Chapter 3

Annotated Alignments

In the previous chapter we described formally the basic concepts behind the formulation, scoring and application of profiles and alignments. This chapter can be viewed as an amalgamation and extension of the two issues addressed there; the main contribution is the theory for inclusion of profiles into alignments yielding the so-called *pair-profile* hits.

The initial part of this chapter is in line with the standard alignments of Section 2.2. We present modifications to extend the standard alignment dynamic programming algorithm to include profiles. Building upon the previously introduced fundamentals of profiles, in the second part of this chapter we describe a statistically motivated choice for the additional pair-profile parameters. The basic framework underlies the tool SimAnn [12]. As an extension to the basic parameter choice, in Section 3.2.2 we describe the possibility of incorporating evolution of binding sites into annotated alignments. Pairwise alignments can be interpreted as pair Hidden Markov Models (HMMs), and it is but natural to formalize the annotated alignment approach in the pairHMM framework and highlight the subtle differences involved. This constitutes the second last contribution of this chapter. Finally, we provide simple simulation results, on random sequences, discussing the characteristics and complexity of annotated alignments (Sections 3.4 and 3.5).

3.1 Dynamic programming algorithm

In the following, we assume that the profile P as well as the position-specific scoring matrix `PSSM` are provided. They can be obtained simply by following the procedure outlined in Section 2.1. Again, the background profile is taken to be $\Pi = \Pi_i = \pi$, $i \in 1 \dots l$.

Linear Gap Costs For the sequences x and y, the dynamic programming matrix `M` is initialized as in the standard case, with $\mathtt{M}(i,0) = \mathtt{M}(0,j) = 0$. We motivated the recursion rule (Section 2.2.1) for the dynamic programming algorithm by considering the possible alternatives – substitution, insertion or deletion – in which an optimal alignment of a pair of prefixes (x_{1i}, y_{1j}) could end.

In the annotated alignment approach, this is extended to include the possibility that (x_i, y_j) is the *last* nucleotide pair of a pair-profile hit. In other words, the optimal alignment ends with a pair-profile state.

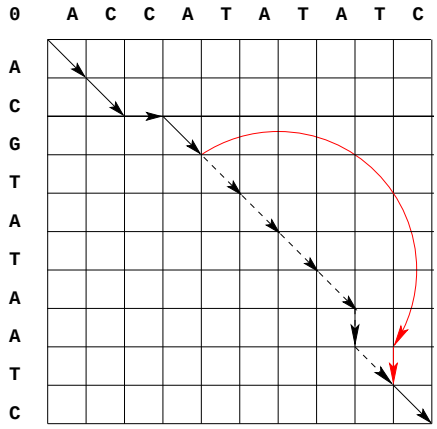


Figure 3.1: *Dynamic Programming Matrix* The annotated alignment path in the dynamic programming matrix is shown. As before, the example is for the sequences in Figure 1.6. Instead of just the three predecessor points, those corresponding to the length of the profile previously are also considered. The pair-profile jump is highlighted in red.

The possible alternatives, therefore are as shown below:

$\ldots x_i$	$\ldots x_{i-1} \quad x_i$	$\ldots x_i \quad -$	$\ldots [x_{i-l+1} \ldots x_{i-1} x_i]$
$\ldots y_j$	$\ldots y_j \quad -$	$\ldots y_{j-1} \quad y_j$	$\ldots [y_{j-l+1} \ldots y_{j-1} y_j]$

Observe that in the pair-profile case, it is essential that (x_{i-l+1}, y_{j-l+1}) are aligned, otherwise the pair-profile ending at (x_i, y_j) would contain gaps which is contrary to our objective of retrieving perfectly aligned TFBS pairs. Let us denote $i' = i - l + 1$ and $j' = j - l + 1$ and the substrings $x_{i'i} = (x_{i'} \ldots x_i)$ and $y_{j'j} = (y_{j'} \ldots y_j)$.

Keeping the above observation in mind, the optimal alignment at (i', j') should end only with a substitution for a possibility to have a pair-profile hit starting there. The new recursion rule includes this as following:

$$\mathtt{M}(i,j) = \max \begin{cases} 0, \\ \mathtt{M}(i-1, j-1) + \mathtt{s}(x_i, y_j), \\ \mathtt{M}(i-1, j) - \mathrm{g}, \\ \mathtt{M}(i, j-1) - \mathrm{g}, \\ \mathtt{M}(i', j') + \mathtt{PSA}(x_{i'i}, y_{j'j}) - pen \end{cases} \tag{3.1}$$

Hence, if the optimal alignment of the substrings x_{1i}, y_{1j}, where $i, j \geq l$, ends with a pair-profile hit then this implies that a pair-profile of length l ends at (x_i, y_j) and scores highest amongst all other possibilities. The score for such a pair-profile, as shown, is calculated by summing the PSA score of the last l nucleotide pairs from x_{1i}, y_{1j} and subtracting penalty pen from the optimal alignment score at (i', j') given by $\mathtt{M}(i', j')$. Diagrammatically, Fig. 3.1 illustrates how the score is now calculated using the scores of the three predecessor points as well as the $(i', j')^{th}$ point.

Finally, the optimal alignment is the one with the maximum score and is generated by traceback from the point (k, l) with the maximum score at $\max_{k,l} \mathtt{M}(k, l)$, where for every pair-profile jump corresponding l nucleotide pairs are emitted. As can be seen from the recursion rule above, the time and space complexity are quadratic in the sequence length. For each additional profile, an extra comparison needs to be performed where the corresponding PSA score is calculated using the previous l nucleotide pairs. This yields a time complexity which is linear in the number and length of profiles. In Section 3.5, we investigate the dependence of on sequence length as well number and length of profiles through simulations on random sequences.

Affine Gaps The extension to affine gap costs is equally straightforward with the new recursion rules presented next. Keeping the notations for the gap costs g_o and g_e, the entries of the matrix corresponding to an end with a substitution (that is matrix $\mathtt{M}$), are calculated with the modified recursion rule:

$$\mathtt{M}(i,j) = \max \begin{cases} 0, \\ \mathtt{M}(i-1,j-1) + \mathtt{s}(x_i, y_j), \\ \mathrm{D}(i-1,j-1) + \mathtt{s}(x_i, y_j), \\ \mathrm{I}(i-1,j-1) + \mathtt{s}(x_i, y_j), \\ \mathtt{M}(i',j') \;+\; \mathtt{PSA}(x_{i'i}, y_{j'j}) \;-\; pen \\ \mathrm{D}(i',j') \;+\; \mathtt{PSA}(x_{i'i}, y_{j'j}) \;-\; pen \\ \mathrm{I}(i',j') \;+\; \mathtt{PSA}(x_{i'i}, y_{j'j}) \;-\; pen \end{cases}$$

The recursion rules for D and I remain the same. This is simply because ending with a pair-profile hit implies the last character pair of the alignment is an aligned pair or a substitution. Therefore the score can be calculated by using the matrix for the substitution. It is possible to arrive at P state from a substitution, insertion, deletion or even another pair-profile state. Coming from the pair-profile instance is equivalent again to coming from a substitution. The optimal alignment is again generated by traceback.

3.2 Choice of score parameters

The proposed annotated alignment algorithm combines standard alignments with profiles. Since inter-linking these concepts arbitrarily may lead to an unrealistic bias in the results, therefore an appropriate parameter choice is crucial. In this section, we provide strategies for the same.

We assume that the score parameters related to standard alignments are derived as discussed in Section 2.2.2. This means that the substitution scoring matrix $\mathtt{s}$ is derived as a log-likelihood ratio of a distribution ϕ for evolutionarily related nucleotide pairs with respect to an independent sampling of two letters from a background distribution π.

On the same lines, the pair-profile related score for a profile P of length l is derived using the log-likelihood ratio of a distribution on pairs $(\mathbf{u}, \mathbf{v})$ of strings of length l with respect to the *same* background distribution π. Such a distribution needs to reflect the properties of the binding site profile we wish to search for. The two strategies for parameter derivation differ in the choice of this distribution, as we see next.

A basic derivation for profile-related parameters, underlying the tool **SimAnn** [12], is presented first. This is followed by an advanced formulation where TFBS-specific evolutionary constraints are explicitly incorporated. This forms the basis of the improved version **eSimAnn**.

3.2.1 Basic formulation – SimAnn

Calculation of Profile Scoring Array (PSA) In the basic formulation, we start with the position-specific letter distribution $\mathrm{P} = (\mathrm{P}^1, \ldots, \mathrm{P}^l)$ of a signal profile and consider two strings $\mathbf{u}$ and $\mathbf{v}$ to be sampled *independently* from P. In the corresponding background distribution all letters occurring in the strings are sampled independently from π. More formally, this leads to:

$$\mathtt{PSA}(\mathbf{u}, \mathbf{v}) := \log \left(\frac{\mathrm{P}(\mathbf{u})\mathrm{P}(\mathbf{v})}{\prod_{i=1}^{l} \pi(u_i)\pi(v_i)} \right) \tag{3.2}$$

which on rewriting yields:

$$\begin{aligned} \mathtt{PSA}(\mathbf{u}, \mathbf{v}) &:= \sum_{i=1}^{l} \log \left(\frac{\mathrm{P}^i(u_i)\mathrm{P}^i(v_i)}{\pi(u_i)\pi(v_i)} \right) \\ &= \sum_{i=1}^{l} \log \left(\frac{\mathrm{P}^i(u_i)}{\pi(u_i)} \right) + \sum_{i=1}^{l} \log \left(\frac{\mathrm{P}^i(v_i)}{\pi(v_i)} \right) \\ &=: \mathtt{PSSM}(\mathbf{u}) + \mathtt{PSSM}(\mathbf{v})) \end{aligned} \tag{3.3}$$

where PSSM denotes the position-specific scoring matrix (see Section 2.1.2). Note the simple additive form where the PSA score can be derived solely using the PSSM of the corresponding profile. This additivity arises from the fact that we sample two strings independently from P.

Calculation of Profile Penalty (*pen*) Recall that the profile penalty *pen* has been introduced for fine-tuning the balance between the two gapless scoring alternatives of the two strings $\mathbf{u}$ and $\mathbf{v}$: l substitutions or one pair-profile hit. Comparing the pair-profile hit and l substitution score distributions yields a log-likelihood ratio (LLR) test, wherein the profile penalty can be chosen based on desired type I and type II error levels. That is,

$$\mathtt{PSA}(\mathbf{u}, \mathbf{v}) - pen > \sum_{i=1}^{l} \mathtt{s}(u_i, v_i)$$

Using the definition of s presented in Section 2.2.2, this can be written as:

$$\log \left(\prod_{i=1}^{l} \frac{\mathrm{P}^i(u_i)\mathrm{P}^i(v_i)}{\phi(u_i, v_i)} \right) > pen$$

Since the calculation of all scores involved is based on the same background model π, it cancels out here. This is equivalent to:

$$\mathrm{LLR}_{\mathrm{P}^2, \phi^l}(\mathbf{u}, \mathbf{v}) := \log \frac{\mathrm{P}(\mathbf{u})\mathrm{P}(\mathbf{v})}{\prod_{i=1}^{l} \phi(u_i, v_i)} > pen \tag{3.4}$$

Hence the log-likelihood ratio directly compares the pair profile measure P^2 and the measure ϕ^l which arises from independently sampling l evolutionarily related letter pairs from ϕ.

In statistical testing language, this can be viewed as following. The null hypothesis H_0 is that the l nucleotide pairs are aligned as consecutive substitutions from the standard

alignment model. The alternative hypothesis H_1 is that they are aligned as one pair-profile hit of the profile P. If the test statistic is taken to be the log-likelihood ratio $\mathrm{LLR}_{\mathrm{P}^2,\phi^l}$ defined above, then *pen* is the *threshold* which has to be surpassed for the null hypothesis to be rejected. That is, it can be interpreted as a cutoff in a log-likelihood ratio test. Thus if the exact distribution of $\mathrm{LLR}_{\mathrm{P}^2,\phi^l}(\mathbf{u},\mathbf{v})$ under the two measures P^2 and ϕ^l are known, then *pen* can be calculated based on desired type I and type II error levels, as described in Section 2.1.3.

Calculation of Score distributions Following the lines of the calculation of exact distributions under the signal and background profile in the single sequence case (Section 2.1.3), we calculate the exact distributions under P^2 and ϕ^l of the LLR score by dynamic programming. It is interesting to see how the score distribution in the pair-profile scenario (pair of strings) compares with respect to that in the single string case.

Similar to the single sequence case, we use three natural choices. First we choose *pen* such that for a pre-specified level α the type-I error probability $\mathbb{P}_{\phi^l}(\mathrm{LLR}_{\mathrm{P}^2,\phi^l}(\mathbf{u},\mathbf{v}) > pen)$ is smaller than α. We call this the *level α type-I error penalty*. Second, we choose *pen* such that the corresponding type-II error probability $\mathbb{P}_{\mathrm{P}^2}(\mathrm{LLR}_{\mathrm{P}^2,\phi^l}(\mathbf{u},\mathbf{v}) < pen)$ is smaller than a pre-specified level β. We call this the *level β type-II error penalty*. Finally, we choose *pen* such that the two error probabilities are equal, in which case we speak of the *balanced penalty*.

Comments SimAnn uses the above-described basic parameter formulation where profile parameters are derived assuming independence between a pair of aligned putative binding sites. In Section 1.6, we discussed the ambiguities and problems associated with defining and predicting a conserved TFBS hit. We mentioned also that the independent scoring of individual hits disregards the binding site-specific evolutionary characteristics. It relies solely on the single sequence hit scores and hence, may unduly penalize or favor a pair of strings irrespective of their mutual relatedness. For example, for a poor profile, the binding site may be quite degenerate. For a pair of closely related sequences, if a random aligned stretch of contiguous nucleotide pairs consist mostly of distinct (mismatches) but relatively high-scoring nucleotides, then the independent scoring may still score the stretch as a conserved binding site. On the other hand, if the binding site evolution is taken into account, it is quite unlikely that a high proportion of mutations (ie. distinct nucleotides) is allowed to be considered as a conserved hit.

In the following, we provide modifications to the basic strategy to tackle this problem. By incorporating binding site evolution into a simultaneous alignment approach, the aim is to associate each aspect of the annotated alignment algorithm with an underlying evolutionary process.

3.2.2 Incorporating evolution of binding sites – eSimAnn

Extending the SimAnn framework to treat a pair of strings as evolutionarily related binding sites involves modifying the parameter derivation; the underlying alignment algorithm remains same. Still, for writing simplicity, we refer to the annotated alignment approach with explicit incorporation of binding site evolution as eSimAnn.

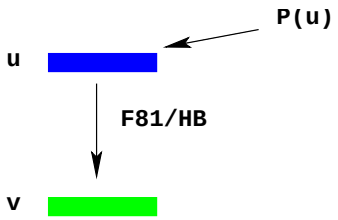


Figure 3.2: *Incorporating binding site evolution.* A motif is sampled from the position-specific letter distribution of the profile and evolved according to a position-specific evolutionary model (F81 or HB).

In this setting, the alternative hypothesis (H_1) now is that a pair of strings represents evolutionarily related binding sites. The first string in a pair is sampled from the profile and then evolved to the second according to a profile-based evolutionary model (Figure 3.2). Predicted hit-pairs are "conserved" binding sites – gaplessly aligned, evolutionarily related profile instances. Beginning with a general setting, we demonstrate how any suitable evolutionary model can be explicitly incorporated into the SimAnn framework. Next, we demonstrate through two evolutionary models employed previously for modeling TFBS evolution how such an incorporation can be carried out in practice.

General Procedure

Equipped with an evolutionary model, the $\texttt{PSA}(\mathbf{u}, \mathbf{v})$ can be calculated as the log-likelihood ratio of observing the pair $\mathbf{u}, \mathbf{v}$ as evolutionarily related binding sites represented by the profile P versus each u_i, v_i sampled independently from the background distribution π. The PSA score at a position i can then be calculated by considering the corresponding position-specific letter distribution of the profile and the evolutionary distribution at that position. Let $\tilde{\rho} = \tilde{\rho}^1(u_1, v_1, t), \ldots \tilde{\rho}^l(u_l, v_l, t)$ give the position-specific time-dependent transition probabilities under the profile, then the PSA score at a position i in the profile is given by:

$$\texttt{PSA}(u_i, v_i) := \log \left(\frac{\mathrm{P}^i(u_i)\tilde{\rho}^i(u_i, v_i, t)}{\pi(u_i)\pi(v_i)} \right) \tag{3.5}$$

Hence the score of u_i, v_i at position i is calculated by comparing the probability of observing u_i in the first string and then evolving it according to $\tilde{\rho}^i$ to v_i.

Comparing the above with Equation (3.3), we can see that the incorporation of binding site relatedness into the score derivation leads to a loss of the simple additive form. At the same time, the above derivation requires additional rate parameters. For a first approach to estimate these parameters, given a substitution scoring matrix, we adopt the simple strategy of assuming an evolutionary model like the Jukes-Cantor [90] for background sequence evolution (details later).

The profile penalty *pen* now compares between the two alternatives of $\mathbf{u}, \mathbf{v}$ being evolutionarily related samples of P versus each (u_i, v_i) being a standard substitution. If we denote the former by $\tilde{\phi}$, then at position i, $\tilde{\phi}^i(u_i, v_i) = \mathrm{P}^i(u_i)\tilde{\rho}^i(u_i, v_i, t)$. This simply leads to a log-likelihood score formulated using the two distributions $\tilde{\phi}$ and ϕ:

$$\mathrm{LLR}_{\tilde{\phi}^l, \phi^l}(\mathbf{u}, \mathbf{v}) := \log \prod_{i=1}^{l} \frac{\tilde{\phi}(u_i, v_i)}{\phi(u_i, v_i)} > pen \tag{3.6}$$

Hence, whereas previously (Equation (3.4)) we compared the distribution under independent sampling of profile versus that under background evolution, here we compare the distribution under TFBS evolution versus that under background evolution. The exact distributions under $\tilde{\phi}$ and ϕ can again be derived as described in Section 2.1.3 and the profile penalty chosen according to desired type I and type II error levels.

Estimating the rate parameters Given an appropriate substitution scoring matrix $\mathtt{s}$, we retrace the rate parameters by assuming that the background sequences evolved according to a simple evolutionary model. We use the Jukes-Cantor (JC) model [90] (Section 2.2.3) model for simplicity, although more sophisticated models can be similarly employed. Given a substitution scoring matrix $\mathtt{s}$, we can write the probability that a pair of nucleotides is related in terms of the log-likelihood scores:

$$\phi(u, v) = (e^{\mathtt{s}(u,v)}) * \pi(u)\pi(v)$$

Using the transition probabilities as derived from the JC model, we also get:

$$\phi(u, v) = \pi(u)[e^{-\mu t}\delta(u, v) + (1 - e^{-\mu t})\pi(v)] \qquad \forall \quad u, v$$

Hence, the unknown parameter pair μt, where μ is the mean instantaneous substitution rate and t is the time elapsed, can be estimated from the above two equations.

After providing the generic approach of incorporating any evolutionary model for TFBS evolution, we now describe how two evolutionary models, used commonly for modeling the position-specific evolutionary properties of TFBSs, can be considered. It is worthwhile to mention here that while both models provide a better approach to modeling position-specific evolution in binding sites as compared to models that treat all positions similarly, each relies on simplifying assumptions. Nevertheless, they provide a more realistic representation of binding site evolution, as shown by Moses and colleagues [132]. We begin with the F81 model because of its simplicity and ease of incorporation into the SimAnn framework.

Using the Felsenstein 1981 model

To ensure that each position in a profile is treated differently with regards to evolutionary characteristics, for an initial choice we adapted the Felsenstein 1981 model (F81) (Section 2.2.3). Here, the probability of a substitution is proportional to the stationary distribution of the incoming nucleotide. Setting the stationary distribution at each position i to the position-specific letter distribution under the profile P^i, the model respects the initial base composition through position-specific substitution rates. The position-specific transition probabilities at a position i are then given by:

$$\tilde{\rho}^i(u_i, v_i, t) = e^{-\mu t}\delta(u_i, v_i) + (1 - e^{-\mu t})\mathrm{P}^i(v_i) \qquad \forall \quad u_i, v_i \tag{3.7}$$

where μ is the rate of mutations per site and δ is the Kronecker delta function with $\delta(u, v) = 1$ if $(u = v)$ and 0 otherwise.

Inserting the above probabilities into Equation (3.5) and rearranging, the PSA score for the pair of strings is given as:

$$\mathtt{PSA}(\mathbf{u}, \mathbf{v}) := \mathtt{PSSM}(\mathbf{u}) + \mathtt{PSSM}(\mathbf{v}) + \sum_{i=1}^{l} \log\left[(\frac{\delta(u_i, v_i)}{\mathrm{P}^i(v_i)} - 1)e^{-\mu t} + 1\right]$$

which is symmetric, since:

$$\mathrm{P}^i(u_i)\tilde{\rho}^i(u_i, v_i, t) = \mathrm{P}^i(v_i)\tilde{\rho}^i(v_i, u_i, t)$$

Note how as time goes to infinity, the previous equation reduces to the simple additive form in the case of independent scoring in Equation (3.3). However, when $u_i \neq v_i$, the contribution of the profile letter distribution in the additional term is lost and the score depends purely on the evolutionary rate. While considering a rate slower than background sequence has been proposed [132] it does not fully reflect the profile conservation properties at each position. Currently, we use the rate as derived from the substitution scoring matrix for background sequence (Section3.2.2).

Using the Halpern Bruno model

As discussed in Section 2.2.3, according to this model, a position-invariant mutation rate is combined with a position-dependent fixation rate to yield position-specific mutation rates $\mathtt{q}^i(u, v)$. Using a similar line of approach as the original article, Moses *et al.* [133] showed that the mutation rate at a position i in the profile is given as the following proportionality:

$$\mathtt{q}^i(u_i, v_i) \propto \mathtt{q}_\mathtt{B}(u_i, v_i) \times \frac{\ln x}{1 - 1/x} \tag{3.8}$$

where

$$x = \frac{\mathrm{P}^i(v_i)\mathtt{q}_\mathtt{B}(v_i, u_i)}{\mathrm{P}^i(u_i)\mathtt{q}_\mathtt{B}(u_i, v_i)}$$

and $\mathtt{q}_\mathtt{B}(u, v)$ gives the background evolutionary model. If $x = 1$, then the rate is equal to the background mutation rate $\mathtt{q}_\mathtt{B}(u, v)$. Let us see how the above equations help in modeling binding site evolution better.

We mentioned before (Section 1.5) that functionally relevant positions in binding sites evolve slower. For the time being, let us assume that the background evolutionary model is the JC model, hence $\mathtt{q}_\mathtt{B} = \mu t/4$. Hence, for equally likely nucleotides ($\mathrm{P}^i(u_i) = \mathrm{P}^i(v_i)$), the rate of substitution purely depends on the background evolutionary distance.

If $x \neq 1$, then the rate equation looks like:

$$\mathtt{q}^i(u_i, v_i) \propto \mu t \cdot \frac{\ln(\mathrm{P}^i(v_i)/\mathrm{P}^i(u_i))}{1 - (\mathrm{P}^i(u_i)/\mathrm{P}^i(v_i))} \tag{3.9}$$

where, for increasing probability values at position i under the profile for u_i, the rate of substitution to v_i decreases. Thus, non-degenerate positions are conserved. On the other hand, if $\mathrm{P}^i(u_i)$ is much lower than $\mathrm{P}^i(v_i)$, then the process favors substitution to the more likely nucleotide at this position under the profile. Hence, the model suitably reflects the position-specific evolution in the profile whereby degenerate positions mutate more while non-degenerate positions are more conserved.

Given the rate matrix, the transition probabilities $\tilde{\rho}^i$ at each position can again be derived by exponentiating $\mathbf{Q}t$. For a more detailed discussion on the use of the HB model for TFBS evolution, see [134, 132]. Finally, plugging the resulting $\tilde{\rho}^i$'s in Equations (3.5 and 3.6), the `PSA` and *pen* under the HB model can be derived as before. Note again that the above-derived scores are symmetric.

Summary In this section, we provided strategies to determine the profile-related parameters based on statistical considerations. We also demonstrated how position-specific evolutionary constraints in the profile can be incorporated in the proposed framework. Besides the algorithm and parameter choice, another relevant aspect of pairwise alignments in general, is their formalization in a probabilistic framework as pair Hidden Markov models (pairHMMs). Since annotated alignments build upon standard alignments, it is possible to visualize them as an extended version of the standard pairHMM and in the following we illustrate how.

3.3 A pairHMM perspective

Standard pairwise alignments can be formalized in a probabilistic framework as pair Hidden Markov models (pairHMMs). Since annotated alignments build upon standard alignments, it is intuitive to visualize them as an extended version of the standard pairHMM. In the following, we formally present a pairHMM interpretation of annotated alignments. However, it should be stressed here that while the formalization of the model is in line with the standard pairHMM methodology, the estimation of parameters is not performed via training, as is the case in HMM-based approaches.

Brief background

A hidden markov model (HMM) consists of a set of hidden states connected by directed transitions. Each state (except the *silent* state), emits a character with a certain emission probability. Usually, we know the sequence of characters and are interested in finding the corresponding state sequence that could have generated it. The standard approach is to use the dynamic programming based Viterbi algorithm. Given a set of observed values and a model, the Viterbi algorithm calculates the most probable state path for this observation set. Mostly studied in the context of speech recognition (the tutorial by Rabiner [157] is a classic reference), hidden markov models are increasingly becoming popular in biological applications.

In a *pairHMM*, instead of single characters, the states emit pairs of characters. Durbin *et al.* [52] use the concepts of pairHMM to formalize pairwise alignments. A pairHMM for standard alignments consists of three states (Figure 3.3): the *match* state `M` which emits pairs of nucleotides, and the *insert* (`I`) and *delete* (`D`) states emitting a nucleotide and a gap. PairHMMs have been widely used for *ab initio* gene prediction approaches [100, 4, 130]. In the following, we build upon the established concepts to put annotated alignments in a pairHMM framework.

Standard pairHMMs

Let us denote the transition probability matrix of the standard pairHMM (henceforth referred to as pHMM) by T. The transition probabilities have to satisfy the constraint that the probabilities of all transitions leaving a state sum to one. Usually, direct transitions between `D` and `I` are not allowed and the states `I` and `D` are symmetric. Hence, $T_{\mathtt{I},\mathtt{D}} = T_{\mathtt{D},\mathtt{I}} = 0$, $T_{\mathtt{M},\mathtt{D}} = T_{\mathtt{M},\mathtt{I}}$ and $T_{\mathtt{I},\mathtt{I}} = T_{\mathtt{D},\mathtt{D}}$.

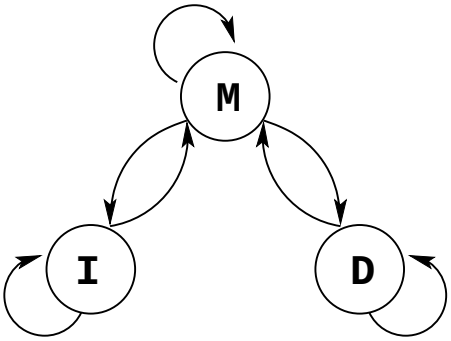


Figure 3.3: *Pair HMM for standard alignments.*

To make the forthcoming discussion on annotated pairHMMs easier, we add two silent states $\mathtt{S}_\mathtt{I}$ and $\mathtt{S}_\mathtt{D}$ to the pHMM, as shown in Figure 3.4. We refer to this as the extended pairHMM epHMM whose transition probabilities are given by the matrix $\tilde{T}$. To make the two topologies equivalent, we can construct $\tilde{T}$ using T, as follows.

For the match state $\mathtt{M}$, we have $\tilde{T}_{\mathtt{M},\mathtt{S}_\mathtt{I}} + \tilde{T}_{\mathtt{M},\mathtt{S}_\mathtt{D}} = 1$. Since, again the two states $\mathtt{D}$ and $\mathtt{I}$ are symmetric, this implies $\tilde{T}_{\mathtt{M},\mathtt{S}_\mathtt{I}} = \tilde{T}_{\mathtt{M},\mathtt{S}_\mathtt{D}} = 0.5$. For the self-transition to $\mathtt{M}$, hence:

$$\tilde{T}_{\mathtt{M},\mathtt{S}_\mathtt{I}}\tilde{T}_{\mathtt{S}_\mathtt{I},\mathtt{M}} + \tilde{T}_{\mathtt{M},\mathtt{S}_\mathtt{D}}\tilde{T}_{\mathtt{S}_\mathtt{D},\mathtt{M}} = T_{\mathtt{M},\mathtt{M}}$$

which yields $\tilde{T}_{\mathtt{S}_\mathtt{I},\mathtt{M}} = \tilde{T}_{\mathtt{S}_\mathtt{D},\mathtt{M}} = T_{\mathtt{M},\mathtt{M}}$. Similarly, for the silent states and the insert and delete states, we have the following constraints:

$$\tilde{T}_{\mathtt{S}_\mathtt{I},\mathtt{I}} + \tilde{T}_{\mathtt{S}_\mathtt{I},\mathtt{M}} = 1 = \tilde{T}_{\mathtt{S}_\mathtt{D},\mathtt{D}} + \tilde{T}_{\mathtt{S}_\mathtt{D},\mathtt{M}} \tag{3.10}$$

$$\tilde{T}_{\mathtt{I},\mathtt{I}} + \tilde{T}_{\mathtt{I},\mathtt{S}_\mathtt{I}} = 1 = \tilde{T}_{\mathtt{D},\mathtt{D}} + \tilde{T}_{\mathtt{D},\mathtt{S}_\mathtt{D}} \tag{3.11}$$

where the symmetry assumption of $\mathtt{D}$ and $\mathtt{I}$ implies $\tilde{T}_{\mathtt{I},\mathtt{I}} = \tilde{T}_{\mathtt{D},\mathtt{D}}$ and $\tilde{T}_{\mathtt{I},\mathtt{S}_\mathtt{I}} = \tilde{T}_{\mathtt{D},\mathtt{S}_\mathtt{D}}$. Again, considering the self-transitions at $\mathtt{I}$, we have:

$$\tilde{T}_{\mathtt{I},\mathtt{I}} + \tilde{T}_{\mathtt{I},\mathtt{S}_\mathtt{I}}\tilde{T}_{\mathtt{S}_\mathtt{I},\mathtt{I}} = T_{\mathtt{I},\mathtt{I}}$$

which on using (3.11) leads to:

$$\tilde{T}_{\mathtt{I},\mathtt{S}_\mathtt{I}} = \frac{1 - T_{\mathtt{I},\mathtt{I}}}{T_{\mathtt{M},\mathtt{M}}}$$

Plugging the above in Equation (3.11):

$$\tilde{T}_{\mathtt{I},\mathtt{I}} := \frac{T_{\mathtt{M},\mathtt{M}} + T_{\mathtt{I},\mathtt{I}} - 1}{T_{\mathtt{M},\mathtt{M}}} = \frac{T_{\mathtt{I},\mathtt{I}} - 2T_{\mathtt{M},\mathtt{I}}}{T_{\mathtt{M},\mathtt{M}}}$$

All other entries of $\tilde{T}$ are zero and the values for the states $\mathtt{S}_\mathtt{D}$ and $\mathtt{D}$ can be symmetrically calculated.

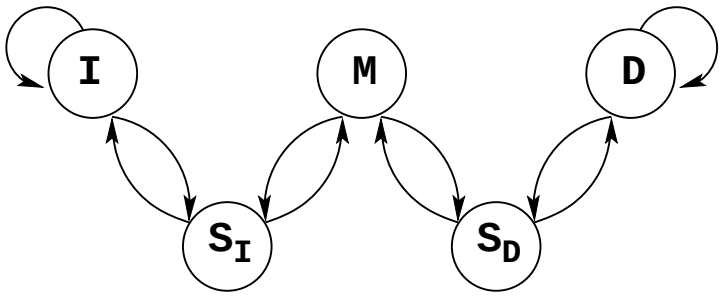


Figure 3.4: *Extended Pair HMM.* The modified version of the standard pairHMM introducing silent states $\mathtt{S}_\mathrm{I}$ and $\mathtt{S}_\mathrm{D}$.

Annotated pairHMMs

To the modified pairHMM described above, we now add states corresponding to profiles $\mathrm{P}_1, \ldots \mathrm{P}_m$ of lengths $l_1, \ldots, l_m$, respectively. We refer to the new pairHMM as annotated pairHMM – a pairHMM with pair-profile states included (denoted as ppHMM). The extended topology is shown in Figure 3.5. Each of the pair-profile states emits pairs of strings which are either independent (SimAnn) or evolutionarily related (eSimAnn).

Transitions Let us denote the transition matrix for ppHMM by $\bar{T}$. Again, assuming the states $\mathtt{S}_\mathrm{I}$ and $\mathtt{S}_\mathrm{D}$ to be symmetric, we set $\bar{T}_{\mathtt{S}_\mathrm{I},\mathtt{PP}_i} = \bar{T}_{\mathtt{S}_\mathrm{D},\mathtt{PP}_i} =: p_i$ and $\bar{T}_{\mathtt{PP}_i,\mathtt{S}_\mathrm{I}} = \bar{T}_{\mathtt{PP}_i,\mathtt{S}_\mathrm{D}} = 0.5$ for all $1 \leq i \leq m$. If we define

$$\bar{p} := \sum_{i=1}^{m} p_i < 1,$$

we can construct the rest of the $\bar{T}$ by multiplying the transition probabilities from $\mathtt{S}_\mathrm{I}$ or $\mathtt{S}_\mathrm{D}$ to the $\mathtt{M}, \mathtt{D}$ and $\mathtt{I}$ states in $\tilde{T}$ by $1 - \bar{p}$. That is,

$$\bar{T}_{\mathtt{S}_\mathrm{I},\mathtt{M}} = (1-\bar{p})\tilde{T}_{\mathtt{S}_\mathrm{I},\mathtt{M}} \quad \bar{T}_{\mathtt{S}_\mathrm{I},\mathtt{I}} = (1-\bar{p})\tilde{T}_{\mathtt{S}_\mathrm{I},\mathtt{I}}$$
$$\bar{T}_{\mathtt{S}_\mathrm{D},\mathtt{M}} = (1-\bar{p})\tilde{T}_{\mathtt{S}_\mathrm{D},\mathtt{M}} \quad \bar{T}_{\mathtt{S}_\mathrm{D},\mathtt{D}} = (1-\bar{p})\tilde{T}_{\mathtt{S}_\mathrm{D},\mathtt{D}}$$

The estimation of these unknown probabilities $p_1, \ldots, p_m$ is crucial since it influences the proportion of pair-profile jumps. We will show how these can be estimated using the concepts of log-likelihood ratio tests in a little while. First, we describe emission characteristics in the ppHMM.

Emissions The background alignment states $\mathtt{M}, \mathtt{D}$ and $\mathtt{I}$ behave as before, emitting single pairs of nucleotides (from $\mathtt{M}$) or a nucleotide and a gap (from $\mathtt{D}$ or $\mathtt{I}$). The pair-profile states emit *pairs of strings* which are either independent samples of the corresponding profile or evolutionarily related ones. We resort to previously introduced notations: π corresponds to the background letter distribution and $\mathtt{PSPM}_i$ to the position-specific letter distribution of the i^{th} profile. The distributions reflecting the evolutionary behavior of background sequences and profiles are again taken to be ϕ and $\tilde{\phi}$, respectively, as in Section 3.2.2.

The state $\mathtt{M}$ emits pairs of letters as following: a nucleotide is sampled from the background letter distribution π and evolved according to an appropriate evolutionary model for a given time t to produce the second nucleotide.

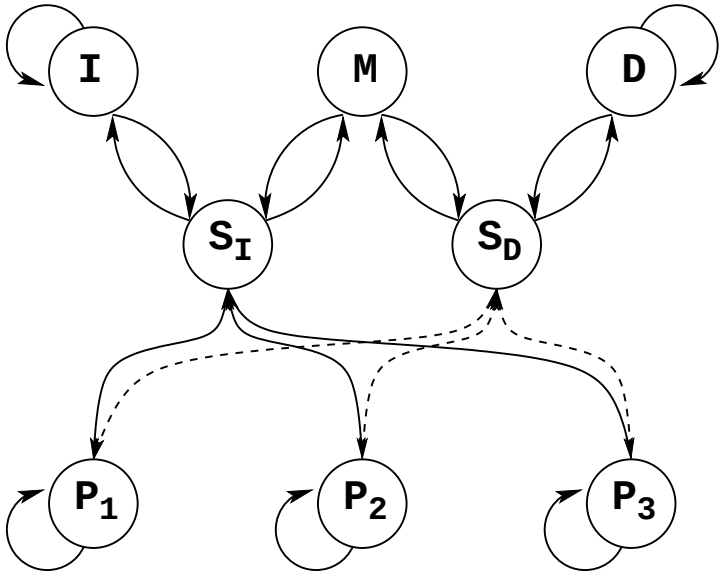


Figure 3.5: *The annotated pairHMM.* Additional states corresponding to profiles are added to the extended pairHMM of standard alignments.

The pair-profile state PP_i emits pairs of strings $(\mathbf{u}^{(i)}, \mathbf{v}^{(i)})$ of length l_i. Let us consider the basic setting first where the pair is independently sampled from the position-specific letter distribution of the profile P_i. The probability of observing this pair of strings from the i^{th} pair-profile state PP_i is given by:

$$\mathbb{P}_{\mathrm{P}_i}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)}) := \prod_{j=1}^{l_i} \mathrm{P}_i^j(u_j^{(i)})\mathrm{P}_i^j(v_j^{(i)})$$

where each string is independently sampled from the profile.

In the extended setting, the pair-profile states emit pairs of evolutionarily related instances of the profile. Hence, the first string $\mathbf{u}^{(i)}$ is sampled from the profile P_i. For a given evolutionary distance t, the second string $\mathbf{v}^{(i)}$ is derived by evolving each position j of $\mathbf{u}^{(i)}$ according to $\tilde{\rho}_i^j$. The overall probability of observing the pair under the pair-profile model is then:

$$\mathbb{P}_{\mathrm{P}_i}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)}) := \prod_{j=1}^{l_i} \mathrm{P}_i^j(u_j^{(i)})\tilde{\rho}_i^j(u_j^{(i)}, v_j^{(i)})$$

In both settings, the emission probabilities from the pair-profile states reflect the position-specific probabilities under the corresponding profile.

Log-likelihood ratio thresholds

We now turn to the issue of estimating the transition probabilities in the annotated pairHMM. Beginning with an intuitive example, we hope to show theoretically how the transition probabilities in a simple HMM can be related to thresholds in log-likelihood ratio based approaches.

An example: Consider the single sequence TFBS scanning problem for a single profile P of length l (discussed in Section 2.1.3). Modeling it as a simple HMM would imply combining two states – a *background state* B and a *profile state* P. At every visit of B a single letter from the background distribution π is emitted. At every visit to P, a single string sampled from the respective position-specific probability distribution `PSPM` is emitted. If the probability of jumping from the background to the profile state is given by p, then the transition matrix for this simple HMM is:

	$\to \mathsf{B}$	$\to \mathsf{P}$
$\mathsf{B} \to$	$1-p$	p
$\mathsf{P} \to$	$1-p$	p

where p is unknown.

The sequence scanning problem can be re-phrased as following. Given an observed sequence $\mathbf{x}$, we wish to find the sequence of hidden states that gave rise to $\mathbf{x}$. This in turn tells us the putative instances of the underlying profile in $\mathbf{x}$. Usually, this can be found by determining the Viterbi path, that is, the path through the states of the HMM for which the probability of emitting $\mathbf{x}$ is maximized. Clearly, the jumping probability p strongly influences the Viterbi path – the more likely it is to jump to the profile state, the higher chances of a putative profile instance. We now describe how a reasonable choice of p can be made using a log-likelihood ratio based approach.

The comparable log-likelihood based approach for the above-described simple HMM is as following. The two states B and P in the HMM correspond to two different probabilistic sequence models – $\mathbb{P}_{\mathsf{B}}(\mathbf{u})$ and $\mathbb{P}_{\mathsf{P}}(\mathbf{u})$ – for generating a string $\mathbf{u}$. The log-likelihood ratio (LLR) of observing the string $\mathbf{u}$ of length l in a sequence x is:

$$\mathrm{LLR}_{\mathsf{P},\mathsf{B}}(\mathbf{u}) := \log \frac{\mathbb{P}_{\mathsf{P}}(\mathbf{u})}{\mathbb{P}_{\mathsf{B}}(\mathbf{u})}$$

Here, both the probabilities are given by:

$$\mathbb{P}_{\mathsf{P}}(\mathbf{u}) := \prod_{j=1}^{l} \mathrm{P}^{j}(u_j)$$

$$\mathbb{P}_{\mathsf{B}}(\mathbf{u}) := \prod_{j=1}^{l} \pi(u_j).$$

If the LLR score exceeds a given threshold t, then $\mathbf{u}$ is considered an instance of the profile given by P, else not. For a given sequence $\mathbf{x}$ of length n, this is carried out for all windows of length l to find putative instances of the motif. In Section 2.1.3, we presented how a concrete choice of the threshold t can be made based on the type I and type II error levels. Modifying the notations introduced there for the purpose at hand, we have the type I error given by:

$$\alpha_{\mathsf{P},\mathsf{B}}(t) := \mathbb{P}_{\mathsf{B}}(\mathrm{LLR}_{\mathsf{P},\mathsf{B}}(\mathbf{x}) > t) \tag{3.12}$$

and the type II error given by:

$$\beta_{\mathsf{P},\mathsf{B}}(t) := \mathbb{P}_{\mathsf{P}}(\mathrm{LLR}_{\mathsf{P},\mathsf{B}}(\mathbf{x}) \leq t) \tag{3.13}$$

Thus, any choice of t results in some balance between the two error probabilities and we can calculate this balance exactly. Depending on the desired type I and type II error levels, we can thus calculate the threshold t which in turn influences the true and false positive rates correspondingly.

Coming back to the HMM, we use these insights from the LLR based approach as follows. At every step in the HMM, the Viterbi path has the choice between the background state B and the profile state P. A string $\mathbf{u}$ of length l can be emitted in the given sequence through two cases:

Case 1: The next state visited is P and $\mathbf{u}$ is emitted from this state. The probability of this event is

$$p\mathbb{P}_{\mathsf{P}}(\mathbf{u})$$

Case 2: The next l successive steps go to B and $\mathbf{u}$ is emitted from them. Here the probability is

$$(1-p)^l\mathbb{P}_{\mathsf{B}}(\mathbf{u}).$$

Clearly, there are other scenarios that the final Viterbi path chooses from, but it will prefer Case 1 over Case 2 whenever

$$\mathrm{LLR}_{\mathsf{P},\mathsf{B}}(\mathbf{u}) := \log\frac{\mathbb{P}_{\mathsf{P}}(\mathbf{u})}{\mathbb{P}_{\mathsf{B}}(\mathbf{u})} > \log\frac{(1-p)^l}{p} =: t_p. \tag{3.14}$$

Above, the terms are rewritten to highlight that the choice of p can be formulated in terms of an LLR cutoff. Thus, in this simple HMM, a reasonable estimate of the transition probability can be made using the standard log-likelihood ratio test. Usually, given a testset with known instances of profile hits, the transition probability is estimated using training. The training set contains background sequences which have implanted samples from the profile. The HMM is used to annotate the sequences for putative locations of the profile hits. Comparing with the true knowledge, the corresponding true and false positive rates can be calculated. Provided that the training set is sufficiently large and unbiased, these estimates are fairly accurate. However, usually the availability of an appropriate reference training set poses problems. Through the approach of estimating the transition probability p using log-likelihood ratio test, we propose that the resulting type I and type II errors provide reasonable estimates to the true and false positive rates characterizing the statistical behaviour of the HMM given a choice of p.

Probability choice in the annotated pairHMM

We can now bring the same considerations to the annotated pair HMM. Suppose $(\mathbf{u}^{(i)}, \mathbf{v}^{(i)})$ is a pair of strings of length l_i. When focusing on PP_i we again have two cases to consider

Case 1: The next state visited is PP_i and $(\mathbf{u}^{(i)}, \mathbf{v}^{(i)})$ is emitted from it. The probability of this is

$$p_i\mathbb{P}_{\mathrm{P}_i}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)}).$$

Case 2: The next l_i successive steps go to $\mathtt{M}$ and $(\mathbf{u}^{(i)}, \mathbf{v}^{(i)})$ is emitted from them. The probability is

$$(1-\bar{p})^{l_i} T_{\mathtt{M},\mathtt{M}}^{l_i} \mathbb{P}_{\mathtt{M}}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)}),$$

where

$$\mathbb{P}_{\mathtt{M}}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)}) := \prod_{j=1}^{l_i} \pi(u_j^{(i)}) \phi(u_j^{(i)}, v_j^{(i)}).$$

In the same sense as before we see that Case 1 is preferred over Case 2 whenever

$$\mathrm{LLR}_{\mathrm{P}_i,\mathtt{M}}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)}) := \log \frac{\mathbb{P}_{\mathrm{P}_i}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)})}{\mathbb{P}_{\mathtt{M}}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)})} > \log \frac{(1-\bar{p})^{l_i} T_{\mathtt{M},\mathtt{M}}^{l_i}}{p_i} =: t_{p,i}.$$

(Recall that T is the transition matrix which would have been used in the pHMM, that is the version without any silent states.)

Since both P_i and $\mathtt{M}$ provide a probabilistic model for the pair $(\mathbf{u}^{(i)}, \mathbf{v}^{(i)})$ of length l_i sequences, we can again calculate the type-I and type-II errors, which are

$$\alpha_{\mathrm{P}_i,\mathtt{M}}(t) := \mathbb{P}_{\mathtt{M}}(\mathrm{LLR}_{\mathrm{P}_i,\mathtt{M}}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)}) > t)$$

and

$$\beta_{\mathrm{P}_i,\mathtt{M}}(t) := \mathbb{P}_{\mathrm{P}_i}(\mathrm{LLR}_{\mathrm{P}_i,\mathtt{M}}(\mathbf{u}^{(i)}, \mathbf{v}^{(i)}) \leq t).$$

Hence, using the ppHMM for TFBS annotation on a pair of sequences relies on the choice of the transition probability to the corresponding pair-profile state. We presented how this probability can be reasonably estimated without relying on extensive training sets and based on sole statistical considerations. This jumping probability p influences the true and false positive rates of the ppHMM and can be interpreted as the threshold in a log-likelihood ratio test governing the corresponding type I and II errors $\alpha_{\mathrm{P}_i,\mathtt{M}}(t_{p,i})$ and $\beta_{\mathrm{P}_i,\mathtt{M}}(t_{p,i})$.

3.4 Studying the properties of annotated alignments

Modifying the standard alignments by introducing additional states clearly influences alignment properties. In our case, these states correspond to profiles and hence the ensuing impact depends on the profile considered. In this section, our objective is twofold – study how the quality of profile exerts influence and analyze the effect of varying standard alignment and profile parameters on the proportion of substitutions, indels and pair-profiles. To this end, we run SimAnn (basic formulation) on simulated random sequence pairs of a fixed sequence length using different parameter combinations and profiles.

3.4.1 Simulation setting

The experimental set-up is as follows. Random sequence pairs of a fixed length are generated and analyzed with SimAnn using three profiles, each of length 9, of good, medium and poor quality, respectively. Here, and in upcoming sections, we use the *balanced quality* for profiles as described in Section 2.1.3.

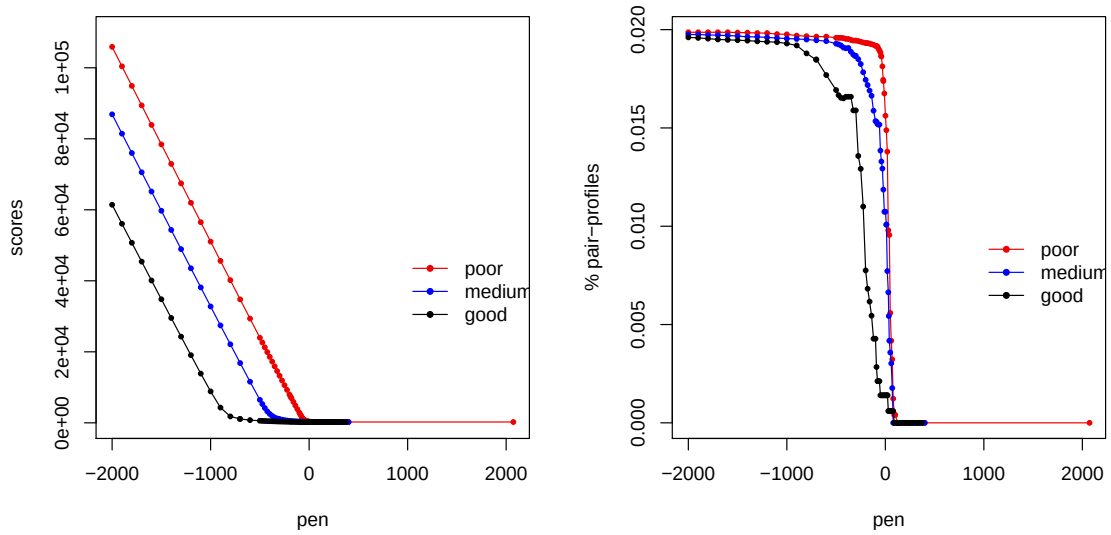


Figure 3.6: *Characteristics of annotated alignments.* The left plot depicts the almost linear increase in alignment scores with decreasing profile penalties for three profiles of poor, medium and good quality. The alignment scores are highest in the case of the poor profile and decrease as quality of profile improves. The right plot shows the variation in the proportion of pair-profile hits as penalty decreases. The alignments get saturated with pair-profile hits at extremely low cutoffs. As profile quality improves, the rise to this saturation is slower.

The respective count matrices are retrieved from the TRANSFAC [123] database. Each is used to formulate the *basic* `PSA`. A wide range of values for the profile penalty is used. For the standard alignment parameters, we consider a fixed match score and mismatch and gap costs corresponding to two main settings. First, where mismatches and gaps are discouraged (high costs) and second, where they are allowed (low costs). Since our focus is on the broad interplay between substitutions and indels on one side and pair-profiles on the other side, we take the gap extension costs arbitrarily to be half of the gap opening costs.

3.4.2 Results and analysis

Consider the first standard alignment parameter setting along with high profile penalties. Here, only matches are allowed. It is clear that while aligning random sequences, the algorithm yields short consecutive stretches of matches as all else is prohibited. Indeed, the proportion of matches in the alignment approaches 1 for all profiles (Figure 3.8(a)), while that for mismatches, indels and pair-profiles it is almost negligible (Figures 3.8(b), 3.7).

As profile penalties decrease, pair-profiles start competing with the matches. However, the balance is still tilted towards matches since the algorithm is unable to extend the alignment by filling in gaps, etc. Alignment scores increase almost linearly with decreasing profile penalty. The scores are highest for the poor profile and lowest for the good profile. Being more degenerate, the poor profile has a `PSA` which assigns greater scores to random pairs of strings. In contrast, the good profile severely penalizes non-consensus pairs at a

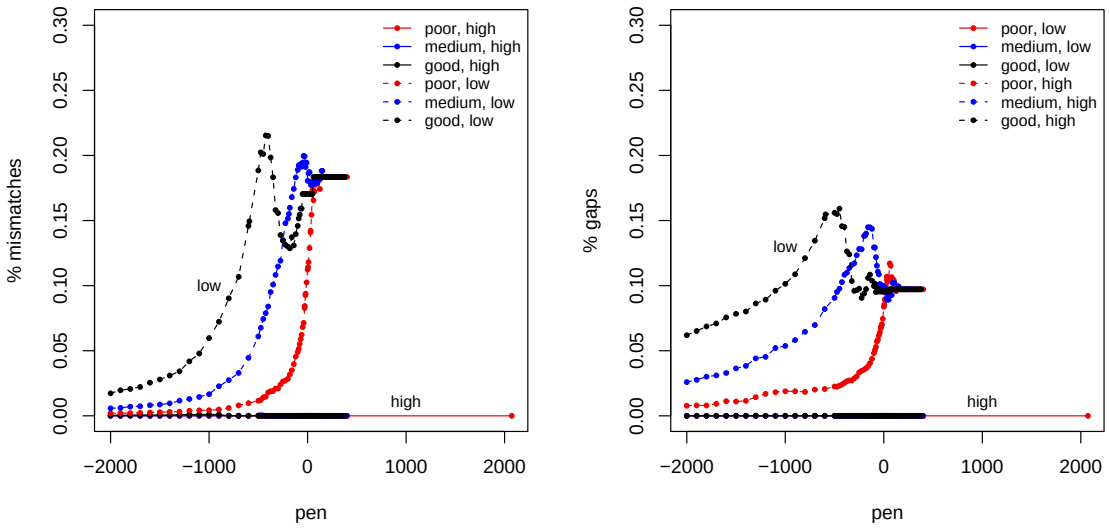


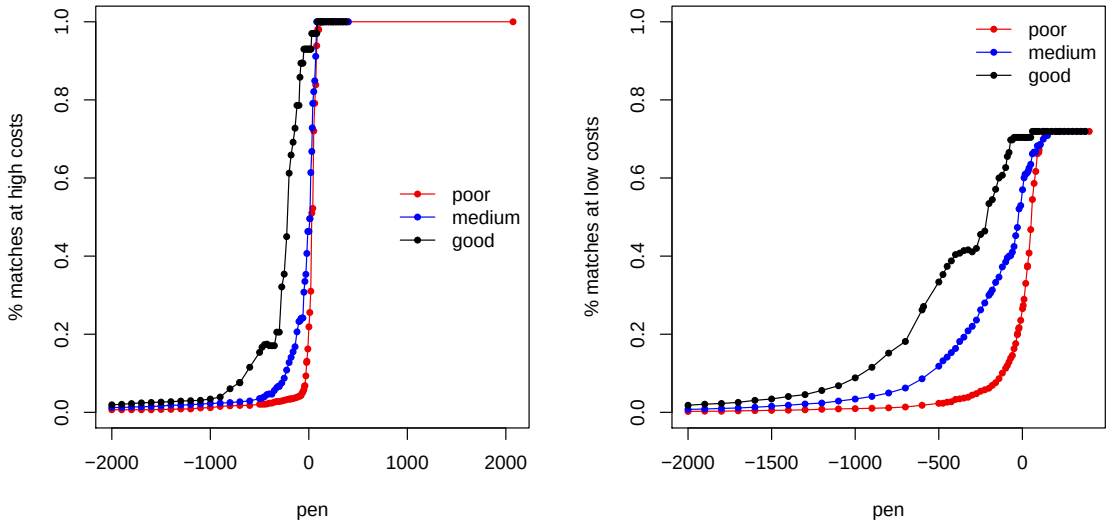
Figure 3.7: *Proportion of mismatches and indels.* As would be expected, the proportion of both mismatches and indels is negligible at high mismatch and gap cost settings (marked "high") and increases at lower cost settings (marked "low"). For each profile, as penalty is lowered, the proportion of both mismatches (left) and gaps (right) attains a peak, with the signal being weakest for the poor profile.

position yielding lesser scores to random sequences. For instance, the worst case PSA score for the good quality profile here is −1610 while that for the poor quality profile is −403. In the pair-profile framework, this is a directly emphasizes that a good quality profile better distinguishes between random strings and profile instances. Relevant figures are shown in Figure 3.6. This also highlights that the choice of profile-related parameters needs to be equally sound as that of the standard alignment parameters, especially the profile penalty. One needs a generic strategy to estimate the profile penalty for any existing or unknown binding site profile.

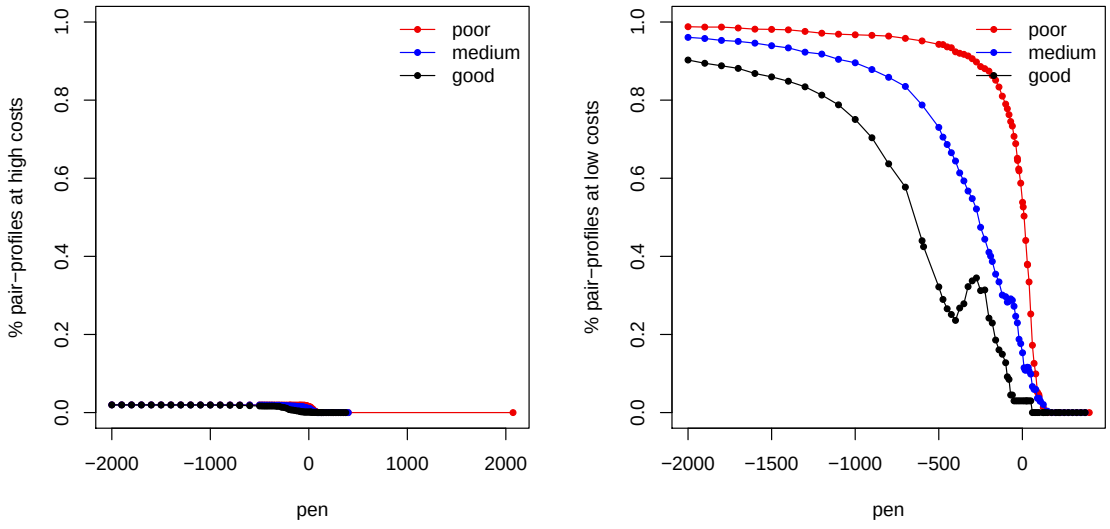
Now, consider the last (mismatches and gaps allowed) standard alignment parameter setting. At high profile penalties, the annotated alignments behave as standard alignments. Lower costs imply an increase in the proportion of mismatches and indels. It also implies that the alignments are in general longer. For all profiles, the proportion of substitutions and indels reaches the same constant level, with no pair-profiles.

As the profile penalties decrease, there is a competition/trade-off between the matches and pair-profile states (Figure 3.8, right plots). The increase in proportion of the pair-profiles is accompanied with a corresponding decrease in that of matches. The algorithm optimizes the alignment score by introducing additional mismatches and indels to accommodate more pair-profiles. This results in a rise in the proportion of mismatches and gaps for each profile, as profile penalty decreases (the peaks in Figure 3.7). With further decrease in profile penalty, the algorithm increasingly extends the alignment through pair-profiles themselves and the peak in the proportion of mismatches and gaps starts falling.

For the good profile, curves for the substitutions and gap proportions remain higher than



(a)

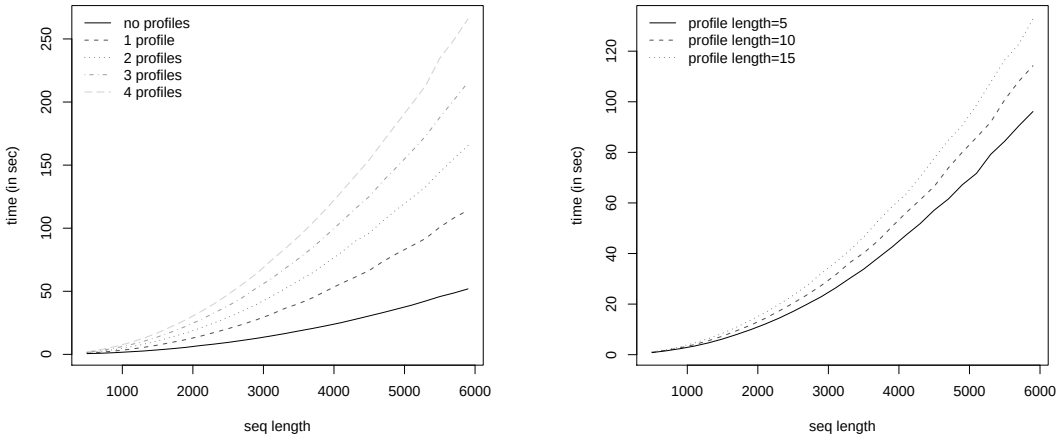


(b)

Figure 3.8: *Proportion of matches and pair-profile hits.* At low mismatch and gap costs, it becomes easier for the alignment algorithm to extend the alignment using more pair-profile states (bottom). With decreasing profile penalties, pair-profiles dominate the alignment, the rate being fastest in case of the poor profile (in red, bottom right). Overall, at high cost settings, the alignment length is smaller since short contiguous stretches of matches are output.

the other two. This indicates that for the good profile, the algorithm does not falsely (since random sequences) annotate as much pair-profiles as the weaker profiles. Hence, the lowest pair-profile curve for the good profile (Figure 3.8).

Comments The straightforward observation from the above simple analysis is that standard alignments and pair-profiles compete for alignment space in annotated alignments. Additionally, in the case of random sequences, the competition favors the pair-profile part more when the profile considered is poorer as opposed to more specific. The basic `PSA` better distinguishes random sequence pairs from TFBS hit pairs for the good profile. And finally, the profile penalty needs to be chosen such that it takes this aspect into consideration.



(a) (b)

Figure 3.9: *Time complexity.* **a)** Running time of SimAnn at varying sequence lengths and increasing number of profiles. **b)** Running time at varying sequence lengths with a single profile of lengths $5, 10, 15$.

3.5 Influence of number of profiles and sequence lengths

We now turn to the empirical validation of the computational complexity (Section 3.1) of the proposed algorithm. For random sequence pairs of increasing lengths $(500 - 6000)$ and increasing number of profiles $(0 - 4)$, Figure 3.9(a) depicts the running time required. As expected, with increasing sequence lengths, the computational time increases quadratically. When no profiles are present, the case is reduced to that of a standard alignment, and with increasing number of profiles the curves shift higher. In Figure 3.9(b), the running time for simulations with a single profile but of varying lengths is plotted. As profile length increases, so does the computation time. The current implementation uses a modified dynamic programming matrix structure where each cell has additional parameters in the presence of profiles. This implies an increase in the space requirements as compared to the standard alignment (no profile) case, although still remaining quadratic in the sequence length (data not shown). Since the parameter calculation is not part of the core algorithm, we choose the basic formulation (independent samples) for the `PSA`s and the respective penalties. All experiments were performed on an Intel(R) Xeon(TM) (2.40GHz) machine.