Universität Bielefeld
Forschungsschwerpunkt Mathematisierung —
Strukturbildungsprozesse

Materialien/Preprints 111

On Simultaneous versus Iterative Multiple Sequence Alignment

S.W. Perrey, J. Stoye, V. Moulton, A.W.M. Dress

Bielefeld 1997

Classification code(s) according to the
1980 Mathematics Subject Classification of
Mathematical Reviews:

92-04
92A10
68C05

Forschungsschwerpunkt Mathematisierung —
Strukturbildungsprozesse
Universität Bielefeld
Postfach 10 01 31
D-33501 Bielefeld

Telefon: (05 21) 1 06-47 64
Fax: (05 21) 1 06-60 07

e-mail:
fsp@mathematik.uni-bielefeld.de

On Simultaneous versus Iterative Multiple Sequence Alignment

S.W. Perrey, J. Stoye, V. Moulton, A.W.M. Dress

July 7, 1997

Abstract

One of the main problems in computational biology is the construction of biologically plausible alignments for given sequence families. Many procedures have been developed for this purpose; most are based on clustering the sequences hierarchically prior to the computation of their alignment and then proceeding recursively.

In this paper, we compare results obtained with a new algorithm for *simultaneous* multiple sequence alignment, the so-called *Divide & Conquer Alignment* procedure (DCA), with results obtained with *iterative* procedures. DCA allows one to simultaneously align relatively large families of sequences using computationally efficient heuristics which – relative to the sum-of-pairs scoring function which DCA accepts as the standard of truth – give rise to good (though not necessarily optimal) alignments.

As expected, the simultaneous alignments often prove superior in picking up biologically important signals contained in a family of highly diverged sequences, which are sometimes overlooked by successive pairwise alignment.

1 Introduction

Multiple sequence alignment is a well-studied but still not satisfactorily solved problem in string processing, having its most important application in computational molecular biology. Indeed, many important conclusions to be drawn from the sequence of *residues* in a big biomolecule (amino acids in a given protein or nucleotides in a given RNA or DNA molecule) depend crucially on *comparing* that sequence with other such sequences by means of appropriately constructed *alignments*. They are used to detect homologues among sequences in genome databases, to study phylogenetic relationships, or to identify structurally and/or functionally important parts of the molecule in question.

Consequently, establishing fast and reliable tools for multiple sequence alignment is one of the most fundamental tasks in present day computational biology, enjoying an abundance of publications and software contributions (see [34, 8, 59]).

The overall strategy one has to follow for producing reliable alignments is quite obvious: by inserting *gaps* here and there into the sequences one wants to align, one tries to come up with sequences of equal length so that the sequence entries at each site – that is, in each column when the (aligned) sequences themselves are spelled out horizontally, one below the other – exhibit a biologically meaningful diversity, possibly of not too large a degree, which can be interpreted in a coherent way. For example, as mentioned above, one may head for a phylogenetic interpretation implying that the sequence entries at a given site have evolved from a common ancestor entry, or for a structural interpretation implying that the aligned residues are placed at similar locations within the folded molecule.

Consequently, because it is the *similarity* of sequence patterns which is supposed to signal phylogenetic and/or structural kinship between the sequences, the aim of sequence-alignment procedures is to *maximize* overall similarity. Thus, all that is required is

- specifying in a biologically suitable and simultaneously quantifiable way the term *overall similarity*, and
- constructing algorithms for producing alignments which maximize – or, if this turns out to be too time consuming, at least exhibit a rather high degree of – that overall similarity.

While the first task needs input from biology as well as from mathematical modeling, the second task is a purely mathematical one. Unfortunately, many ideas relating to the first task cannot be tested, and important structural parameters suggested by these ideas cannot be evaluated easily unless the second task has been dealt with appropriately.

To tackle that second task, the starting point is clearly to find good methods for aligning *two* sequences – that is, for *pairwise* alignment –, and algorithms for solving this problem were already successfully developed a quarter of a century ago [36, 56]. These algorithms follow the well-known *dynamic programming* method. However, their natural and straightforward generalizations for aligning three or more sequences *simultaneously* (together with the natural extension of quantifying overall similarity in terms of the so-called *sum-of-pairs score* [3]) quickly run into prohibitive memory and time constraints as number or length of the sequences in question increase. Therefore, almost all techniques for aligning larger sets of sequences are based on first calculating a proper order – or, more precisely, a binary hierarchy – for the sequences to be aligned, and then constructing a multiple alignment in a "bottom-up" or "hill-climbing" manner by

performing a series of pairwise alignments (using, if necessary, appropriate adaptations of the standard algorithm aligning appropriately defined "profiles" of sequences rather than sequences) according to that precalculated hierarchy (see for example [10],[31] for reviews).

However, these methods (e.g. DFALIGN [14], GENALIGN [30], TREEALIGN [20], MULTAL [50], PILEUP [16], CLUSTAL W [51], MALIGN [60]) – though fast – can be used with some reservation only as they rarely allow one to reconsider an alignment of a subfamily in the light of information coming from sequences outside that subfamily.

On the other hand, significant progress with implementing dynamic programming procedures for simultaneous multiple alignment was made by H. Carrillo and D.J. Lipman [7] in the late eighties: it became possible to align simultaneously and optimally between six and eight protein sequences (of medium length and comparatively high pairwise similarity) in some minutes. This was achieved by a branch-and-bound approach, cutting down the (high-dimensional) search space used in standard versions of dynamic programming by considering projections of precalculated heuristic alignments onto the (two-dimensional) "boundaries" of that search space.

Yet, even when implementing the proposals by H. Carrillo and D.J. Lipman using highly sophisticated implementation techniques, the resulting program, called `MSA`, often requires more time and/or memory space than available whenever it has to deal with larger data sets (regarding the number as well as the length of the sequences) [19].

However, there exists an algorithm which is based on `MSA` and yet is significantly faster, providing very good, if not optimal alignments for comparatively large data sets (even those including long sequences), called the *Divide & Conquer alignment* procedure (DCA). The performance of this algorithm has been discussed in some detail in [53, 49, 47].

In this paper, DCA is used to compare optimal (or, at least close-to-optimal) sum-of-pairs-score alignments of some protein and some RNA sequence families with results produced by various iterative alignment procedures, highlighting advantages offered by the former.

The contents of this paper are as follows: First, we discuss, from a general point of view, the differences between simultaneous and iterative multiple sequence alignment methods. Next, we briefly recall the description of the problem in question in formal, purely mathematical terms and summarize the workings of the DCA algorithm. And then, we discuss in detail some biological data sets: First, we give an alignment of nine serine protease sequences which significantly extends results obtained in [29]. Next, we discuss a set of eight RNase MRP RNA sequences, and we briefly interpret the results phylogenetically, using the program SplitsTree [13], an independent program developed for phylogenetic analysis. And finally, we report and briefly discuss the results of a recent evaluation of five multiple sequence alignment procedures, including DCA, using a set of ten rRNA sequences [23].

2 Some Differences between Simultaneous and Iterative Multiple Sequence Alignment Methods

Iterative methods are very fast, and they can align, in principle, any number of sequences. Over the years, they have successfully incorporated growing biological knowledge regarding e.g. the use of appropriate substitution matrices, region-dependent gap-

penalties and weighting schemes (see for instance [51]). The resulting alignments are quite acceptable for families of moderately diverged sequences. Yet, when it comes to align a family of highly diverged sequences, they easily run into local, but not necessarily global optima, – a risk which is of course inherent in any hill-climbing bottom-up method.

In addition, while a simultaneous alignment procedure tries to optimise some well-defined score function for multiple sequence alignment [3], iterative methods often do not even accept such a function (which might be tested for its biological significance and improved) as a standard of truth.

A further problem of iterative alignment procedures is that their results depend crucially on the precalculated order in which the sequences are processed[1], while simultaneous alignment procedures take all sequences to be aligned into account simultaneously and therefore do not depend on any precalculated grouping of the sequences in question.

Most importantly, iterative methods perform a series of pairwise alignments whereby pairs of more closely related sequences – or pairs of already aligned subfamilies of sequences – are considered first. This can go wrong for the following reasons:

1. There is often *more than one optimal* alignment for any given pair of sequences – or pair of already aligned subfamilies – so that an *arbitrary* decision has to be made for choosing one of those for further consideration (at least, unless additional effort is spent to find out which of those might be biologically the most plausible one – or to keep track of all of them).

2. Optimal alignments are often highly sensitive with respect to parameter changes, in particular, when more distantly related sequences (or subfamilies) are to be aligned. Consequently, the set of optimal pairwise alignments one has to choose from may depend strongly on the parameters used in the calculation.

3. Once all members of some closely related subfamily have been considered, the alignment of this subfamily is *locked* so that any more distantly related sequences can never have an influence on this sub-alignment. Yet, it was noticed that reopening such a locked alignment for performing pairwise alignments of two subfamilies (e.g. [50]) or simultaneous three-way alignments [55] often improves the overall alignment. However, even these procedures are still based on locking some sub-alignments.

Instead, *simultaneous* alignment procedures offer the following advantages:

1. Because such procedures align a family of sequences in *one* step, the "multiple optima problem" does not present itself at all in the construction process. Of course, there can also be more than one "final" optimal alignment of the whole sequence family. However, at this final stage, that might actually present quite interesting and valuable information, e.g. for figuring out "uncertain alignment sites".

2. More often than not, a simultaneous alignment procedure should be more robust against parameter changes than procedures based on pairwise alignments, particularly for data sets consisting of a comparatively large number of sequences.

[1] For example, CLUSTAL V [24] and PILEUP [16] use an UPGMA tree (derived from pairwise alignment scores transformed into distances) for guiding the successive alignment stages while CLUSTAL W uses a Neighbour-Joining tree and allows the user to specify another tree if he doesn't agree with that tree, or just wants to check out an alternative tree [51].

3. Already D.J. LIPMAN *et al.* observed that the quality of a simultaneous alignment of a sequence family (quality in terms of the number of alignment sites which agree with an alignment based on the structure of the molecules involved) generally increases with the number of members in that family [29]. And they observed that that quality can even be increased by including one or two "outgroup sequences" – that is, more distantly related sequences – in a simultaneous alignment procedure, a fact which almost by definition cannot hold for iterative procedures.

A further problem in biological sequence alignment is that given any two sequences, their biologically most plausible pairwise alignment often is not optimal but just *suboptimal*, and sometimes remains suboptimal even after trying carefully and specifically to optimise the employed set of alignment parameters with regard to the sequences in question. The reason for this is that the mutation process in different areas of a biomolecule can hardly be modeled by one and the same model, and in particular not by a model which presupposes the validity of some kind of a general stochastic Fermat Principle for such a process. For instance, it is well-known that the residues in biomolecular sequences are often correlated and do *not* evolve by some purely stochastic mutation process, independently of the remaining residues (see for instance [32]). Yet, because these correlations are very difficult to model correctly and, provided even that that could be achieved, are almost impossible to handle computationally, almost all alignment procedures (including `MSA` and DCA, so far) employ a single substitution cost matrix across the entire sequence.

And even more problems arise when it comes to model appropriately the probabilities for the occurrence of insertions and deletions.

Yet, it is virtually impossible to handle all possibly relevant suboptimal alignments in a computationally acceptable way because, unfortunately, the number of alignments to be taken into consideration generally grows enormously upon even the slightest relaxation of the optimality criterion [58]. Therefore, iterative methods can go wrong in

- always choosing the optimal alignment for the most closely related pairs of sequences, as well as
- choosing the optimal alignment of already aligned subfamilies, – thereby inducing, of course, some sort of suboptimal alignment between members of different subfamilies which in turn strongly depend on the alignment parameters and the precalculated hierarchy.

Instead, a simultaneous alignment procedure forces *every* pair of sequences into a certain suboptimal pairwise alignment, depending on *all* other sequences of the family to be aligned, because it attempts to optimise an overall score function. As it happens, these suboptimal pairwise alignments often are the biologically (more) plausible ones, because they are calculated relative to the other sequences of the family under consideration (see the example below).

It was also noticed that phylogenetic trees calculated from sequence alignments produced by iterative methods had a bias towards the tree that these methods used just for processing the sequences [28, 52]. In contrast, an alignment procedure which takes all the sequences to be aligned into account *simultaneously*, produces alignments for subsequent phylogenetic tree reconstruction which are not biased by a previously chosen "guide tree".

A simple Example:

As a very simple biological example, we consider a short subsequence of ϵ-globin noncoding DNA[2] of human ($\mathbf{s}_1$), chimpanzee ($\mathbf{s}_2$) and orangutan ($\mathbf{s}_3$) in order to highlight the difference between iterative and simultaneous sequence alignment methods.

As a first step, any iterative sequence alignment procedure calculates a proper order for the series of pairwise alignments which in this case is $(\,(\mathbf{s}_1, \mathbf{s}_2),\, \mathbf{s}_3)$ because the human and chimpanzee sequences are more closely related to one another than either of them is to the orangutan sequence. One rather reasonably looking alignment of $S = \{\mathbf{s}_1, \mathbf{s}_2\}$ with $\mathbf{s}_1 = \mathtt{GGAAGG}$ and $\mathbf{s}_2 = \mathtt{GGGAGG}$ is given by the matrix

$$M_1 := \begin{pmatrix} \mathtt{G\,G\,A\,A\,G\,G} \\ \mathtt{G\,G\,G\,A\,G\,G} \end{pmatrix}.$$

There are two other alignments which might also be (biologically) plausible. These contain some gaps and thereby avoid the substitution $\mathtt{A} \leftrightarrow \mathtt{G}$:

$$M_2 := \begin{pmatrix} \mathtt{G\,G\,A\,-\,A\,G\,G} \\ \mathtt{G\,G\,-\,G\,A\,G\,G} \end{pmatrix}, \qquad M_2' := \begin{pmatrix} \mathtt{G\,G\,-\,A\,A\,G\,G} \\ \mathtt{G\,G\,G\,-\,A\,G\,G} \end{pmatrix}.$$

Any reasonable parameter choice for pairwise alignment would result either in M_1 as the optimal alignment or in M_2 *and* M_2' as optimal alignments (as it would be hard to prefer one of the latter two alignments as being more biologically plausible than the other). Iterative methods have to make a somehow arbitrary decision unless they have decided for the M_1 alignment (which they most probably do since the penalty for introducing two gaps is in general considerably higher than that for accepting just one transition). In this case, M_2 and M_2' would be regarded as suboptimal or near-to-optimal alignments.

Considering the gorilla sequence of this part of the DNA doesn't help because, here, it is identical to the chimpanzee sequence. But including also the sequence $\mathbf{s}_3 = \mathtt{GGAGAGG}$ of the more distantly related orangutan clarifies the situation: iterative procedures would produce one of the alignments

$$M_{11} := \begin{pmatrix} \mathtt{G\,G\,A\,-\,A\,G\,G} \\ \mathtt{G\,G\,G\,-\,A\,G\,G} \\ \mathtt{G\,G\,A\,G\,A\,G\,G} \end{pmatrix}, M_{12} := \begin{pmatrix} \mathtt{G\,G\,-\,A\,A\,G\,G} \\ \mathtt{G\,G\,-\,G\,A\,G\,G} \\ \mathtt{G\,G\,A\,G\,A\,G\,G} \end{pmatrix},$$

unless they reopen the alignment for aligning e.g. some "profile" of the human and orangutan sequence with the chimpanzee sequence.

A simultaneous alignment procedure would instead produce (for appropriate parameter choices) the alignment

$$M_{21} := \begin{pmatrix} \mathtt{G\,G\,A\,-\,A\,G\,G} \\ \mathtt{G\,G\,-\,G\,A\,G\,G} \\ \mathtt{G\,G\,A\,G\,A\,G\,G} \end{pmatrix},$$

which now appears to be at least as plausible biologically as any one of the alignments M_{11} or M_{12} – implying that the last common ancestor of man, chimpanzee and gorilla as well as that of man and chimpanzee still carried the GGAGAGG sequence and that the loss of one of the residues at sites 3 and 4 precisely occurred independently and more or less as a chance event in all three lineages after speciation. This hypothesis could further be confirmed by some additional outgroup sequence of the form $\mathbf{s} = \mathtt{GG} * \mathtt{A} * \mathtt{G} * \mathtt{AGG}$ (where "$*$" stands for arbitrary, short – hopefully even empty – insertions) (see [38]).

[2]We have taken the sequences from the EMBL database: gde21301, alignment positions 973-978. The alignment-positions 720 up to 972 and 979-1100 are without any gaps and contain only 19 non-constant positions which are scattered across the sequences.

3 The Sum-of-Pairs Score for Multiple Sequence Alignment

In this section, we define multiple alignments formally and we describe the basic principles of evaluating quantitatively the quality of a given multiple alignment (for further reference see [10],[41], and [59]).

Suppose that we are given a family $S = (\mathbf{s}_1, \ldots, \mathbf{s}_k)$ of k sequences:

$$\begin{aligned} \mathbf{s}_1 &= s_{11}\, s_{12} \ldots s_{1n_1} \\ &\vdots \\ \mathbf{s}_k &= s_{k1}\, s_{k2} \ldots s_{kn_k} \end{aligned}$$

of various lengths n_1 to n_k, where each sequence entry s_{ij} represents a letter from a given finite alphabet $\mathcal{A}$. An *alignment* of the sequences S is a matrix $M = (m_{ij})_{1 \le i \le k, 1 \le j \le N}$ where

- $m_{ij} \in \mathcal{A} \cup \{-\}$, with '$-$' denoting the *gap letter* supposed not to be contained in $\mathcal{A}$,
- the rows $\mathbf{m}_l := m_{l1} \ldots m_{lN}$ of M considered as sequences of symbols from $\mathcal{A} \cup \{-\}$, reproduce the sequences $\mathbf{s}_l$ upon elimination of the gap letters $(1 \le l \le k)$,
- the matrix M has no column, only containing gaps.

We denote the set of all alignments of S by M_S (see also [33] for a more general definition of *alignment*).

Next, assume that for each pair of rows $\mathbf{m}_p$ and $\mathbf{m}_q$ in an alignment, a (non-negative) score function $w_{p,q}(\mathbf{m}_p, \mathbf{m}_q)$ has been defined which measures the quality of the alignment of $\mathbf{s}_p$ and $\mathbf{s}_q$ defined by these two rows (upon elimination of common gaps), and denote by $w_{opt}(\mathbf{s}_p, \mathbf{s}_q)$ the minimum of $w_{p,q}(\mathbf{m}_p, \mathbf{m}_q)$, taken over all alignments $M \in M_S$ which we suppose to vanish if and only if $\mathbf{s}_p$ coincides with $\mathbf{s}_q$ (see [59] for a thorough discussion of score functions for pairwise alignment and note that, as an alternative to minimising a *dissimilarity* score, one may also aim to maximise an appropriately defined *similarity* score).

The *weighted sum-of-pairs score* [3] for an alignment $M \in M_S$ relative to a given family of (generally non-negative) weight parameters $\alpha_{p,q}\, (1 \le p < q \le k)$ is now defined by

$$w(M) := \sum_{1 \le p < q \le k} \alpha_{p,q} \cdot w_{p,q}(\mathbf{m}_p, \mathbf{m}_q).$$

The *multiple alignment problem* then is to find matrices $M \in M_S$ whose weighted sum of pairs score $w(M)$ is small.

The logic for introducing the weight parameters $\alpha_{p,q}$ (from which procedures for choosing them appropriately are to be deduced) is the following: In general, any set of related biological sequences contains some sequences which are more closely related to one another than to the remaining ones, and highlighting their similarity, at least to some extent, might often be more important than forcing them to independently conform to the patterns of the other sequences. On the other hand, as almost any sample of sequences is biased in one way or the other (even, most probably, the sample provided by Nature itself), a perhaps over-represented subset of highly homologous

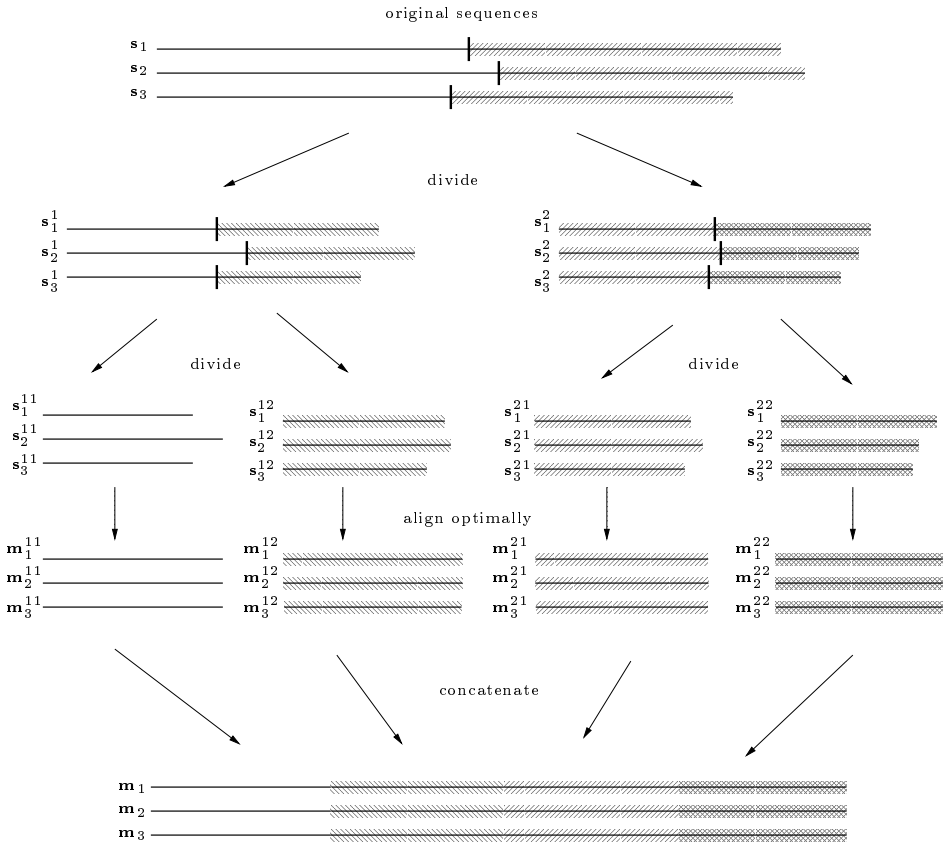

Figure 1: Schematic representation of the divide and conquer method.

sequences in a data set should not be allowed through its sheer size to force all the other sequences to conform to its patterns. Both goals, highlighting similarity between closely related sequences and discounting over-representation of certain subclasses of sequences can (hopefully) be achieved by choosing appropriate weight factors.

4 The Divide &Conquer Alignment

As mentioned above, $w(M)$ can be optimised in principle by straightforward *dynamic programming*, provided $w_{opt}(\mathbf{s}_p, \mathbf{s}_q)$ can be computed in this way [36, 45]. However, this is possible only in theory at present: in practice, the space and time requirements for dynamic programming, even in its most sophisticated forms, make it virtually impossible to deal with, say, five not *highly* homologous sequences of length approximately 1000. However, such tasks present themselves often in biological sequence analysis. It is here, therefore, where we have to invoke DCA.

The general idea of DCA is rather simple (cf. Fig. 1): Each sequence is cut in two by cutting it just behind a suitable *slicing site* somewhere close to its midpoint. This way, the problem of aligning one family of (long) sequences is divided into the two problems of aligning two families of (shorter) sequences, the prefix and the suffix sequences. This procedure is re-iterated until the sequences are sufficiently short – say, shorter than a pregiven stop size L – so that they can be aligned optimally by `MSA`. Finally, the resulting short alignments are concatenated, yielding a multiple alignment of the original sequences.

Of course, the main difficulty with this approach is how to identify those slicing-site

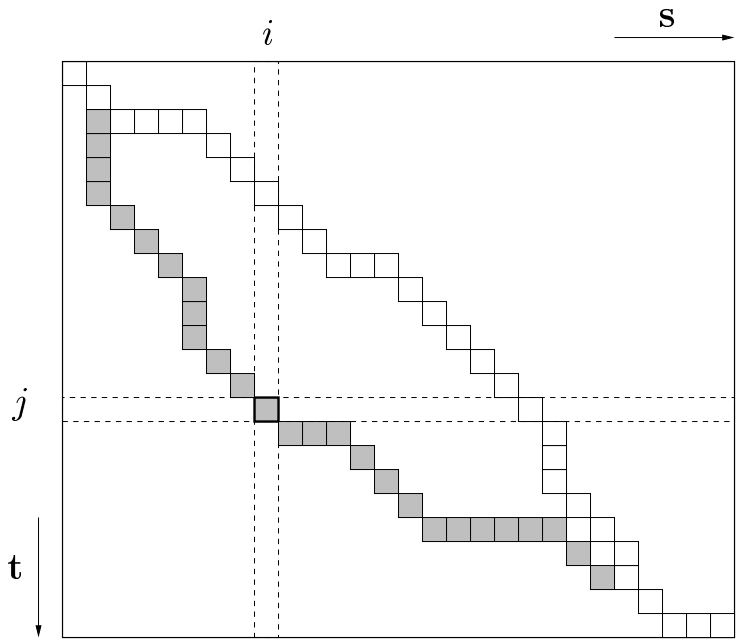

Figure 2: The definition of secondary charges: White boxes present an optimal alignment M of the sequences $\mathbf{s}$ and $\mathbf{t}$, shaded boxes present the concatenation of an optimal alignment M_1 of the two prefix and an optimal alignment M_2 of the two suffix sequences defined by the slicing sites (i, j). $C_{\mathbf{s},\mathbf{t}}$ is then defined by $w(M_1) + w(M_2) - w(M)$.

combinations which lead to an optimal or – at least – close to optimal concatenated alignment. Here, a heuristic based on so-called *secondary-charge matrices* which are used for quantifying the compatibility of slicing sites in distinct sequences proved to be successful:

More precisely, given a sequence $\mathbf{s} = s_1 s_2 \dots s_n$ of length n and a slicing site c ($0 \leq c \leq n$), we denote by $\mathbf{s}(\leq c)$ the prefix sequence $s_1 s_2 \dots s_c$ and by $\mathbf{s}(> c)$ the suffix sequence $s_{c+1} s_{c+2} \dots s_n$, and we use the dynamic programming procedure to compute, for all pairs of sequences $(\mathbf{s}_p, \mathbf{s}_q)$ and for all slicing sites c_p of $\mathbf{s}_p$ and c_q of $\mathbf{s}_q$, the *secondary charge* $C_{\mathbf{s}_p,\mathbf{s}_q}[c_p, c_q]$ defined by[3]

$$C_{\mathbf{s}_p,\mathbf{s}_q}[c_p, c_q] := w_{opt}(\mathbf{s}_p(\leq c_p), \mathbf{s}_q(\leq c_q)) + w_{opt}(\mathbf{s}_p(> c_p), \mathbf{s}_q(> c_q)) - w_{opt}(\mathbf{s}_p, \mathbf{s}_q)$$

which quantifies the additional charge imposed by forcing the alignment of $\mathbf{s}_p$ and $\mathbf{s}_q$ to optimally align the two prefix sequences $\mathbf{s}_p(\leq c_p)$ and $\mathbf{s}_q(\leq c_q)$ as well as the two suffix sequences $\mathbf{s}_p(> c_p)$ and $\mathbf{s}_q(> c_q)$, rather than aligning $\mathbf{s}_p$ and $\mathbf{s}_q$ optimally (cf. Fig. 2).

The calculation of the matrices $C_{\mathbf{s}_p,\mathbf{s}_q}$ can be performed by computing *forward* and *reverse* matrices in a similar way as described in [25, 35, 54]. Note that there exists, for every fixed slicing site $\hat{c}_p$ of $\mathbf{s}_p$, at least one slicing site $c_q(\hat{c}_p)$ of $\mathbf{s}_q$ with $C_{\mathbf{s}_p,\mathbf{s}_q}[\hat{c}_p, c_q(\hat{c}_p)] = 0$ which can be computed easily from any optimal *pairwise* alignment of $\mathbf{s}_p$ and $\mathbf{s}_q$. The problem *multiple* alignments have to face, is that given a slicing site $\hat{c}_r$ of $\mathbf{s}_r$, the optimal slicing site $c_q\,(c_p(\hat{c}_r))$ of $\mathbf{s}_q$ relative to the slicing site $c_p(\hat{c}_r)$ of $\mathbf{s}_p$ might not coincide with the optimal slicing site $c_q\,(\hat{c}_r)$ of $\mathbf{s}_q$ relative to the slicing site $\hat{c}_r$ of $\mathbf{s}_r$. In other words, pairwise optimal alignments may be incompatible with one another - much in analogy to *frustrated systems* considered in statistical physics.

To search for good k-tuples of slicing sites, we therefore define the *multiple additional charge* $C(c_1, \dots, c_k)$ imposed by slicing the sequences at any given k-tuple of slicing sites

[3] or just the negative of this if one deals with similarity rather than with dissimilarity scores.

$(c_1, \ldots, c_k)$ as a weighted sum of secondary charges over all projections (c_p, c_q), that is, we put

$$C(c_1, c_2, \ldots, c_k) \; := \sum_{1 \le p < q \le k} \alpha_{p,q} \cdot C_{\mathbf{s}_p, \mathbf{s}_q}[c_p, c_q],$$

where the $\alpha_{p,q}$ are the same sequence-dependent weight factors as above.

Our proposition is now that using those k-tuples as the preferred slicing-site combinations that minimise – for a given fixed slicing site $\hat{c}_p$ of, say, the longest sequence $\mathbf{s}_p$ which we choose somewhere in the middle of that sequence – the value $C(c_1, \ldots, c_{p-1}, \hat{c}_p, c_{p+1}, \ldots, c_k)$ over all slicing sites $c_1, \ldots, c_{p-1}, c_{p+1}, \ldots, c_k$ of $\mathbf{s}_1, \ldots, \mathbf{s}_{p-1}, \mathbf{s}_{p+1}, \ldots, \mathbf{s}_k$, respectively, will result in very good, if not optimal multiple alignments because, in this way, the mutual *frustration* is distributed as fairly as possible.

In conclusion, reiterating this process until all sequences in any given subsequence family have a length not exceeding the stop size L, DCA simply performs the following general procedure:

Algorithm *DCA* $(\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k; L)$

`If` $\min_{i \in \{1,2,\ldots,k\}}\{n_i\} \le L$
`then` return an optimal alignment of $\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k$ (using e.g. `MSA`);
`else` return the concatenation of
$$DCA(\mathbf{s}_1(\le c_1), \mathbf{s}_2(\le c_2), \ldots, \mathbf{s}_k(\le c_k); L)$$
and $DCA(\mathbf{s}_1(> c_1), \mathbf{s}_2(> c_2), \ldots, \mathbf{s}_k(> c_k); L)$;
`where` $(c_1, c_2, \ldots, c_k) := \mathit{calc\text{-}cut}(\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_k)$,

where the subroutine *calc-cut* computes a *C-optimal* k-tuple of slicing sites (see [49, 39, 6] for details regarding suitable branch-and-bound approaches towards that optimization problem).

5 The Present and the Future State of the Implementation of DCA

The program DCA is a C-implementation of the divide and conquer alignment method, written in ANSI C, so that it will run on any computer with an ANSI C compiler available. Current details on the availability of DCA can be found on the DCA World-Wide Web Homepage [48].

The optimal alignment calculations required by the divide and conquer method are performed by the program `MSA` (version 2.1) [19, 29, 42] which also has to be installed on the same computer, and has to be executable in the same directory as DCA.

In general terms, DCA takes as input a family of sequences and a matrix of pairwise letter distances (e.g. PAM [9], Blosum [21]) including affine gap penalties, and it then produces a multiple alignment of that sequence family. The input format for the sequences and the score parameters are the same as those for `MSA` [4] [42]. It is possible to penalise gaps at the beginning and the end of an alignment either just like internal gaps, or not to penalise terminal gaps at all, allowing for what people call *free shift*.

[4] Unfortunately, the present implementation of `MSA` puts the gap penalty parameters and the substitution matrix into a single file. So, in the present format, they cannot easily be varied independently of each other.

Several other parameters can be chosen for DCA: stop size L, window size W, and weight intensity λ (W and *lambda* are described below).

As explained in Section 4, the stop size L of the recursion gives an upper bound for the size of the sequences to be aligned optimally by `MSA`. Thus, too large an L (e.g. $L > 50$) can result in very long running times and very large memory usage due to the resulting `MSA` runs. On the other hand, too small an L (e.g. $L < 5$) can result in empty subsequences which easily lead to unsuitable alignments.

We recommend choosing L between 16 and 30. More precisely, for larger sequence families or for families of closely related sequences a small L results in a short running time as well as in quite acceptable alignments, whereas for a small number of sequences or for sequences which are highly diverged a larger L is advisable to get high quality alignments.

In case one wishes to check and perhaps correct the alignment in the proximity of slicing sites, a *windowing approach* [49] has also been implemented: Inside a window of size W, placed across each slicing site, the alignment can be re-aligned optimally.

In the present implementation, the weighting factors $\alpha_{p,q}$ for the cost function C described in Section 4, are computed using the following formula:

$$\alpha_{p,q} \;:=\; 1 - \lambda \cdot \left(\frac{w_{opt}(s_p, s_q) - \min_{1 \le i < j \le k} \{ w_{opt}(\mathbf{s}_i, \mathbf{s}_j) \}}{\max_{1 \le i < j \le k} \{ w_{opt}(\mathbf{s}_i, \mathbf{s}_j) \}} \right)$$

where λ is defined to be the *weight intensity*. That intensity may be set to any value between $\lambda = 1$ (maximum weighting) and $\lambda = 0$ (no weighting: $\alpha_{p,q} = 1$ for all $1 \le p < q \le k$).

For more than ten or twelve highly diverged sequences, the calculation of the first slicing-site combinations may take a rather long time. So, in that case, a perhaps slightly less accurate alignment can be obtained by using the option *Off* at the heading "C-optimal slicing sites:" (the default is *On*). In this case, the sub-routine *calc-cut* which minimizes $C(\hat{c}_1, c_2, \ldots, c_k)$ exactly, is replaced by an appropriate and fast search for C-suboptimal k-tuples $(\hat{c}_1, c_2, \ldots, c_k)$ using standard (heuristic) combinatorial optimization techniques [40]. This allows one to obtain still rather reasonable looking alignments with time and memory space requirements of the order of $\mathcal{O}(k^2 n^2 + \frac{n}{L} L^k)$, where k is the number of sequences and n is the length of the longest sequence, making the procedure for small L comparable – with respect to time and space requirements – to iterative alignment procedures, the advantage still being that the sequences are being aligned simultaneously.

In the near future, following ideas proposed in [6] and in [51], preprocessing procedures will be implemented in order to speed up the *calc-cut* subroutine and to apply more sophisticated weighting schemes, region-dependent gap-penalties and pair-dependent substitution matrices.

6 Examples

In the current implementation of DCA, the parameters used for the (affine) gap penalty function and substitution matrix are the same for all pairwise alignments (and also in the `MSA`-subroutine). In addition, we employ the *unweighted* sum-of-pairs score, i.e. the weight intensity λ is put equal to 0. However, even with these simplifications, the algorithm produces high quality alignments, in particular if a reasonably large number of biological sequences is considered (a fact which is well in accordance with the

Proteins	Brookhaven Code (Chain)	Seq. length
Rat tonin	1TON	235
Porcine kallikrein	2PKA (A)	232
Bovine trypsin (orthorhombic)	2PTN	223
Rat trypsin	2TRM	223
Bovine α-chymotrypsin	4 CHA (A)	245
Rat mast cell proteinase	3RP2 (A)	224
Streptomyces griseus proteinase A	2SGA	181
Streptomyces griseus proteinase B	3SGB (E)	185
Lysobacter enzymogenes Alpha-lytic protease	2alp	198

Table 1: The nine serine protease sequences used in Example 6.1.

observation that with the number of sequences to be aligned, the agreement increases between biologically "plausible alignments" – that is, alignments based on phylogenetic or structural information concerning the molecules involved – and alignments which are (sub)optimal relative to an appropriately defined sum-of-pairs score [29]).

Following the principles proposed by M. McClure, T. Vasi, and W. Fitch [31], we have evaluated our results[5] rigorously by measuring the degree by which they identify certain structurally conserved subsequences (often called *motifs*) which are known (or supposed) to be important for the structure and/or function of the molecule. In principle, that degree is measured not just by the size of the *largest* set of sequences correctly aligned with each other relative to a pre-given motif, but by the *number and size* of the various subclasses of correctly aligned sequences [11].

6.1 Alignment of Serine Protease Sequences

As a first example, we consider the protein family of *serine proteases*. A number of mammalian and microbial protein structures of that family have been determined by X-ray crystallography and, although there is less than 21 percent sequence identity between the mammalian and microbial serine proteases, it has been observed that they all adopt similar three-dimensional structures.

D.J. Lipman *et al.* [29] aligned five mammalian sequences using the `MSA` procedure. They observed a high agreement of the resulting multiple sequence alignment with the structural alignment presented in [18].

In the following, we extend their work by presenting an alignment of nine serine protease sequences with known three-dimensional structure (cf. Table 1), consisting of six mammalian (rat tonin, porcine kallikrein, bovine trypsin, rat trypsin, bovine α-chymotrypsin, and rat mast cell proteinase) and three microbial ones (proteinase A and B from Streptomyces griseus, and α-lytic protease from Lysobacter enzymogenes). As a standard of truth, we take the structural alignment of the EMBL 3d-ali database[6].

[5] For an evaluation of the results of DCA regarding the four protein families considered in [31], cf. [47].

[6] This is a collection of protein structural families, each consisting of (a) non-redundant and multiple tertiary structural superpositions and (b) resulting primary sequence alignments [37] which can be found on the World Wide Web at `http://www.embl-heidelberg.de/argos/ali/ali.html`.

```
                                  motif 1        motif 2
1ton   ---------------ivggykceknsqpwQVAV----IneyL--CG-GVLIdps---WVI
2pka   ---------------iiggreceknshpwQVAI--YhyssfQ--CG-GVLVnpk---WVL
2ptn   ---------------ivggytcgantvpyQVSL----NsgyhF-CG-GSLInsq---WVV
2trm   ---------------ivggytcqensvpyQVSL----NsgyhF-CG-GSLIndq---WVV
4cha   cgvpaiqpvlsglsrivngeeavpgswpwQVSL--QdktgfhF-CG-GSLInen---WVV
3rp2   ---------------iiggvesiphsrpyMAHLDivtekglrvICG-GFLIsrq---FVL
2sga   ---------------iagGEA---------IT-----tggsR--CSLGFNVsvngvaHAL
3sgb   ---------------isgGDA---------IY-----sstgR--CSLGFNVrsgstyYFL
2alp   -------------anivgGIE-------YSin-----nasL---CSVGFSVtrgatkGFV

       motif 3  motif 4                    motif 5
1ton   TAAHCY---snNYQvllgrnnlfkdep---fa----qrrlVRQSFRhpdy--iplivtnd
2pka   TAAHCK---ndNYEvwlgrhnlfenen---ta----qffgVTADFPhpgf----nlsadg
2ptn   SAAHCY---ksGIQvrlgedninvveg---ne----qfisASKSIVhpsy--nsntl---
2trm   SAAHCY---ksRIQvrlgehninvleg---ne----qfvnAAKIIKhp------nfd--r
4cha   TAAHCGvttSDVvvagefdqg---------sssekiqklkIAKVFKnsky--nslti---
3rp2   TAAHCK---grEITvilgahdvrkres---tq----qkikVEKQIIhesynsvpn-----
2sga   TAGHCT---nisaSWSiGTRTG--------Ts------------fp--------------
3sgb   TAGHCT---dgatTWWansarttvlGTTSGSs------------fp--------------
2alp   TAGHCG---tvnaTARiggavvG-------TF--------AARvfp--------------

               motif 6                          motif 7  motif 8
1ton   teqpvhdhSNDLMLLHLSepa----di---tggvkvidlptk--EPKVgSTCLASGWgst
2pka   kdy-----SHDLMLLRLQspa----ki---tdavkvlelptq--EPELgSTCEASGWgsi
2ptn   --------NNDIMLIKLKsaa----sl---nsrvasislpts--CASAgTQCLISGWgnt
2trm   ktl-----NNNIMLIKLSspv----kl---narvatvalpss--CAPAgTQCLISGWgnt
4cha   --------NNDITLLKLStaa----sf---sqtvsavclpsasdDFAAgTTCVTTGWglt
3rp2   --------LHDIMLLKLEkkv----el---tpavnvvplpspsdFIHPgAMCWAAGWgkt
2sga   --------NNDYGIIRHSnpaaadgrvylyngsyqdit-tag--NAFVgQAVQRSGSttg
3sgb   --------NNDYGIVRYTnttipk-dg---tvggqdit-saa--NATVgMAVTRRGSttg
2alp   --------GNDRAWVSLTsaqtllprv---angssfvtvrgst-EAAVgAAVCRSGRttg

                            motif 9               motif 10
1ton   --------npsemvvshdlqCVNIHLLSn---ekcietykdnvtdvMLCAgemeggkDTC
2pka   epgpddfefpde------iqCVQLTLLQn---tfcadahpdkvtesMLCAgylpggkDTC
2ptn   --kssgtsypdv------lkCLKAPILSd---sscksaypgqitsnMFCAgyleggkDSC
2trm   --lssgvnepdl------lqCLDAPLLPq---adceasypgkitdnMVCVgfleggkDSC
4cha   --rytnantpdr------lqQASLPLLSn---tnckkywgtkikdaMICAg--asgvSSC
3rp2   -----gvrdptsyt----lrEVELRIMDe---kacvd-yryyeykfQVCVgspttlrAAF
2sga   ------------------LRSGSVTGl-natvnygssgivygMIQTnvCAQ---------
3sgb   ------------------THSGSVTAl-natvnygggdvvygMIRTnvCAE---------
2alp   ------------------YQCGTITAk-nvt-anyaegavrgLTQGnaCMG---------

        motif 11          motif 12                      motif 13
1ton   AGDSGGPL--ICdg----VLQGITSGGAt-----pca--kpktpaIYAKLIKFTSWIkkv
2pka   MGDSGGPL--ICng----MWQGITSWGHt-----pcg--sankpsIYTKLIFYLDWIddt
2ptn   QGDSGGPV--VCsg----KLQGIVSWGSg------ca--qknkpgVYTKVCNYVSWIkqt
2trm   QGDSGGPV--VCng----ELQGIVSWGYg------ca--lpdnpgVYTKVCNYVDWIqdt
4cha   MGDSGGPL--VCkkngawTLVGIVSWGSs-----tc---ststpgVYARVTALVNWVqqt
3rp2   MGDSGGPL--LCag----VAHGIVSYGH-------p---dakppaIFTRVSTYVPWInav
2sga   PGDSGGSL--FAgs----TALGLTSGGSg-----nc---rtggttFYQPVTEALSAYgat
3sgb   PGDSGGPL--YSgt----RAIGLTSGGSg-----nc---ssggttFFQPVTEALVAYgvs
2alp   RGDSGGS-WITsag----QAQGVMSGGNvqsngnncgipasqrssLFERLQPILSQYgls

1ton   mkenp
2pka   itenp
2ptn   iasn-
2trm   iaan-
4cha   laan-
3rp2   in---
2sga   vl---
3sgb   vy---
2alp   lvtg-
```

Table 2: The nine serine protease sequences used in Example 6.1.

	Organism	GenBank acc.no.	Seq. length
Mouse	Mus musculus	J03151	274
Rat	Rattus norvegicus	(see caption)	273
Human	Homo sapiens	X51867	264
Bovine	Bos taurus	Z25280	278
Frog	Xenopus laevis	Z11844	276
Yeast1	Saccharomyces cerevisiae	Z14231	339
Yeast2	Schizosaccharomyces pombe	X52530	399
Plant	Arabidopsis thaliana	(see caption)	261

Table 3: The eight currently known RNase MRP RNA sequences used in Example 6.2. The Arabidopsis sequence and the Rat sequence are not contained in GenBank. They were obtained from [43] and from [27], respectively.

In Figure 3, we present an alignment produced by DCA, using stop size $L = 30$, free shift, and the Blosum30 substitution matrix [21] (converted to distances with integer values from 0 to 27) with an affine gap penalty function of $g(l) = 5 + 10 \cdot l$.

There are thirteen motifs for which the predominant secondary structure is known [26]. Six motifs are aligned correctly over all nine sequences (Figure 3: motif 3, 6, 7, 8, 12, and 13), the motifs 2 and 11 are misaligned, though only at their very beginning or end, respectively, and three motifs are correctly aligned within each group of mammalian and microbial sequences (motifs 4, 9, and 10), but the alignment across the subfamilies is wrong. For the remaining two motifs, the alignment of the (larger) set of mammals is correct (motif 5) or "reasonable" (motif 1), while – with respect to the microbial sequences – only one is almost correct (motif 1) and the other is incorrect (motif 5). Altogether, there are 77 of 111 positions belonging to some motif correctly aligned over *all* sequences.

Even though commonly used alignment procedures (e.g. CLUSTAL W [51]) perform well within the group of mammalian or microbial serine protease, respectively, the quality of the alignments decline significantly when members of both groups are to be aligned. None of the alignment procedures we have tested was able to align more than the motifs 2, 3, 6, 12, and 13 correctly.[7]

One crucial set of parameters for sequence alignment is the substitution matrix. If the mutation process of the (sub)sequences used for scoring the alignment is modelled well by a substitution matrix, then the optimal alignments (with respect to the resulting scoring system) are relatively independent of the gap penalties (within some reasonable range, see for example Section 6.3). We suspect that the main difficulty of generating a good alignment across the subfamilies of mammalian and microbial serine protease sequences is the (present) lack of the option to use pair-dependent substitution matrices from a parametrised set of such matrices appropriate for this protein family.

6.2 Alignment and Phylogeny of RNase MRP RNA Sequences

As a second example, we have tested DCA on a set of highly diverged RNase MRP RNA sequences. The eight currently known sequences[8] are from human, bovine, mouse, rat, frog, two yeasts (Saccharomyces cerevisiae and Schizosaccharomyces pombe) and plant (Arabidopsis) (see Table 2). In the following, we will refer to Saccharomyces cerevisiae and Schizosaccharomyces pombe as yeast1 and yeast2, respectively.

There are three motifs which we have highlighted in Figure 4 by using capital letters. Note that – with the exception of the second and third motif in yeast2 – all motifs are reasonably well aligned. The reason why yeast2 appears to be difficult to align is that this sequence is significantly longer than all the other sequences (in fact it contains an additional subsequence of length 60 between the first and the second single-stranded region) and the motifs 2 and 3 of yeast2 differ significantly from the corresponding motifs of the other sequences. The gaps occurring in motif 2 in all sequences other than yeast1 correspond to an (unusual) loop in the secondary structure of the yeast1 sequence, unique among the RNase MRP RNA sequences found so far. None of the commonly used programs we have tested on this data set (CLUSTAL W, TREEALIGN, MALIGN, PILEUP) was able to align this RNA family with respect to the motifs as well as DCA – in particular, no other program predicted the loop.

In order to analyse why successive alignment procedures fail to produce a reasonable alignment for this data set, we investigated the phylogenetic relationships between these species. In particular, we looked at the potential (phylogenetic) trees guiding an iterative alignment procedure. To this end, we applied the SplitsTree program [13]. This program does not always return a (proposal for a) phylogenetic *tree*, – in general, it returns *graphs* which indicate conflicting "phylogenetic" information in the data set which does not all fit into a single tree. In Figure 5, we show the graph generated by the SplitsTree program for the set of Hamming distances derived from pairwise alignments.

Obviously, any reasonable tree guiding a successive alignment puts the two yeasts and the Arabidopsis sequence close together. But any successive alignment of these three sequences is incorrect for the second and third motif due to the fact that pre-aligning any pair of these sequences goes wrong in this respect. Moreover, a simultaneous alignment of only these three sequences, e.g. by `MSA`, does neither identify motif 2 nor motif 3, not even for any two of these three sequences. But including the vertebrate sequences, a *simultaneous* alignment clearly identifies these motifs at least for yeast1 and the plant sequence (see Figure 5).

6.3 Alignment of 12S RNA Sequences

In the third and final example, we present an alignment of ten 12S RNA sequences of average length 290 for which an alignment based on secondary structure had been worked out in [22]. Five of these sequences represent phyla of vertebrates (cow, bird,

[7] Motif 1 appears to be exceptional in that a surprisingly high sequence identity between the amino acids at positions 16–19 is observed (which for all except the bovine α-chymotrypsin and the Lysobacter enzymogenes α-lytic protease sequence are actually the first non-gap positions) although, for the microbial sequences, that position 19 is actually the first site of motif 1 and therefore corresponds to position 30 rather than 19 of the mammalian sequence alignment.

[8] There are actually two more tobacco sequences. Because they are almost identical to the Arabidopsis sequence, we have excluded them here.

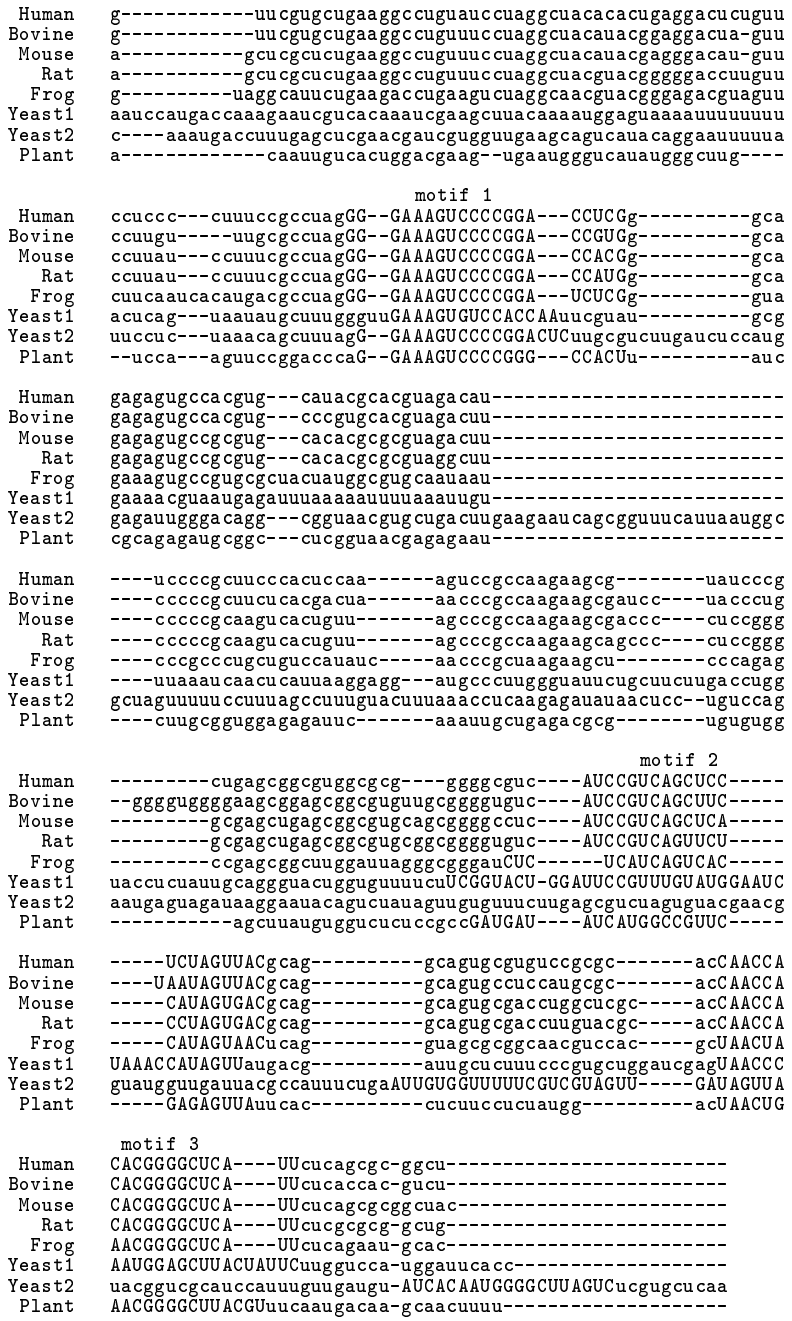

Figure 3: An alignment of eight RNase MRP RNA sequences, computed with DCA using a weighted transition/transversion matrix for substitutions of 2/5, and a linear gap penalty function of $g(l) = 6 + 2 \cdot l$.

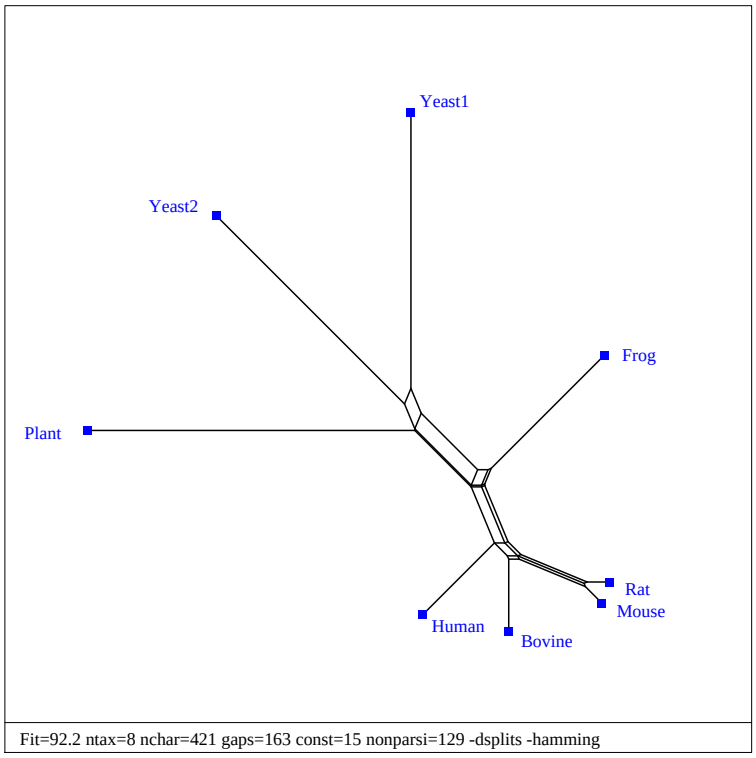

Figure 4: The SplitsTree graph for an alignment of eight RNase MRP RNA sequences obtained by DCA using Hamming distance.

	Organism	GenBank acc.no.	Seq. length
Cow	Bos taurus	J01394	302
Moa	Megalapteryx didinus (Ratite Bird)	X67634	295
Scincid Lizard	Leiolopisma nigriplantare polychroma	unpublished	290
Toad	Xenopus laevis (Xenopus; Pipidae)	M10217	300
Cyprinid Fish	Crossostoma lacustre	M91245	297
Sea Urchin	Paracentrotus lividus	J04815	287
Fly	Drosophila yakuba	X03240	288
Honeybee	Apis mellifera	L06178	278
Cicada	Magicicada tredecim	unpublished	271
Earthworm	Aporrectodea rosea	L02392	285

Table 4: Ten 12S RNA sequences used in Example 6.3.

	number and sizes of correctly aligned subclasses					
	transition:transversion penalty					
	2:5					2:6
motif	CLUSTAL W	MALIGN	PILEUP	TREEALIGN	DCA	
1	1 (10)	1 (10)	1 (10)	1 (10)	1 (10)	1 (10)
2	1 (10)	1 (10)	1 (10)	1 (10)	1 (10)	1 (10)
3	3 (6,2,2)	2 (8,2)	3 (8,1,1)	2 (8,2)	2 (8,2)	2 (9,1)
4	1 (10)	2 (8,2)	2 (9,1)	2 (9,1)	2 (8,2)	1 (10)
5	1 (10)	1 (10)	1 (10)	1 (10)	1 (10)	1 (10)
6	1 (10)	1 (10)	1 (10)	1 (10)	1 (10)	1 (10)
7	2 (9,1)	2 (9,1)	2 (9,1)	2 (8,2)	2 (9,1)	2 (9,1)
8	2 (8,2)	4 (7,1,1)	4 (5,3,1,1)	2 (8,2)	3 (7,2,1)	2 (9,1)
9	1 (10)	1 (10)	2 (9,1)	1 (10)	1 (10)	1 (10)
10	2 (7,3)	2 (7,3)	2 (6,4)	2 (7,3)	2 (7,3)	2 (8,2)

Table 5: The number and sizes of correctly aligned subclasses among the ten 12S RNA sequences relative to the motifs $1-10$ achieved by the various alignment programs using a fixed penalty of 2 for transitions and 5 for transversions, and those gap penalties which optimise the sum of the maximal number of correctly aligned sequences taken over all motifs. The last column contains the corresponding number and sizes achieved by DCA using a penalty of 6 for transversions (and 2 for transitions).

lizard, frog, and fish), and five represent phyla of invertebrates (sea urchin, earthworm, fly, honeybee, and cicada). From the structural alignment which we accept as the one to aim for, ten highly conserved "motifs" could be identified, comprising altogether 69 sites, among which 30 are constant and 34 include transitions but no transversions, while only 5 include transitions as well as transversions, and none contain gaps. The 245 "non-motif" sites include 72 gap sites. Among the 240 sites including at least two "non-gaps", there are 30 constant sites, 128 sites include transitions only, and 82 sites include both, transitions and transversions.

These ten sequences have been used already in [23] where five alignment programs (CLUSTAL W, MALIGN, PILEUP, TREEALIGN, and DCA) were tested regarding their ability to correctly align the motifs within the sequences. As one of these programs (TREEALIGN) has its transition:transversion penalties fixed at 2:5, the same values were also used as the transition:transversion penalties in the remaining programs to get a "fair" comparison. Then, in all five programs, the [gap open/gap extension] parameters were allowed to vary, and the resulting alignments were scored by summing up – from motif 1 to motif 10 – the maximal number $N(1), \ldots, N(10)$ of sequences correctly aligned with respect to motif $1, \ldots,$ motif 10, respectively. Finally, those [gap open/ gap extension] parameters were selected for each program which resulted in the highest alignment score. Table 4 summarises the results obtained in this way in [23]. As one can see, all programs scored at least 86 and none scored more than 90 points.

Here, we add to this investigation as follows: As pointed out above, the transversion:transition rate 5:34 at sites within motifs is far below the ratio 2:5. Hence, at least for the DCA procedure, we varied not only the gap parameters but also the transition:transversion penalties. Amazingly enough, enlarging the transversion penalty by just 1, from 5 to 6, and keeping the [gap open/gap extension] parameters at [3/2] (which – for DCA – achieved the best score for the transition:transversion penalties 2:5, investigated before), we immediately hit upon an alignment which scored 95 out of the

```
          motif 1              motif 2             motif 3
     Cow  UGGCGGUgcuuuauauccuuCUAGAGGAgccuguu--cuauaA-UCGauaaaccccgaua
     Moa  UGGCGGUgccccaaacccacCUAGAGGAgccuguu--cuauaA-UCGauaacccacguua
    Skink UGGCGGUgcuccacaucaacCUAGAGGAgccuguc--cuauaA-UCGauacccccgauc
    Uoad  UGGCGGUgcuccaaacccacCUAGAGGAgccuguu--cuguaA-UCGauaccccucgcua
    Fish  UGGCGGUgccuuagacccccCUAGAGGAgccuguu--cuagaA-CCGauaaccccgcuua
  Urchin  UGGCGGUuuuccaaaccuccCUGGAGGAgcuugcc-auug-aA-UCGauaacccacgaaa
     Fly  UGGCGGUauuuua-gucuauCCAGAGGAaccuguu--uuguaA-UCGauaauccacgaug
Honeybee  UGGCGGCuuuuuauaucuagUUAGAGAUguuugucgauua-aU-UUGauaguccacgaua
  Cicada  UGGCGGUaauuua--ucuaaUCAGAGGAaucuguu--uuguaA-UUGauaauccacgaua
    Worm  UGGCGGUgucuua--ucaacCCAGGGGAaccuguc--ucauaaCUCGauaacccacga-a

              motif 4                        motif 5
     Cow  aaccucACCaauucuugcuaa--uacagucUAUAUACCGCCAUCuucagcaaacccuaaa
     Moa  cacccgACCaucucuugccca--ugcagccUACAUACCGCCGUCcccagcccgccu-aug
    Skink uaccucACCgcuuuuugaaac---ucagccUAUAUACCGCCGUCgucagccuaccuuaug
    Uoad  aaccucACCacuucuugccaa--acccgccUAUAUACCACCGUCgccagcccaccucgug
    Fish  aaccucACCacuucuagucau--ccccgccUAUAUACCGCCGUCgucagcuuaccccgug
  Urchin  uaccucACCaacuuuuguaac--aacagcuUGUAUACCAUCGUCguaagucuacuucuu-
     Fly  gaccuuACUuaaauuuguaau----caguuUAUAUACCGUCGUUaucagaauauuuuau-
Honeybee  agaucuACUuaaguuua-aau--------uUAUGUAUUGUUGUUuuaa--uuagcuuga-
  Cicada  gauucuAUUuua------aau----aaauuUGUAUACCUCUGUCaaga--auguuuuau-
    Worm  uuccucACCcucucua--gauucuacagccUGUGUACUGCCGUCguaagcacaccccua-

                                                              motif 6
     Cow  aaggaaaaaaaguaagcguaauuaugauac---auaaaaacguUAGGUCAAGGUGUAacc
     Moa  aaagaacaauagcgagc--acaacagccac-ccgcuaacaagaCAGGUCAAGGUAUAgca
    Skink aaagaaguauaguaagc--aaaauagucac-caacuaaaacguCAGGUCAAGGUGUAgca
    Uoad  agagauucuuaguaggcuuaaugauuuuuc---aucaacacguCAGGUCAAGGUGUAgca
    Fish  aaggcucaauaguaagc--aaagugggcac-aacccaaaacguCAGGUCGAGGUGUAgcg
  Urchin  -----gagaaaguugac---uuuaagggag-aacccuggacguCAGAUCAAGGUGCAgcc
     Fly  -aagaauaauaauauuc--aauaauuuuaauaaaaauuuauauCAGAUCAAGGUGUAgcu
Honeybee  -auuuauuuggguuaga---gaaauuuuauauaauaauaau-uCAAAUCAAAAUGUAgua
  Cicada  ----cagaauaauuuuc--auuuguuuuau--uaaaaaaaaguCAGGUCAAGGUGCAguu
    Worm  -aaagaaagaagugugc-agacaugauuaa-acucauauacguCAGGUCAAAGUGCAgcc

                          motif 7
     Cow  uaugaaaugggaagaaaugggcUACAuucucua----caccaagagaaucaagcacgaaa
     Moa  uaugaga-uggaagaaaugggcUACAuuuucuaa---cauagaacaccacga------aa
    Skink cauaaag-uggaagagaugggcUACAcucucu-----cccagagaacacga-------ac
    Uoad  uaugaaguggggaagaaaugggcUACAuuuucuac---cuuagaauaaacga-------aa
    Fish  uacgaaguggggaagagaugggcUACAuuuucu-----acuagaauaagacga------au
  Urchin  uauaaguuggggauaggugagcUACAauguuug----aacaaaccagugg--------aa
     Fly  uauauuu-aaguaauaaauggguUACAauaaaauuua--uuuaaacggauaa--------aa
Honeybee  uauuuaa-gauauaagaugaaUAUCAauaaauuuaauuuuagaauua-----------uu
  Cicada  aauuuua-aagaaagaauggauUACAuuau-------uguaaaaaauga---------au
    Worm  cauggga-gggagaugaugggUACAcccua------aacaaagauacgg--------aa

                  motif 8                     motif 9
     Cow  guua-----uuaUGAAAccaauaa--cc----aaaggaggAUUUAGcaguaaa----cu-
     Moa  gaga-----agaUGAAAcucucc---uca---gaaggcggAUUUAGcaguaaa----au-
    Skink agcau----caaUGAAAcacugc---uc----aaagguggAUUUAGuaguaag----au-
    Uoad  gaucu----cuaUGAAAccaga----ucgagaaaaggcggAUUUAGcaguaaa----ga-
    Fish  agca-----ucaUGAAAacuuaaugcuu----gaaggaggAUUUAGuaguaaa----aa-
  Urchin  ggag-----ggaUGAAAuacccc---ucg---gaaauuggAUUCAGcaguaag-cccca-
     Fly  ---------uuaUGAAAaaauuu---uu----gaagguggAUUUGGuaguaaa----auu
Honeybee  uauuu----UAAUAuaaaua------u-----gaaagagaAUUUAAaaguaaa----uu-
  Cicada  uguuuucuauaaUGAAAauca-----u-----gaaacuggAUUUGAaaguaaa-uuuca-
    Worm  uauag----uacUAAAAgcuaua---u-----aaauuauuACUUGGuuguaacguuucu-

                          motif 10
     Cow  ---aagaauagag--ugcuuaguUGAAuuaggcca
     Moa  ---aggacaagaa-cgcccauuuUAAGcuggccc-
    Skink ----aaacaagag--acuuaucuUAAAccagccc-
    Uoad  ---gaaacaagagaguuccucuuUAAAacggccc-
    Fish  ---ggaaauagag--uguccuuuUGAAcccggcuc
  Urchin  -cuaagacaaugg------gacUGAAaagagcuc-
     Fly  auaaagauuauaa----auaauuUGAUuuuagcuc
Honeybee  ---aaaguaauuu---cuuuaauUGAAgauaguau
  Cicada  ---uuaaauaugu------guuuUGAAuuuagguc
    Worm  -ucaaaacuaaag---------UGAAuaugaauc-
```

Figure 5: An alignment of ten 12S RNA sequences, calculated by DCA using a weighted transition/transversion matrix for substitutions of 2/6, and a linear gap penalty function of $g(l) = 3 + 2 \cdot l$.

maximally 100 points – that is, it basically halved the number of mistakes (see Fig. 6 and the last column of Table 4). Moreover, varying the [gap open/gap extension] parameters $[i, j]$ within the range $\{2, 3, 4\}$ for i and $\{1, \ldots, 6\}$ for j, respectively, never produced an alignment which scored less than 88 points. Similar results were obtained for transversion penalties from 7 to 9, while above 10 the quality of the resulting alignments started to decrease again.

In view of similar observations regarding further data sets, it seems to be characteristic for (close-to) optimal sum-of-pairs score alignments that, for them, "good" substitution parameters may be more important than "good" gap penalty parameters and that, for appropriate substitution parameters, the quality of the resulting alignments are relatively robust against changes of the gap penalties. Consequently, simultaneous alignment procedures appear to have the additional advantage compared with iterative procedures that they are almost independent of gap penalty parameters provided good estimates for the substitution parameters are available and a reasonable large number of homologous sequences are being aligned.

Acknowledgements

We wish to thank Lesley Collins for providing us with the RNase MRP RNA sequences, their structural alignment and references and Robert Hickson for the 12S RNA sequences, their various alignments and references. Also, the first and the last named authors worked on the final stages of this paper as visitors of the Biomathematical Research Centre at the Mathematics Department, University of Canterbury (NZ), and they want to thank its Director Dr. Mike Steel and the Mathematics Department for their great and generous hospitality.

References

[1] S.F. Altschul. Gap Costs for Multiple Sequence Alignment. *J. Theor. Biol.* 138, pages 297–309, 1989.

[2] S.F. Altschul and B.W. Erickson. Optimal Sequence Alignment Using Affine Gap Costs. *Bull. Math. Biol.*, 48(5/6), pages 603–616, 1986.

[3] S.F. Altschul and D.J. Lipman. Trees, Stars, and Multiple Biological Sequence Alignments. *SIAM J. of Appl. Math.*, 49, pages 197–209, 1989.

[4] H.-J. Bandelt and A.W.M. Dress. A Canonical Decomposition Theory for Metrics on a Finite Set. *Adv. Math.*, 92, pages 47–105, 1992.

[5] H.-J. Bandelt and A.W.M. Dress. Split Decomposition: A New and Useful Approach to Phylogenetic Analysis of Distance Data. *Molecular Phylogenetics and Evolution*, 1(3), pages 242–252, 1992.

[6] G. Brinkmann, A.W.M. Dress, S.W. Perrey, and J. Stoye. Two Applications of the Divide&Conquer Principle in the Molecular Sciences. to appear in *Proceedings of the Conference of the International Society on Mathematical Programming*, Lausanne, 1997.

[7] H. Carrillo and D.J. Lipman. The Multiple Sequence Alignment Problem in Biology. *SIAM J. Appl. Math.*, 48(5), pages 1073–1082, 1988.

[8] S.C. Chan, A.K.C. Wong, and D.K.Y. Chiu. A Survey of Multiple Sequence Comparison Methods. *Bull. Math. Biol.*, 54(4), pages 563–598, 1992.

[9] M.O. Dayhoff, R.M. Schwartz, and B.C. Orcutt. A Model of Evolutionary Change in Proteins. In M.O. Dayhoff, editor, *Atlas of Protein Sequences and Structure*, volume 5, suppl. 3, pages 345–352. National Biomedical Research Foundation, Washington, D.C., 1979.

[10] R.F. Doolittle. Computer Methods for Macromolecular Sequence Analysis. *Methods in Enzymology* 266, Academic Press, San Diego, USA, 1996.

[11] A.W.M. Dress and S.W. Perrey. The Quality of Multiple Sequence Alignment – A Quantitative Approach. In preparation, Universität Bielefeld, 1997.

[12] A.W.M. Dress, G.Füllen, and S.W. Perrey. A Divide and Conquer Approach to Multiple Alignment. In *Proc. of the Third Conference on Intelligent Systems for Molecular Biology, ISMB 95*, pages 107–113. AAAI Press, Menlo Park, CA, USA, 1995.

[13] A.W.M. Dress, D. Huson, and V. Moulton. Analyzing and Visualizing Sequence and Distance Data Using SplitsTree. *Disc. Appl. Math.*, 71, pages 95–109, 1996.

[14] D.-F. Feng and R.F. Doolittle. Progressive Sequence Alignment as a Prerequisite to Correct Phylogenetic Trees. *J. Mol. Evol.*, 21, pages 112-125, 1987.

[15] W.M. Fitch and T.F. Smith. Optimal Sequence Alignments. *Proc. Natl. Acad. Sci. USA*, 80, pages 1382–1386, 1983.

[16] Genetics Computer Group, Inc. Wisconsin Sequence Analysis Package, version 8.1. Madison, Wisconsin, 1995.

[17] O. Gotoh. An Improved Algorithm for Matching Biological Sequences. *J. Mol. Biol.*, 162, pages 705–708, 1982.

[18] J. Greer. A Model of a Specific Interaction: Salt-bridges Form Between Prothrombin and its Activating Enzyme Blood Clotting Factor X_a. *J. Mol. Biol.*, 153, pages 1043–1053, 1981.

[19] S.K. Gupta, J.D. Kececioglu, and A.A. Schäffer. Improving the Practical Space and Time Efficiency of the Shortest-Paths Approach to Sum-of-Pairs Multiple Sequence Alignment. *J. Comp. Biol.*, 2(3), pages 459–472, 1995.

[20] J.J. Hein. TreeAlign. In Griffin, A. M. and Griffin, H. G., editor, *Computer Analysis of Sequence Data, Part II*, volume 24 of *Methods in Molecular Biology*, chapter 28, pages 349–364 Humana Press, Totowa, NJ, USA, 1994.

[21] S. Henikoff and J.G. Henikoff. Amino Acid Substitution Matrices from Protein Blocks. *Proc. Natl. Acad. Sci. USA*, 89, pages 10915–10919, 1992.

[22] R.E. Hickson, C. Simon, A.C. Cooper, G.S. Spicer, J. Sullivan, D. Penny. Conserved Sequence Motifs, Alignment, and Secondary Structure for the Third Domain of Animal 12S rRNA. *J. Mol. Biol. Evol.*, 13, pages 150–169, 1996.

[23] R.E. Hickson, C. Simon, and S.W. Perrey. An Evaluation of Multiple Sequence Alignment Programs Using an rRNA Data Set, Submitted, 1997.

[24] D.G. Higgins, A.J. Bleasby, and R. Fuchs. CLUSTAL V: Improved Software for Multiple Sequence Alignment. *CABIOS*, 8, pages 189–191, 1991.

[25] D.S. Hirschberg. A Linear Space Algorithm for Computing Maximal Common Subsequences. *Communications of the ACM* 18(6), pages 341–343, 1975.

[26] W. Kabsch and C. Sander. A Dictionary of Protein Secondary Structure. *Biopolymers* 22, pages 2577–2637, 1983.

[27] T. Kiss, C. Marshallsay, and W. Filipowicz. 7-2/MRP RNAs in Plant and Mammalian Cells: Association with Higher Order Structures in the Nucleolus. *The EMBO Journal*, 11(10), pages 3737–3746, 1992.

[28] J.A. Lake. The Order of Sequence Alignment Can Bias the Selection of Tree Topology. *J. Mol. Biol. Evol.*, 8, pages 378–385, 1991.

[29] D.J. Lipman, S.F. Altschul, and J.D. Kececioglu. A Tool for Multiple Sequence Alignment. *Proc. Natl. Acad. Sci. USA*, 86, pages 4412–4415, 1989.

[30] H.M.Martinez. A Flexible Multiple Sequence Alignment Program. *Nucl. Acids Res.*, 16, pages 1683–1691, 1988.

[31] M.A. McClure, T.K. Vasi, and W.M. Fitch. Comparative Analysis of Multiple Protein-Sequence Alignment Methods. *J. Mol. Biol. Evol.*, 11(4), pages 571–592, 1994.

[32] B.R. Morton. The Influence of Neighboring Base Composition on Substitutions in Plant Chloroplast Coding Sequences. *J. Mol. Biol. Evol.*, 14(2), pages 189–194, 1997.

[33] B. Morgenstern, J. Stoye, and A.W.M. Dress. Some Theoretical Aspects of Pairwise and Multiple Sequence Alignment. In preparation, Universität Bielefeld, 1997.

[34] E.W. Myers. An Overview of Sequence Comparison Algorithms in Molecular Biology. Technical Report TR 91-29, University of Arizona, Tucson, Department of Computer Science, 1991.

[35] E.W. Myers and W. Miller. Optimal Alignments in Linear Space. *CABIOS*, 4(1), pages 11–17, 1988.

[36] S.B. Needleman and C.D. Wunsch. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *J. Mol. Biol.*, 48, pages 443–453, 1970.

[37] S. Pascarella and P. Argos. A Data Bank Merging Related Protein Structures and Sequences. *Prot. Eng.*, 5(2), pages 121–137, 1992.

[38] S.W. Perrey and M.D. Hendy. Alignment Error Detection Using Extended Hadamard Conjugation. In preparation, Massey University, Palmerston North, 1997.

[39] S.W. Perrey and J. Stoye. Fast Approximation to the NP-hard Problem of Multiple Sequence Alignment. Information and Mathematical Sciences Reports, Series B:96/06 (ISSN 1171-7637), May 1996.

[40] S.W. Perrey, J. Stoye, and V. Moulton. Fast and Accurate Approximation to Sum-of-Pairs Score Optimal Multiple Sequence Alignment. Manuscript, Christchurch, 1997.

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

[42] A.A. Schäffer. `MSA` version 2.1. `http://alfredo.wustl.edu/msa/msa.tar.gz`, 1995.

[43] M.E. Schmitt, D.J. Bennett, D.J. Dairaghi, and D.A. Clayton. Secondary Structure of RNase MRP RNA as Predicted by Phylogenetic Comparison. *The FASEB Journal*, 7, pages 208–213, 1993.

[44] G.D. Schuler, S.F. Altschul, and D.J. Lipman. A Workbench for Multiple Alignment Construction and Analysis. *PROTEINS, Structure, Function, and Genetics* 9, pages 180–190, 1991.

[45] T.F. Smith and M.S. Waterman. The Identification of Common Molecular Subsequences. *J. Mol. Biol.* 147, pages 195–197, 1981.

[46] R.F. Smith and T.F. Smith. Pattern-Induced Multi-Sequence Alignment (PIMA) Algorithm Employing Secondary Structure-Dependent Gap Penalties for Use in Comparative Protein Modelling. *Protein Engng.* 5(1), pages 35–41, 1992.

[47] J. Stoye. Divide-and-Conquer Multiple Sequence Alignment. *Dissertation*, Technische Fakultät der Universität Bielefeld, 1997.

[48] J. Stoye. DCA: Divide and Conquer Multiple Sequence Alignment. `http://bibiserv.techfak.uni-bielefeld.de/dca/`, 1997.

[49] J. Stoye, S.W. Perrey, and A.W.M. Dress. Improving the Divide-and-Conquer Approach to Sum-of-Pairs Multiple Sequence Alignment. *Appl. Math. Lett.*, 10(2), pages 67–73, 1997.

[50] W.R. Taylor. A Flexible Method to Align Large Numbers of Biological Sequences. *J. Mol. Evol.*, 28, pages 161-169, 1988.

[51] J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTAL W: Improving the Sensitivity of Progressive Multiple Sequence Alignment through Sequence Weighting, Position-specific Gap Penalties and Weight Matrix Choice. *Nucl. Acids Res.*, 22(22), pages 4673–4680, 1994.

[52] J.L. Thorne and H. Kishino. Freeing Phylogenies from Artifacts of Alignment. *J. Mol. Biol. Evol.*, 9(6), pages 1148–1162, 1992.

[53] U. Tönges, S.W. Perrey, J. Stoye, and A.W.M. Dress. A General Method for Fast Multiple Sequence Alignment. *Gene* 172, pages GC33–GC41, 1996.

[54] M. Vingron and P. Argos. Motif Recognition and Alignment for Many Sequences by Comparison of Dotmatrices. *J. Mol. Biol.*, 218, pages 33-43, 1991.

[55] M. Vingron and A. von Haeseler. Towards Integration of Multiple Alignment and Phylogenetic Tree Reconstruction. *J. Com. Biol.*, 4(1), pages 23–34, 1997.

[56] R.A. Wagner and M.J. Fischer. The String-to-String Correction Problem. *Journal of the ACM*, 21(1), pages 168–173, 1974.

[57] L. Wang and T. Jiang. On the Complexity of Multiple Sequence Alignment. *J. Comp. Biol.* 1(4), pages 337–348, 1994.

[58] M.S. Waterman. Sequence Alignments in the Neighbourhood of the Optimum with the General Application to Dynamic Programming. *Proc. Natl. Acad. Sci. USA*, 80, pages 3123–3124, 1983.

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

[60] W. Wheeler and D. Gladstein. Malign version 2.3. American Museum of Natural History, 1994.