2 General Introduction to Computational Methods Used in this Thesis

2.1 The Application of Bayesian Methods in Sequence Analysis

Bayesian analysis (Grate *et al.*, 1996), a general class of stochastic modelling techniques based on Bayes' theorem of conditional probability (Equation 2.1), represent an important approach for studying biological sequences. The idea is to construct a model that describes a set of sequences. The model can then be used to find a set of related sequences or examined further to determine properties of the sequences. A model in this case can be described as a "black box" which does not necessarily represent a "real world" mechanism. The model's value depends solely on the accuracy of its predictions and not by the mechanism used to make those predictions.

Equation 2.1: Bayes' theorem of conditional probability. In the context of biological sequence analysis, *M* represents a Bayesian model and *s* a DNA or protein sequence.

$$P(M \mid s) = \frac{P(s \mid M)P(M)}{P(s)}$$

Bayes' theorem (Equation 2.1) is based on the idea that in many situations, an analysis can be commenced with an estimated prior probability for an event of interest. This probability can come, for example, from historical data or previous experience. The idea is to receive additional information such that the prior probabilities in Equation 2.1 can be updated. The updated probabilities are referred to as the posterior probabilities.

In Equation 2.1, above, one of two conditional probabilities to update is *P(M|s)*. This probability value answers the question "Given the sequence *s*, what is the probability that it came from the distribution described by *M*?". The other conditional probability to update is *P(s|M)*, which is the probability of the sequence *s* given *M*. Two prior probabilities are required to estimate these values: *P(M)*, the

probability that *s* is drawn from model *M* and *P(s)*, the probability of the sequence *s*. It is not possible to know the real probabilities of *P(M)* and *P(s)* but a different approach can be used to overcome this. The approach is to calculate the odds that the sequence *s* came from model *M* rather than a null model *N* (Equation 2.2). As can be seen from Equation 2.2, *P(s)* is no longer required. The model probabilities *P(M)* and *P(N)* can be estimated using iterative training methods (the procedure for hidden markov models is described in Section 2.2.3).

Equation 2.2: Relative probability of model M and the null model N.

$$\frac{P(M \mid s)}{P(N \mid s)} = \frac{P(s \mid M)P(M)}{P(s)} \times \frac{P(s)}{P(s \mid N)P(N)} = \frac{P(s \mid M)}{P(s \mid N)} \times \frac{P(M)}{P(N)}$$

The null model defines what the null hypothesis is. Choosing a good null model is a tricky problem and depends on the problem at hand. A sequence *s* can then be said to fit model *M* if $P(M|s) > P(N|s)$. Usually, this result is scored in log values and the value $\log P_M(s) - \log P_N(s)$ is referred to as the log-likelihood of the sequence. In practice, a threshold score is chosen: the higher the log likelihood score is than the threshold, the greater the confidence in the result. Bayesian methods have been used in this thesis in Chapters 3 and 5.

2.2 Hidden Markov Model Theory

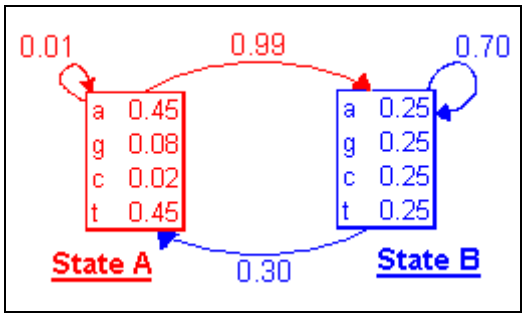
2.2.1 A general introduction to hidden markov models

Hidden Markov Model (HMM) analysis has widespread applications in Bioinformatics particularly in DNA and protein sequence analysis. These include creating multiple alignments of sequences to model protein families (Bateman *et al.*, 2002) and gene prediction (Meyer & Durbin, 2002). HMMs have also found importance as a pattern discovery tool; an example was seen recently where it was used to learn local composition patterns from chromosome 2 in the malarial genome *P. falciparum* and use that information to predict corresponding features in chromosome 3 (Pocock MR *et al.*, 2000). It has also been used as a discovery tool to find patterns that could be involved in nucleosome rotational positioning (Baldi *et al.*, 1996). This approach used a special kind of HMM referred to as the cyclical HMM. In this thesis, this approach has been extended to try to gain further insights into the patterns which were originally reported using cyclical HMMs: this is the focus of Chapter 3. This section will briefly introduce some basic HMM terminology and then introduce two algorithms which were used in this thesis for HMM prediction and training respectively (Sections 2.2.2, 2.2.3).

- **HMM terminology**

A hidden markov model (HMM) is in essence a vector of "states" connected with "transition paths"; each state contains 2 kinds of probability distributions associated with it: an emission spectrum and a transition spectrum respectively. Figure 2.1 shows a HMM which has an architecture of 2 states connected by a number of transitions.

Figure 2.1: A 2-state hidden markov model which emits symbols from the DNA alphabet. Boxes represent states and arrows represent transitions. The emission and transition distributions for State A are shown in red; State B's corresponding distributions are shown in blue.



To model a specific kind of sequence with a HMM, it is first necessary to define the alphabet from which that sequence is composed; this alphabet is called the "emission alphabet". To model DNA sequences with a HMM, for example, it needs to be defined that DNA is composed of an emission alphabet of 4 symbols, "a,c,g,t".

The HMM shown in Figure 2.1 is a 2-state HMM, based on the DNA alphabet. *State A* has a strong probability of emitting "a" (0.45) or "t" (0.45) and a much weaker probability of emitting "g" (0.08) or "c" (0.02). *State A* has 2 transition paths out of it: one path to *State B* and one path back to itself. These paths form the transition spectrum of *State A*. In this case, it has a weak transition probability of going back to itself (0.01) and a strong transition probability of going to *State B* (0.99). *State B* has a random emission distribution (each symbol emitted at equal probability) and a set of 2 transitions (0.70 probability of going back to itself and 0.30 probability of going to *State A*). The entire set of emission and transition probabilities in the HMM define the HMM's parameters. This model can be used to score a sequence; this score is usually the product of all the emission and transition probabilities in the "path" of the model in that sequence (described below).

Figure 2.2: 2 DNA sequences which are likely to receive a high score and a weak score respectively with the model of Figure 2.1. The locations of [W] regions are underlined.

(a) Possible High Scoring Sequence:

GAGCCGGCCGGGGGCCCGGGCCCGGGCTCGGGGACCCGCCCCCTCGCCCCAACCGCGG

(b) Possible Low Scoring Sequence:

AAAACCCTTAAAAATTTCGGGCCCTTTTTCCCTGTTTAAACGGTCCCTATTTACCCGG

To introduce HMM paths and HMM-based scoring, the 2 sequences in Figure 2.2 are considered. The first assumption is that the sequences in Figure 2.2 have been generated by the states of the HMM of Figure 2.1. But it is not known which part of the sequence was emitted by *State A* or *State B*; this is a "hidden" path from which the "hidden" term of HMMs is derived. However, it can be guessed that the sequence of Figure 2.2(a) was more likely to have been produced by a path through the HMM than the second sequence (Figure 2.2(b)). This is firstly because *State A*, whose emission spectrum represents [W][9] motifs, has only a weak transition probability of going back to itself but a strong transition probability of going to *State B* (whose emission spectrum is random). Secondly, *State B* has a stronger probability of going back to itself compared to going back to *A*. This means that the HMM is more likely to spend more of its "energy" in *State B* than in *State A*. It effectively makes this HMM a model or predictor for sequences which display "short spurts" of [W] (*State A*) compared to a random background (*State B*). A path through the HMM which could have produced the sequence in Figure 2.2(a) could be as shown in Figure 2.3.

[9] Please refer to the ambiguity symbols for DNA at the beginning of the thesis

Figure 2.3: (a) A possible path through the HMM which could have emitted (b) the corresponding DNA sequence.

```
 (a) Possible path through the HMM:

BABBBBBBBBBBBBBBBBBBBBBBBBBBBABBBBBBABBBBBBBBBBABBBBBBBBB

(b) DNA sequence:

GAGCCGGCCGGGGGCCCGGGCCCGGGCTCGGGGACCCGCCCCCTCGCCCCAACCCCCA
```

An algorithm for predicting the hidden path of states is described next.

2.2.2 Predicting the most likely path of a HMM through a sequence using the Viterbi algorithm

The Viterbi algorithm can be used to predict the most probable path, $\Pi_{(a)}$, through a HMM's states that could have emitted a given sequence. It uses a "dynamic programming" matrix where the columns are indexed by the states of the HMM, S, and the rows are indexed by the position x_i of the sequence X. The algorithm is outlined below using the following notations (Karchin, 1999; Shamir, 2001):

A general hidden markov model (HMM) is defined as $M=(A,S,Y)$ where:

- A = finite set of symbols (also called the emission alphabet).
- S = finite set of emission states.
- Y = finite set of probabilities comprised of:
    - State transition probabilities, denoted by t_{kl} for each $k,l \in S$.
    - Emission transition probabilities, denoted by $e_k(b)$ for each $k \in S$ and $b \in A$.

A sequence X, of length L, is defined whose positions are indexed as $(x_1,...,x_i)$. $v_k(i)$ is denoted as the probability of the most probable path for the sequence that ends with state k ($k \in S$ and $1 \leq i \leq L$). $\Pi_{(a)}$ is found using the following steps:

- **Initialization:**

$v_{begin}(0) = 1$

For all $k \neq begin$, $v_k(0) = 0$

- **Recursion:**

For each $i = 0, \ldots, L-1$ and for each $l \in S$ the following is calculated recursively:

$$v_l(i+1) = e_l(x_{i+1}) \cdot \max_{k \in S} \{v_k(i) \cdot t_{kl}\}$$

During each recursive step, a backpointer is assigned from l back to the k.

- **Termination:**

$$P(X \mid \prod_{(a)}) = \max_{k \in S} \{v_k(L) \cdot t_{k,end}\}$$

- **Path Reconstruction:**

$\Pi_{(a)}$ is found by re-tracing the backpointers.

2.2.3 Training a HMM using the Baum Welch algorithm

The HMM, shown in Figure 2.1, can be used to score any DNA sequence, for example by obtaining the Viterbi score, $P(X|\Pi_{(a)})$, as explained above. But the parameters of the HMM itself, Y, may not be realistic. To obtain realistic probabilities, it is necessary firstly to obtain a set of related sequences which contain a known motif or a set of known motifs. These sequences form the training set, $X_{(1)},\ldots,X_{(n)}$, from which Y must be "learnt" or "trained". Training is an iterative process which keeps refining the parameters of the HMM to obtain an optimal score for $X_{(1)},\ldots,X_{(n)}$ denoted as $Score(X_{(1)},\ldots,X_{(n)}|Y)$. The Baum Welch algorithm is one such training algorithm, which was used in this thesis.

Before the Baum Welch algorithm can be introduced, it is important to point out that the individual statepaths of the HMM, $\Pi_{(1)},\ldots,\Pi_{(n)}$, which produced $X_{(1)},\ldots,X_{(n)}$

are unknown. The Baum Welch procedure has a step to overcome this. The step involves computing the probability of every statepath $\Pi_{(i,j)} = (\pi_{I(i,j)},..,\pi_{L(i,j)})$ for every $X_{(j)}$ in $X_{(1)},...,X_{(n)}$. These probabilities, $P(\pi_{(i,j)} = k|X_{(j)})$, can be calculated using the forward and backward algorithms which are outlined first:

Forward algorithm (outlined for a single sequence *X*):

The parameter $f_k(i)$ denotes the probability of emitting X using the statepath $\pi_i = k$.

- **Initialization:**

$f_{begin}(0) = 1$

For all $k \neq begin$, $f_k(0) = 0$

- **Recursion:**

$$f_l(i+1) = e_l(x_{i+1}) \cdot \sum_{k \in S} f_k(i) \cdot t_{kl}$$

- **Termination:**

$$P(X) = \sum_{k \in S} f_k(L) \cdot t_{k,end}$$

Backward algorithm:

The Backward algorithm works in exactly the same way as the forward algorithm except it is computed backwards from the end of X. The parameter $b_k(i)$ denotes the backward probability of emitting X using the statepath $\pi_i = k$.

Finally, it can be shown that $P(X, \pi_i = k) = f_k(i) \cdot b_k(i)$ (Shamir, 2001).

Baum Welch algorithm:

- **Initialization**

Y is initialized with reasonably-guessed parameters. For work done in this thesis, all $e_k(b)$ were initialized randomly and a reasonable guess was made for t_{kl}.

- **Expectation**

The probabilities $P(X_{(i,j)})$ for every statepath $\Pi_{(i,j)}$ for all $X_{(1)},\ldots,X_{(n)}$ is calculated as above.

The following 2 parameters can now be estimated:

- T_{kl} – the number of transitions from state k to state l.
- $E_k(b)$ – the number of times that an emission of the symbol b occurred in state k.

These are estimated as follows:

$$T_{kl} = \sum_{j=1}^{n} \frac{1}{P(X_{(j)})} \cdot \sum_{i=1}^{L_{(j)}} f_{k(j)}(i) \cdot t_{kl} \cdot e_l(x_{i+1(j)}) \cdot b_{l(j)}(i+1)$$

$$E_k(b) = \sum_{j=1}^{n} \frac{1}{P(X_{(j)})} \cdot \sum_{\{i|x_{i(j)}=b\}} f_{k(j)}(i) \cdot b_{k(j)}(i)$$

- **Maximization**

The new values of Y are estimated from T_{kl} and $E_k(b)$. These are estimated using maximum likelihood estimators for the transition and emission probabilities respectively. The maximum likelihood estimators are:

$$a_{kl} = \frac{T_{kl}}{A_{q \in S} A_{kq}}$$

$$e_k(b) = \frac{E_k(b)}{A_{a \in A} E_k(a)}$$

- **Terminaton**

Steps 2 and 3 are repeated until the improvement in $Score(X_{(1)},\ldots,X_{(n)}|Y)$ is less than a given parameter ε.

2.3 The Use of Flexibility Sequences

2.3.1 An Introduction to flexibility sequences

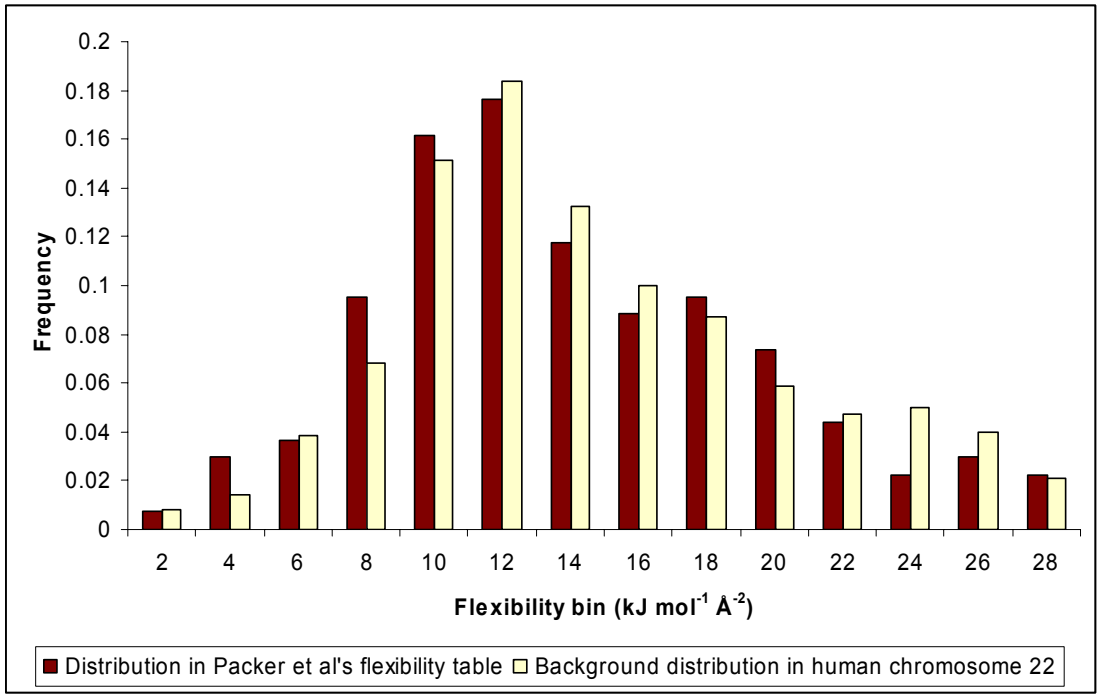
One of the fundamental concepts of nucleosome positioning is that it is an effect of the physical properties of the underlying DNA sequence. This made it necessary to model DNA sequences as sequences of physical DNA parameters. This section will introduce these kinds of sequences, herein referred to as "flexibility sequences". The flexibility sequences described in this section was used for wavelet analysis (discussed in Section 2.4.1). Section 2.3.2 will introduce a simpler kind of flexibility sequence for using as emission symbols for HMMs.

For the work carried out in this thesis, a table which provides flexibility values for all 256 possible tetranucleotide steps (4^4 combinations) (Packer *et al.*, 2000b) was used to translate a given DNA sequence into its corresponding flexibility sequence. According to these studies, certain dinucleotide steps, represented within the larger tetranucleotide steps, were 'sequence-independent'. Their conformation appears to be constant regardless of neighbouring sequences; an example of this is [AA/TT] whose physical basis was discussed earlier (Section 1.4.1). At the other extreme, sequences such as [CA/TG] are 'sequence-dependent' as their conformation is strongly influenced by the immediate DNA sequence context. This is why a tetranucleotide-based flexibility table was used rather than a lower di- or tri- nucleotide based flexibility table since it would be able to model the contexts of the sequence-dependent dinucleotides slightly better.

The parameters in this table were estimated using force field measurements, which are mathematical formulas for expressing energy as a function of physical conformation (Sprous, 1996). Such functions are usually sums of terms which

correspond to bond angle, torsion, Van der Waals forces and electrostatic interaction energies. These parameters correlated reasonably well with the limited tetranucleotide parameters available from X-ray crystallography (Hunter & Lu, 1997; Packer *et al.*, 2000b). The values in the flexibility table range from 1.9 (most flexible) to 27.2 (most rigid) and there are a total of 102 unique flexibility values. As can be seen from Figure 2.4, the distribution of the flexibility values is negatively skewed in both the flexibility table and in background human genomic DNA. Those tetranucleotide sequences which exhibit the highest rigidity generally contain [AA/TT] dinucleotides.

Figure 2.4: Histogram of DNA flexibility values (Packer *et al.*, 2000b)



A DNA sequence was converted to this kind of flexibility sequence using the following steps:

- A 4 bp window was taken at position 1 of the DNA sequence.
- Its corresponding flexibility value was looked up and stored as the first symbol of the flexibility sequence.

- The window was shifted by 1 bp and the next value looked up; this was stored as the second symbol of the flexibility sequence.
- Steps 2-3 were repeated until reaching 3 bp from the end of the DNA sequence.

2.3.2 Flexibility emission alphabet for using with HMMs

A simple flexibility emission alphabet was derived from the tetranucleotide-based flexibility table described above for using with HMMs. In the original form of this table, 102 unique symbols would have been an exhaustive emission alphabet for HMM training (compare with 4 symbols for the DNA alphabet for example). Therefore, the number of symbols had to be sized down to form a reasonable emission alphabet. This was done by firstly splitting the 256 unique tetranucleotide sequences into 6 equally binned categories ranked by ascending values of flexibility. Each of the 6 bin categories represented a symbol of the new compressed alphabet: these new symbol values were assigned from 1 for most flexible to 6 for most rigid. So for example, the 'most flexible' category would contain the 42 (256/6) most flexible tetranucleotide sequences of the original table. In this way, a compressed 6-symbol flexibility lookup table for tetranucleotide DNA sequences was derived. This table was used to convert a DNA sequence into its corresponding 6-symbol flexibility sequence using the same steps outlined in Section 2.3.1.

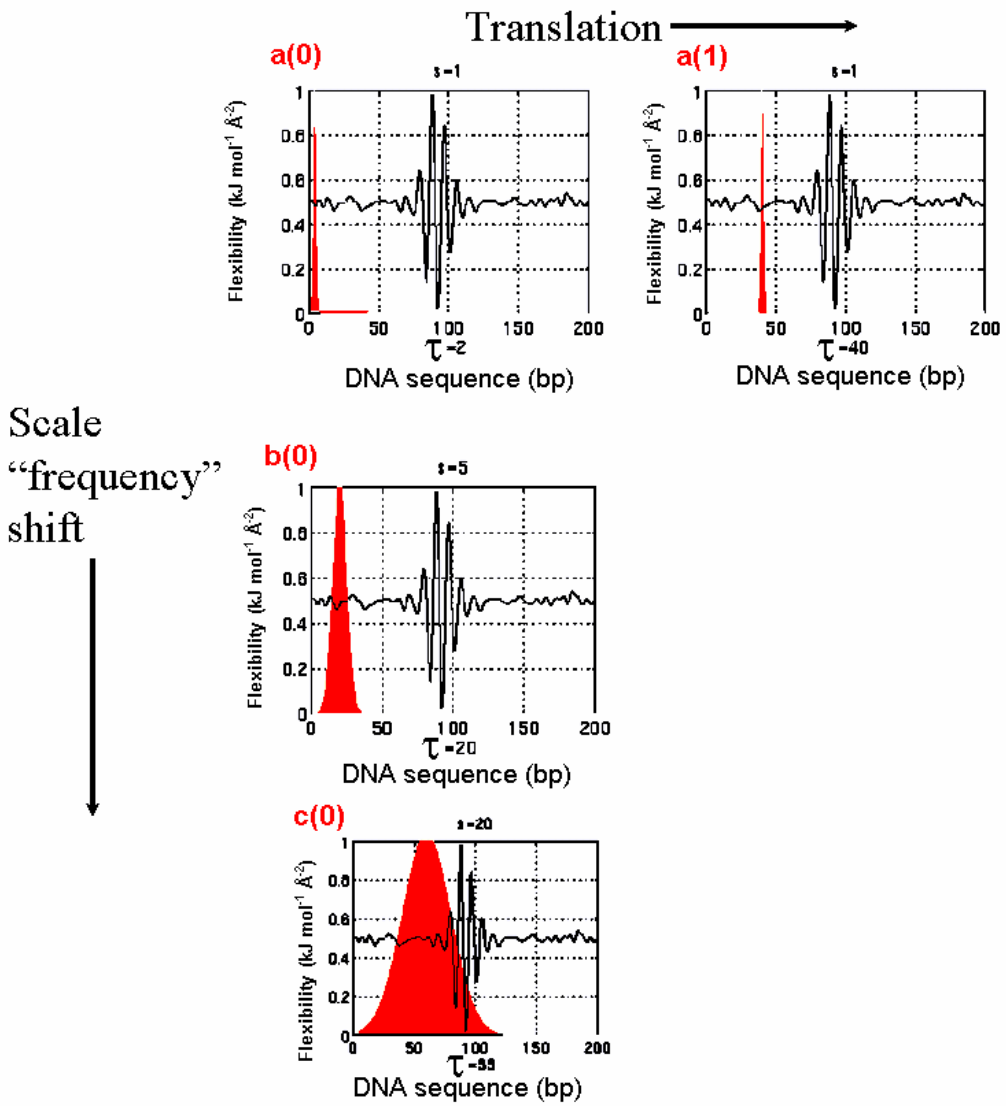
2.4 A Basic Introduction to Wavelets

2.4.1 An introduction to wavelets

Wavelets are a family of mathematical transformations which reveal information about the strength and localisation of periodic patterns in a signal; this information is not apparent in the raw format of the signal. A DNA sequence can be considered as a specific kind of signal. The flexibility sequence is another representation of the same signal but from which it is easier to derive information about the sequence of structural features in the DNA sequence. There are 2 parameters which define a wavelet (Figure 2.5):

- Translation (τ) which defines a specific position along a signal and
- Scale (s) which defines a specific frequency.

Figure 2.5: The concept of translation and scale in wavelet terminology. This figure is a slightly modified version of a figure from Robi Polikar's 'Introduction to Wavelets' online tutorial (Polikar, 2000).



In Figure 2.5a(0), the wavelet function is seen as a red sine curve; it is located at its initial position 2 (the value of τ) along the DNA sequence and with a scale parameter of 1 (the value of s). This is the wavelet function at its original position and is called the mother wavelet. The following shifts in size and location are then applied to the mother wavelet:

- Firstly, the function is moved or 'translated' along a sequence to scan for any localised frequencies which correspond to the present value of s = 1 (Figure 2.5a(1)). In Figure 2.5a(1), the function has been shifted to a τ value of 40. τ = 80 will receive a high score at this present s value as it is very similar in size and shape to the current value of s. In this way, a score is obtained for each point along the DNA sequence which represents how strongly correlated the part of the sequence is to the present shape and size of the wavelet function.
- The scale parameter, 's', is now 'dilated' to 5 (Figure 2.5b(0)) increasing the width of the function. It is also translated across the sequence to obtain a score for each point along the DNA sequence. One important feature is that since the scale has increased, the resolution along the 'x' axis has also diminished. This is a property of multiresolution which is explained in the next section. Note that the initial τ value is now at 20 which is due to the increase in width of the wavelet function.
- In Figure 2.5c(0), 's' is further dilated to 20. In this way, a number of co-efficient scores are obtained for different values of 's' and τ. The results can be plotted as a 2D contour map as in Figure 4.2 (page 4-122), where the intensity of the colours represent the strength of different frequencies in different regions of the DNA sequence (dark blue is strongest).

Equation 2.3 is the formula for the continuous wavelet transform. For different values of τ and s, the wavelet function is obtained as the product of the original sequence, x(t), and the wavelet function. This product is referred to as the convolution of the signal and the wavelet function; it is analogous to a correlation co-efficient between the wavelet function and a specific region of the signal. The

convolved product is further multiplied by a normalisation factor $1/\sqrt{|s|}$, which ensures that the energy of the co-efficients is distributed evenly along different scales.

Equation 2.3: Continuous wavelet transform

$$CWT_x^{\psi}(\tau, s) = \Psi_x^{\psi}(\tau, s) = \frac{1}{\sqrt{|s|}} \int x(t)\psi^* \left(\frac{t-\tau}{s}\right) dt$$

2.4.2 The multiresolution property of wavelets

The output from a wavelet transform provides a 2 dimensional representation where the strengths of different frequencies against a DNA sequence can be viewed. However, an important feature with this kind of transformation is the multiresolution property. This states that high frequency components are resolved well in time and low frequency components are resolved well in frequency. As can be seen in Figure 2.6, as the frequencies get higher, the width of the boxes get narrower; thus this value can be resolved well along the DNA sequence. The reverse is true for low frequencies which will be resolved poorly along the DNA sequence but better along the frequency axis; this is seen as the wide box at the bottom of the frequency axis.

Figure 2.6: The multiresolution property of wavelets. The x and y axes represent increasing values along the DNA sequence co-ordinates and frequency values respectively.