This chapter is based on the following sources, which are all recommended reading:


12.1 Introduction

In the 1960s, it was discovered that a gene and its protein product are colinear structures with a direct correlation between the triplets of nucleotides in the gene and the amino acids in the protein.

It soon became clear that genes can be difficult to determine, due to the existence of overlapping genes, and genes within genes etc.

Moreover, the paradox arose that the genome size of many eukaryotes does not correspond to “genetic complexity”, for example, the salamander genome is 10 times the size of that of human.

In 1977, the surprising discovery of “split” genes was made: genes that consist of multiple pieces of coding DNA called exons, separated by stretches of non-coding DNA called introns.

Given a string of genomic DNA. The gene prediction problem is to reliably predict all genes contained in the sequence.

Why is gene prediction so hard?

- The Genome of many eukaryotes contain only relatively few genes (Human genome 3%).
- Many false splice sites & other signals
- Very short exons (3bp), especially initial
- Many very very long introns
12.2 Three approaches to gene finding

One can distinguish between three types of approaches:

- **Statistical or ab initio methods.** These methods attempt to predict genes based on statistical properties of the given DNA sequence. Programs are e.g. Genscan, GeneID, GENIE and FGENEH.

- **Homology methods.** The given DNA sequence is compared with known protein structures. Programs are e.g. TBLASTN or TBLASTX, Procrustes and GeneWise.

- **Comparative methods.** The given DNA string is compared with a similar DNA string from a different species at the appropriate evolutionary distance and genes are predicted in both sequences based on the assumption that exons will be well conserved, whereas introns will not. Programs are e.g. CEM (conserved exon method) and Twinscan.

12.2.1 Ab initio methods

*Ab initio* gene prediction methods use statistical properties of the different components of such a gene model to predict genes in unannotated DNA. For example, for the bases around the transcription start site we may have the following observed frequencies (given by this *position specific weight matrix* (PSWM)):

<table>
<thead>
<tr>
<th>Pos.</th>
<th>-8</th>
<th>-7</th>
<th>-6</th>
<th>-5</th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
<th>+7</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>.16</td>
<td>.29</td>
<td>.20</td>
<td>.25</td>
<td>.22</td>
<td>.66</td>
<td>.27</td>
<td>.15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>.28</td>
<td>.24</td>
<td>.11</td>
<td>.26</td>
</tr>
<tr>
<td>C</td>
<td>.48</td>
<td>.31</td>
<td>.21</td>
<td>.33</td>
<td>.56</td>
<td>.05</td>
<td>.50</td>
<td>.58</td>
<td>0</td>
<td>0</td>
<td>.16</td>
<td>.29</td>
<td>.24</td>
<td>.40</td>
<td></td>
</tr>
<tr>
<td>G</td>
<td>.18</td>
<td>.16</td>
<td>.46</td>
<td>.21</td>
<td>.17</td>
<td>.27</td>
<td>.12</td>
<td>.22</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.48</td>
<td>.20</td>
<td>.45</td>
<td>.21</td>
</tr>
<tr>
<td>T</td>
<td>.19</td>
<td>.24</td>
<td>.14</td>
<td>.21</td>
<td>.06</td>
<td>.02</td>
<td>.11</td>
<td>.05</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>.09</td>
<td>.26</td>
<td>.21</td>
<td>.21</td>
</tr>
</tbody>
</table>

This can then be used together in a log-likelihood scoring model in order to distinguish certain recognition sites (such as transcription start sites, or promoter regions) from non-recognition sites.

12.3 Gene prediction in prokaryotes

In prokaryotic cells most of the DNA sequence is coding for genes/proteins. In comparison, almost 70% of the genome of *H. influenzae* is coding, while only about 3-5% of the human genome codes for proteins. Also the gene structure is quite different, e.g. there are no introns in the coding regions in prokaryotic genes.
During the transcription process the RNA polymerase copies one DNA strand into the mRNA. The polymerase attaches at the transcription start site (TSS) to the DNA from which the transcription is started and stops the transcription when the signal for the transcription end is reached. The signal or pattern for the transcription start is (mostly) upstream of the start codon, and the one for the transcription end downstream of the stop codon. Thus there are regions at both ends of the coding region that become transcribed but not translated, thus they are called untranslated regions. Again upstream of the TSS is a regulatory region that contains the promoters.

12.3.1 ORF prediction

The simplest way to detect potential coding regions is to look at Open Reading Frames (ORFs). An ORF is a sequence of codons in DNA that starts with a Start codon (ATG), ends with a Stop codon (TAA, TAG or TGA) and has no other (in-frame) stop codons inside.

**Evaluate lengths of ORFs:**
The average distance between stop codons in “random” DNA is \( \frac{64}{3} \approx 21 \), much smaller than the number of codons in an average protein (\( \approx 300 \)).

An algorithm would then take a given DNA sequence and search within each of the possible reading frames stop codons and its corresponding start codon. For each such potential ORF determine the length and evaluate that.

**Evaluate codon usage:**
Here we use the fact that codon usage in coding regions differs substantially from that in non-coding regions. A number of these measures have been proposed, such as codon usage or hexamer counts. The codon usage of a string of DNA is given by a 64-component vector that counts how many times each codon is present in the string.

Example Codon Preference in E. Coli:

<table>
<thead>
<tr>
<th>AA</th>
<th>Codon</th>
<th>/ 1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gly</td>
<td>GGG</td>
<td>1.89</td>
</tr>
<tr>
<td>Gly</td>
<td>GGA</td>
<td>0.44</td>
</tr>
<tr>
<td>Gly</td>
<td>GGU</td>
<td>52.99</td>
</tr>
<tr>
<td>Gly</td>
<td>GGC</td>
<td>34.55</td>
</tr>
<tr>
<td>Glu</td>
<td>GAG</td>
<td>15.68</td>
</tr>
<tr>
<td>Glu</td>
<td>GAA</td>
<td>57.20</td>
</tr>
<tr>
<td>Asp</td>
<td>GAU</td>
<td>21.63</td>
</tr>
<tr>
<td>Asp</td>
<td>GAC</td>
<td>43.26</td>
</tr>
</tbody>
</table>

The in-phase hexamer feature measures the frequency of occurrence of oligonucleotides of length six in a specific reading frame. In a study by Fickett and Tung (1992) it has been shown to be the most effective. Hexamer counts are mostly modeled as fifth-order Hidden Markov Models.

Fifth-order: \( P(x_n = s | \cap_{j<n} x_j) = P(x_n = s | x_{n-1} x_{n-2} x_{n-3} x_{n-4} x_{n-5}) \)
For each reading frame a codon preference statistic $p_c$ at each position is computed. The statistic is calculated over a window of length $l_w$ ($l_w$ is usually between 25 and 50), where the window is moved along the sequence in increments of three bases, maintaining the reading frame. The magnitude of the codon preference statistic is a measure of the likeness of a particular window of codons to a predetermined preferred usage.

The statistic is based on the concept of synonymous codons. Synonymous codons are those codons specifying the same amino acid.

For example, the bases Leucine, Alanine and Tryptophan are coded by 6, 4 and 1 different codons respectively. So in a uniformly random DNA sequence, the bases should occur in the ratio 6:4:1. But in a protein they occur in a different ratio – eg. 6.9:6.5:1. Therefore coding DNA is not random.

A codon parameter is calculated for each codon $c$ in the reading frame based on the codon’s frequency of occurrence ($f_c$) and the total number of occurrences of its synonymous family ($F_c$) in the codon frequency table, and the calculated occurrences of the codon ($r_c$) and its synonymous family ($R_c$) in a random sequence with the same base composition as the sequence being analyzed. The codon preference statistic for each codon $c$, $p_c$, is then given by:

$$p_c = \frac{f_c}{F_c} \cdot \frac{r_c}{R_c}$$  \hspace{1cm} (12.1)

A $p_c$ value of 1.0 indicates that a codon is used equally in the random sequence and the codon frequency table. Values greater than 1.0 indicate the codon is present at higher than the random frequency in the codon frequency table, and values less than 1 indicate a codon is present at less than the random frequency in the codon frequency table.

The probability of the sequence in the window $w$ is then

$$P(w) = \prod_{i=1}^{l_w} p_{c_i}$$  \hspace{1cm} (12.2)

Again a log-based score is used and the codon preference statistic for each window ($P_w$) is given by

$$P_w = e^{\left(\sum_{i=1}^{l_w} \log p_{c_i}\right)/l_w}$$  \hspace{1cm} (12.3)

Since the statistic is strongly affected by codons whose occurrence is zero in the codon frequency table, these codons are assigned an occurrence of 1. This is equivalent to saying that a zero value in the table doesn’t mean that these codons are never seen, it only means that they haven’t been seen in $F_c$ observations, and that the upper bound on their occurrence as a fraction of their synonymous family is $1/F_c$.

An example

---

1based on Gribskov et al., Nucl. Acids Res. 12; 539-549, 1984
ORF prediction using Markov models and HMMs:
There are many more ORFs than real genes. For example, the *E. coli* genome contains about 6600 ORFs but only about 4400 real genes. Here we want to briefly mention how a Markov model and an HMM can be used to distinguish between non-coding ORFs and real genes.

In principle, a model as described for distinguishing CpG-islands can be set up, which can distinguish between coding and non-coding ORFs. One possibility is to model the DNA sequences as 64-states (alphabet thus consists now of 64 letters) Markov chains of codons. The transition probabilities are then the probabilities that a certain codon is followed by another codon in a coding ORF. We can thus compute the log-odds scores. Non-coding ORFs have log-odds distribution centered around zero (as can be seen from the following figure), from which one concludes that codon usage in such regions is essentially random.

Krogh et al.\(^2\) have set up an HMM to model prokaryotic genes, with the aim to combine all signal-based methods for locating a gene within one framework.

The architecture models the structure of a typical (prokaryotic) gene. Since this is similar to the idea

\(^2\)Krogh, Mian & Haussler, NAR 22, 4768-4778, 1994
of GenScan, we will skip a detailed description here and refer the reader to the paper.

### 12.4 Eukaryotic gene structure

The gene structure and the expression mechanism in eukaryotic cells is much more complicated than in prokaryotes. Here genes are not organized as operons. Genes involved in the same metabolic pathway often lie scattered across several chromosomes. Thus, most genes have their own promoter and transcription start sites. In typical eukaryotes the coding region is not continuous, but is composed of alternating stretches of exons and introns. In the initial transcription phase both exons and introns are transcribed into a pre-mRNA, and then in a splicing process intron sequences are excised and removed from the pre-mRNA.

For our purposes, a eukaryotic gene has the following structure:

- **Initial exon** occurs at the very 5’ end of the gene
- **Internal exons**
- **Terminal exons**
- **Single-exon genes**, i.e. genes without introns.

Since in the exon-intron junctions there is a large similarity to the consensus sequence, this naturally leads to the idea using an algorithm based on position specific weight matrices. However, this is far too simple, since it does not use all the information encoded in a gene. Thus more integrated approaches are sought. This naturally leads us to Hidden Markov Models.

### 12.5 A simple HMM for gene detection

A simple HMM $M$ for identifying a gene is shown below. Here the states are exons and introns and with a certain probability $p$ and/or $q$ the process stays within the exon and/or intron state, and thus with probability $1 - p$ and/or $1 - q$ changes between the states occur. Then the probability that an exon has length $k$ is

$$
\mathbb{P}(\text{exon of length } k \mid M) = p^k \cdot (1 - p)
$$

However, see also M.Q. Zhang (Nature Rev. Gen. 3, 698-709, 2002), who proposed a more detailed classification scheme for exons.
which has a geometric distribution.

\[
\begin{array}{c}
\text{Exon} \\
\begin{array}{c}
P(A)=0.2 \\
P(C)=0.3 \\
P(G)=0.3 \\
P(T)=0.2 \\
\end{array}
\end{array}
\quad
\begin{array}{c}
\text{Intron} \\
\begin{array}{c}
P(A)=0.25 \\
P(C)=0.25 \\
P(G)=0.25 \\
P(T)=0.25 \\
\end{array}
\end{array}
\]

Unfortunately from the following figures we see that the exon length does not have a geometric distribution. If an exon is too short (under 50bp), the spliceosome (enzyme that performs the splicing) has not enough room. On the other hand, exons that are longer than 300 bp are difficult to locate. Therefore we need other models that can model biological exon lengths.

Typical numbers for vertebrates: mean gene length \(\approx 30\text{kb}\), mean coding region length \(\approx 1 - 2\text{kb}\).

Empirical length distributions for introns and exons:

\[
\begin{array}{c}
\text{Introns} \\
\text{Initial exons} \\
\text{Internal exons} \\
\text{Terminal exons}
\end{array}
\]

12.6 GENSCAN’s model

We are going to discuss the popular program GENSCAN in detail, which is an explicit state duration HMM or semi-Markov model. This is an HMM in which, additionally, a duration period is explicitly modeled for each state, using a probability distribution.

The model consists of 27 states and 46 transition probabilities:
The states correspond to the functional units of a eukaryotic gene, with state transitions chosen to allow only a biologically sensible order of these units.

Starting at the state representing intergenic region, we will now see how the states are modeled. We first notice that the model includes forward and reverse strand genes, and one is basically a mirror of the other one, therefore we will look only at the forward strand model.

1. **Intergenic Region:** The intergenic region is modeled as a homogeneous fifth-order HMM, i.e., the generation of any nucleotide depends on the previous five nucleotides generated. This reflects the fact that nucleotides tend to occur in hexamers with different frequencies in coding and noncoding regions.

2. **Promoter:** The intergenic region is followed by the promoter ($P^+$), modeled as a PSWM.
3. **5' UTR:** The next state is the 5' UTR ($F^+$), also modeled as a fifth-order HMM.

4. **Exons:** Then either an initial exon follows ($E_{init}^+$), or a single exon ($E_{sngl}^+$).

5. The initial exon passes into an intron ($I_j^+, j = 0, 1, 2$) that is either in phase 0, 1 or 2, depending on the position of the exon-intron boundary with respect to the reading frame. Like the intergenic region and the 5' UTR, introns are modeled using the fifth-order HMM. Introns are always followed by an exon, either by an internal one ($E_j^+, j = 0, 1, 2$) or by a terminal one ($E_{term}^+$). Clearly, a phase $j$ intron can only be followed by the same phase for the internal exon.

$I_0$ falls between codons, $I_1$ falls after the 1st position, $I_2$ falls after the 2nd position. Then $E_0$ starts with a whole codon, $E_1$ starts with the 2nd position, $E_2$ starts with the 3rd position so that a consistent reading frame is forced.

6. **3' UTR:** Terminal exons are followed by the 3' UTR ($T^+$), again modeled by a fifth-order HMM.

7. **Poly-A:** Finally the model passes into the poly-A state ($A^+$), which eventually passes into the next intergenic region.

### 12.6.1 GENSCAN’s sequence generation

Now we will consider how the model generates DNA sequences of a predefined length $L$ by moving through a sequence of states. The model is thought of generating a parse $\phi$, consisting of:

- a sequence of states $q = (q_1, q_2, \ldots, q_n)$, and
- an associated sequence of durations $d = (d_1, d_2, \ldots, d_n)$,

which, using probabilistic models for each of the state types, generates a DNA sequence $S$ of length $L = \sum_{i=1}^n d_i$.

The generation of a parse of a given sequence length $L$ proceeds as follows:

1. An initial state $q_1$ is chosen according to an initial distribution $\pi$ on the states, i.e. $\pi_i = P(q_1 = Q^{(i)})$, where $Q^{(i)} (i = 1, \ldots, 27)$ is an indexing of the states of the model.

2. A state duration or length $d_1$ is generated conditional on the value of $q_1 = Q^{(i)}$ from the duration distribution $f_{Q^{(i)}}$.

3. A sequence segment $s_1$ of length $d_1$ is generated, conditional on $d_1$ and $q_1$, according to an appropriate sequence-generating model for state type $q_1$.

4. The subsequent state $q_2$ is generated, conditional on the value of $q_1$, from the (first-order Markov) state transition matrix $P$, i.e. $p_{i,j} = P(q_{k+1} = Q^{(j)} | q_k = Q^{(i)})$.

This process is repeated until the sum $\sum_{i=1}^n d_i$ of the state durations first equals or exceeds $L$, at which point the last state duration is appropriately truncated, the final stretch of sequence is generated and the process stops.

The resulting sequence is simply the concatenation of the sequence segments, $S = s_1 s_2 \ldots s_n$.

In addition to its topology involving the 27 states and 46 transitions depicted above, the model $M$ has four main components:

- a vector of initial probabilities $\pi$,
- a matrix of state transition probabilities $P$, 

• a set of length distributions \( f \), and
• a set of sequence generating models \( G \).

This type of HMM is also called a generalized hidden Markov model, since the output of a state may be not a single symbol, but can be string of finite length.

Note that the generated sequence is not restricted to correspond to a single gene, but could represent multiple genes, in both strands, or none.

### 12.7 GENSCAN optimizes a probability model

Given a DNA sequence, find the annotation that maximizes the joint probability

\[
\begin{align*}
5' & \text{TTTACAGGACCATGCTACCGGTGGATT} \ 3' \\
& \text{FFFFFFFFFFFFFFFFFFEIIIIIIIIIIII}
\end{align*}
\]

\( F = 5' \text{ Untranslated}; \ E = \text{Exon}; \ I = \text{Intron} \)

### 12.8 Maximum likelihood prediction

Given such a model \( M \). The aim is to compute which of the many possible gene structures is most likely. For a fixed sequence length \( L \), consider

\[
\Omega = \Phi_L \times S_L,
\]

where \( \Phi_L \) is the set of all possible parses of \( M \) of length \( L \) and \( S_L \) is the set of all possible sequences of length \( L \).

The model \( M \) assigns a probability density to each point (parse/sequence pair) in \( \Omega \). Thus, for a given sequence \( S \in S_L \), a conditional probability of a particular parse \( \phi \in \Phi_L \) according to Bayes formula is given by:

\[
\mathbb{P}(\phi \mid S) = \frac{\mathbb{P}(\phi, S)}{\mathbb{P}(S)} = \frac{\mathbb{P}(\phi, S)}{\sum_{\phi' \in \Phi_L} \mathbb{P}(\phi', S)}.
\]

The essential idea is to specify a precise probabilistic model of what a gene looks like in advance and then to select the parse \( \phi \) through the model \( M \) that has highest likelihood, given the sequence \( S \).

### 12.9 Computational issues

Given a sequence \( S \) of length \( L \), the joint probability \( \mathbb{P}(\phi, S) \) of generating the parse \( \phi \) and the sequence \( S \) is given by:

\[
\mathbb{P}(\phi, S) = \pi_{q_1} f_{q_1}(d_1) \mathbb{P}(s_1 \mid q_1, d_1) \times \prod_{k=2}^{n} p_{q_{k-1}, q_k} f_{q_k}(d_k) \mathbb{P}(s_k \mid q_k, d_k),
\]

where the states of \( \phi \) are \( q_1, q_2, \ldots, q_n \) with associated state lengths \( d_1, d_2, \ldots, d_n \), which break the sequence into segments \( s_1, s_2, \ldots, s_n \).

Here, \( f_{q_k}(d_k) \) is the probability to choose a sequence length \( d_k \) from the length distribution associated with state \( q_k \), \( p_{q_{k-1}, q_k} \) is the state transition probability of state \( q_k \) given the current state \( q_{k-1} \), and \( \mathbb{P}(s_k \mid q_k, d_k) \) is the probability of generating the segment \( s_k \) under the appropriate sequence generating model of state \( q_k \) with length \( d_k \).
A modification of the Viterbi algorithm may be used to calculate $\phi_{opt}$, the parse with maximal joint probability (under $M$), that gives the predicted gene or set of genes in the sequence.

We can compute $\mathbb{P}(S)$ using the “forward algorithm” discussed under HMMs. With the help of the “backward algorithm”, certain additional quantities of interest can also be computed.

For example, consider the event $E_{[x,y]}^{(k)}$ that a particular sequence segment $[x, y]$ is an internal exon of phase $k \in \{0, 1, 2\}$. Under $M$, this event has probability

$$
\mathbb{P}(E_{[x,y]}^{(k)} \mid S) = \frac{\sum_{\phi; E_{[x,y]}^{(k)} \in \phi} \mathbb{P}(\phi, S)}{\mathbb{P}(S)},
$$

where the sum is taken over all parses that contain the given exon $E_{[x,y]}^{(k)}$. This sum can be computed using the forward and backward algorithms.

### 12.10 Details of the model

So far, we have discussed the topology and the other main components of the Genscan model in general terms. The following details need to be discussed:

- the initial and transition probabilities,
- the state length distributions,
- transcriptional and translational signals,
- splice signals, and
- reverse-strand states.

#### 12.10.1 Initial and transition probabilities

Whenever a transition is obligatory, the transition probability is set to 1, such as $p(P+ \rightarrow F+)$, $p(E_{term}+ \rightarrow T+)$, $p(E_{singl}+ \rightarrow T+)$, $p(T+ \rightarrow A+)$ and $p(A+ \rightarrow N)$.

The initial probability of each states should be chosen proportionally to its estimated frequency in bulk (human) genomic DNA.

This is a non-trivial problem, because gene density and certain aspects of gene structure vary significantly in regions of differing C+G content (so-called “isochores”) of the human genome, with a much higher gene density in C+G-rich regions.

#### 12.10.2 Isochores

- Vertebrate genomes have large variability in GC content on a megabase scale.
- High GC regions have
  - much greater gene density (10:1 not unusual)
  - shorter introns
  - shorter intergenic regions
  - fewer repeats

Genscan uses different parameter sets for different sochores

Initial and transitional probabilities are estimated for four different categories:
Also 4 sets for intron/intergenic length are used.

The following initial probabilities were obtained from a training set of 380 human genes by comparing the number of bases corresponding to each of the different states:

<table>
<thead>
<tr>
<th></th>
<th>Group I</th>
<th>Group II</th>
<th>Group III</th>
<th>Group IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>C+G-range</td>
<td>&lt; 43%</td>
<td>43 − 51%</td>
<td>51 − 57%</td>
<td>&gt; 57%</td>
</tr>
</tbody>
</table>

Initial probabilities:

<table>
<thead>
<tr>
<th>State Type</th>
<th>Intergenic (N)</th>
<th>Intron (I⁺, I⁻)</th>
<th>5' UTR (F⁺, F⁻)</th>
<th>3' UTR (T⁺, T⁻)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.892</td>
<td>0.095</td>
<td>0.008</td>
<td>0.005</td>
</tr>
<tr>
<td></td>
<td>0.867</td>
<td>0.103</td>
<td>0.018</td>
<td>0.011</td>
</tr>
<tr>
<td></td>
<td>0.540</td>
<td>0.338</td>
<td>0.077</td>
<td>0.045</td>
</tr>
<tr>
<td></td>
<td>0.418</td>
<td>0.388</td>
<td>0.122</td>
<td>0.072</td>
</tr>
</tbody>
</table>

For simplicity, the initial probabilities for the exon, promoter and poly-A states were set to 0.

Transition probabilities were obtained in a similar way.

12.10.3 State length distributions

In general, the states of the model correspond to sequence segments of highly variable length. For example, the poly(A) site is about 6 bp long, while introns in some human genes can be longer than 2,000.

For certain states, most notably for internal exon states $E_k$, the length is probably important for proper biological function, i.e. proper splicing and inclusion in the final processed mRNA.

It has been shown in vivo that internal deletions of exons to sizes below about 50 may often lead to exon skipping, and there is evidence that steric interference between factors recognizing splice sites may make splicing of small exons more difficult. There is also evidence that spliceosomal assembly is inhibited if internal exons are expanded beyond 300.

In summary, these arguments support the observation that internal exons are usually $\approx 120 − 150$ long, with only a few of length less that 50 or more than 300.

Constraints for initial and terminal exons are slightly different.

The duration in initial, internal and terminal exon states is modeled by a different empirical distribution for each of the types of states.

In contrast to exons, the length of introns does not seem critical, although a minimum length of 70 − 80 may be preferred.

The length distribution for introns appears to be approximately geometric (exponential). However, the average length of introns differs substantially between the different C+G groups:

<table>
<thead>
<tr>
<th>Group</th>
<th>C+G-range</th>
<th>Average Intron Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt; 43%</td>
<td>2069</td>
</tr>
<tr>
<td></td>
<td>43 − 51%</td>
<td>1086</td>
</tr>
<tr>
<td></td>
<td>51 − 57%</td>
<td>801</td>
</tr>
<tr>
<td></td>
<td>&gt; 57%</td>
<td>518</td>
</tr>
</tbody>
</table>

Hence, the duration in intron states is modeled by a geometric distribution with parameter $q$ estimated for each C+G group separately.

Note that the exon lengths generated must be consistent with the phases of adjacent introns. To

*Remark K. Nieselt: why not at least for initial exon and/or single exon?*
account for this, first the number of complete codons is generated from the appropriate length distribution, then the appropriate number (0, 1 or 2) of bp is added to each end to account for the phases of the preceding and subsequent states.

For example, if the number of complete codons \( C \) generated for an internal exon is 6, and the phase of the previous and next intron is 1 and 2, respectively, then the total length of the exon is \( l = 3C + 2 + 2 = 22 \):

```
phase 1 intron   phase 1 exon   phase 2 exon   phase 2 intron
```

For the 5' UTR and 3' UTR states, geometric distributions are used with mean values of 769 and 457, respectively.

12.10.4 Simple signal models

There are a number of different models of biological signal sequences, such as donor and acceptor sites, promoters, etc.

One of the earliest and must influential approaches is the **position specific weight matrix (PSWM)** method, in which the frequency \( p_a^{(i)} \) of each nucleotide \( a \) at position \( i \) of a signal of length \( n \) is derived from a collection of aligned signal sequences.

The product \( \prod_{i=1}^{n} p_a^{(i)} \) is used to estimate the probability of generating a particular sequence \( A = a_1a_2 \ldots a_n \).

The **weight array matrix (WAM)** is a generalization that takes dependencies between two adjacent positions into account. In this model, the probability of generating a particular sequence is:

\[
P(A) = p_1^{(1)} \prod_{i=2}^{n} p_{a_{i-1},a_{i}}^{(i)}
\]

where \( p_{v,w}^{i-1,i} \) is the conditional probability of generating a particular nucleotide \( w \) at position \( i \), given nucleotide \( v \) at position \( i-1 \).

Here is a PSWM for a region around the start codon:

<table>
<thead>
<tr>
<th>Pos.</th>
<th>8</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
<th>+7</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>.16</td>
<td>.29</td>
<td>.20</td>
<td>.25</td>
<td>.22</td>
<td>.66</td>
<td>.27</td>
<td>.15</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>.28</td>
<td>.24</td>
<td>.11</td>
<td>.26</td>
</tr>
<tr>
<td>C</td>
<td>.48</td>
<td>.31</td>
<td>.21</td>
<td>.33</td>
<td>.56</td>
<td>.05</td>
<td>.50</td>
<td>.58</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>.16</td>
<td>.29</td>
<td>.24</td>
<td>.40</td>
</tr>
<tr>
<td>G</td>
<td>.18</td>
<td>.16</td>
<td>.46</td>
<td>.21</td>
<td>.17</td>
<td>.27</td>
<td>.12</td>
<td>.22</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>.48</td>
<td>.20</td>
<td>.45</td>
<td>.21</td>
</tr>
<tr>
<td>T</td>
<td>.19</td>
<td>.24</td>
<td>.14</td>
<td>.21</td>
<td>.06</td>
<td>.02</td>
<td>.11</td>
<td>.05</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>.09</td>
<td>.26</td>
<td>.21</td>
<td>.21</td>
</tr>
</tbody>
</table>

Under this model, the sequence \( \ldots CCGCCACC ATG GCGC \ldots \) has the highest probability of containing a start site, namely: \( P = 0.48 \cdot 0.31 \cdot 0.46 \cdot 0.33 \cdot 0.56 \cdot 0.66 \cdot 0.5 \cdot 0.58 \cdot 1 \cdot 1 \cdot 0.48 \cdot 0.29 \cdot 0.45 \cdot 0.4 = 0.006 \).

The sequence \( \ldots AGTTTTTT ATG TGAT \ldots \) has the lowest non-zero probability of containing a start site at the indicated position, namely: \( P = 0.16 \cdot 0.16 \cdot 0.14 \cdot 0.21 \cdot 0.06 \cdot 0.02 \cdot 0.11 \cdot 0.05 \cdot 1 \cdot 1 \cdot 1 \cdot 0.09 \cdot 0.2 \cdot 0.11 \cdot 0.21 = 2.1 \cdot 10^{-12} \).

12.10.5 Transcriptional and translational signals

Poly-A signals are modeled as a 6 PSWM model with consensus sequence AATAAA.

A 12 PSWM, beginning 6 prior to the start codon, is used for the translation initiation signal.

In both cases, one can estimate the probabilities using the GenBank annotated “polyA_signal” and “CDS” features of sequences.

Approximately 30% of eukaryotic promoters lack a TATA signal. Hence, a TATA-containing promoter is generated with 0.7 probability, and a TATA-less one with probability 0.3. TATA-less ones are modeled as intergenic regions of 40.

For TATA-containing promoters, a more elaborate model is used. It basically generates 3 strings and concatenates them to generate the promoter. The first string is obtained using a 15 TATA-box PSWM. For the second string, the length is randomly generated uniformly from the range 14 – 20.
The bases in the second string itself are chosen independently from the intergenic base frequencies. The third string is generated with an 8 cap site PSWM.

12.10.6 Splice signals

The donor and acceptor splice signals are probably the most important signals, as the majority of exons are internal ones. Previous approaches use PSWMs or WAMs to model them, thus assuming independence of sites, or that dependencies only occur between adjacent sites.

The consensus region of the donor splice sites covers the last 3 of the exon (positions -3 to -1) and the first 6 of the succeeding intron (positions 1 to 6) (see also section 8.4 p. 148):

<table>
<thead>
<tr>
<th>Position</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Consensus</td>
<td>c/a</td>
<td>A</td>
<td>G</td>
<td>T</td>
<td>a/g</td>
<td>A</td>
<td>G</td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>WMM:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>.33</td>
<td>.60</td>
<td>.08</td>
<td>0</td>
<td>0</td>
<td>.49</td>
<td>.71</td>
<td>.06</td>
<td>.15</td>
</tr>
<tr>
<td>C</td>
<td>.37</td>
<td>.13</td>
<td>.04</td>
<td>0</td>
<td>0</td>
<td>.03</td>
<td>.07</td>
<td>.05</td>
<td>.19</td>
</tr>
<tr>
<td>G</td>
<td>.18</td>
<td>.14</td>
<td>.81</td>
<td>1</td>
<td>0</td>
<td>.45</td>
<td>.12</td>
<td>.84</td>
<td>.20</td>
</tr>
<tr>
<td>T</td>
<td>.12</td>
<td>.13</td>
<td>.07</td>
<td>0</td>
<td>1</td>
<td>.03</td>
<td>.09</td>
<td>.05</td>
<td>.46</td>
</tr>
</tbody>
</table>

12.10.7 Donor site model

However, donor sites show significant dependencies between non-adjacent positions, which probably reflect details of donor splice site recognition by U1 snRNA and other factors.

Given a sequence \( S = (s_1, s_2, \ldots) \) and the consensus sequence \( Cons \) of an alignment. The consensus indicator variable \( C = (C_1, C_2, \ldots) \) is defined as:

\[
C_i = \begin{cases} 
1 & \text{if } s_i = Cons_i \\
0 & \text{else}
\end{cases}
\]

For example, consider:

<table>
<thead>
<tr>
<th>Position</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>+1</th>
<th>+2</th>
<th>+3</th>
<th>+4</th>
<th>+5</th>
<th>+6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cons</td>
<td>c/a</td>
<td>A</td>
<td>G</td>
<td>T</td>
<td>a/g</td>
<td>A</td>
<td>G</td>
<td>t</td>
<td></td>
</tr>
<tr>
<td>( S )</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( C )</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

For each pair of positions \( i \neq j \), consider the \( C_i \) versus \( s_j \) contingency table computed from the given learning set of gene structures:

\[
\begin{array}{ccccc}
& A & C & G & T \\
C_i & f_0(A) & f_0(C) & f_0(G) & f_0(T) \\
0 & f_1(A) & f_1(C) & f_1(G) & f_1(T) \\
1 & & & & \\
\end{array}
\]

where \( f_c(x) \) is the frequency at which the training set has the consensus indicator value \( c \) at position \( i \) and the base \( x \) at position \( j \). Now we would like to use a statistical test that computes a score testing the null-hypothesis that \( s_j \) does not depend on \( C_i \).

For this we use the (Pearson’s) \( \chi^2 \)-test\(^5\):

\[
\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}
\]  
(12.4)

\(^5\)Let \( X \) be a standard normally distributed variable \( N(0,1) \), then the distribution of the variable \( X^2 \) is called \( \chi^2 \) distribution with one degree of freedom.
The (Pearson’s) $\chi^2$-test assigns a score $\chi^2(C_i, s_j)$ to each pair of variables $C_i$ and $s_j$:

$$\chi^2(C_i, s_j) = \sum_{c \in \{0,1\}} \sum_{x \in \{A,C,G,T\}} \left( \frac{f_c(x) - f(x)}{f(x)} \right)^2,$$

where $f(x)$ denotes the frequency with which we observe $s_j = x$ in the training set (corresponding to the null-hypothesis that $s_j$ does not depend on $C_i$).

A significant score indicates that a dependency exists between $C_i$ and $S_j$.

In donor site prediction, the following procedure, a so-called maximal dependence decomposition, is carried out:

1. the positions $i$ are ordered by decreasing discriminatory power $Z_i = \sum_{j \neq i} \chi^2(C_i, S_j)$

2. Choose the position $i_1$ for which $Z_{i_1}$ is maximal and subdivide the alignment into 2 subsets: one contains all sequences that have the consensus symbol at position $i_1$, and the other subset, that don’t.

Then repeat steps 1.) and 2.) on the subsets.

<table>
<thead>
<tr>
<th>Pos</th>
<th>A%</th>
<th>C%</th>
<th>G%</th>
<th>U%</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>33</td>
<td>36</td>
<td>19</td>
<td>13</td>
</tr>
<tr>
<td>-2</td>
<td>56</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>-1</td>
<td>9</td>
<td>4</td>
<td>78</td>
<td>9</td>
</tr>
<tr>
<td>+3</td>
<td>44</td>
<td>3</td>
<td>51</td>
<td>3</td>
</tr>
<tr>
<td>+4</td>
<td>75</td>
<td>4</td>
<td>13</td>
<td>9</td>
</tr>
<tr>
<td>+6</td>
<td>14</td>
<td>18</td>
<td>19</td>
<td>49</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pos</th>
<th>A%</th>
<th>C%</th>
<th>G%</th>
<th>U%</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>35</td>
<td>44</td>
<td>16</td>
<td>6</td>
</tr>
<tr>
<td>-2</td>
<td>85</td>
<td>4</td>
<td>7</td>
<td>5</td>
</tr>
<tr>
<td>-1</td>
<td>2</td>
<td>1</td>
<td>97</td>
<td>0</td>
</tr>
<tr>
<td>+3</td>
<td>81</td>
<td>3</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>+4</td>
<td>51</td>
<td>28</td>
<td>9</td>
<td>12</td>
</tr>
<tr>
<td>+6</td>
<td>22</td>
<td>20</td>
<td>30</td>
<td>28</td>
</tr>
</tbody>
</table>

All donor splice sites (1254)

<table>
<thead>
<tr>
<th>Pos</th>
<th>A%</th>
<th>C%</th>
<th>G%</th>
<th>U%</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>29</td>
<td>31</td>
<td>21</td>
<td>18</td>
</tr>
<tr>
<td>-2</td>
<td>43</td>
<td>30</td>
<td>17</td>
<td>11</td>
</tr>
<tr>
<td>+3</td>
<td>56</td>
<td>0</td>
<td>43</td>
<td>0</td>
</tr>
<tr>
<td>+4</td>
<td>93</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>+6</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td>76</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pos</th>
<th>A%</th>
<th>C%</th>
<th>G%</th>
<th>U%</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>37</td>
<td>42</td>
<td>18</td>
<td>3</td>
</tr>
<tr>
<td>+3</td>
<td>39</td>
<td>3</td>
<td>51</td>
<td>5</td>
</tr>
<tr>
<td>+4</td>
<td>62</td>
<td>5</td>
<td>22</td>
<td>11</td>
</tr>
<tr>
<td>+6</td>
<td>19</td>
<td>25</td>
<td>25</td>
<td>56</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pos</th>
<th>A%</th>
<th>C%</th>
<th>G%</th>
<th>U%</th>
</tr>
</thead>
<tbody>
<tr>
<td>-3</td>
<td>35</td>
<td>43</td>
<td>15</td>
<td>2</td>
</tr>
<tr>
<td>+3</td>
<td>46</td>
<td>6</td>
<td>46</td>
<td>3</td>
</tr>
<tr>
<td>+4</td>
<td>69</td>
<td>5</td>
<td>20</td>
<td>7</td>
</tr>
</tbody>
</table>

(Source: Burge and Karlin 1997)

Here, $H = A|C|U$, $B = C|G|U$ and $V = A|C|G$. For example, $G_5$, or $H_5$, is the set of donor sites with, or without, a $G$ at position +5, respectively.

### 12.10.8 Acceptor site model

Introns/exons junctions are modeled by a (first-order) WAM for bases $-20$ to $+3$, capturing the pyrimidine (C,T) rich region and the acceptor splice site itself.

It is difficult to model the branch point in the preceding intron, and only 30% of the test data had an YYRAY sequence in the appropriate region $[-40, -21]$.

A modified variant of a second-order WAM is employed in which nucleotides are generated conditional on the previous two ones, in an attempt to model the weak but detectable tendency toward YYY triplets as well as certain branch point-related triplets such as TGA, TAA, GAC, and AAC in this region, without requiring the occurrence of any specific branch point consensus.

(A windowing and averaging process is used to obtain estimates from the limited training data.)
12.10.9 Exon models

Coding portions of exons are modeled using an inhomogeneous 3-periodic fifth order Markov model. Here, separate Markov transition matrices, $C_1$, $C_2$ and $C_3$, are determined for hexamers ending at each of the three codon positions, respectively:

\[
\begin{array}{ccccccc}
\hline
\text{xxxxxxxxxx} & \text{x1} & \text{x2} & \text{x3} & \text{y1} & \text{y2} & \text{y3} \\
\hline
\text{C1} & \hline
\text{C2} & \hline
\text{C3} & \hline
\end{array}
\]

This is based on the observation that frame-shifted hexamer counts are generally the most accurate compositional discriminator of coding versus non-coding regions.

However, A+T rich genes are often not well predicted using hexamer counts based on bulk DNA and so GENSCAN uses two different sets of transition matrices, one trained for sequences with $< 43\%$ C+G content and one for all others.

12.11 Example of GENSCAN summary output

```
GENSCAN 1.0 Date run: 11-Dec-2005 Time: 11:35:14
Sequence CFTR : 194880 bp : 35.98% C+G : Isochore 1 ( 0.00 - 43.00 C+G)
Parameter matrix: HumanIso.smat
Predicted genes/exons: ---------

<table>
<thead>
<tr>
<th>Gene</th>
<th>Type</th>
<th>S</th>
<th>Begin</th>
<th>End</th>
<th>Len</th>
<th>Fr</th>
<th>Ph</th>
<th>I/Ac</th>
<th>Do/T</th>
<th>CodRg</th>
<th>P</th>
<th>Tscr</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.03</td>
<td>PlyA</td>
<td>-</td>
<td>1239</td>
<td>1234</td>
<td>6</td>
<td>1.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.02</td>
<td>Term</td>
<td>-</td>
<td>5774</td>
<td>5421</td>
<td>354</td>
<td>2</td>
<td>0</td>
<td>112</td>
<td>44</td>
<td>285</td>
<td>0.847</td>
<td>20.01</td>
</tr>
<tr>
<td>1.01</td>
<td>Init</td>
<td>-</td>
<td>17773</td>
<td>17669</td>
<td>105</td>
<td>1</td>
<td>0</td>
<td>65</td>
<td>64</td>
<td>69</td>
<td>4.47</td>
<td></td>
</tr>
<tr>
<td>1.00</td>
<td>Prom</td>
<td>-</td>
<td>22590</td>
<td>22551</td>
<td>40</td>
<td>-5.75</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.04</td>
<td>PlyA</td>
<td>-</td>
<td>22626</td>
<td>22621</td>
<td>6</td>
<td>1.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.03</td>
<td>Term</td>
<td>-</td>
<td>23609</td>
<td>23492</td>
<td>118</td>
<td>1</td>
<td>1</td>
<td>90</td>
<td>42</td>
<td>128</td>
<td>0.992</td>
<td>5.43</td>
</tr>
<tr>
<td>2.02</td>
<td>Instr</td>
<td>-</td>
<td>40636</td>
<td>40386</td>
<td>252</td>
<td>2</td>
<td>0</td>
<td>66</td>
<td>72</td>
<td>131</td>
<td>0.478</td>
<td>7.57</td>
</tr>
<tr>
<td>2.01</td>
<td>Init</td>
<td>-</td>
<td>41269</td>
<td>41056</td>
<td>215</td>
<td>1</td>
<td>2</td>
<td>58</td>
<td>86</td>
<td>65</td>
<td>4.16</td>
<td></td>
</tr>
<tr>
<td>2.00</td>
<td>Prom</td>
<td>-</td>
<td>41347</td>
<td>41308</td>
<td>40</td>
<td>-3.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.00</td>
<td>Prom</td>
<td>-</td>
<td>43102</td>
<td>43141</td>
<td>40</td>
<td>-7.36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.01</td>
<td>Init</td>
<td>-</td>
<td>43192</td>
<td>43244</td>
<td>53</td>
<td>0</td>
<td>2</td>
<td>59</td>
<td>36</td>
<td>101</td>
<td>0.444</td>
<td>2.68</td>
</tr>
<tr>
<td>3.02</td>
<td>Instr</td>
<td>-</td>
<td>52104</td>
<td>54935</td>
<td>2632</td>
<td>2</td>
<td>0</td>
<td>94</td>
<td>53</td>
<td>936</td>
<td>76.29</td>
<td></td>
</tr>
<tr>
<td>3.03</td>
<td>Prom</td>
<td>-</td>
<td>56216</td>
<td>57239</td>
<td>126</td>
<td>1</td>
<td>0</td>
<td>36</td>
<td>43</td>
<td>72</td>
<td>1.67</td>
<td></td>
</tr>
<tr>
<td>3.04</td>
<td>Instr</td>
<td>-</td>
<td>66166</td>
<td>66412</td>
<td>247</td>
<td>1</td>
<td>1</td>
<td>74</td>
<td>95</td>
<td>113</td>
<td>7.14</td>
<td></td>
</tr>
<tr>
<td>3.05</td>
<td>Prom</td>
<td>-</td>
<td>68082</td>
<td>68174</td>
<td>93</td>
<td>2</td>
<td>0</td>
<td>85</td>
<td>54</td>
<td>70</td>
<td>1.64</td>
<td></td>
</tr>
<tr>
<td>3.06</td>
<td>Prom</td>
<td>-</td>
<td>74707</td>
<td>74889</td>
<td>183</td>
<td>0</td>
<td>82</td>
<td>63</td>
<td>140</td>
<td>988</td>
<td>9.96</td>
<td></td>
</tr>
<tr>
<td>3.07</td>
<td>Instr</td>
<td>-</td>
<td>85630</td>
<td>85721</td>
<td>192</td>
<td>2</td>
<td>0</td>
<td>97</td>
<td>94</td>
<td>98</td>
<td>10.07</td>
<td></td>
</tr>
<tr>
<td>3.08</td>
<td>Instr</td>
<td>-</td>
<td>113805</td>
<td>113899</td>
<td>96</td>
<td>2</td>
<td>2</td>
<td>80</td>
<td>63</td>
<td>129</td>
<td>7.44</td>
<td></td>
</tr>
<tr>
<td>3.09</td>
<td>Prom</td>
<td>-</td>
<td>116419</td>
<td>116506</td>
<td>87</td>
<td>1</td>
<td>89</td>
<td>95</td>
<td>26</td>
<td>397</td>
<td>1.47</td>
<td></td>
</tr>
<tr>
<td>3.10</td>
<td>Term</td>
<td>-</td>
<td>118000</td>
<td>118771</td>
<td>772</td>
<td>1</td>
<td>1</td>
<td>50</td>
<td>48</td>
<td>415</td>
<td>25.26</td>
<td></td>
</tr>
<tr>
<td>3.11</td>
<td>PlyA</td>
<td>-</td>
<td>119067</td>
<td>119072</td>
<td>6</td>
<td>1.05</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Explanation

```
Gn.Ex : gene number, exon number (for reference)
Type   : Init = Initial exon Instr = Internal exon Term = Terminal exon
Sngl   = Single-exon gene Prom = Promoter PlyA = poly-A signal
S : DNA strand (+ = input strand; - = opposite strand)
Begin : beginning of exon or signal (numbered on input strand)
End   : end point of exon or signal (numbered on input strand)
Len   : length of exon or signal (bp)
Fr    : reading frame (a codon ending at x is in frame f = x mod 3)
Ph    : net phase of exon (length mod 3)
I/Ac  : initiation signal or acceptor splice site score (x 10)
Do/T  : donor splice site or termination signal score (x 10)
CodRg : coding region score (x 10)
P    : probability of exon (sum over all parses containing exon)
Tscr  : exon score (depends on length, I/Ac, Do/T and CodRg scores)
```
12.12 Performance studies

The performance of a gene prediction program is evaluated by applying it to DNA sequences for which all contained genes are known and annotated with high confidence.

For our example, we can test on several levels. Here the predicted peptide sequences were compared via Blastp against GenBank:

```
Query= CFTR|GENSCAN_predicted_peptide_4|627_aa
(627 letters)
Score E
Sequences producing significant alignments: (Bits) Value
ref|XP_416011.1| PREDICTED: similar to chloride channel [Gallus] 533 6e-150
emb|CAO08756.1| unnamed protein product [Tetraodon nigroviridis] 403 1e-110
gb|AAB46340.2| unknown [Homo sapiens] 384 6e-105
gb|EAL24353.1| cystic fibrosis transmembrane conductance regul... 384 6e-105
...
```

```
Query= CFTR|GENSCAN_predicted_peptide_3|1559_aa
(1559 letters)
Score E
Sequences producing significant alignments: (Bits) Value
gi|5052951|gb|AAD38785.1| unknown [Homo sapiens] 1802 0.0
gi|2072948|gb|AAG51261.1| putative p150 [Homo sapiens] 1799 0.0
gi|339777|gb|AAB59368.1| ORF2 contains a reverse transcriptase d 1792 0.0
gi|106332|pir||B34087 hypothetical protein (LiH 3' region) - hum 1759 0.0
gi|225047|pr||1207289A reverse transcriptase related protein 1737 0.0
gi|126295|sp|P08547|LIN1_HUMAN LINE-1 reverse transcriptase homo 1726 0.0
...
```

```
Query= CFTR|GENSCAN_predicted_peptide_2|161_aa
(161 letters)
Score E
Sequences producing significant alignments: (Bits) Value
gi|55630186|ref|XP_519595.1| PREDICTED: similar to FLJ46489 prot 87.8 1e-16
gi|21752586|dbj|BAC04215.1| unnamed protein product [Homo sapien 83.6 2e-15
gi|126295|sp|P08547|LIN1_HUMAN LINE-1 reverse transcriptase homo 82.8 4e-15
...
```

To calculate accuracy statistics, each nucleotide of a test sequence is classified as:

- a **predicted positive (PP)** if it is predicted to be contained in a coding region,
- a **predicted negative (PN)** if it is predicted to be contained in non-coding region,
- an **actual positive (AP)** if it is annotated to be contained in coding region, and
- an **actual negative (AN)** if it is annotated to be contained in non-coding region.

Based on these assignments, we compute the number of: 2pt

- true positives, \( TP = PP \cap AP \),
• false positives, $FP = PP \cap AN$,
• true negatives, $TN = PN \cap AN$, and
• false negatives, $FN = PN \cap AP$.

The sensitivity $Sn$ and specificity $Sp$ of a method are then defined as

$$Sn = \frac{TP}{AP} \quad \text{and} \quad Sp = \frac{TP}{PP},$$

respectively, measuring both the ability to predict true genes and to avoid predicting false ones.

Remark: sometimes one sees $Sn = \frac{TP}{TP + FN}$ and $Sp = \frac{TN}{TN + FP}$ or $Sp = \frac{TP}{TP + FP}$.

At the exon level, predicted exons (PE) are compared to annotated exons (AE). True exons (TE) is the number of predicted exons which are exactly identical to an annotated exon (i.e. both endpoints correct). Accuracy is again measured by:

Sensitivity: $Sn = TE/AE$

Specificity: $Sp = TE/PE$

The average of $Sn$ and $Sp$ is typically used as an overall measure of accuracy at the exon level. Two additional accuracy measures are also calculated at the exon level: Missing Exons (ME), the fraction of annotated exons not overlapped by any predicted exon; and Wrong Exons (WE), the fraction of predicted exons not overlapped by any true exon. Accuracy measures for a set of sequences are calculated by averaging the values obtained for each sequence separately, the average being taken over all sequences for which the measure is defined.

### 12.12.1 Performance of GENSCAN

GENSCAN was run on a test set of 570 vertebrate sequences and the forward strand exons in the optimal GENSCAN parse of the sequence were compared to the annotated exons. The following table shows the results and compares them with results obtained using other programs:

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy per nucleotide</th>
<th>Accuracy per exon</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$Sn$</td>
<td>$Sp$</td>
</tr>
<tr>
<td>GENSCAN</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>FGENEH</td>
<td>0.77</td>
<td>0.85</td>
</tr>
<tr>
<td>GeneID</td>
<td>0.63</td>
<td>0.81</td>
</tr>
<tr>
<td>GeneParser2</td>
<td>0.66</td>
<td>0.79</td>
</tr>
<tr>
<td>GenLang</td>
<td>0.72</td>
<td>0.75</td>
</tr>
<tr>
<td>GRAILII</td>
<td>0.72</td>
<td>0.84</td>
</tr>
<tr>
<td>SORFIND</td>
<td>0.71</td>
<td>0.85</td>
</tr>
<tr>
<td>Xpound</td>
<td>0.61</td>
<td>0.82</td>
</tr>
</tbody>
</table>

(Source: Burge and Karlin 1997)
211

Summary: GENSCAN performs very well and is currently the most popular gene finding method.

12.13 Comparative gene finding

GENSCAN’s model makes use of statistical features of the genome under consideration, obtained from an annotated training set.

More recently, a number of methods have been suggested that attempt to also make use of comparative data. They are based on the observation that

the level of sequence conservation between two species depends on the function of the DNA, e.g. coding sequence is more conserved than intergenic sequence.

One such program is Rosetta\(^6\) which first computes a global alignment of two homologous sequences and then attempts to predict genes in both sequences simultaneously. A second is the conserved exon method\(^7\) that uses local conservation.

The Twinscan program is an extension of GENSCAN, that additionally models a conserved sequence.

Orthologous Genes: homologous genes in two species that have a common ancestor\(^8\)

Example: the human CFTR gene and its mouse ortholog.

Most human genes have mouse orthologs\(^9\). Moreover, 95% of coding exons are in a one-to-one correspondence between the two genomes. 75% of orthologous coding exons have equal length, and 95% have equal length modulo 3. Intron lengths differ by an average of 50%. The coding sequence similarity between the two organisms is around 85%, the intron sequence similarity is around 35%, 5’ UTRs and 3’ UTRs around 68%.

We want to take into account that the sequences are genome sequences:

Example: a pair of syntenic genomic regions

Finding genes in two genome sequences then tries to identify the islands (eg., exons) of high similarity.


\(^7\)Bafna, and Huson, D., 2000

\(^8\)Question: and what are paralogous genes?

\(^9\)Makalowski et al., 1996
12.14 TWINSCAN

TWINSCAN (Korf et al., 2001) extends the probability model of GENSCAN, allowing it to exploit homology between two related genomes.

The input to TWINSCAN consists of a target sequence, i.e. a genomic sequence in which genes are to be predicted, and an informant sequence, i.e. a genomic sequence from a related organism similar to the target sequence.

For example, the target may come from the mouse genome and the informant from the human genome.

Given a target and an informant, in a preprocessing step, one determines a set of top homologs (e.g. using BLAST) from the informant sequence, i.e. one or more sequences from the informant sequence that best match the target sequence.

mouse

conserved human (top homologs)

The top homologs represent the regions of conserved informant sequence, which we will simply call “the informant sequence” in the following.

12.14.1 Conservation sequence

Similarity is represented by a conservation sequence, which pairs one of three symbols with each nucleotide of the target:

. unaligned | matched : mismatched

Gaps in the informant sequence become mismatch symbols, gaps in the target sequence are ignored. Consider:

123456789 position
GAATTCCGT target sequence

and suppose that BLAST yields the following HSP:

345 6789 target position
ATT-CCGT target alignment
|| || | BLAST match symbol
6||:||:| conservation sequence
ATCACC-T Informant alignment

The following algorithm takes a list of HSPs and computes the conservation sequence $C$:

```
Algorithm
Input: target sequence, list of HSPs
Output: conservation sequence $C$
Init.: $C[1..n] := \text{unaligned}$
Sort HSPs by alignment score
for each position $i$ in the target sequence:
  for each HSP $H$ from best to worst score:
    if $H$ covers position $i$:
      if $C[i] = \text{unaligned}$:
        $C[i] := \text{'}$, in case of a match, and $C[i] := \text{'}.$ otherwise
      if $C[i] = \text{unaligned}$:
        $C[i] := \text{'}.$
```
Note that the conservation symbol assigned to the target nucleotide at position $i$ is determined by the best HSP that covers $i$, regardless of which homologous sequence it comes from. Position $i$ is classified as unaligned only if none of the HSPs overlap it.

12.14.2 Probability of sequence and conservation sequence

Recall that GENSCAN assigns each nucleotide of an input sequence to one of seven categories: promoter, 5' UTR, exon, intron, 3' UTR, poly-A signal and intergenic.

GENSCAN chooses the most likely assignment of categories to nucleotides according to the GENSCAN model, using an optimization algorithm (that is, a modification of the Viterbi algorithm).

Given a sequence, the GENSCAN model assigns a probability to each parse of the sequence (that is, a path of states and durations through the model that generates the sequence.)

The TWINSAN model assigns a probability to any parsed DNA sequence together with a parallel conservation sequence. Under this model, the probability of a DNA sequence and the probability of the parallel conservation sequence are independent, given the parse.

Consider the following example:

| 10 | 20 | 30 |
|-----------------------------|
| 123456789 | 123456789 | 123456789 | 123456789 |
| ATTTAGCTACTGAAATGGACCGCTTCAGCATGGTATCC target sequence T
| ||:|||............|:|:|||||||||:||:|||::|| conservation sequence C

What is the probability of observing the target sequence $T_{7,33}$ and conservation sequence $C_{7,33}$ extending from position 7 to 33, given the hypothesis $E_{7,33}$ that an internal exon extends from position 7 to 33?

This is simply the probability of the target sequence $T_{7,33}$ under the GENSCAN model times the probability of the conservation sequence $C_{7,33}$ under the conservation model, assuming the parse $E_{7,33}$:

$$P(T_{7,33}, C_{7,33} \mid E_{7,33}) = P(T_{7,33} \mid E_{7,33})P(C_{7,33} \mid E_{7,33}).$$

12.14.3 TWINSAN’s model

TWINSAN consists of a new, joint probability model on DNA sequences and conservation sequences, together with the same optimization algorithm used by GENSCAN.

TWINSAN augments the state-specific sequence models of GENSCAN with models for the probability of generating any given conservation sequence from any given state.

Coding, UTR, and intron/intergenic states all assign probabilities to stretches of conservation sequence using homogeneous 5th-order Markov chains:

```
ccccccccccccc c1 c2 c3 c4 c5 c6 ccccccccccccc
```

One set of parameters is estimated for each of these types of regions.
Again, consider:

```
10 20 30
123456789|123456789|123456789|123456789
ATTAGCCTACTGAAATGGACCGCTTCAGCATGGTATCC target sequence T
||:|||.........|:|:|||||||||:||:|||::|| conservation sequence C
```

The probability of observing $C_{7,33}$, given $E_{7,33}$, is:

$$P_C(C_{7,33} \mid E_{7,33}) = P_E(C_{7,7} \mid C_{2,6}) \cdot \cdots \cdot P_E(C_{33,33} \mid C_{28,32}),$$

where $P_E(C_{33,33} \mid C_{28,32})$, for example, is the estimated probability of a ‘:’ (match) following the given context symbols “|||" in the conservation sequence of an exon.

Models of conservation at splice donor and acceptor sites are modeled using 2nd-order WAMs of length 9 and 43, respectively (lengths as in GENSCAN).

### 12.15 TWINSCAN’s performance

TWINSCAN was tested on two data sets. The first set consists of 86 mouse sequences totaling 7.6 and used top homologs from human:

<table>
<thead>
<tr>
<th>Program</th>
<th>Exons</th>
<th>Exon Sn</th>
<th>Exon Sp</th>
<th>Genes</th>
<th>Genes Sn</th>
<th>Genes Sp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation</td>
<td>2758</td>
<td>275</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GENSscan</td>
<td>2997</td>
<td>0.631</td>
<td>0.581</td>
<td>395</td>
<td>0.153</td>
<td>0.106</td>
</tr>
<tr>
<td>TWINSCAN</td>
<td>2854</td>
<td>0.683</td>
<td>0.660</td>
<td>464</td>
<td>0.244</td>
<td>0.144</td>
</tr>
</tbody>
</table>

The second set is a subset containing 8 pairs of finished orthologs:

<table>
<thead>
<tr>
<th>Program</th>
<th>Exons</th>
<th>Exon Sn</th>
<th>Exon Sp</th>
<th>Genes</th>
<th>Genes Sn</th>
<th>Genes Sp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Annotation</td>
<td>610</td>
<td>48</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GENSscan</td>
<td>731</td>
<td>0.798</td>
<td>0.666</td>
<td>51</td>
<td>0.167</td>
<td>0.157</td>
</tr>
<tr>
<td>TWINSCAN</td>
<td>684</td>
<td>0.854</td>
<td>0.752</td>
<td>50</td>
<td>0.271</td>
<td>0.260</td>
</tr>
</tbody>
</table>

### 12.15.1 Twinscan versus Genscan:

The crucial difference between Twinscan and Genscan is, that while GenScan uses sophisticated species-dependent statistical models to distinguish coding from non-coding regions, Twinscan is based on a simple and universally applicable measure of local sequence similarity and on basic models for splice junctions. These two approaches therefore complement each other in that they use different types of input information.

Moreover, since Twinscan being a comparative gene-prediction approach does not rely on statistical models derived from known genes of a given species, it can be applied to genome sequences from newly sequenced organisms where no training data are available - provided syntenic sequences are available from a second species at an appropriate evolutionary distance. With the increasing number of whole-genome sequencing projects, it will become feasible to find syntenic sequence pairs from related organisms.

### 12.16 Homology-based methods: BLAST

Comparison of a genomic DNA versus a database of CDS-sequences. Which BLAST do you use?

Example:
Application of tBLASTx

Suppose we are given two similar DNA sequences $S$ and $T$.

The program tBLASTx produces a list of high-scoring pairs (HSPs) of locally aligned substrings of $S$ and $T$, where the two substrings are interpreted as amino-acid coding strings and the score of the alignment is computed using a BLOSUM or PAM protein scoring matrix.

This is how an HSP is reported by tBLASTx:

Score = 214 (98.4 bits), Expect = 0.0, Sum P(24) = 0.0
Identities = 44/46 (95%), Positives = 46/46 (100%), Frame = +1 / +1

Query: 5284 RLVLRIATDDSKAVCRLSVKFGATLRTSRLLERAKELNIDVVGVR 5421
       RLVLRIATDDSKAVCRLSVKFGATLTSRLLERAKELNIDV+GVR
Sbjct: 3871 RLVLRIATDDSKAVCRLSVKFGATLKTSRLLERAKELNIDVIGVR 4008

In this example, the positions 5284–5421 of sequence $S$ and positions 3871–4008 of sequence $T$ are aligned together and interpreted as amino-acids as shown. The “frame” indicates the directions and the offsets of the two substrings.

12.17 Homology method: Procrustes

Any newly sequence gene has a good chance of having an already known relative and progress in large-scale sequencing projects is rapidly increasing the number of known genes and protein sequences.

Hence, homology-based gene prediction methods have therefore proven to be very useful. In particular, such a method may be able to detect exons that are missed by statistical methods because they are small, or statistically unusual.

Procrustes is a popular program that uses homology to predict genes: use a related protein in one genome to reconstruct the exon-intron structure of a gene in another genome.

12.17.1 About the name: Procrustes

PROCRUSTES, also called POLYPEMON, DAMASTES, or PROCOPTAS, in Greek legend, a robber dwelling somewhere in Attica—in some versions, in the neighbourhood of Eleusis. His father was said to be Poseidon. Procrustes had an iron bed (or, according to some accounts, two beds) on which he compelled his victims to lie. Here, if a victim was shorter than the bed, he stretched him by hammering or racking the body to fit. Alternatively, if the victim was longer than the bed, he cut off the legs to make the body fit the bed’s length. In either event the victim died. Ultimately Procrustes was slain by his own method at the hands of the young Attic hero Theseus, who as a young man went about slaying robbers and monsters that pervaded the countryside.

The "BED OF PROCRUSTES" or "PROCRUSTEAN BED" has become proverbial for arbitrarily - and perhaps ruthlessly - forcing someone or something to fit into an unnatural scheme or pattern.
PROCRUSTES forces genomic DNA sequence to fit into a related target protein.

12.17.2 Spliced-alignment problem

Idea: Given a genomic sequence $G$, a set of candidate exons (blocks) $B$ and a target sequence $T$. Determine a chain $\Gamma$ of non-overlapping blocks in $B$ that has the highest spliced-alignment score with target $T$. These blocks are interpreted as exons and the chain $\Gamma$ is the predicted gene structure.

12.17.3 Example

Given the genomic sequence $G = \text{baabaablacksheep} \text{have you any wool}$ and the target sequence $T = \text{barbarasleepsonwool}$, find the best spliced alignment of $T$ to $G$ and thus obtain a gene prediction in $G$:

Genome sequence:

\[
\begin{align*}
\text{baa} & \quad \text{baa} & \quad \text{black} & \quad \text{sheep} & \quad \text{have} & \quad \text{you} & \quad \text{any} & \quad \text{wool} \\
\text{baa} & \quad \text{baa} & \quad \text{black} & \quad \text{sheep} & \quad \text{have} & \quad \text{you} & \quad \text{any} & \quad \text{wool} \\
\text{baa} & \quad \text{baa} & \quad \text{black} & \quad \text{sheep} & \quad \text{have} & \quad \text{you} & \quad \text{any} & \quad \text{wool}
\end{align*}
\]

Best spliced alignment:

\[
\begin{align*}
\text{barbara} & \quad \text{sleeps} & \quad \text{on} & \quad \text{wool}
\end{align*}
\]

Resulting gene structure prediction:

\[
\begin{align*}
\text{baa} & \quad \text{baa} & \quad \text{sheep} & \quad \text{any} & \quad \text{wool}
\end{align*}
\]

There are many possible chainings of blocks in the given example:

\[
\begin{align*}
\text{baa} \rightarrow \text{baa} \rightarrow \text{black} \rightarrow \text{sheep} \rightarrow \text{have} \rightarrow \text{you} \rightarrow \text{any} \rightarrow \text{wool} \\
\text{baa} \rightarrow \text{baa} \rightarrow \text{black} \rightarrow \text{sheep} \rightarrow \text{have} \rightarrow \text{you} \rightarrow \text{any} \rightarrow \text{wool} \\
\text{baa} \rightarrow \text{baa} \rightarrow \text{black} \rightarrow \text{sheep} \rightarrow \text{have} \rightarrow \text{you} \rightarrow \text{any} \rightarrow \text{wool}
\end{align*}
\]

However, we choose the one that yields the best alignment to the given target sequence.

12.17.4 Preprocessing: determining the blocks

Given a genomic sequence $G$. The first computational step is to determine the set $B$ of all candidate blocks for $G$, which should contain all true exons. Naively, this is done by selecting all blocks between potential acceptor and donor sites, which are detected using e.g. a PSWM:
Clearly, this set of blocks will contain many false exons. One obvious way is to remove all blocks that contain a stop codon in each of the three reading frames. Statistical methods may be used in an attempt to remove blocks that are obviously not true exons.

Any chain of blocks corresponds to a gene prediction and the number of such chains can be huge. Dynamic programming is used to obtain an algorithm that runs in polynomial time.

12.17.5 The spliced alignment problem

Let \( G = g_1 \ldots g_n \) be a string of letters, and \( B = g_i \ldots g_j \) and \( B' = g_{i'} \ldots g_{j'} \) be substrings of \( G \). We write \( B \prec B' \), if \( B \) ends before \( B' \) starts, i.e. \( j < i' \). A sequence \( \Gamma = (B_1, \ldots, B_b) \) of substrings of \( G \) is a chain, if \( B_1 \prec B_2 \prec \ldots \prec B_b \). We denote the concatenation of the strings in \( \Gamma \) by \( \Gamma^* = B_1* B_2* \ldots * B_b \).

For two strings \( A \) and \( B \), we set \( s(A, B) \) to the score of an optimal alignment between \( A \) and \( B \).

Spliced Alignment Problem (SAP)
Let \( G = g_1 \ldots g_n \) be a genomic sequence, \( T = t_1 \ldots t_m \) a target sequence and \( B = \{B_1, \ldots, B_b\} \) a set of blocks in \( G \). Given \( G, T \) and \( B \), the Spliced Alignment Problem is to find a chain \( \Gamma \) of strings from \( B \) such that the score \( s(\Gamma^*, T) \) is maximum among all chains of blocks from \( B \).

12.17.6 Solving the spliced alignment problem

The SAP can be reduced to the search of a path in some (unweighted) directed graph. Vertices of this graph correspond to the blocks, arcs correspond to potential transitions between blocks, and the path weight is defined as the weight of the optimal alignment between the concatenated blocks of this path and the target sequence.

For simplicity, we will consider sequence alignment with linear gap penalties and define the \( \Delta_{\text{match}} \), \( \Delta_{\text{mismatch}} \) and \( \Delta_{\text{indel}} \) scores as usual.

We set \( \Delta(x, y) = \begin{cases} \Delta_{\text{match}} & \text{if } x = y, \\ \Delta_{\text{mismatch}} & \text{else.} \end{cases} \)

Compute the best alignment between the \( i \)-prefix of the genomic sequence \( G \) and the \( j \)-prefix of target \( T \): \( S(i, j) \)

But what is the \( i \)-prefix of \( G \)?

There may be a few \( i \)-prefixes of \( G \) depending on which block \( B \) we are in.

Compute the best alignment between the \( i \)-prefix of the genomic sequence \( G \) and the \( j \)-prefix of target \( T \) under the assumption that the alignment uses the block \( B \) at position \( i \): \( S(i, j, B) \)

12.17.7 The score of a prefix alignment

For a block \( B_k = g_m \ldots g_l \) in \( G \), define \( \text{first}(k) = m \), \( \text{last}(k) = l \) and \( \text{size}(k) = l - m + 1 \). Let \( B_k(i) \) denote the \( i \)-prefix \( g_m \ldots g_i \) of \( B_k \), if \( m \leq i \leq l \).

Given a position \( i \) and let \( \Gamma = (B_1, \ldots, B_k, \ldots, B_t) \) be a chain such that some block \( B_k \) contains \( i \). We define

\[
\Gamma^*(i) = B_1* B_2* \ldots * B_k(i)
\]
as the concatenation of $B_1 \ldots B_{k-1}$ and the $i$-prefix of $B_k$.

Then

$$S(i, j, k) = \max_{\text{all chains } \Gamma \text{ containing block } B_k} s(\Gamma^*(i), T(j)),$$

is the optimal score for aligning a chain of blocks up to position $i$ in $G$ to the $j$-prefix of $T$. As we will see, the values of this matrix are computed using dynamic programming.

### 12.17.8 The dynamic program

Let $\mathcal{B}(i) = \{ k \mid \text{last}(k) < i \}$ be the set of all blocks that end (strictly) before position $i$ in $G$.

We now distinguish 5 cases. First we look at the two trivial ones:

1. $j = 0$: By definition of $S(i, 0, k)$ we align only gaps to symbols of certain blocks that lie in front of block $B_k$ with symbols from $G[\text{first}(B_k)..i]$. We assume $\Delta_{\text{mismatch}} < 0$, we then get

$$S(i, 0, k) = \sum_{l=\text{first}(B_k)}^{i} \Delta(G(l, -))$$

2. $j > 0$ and there is no block $B < B_k$: then we can only align the string $G[\text{first}(B_k)..i]$ with $T[1..j]$, and therefore the score is

$$S(i, j, k) = s(G[\text{first}(B_k)..i], T[1..j])$$

3. $j > 0$, and there is a block $B < B_k$: We distinguish now whether position $i$ is the first position of block $B_k$ or not. For the latter case, then $B_k$ contains at least two further symbols, $G(i-1)$ and $G(i)$. Then the following recurrence computes $S(i, j, k)$ for $1 \leq i \leq n$, $1 \leq j \leq m$ and $1 \leq k \leq b$:

$$S(i, j, k) = \max \begin{cases} S(i-1, j-1, k) + \Delta(g_i, t_j), & \text{if } i \neq \text{first}(k) \\ S(i-1, j, k) + \Delta_{\text{indel}}, & \text{if } i \neq \text{first}(k) \\ S(i, j-1, k) + \Delta_{\text{indel}}, & \text{if } i \neq \text{first}(k) \end{cases}$$

If $i = \text{first}(k)$, then we need to find another block from $\mathcal{B}()$ such that its last symbol lies strictly in front of $i$.

$$S(i, j, k) = \max \begin{cases} \max_{l \in \mathcal{B}(i)} S(\text{last}(l), j-1, l) + \Delta(g_i, t_j), & \text{if } i = \text{first}(k) \\ \max_{l \in \mathcal{B}(i)} S(\text{last}(l), j, l) + \Delta_{\text{indel}}, & \text{if } i = \text{first}(k). \end{cases}$$
The score of the optimal spliced alignment can be found as:

$$\max_k S(last(k), m, k).$$

Note that $S(i, j, k)$ is only defined if $i \in B_k$ and therefore only a portion of entries in the three-dimensional $n \times m \times b$ matrix $S$ needs to be computed, where $n = |G|$, $m = |T|$ and $b$ is the number of blocks.

Alltogether we have for the entries $S(i, j, k)$ to combine $n + 1$ values for $i$, $m + 1$ for $j$ and $b$ for $k$, making this $nmb$ expensive times $b$ for the second part of the recurrence where we take the maximum over $O(b)$ many blocks.

Hence, a naive implementation of the recurrence runs in $O(nmb^2)$ time.

### 12.17.9 Example

Consider the following string $G$ with the possible blocks indicated by boxes:

```
baa baa  black sheep have you any wool
```

The recurrence corresponds to the following graph:

```
\[ S(i, j, k) = \begin{cases} 
S(i-1, j-1, k) + \Delta(g_i, t_j), & \text{if } i \neq \text{first}(k) \\
S(i-1, j, k) + \Delta_{\text{indel}}, & \text{if } i \neq \text{first}(k) \\
S(i, j-1, k) + \Delta_{\text{indel}}, & \text{if } i \neq \text{first}(k) \\
P(i, j-1) + \Delta(g_i, t_j), & \text{if } i = \text{first}(k) \\
P(i, j) + \Delta_{\text{indel}}, & \text{if } i = \text{first}(k) 
\end{cases} \]
```

The target sequence is:

```
barbara sleeps on wool
```

Here are four spliced alignments of $G$ and $T$ obtainable from the above graph:

```
barbara sleeps on wool

baa baa black sheep any wool
baa baa black sheep have you any wool
baa baa sheep have you any wool
black sheep any wool
```

### 12.17.10 Speed up

The time and space requirements of the algorithm can be reduced significantly. Here we only discuss one such improvement.

Define $P(i, j) = \max_{l \in B(i)} S(last(l), j, l)$. The recurrence can be rewritten as follows:

```
S(i, j, k) = \begin{cases} 
S(i-1, j-1, k) + \Delta(g_i, t_j), & \text{if } i \neq \text{first}(k) \\
S(i-1, j, k) + \Delta_{\text{indel}}, & \text{if } i \neq \text{first}(k) \\
S(i, j-1, k) + \Delta_{\text{indel}}, & \text{if } i \neq \text{first}(k) \\
P(i, j-1) + \Delta(g_i, t_j), & \text{if } i = \text{first}(k) \\
P(i, j) + \Delta_{\text{indel}}, & \text{if } i = \text{first}(k) 
\end{cases} 
```
where

\[ P(i, j) = \max \left\{ P(i-1, j) \big\| \max_{l \in B(i): \text{last}(l) = i-1} S(i-1, j, l) \right\}. \]

With this modification, we maintain and update the maximal score for all preceding blocks explicitly and thus do not reconsider all preceding blocks in each evaluation of the recurrence.

The corresponding network that indicates which computations are performed looks like this:

12.17.11 Evaluation of the method

The authors of Procrustes evaluated the performance of the program on a test sample of human genes with known mammalian relatives. In their study, the average correlation between the predicted and actual proteins was 99%. The algorithm correctly reconstructed 87% of the genes.

They also reported that the algorithm predicts human genes reasonably well when the homologous protein is non-vertebrate or even prokaryotic.

Additionally, predictions were made using simulated targets that gradually diverged from the analyzed gene. For targets up to 100 PAM distance, the predictions were almost 100% correct. (This distance roughly corresponds to 40% similarity).

This indicates that for an average protein family the method is likely to correctly predict a human gene given a mammalian relative.

12.18 Summary of Gene prediction computational approaches

- Homology
  - Pure protein homology (TBLASTN)
  - Pure genomic homology (TBLASTX, Exofish)
  - Comparative (Procrustes)

- Ab initio
  - Pure ab initio (E.g., Genscan, Genie)
  - Ab initio + protein homology
    * Align first approach: Genomescan
    * Pair HMM approach: Genewise
  - Ab initio + genomic homology
    * Align first approach (E.g., Twinscan, sgp2)
    * Pair HMM approach (E.g., SLAM)