Chapter 2

The pair HMM underlying DOUBLESCAN and PROJECTOR

2.1 Introduction and motivation

It is clear from previous comparative studies discussed in Section **1.4** that we cannot reliably infer the gene structures within two **DNA** sequences from a set of matching subsequences only. In order to comparatively predict gene structures, we would like a method which

- is symmetric with respect to the two input sequences
- keeps track of a valid splicing pattern simultaneously in each of the two **DNA** sequences
- does not assume that the input sequences contain a certain gene structure, for example one single complete gene
- can make use of different notions of similarity such as similarity at protein level as well as similarity at **DNA** level
- is able to incorporate information about sequence signals such as splice sites

As we wanted to see if two related **DNA** sequences would enable us to predict genes in an ***ab initio*** way, the method should

- use the two **DNA** sequences as the only input information and should in particular not have to know the amino-acid sequences they encode or how the two **DNA** sequences should be aligned.

The mathematical concept of pair hidden Markov models is well suited for achieving all of the above aims. It can treat each of the two input sequences on an equal footing. The pair HMM's states and transitions can be defined to enforce a valid splicing pattern in each of the two DNA sequences and to enable the prediction of a variety of different gene configurations which is not limited to predicting single complete genes. The different notions of similarity can be incorporated into the emission probabilities of the pair HMM's states. The strength of a variety of sequence signals such as translation start sites, splice sites and other functional elements can be translated into scores which are then used within the pair HMM to modify the nominal values of the transition probabilities. This is a generalisation of the standard form of pair HMMs as introduced in Section 1.5.1 which facilitates the efficient treatment of sequence dependent scores (see Chapter 6 for details).

In developing the states and transitions of the pair HMM underlying both DOUBLESCAN and PROJECTOR, we want the gene prediction to be mainly guided by the similarity information between the two DNA sequences. The different types of conservation between the two DNA sequences should, together with the constraint to produce a valid splicing pattern as imposed by the architecture of the pair HMM, enable the simultaneous comparative prediction of pairs of related genes.

The pair HMM can distinguish two different types of conservation as shown in Figure 2.1. The patterns of conservation between two DNA sequences can be different even if the overall percent identity between the two sequences is the same. It is this pattern of conservation which is used in order to distinguish conserved protein coding DNA from conserved non protein coding DNA. If the pattern of conservation has a three base pair periodicity and if the bases of the DNA can be grouped into triplets which could be interpreted as codons encoding the same or a chemically similar amino-acid, the DNA is likely to be protein coding and not protein coding otherwise.

We try to keep the number of assumptions on how a gene in isolation should look like, to a minimum of biologically well motivated assumptions and rather focus on implementing assumptions on the similarities which two related genes should exhibit. In particular we refrain from explicitly modelling the length distributions of exons and introns within a gene or the number of exons within a gene. Instead, we implement the assumption that two related genes should encode similar sequences of amino-acids which should be distributed onto the same or a similar number of exons of the same or similar length. This approach enables us

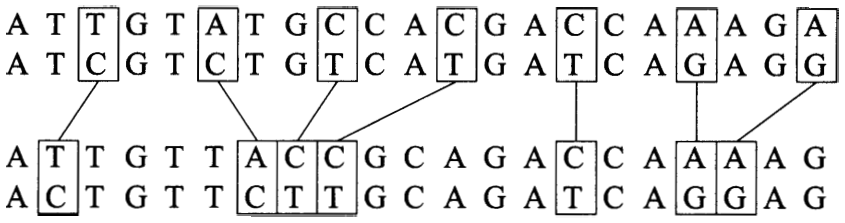

Figure **2.1:** Two alignments with the same percent identity, but different types of conservation. Boxes show un-conserved bases. The upper alignment exhibits a conservation pattern with a periodicity of three bases indicating pairs of codons which encode the same or a similar amino-acid, whereas the lower alignment shows no apparent pattern of conservation. The lower alignment is related to the upper alignment by permutations of the columns of aligned bases.

not only to detect novel genes whose amino-acid sequence is not yet known, but also to detect pairs of unusual genes. The main reason for choosing this approach is that genes, however unusual they might be, should be similarly unusual in a related organism and should therefore be detectable by our comparative method.

In the following, we first describe the pair HMM of DOUBLESCAN and PROJECTOR, its states and transitions, then explain how the parameters of the model are derived and conclude with a presentation of a new algorithm by which gene predictions are generated with essentially linear time and memory requirements.

2.2 States and transitions of the pair HMM

The aim in defining the states and transitions of the pair HMM is to be able to capture the most important configurations which can arise from the generic alignment of two homologous genes, see Figure 2.2. Suppose that one of the two **DNA** sequences, the first sequence in Figure **2.2,** comprises all exons and introns which correspond to one protein. The homologous gene in the second **DNA** sequence, originating from the same or a different organism, may contain the same number of introns (a), an additional intron (b) or one intron less *(c).* It can thus happen that the level of similarity between two **DNA** sequences is not high, even though they encode very similar amino-acid sequences.

The states and transitions of the pair HMM were defined so that the exons of two homologous genes can be aligned even if the genes are related by events of exon-fusion or exon-splitting. The pair HMM consists of **54** states. Every state of the pair HMM classifies every letter it

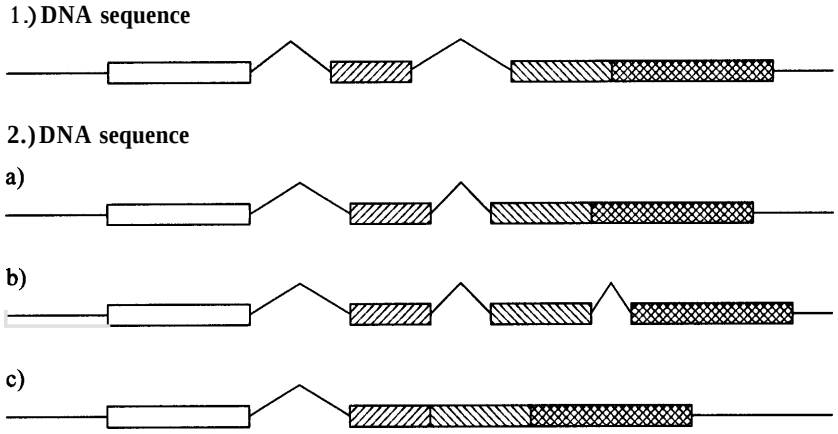

Figure **2.2:** Different pairs of homologous genes. Boxes represent exons, kinked lines introns and straight lines intergenic sequences. Similar exons have the same hatching.

reads into four mutually exclusive classes: intergenic, non protein coding exon, intron and exon. Every ***match*** state reads the same number of letters from both sequences and assigns them to the same class, say exon DNA. We do not expect two homologous DNA sequences to exhibit the same features in the same length: even though the two encoded amino-acid sequences may be very similar, they may not have exactly the same length, see Figure **2.3.** And we expect the non protein coding subsequences, e.g. introns and intergenic regions, to be more diverged than the exons. To be able to align two subsequences of the same class, but of different length, in addition to the ***match*** state we need two corresponding ***emit*** states which read non-matching letters from only one sequence at a time.

Even though the pair HMM can deal with the prevailing configurations which arise from the generic alignment of pairs of homologous gene structures, some configurations cannot be modeled with the pair HMM and would require the introduction of extra transitions or states into the pair HMM. To name just two examples which the pair HMM cannot model: **(1)** a pair of homologous genes in which one exon in one gene exhibits no homology to any exon of the other gene, **(2)** a pair of homologous genes in which the pairs of homologous exons do not appear in collinearity. In the default implementation of the model, only the pair of input DNA sequences, but not the pair of their corresponding reversecomplemented sequences, is analysed. The analysis of the reversecomplemented sequence pair requires a separate run. Simultaneous search for genes in both orientations could be obtained by essentially doubling the number of states in the pair HMM.

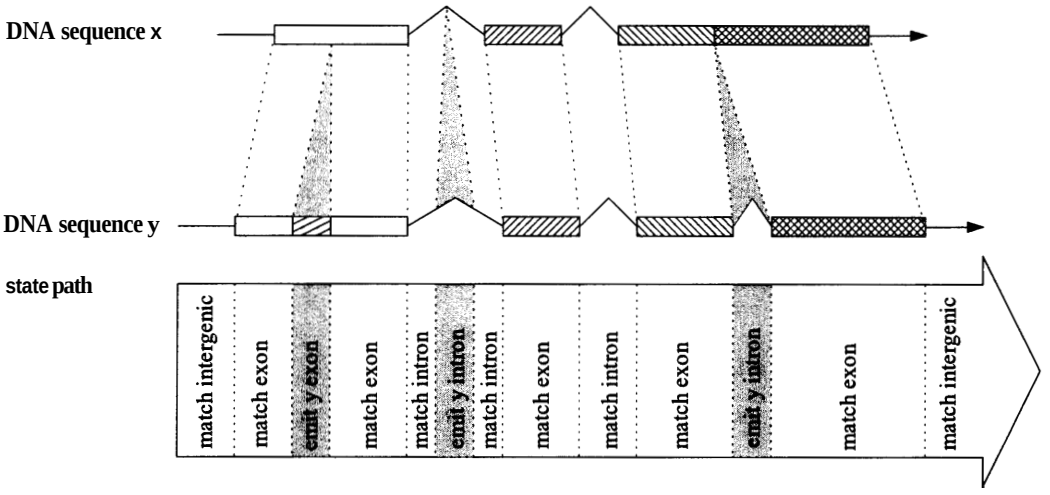

Figure **2.3:** Sample alignment of two annotated **DNA** sequences with a possible state path. Boxes represent exons, kinked lines introns and straight lines intergenic sequences. Similar exons have the same hatching.

We now explain the different sets of states:

begin* and *end* states** By definition, each state path begins in the ***begin state and ends in the ***end*** state. Both states are silent, i.e. they do not read any letters, and are used exactly once in every state path.

As we do not want to make assumptions on the annotations with which the two sequences start or end, the ***begin*** state is connected to every other state except for the ***end*** state. Likewise, the ***end*** state can be reached by every other state except for the ***begin*** state. The pair HMM can thus not only predict single complete genes, but also partial genes, no genes, multiple genes and other configurations of gene structures.

START START* and STOP *STOP* states** We want to align pairs of genes and thus have a one-to-one correspondence between the start and the stop codons of the genes in the two **DNA** sequences. The start codons of the pair of initial exons can be aligned using the ***START START state and the stop codons of the terminal exons can be aligned using the STOP ***STOP*** state.

All potential start codons, i.e. all ATG triplets, are scored using a weight matrix model of **21** base pairs width that starts 9 base pairs 5' to the potential start codon.

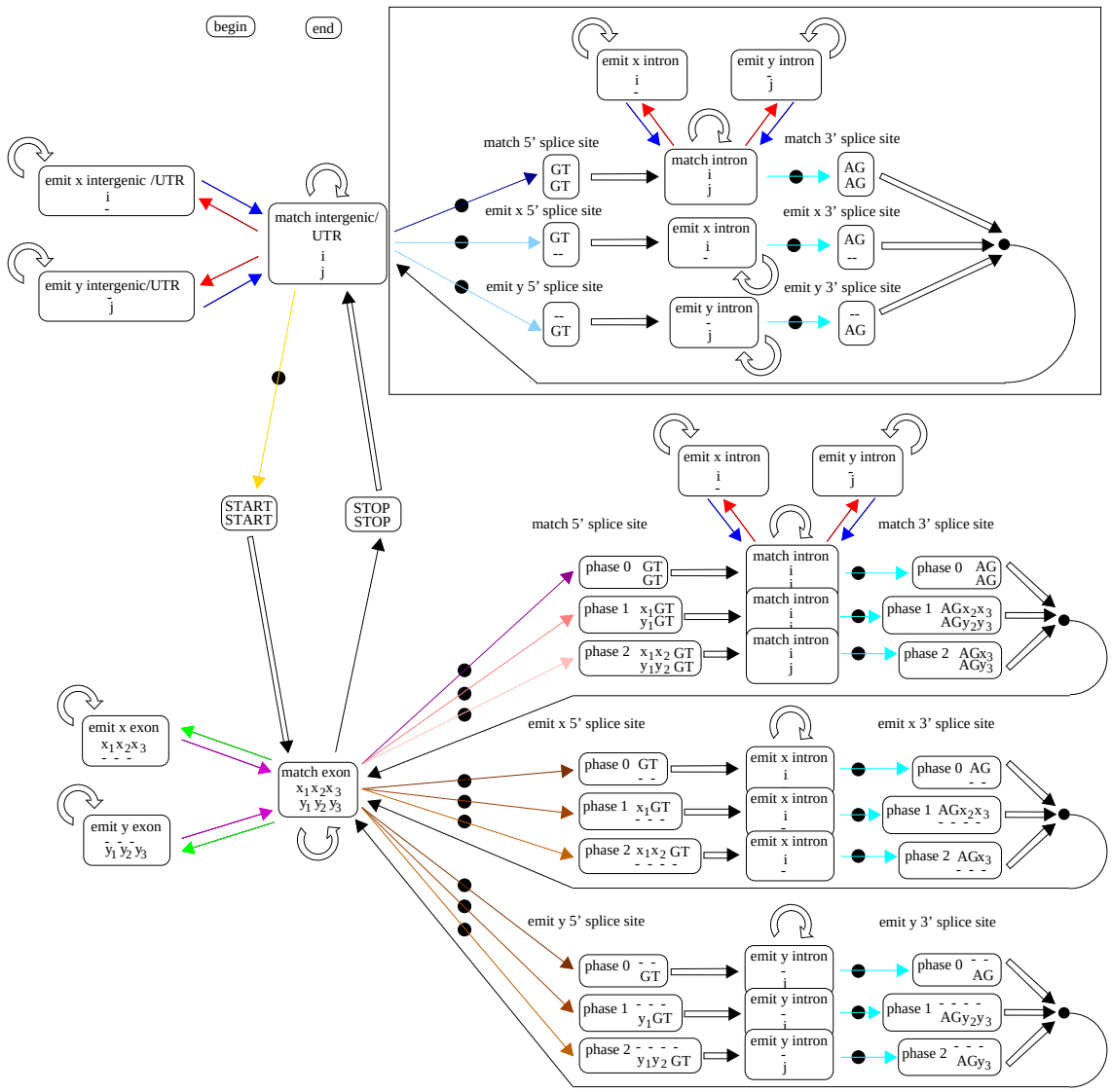

Figure 2.4: States and transitions of the pair HMM underlying DOUBLESCAN and PROJECTOR. States are shown as boxes with rounded corners, transitions as arrows. The *begin* state is connected to every state except for itself and the *end* state. Likewise, there are transitions to the *end* state from every state except for the *begin* state and itself. The arrows corresponding to these transitions are not shown for clarity. Each open arrow corresponds to a transition probability which is defined by the constraint that the probabilities of the transitions emerging from every state have to add up to one. Coloured arrows of the same colour correspond to transitions of the same probability. Arrows marked by a black dot are special transitions, see Section 2.3. The *match intron* states for phase 1 and 2 are connected to the corresponding *emit x* and *emit y* intron states as is the *match intron* state for phase 0. These states have been omitted for clarity (see Figure B.3 in Appendix B). The large box at the top right contains the states which model introns within untranslated regions (UTR-splicing).

Exons We expect evolutionarily related genes to encode similar amino-acid sequences which at **DNA** level correspond to similar sequences of codons. These codons *can* be aligned using the *match* exon state which reads one codon from each of the two DNA sequences at a time. The emission probability of the match exon state for aligning two codons which encode the same amino-acid is high compared to that for aligning two codons which encode chemically dissimilar amino-acids. The wobble position, the last (most **3')** position in a codon, has thus less importance in defining the level of similarity between two codons than the first (most **5')** codon position.

Closely related genes which encode similar proteins need not have the same number **of** amino-acids and thus need not correspond to the same number of codons at **DNA** level. This motivates the definition of the emit x exon and the emit y exon states which read a codon from only one **DNA** sequence at a time.

Closely related genes which encode similar proteins may not only have a different number of amino-acids, but these amino-acids may also be encoded on a different number of exons. These pairs of genes which are related by events of exon-fusion or exon-splitting *can* be aligned using the sets of emit x and emit y states of splice site and intron states.

Note that all match and emit exon states can read in-frame **ATG** codons encoding methionine, but that their emission probability for reading any of the three stop codons in frame is zero.

Splice sites and introns within translated regions Introns within protein coding regions can come in three different phases depending on where they are inserted into the codons. **As** we want to be able to align genes which are related by events of exon-fusion or exon-splitting, we have to take into account introns which are present in only one of the two genes. These introns can be modelled using the emit x or emit y sets of splice site states and intron states.

In the default implementation of the model, all splice sites are assumed to obey the **GT-AG** rule, stating that an intron should start with a **GT** at the 5' side and end with an **AG** at the 3' side. This rule accounts for **99** % of introns in the set of known mammalian **DNA** sequences **[BSSOO]. All** potential splice sites of the input **DNA** sequences are scored by a splice site prediction program [LD01] similar to that used in [BK97].

Splice sites and introns within untranslated regions (UTR-splicing) **A** special feature of our model is that it allows for introns within the untranslated regions of genes using a

set of states similar to those for introns within translated regions. The states for UTR-splicing are shown within the box in Figure 2.4. The main reason for introducing introns within the untranslated regions is the observation that the model without them has difficulties to detect start codons properly. Some start codons were missing in the predictions and were hidden within internal exons. As there are true splice sites also to be found within the untranslated regions and as all potential splice sites – also those within the untranslated regions – are scored by the splice site predictor, the model without UTR-splicing had no means of selectively ignoring the high scoring splice sites within the untranslated regions and of taking only those within the translated regions into consideration. The addition of the UTR-splicing states handles this better and helps to detect both start and stop codons.
Unlike introns within translated regions, introns within untranslated regions do not have a phase. As for introns within protein coding regions, all splice sites are by default assumed to obey the GT–AG rule and are scored by the splice site prediction program.

Intergenic/UTR states We put the least constraints on the intergenic/UTR subsequences even though we know that they can have a rich functional structure, comprising for example promoters and sequences which bind molecules which determine the three-dimensional structure of the DNA sequence. We do not attempt to model these features with this pair HMM, as the ability to predict them is poor. If these functional elements are conserved, they will be predicted as conserved intergenic/UTR subsequences and they can be further investigated.

2.3 Parameters of the model and their determination

The parameters of the pair HMM can be subdivided into transition and emission probabilities. While the transition probabilities are the same for the different pairs of genomes investigated in this dissertation (with one exception, see Table D.1 in Appendix D), the emission probabilities are adapted to the pair of related organisms that is studied and are derived from a *training set* of known genes.

Transition probabilities and splice site scores Non-zero transition probabilities are represented by arrows in Figure 2.4. The *begin* state is connected to every other state except the *end* state. Likewise, the *end* state can be reached by all other states except the *begin* state. The corresponding arrows have been omitted for clarity. Every open arrow corresponds

to a transition probability whose value is defined by the constraint that the probabilities of the transitions which emerge from every state must add up to one.

As we assume that there is no systematic bias in the number of exons or the length of exon, intron and intergenic sequences between the two organisms from which the two DNA sequences derive, the transition probabilities of the emit **x** states are the same **as** those of the corresponding emit y states.

As opposed to the emission probabilities which are derived from the training set, there is not straightforward way to derive the values of the transition probabilities. First, the training sets are generally too small to reliably estimate the transition probabilities, e.g. the probability for the transition from the match intergenic to the ***START START*** state, from the corresponding frequencies in the training set. Second, the probabilities for transitions between match and emit states can only be derived from a training set of pre-aligned sequences which is not available. We therefore derive only the relative probabilities for introns of phase zero, one and two from their respective frequencies within the training set. All other transition probabilities are set to estimated values which are then tuned by hand during the optimisation of the performance with the training set. All transitions emerging from the begin state have the same probability **as** well **as** those leading to the end state.

The values of the transition probabilities are generally fixed. This means that the probability of each transition is independent of the positions within the two sequences at which the transition in used in the pair HMM. However, there are sequence signals whose strength varies along the sequence and which cannot be adequately described by the emission probabilities of the pair HMM's states. One example for this are splice sites. In the pair HMM, they are modelled by states which recognise the consensus (GT in the case of a 5' splice site and **AG** for a 3' splice site). The emission probabilities of these states cannot take into account the splice site signal **as** it is wider than the window of letters that the splice site states read. In order to incorporate the splice site signal into the pair HMM, every potential splice site in the two DNA sequences is scored by a splice site predictor program [LD01] similar to that in [BK97]. These scores are transformed into posterior probabilities which modify the nominal transition probabilities leading into the splice site states. A potential splice site with a high score leaves the nominal transition probability almost unchanged, whereas a low score decreases it. The probabilities of all the other transitions emerging from the same state are rescaled accordingly by a common factor so that the sum of all transition probabilities emerging from that state

always remains one. This is an extension of the pair HMMs described in Section 1.5.1. We call transitions ***special*** if their value is affected by position dependent sequence signals. The implementation of special transitions is explained in detail in Chapter 6.
Similarly to splice sites, all potential start codons are scored using a weight matrix model of 21 base pairs width that starts 9 base pairs 5' to the potential start codon.

Emission probabilities of the *match exon* state The emission probabilities of the ***match exon*** state are derived from a training set, see Section A.1 in Appendix **A** and Section C.1 in Appendix C. The main idea is to base the emission probabilities of all states except for those of the ***START START*** and the *STOP STOP* state on the emission probabilities of the ***match exon*** state,

$$p(x_1, x_2, x_3, y_1, y_2, y_3),$$

where (x_1, x_2, x_3) denotes the letters read from sequence $\boldsymbol{X}$ an (y_1, y_2, y_3) denotes those read from sequence $\boldsymbol{Y}$. These probabilities are derived from pairs of orthologous genes that have identical coding length, the coding length of a gene being defined as the sum of lengths of its exons. For every such pair of genes, the exons of each gene are concatenated into a continuous sequence of codons finishing in a stop codon. From each such pair of codon sequences, the aligned terminal stop codons are used to derive the emission probabilities of the *STOP* STOP state and the rest of the aligned codon pairs is used to derive the emission probabilities of the ***match exon*** state. One of our training sets, see Section A.1 in Appendix **A,** is not large enough to avoid zero counts for some legal codon pairs. We could thus not simply use the maximum likelihood method to estimate the emission probabilities of the ***match exon*** state. In order to be able to apply the same estimation method to training sets of variable size, we refrained from adding simple pseudo-counts and instead chose a Dirichlet distribution with the following posterior mean estimator ($i := (x_1, x_2, x_3)$, $j := (y_1, y_2, y_3)$):

$$p(i,j) = \frac{n(i,j) + A \cdot q(i,j)}{\sum_{i,j} n(i,j) + A}$$

where $n(i,j)$ is the number of aligned, unordered codon pairs with codon i and codon j in the training set, $\boldsymbol{A}$ is the number of unordered non-stop codon pairs, i.e. $61 \cdot 60/2 + 61 = 1891$, and $q(i,j) = (c(i) + c(j)) / \sum_{i,j} (c(i) + c(j))$, where $c(i)$ is the number of codons i in the training set of aligned exons. This formula introduces a symmetry with respect to the two sequences X

and Y as $p(i,j) = p(j,i)$. The advantage of this method is that it scales well from rather small training sets for which the probability of rare events is $p(i,j) \approx q(i,j)$ to large sets for which this probability converges to the maximum likelihood result, i.e. $p(i,j) \approx n(i,j)/\sum_{i,j} n(i,j)$. We have investigated whether the aligned concatenated exons of the training set give sensible emission probabilities for the *match exon* state by comparing the frequencies of pairs of amino-acids, see Figure B.1 in Appendix B and Figure D.1 in Appendix D. As one would expect, each codon is found to be preferentially aligned to codons which encode the same amino-acid. The DNA sequences of the training set are therefore evolutionarily well conserved.

Another question which we address is whether the codons in the pairs which encode the same amino-acid are likely to be the same or whether there exists a bias. The results of this study are shown in Figure B.2 in Appendix B and in Figure D.2 in Appendix D. Each diagram corresponds to one amino-acid and shows all possible *unordered* codon pairs and the frequency with which they are observed. It is apparent from the mouse human training set that the results for most amino-acids do not follow any simple rule: the frequency of pairs where the two codons are the same need not be higher than that of pairs where the two codons differ, as shown for example for isoleucine (I). However, the results for serine (S), which is encoded by six different codons which differ in any of the three codon positions, follow some basic rules: codon pairs where the two codons are the same, dominate, {TCC, TCC} = 0.1980, {AGC, AGC} = 0.1962, {TCT, TCT} = 0.1058, {AGT, AGT} = 0.0939, {TCA, TCA} = 0.0870, and {TCG, TCG} = 0.0239. Differences of the codons in the wobble position are tolerable (with observed frequencies ranging from {TCA, TCT} = 0.0205 to {TCC, TCT} = 0.0819) and pairings between codons which differ in their first codon position almost never occur (the frequencies of these events being lower than 0.0051).

Concerning the pairs of stop codons, codon pair {TGA, TGA} dominates, followed by the pairs {TAA, TAA} and {TAG, TAG} of approximately the same frequency.

It would be interesting to investigate if the codon pair frequencies shown in Figure B.2 in Appendix B and in Figure D.2 in Appendix D correspond to a deviation from the codon pair frequencies that would be obtained by assuming that the frequency of *each* codon pair is equal to the product of the two individual codon frequencies.

Emission probabilities of the other states The emission probabilities of the other states except for those of the *START START* and *STOP STOP* state are calculated using the above

match exon emission probabilities $p(x_1, x_2, x_3, y_1, y_2, y_3)$. This is done by marginalising over one or more codon positions. The emission probabilities for the two emit exon states are given by:

$$p_{\text{emit x exon}}(x_1, x_2, x_3) = \sum_{(y_1, y_2, y_3)} p(x_1, x_2, x_3, y_1, y_2, y_3)$$

and the analogous expression for the emit y exon state.

The match intergenic and match intron states have the same emission probabilities. They are given by:

$$p_{\text{match intron}}(i, j) = \frac{1}{3}\Bigg(\sum_{(x_2, x_3, y_2, y_3)} p(i, x_2, x_3, j, y_2, y_3) + \sum_{(x_1, x_3, y_1, y_3)} p(x_1, i, x_3, y_1, j, y_3) + \sum_{(x_1, x_2, y_1, y_2)} p(x_1, x_2, i, y_1, y_2, j) \Bigg)$$

The *emit* intergenic states and the corresponding emit intron states also have the same emission probabilities. They are obtained by marginalising over one of the two indices of the match intron emission probabilities.

The emission probabilities for the three match 5' splice site states are obtained by:

$$\begin{aligned} p_{\text{phase 0}}(x_1, x_2, y_1, y_2) &= \delta_{x_1 G}\delta_{x_2 T}\delta_{y_1 G}\delta_{y_2 T} \\ p_{\text{phase 1}}(x_1, x_2, x_3, y_1, y_2, y_3) &= \delta_{x_2 G}\delta_{x_3 T}\delta_{y_2 G}\delta_{y_3 T} \sum_{(x'_2, x'_3, y'_2, y'_3)} p(x_1, x'_2, x'_3, y_1, y'_2, y'_3) \\ p_{\text{phase 2}}(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4) &= \delta_{x_3 G}\delta_{x_4 T}\delta_{y_3 G}\delta_{y_4 T} \sum_{(x'_3, y'_3)} p(x_1, x_2, x'_3, y_1, y_2, y'_3) \end{aligned}$$

In a similar way, the emission probabilities for the three match 3' splice site states axe determined using:

$$
\begin{aligned}
p_{\text{phase 0}}(x_1, x_2, y_1, y_2) &= \delta_{x_1 A}\delta_{x_2 G}\delta_{y_1 A}\delta_{y_2 G} \\
p_{\text{phase 1}}(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4) &= \delta_{x_1 A}\delta_{x_2 G}\delta_{y_1 A}\delta_{y_2 G} \sum_{(x_1', y_1')} p(x_1', x_2, x_3, y_1', y_2, y_3) \\
p_{\text{phase 2}}(x_1, x_2, x_3, y_1, y_2, y_3) &= \delta_{x_1 A}\delta_{x_2 G}\delta_{y_1 A}\delta_{y_2 G} \sum_{(x_1', x_2', y_1', y_2')} p(x_1', x_2', x_3, y_1', y_2', y_3)
\end{aligned}
$$

The emission probabilities of the match **5'** splice site and match *3'* splice site states can then be used to derive the emission probabilities of the emit *5'* splice site and emit **3'** splice site states by summing over the relevant indices in the same way the emit *exon* state emission probabilities are obtained from the match exon state emission probabilities.
The emission probabilities of the *START START* state are simply given by:

$$p_{\text{START START}}(x_1, x_2, x_3, y_1, y_2, y_3) = \delta_{x_1 A}\delta_{x_2 T}\delta_{x_3 G}\delta_{y_1 A}\delta_{y_2 T}\delta_{y_3 G}$$

The emission probabilities of the *STOP STOP* state are determined from the training set in the same way the emission probabilities are determined for the match *exon* state by using a Dirichlet distribution. The observed frequencies for all possible pairs of stop codons in the training set are shown in Figure B.2 in Appendix B and Figure D.2 in Appendix D.
During the training of **DOUBLESCAN,** we implemented a fifth order Markov model in order to use hexamer frequencies which are frequently used **to** distinguish protein-coding from non-protein coding DNA [BK97, GF95], but abandoned the use of hexamer frequencies **as** they did not improve the performance.

2.4 The Stepping Stone algorithm

For a given pair HMM with N states and T transitions and two input sequences X and Y of length L_x and L_y, the optimal state path can be found with a memory requirement which scales only linearly with the length of the input sequence, $O(N \cdot min\{L_x, L_y\})$, and a time requirement which scales quadratically with the length of the input sequence, $O(T \cdot L_x \cdot L_y)$, using the Hirschberg algorithm, see Section 1.5.2. The memory requirement of the Viterbi

algorithm can thus be linearised, but the time requirement is still quadratic and imposes a serious constraint on the analysis of long DNA sequences.

Our aim in introducing the Stepping Stone algorithm is to invent a method by which a nearly optimal state path can be found with time and memory requirements which scale linearly with the sequence length. The main idea is to first employ a simple local alignment program to search for subsequences of strong similarity between the two input DNA sequences and then to use these matches as guidelines to search for the optimal state path only in a sub-space of the Viterbi matrix.

In the first step, the local alignment program BLASTN [AGM+90] is used to search the two input DNA sequences for regions of high similarity. The set of matches returned by BLASTN is then turned into a set of mutually compatible constraints in the following way. We select the highest scoring match and define its middle point as its reference point. We then take this middle point to find the next highest scoring match whose middle point is compatible with it and repeat this scheme until no more compatible middle points can be added. A new middle point is compatible with an already selected set of middle points if their pairs of (x, y) coordinates can be simultaneously ordered by their x and y coordinates. Although we then have a set of (x, y) constraints at which the two DNA sequences match, we do not know whether these matches correspond to exons, introns or intergenic regions as BLASTN does not assign any functional annotation to the matches. We allow for this uncertainty by allowing all states at the (x, y) midpoint. The overlap between two adjacent sub-matrices is thus a line whose projection onto the (X,Y) plane is a point at (x, y). In particular, BLASTN does not know about codons and phases. It may thus happen that a match corresponds to aligned exons whose codons are out of phase. To allow DOUBLESCAN to correct for this phase difference, we increase the overlap at (x, y) to a small 15 base pairs by 15 base pairs region around (x, y). Two adjacent sub-matrices thus overlap in a small volume of 15 base pairs by 15 base pairs by N. The set of concatenated sub-matrices defines a continuous sub-space of the Viterbi matrix, see the hatched area in Figure 2.5, which is searched for the highest scoring state path in the following step.

In the next step, the optimal state path in the thus restricted sub-space of the Viterbi matrix is retrieved by first calculating the elements in the sub-space and by then applying a traceback procedure. The calculation is started at the lower left sub-matrix, see Figure 2.5, using the Viterbi algorithm. During the calculation we keep only the values in a narrow strip like

volume in memory with which the calculation can be continued. Once this sub-matrix has been calculated, only the values in the small volume where this sub-matrix overlaps the next one are used to initialise the calculation of the next sub-matrix. This process is iterated until the calculation of the upper right sub-matrix is finished and the ends of the sequences are reached. We then know the score of the highest scoring path that lies within the sub-space of the Viterbi matrix. The corresponding state path is retrieved by proceeding from the upper right to the lower left sub-matrix, recalculating each sub-matrix with now partially known boundaries either using the Viterbi algorithm, if there is sufficient memory, or the Hirschberg algorithm.

The benefits of the Stepping Stone algorithm are that the time and memory requirements are reduced with respect to the Viterbi algorithm. If we assume that there is a minimum number of BLASTN matches per sequence length, both memory and time requirements depend essentially linearly on the sequence length, i.e. are of order $U(N \cdot \sqrt{L_x^2 + L_y^2})$ and $\mathcal{O}(T \cdot \sqrt{L_x^2 + L_y^2})$, respectively, as the number of rectangles is expected to increase asymptotically as $\sqrt{L_x^2 + L_y^2}$. The disadvantage of the Stepping Stone algorithm is that the state path which is optimal within the sub-space of the Viterbi matrix is not necessarily identical to the optimal state which would have been found by calculating the whole Viterbi matrix. In Section 3.4 we show for a test set of mouse and human DNA sequences that the Stepping Stone algorithm finds the true optimal state path in 81 % of all cases and that 97 % of the predicted genes are the same as those predicted by the Hirschberg algorithm. The Stepping Stone algorithm therefore provides a very good practical solution.

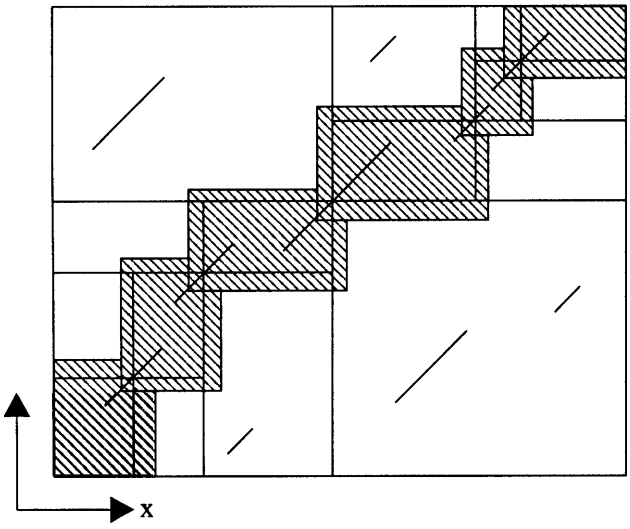

Figure **2.5:** The Stepping Stone algorithm: Shown is the projection of the Viterbi matrix onto the (X,*Y*)plane spanned by the two sequences X and *Y*. Diagonals represent similar subsequences retrieved by **BLASTN**, hatched areas correspond to sub-matrices which are calculated using the Viterbi algorithm or the Hirschberg algorithm. Note that there is no restriction imposed on the third dimension, the state dimension.