed by the cost of suppressing such transcription. This is in line with the fact that only 7-20% of the novel transcribed regions on human chromosomes 21 and 22 are conserved in the mouse genome (Rinn et al., 2003, Kampa et al., 2004). Though the vast majority of such leaky transcription would not be functional it may serve as a source of recruitment of functional ncRNA.

Since the large amounts of novel transcription has been observed by many different groups using different approaches these novel findings can hardly be attributed to experimental artefacts alone. It is likely, however, that some issues for the tiling array approaches such as genomic contamination of RNA samples, contamination with un-spliced mRNA, unintended double-stranded labelling, and cross-hybridisation may result in a considerable amount of artefacts.

5.3 Analysis of tiling array data

The basic tasks associated with the analysis of tiling microarray data are twofold. First, raw fluorescence scores must be normalised for both chip and sequence specific signals. Secondly, genomically consecutive probes must be categorised into regions of expression or non-expression.

One approach to normalisation is modelling of the contribution of each nucleotide based on its position in the probe (Naef and Magnasco, 2003, Wu et al., 2004, Wu and Irizarry, 2005). Using negative controls the signal attributable to probe sequence alone can be subtracted. By modelling probes for each chip it is also possible to normalise for chip-specific background signal.

As an alternative approach, probe and array normalisation can be addressed by, in addition to perfect match (PM) probes, also designing mismatch (MM) probes, where the complement of the central base in the probe is used instead of the actual base. The mismatch probe is expected to capture much of the probe sequence and array-specific contributions to the signal as well as problematic probe eccentricities such as secondary-structure and cross-hybridisation effects while capturing little real signal. The difference in fluorescence score for the PM and MM probe score is then used to estimate expression signal from the queried genomic position. Though there is room for improvement, the PM-MM normalisation procedure does help (Royce et al., 2005).

Several approaches for categorisation of probes have been applied. The following three are representative. Kampa *et al.* (2004) have used a smoothening

approach in which the median of pairwise averages among the PM-MM scores is calculated within a window of 100 nucleotides centred at the normalised probe. Probes with a normalised score above a threshold determined by spiked in bacterial negative controls is considered expressed. This procedure serves to smooth the signal and to take care of outliers. Sets of consecutive expressed probes are constructed by merging all positive probes within a cutoff distance, disregarding sets of length below a threshold.

Bertone *et al.* (2004) have identified regions of transcription based on p values calculated in windows of consecutive probes as the probability of the observed number of scores above and below the global array median score. Overlapping windows considered expressed based on a p value cutoff are joined to form predictions of expression.

A third approach used by Schadt *et al.* (2004) analyses multiple replicates (tissues) simultaneously by performing a principal component analysis on all replicates for probes scores in a sliding window of 500 probes. The first component agrees with average probe intensity across tissues and the second dimension correlates with variation across tissues. Probes with small second component values are discarded. The Mehalanobis distance (MD) of each probe to the centre of this two-dimensional PCA space is calculated. Probes with a significantly larger MD than expected from an intron distribution is then considered expressed. Expressed regions are constructed by grouping positive probes using a one-dimensional hierarchical clustering of the MDs of expressed probes. This procedure clusters probes exhibiting similar intensities as would be expected from probes from the same transcript.

A major issue when making inference on tiling array data is the lack of proper positive and negative controls. The noise level in tiling array data is very high especially for arrays using short probes. Add to this that the number of truly positive probes is relatively small. This makes the interpretation of data challenging but interesting. The amount of tiling array data grows rapidly. To further elucidate the nature and amount of "dark matter" transcription in the genome new statistical procedures and good models of inference are called for.

Chapter 6

A probabilistic approach for determining transcripts from genomic tiling microarrays [Insert]

The contents of this chapter is a manuscript submitted to BMC Bioinformatics. Supplementary information is attached at the end.

Chapter 7

Future directions

The approach to length modelling in Agene is powerful but suffer from the drawback that length distributions must be trained separately and then plugged into the full model before this is trained. This limits the use of length modelling to subsequences that are annotated in the training set, exemplified in Figure 1.4. In some situations, however, the subsequences representing characteristic length cannot be extracted from the training set. An example is the distance from the branch point to the acceptor site. This length distribution can not be trained in advance using acyclic discrete phase type (ADPH) distributions (see Chapter 3) if the branch point is not annotated. In some cases the branch point constitutes a well defined consensus sequence. By priming a separate weight array matrix structure in the model with this consensus sequence the distance between predictions of branch points and acceptor sites can be captured by a length distribution if this is trained simultaneously as part of the full model. The limitations of this approach is that so far only geometric and negative binomial mixtures can be trained using Baum-Welsh. It would be interesting to investigate the possibilities of fitting more detailed distributions using Baum-Welsh.

Concerning prediction of multiple transcript genes, the N-best algorithm has inherent problems that hampers its ability to predict alternative exons as part of a reasonable number of alternative labellings (see Chapter 4). Development of an algorithm that returns a more representative set of alternative exons should be considered.

HMMs are stochastic regular grammars, meaning that they can only model non-overlapping non-nested dependencies of the sequence. A large part of the information in gene structures conform to these constraints which is why HMMs have been so successful in gene prediction. *Ab initio* HMM gene finders, however, do seem to have reached an upper limit of performance that cannot be exceeded without including new types of dependencies. A more elaborate modelling of genes is needed if we are to capture the dependencies that govern the large number of possible splice forms. It seems that there are long range dependencies between

the splice sites. Strong splice sites can ”rescue” week splice sites at the other end of an exon or intron (Berget, 1995). Such intron and exon bridging interactions can be modelled by stochastic context free grammars (SCFGs).

It is likely that the length of a gene structure element is dependent on the length of an adjacent element. A way to model dependent length distributions perhaps using SCFGs would be interesting to pursue.

With the emerging picture that non-coding genes are much more prevalent than anticipated, it would also be interesting to remove modelling of the coding region from Agene and thus turn it into a non-coding gene finder. The performance of such a gene finder is not expected to be high but may return interesting predictions that can be filtered by other means. Prediction of expression returned by ExpressHMM can be used as supporting evidence by such a predictor. This could result in a predictor that fits expression evidence to consistent gene structures. Also, expression evidence may aid the correspondence between suboptimal labellings from Agene and true alternative transcripts.

We intend to improve ExpressHMM with respect to modelling of gaps in the tiling. The current modelling of gaps does not take into account that the label (expressed or non-expressed) is not likely to change over short gaps. With an extended gap modelling the probability of returning to the label before the gap can be trained for gaps up to a sensible maximum length.

Bibliography

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P., 2002. *Molecular biology of the cell*. Garland Science, 4. edition.

Baum, L. E., 1972. An equality and associated maximization technique in statistical estimation for probabilistic functions of markov processes. *Inequalities*, **3**:1–8.

Berget, S. M., 1995. Exon recognition in vertebrate splicing. *J Biol Chem*, **270**(6):2411–2414.

Bertone, P., Stolc, V., Royce, T. E., Rozowsky, J. S., Urban, A. E., Zhu, X., Rinn, J. L., Tongprasit, W., Samanta, M., Weissman, S., *et al.*, 2004. Global identification of human transcribed sequences with genome tiling arrays. *Science*, **306**(5705):2242–2246.

Bobbio, A., Horvath, A., Scarpa, and Telek, M., 2003. Acyclic discrete phase type distributions: Properties and a parameter estimation algorithm. *Performance Evaluation*, **54**:1–32.

Burset, M., Seledtsov, I. A., and Solovyev, V. V., 2000. Analysis of canonical and non-canonical splice sites in mammalian genomes. *Nucleic Acids Res*, **28**(21):4364–4375.

Carninci, P., Kasukawa, T., Katayama, S., Gough, J., Frith, M. C., Maeda, N., Oyama, R., Ravasi, T., Lenhard, B., Wells, C., *et al.*, 2005. The transcriptional landscape of the mammalian genome. *Science*, **309**(5740):1559–1563.

Cheng, J., Kapranov, P., Drenkow, J., Dike, S., Brubaker, S., Patel, S., Long, J., Stern, D., Tammana, H., Helt, G., *et al.*, 2005. Transcriptional maps of 10 human chromosomes at 5-nucleotide resolution. *Science*, .

Curwen, V., Eyras, E., Andrews, T. D., Clarke, L., Mongin, E., Searle, S. M., and Clamp, M., 2004. The ensembl automatic gene annotation system. *Genome Res*, **14**(5):942–950.

Dempster, A. P., Laird, N. M., and Rubin, D. B., 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society B*, **39**:1–38.

Durbin, R., Eddy, S., Krogh, A., and Mitchison, G., 1998. *Biological Sequence Analysis*. Cambridge Universiy Press, 1. edition.

Johnson, J. M., Castle, J., Garrett-Engele, P., Kan, Z., Loerch, P. M., Armour, C. D., Santos, R., Schadt, E. E., Stoughton, R., and Shoemaker, D. D., *et al.*, 2003. Genome-wide survey of human alternative pre-mrna splicing with exon junction microarrays. *Science*, **302**(5653):2141–2144.

Johnson, J. M., Edwards, S., Shoemaker, D., and Schadt, E. E., 2005. Dark matter in the genome: evidence of widespread transcription detected by microarray tiling experiments. *Trends Genet*, **21**(2):93–102.

Kampa, D., Cheng, J., Kapranov, P., Yamanaka, M., Brubaker, S., Cawley, S., Drenkow, J., Piccolboni, A., Bekiranov, S., Helt, G., *et al.*, 2004. Novel rnas identified from an in-depth analysis of the transcriptome of human chromosomes 21 and 22. *Genome Res*, **14**(3):331–342.

Kapranov, P., Cawley, S. E., Drenkow, J., Bekiranov, S., Strausberg, R. L., Fodor, S. P., and Gingeras, T. R., 2002. Large-scale transcriptional activity in chromosomes 21 and 22. *Science*, **296**(5569):916–919.

Katayama, S., Tomaru, Y., Kasukawa, T., Waki, K., Nakanishi, M., Nakamura, M., Nishida, H., Yap, C. C., Suzuki, M., Kawai, J., *et al.*, 2005. Antisense transcription in the mammalian transcriptome. *Science*, **309**(5740):1564–1566.

Krogh, A., 1994. Hidden markov models for labelled sequences. *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, :140–144.

Krogh, A., 1997. Two methods for improving performance of an hmm and their application for gene finding. *Proc Int Conf Intell Syst Mol Biol*, **5**:179–186.

Naef, F. and Magnasco, M. O., 2003. Solving the riddle of the bright mismatches: labeling and effective binding in oligonucleotide arrays. *Phys Rev E Stat Nonlin Soft Matter Phys*, **68**(1 Pt 1):011906.

Okazaki, Y., Furuno, M., Kasukawa, T., Adachi, J., Bono, H., Kondo, S., Nikaido, I., Osato, N., Saito, R., Suzuki, H., *et al.*, 2002. Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cdnas. *Nature*, **420**(6915):563–573.

Ota, T., Suzuki, Y., Nishikawa, T., Otsuki, T., Sugiyama, T., Irie, R., Wakamatsu, A., Hayashi, K., Sato, H., Nagai, K., *et al.*, 2004. Complete sequencing and characterization of 21,243 full-length human cdnas. *Nat Genet*, **36**(1):40–45.

Rabiner, L. R., 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, **77**(2):257–286.

Rinn, J. L., Euskirchen, G., Bertone, P., Martone, R., Luscombe, N. M., Hartman, S., Harrison, P. M., Nelson, F. K., Miller, P., Gerstein, M., *et al.*, 2003. The transcriptional activity of human chromosome 22. *Genes Dev*, **17**(4):529–540.

Royce, T. E., Rozowsky, J. S., Bertone, P., Samanta, M., Stolc, V., Weissman, S., Snyder, M., and Gerstein, M., 2005. Issues in the analysis of oligonucleotide tiling microarrays for transcript mapping. *Trends Genet*, **21**(8):466–475.

Schadt, E. E., Edwards, S. W., GuhaThakurta, D., Holder, D., Ying, L., Svetnik, V., Leonardson, A., Hart, K. W., Russell, A., Li, G., *et al.*, 2004. A comprehensive transcript index of the human genome generated using microarrays and computational approaches. *Genome Biol*, **5**(10):R73.

Schneider, T. D. and Stephens, R. M., 1990. Sequence logos: a new way to display consensus sequences. *Nucleic Acids Res*, **18**(20):6097–6100.

Shoemaker, D. D., Schadt, E. E., Armour, C. D., He, Y. D., Garrett-Engele, P., McDonagh, P. D., Loerch, P. M., Leonardson, A., Lum, P. Y., Cavet, G., *et al.*, 2001. Experimental annotation of the human genome using microarray technology. *Nature*, **409**(6822):922–927.

Stolc, V., Gauhar, Z., Mason, C., Halasz, G., van Batenburg, M. F., Rifkin, S. A., Hua, S., Herreman, T., Tongprasit, W., Barbano, P. E., *et al.*, 2004. A gene expression map for the euchromatic genome of drosophila melanogaster. *Science*, **306**(5696):655–660.

Wu, Z. and Irizarry, R. A., 2005. Stochastic models inspired by hybridization theory for short oligonucleotide arrays. *J Comput Biol*, **12**(6):882–893.

Wu, Z., R.A., I., Gentleman, R., Martinez-Murillo, F., and Spencer, F., 2004. A model-based background adjustment for oligonucleotide expression arrays. *Journal of the American Statistical Association*, **99**(468):909–917.

Yamada, K., Lim, J., Dale, J. M., Chen, H., Shinn, P., Palm, C. J., Southwick, A. M., Wu, H. C., Kim, C., Nguyen, M., *et al.*, 2003. Empirical analysis of transcriptional activity in the arabidopsis genome. *Science*, **302**(5646):842–846.