Aligning alignments*

John D. Kececioglu[†] Weiqing Zhang[‡]

Abstract While the area of sequence comparison has a rich collection of results on the alignment of two sequences, and even the alignment of multiple sequences, there is little known about the alignment of two *alignments*. The problem becomes interesting when the alignment objective function counts gaps, as is common when aligning biological sequences, and has the form of the sum-of-pairs objective. We begin a thorough investigation of aligning two alignments under the sum-of-pairs objective with general linear gap costs when either of the two alignments are given in the form of a *sequence* (a degenerate alignment containing a single sequence), a *multiple alignment* (containing two or more sequences), or a *profile* (a representation of a multiple alignment often used in computational biology). This leads to five problem variations, some of which arise in widely-used heuristics for multiple sequence alignment, and in assessing the relatedness of a sequence to a sequence family. For variations in which exact gap counts are computationally difficult to determine, we offer a framework in terms of *optimistic* and *pessimistic* gap counts. For optimistic and pessimistic gap counts we give efficient algorithms for the sequence vs. alignment, sequence vs. profile, alignment vs. alignment, and profile vs. profile variations, all of which run in essentially $O(mn)$ time for two input alignments of lengths m and n. For exact gap counts, we give the first provably efficient algorithm for the sequence vs. alignment variation, which runs in essentially $O(mn \log n)$ time using the candidate-list technique developed for convex gap-costs, and we conjecture that the alignment vs. alignment variation is NP-complete.

Keywords Sequence comparison, sum-of-pairs alignment, affine gap costs, quasi-natural gap costs, profiles

1 Introduction

While the comparison of two or more *sequences* is a richly developed area that has now made its way into several texts [19, 24, 20, 12], it is interesting that the seemingly similar problem of comparing two *alignments* has received little

---

*To appear in Proceedings of the 9th Annual Symposium on Combinatorial Pattern Matching (CPM'98).

[†]Department of Computer Science, The University of Georgia, Athens, GA 30602-7404. Electronic mail: `kece@cs.uga.edu`. Research supported by a National Science Foundation CAREER Award, Grant DBI-9722339.

[‡]Department of Medicinal Chemistry, The University of Georgia, Athens, GA 30602. Electronic mail: `wzhang@rx.uga.edu`

attention, even though it arises naturally when constructing approximate multiple sequence alignments [2, 18], and evaluating the relatedness of a sequence to a sequence family [25]. Given that an alignment can be viewed as a kind of sequence where each element is a column of characters, it may seem at first glance that the problem of aligning two alignments is in principle no harder than aligning two sequences, yet the problem becomes challenging when the objective function scores the resulting multiple alignment under the commonly-used sum-of-pairs measure with gaps having a cost that is a general linear function of their length.[1] In this context, the well-known dynamic programming solution for two sequences [7], which corresponds to a system of three mutual recurrences, breaks down.

In aligning two alignments under the sum-of-pairs objective with linear gap costs, we consider three forms in which an alignment may be given: (1) as an ordinary *sequence*, which is really a degenerate multiple alignment, (2) as a nondegenerate *alignment* of two or more sequences, or (3) as a *profile* [10], a representation of a multiple alignment often used in computational biology in which each column of the alignment is reduced to a distribution giving the frequency of each character of the alphabet at the particular column. Combining all pairs of forms gives rise to five new problem variations, and we consider each variation in turn, except for the alignment vs. profile variation which does not appear relevant. For many variations, accurately counting gaps is either difficult or impossible, and in this context we revisit the "quasi-natural gap costs" of Altschul [1] within a uniform framework of *optimistic* and *pessimistic* gap counts, which generalize to variations involving profiles. For optimistic and pessimistic gap counts, we give efficient algorithms for all problem variations that run in essentially $O(mn)$ time for two input alignments of lengths m and n. For exact gap counts, we give the first provably efficient algorithm for the sequence vs. alignment variation, which runs in essentially $O(mn \log n)$ time and corresponds to solving a system of 13 recurrence equations.

Our initial motivation for this study came from the problem in computational biology of assessing the relatedness, or homology, of a given sequence to a known sequence family. The second author and E. Will Taylor approached the first author with the problem of objectively assessing the significance of homology between an aligned protein family and a proposed member where the pairwise similiarity between the sequence and any isolated member of the family was quite weak. (Specifically, the problem was to determine whether a newly-sequenced HIV gene was homologous to the selenium dependent glutathione peroxidase protein family, which if true would implicate selenium deficiency in the onset of AIDS [21].) The traditional solution to this problem for two sequences is to quantify the significance of their homology by measuring a shuffling statis-

[1] The *sum-of-pairs* objective function scores a multiple alignment by the sum of the scores of its induced pairwise alignments. A *gap* in a pairwise alignment is a maximal run of insertions or deletions. We say the alignment scoring function uses *linear gap costs* when a gap of length $x > 0$ costs $a + bx$ for constants a and b, which is sometimes referred to as *affine gap costs* in the literature.

tic, which tests how many standard deviations their optimal alignment score is above a sample of optimal alignment scores for random sequences having the same composition of letters. Since pairwise similarity between the sequence and the family members was low, standard programs that evaluate the shuffling statistic between two sequences gave low measures of significance of homology. A much more sensitive approach is to use an algorithm for optimal alignment of the sequence to the family (which we call here the sequence vs. alignment variation) when evaluating the shuffling statistic. This in fact gave a significance of homology that was sufficiently high to support membership of the HIV gene in the peroxidase family. (A full study measuring sequence-family homology using the algorithm in this paper for the sequence vs. alignment variation with pessimistic gap costs, with a comparison among several definitions of the shuffling statistic, is reported in [25].)

The plan of the paper is as follows. In the first section we review the solution to the sequence vs. sequence problem with linear gap costs. The following sections generalize this to sequence vs. alignment, alignment vs. alignment, sequence vs. profile, and profile vs. profile variations, discussing optimistic, pessimistic, and exact gap counts. For each problem, we present the solution in terms of dynamic programming recurrences with complete boundary conditions in a form that can be readily implemented. (In this proceedings we omit proofs of correctness for the recurrences and time bounds for reasons of limited space.) We then conclude with some open problems.

2 Sequence vs. sequence

We first review how to efficiently compute an optimal alignment of a sequence A with a sequence B with linear gap costs, which was solved by Gotoh [7]. To specify the input, we denote A by the sequence $a_1 a_2 \cdots a_m$ and B by the sequence $b_1 b_2 \cdots b_n$. We refer to a *substring* of a sequence by subscripting with the range of positions: $A_{i,j} := a_i a_{i+1} \cdots a_j$.

To describe the alignment scoring function, we use δ to denote the *substitution costs* (such as the PAM [4] or BLOSUM [13] matrices in the form of dissimilarity scores), and we write $\delta(a, b)$ for the cost of substituting character a by character b. For the gap costs we use γ to denote the *gap start-up cost* and λ to denote the *gap extension cost*. Thus a gap of length $x > 0$ has cost $\gamma + \lambda x$. We denote the *null character* (usually represented in an alignment by a dash) by ϵ, and assume that for any non-null character a, $\delta(a, \epsilon) = \delta(\epsilon, a) = \lambda$, while $\delta(\epsilon, \epsilon) = 0$. Together, δ, γ, and λ specify the alignment scoring function.

Recall that the key to solving the alignment problem with linear gap costs is to record how the optimal alignment over a prefix of A and a prefix of B *ends*, since the gap cost associated with a given alignment column is completely determined by the preceding column. An optimal alignment can end in one of three ways. Let

- $D(i, j)$ be the cost of an optimal aligment of $A_{1,i}$ versus $B_{1,j}$ that ends by substituting a_i with b_j (which corresponds to a *diagonal* edge from node $(i-1, j-1)$ to (i, j) in the standard dynamic programming graph),

- $V(i, j)$ be the cost of an optimal alignment of $A_{1,i}$ versus $B_{1,j}$ that ends by deleting a_i (which corresponds to a *vertical* edge into node (i, j)), and
- $H(i, j)$ be the cost of an optimal alignment of $A_{1,i}$ versus $B_{1,j}$ that ends by inserting b_j (which corresponds to a *horizontal* edge into node (i, j)).

Then the cost of an optimal alignment of A and B is

$$\min\{D(m,n), V(m,n), H(m,n)\}.$$

We now give the recurrences for D, V, and H, with complete boundary conditions. In the recurrences, cost γ is incurred once at the start of a gap.

For $i > 0$ and $j > 0$,

$$H(i,j) = \lambda + \min \begin{cases} H(i,j-1), \\ V(i,j-1) + \gamma, \\ D(i,j-1) + \gamma \end{cases} \tag{1}$$

$$V(i,j) = \lambda + \min \begin{cases} H(i-1,j) + \gamma, \\ V(i-1,j), \\ D(i-1,j) + \gamma \end{cases} \tag{2}$$

$$D(i,j) = \delta(a_i, b_j) + \min \begin{cases} H(i-1,j-1), \\ V(i-1,j-1), \\ D(i-1,j-1) \end{cases} \tag{3}$$

For $i = 0$ and $j > 0$,

$$H(0,j) = H(0,j-1) + \lambda, \tag{4}$$
$$V(0,j) = H(0,j) + \gamma, \tag{5}$$
$$D(0,j) = H(0,j). \tag{6}$$

For $i > 0$ and $j = 0$,

$$V(i,0) = V(i-1,0) + \lambda, \tag{7}$$
$$H(i,0) = V(i,0) + \gamma, \tag{8}$$
$$D(i,0) = V(i,0). \tag{9}$$

For $i = 0$ and $j = 0$,

$$H(0,0) = \gamma, \tag{10}$$
$$V(0,0) = \gamma, \tag{11}$$
$$D(0,0) = 0 \tag{12}$$

Recurrences (1) through (3) are often presented in the literature in an optimized form. The recurrence for $D(i,j)$ is replaced by a recurrence for a quantity $S(i,j)$ which represents $\min\{D(i,j), V(i,j), H(i,j)\}$. This yields a system that references only seven terms at an (i, j)-position, instead of nine as above. Interestingly, this optimization does not generalize to the alignment problems considered in this paper. Hence we present the basic sequence-sequence recurrences in the above unoptimized form as this makes comparison with the more general recurrences more straightforward.

Evaluating the above recurrences in three $(m+1) \times (n+1)$ tables that correspond to H, V, and D above, and tracing back through the tables to recover the optimal alignment once the solution value is known, gives the following result.

Theorem 1 (Gotoh [7]) *An optimal alignment of two sequences of lengths m and n with linear gap costs can be computed in $O(mn)$ time.*

Myers and Miller [17] show that an optimal alignment with linear gap costs can be found in the above time using only $O(\min\{m, n\})$ working space, by applying a technique of Hirschberg [14].

3 Sequence vs. alignment

As soon as one generalizes from comparing two sequences to comparing several sequences or, as considered in this paper, to comparing two objects one or both of which may be alignments, the problem of optimal alignment with linear gap costs becomes more difficult. When aligning several sequences or two alignments, the output is a multiple sequence alignment, and a common objective function on multiple alignments is the *sum-of-pairs* objective [3], which takes as the cost of a multiple alignment the sum of the costs of all induced pairwise alignments. If in the sum-of-pairs objective pairwise alignments are scored with linear gap costs, the problem of counting gaps in the dynamic programming recurrences arises. Unfortunately in this more general setting it is no longer possible to determine when a gap is started by keeping track of the preceding column of the alignment, or for that matter any fixed number of preceding columns. To deal with this difficulty, Altschul [1] introduced what were termed "quasi-natural gap costs." This is a practical suggestion, in which an induced pairwise alignment that ends with the pattern $\begin{smallmatrix}-\mathtt{X}\\--\end{smallmatrix}$ or $\begin{smallmatrix}--\\-\mathtt{X}\end{smallmatrix}$, where `X` represents a non-null character, is considered to start a gap, even though it may not. (For all other patterns there is no difficulty in determining the start of a gap.) While this will not give exact gap counts, it allows gap start-ups to be counted on the basis of the preceding column alone as in Gotoh's solution of Section 2, and hopefully overestimates the true count only rarely.

In this paper we refer to this type of compromise in terms of *optimistic* and *pessimistic* gap counts. Pessimistic gap counts count one gap start-up in the situation $\begin{smallmatrix}-\mathtt{X}\\--\end{smallmatrix}$ and $\begin{smallmatrix}--\\-\mathtt{X}\end{smallmatrix}$, while optimistic gap counts do not. Pessimistic counts are used in the multiple alignment program `MSA` [15, 11] which scores alignments under the sum-of-pairs objective; pessimistic counts are needed in this context instead of optimistic counts so that the sum of all optimal pairwise alignments yields a lower bound on the objective function. However, pessimistic gap counts can overestimate the exact count by an arbitrarily large value, and it can be argued that optimistic counts are likely to yield a more accurate estimate.

The next section develops recurrences for the sequence vs. alignment variation with optimistic or pessimistic gap counts. Following this we give a more involved system of recurrences that solves the problem with *exact* gap counts.

3.1 Optimistic and pessimistic gap counts

In the sequence vs. alignment variation, the input is a sequence A and an alignment B. We again denote A by the sequence $a_1 \cdots a_m$, but now B is represented by a $k \times n$ matrix (b_{ij}), where each character b_{ij} is either a symbol from the alphabet Σ or the null character ϵ. (Thus B is a multiple alignment of length n of k sequences.) The problem is to align the characters of A to the columns of B so as to minimize the sum over all rows of B of the cost of the induced pairwise alignment between A and the given row of B, where pairwise alignments are scored with linear gap costs. This differs from general sum-of-pairs multiple alignment [3] (which is intractable [22]) in that the alignment on the rows of B is fixed. The alignment cost function is again given by substitution costs δ, gap start-up cost γ, and gap extension cost λ.

To count whether or not an element b_{ij} of alignment B is the null character, we define

$$\beta_{ij} \;:=\; \begin{cases} 1, & b_{ij} \neq \epsilon \text{ and } j > 0; \\ 0, & b_{ij} = \epsilon \text{ and } j > 0; \\ 1, & j = 0. \end{cases}$$

(This treats alignment B as starting with an artificial 0th column of non-null characters for the boundary conditions of the recurrences.) The number of non-null characters in column j of B is then $f_1(j) \;:=\; \sum_{1 \le i \le k} \beta_{ij}$.

To count gap start-ups we use a function $\Gamma(a_1 a_2, b_1 b_2)$ given in tabular form below. Essentially, $a_1 a_2$ is a binary string that represents whether the last two entries of the row for A are non-null characters in the alignment, while $b_1 b_2$ is a binary string that represents whether the last two entries of a given row for B are non-null characters in the alignment. In the table for Γ, rows are indexed by $a_1 a_2$, and columns by $b_1 b_2$.

	00	01	10	11
00	0	$\{0,1\}$	0	0
01	$\{0,1\}$	0	1	0
10	0	1	0	1
11	0	0	1	0

The two entries of Γ that differ for optimistic and pessimistic gap counts contain the set $\{0,1\}$; with optimistic counts both entries are 0, while for pessimistic counts both entries are 1.

We can now write down recurrences for the value of an optimal alignment with optimistic or pessimistic gap costs. As in Section 2, let $H(i,j)$, $V(i,j)$, and $D(i,j)$ be the cost of an optimal alignment of the first i characters of A against the first j columns of B that ends with an insertion, deletion, or substitution, respectively. For $i > 0$ and $j > 0$,

$$H(i,j) \;=\; \lambda\, f_1(j) \;+\; \min \begin{cases} H(i,j-1) \;+\; \gamma \displaystyle\sum_{1 \le r \le k} \Gamma(00, \beta_{rj-1}\beta_{rj}), \\ V(i,j-1) \;+\; \gamma \displaystyle\sum_{1 \le r \le k} \Gamma(10, 0\beta_{ij}), \\ D(i,j-1) \;+\; \gamma \displaystyle\sum_{1 \le r \le k} \Gamma(10, \beta_{rj-1}\beta_{rj}) \end{cases} \tag{13}$$

$$V(i,j) \;=\; \lambda\, k \;+\; \min \begin{cases} H(i-1,j) \;+\; \gamma \displaystyle\sum_{1 \le r \le k} \Gamma(01, \beta_{rj}0), \\ V(i-1,j) \;+\; \gamma \displaystyle\sum_{1 \le r \le k} \Gamma(11, 00), \\ D(i-1,j) \;+\; \gamma \displaystyle\sum_{1 \le r \le k} \Gamma(11, \beta_{rj}0) \end{cases} \tag{14}$$

$$D(i,j) \;=\; \sum_{1 \le r \le k} \delta(a_i, b_{rj}) \;+\; \min \begin{cases} H(i-1,j-1) \;+\; \gamma \displaystyle\sum_{1 \le r \le k} \Gamma(01, \beta_{rj-1}\beta_{rj}), \\ V(i-1,j-1) \;+\; \gamma \displaystyle\sum_{1 \le r \le k} \Gamma(11, 0\beta_{rj}), \\ D(i-1,j-1) \;+\; \gamma \displaystyle\sum_{1 \le r \le k} \Gamma(11, \beta_{rj-1}\beta_{rj}) \end{cases} \tag{15}$$

The recurrences can be simplified using the following counts. Let $f_b(j)$ count the number of occurrences of character $b \in \Sigma$ in column j of B, and for $x, y \in \{0,1\}$, let us count patterns of null- and non-null characters by

$$\begin{aligned} f_x(j) \;&:=\; \sum_{1 \le r \le k} \xi(x, \beta_{rj}), \\ f_{xy}(j) \;&:=\; \sum_{1 \le r \le k} \xi(x, \beta_{rj-1})\, \xi(y, \beta_{rj}), \end{aligned}$$

where $\xi(x,y)$ is 1 if $x = y$ and 0 otherwise. Finally, let $f_{\perp 1}(j)$ denote the number of non-null characters in column j of B that are the *first* non-null character in their row. The recurrences in terms of these functions, together with complete boundary conditions, are given below. The boundary conditions count gaps exactly.

For $i > 0$ and $j > 0$,

$$H(i,j) \;=\; \lambda\, f_1(j) \;+\; \min \begin{cases} H(i,j-1) \;+\; \gamma \begin{Bmatrix} 0, & \text{opt;} \\ f_{01}(j), & \text{pess.} \end{Bmatrix}, \\ V(i,j-1) \;+\; \gamma\, f_1(j), \\ D(i,j-1) \;+\; \gamma\, f_1(j) \end{cases} \tag{16}$$

$$V(i,j) \;=\; \lambda\, k \;+\; \min \begin{cases} H(i{-}1,j) \;+\; \gamma \left\{ \begin{array}{ll} f_1(j), & \text{opt;} \\ k, & \text{pess.} \end{array} \right\}, \\ V(i{-}1,j), \\ D(i{-}1,j) \;+\; \gamma\, f_1(j) \end{cases} \tag{17}$$

$$D(i,j) \;=\; \sum_{b \,\in\, \Sigma\cup\{\epsilon\}} f_b(j)\, \delta(a_i, b) \;+\; \min \begin{cases} H(i{-}1,j{-}1) \;+\; \gamma \left\{ \begin{array}{ll} f_{10}(j), & \text{opt;} \\ f_0(j), & \text{pess.} \end{array} \right\}, \\ V(i{-}1,j{-}1), \\ D(i{-}1,j{-}1) \;+\; \gamma\, f_{10}(j) \end{cases} \tag{18}$$

For $i = 0$ and $j > 0$,

$$H(0,j) \;=\; H(0,j{-}1) \;+\; \lambda\, f_1(j) \;+\; \gamma\, f_{\perp 1}(j), \tag{19}$$
$$V(0,j) \;=\; H(0,j) \;+\; \gamma\, k, \tag{20}$$
$$D(0,j) \;=\; V(0,j). \tag{21}$$

For $i > 0$ and $j = 0$,

$$V(i,0) \;=\; V(i{-}1,0) \;+\; \lambda\, k, \tag{22}$$
$$H(i,0) \;=\; V(i,0) \;+\; \gamma\, f_1(1), \tag{23}$$
$$D(i,0) \;=\; H(i,0). \tag{24}$$

For $i = 0$ and $j = 0$,

$$V(0,0) \;=\; \gamma\, k, \tag{25}$$
$$H(0,0) \;=\; 0, \tag{26}$$
$$D(0,0) \;=\; 0. \tag{27}$$

It is straightforward to verify that recurrences (16) through (18) are faithful to (13) through (15), and to check boundary conditions (19) through (27). Evaluating these recurrences in three tables, and tracing back to recover the optimal alignment, gives the following result.

Theorem 2 *An optimal alignment of a sequence versus an alignment under the sum-of-pairs objective with optimistic or pessimistic gap counts can be computed in time*

$$O(kn \;+\; \min\{s,k\}mn),$$

where m is the length of the input sequence, n is the length of the input alignment, k is the number of sequences in the input alignment, and s is the size of the alphabet.

The factor of $\min\{s,k\}$ in the running time comes from evaluating the substitution cost for $D(i,j)$ by summing over the s symbols in the alphabet, as shown in (18), or by summing over the k entries in the column of B.

3.2 Exact gap counts

Examining the optimistic-pessimistic recurrences, the only source of inaccuracy is in counting gap start-ups in the HH, HV, and HD cases of equations (16) through (18), in which the second-to-last operation is an insertion of a column of B, corresponding to the H-term in the minimum. These are the only cases in which the pattern $\begin{smallmatrix} - & - \\ - & \mathtt{X} \end{smallmatrix}$ or $\begin{smallmatrix} - & \mathtt{X} \\ - & - \end{smallmatrix}$ can arise, which are the only situations in which the optimistic-pessimistic gap counts are inaccurate. We now develop recurrences that handle these three cases exactly.

Developing an exact system of recurrences

The key observation is the following. In the HH, HV, and HD cases, the insertion that is the second-to-last operation may, in an optimal alignment, be itself preceded by an insertion, giving rise to cases HHH, HHV, and HHD. This third-to-last insertion may also contain the pattern $\begin{smallmatrix} - \\ - \end{smallmatrix}$, and may, in an optimal alignment, be preceded by yet another insertion. Notice, however, that in these cases with an HH$\cdots$H run, eventually the run of insertions must be preceded by (1) a substitution, corresponding to $\cdots$DHH$\cdots$H, (2) a deletion, corresponding to $\cdots$VHH$\cdots$H, or (3) go all the way back to the beginning of B, corresponding to HH$\cdots$H. In all three situations, any gap start-ups in the run of H's can be counted exactly, by determining whether a given character b_{ij} in B is the first non-null character in its row encountered during the run. Incorporating this idea results in the following system of recurrences for exact gap counts.

For $i > 0$ and $j > 0$,

$$H(i,j) \;=\; \lambda\, f_1(j) \;+\; \min \begin{cases} H_H(i, j-1), \\ V(i, j-1) \;+\; \gamma\, f_1(j), \\ D(i, j-1) \;+\; \gamma\, f_1(j) \end{cases} \tag{28}$$

$$V(i,j) \;=\; \lambda\, k \;+\; \min \begin{cases} H_V(i-1, j), \\ V(i-1, j), \\ D(i-1, j) \;+\; \gamma\, f_1(j) \end{cases} \tag{29}$$

$$D(i,j) \;=\; \sum_{b \,\in\, \Sigma \cup \{\epsilon\}} f_b(j)\, \delta(a_i, b) \;+\; \min \begin{cases} H_D(i-1, j-1), \\ V(i-1, j-1), \\ D(i-1, j-1) \;+\; \gamma\, f_{10}(j) \end{cases} \tag{30}$$

$$H_H(i,j) \;=\; \begin{cases} \displaystyle\min_{0 \le j' < j} \left\{ \min \begin{Bmatrix} V(i, j'), \\ D(i, j') \end{Bmatrix} + G_H(j', j) \right\}, & j > 0; \\ H(i, 0), & j = 0. \end{cases} \tag{31}$$

$$H_V(i,j) \;=\; \begin{cases} \min \begin{Bmatrix} \displaystyle\min_{0 \le j' < j} \left\{ V(i, j') \;+\; G_{VV}(j', j) \right\}, \\ \displaystyle\min_{0 \le j' < j} \left\{ D(i, j') \;+\; G_{DV}(j', j) \right\} \end{Bmatrix}, & i > 0; \\ H(0, j) \;+\; \gamma\, k, & i = 0. \end{cases} \tag{32}$$

$$H_D(i,j) = \begin{cases} \min\left\{ \begin{array}{l} \min\limits_{0\le j'<j} \left\{ V(i,j') + G_{VD}(j',j) \right\}, \\ \min\limits_{0\le j'<j} \left\{ D(i,j') + G_{DD}(j',j) \right\} \end{array} \right\}, & i>0 \text{ and } j>0; \\ V(i,0), & i>0 \text{ and } j=0; \\ H(0,j) + \gamma\, f_0(j{+}1), & i=0. \end{cases} \tag{33}$$

$$G_H(j',j) = F(j',j) + g_H(j, j-j'), \tag{34}$$

$$G_{VV}(j',j) = F(j',j) + g_V(j, j-j'), \tag{35}$$

$$G_{DV}(j',j) = F(j',j) + g_V(j, j-j'+1), \tag{36}$$

$$G_{VD}(j',j) = F(j',j) + g_D(j, j-j'), \tag{37}$$

$$G_{DD}(j',j) = F(j',j) + g_D(j, j-j'+1), \tag{38}$$

$$F(j',j) = \left\{ \begin{array}{ll} F(j',j-1) + \lambda\, f_1(j) + g_H(j-1, j-j'-1), & j'<j; \\ 0, & j'=j. \end{array} \right\}, \tag{39}$$

$$L(i,j) = \left\{ \begin{array}{ll} L(i,j-1) + 1, & j>0 \text{ and } b_{ij} = \epsilon; \\ 0, & \text{otherwise.} \end{array} \right\}, \tag{40}$$

where we define

$$g_H(j,\ell) := \gamma \sum_{1\le r\le k} \begin{cases} 1, & b_{r\,j+1} \ne \epsilon \text{ and } L(r,j) \ge \ell; \\ 0, & \text{otherwise.} \end{cases}$$

$$g_V(j,\ell) := \gamma \sum_{1\le r\le k} \begin{cases} 1, & L(r,j) < \ell; \\ 0, & \text{otherwise.} \end{cases}$$

$$g_D(j,\ell) := \gamma \sum_{1\le r\le k} \begin{cases} 1, & b_{r\,j+1} = \epsilon \text{ and } L(r,j) < \ell; \\ 0, & \text{otherwise.} \end{cases}$$

In the above, $H_H(i,j)$, $H_V(i,j)$, and $H_D(i,j)$ are the cost of an optimal alignment of the first i characters of A versus the first j columns of B that ends by inserting the jth column of B, *plus* the cost of any gaps started by following with an insertion, deletion, or substitution, respectively. Quantity $F(j',j)$ is the cost of insertions plus gap startups for inserting columns $j'{+}1$ through j of B, while $L(i,j)$ is the length of the gap in row i of B measured starting from column j of B and extending to the left. Quantities $g_H(j,\ell)$, $g_V(j,\ell)$, and $g_D(j,\ell)$ give the cost of gap startups for following a run of insertions up through column j of B with an insertion, deletion, or substitution, respectively, where ℓ varies depending on the length of the run and whether the run is preceded by a deletion or a substitution.

The boundary conditions for the recurrences are the same as equations (19) through (27) for optimistic-pessimistic gap counts, except that equation (24) is replaced by

$$D(i,0) := V(i,0). \tag{41}$$

Straightforwardly evaluating the recurrences, using 9 tables to store V, D, F, g_H, g_V, g_D, L, f_1, f_{10}, and $|\Sigma|+1$ additional tables to hold the f_c counts for

$c \in \Sigma \cup \{\epsilon\}$, and tracing back to recover the optimal alignment, takes time

$$O(kn^2 \, + \, \min\{s,k\}mn \, + \, mn^2),$$

where m is the length of A, n is the length of B, k is the number of sequences in B, and s is the size of Σ. The first term is the time to precompute g_H, g_V, g_D, and F, the second term is the time to evaluate H, V, and D assuming H_H, H_V, and H_D are known, and the third term is the time to evaluate H_H, H_V, and H_D directly from their recurrences.

When m and n are large compared to k, which corresponds to the common situation of aligning a few long sequences, the final $O(mn^2)$ term dominates the running time, which is then cubic in the length of the inputs. We now show that by applying the candidate-list technique for sequence alignment with convex gap-costs pioneered independently by Miller and Myers [16] and Galil and Giancarlo [6] (and suggested by Waterman [23]), we can reduce the time to compute the 1-dimensional minimizations for H_H, H_V, and H_D to $O(mn \log n)$ time in total, which is close to quadratic in the length of the inputs.

Applying the candidate-list technique

Examining equations (31) through (33), we can abstract the five 1-dimensional minimization problems for H_H, H_V, and H_D into the problem of computing a function $h(j)$ for $j = 1, 2, \ldots, n$, where

$$h(j) \;\; := \;\; \min_{0 \le i < j} \Big\{ h_i(j) \Big\},$$

and each function $h_i(j)$ has the form

$$h_i(j) \;\; := \;\; a(i) \, + \, b(i,j).$$

In our context the function $a(i)$, for a fixed row r of $H_H(r,\cdot)$, $H_V(r,\cdot)$, or $H_D(r,\cdot)$, is either $V(r,i)$ or $D(r,i)$ or the minimum of the two, and the function $b(i,j)$ is either $g_H(j,j{-}i)$, $g_V(j,j{-}i)$, $g_D(j,j{-}i)$, or a slight variant. From this simplified perspective, computing function h is the problem of determining the minimum envelope of a family of curves, where curve h_i of the family starts at position $i+1$ in the domain.

If it can be shown that any two curves in the family satisfy a *crossing-once* property in the sense that once two such curves cross they never cross again, then the minimum envelope of the family can be found very efficiently [16, 6]. The key to proving the crossing-once property is to consider forward differences of the functions. Let $\Delta h_i(j)$ denote $h_i(j) \, - \, h_i(j{-}1)$.

Lemma 1 *Let functions $h_i(j)$ be defined as above for H_H, H_V, and H_D, and suppose $0 \le i \le i'$. Then for all $j > i'$,*

$$\Delta h_i(j) \;\; \le \;\; \Delta h_{i'}(j).$$

Proof sketch We can prove separately that the inequality holds when the functions h_i are defined as appropriate for H_H, H_V, or H_D.

First consider the functions h_i defined for H_H. By straightforward manipulation,

$$\Delta h_{i'}(j) \; - \; \Delta h_i(j) \;\; = \;\; g_H(j, j-i') \; - \; g_H(j, j-i).$$

Since $g_H(j, \ell)$ is monotonically decreasing in ℓ, this difference is at least zero.

Next consider the functions h_i defined for H_V. It is not hard to show that

$$\Delta h_{i'}(j) \; - \; \Delta h_i(j) \;\; = \;\; 2\Big(g_H(j-1, j-i'-1) \; - \; g_H(j-1, j-i-1)\Big),$$

which as argued above is at least zero.

Finally considering functions h_i defined for H_D, it can be shown that

$$\begin{aligned}\Delta h_{i'}(j) \; - \; \Delta h_i(j) \;\; &= \;\; \Big(g_H(j-1, j-i'-1) \; - \; g_H(j-1, j-i-1)\Big) \\ &\quad + \Big(g_H(j, j-i') \; - \; g_H(j, j-i)\Big),\end{aligned}$$

which again is at least zero. □

An easy consequence of Lemma 1 is the following.

Corollary 1 (Crossing-Once Property) *Let functions $h_i(j)$ be defined as for H_H, H_V, and H_D, and suppose $0 \le i \le i' < j$. If*

$$h_i(j) \;\; \le \;\; h_{i'}(j),$$

then for all $j' \ge j$,

$$h_i(j') \;\; \le \;\; h_{i'}(j').$$

Given Corollary 1 we can determine $h(1)$, $h(2)$, ..., $h(n)$ in $O(n \log n)$ total time. The idea is to incrementally update, as successive curves are considered, a partition of the domain of h into maximal intervals in which one curve of the family gives the minimum envelope. The extents of the intervals in the partition are determined by computing crossover points between pairs of curves in $O(\log n)$ time using binary search. (For a very readable exposition of this idea in the context of sequence alignment with convex gap-costs, see [12, pp. 293–302].) We summarize this result in the following theorem.

Theorem 3 *An optimal alignment of a sequence versus an alignment under the sum-of-pairs objective with exact gap counts can be computed in time*

$$O(kn^2 \; + \; \min\{s, k\}mn \; + \; mn \log n),$$

where m is the length of the sequence, n is the length of the input alignment, k is the number of sequences in the input alignment, and s is the size of the alphabet.

4 Alignment vs. alignment

We can generalize further to aligning two multiple alignments, which arises in many multiple alignment heuristics [18]. Now A is an alignment, denoted by a $k \times m$ matrix (a_{ij}), and B is an alignment, denoted by an $\ell \times n$ matrix (b_{ij}). The cost of an alignment of the columns of A against the columns of B is the sum of the costs of all pairwise alignments induced by choosing a row from A and a row from B. We first give recurrences for optimistic-pessimistic gap counts, and then discuss the situation with exact gap counts.

4.1 Optimistic and pessimistic gap counts

Recurrences for the alignment vs. alignment variation with optimistic or pessimistic gap counts can be derived in a similar manner as for the sequence vs. alignment case by summing over all pairs of rows from A and B and counting gap startups using table Γ. The recurrences can then be simplified by rewriting them in terms of functions f^A_{00}, f^B_{00}, $\ldots$, f^A_{11}, f^B_{11}, and $f^A_{\perp 1}$, $f^B_{\perp 1}$, which count the occurrences of patterns of null and non-null characters in the columns of A and B, defined as for the sequence vs. alignment case.

We give the alignment vs. alignment optimistic-pessimistic recurrences in Section A of the Appendix, and state the final result here.

Theorem 4 *An optimal alignment of two alignments under the sum-of-pairs objective with optimistic or pessimistic gap counts can be computed in time*

$$O(km + \ell n + \min\{s^2, k\ell\}mn),$$

where m and n are the lengths of the two alignments, k and ℓ are the corresponding numbers of sequences in the alignments, and s is the size of the alphabet.

4.2 Exact gap counts

In the alignment vs. alignment variation, the idea used to derive recurrences for exact gap counts for the sequence vs. alignment case no longer works. Since A is an alignment as well, preceding a run of insertions by a deletion or a substitution can still give rise to a pair of rows containing $\genfrac{}{}{0pt}{}{-}{-}$, in which case gap startups still cannot be resolved. In fact, we have been unable to find a representation of an alignment that allows us to count gap startups exactly and gives rise to a polynomial number of cases in the dynamic program. (In the sequence vs. alignment case, there is such a representation, namely an integer describing the length of a terminal run of insertions together with a bit describing whether the run was preceded by a deletion or a substitution.) This suggests that it may not be possible to find an optimal solution for the alignment vs. alignment case with exact gap counts in polynomial time.

Gotoh [8] suggests a procedure for the alignment vs. alignment variation that considers a set of candidate alignments at each (i, j)-subproblem in the dynamic program. The set of candidates is apparently potentially all alignments, which is reduced by testing certain inequalities. No worst-case time bound for the procedure is given, but the paper states "the total arithmetic operations used ... is close to [quadratic] in typical cases."

5 Sequence vs. profile

In the sequence vs. profile variation, B is now a profile [10], which is essentially a sequence of distributions over the alphabet, rather than a sequence of characters, obtained by reducing the columns of a multiple alignment to character counts. Many protein families currently are characterized by profiles, thus it is worthwhile developing an algorithm for aligning profiles; the scoring of such alignments with linear gap costs can be put on an objective basis within the optimistic-pessimistic framework.

We describe profile B by a sequence $f_c(1)$, $f_c(2)$, ..., $f_c(n)$, where n is the length of B, $c \in \Sigma \cup \{\epsilon\}$ is a character, and $f_c(i)$ gives the number of occurrences of character c at position i in B. We use k to denote the total number of characters (including the null character) at a position of B, which is constant across positions.

Given that the gapping structure of the individual sequences on which a profile is based is lost, it is not possible to define exact gap counts, but it is possible to define optimistic and pessimistic gap counts. We define *optimistic* gap counts for a sequence-profile alignment to be the minimum, over all multiple alignments obtained by arranging the non-null characters at each position in the profile, of the cost of the resulting sequence-alignment alignment under optimistic gap counts. Similarly, we define *pessimistic* gap counts for a sequence-profile alignment to be the maximum, over all multiple alignments obtained by arranging the non-null characters in the profile, of the cost of the resulting sequence-alignment alignment under pessimistic gap counts.

We give the recurrences for sequence-profile alignment with optimistic or pessimistic gap counts in Section B of the Appendix, and state the final result here.

Theorem 5 *An optimal alignment of a sequence versus a profile under the sum-of-pairs objective with optimistic or pessimistic gap counts can be computed in time $O(smn)$, where m is the length of the sequence, n is the length of the profile, and s is the size of the alphabet.*

6 Profile vs. profile

We can generalize to profile-profile alignment using the same notion of optimistic and pessimistic gap counts developed for sequence-profile alignment. We give the recurrences in Section C of the Appendix, and state the final result here.

Theorem 6 *An optimal alignment of two profiles under the sum-of-pairs objective with optimistic or pessimistic gap counts can be computed in time*

$O(s^2mn)$, where m and n are the lengths of the profiles, and s is the size of the alphabet.

7 Conclusions

We have presented recurrences with complete boundary conditions for the sequence vs. alignment, alignment vs. alignment, sequence vs. profile, and profile vs. profile variations under the sum-of-pairs objective with linear gap costs and optimistic or pessimistic gap counts, and recurrences for the sequence vs. alignment variation with exact gap counts. To our knowledge this is the first presentation in the literature of recurrences for each of these forms of the problem that permits direct implementation. Evaluating the recurrences yields $O(mn)$ time algorithms for all variations with optimistic or pessimistic gap counts, and by applying the candidate-list technique, an $O(mn \log n)$ time algorithm for the sequence vs. alignment variation with exact gap counts, where time is measured in terms of the lengths m and n of the input.

It may be worth noting one interesting observation arising from the sequence vs. alignment and alignment vs. alignment recurrences: With respect to sum-of-pairs alignment with optimistic or pessimistic gap counts, by storing just four additional integers at each position in a profile (namely the counts f_{00}, f_{01}, f_{10}, and f_{11}), profiles become equivalent in descriptive power to the original multiple alignment of the family.

Further research

The main unresolved theoretical problem is the alignment vs. alignment variation with exact gap counts. We suspect one can construct an input that forces Gotoh's procedure [8] for the alignment vs. alignment variation to take exponential time, and we conjecture that the problem in general is NP-complete. An intractability result for this problem would clarify why it is so hard to incorporate exact gap counts into many already NP-complete multiple alignment problems (see for instance [11]).

Finally from a practical point of view, it would be interesting to perform a computational study comparing optimistic versus pessimistic versus exact gap counts, and full alignments versus profiles, now that complete recurrences are available for the problem variations.

References

[1] Altschul, S.F. "Gap costs for multiple sequence alignment." *Journal of Theoretical Biology* 138, 297–309, 1989.

[2] Anson, E.L. and E.W. Myers. "ReAligner: a program for refining DNA sequence multi-alignments." Proceedings of the 1st ACM Conference on *Computational Molecular Biology*, 9–13, 1997.

[3] Carrillo, H. and D. Lipman. "The multiple sequence alignment problem in biology." *SIAM Journal on Applied Mathematics* 48, 1073–1082, 1988.

[4] Dayhoff, M.O., R.M. Schwartz and B.C. Orcutt. "A model of evolutionary change in proteins." In *Atlas of Protein Sequence and Structure* 5:3, M.O. Dayhoff editor, 345–352, 1978.

[5] Fredman, M.L. "Algorithms for computing evolutionary similarity measures with length independent gap penalties." *Bulletin of Mathematical Biology* 46:4, 553–566, 1984.

[6] Galil, Z. and R. Giancarlo. "Speeding up dynamic programming with applications to molecular biology." *Theoretical Computer Science* 64, 107–118, 1989.

[7] Gotoh, O. "An improved algorithm for matching biological sequences." *Journal of Molecular Biology* 162, 705–708, 1982.

[8] Gotoh, O. "Optimal alignment between groups of sequences and its application to multiple sequence alignment." *Computer Applications in the Biosciences* 9:3, 361–370, 1993.

[9] Gotoh, O. "Further improvement in methods of group-to-group sequence alignment with generalized profile operations." *Computer Applications in the Biosciences* 10:4, 379–387, 1994.

[10] Gribskov, M., A.D. McLachlan, and D. Eisenberg. "Profile analysis: detection of distantly related proteins." *Proceedings of the National Academy of Sciences USA* 84, 4355–4358, 1987.

[11] Gupta, S., J. Kececioglu and A. Schäffer. "Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment." *Journal of Computational Biology* 2:3, 459-472, 1995.

[12] Gusfield, D. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, New York, 1997.

[13] Henikoff, S. and J.G. Henikoff. "Amino acid substitution matrices from protein blocks." *Proceedings of the National Academy of Sciences USA* 89, 10915–10919, 1992.

[14] Hirschberg, D.S. "A linear space algorithm for computing longest common subsequences." *Communications of the ACM* 18, 341–343, 1975.

[15] Lipman, D.G., S.F. Altschul and J.D. Kececioglu. "A tool for multiple sequence alignment." *Proceedings of the National Academy of Sciences USA* 86, 4412-4415, 1989.

[16] Miller, W. and E.W. Myers. "Sequence comparison with concave weighting functions." *Bulletin of Mathematical Biology* 50, 97–120, 1988.

[17] Myers, E.W. and W. Miller. "Optimal alignments in linear space." *Computer Applications in the Biosciences* 4:1, 11–17, 1988.

[18] Myers, G., S. Selznick, Z. Zhang and W. Miller. "Progressive multiple alignment with constraints." Proceedings of the 1st ACM Conference on *Computational Molecular Biology*, 220–225, 1997.

[19] Sankoff, D. and J.B. Kruskal, editors. *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*. Addison-Wesley, Reading, MA, 1983.

[20] Setubal, J. and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Company, Boston, 1997.

[21] Taylor, E.W., A. Bhat, R. Nadimpalli, W. Zhang and J.D. Kececioglu. "HIV-1 encodes a sequence overlapping env gp41 with highly significant similarity to selenium dependent glutathione peroxidases." *Journal of Acquired Immune Deficiency Syndromes and Human Retrovirology* 15:5, 393–394, 1997.

[22] Wang, L. and T. Jiang. "On the complexity of multiple sequence alignment." *Journal of Computational Biology* 1:4, 337–348, 1994.

[23] Waterman, M.S. "Efficient sequence alignment algorithms." *Journal of Theoretical Biology* 108, 333–337, 1984.

[24] Waterman, M.S. *Introduction to Computational Biology: Maps, Sequences, and Genomes*. Chapman and Hall, London, 1995.

[25] Zhang, W., J.D. Kececioglu and E.W. Taylor. "Assessing distant homology between an aligned protein family and a proposed member through accurate sequence alignment." Technical Report 97-3, Department of Computer Science, The University of Georgia, October 1997. Submitted to *Journal of Molecular Biology*.

A Alignment vs. alignment

The recurrences for the alignment vs. alignment variation with optimistic or pessimistic gap counts are as follows. For $i > 0$ and $j > 0$,

$$
\begin{aligned}
H(i,j) &= \lambda\, k\, f_1^B(j) \\
&\quad + \min \begin{cases}
H(i,j-1) \;+\; \gamma \begin{cases} 0, & \text{opt;} \\ k\, f_{01}^B(j), & \text{pess.} \end{cases} \\
V(i,j-1) \;+\; \gamma \begin{cases} f_1^A(i)\, f_1^B(j), & \text{opt;} \\ k\, f_1^B(j), & \text{pess.} \end{cases} \\
D(i,j-1) \;+\; \gamma \begin{cases} f_1^A(i)\, f_1^B(j), & \text{opt;} \\ f_1^A(i)\, f_1^B(j) \;+\; f_0^A(i)\, f_{01}^B(j), & \text{pess.} \end{cases}
\end{cases} \qquad (42) \\
V(i,j) &= \lambda\, \ell\, f_1^A(i)
\end{aligned}
$$

$$+\ \min \begin{cases} H(i-1,j)\ +\ \gamma \begin{cases} f_1^A(i)\, f_1^B(j), & \text{opt;} \\ \ell\, f_1^A(i), & \text{pess.} \end{cases} \\ V(i-1,j)\ +\ \gamma \begin{cases} 0, & \text{opt;} \\ \ell\, f_{01}^A(i), & \text{pess.} \end{cases} \\ D(i-1,j)\ +\ \gamma \begin{cases} f_1^A(i)\, f_1^B(j), & \text{opt;} \\ f_1^A(i)\, f_1^B(j)\ +\ f_{01}^A(i)\, f_0^B(j), & \text{pess.} \end{cases} \end{cases} \tag{43}$$

$$D(i,j) \;=\; \sum_{a,b\,\in\,\Sigma\cup\{\epsilon\}} f_a^A(i)\, f_b^B(j)\ \delta(a,b)$$

$$+\ \min \begin{cases} H(i-1,j-1)\ +\ \gamma \begin{cases} f_1^A(i)\, f_{10}^B(j), & \text{opt;} \\ f_1^A(i)\, f_0^B(j)\ +\ f_0^A(i)\, f_{01}^B(j), & \text{pess.} \end{cases} \\ V(i-1,j-1)\ +\ \gamma \begin{cases} f_{10}^A(i)\, f_1^B(j), & \text{opt;} \\ f_0^A(i)\, f_1^B(j)\ +\ f_{01}^A(i)\, f_0^B(j), & \text{pess.} \end{cases} \\ D(i-1,j-1)\ +\ \gamma \begin{cases} f_1^A(i)\, f_{10}^B(j)\ +\ f_{10}^A(i)\, f_1^B(j), & \text{opt;} \\ f_1^A(i)\, f_{10}^B(j)\ +\ f_{10}^A(i)\, f_1^B(j) \\ \quad +\ f_{00}^A(i)\, f_{01}^B(j)\ +\ f_{01}^A(i)\, f_{00}^B(j), & \text{pess.} \end{cases} \end{cases} \tag{44}$$

For $i = 0$ and $j > 0$,

$$H(0,j) \;=\; H(0,j-1)\ +\ \lambda\, k\, f_1^B(j)\ +\ \gamma\, k\, f_{\perp 1}^B(j), \tag{45}$$

$$V(0,j) \;=\; H(0,j)\ +\ \gamma\, \ell\, f_{\perp 1}^A(1), \tag{46}$$

$$D(0,j) \;=\; V(0,j). \tag{47}$$

For $i > 0$ and $j = 0$,

$$V(i,0) \;=\; V(i-1,0)\ +\ \lambda\, \ell\, f_1^A(i)\ +\ \gamma\, \ell\, f_{\perp 1}^A(i), \tag{48}$$

$$H(i,0) \;=\; V(i,0)\ +\ \gamma\, k\, f_{\perp 1}^B(1), \tag{49}$$

$$D(i,0) \;=\; H(i,0). \tag{50}$$

For $i = 0$ and $j = 0$,

$$V(0,0) \;=\; 0, \tag{51}$$

$$H(0,0) \;=\; 0, \tag{52}$$

$$D(0,0) \;=\; 0. \tag{53}$$

B Sequence vs. profile

The recurrences for the sequence vs. profile variation with optimistic or pessimistic gap counts are as follows. For $i > 0$ and $j > 0$,

$$H(i,j) \;=\; \lambda\, f_1(j)\ +\ \min \begin{cases} H(i,j-1)\ +\ \gamma \begin{cases} 0, & \text{opt;} \\ \min\big\{ f_1(j),\ f_0(j-1) \big\}, & \text{pess.} \end{cases} \\ V(i,j-1)\ +\ \gamma\, f_1(j), \\ D(i,j-1)\ +\ \gamma\, f_1(j) \end{cases} \tag{54}$$

$$V(i,j) \;=\; \lambda\, k \;+\; \min \begin{cases} H(i-1,j) \;+\; \gamma \begin{cases} f_1(j), & \text{opt;} \\ k, & \text{pess.} \end{cases} \\ V(i-1,j), \\ D(i-1,j) \;+\; \gamma\, f_1(j) \end{cases} \tag{55}$$

$$\begin{aligned} D(i,j) \;=\; & \sum_{b \,\in\, \Sigma \cup \{\epsilon\}} f_b(j)\, \delta(a_i, b) \\ & +\; \min \begin{cases} H(i-1,j-1) \;+\; \gamma \begin{cases} \max\big\{ f_0(j) - f_0(j-1),\; 0 \big\}, & \text{opt;} \\ f_0(j), & \text{pess.} \end{cases} \\ V(i-1,j-1), \\ D(i-1,j-1) \;+\; \gamma \begin{cases} \max\big\{ f_0(j) - f_0(j-1),\; 0 \big\}, & \text{opt;} \\ \min\big\{ f_0(j),\; f_1(j-1) \big\}, & \text{pess.} \end{cases} \end{cases} \end{aligned} \tag{56}$$

The boundary conditions are the same as for the sequence vs. alignment variation, except we now define

$$f_{\perp 1}(j) \;:=\; F_{\perp 1}(j) \;-\; F_{\perp 1}(j-1)$$

for $1 \le j \le n$, where

$$F_{\perp 1}(j) \;=\; \begin{cases} \left\{ \begin{array}{ll} \max\big\{ F_{\perp 1}(j-1),\; f_1(j) \big\}, & \text{opt;} \\ \min\big\{ F_{\perp 1}(j-1) + f_1(j),\; k \big\}, & \text{pess.} \end{array} \right\}, & j > 0; \\ 0, & j = 0. \end{cases} \tag{57}$$

and we also define $f_0(0) := 0$ and $f_1(0) := k$.

C Profile vs. profile

The recurrences for the profile vs. profile variation with optimistic or pessimistic gap counts are as follows, where for $S \in \{A, B\}$ and $x, y \in \{0, 1\}$, we define

$$\begin{aligned} g^S_{xy}(i) \;&:=\; \min\big\{ f^S_y(i),\; f^S_x(i-1) \big\}, \\ h^S_{xy}(i) \;&:=\; \max\big\{ f^S_y(i) - f^S_x(i-1),\; 0 \big\}. \end{aligned}$$

For $i > 0$ and $j > 0$,

$$H(i,j) \;=\; \lambda\, k\, f^B_1(j) \;+\; \min \begin{cases} H(i,j-1) \;+\; \gamma \begin{cases} 0, & \text{opt;} \\ k\, g^B_{01}(j), & \text{pess.} \end{cases} \\ V(i,j-1) \;+\; \gamma \begin{cases} f^A_1(i)\, f^B_1(j), & \text{opt;} \\ k\, f^B_1(j), & \text{pess.} \end{cases} \\ D(i,j-1) \;+\; \gamma \begin{cases} f^A_1(i)\, f^B_1(j), & \text{opt;} \\ k\, g^B_{01}(j) + f^A_1(i)\, h^B_{01}(j), & \text{pess.} \end{cases} \end{cases} \tag{58}$$

$$
V(i,j) \;=\; \lambda\,\ell\, f_1^A(i) \;+\; \min \begin{cases} H(i,j-1) \;+\; \gamma \begin{cases} f_1^A(i)\, f_1^B(j), & \text{opt;} \\ \ell\, f_1^A(i), & \text{pess.} \end{cases} \\[2ex] V(i,j-1) \;+\; \gamma \begin{cases} 0, & \text{opt;} \\ \ell\, g_{01}^A(i), & \text{pess.} \end{cases} \\[2ex] D(i,j-1) \;+\; \gamma \begin{cases} f_1^A(i)\, f_1^B(j), & \text{opt;} \\ \ell\, g_{01}^A(i) + f_1^B(j)\, h_{01}^A(i), & \text{pess.} \end{cases} \end{cases} \tag{59}
$$

$$
\begin{aligned}
D(i,j) \;=\; & \sum_{a,b\,\in\,\Sigma\cup\{\epsilon\}} f_a^A(i)\, f_b^B(j)\, \delta(a,b) \\
& + \min \begin{cases} H(i-1,j-1) \;+\; \gamma \begin{cases} f_1^A(i)\, h_{00}^B(j), & \text{opt;} \\ f_1^A(i)\, f_0^B(j) \;+\; f_0^A(i)\, g_{01}^B(j), & \text{pess.} \end{cases} \\[2ex] V(i-1,j-1) \;+\; \gamma \begin{cases} f_1^B(j)\, h_{00}^A(i), & \text{opt;} \\ f_1^B(j)\, f_0^A(i) \;+\; f_0^B(j)\, g_{01}^A(i), & \text{pess.} \end{cases} \\[2ex] D(i-1,j-1) \;+\; \gamma \begin{cases} g_{11}^A(i)\, h_{00}^B(j) \;+\; h_{00}^A(i)\, g_{11}^B(j) \\ \quad +\, h_{00}^A(i)\, h_{11}^B(j) \;+\; h_{11}^A(i)\, h_{00}^B(j), & \text{opt;} \\ h_{01}^A(i)\, g_{10}^B(j) \;+\; g_{10}^A(i)\, h_{01}^B(j) \\ \quad +\, g_{10}^A(i)\, g_{01}^B(j) \;+\; g_{01}^A(i)\, g_{10}^B(j) \\ \quad +\, g_{01}^A(i)\, h_{10}^B(j) \;+\; h_{10}^A(i)\, g_{01}^B(j), & \text{pess.} \end{cases} \end{cases}
\end{aligned} \tag{60}
$$

The boundary conditions for the recurrences are the same as for the alignment vs. alignment variation with optimistic or pessimistic gap counts, except $f_{\perp 1}^A(i)$ and $f_{\perp 1}^B(j)$ are now defined for the two profiles A and B as in the sequence vs. profile variation.