J. theor. Biol. (1995) **174,** 119–129

Alignment of Molecular Sequences Seen as Random Path Analysis

M. Q. Zhang and T. G. Marr

Cold Spring Harbor Laboratory, P.O. Box 100, *Cold Spring Harbor, New York* 11724, *U.S.A.*

(*Received on* 1 *September* 1992, *Accepted in revised form on* 30 *November* 1994)

We propose a generating functional method—random path analysis (RPA)—that generalizes the classical dynamic programming (DP) method widely used in sequence alignments. For a given cost function, DP is a deterministic method that finds an optimal alignment by minimizing the total cost function for all possible alignments. By allowing uncertainty, RPA is a statistical method that weights fluctuating alignments by probabilities. Therefore, DP maybe thought of as the deterministic limit of RPA when the fluctuations approach zero. DP is the method of choice if one is only interested in optimal alignment. But we argue that, when information beyond the optimal alignment is desired, RPA gives a natural extension of DP for biological applications. As an algebraic approach, RPA is computationally intensive for long sequences, but it can provide better parametric control for developing analytical or perturbational results and it is more informative and biologically relevant. The idea of RPA opens up new opportunities for simulational approaches and more importantly it suggests a novel hardware implementation that has the potential of improving the way a sequence alignment is done. Here we focus on deriving a mathematically rigorous solution to RPA both in its combinatorial form and in its graphical representation; this puts DP in logical perspective under a more general conceptual framework.

1. Introduction

Biologists frequently compare molecular sequences (i.e. DNA and protein sequences) to see if any relevant similarities can be found with sequences already existing in the sequence databases (e.g. GenBank). Typically, the actual comparison is performed by running one of the popular computer programs used for screening the sequence databases (e.g. Pearson, 1990). The result is usually a list of potentially significant "hits", leaving the final decision concerning biological significance up to the user of the program. But, as described by Waterman (1989),

> Sequence alignments are frequently obtained in an *ad hoc* manner, and simply written down in a way that makes the result 'look good'. Sometimes important but implicit criteria have been satisfied... However, alignments that are pleasing to the eye of the scientist may not have much merit beyond that. The commonly used phrases 'aligned so as to maximize homology' and 'gaps inserted in order to maximize homology' might conceal what has been attempted and 'homology' is left undefined.

In 1970, apparently without knowing about the dynamic programming (DP) algorithms introduced by Richard Bellman in computer science, Needleman & Wunsch (1970) provided the first mathematically rigorous dynamic programming method, which has had a great deal of influence in molecular sequence alignment. Incorporating indels (insertions and deletions), on a same footing as mismatches, into a similarity function $S(\mathbf{a}, \mathbf{b})$, their method had the advantage that an explicit criterion for the optimality of alignment could be stated, and an efficient algorithm providing a solution could be given. Later, in the early 1970s, Stan Ulam and colleagues defined a distance function, $D(\mathbf{a}, \mathbf{b})$, which equipped the space of sequences with a *bona fide* metric. Based on minimizing the distance function D, Sellers (1974) gave another DP algorithm which has then been shown by Smith & Waterman (1981) and Fitch *et al.* (1981) to be equivalent to those based on maximizing the similarity function S.

0022–5193/95/100119+11 $08.00/0

To illustrate some basic terminology, we start with a brief discussion of the DP method for sequence alignment. Suppose we want to compare two sequences $\mathbf{a}=a_1a_2,\ldots,a_m$ and $\mathbf{b}=b_1b_2,\ldots,b_n$, where each component can take values from some alphabet set Σ (e.g. the four nucleotides for DNA sequences or the 20 amino acids for protein sequences). Taking a concrete example, a typical alignment for $\mathbf{a}=a_1a_2$ and $\mathbf{b}=b_1b_2b_3$ may be given by

$$\begin{matrix} a_1 & \phi & a_2 & \phi \\ \phi & b_1 & b_2 & b_3 \end{matrix}. \qquad (1)$$

Using the null element ϕ, it is convenient to think of this alignment as an "evolutionary" conversion of sequence **a** to sequence **b**. Traditionally, one chooses a scoring function ϵ (we shall call it the "cost" function) that assigns a cost to each possible conversion of an element (aligned pair): $\epsilon(\phi, b)$ specifies the cost of inserting b, $\epsilon(a, \phi)$ the cost of deleting a and $\epsilon(a, b)$ the cost of substituting b for a (an aligned pair (ϕ, ϕ) of null elements is not allowed). The total cost $E(\Gamma)$ for a particular alignment Γ is given by the sum of the cost of each individual conversion γ along the alignment $E(\Gamma)=\Sigma_{\gamma\in\Gamma}\epsilon(\gamma)(=\epsilon(a_1, \phi)+\epsilon(\phi, b_1)+\epsilon(a_2, b_2)+\epsilon(\phi, \phi_3)$ in the above example). The DP approach is to find an optimal alignment Γ_0 which minimize the total cost E.

Computer scientists have an intuitive graph-theoretic formulation of the problem by using an edge-labeled directed graph $G_{\mathbf{a},\mathbf{b}}$ (the so-called "edit graph"). For our purpose, we prefer to present the same graphical idea in terms of physics, the advantage will become apparent later. It suffices to illustrate the idea with the same example mentioned above.

In Fig. 1, we have plotted the graph $G_{\mathbf{a},\mathbf{b}}$, starting with a null element, we listed **a** and **b** along the columns and the rows, respectively (these columns and rows are also numbered by the integers starting with 0). Anticipating a vivid physical picture, we call the

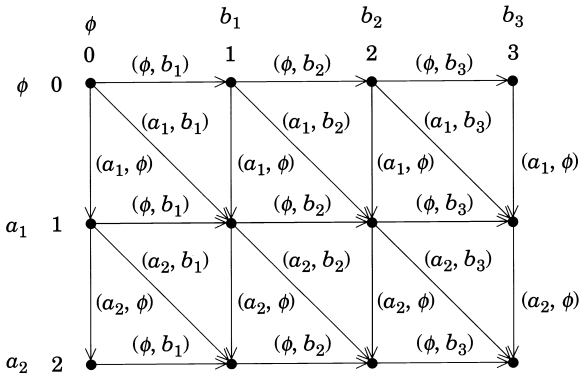

FIG. 1. $G_{\mathbf{a},\mathbf{b}}$ for $\mathbf{a}=a_1a_2$ and $\mathbf{b}=b_1b_2b_3$.

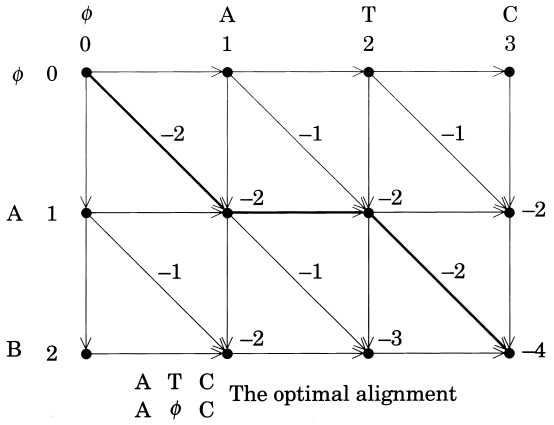

FIG. 2. $G_{\mathbf{a},\mathbf{b}}$ for $\mathbf{a}=ac$, $\mathbf{b}=atc$. The cost for each diagonal jump is indicated on the link, the cost for all horizontal and vertical jumps are zero. Each matrix element $E_{\min}(i,j)$ is indicated (in bold) near the corresponding site. The back-tracking of the optimal alignment is presented by a darker path.

directed graph G a directed *lattice*, its directed edges and vertices are called directed *links* and lattice *sites*, respectively.

Now imagine that a particle starting at the origin (or the source, which is the upper left site (0, 0) of the lattice) can jump from one site to the nearest neighboring site along the direction of a link, and eventually will end up at the end (or the sink, which is the lower right site $(m, n)=(2, 3)$ of the lattice). There is a cost $\epsilon(\gamma)$ for each jump across a link γ. If we associate each link with an aligned pair as indicated in the figure (i.e. a right jump corresponds to an insertion, a down jump to a deletion and a diagonal jump to an substitution), then one can easily see that there is a one-to-one map between the space of all possible alignments of two sequences $\mathbf{a}=a_1a_2,\ldots,a_m$ and $\mathbf{b}=b_1b_2,\ldots,b_n$ and that of all possible directed paths from (0, 0) to (m, n). Furthermore, the problem of finding a best alignment Γ_0 is equivalent (isomorphic) to that of finding an optimal path (also denoted by Γ_0 which minimizes the total cost $(E(\Gamma)=\Sigma_{\gamma\in\Gamma}\epsilon(\gamma))$. For example, the particular alignment [eqn (1)] is indicated by the darker path in Fig. 2. By equating the cost of a link to the length of the link, the problem then becomes the famous shortest path problem which underlies the "action principle" of modern physics (Feynman, 1989). Once the sequence alignment problem is abstracted in this concise mathematical form, it allows a rigorous analysis and its solution may have many general applications as well as theoretical value. From now on, we shall use the particle path representation and the corresponding graphic language inter-changeably with the alignment language whenever possible, and the results may be easily interpreted in the sequence alignment context.

Dynamic programming is a procedure used primarily to improve the computational efficiency of a certain class of deterministic optimization problems by decomposing them into smaller subproblems. These subproblems are staged such that each stage involves exactly one optimizing variable. The computations at the different stages are linked by recursive computations such that an optimal solution to the entire problem is reached when the last stage is solved. Indeed, the optimal path problem we are interested in falls into this class. It is well known (Waterman, 1989) that if $\mathbf{a}=a_1a_2,\ldots,a_m$, $\mathbf{b}=b_1b_2,\ldots,b_n$, the $E_{\min}(i,j)$ (the minimum cost for an optimal path from (0, 0) to (i,j)) can be comptued from the following recurrence relation:

$$E_{\min}(i,j)=\min\{E_{\min}(i-1,j)+\epsilon(a_i,\phi),$$
$$E_{\min}(i-1,j-1)+\epsilon(a_i,b_j),$$
$$E_{\min}(i,j-1)+\epsilon(\phi,b_j)\}, \quad (2)$$

with the boundary conditions:

$$E_{\min}(0,0)=0,\ E_{\min}(0,j)=\sum_{k=1}^{j}\epsilon(\phi,b_k),$$

and

$$E_{\min}(i,0)=\sum_{k=1}^{i}\epsilon(a_k,\phi),$$

where ϵ is the cost function mentioned before. This equation is at the heart of the DP method which allows one to iteratively compute the full minimum cost matrix $E_{\min}(m,n)$, and to find an optimal path. Again, for the sake of later comparison, we provide an explicit example. For simplicity, we assume that $a_1=A$, $a_2=C$, $b_1=A$, $b_2=T$, $b_3=C$ and $\epsilon(\phi,b)=\epsilon(a,\phi)=0$, $\epsilon(a,b)=-1$ if $a\neq b$, $=-2$ if $a=b$. In Fig. 2, all the details of a DP alignment of **a**, **b** are shown.

There have been many variations and generalizations upon the main theme of the DP method. One direction is to try to improve the efficiency of algorithms (Myres, 1991), or even sometimes compromise the optimality for speed (Fickett, 1984); another direction is to enlarge the capability for wider applications [including k-indels (Gotch, 1982) and/or inversions (Schöniger & Waterman, 1992), incorporating statistical significant analysis (Arriatia *et al*., 1985), adaptation to consensus alignment of multiple sequences (Waterman, 1986) or database search (E. W. Myers, unpublished data), etc).

Of course, the heart of the problem is how to choose the right cost function $E(\mathbf{a},\mathbf{b})$ for the application in hand; this is the most difficult part, and it requires a deep understanding of biology to be useful. The key is to understand the problem of *nonuniqueness* in the DP method: (i) different choices of the cost function can give the same optimal alignment (which leads to the problem of parametric decomposition); (ii) for a given cost function, there maybe several optimal alignments which all have the same $E_{\min}$. The first means, mathematically, that the optimal solution is invariant under some symmetry transformations in the space of cost functions; the second is called the degeneracy of the solution set. They are of course inter-related. Nonuniqueness reflects, to some extent, the incompletness of our knowledge about the biological system at hand. In practice, a meaningful alignment is usually expected to be somewhat close to, but maybe *not* exactly in the optimal set for a given cost function (Waterman, 1989). However, one can often take the advantage of the nonuniqueness property by starting off with a simple cost function and adjust the degeneracy of the solution such that it contains the biologically significant alignment. This can be followed by fine-tuning the cost function, sometimes by adding more non-local terms (the correlation terms which cannot be written in a simple additive form) in order to break the degeneracy (Fitch & Smith, 1983), where one hopes to be able to narrow down the optimal solution to the desired alignment. (As always, identifying a desirable alignment that has biological significance requires additional information.) To overcome the potential of missing significant alignments, more recent versions (Waterman, 1983) of DP method also include a cut-off parameter which specifies a neighborhood of the optimal solution (this neighborhood consists of paths with total cost not more than the cut-off parameter above $E_{\min}$), and one hopes the correct alignment maybe found within this neighborhood.

In the same spirit, we propose that every possible alignment is allowed, not just a neighborhood, albeit with a definite probability. This means the evolutionary distance (cost) has an uncertainty, although the minimum has the largest probability. The advantage is that we do not need a sharp cut-off and the DP problem can be thought of as a subproblem (its solution can be found by taking a limit).

Since the optimal path, which is deterministic, is replaced by random paths, and the solution requires a recurrence equation similar to eq (2) but without the awkward minimization procedures, we call our new method "random path analysis" (RPA). Readers familiar with statistical mechanics will recognize that RPA is an analogue of the partition function approach.

A similar partition function approach has been attempted earlier by Howell *et al*. (1980) and McCaskill (1990) in study of RNA secondary structure. But their structure problem is very different from (and more difficult than) the alignment problem: it corresponds to an unusual alignment problem where the length of each sequence is not fixed (a base can jump from one sequence to another) and it could also be complicated by knotted configurations which have no counter part in alignment problems. Furthermore, instead of a general solution, McCaskill was interested in a reduced description of the pair binding probabilities and in approximations that would allow further reduction of computational complexity of the algorithm. We, on the other hand, concentrate on the alignment problem and try to explore the full algebraic structure of DP in its connection to RPA theoretically, but also suggests some interesting applications.

2. Theoretical Formulation of the RPA Method for Sequence Alignments

The setup of RPA is the same as that of DP. Given a pair of sequences **a** and **b**, we have the directed $m+n$ lattice $G_{\mathbf{a},\mathbf{b}}$. A random particle makes an independent jump across a link γ with a probability proportional to

$$e^{u-\beta\epsilon(\gamma)}, \tag{3}$$

where ϵ is the cost function and $\beta>0$, u are real parameters (their meaning will be explained later). If we define a partition function $Z(m, n)$ as

$$Z(m\,n)=\sum_{\Gamma}\prod_{\gamma\in\Gamma} e^{u-\beta\epsilon(\gamma)}=\sum_{\Gamma} e^{|\Gamma|u-\beta E(\Gamma)}, \tag{4}$$

where Γ is a directed path from (0, 0) to (m, n), $|\Gamma|$ is the length (measured by number of the links) of the path and $E(\Gamma)$ is the total cost of the path, then the probability P of the path Γ is given by

$$P(\Gamma)=\frac{e^{|\Gamma|u-\beta E(\Gamma)}}{Z}, \tag{5}$$

which is, of course, normalized i.e.

$$\sum_{\Gamma} P(\Gamma)=1. \tag{6}$$

All the statistical information about the paths (hence the alignments) is contained in Z. For example, all the moments of the cost

$$\overline{E^k}=\frac{1}{Z}\frac{d^k}{d(-\beta)^k}Z,$$

(a bar designates an average by the probability P) and all the moments of the length distribution

$$\overline{|\Gamma|^k}=\frac{1}{Z}\frac{d^k}{du^k}Z,$$

can be obtained by simple differentiations. The parameter β is related to the average cost scale

$$\bar{E}=\frac{1}{Z}\frac{d}{d(-\beta)}Z, \tag{7}$$

and u is related to the averge length scale

$$\overline{|\Gamma|}=\frac{1}{Z}\frac{d}{du}Z. \tag{8}$$

In particular, all the results of the classical DP can be obtained by taking the limit $\beta\to\infty$. For example: the minimum cost matrix

$$E_{\min}(i,j)=\lim_{\beta\to\infty}\bar{E}(i,j), \tag{9}$$

where

$$\bar{E}(i,j)=\frac{d}{d(\beta)}F(i,j),$$

and the "free energy" of the random paths is defined by

$$F(i,j)=-\ln Z(i,j). \tag{10}$$

Or equivalently,

$$E_{\min}(i,j)=\lim_{\beta\to\infty}\frac{F(i,j)}{\beta}.$$

Once $E_{\min}(i,j)$ is known, back-tracking for an optimal path can be carried out just as before. The nice thing about RPA is that it contains much more information and $Z(i,j)$, which is at the heart of the RPA method as $E_{\min}(i,j)$ is for DP, can be solved exactly in a compact form. Although $E_{\min}$ is discontinuous in the cost parameters, F is smooth.

3. Algebraic Solution of $Z(i,j)$

Recall that, in the DP method, one has to solve a recurrence equation [eq (2)] for $E_{\min}(i,j)$. In the RPA method, we have to solve a similar difference equation

$$Z(i,j)=Z(i-1,j)e^{u-\beta\epsilon(a_{i-1},\phi)}+Z(i-1,j-1)\times e^{u-\beta\epsilon(a_i,b_j)}+Z(i,j-1)e^{u-\beta\epsilon(\phi,b_{j-1})} \tag{11}$$

for $Z(i,j)$. Here the Markov property is guaranteed as usual by the locality of the cost functions. It is the elimination of the optimization procedure that gives rise to the hope for an analytical solution to the problem. To avoid immaterial complications, we shall only derive and solve this equation for the case

where $u=0$ and $\epsilon(a, \phi)=\epsilon(\phi, b)=g$, a constant (this case corresponds to many applications). The extension to the more general case presents no more difficulties.

There exists an important symmetry in the probability $P(\Gamma)$ defined in Eqn (5). If we add a constant c both to $\epsilon(a, \phi)$ and to $\epsilon(\phi, b)$, at the same time, we also add 2c to $\epsilon(a, b)$, then P is invariant (i.e. P does not change its value). This is because under these operations, every "Boltzman factor", $\exp(-\beta E)$, gets multiplied by a same constant factor

$$\begin{aligned} &\mathrm{e}^{-\beta[\text{number of indels } +2(\text{number of substitutions})]} \\ &=\mathrm{e}^{-\beta[\text{total number of the sequence elements}]} \\ &=\mathrm{e}^{-\beta(m+n)c}, \end{aligned}$$

which in turn get cancelled out from both the numerator and the denominator in eqn (5) (One should notice that u can not be transformed away, because each alignment (path) may have a different length). This very symmetry dictates that the equation for Z must be homogeneous (we shall see the equation is also linear). This symmetry also implies that if we assume the cost for an indel (a horizontal or vertical jump) is a constant, then we can set this constant to be zero by adjusting a constant in the definition of the substitution cost $\epsilon(a, b)$ (a diagonal jump).

The derivation of the recurrence equation is simple. Remember the definition [eqn (4)], $Z(i, j)$ is a sum of product terms over all possible directed paths from $(0, 0)$ to (i, j). All of these paths have either passed through site $(i-1, j)$ followed by a down link, or through $(i-1, j-1)$ follwed by a diagonal link, or though $(i, j-1)$ followed by a right link. This explains the three terms in the equation [eqn (11)]. If the gap cost g is a constant, by the symmetry mentioned above, we may set $g=0$. We have the folloiwng recurrence equation

$$\begin{aligned} Z(i, j)=&AZ(i-1, j) \\ &+B_{ij}Z(i-1, j-1)+AZ(i, j-1), \quad (12) \end{aligned}$$

with the boundary conditions

$$Z(i, 0)=Z(0, i)=A^i,$$

where

$$A\equiv \mathrm{e}^u, \quad B_{ij}=\mathrm{e}^{u-\beta\epsilon(a_i, b_j)}.$$

As an interesting problem, mathematicians have studied this equation for some special cases (unfortunately less biologically interesting ones). We point out that these special results may also be obtained using a simpler generating functional approach. For example, with $B_{i, j}=B$ (i.e. all the diagonal jumps, alias for substitutions, have the same weight), a constant, the case when $A=1$, $Z(i, j)$ was solved by Laquer (1981) using combinatorial identities (the case when $A=B=1$ was previously solved by Stanton & Cowan, 1970).

Laquer obtained

$$Z(m, n)=\sum_{k=0}^{\min}\binom{m}{k}\binom{n}{k}(1+B)^k, \quad (13)$$

where

$$\min\equiv\min(m, n).$$

Here $Z(m, n)$ is symmetric (i.e. $Z(m, n)=Z(n, m)$) because $B_{i, j}$ is. When $B=0$ (no diagonal jump is allowed), $Z(m, n)=(m+n, m)$. When $B=1$ (i.e. $\beta\to 0$, all jumps have equal weight), $Z(m, n)$ becomes the total number of possible paths which has the following asymptotics Laquer (1981)

$$Z(n, n)\to(1+\sqrt{2})^{2n+1}\sqrt{n}, \quad \text{as } n\to\infty,$$

or

$$\frac{\ln Z(n, n)}{2n}\to\text{cont.} \quad \text{with cont.}=\ln(1+\sqrt{2}).$$

The fact that Z grows exponentially with $2n$ is obvious because there are $2n$ independent variables, the cont. exponent characterizes the geometry of the lattice. If we substitute $Z(i, j)\sim t^i s^j$ (for i, j large) into

$$Z(i, j)=Z(i-1, j)+Z(i-1; j-1)+Z(i, j-1),$$

and then set $s=t$, we find

$$t^2-2t-1=0, \quad \text{or } t=1+\sqrt{2}, \quad (14)$$

which is just the constant that controls the growth of $Z(n, n)$.

We can easily solve the equation for arbitrary constant A, B by using the generating function (by analogy to a random walk problem):

$$G(t, s)=\sum_{i, j=0}^{\infty}Z(i, j)t^i s^j, \quad (15)$$

assuming this is convergent for small t, s, then from the equation

$$\begin{aligned} Z(i, j)=&AZ(i-1, j)+BZ(i-1, j-1) \\ &+AZ(i, j-1) \end{aligned}$$

and $Z(i, 0)=Z(0, i)=A^i$, we obtain

$$G(t, s)=\frac{\dfrac{1-t}{1-At}+\dfrac{1-s}{1-As}-1}{1-t-s-BTs}. \quad (16)$$

Indeed, $G(t, s)$ is analytic for small t, s. Therefore, $Z(i, j)$ is just the expansion coefficient of $g(t, s)$ [eqn (15)] upon expanding in power series. If $A=1$, $G(t, s)=(1-t-s-bts)^{-1}$ which gives the exact solution [eqn (13)]. If we then set $b=1$, $s=t$, the singularity of G is given exactly by eqn (14). The point is that the growth property of Z can be easily seen from the singularities of G.

Now let us return to the general case where $B_{i,j}$ is arbitrary, (we only consider the case where $A=1$, generalizations cause no difficulty). The recurrence equation is

$$Z(i, j)=Z(i-1, j)+B_{i,j}Z(i-1, j-1)+Z(i, j-1), \quad (17)$$

with the boundary conditions: $Z(i, 0)=Z(0, i)=1$. Then the general solution with arbitrary $B_{i,j}$ (i.e. arbitrary cost functions) can be given exactly as

$$Z(m, n)=1+\sum_{i_1}\sum_{j_1}\hat{B}_{i_1,j_1}+\sum_{i_1<i_2}\sum_{j_1<j_2}\prod_{k=1}^{2}\hat{B}_{i_k,j_k}+\sum_{i_1<i_2<i_3}\sum_{j_1<j_2\in j_3}\prod_{k=1}^{3}\hat{B}_{i_k,j_k}+\cdots+\sum_{i_1<i_2<\cdots<i_{\min}}\sum_{j_1<j_2<\cdots<j_{\min}}\prod_{k=1}^{\min}\hat{B}_{i_k,j_k} \quad (18)$$

where we have introduced the convenient variables

$$\hat{B}_{i,j}\equiv 1+B_{i,j},$$

and the summation indices i_k and j_k run from 1 to m and from 1 to n, respectively. The solution may be proved by induction, which we have omitted here. Equation 18 reduces to Laquer's result [eqn (13)] when $\hat{B}_{i,j}$ is a constant. The structure of the solution is remarkably transparent: the sum over paths is organized in number of diagonal links (indexed by k). A graphical representation of the solution can be written in the following way. Denoting 1 by dot "•" and $\hat{B}_{i,j}$ by a link "—", then the sum in eqn (18) may be symbolically written as a graphic sum:

$$\bullet+i_1-j_1+\begin{matrix}i_1-j_1\\ i_2-j_2\end{matrix}+\cdots+\begin{matrix}i_2-j_1\\ i_2-j_2\\ \vdots\\ i_{\min}-j_{\min}\end{matrix}$$

Of course, one has to carry out the summation of all the indices accordingly.

We conclude this section by working out the example discussed above (Fig. 2). Since $m=2$, $n=3$, $\mathbf{a}=AC$, $\mathbf{b}=ATC$, we have

$$Z(2, 3)=\bullet+i_1-j_1+\begin{matrix}i_1-j_1\\ i_2-j_2\end{matrix}.$$

$$=1+\sum_{i_1=1}^{2}\sum_{j_1=1}^{3}\hat{B}_{i_1,j_1}+\hat{B}_{1,1}\hat{B}_{2,2}+\hat{B}_{1,1}\hat{B}_{2,3}+\hat{B}_{1,2}\hat{B}_{2,3}.$$

This provides the complete information about the paths. For instance, the probability that a path $\Gamma=(0, 0)\to(1, 1)\to(2, 2)\to(2, 3)$ is given by $P(\Gamma)=\hat{B}_{1,1}\hat{B}_{2,2}$. Because $\hat{B}_{i,j}=1+B_{i,j}=1+\exp[\beta|\epsilon(a_i, b_j)|]$, if one is only interested in the optimal path (when $\beta\to\infty$), from the cost matrix

$$E(\mathbf{a}, \mathbf{b})=\begin{pmatrix}-2 & -1 & -1\\ -1 & -1 & -2\end{pmatrix},$$

it is clear that the term $\hat{B}_{1,1}\hat{B}_{2,3}$ is the dominant one, which corresponds to the optimal path $(0, 0)\to(1, 1)\to(1, 2)\to(2, 3)$. The minimum cost is

$$E_{\min}=\lim_{\beta\to\infty}-\frac{\mathrm{d}}{\mathrm{d}\beta}\ln Z(2, 3),$$

$$=-\lim_{\beta\to\infty}\frac{\mathrm{d}}{\mathrm{d}\beta}\ln(\hat{B}_{1,1}\hat{B}_{2,3}),$$

$$=-\frac{\mathrm{d}}{\mathrm{d}\beta}(4\beta)=-4.$$

It gives, as expected, the same result as obtained by DP (Fig. 2).

4. Explicit Examples and Parametric Alignment Problem

The algebraic solution of RPA is necessarily complicated because it has to be valid for all parameteric (cost) values for arbitrary pair of sequences. A real numerical computation has to be done by Monte Carlo methods. This computational disadvantage can become an advantage if one is interested in analytical structure and parametric dependence. For example, in the parametric alignment problem (Waterman *et al*. 1992), one needs to know what is expected $E_{\min}$ is for random sequences and how does it depend on cost parameters. In RPA, the expected $E_{\min}$ should be replaced by F averaged over all random sequence pairs. We use the explicit solutions

for small systems ($n=m=1, 2, 3$) to illustrate how it is done by RPA. Without loss of generality, we assume binary sequences (represented by a_i, $b_j=\pm 1$).

A Boltzman factor $B_{i,j}$ for such binary sequences can always be written as

$$B_{i,j}=\alpha+\gamma a_i b_j,$$

for the usual cost parameters: 1 for a match, $-\mu$ for a mismatch and $-\delta$ for an indel, after translating the "energy" scale by $c=\delta$ (as mentioned above) the cost parameters becomes: $1+2\delta$ for a match, $-\mu+2\delta$ for a mismatch and 0 for an indel.

$$\alpha=e^{\beta((1-\mu)/2+2\delta)}\cosh\beta\frac{1+\mu}{2},$$

$$\gamma=e^{\beta((1-\mu/2+2\delta)}\sinh\beta\frac{1+\mu}{2}. \tag{19}$$

The general solution (18) can be written as

$$Z(n,n)=\sum_{k=0}^{n}\sum_{I_{k,n},J_{k,n}}\prod_{m=1}^{k}(1+B_{i_m,j_m})$$

$$=\sum_{k=0}^{n}\sum_{I_{k,n},J_{k,n}}\prod_{t=0}^{k}\sum_{M_{t,k}}\prod_{s=1}^{t}a_{i_{ms}}b_{j_{ms}}, \tag{20}$$

where $I_{k,n}$ means all possible integers $i_1<i_2<\cdots<i_k$ from 1 to n ($J_{k,n}$ and $M_{t,k}$ are defined similarly).

4.1. $n=m=1$

This is the simplest case,

$$Z=2+\alpha+\gamma\alpha_1 b_1,$$

and

$$-F=\ln Z=\tfrac{1}{2}\ln[(2+\alpha)^2-\gamma^2]+\frac{a_1b_1}{2}\ln\frac{2+\alpha+\gamma}{2+\alpha-\gamma}.$$

If p is the expected value for a_1 and b_1, since a and b are assumed to be independent random variables, the expected "free energy" $F_{\exp}$ is given by the same solution with a_1, b_1 replaced by p. For equal probable distributions (i.e. $p=0$), $F_{\exp}$ is given by the first term. With the usual parameterization (19),

$$-F_{\exp}=\tfrac{1}{2}\ln(2+e^{\beta(1+2\delta)})(2+e^{\beta(-\mu+2\delta)}).$$

For the expected $E_{\min}$, we let $\beta\to\infty$ and obtain: $-(2\delta+1/2)$ if $\mu>2\delta$; and $-(2\delta+1/2-2\delta-\mu/2)$.

4.2. $n=m=2$

This is the smallest system where one can see the correlation among different paths.

$$Z=x_1+x_2(a_1b_1+a_2b_2)+x_3(a_1b_2+a_2b_1)+x_4a_1a_2b_1b_2,$$

where $x_1=6+6\alpha+\alpha^2$, $x_2=\gamma(2+\alpha)$, $x_3=\gamma$, $x_4=\gamma^2$. The explicit solution for the "free energy" is given by

$$-F=y_1+y_2(a_1b_1+a_2b_2)+y_3(a_1b_2+a_2b_1)+y_4(a_1a_2+b_1b_2)+y_5a_1a_2b_1b_2,$$

where

$$y_1=\tfrac{1}{8}\ln\frac{[(3+\alpha+\gamma)^2-3][(3+\alpha-\gamma)^2-3][(1+\alpha+\gamma)^2+5+4\alpha][(1+\alpha-\gamma)^2+5+4\alpha]}{(6+6\alpha+\alpha^2+\gamma^2)^{-4}}$$

$$y_2=\tfrac{1}{8}\ln\frac{[(3+\alpha+\gamma)^2-3][(1+\alpha+\gamma)^2+5+4\alpha]}{[(3+\alpha-\gamma)^2-3][(1+\alpha-\gamma)^2+5+4\alpha]}$$

$$y_3=\tfrac{1}{8}\ln\frac{[(3+\alpha+\gamma)^2-3][(1+\alpha-\gamma)^2+5+4\alpha]}{[(1+\alpha+\gamma)^2+5+4\alpha][(3+\alpha-\gamma)^2-3]}$$

$$y_4=\tfrac{1}{8}\ln\frac{[(3+\alpha+\gamma)^2-3][(3+\alpha-\gamma)^2-3]}{[(1+\alpha+\gamma)^2+5+4\alpha][(1+\alpha-\gamma)^2+5+4\alpha]}$$

$$y_5=\tfrac{1}{8}\ln\frac{[(3+\alpha+\gamma)^2-3][(3+\alpha-\gamma)^3-3][(1+\alpha+\gamma)^5+5+4\alpha][(1+\alpha-\gamma)^2+5+4\alpha]}{(6+6\alpha+\alpha^2-\gamma^2)^4}$$

Comparing F with Z, one sees immediately the additional y_4 term which signals the correlation, because, a_1a_2 must have come from different paths (i.e. $=(a_1b_j)^*(a_2b_j)$, remember that $a_i^2=b_j^2=1$), *Path interference will be a general feature for longer sequence alignments*. Again, if $p=0$, we have $-F_{\exp}=y_1$. If we let $\beta\to\infty$ (α, γ become large, see (19)), we have

$$-E_{\min,\exp}\sim\tfrac{1}{8}\ln\alpha^8[1+(\alpha-\gamma)^2]^5[\alpha+(\alpha-\gamma)^2].$$

Depending which of the two terms in each square bracket is dominant, we obtain

$$-E_{\min,\exp}=\begin{cases}\frac{9}{8}(1+2\delta) & \text{if } 2\delta<\mu,\\ \frac{9}{8}(1+2\delta)+\frac{6}{8}(-\mu+2\delta) & \text{if } \delta-\frac{1}{2}<\mu<2\delta,\\ 1-\mu+4\delta & \text{otherwise.}\end{cases}$$

This optimal result could be calculated using probability theory. RPA does the counting systematically. For suboptimal paths (finite β) or unequal distribution ($p\neq 0$), the full expression for F is needed for parametric studies.

4.3. $n=m=3$

Finally, with the assistance of *Mathematica*†, we present the solution for 3×3 (using a symbolic manipulation program, one can obtain solution for longer sequences, but 3×3 is sufficient here for illustrating the general concept). To simplify the notation, we use $r_{i_1i_2},\ldots,j_1j_2,\ldots$ to denote $a_{i_1}a_{i_2},\ldots, b_{j_1}b_{j_2},\ldots$,

$$\begin{aligned}Z=&x_1\times x_2(r_{1,1}+r_{3,3})+x_3r_{2,2}+x_4(r_{1,2}+r_{2,1}\\&+r_{2,3}+r_{3,2})+x_5(r_{1,3}+r_{3,1})+x_6(r_{1,1}r_{2,2}\\&+r_{2,2}r_{3,3}+r_{1,1}r_{3,3})+x_7(r_{1,1}r_{2,3}+r_{1,1}r_{3,2}\\&+r_{1,2}r_{3,3}+r_{2,1}r_{3,3}+r_{1,2}r_{2,3}+r_{2,1}r_{3,2})\\&+x_8r_{1,1}r_{2,2}r_{3,3},\end{aligned}$$

$$E_{\min,\exp}=\begin{cases}-\frac{58}{32}[(1+2\delta) & \text{if } \mu>2\delta\\ -\frac{1}{32}[58(1+2\delta)+29(-\mu+2\delta)] & \text{if } \delta-\frac{1}{2}<\mu<2\delta\\ -\frac{1}{32}[50(1+2\delta)+45(-\mu+2\delta)] & \text{if } \frac{2}{3}(\delta-1)<\mu<\delta-\frac{1}{2}\\ -\frac{48}{32}(1-\mu+4\delta) & \text{otherwise.}\end{cases}$$

where (expressions are written in a form suggesting its generalization for longer sequences)

$$x_1=\sum_{k=0}^{3}\binom{3}{k}^2(1+\alpha)^k$$

$$x_2=b\sum_{k=0}^{2}\binom{2}{k}^2(1+\alpha)^k$$

$$x_3=b\sum_{k=0}^{1}\left[\binom{1}{k}^2+1\right](1+\alpha)^k$$

$$x_4=b\sum_{k=0}^{1}\left[\binom{2}{k}\binom{1}{k}(1+\alpha)^k\right.$$

$$x_5=b$$

$$x_6=b^2\sum_{k=0}^{1}\binom{1}{k}^2(1+\alpha)^k$$

$$x_7=b^2$$

$$x_8=b^3.$$

And the "free energy" F is given by

$$\begin{aligned}-F=&y_1+y_2(r_{1,1}+r_{3,3})+y_3r_{2,2}+y_4(r_{1,2}+r_{2,1}+r_{2,3}\\&+r_{3,2})+y_5(r_{1,3}+r_{3,1})+y_6(r_{12,}+r_{,12}+r_{23,}\\&+r_{,23})+y_7(r_{13,}+r_{,13})+y_8(r_{12,12}+r_{23,23})\\&+y_9r_{13,13}+y_{10}(r_{12,13}+r_{13,12}+r_{13,23}+r_{23,13})\\&+y_{11}(r_{12,23}+r_{23,12})+y_{12}(r_{123,1}+r_{1,123}+r_{123,3}\\&+r_{3,123})+y_{13}(r_{123,2}+r_{2,123})+y_{14}r_{123,123}.\end{aligned}$$

The expressions for the coefficients (the y's) are listed in the Appendix. For $p=0$, $-F_{\exp}=y_1$. As $\beta\to\infty$,

$$\begin{aligned}-F_{\exp}\sim&\tfrac{1}{32}\ln(\alpha^{48}[1+(\alpha-\gamma)]^{15}[1+(\alpha-\gamma)^2]^6\\&\times[\alpha+(\alpha-\gamma)^2]^6[\alpha+\alpha(\alpha-\gamma)\\&+(\alpha-\gamma)^3]^2[\alpha^2+(\alpha-\gamma)^3]),\end{aligned}$$

which, in the limit, has four solutions:

5. Concrete Numerical Example of Path Probabilities

To see how the path (alignment) probabilities change as β is increased, we take a concrete example of two sequences from *Escherichia coli tRNA* sequences (sequence A is from threonine tRNA of Genbank name ECOTRTACU and sequence B from valine tRNA of Genbank name ECOTRV1). The following alignment is optimal, which we call Γ_1(we use "—" to indicate a gap).

Γ_1:

(A) ACTA-CTGCCCTAGCTTGGCGGCTGGGGGAGGAA
(B) ACTATCCGTCTAAGCTTGACGGCTGGAGTGGGAA.

†*Mathematica* is a registered trademark of Wolfram Research, Inc.

TABLE 1

β	0.0	0.7	1.4	2.1	2.8	3.5	4.2	4.9	5.6	6.3
$P(\Gamma_1)$	2.3×10^{-25}	5.8×10^{-5}	5.8×10^{-2}	0.26	0.49	0.68	0.82	0.90	0.95	0.97
$P(\Gamma_2)$	2.3×10^{-25}	2.0×10^{-10}	6.6×10^{-13}	1.0×10^{-17}	6.4×10^{-23}	3.0×10^{-28}	1.2×10^{-33}	4.5×10^{-39}	1.6×10^{-44}	5.5×10^{-50}
$P(\Gamma_3)$	2.3×10^{-25}	2.7×10^{-15}	1.2×10^{-22}	2.6×10^{-32}	2.2×10^{-42}	1.4×10^{-52}	7.9×10^{-63}	4.0×10^{-73}	1.9×10^{-83}	9.2×10^{-94}
$F-\beta E(\Gamma_1)$	56.7	9.75	2.85	1.33	0.710	0.379	0.198	0.101	0.051	0.025

In the spirit of RPA, every alignment is possible with a specific probability. We use a fixed set of scoring parameters (for $a\neq b$):

$$\epsilon(a, a)=-1,\ \epsilon(a, b)=1,\ \epsilon(a, -)=\epsilon(-, a)=2,$$

and compare the probability of the alignment $Gamma_1$ with that of Γ_2 and Γ_3 defined below

Γ_2:

(A) ACTACTGCCCTAGCTTGGCGGCTGGGGGAGGAA-
(B) ACTATCCGTCTAAGCTTGACGGCTGGAGTGGGAA,

and

Γ_3:

(A) ACTACTGCCCTAGCTTGGCGGCTGGGGGAGGAA- - -
(B) - -ACTATCCGTCTAAGCTTGACGGCTGGAGTGGGAA.

In Table 1, we have listed numerical values of the path probabilities and the difference of the "free" energy with the energy of the optimal alignment Γ_1.

We see clearly that, at zero β, all paths (alignments) have equal probability. As β increases, the path costing less energy become more probable. As β become large, the probability for optimal path will approach to one and the "free" energy will approach to the lowest energy.

6. General Features and Practical Applications of RPA

In the above, we have formulated the theory of RPA and shown its relation to DP and its usage in parametric studies. This reltion is a general one, i.e., how to interpret the meaning of *optimality*. What is optimal for a set is often suboptimal for a subset. This is particularly true when the interaction among different subsets are strong. Presumably, if the environment was fixed, optimality would be eventually reached by any system obeying the laws of physics and chemistry. Therefore, to understand nature better, it is important to find out the natural boundaries within the biological hierarchy, which means one has to include a part of the environment into the system so that the new system interacts with the new environment weakly. In the context of sequence alignment, one often wants to find some optimal alignment for a well-characterized group of sequences even though a particular pair may not be optimally aligned. In other words, a cost function should characterize the statistical properties of a known group of related sequences. The best alignment of a particular pair of sequences from a related group need not be optimal. In this situation, a statistical approach like RPA is very relevant. The arameters (like u, β of RPA) in the probability of a possible alignment should no longer be treated as free, they ought to be related to the characteristics of the group (see Fig. 3, u is related to the average length, β to the average cost of the aligned group, for instance).

Another advantage of a statistical approach is that it can help us understand the inverse problem: knowing an optimal alignment of a group of sequences, what can one infer about the corresponding cost function (its functional form as well as its best parameters)? The inverse problem completes the feedback process, which is necessary for scientific progress. With the help of RPA, one starts with a model cost function and fits the parameters by examining statistical properties of a group of aligned sequences. With more information about the alignment from new experiments, one proceeds to modify the cost function in order to better explain the data. In this sense, selecting appropriately related sequences is equivalent to picking a good cost function. There have been dynamic approaches to sequence alignment problems, notably the "maximum likelihood alignment" pioneered by Bishop & Thompson (1986) and extended by Thorne *et al*. (1991, 1991). If we reinterpret $B_{i,j}=e^{u-\beta\epsilon(a_i,b_j)}$ as the transition matrix, the Z function will be proportional to the likelihood function in the "maximum likelihood" method, and the choice of $u=$*average length*, $\beta=$*average cost* will become the maximal likelihood estimates.

Finally, we mention some potential applications of RPA other than its conceptual merits.

(i) RPA opens up the possibility of using Monte Carlo simulations to study the sequence alignment problem, because random paths can be generated by

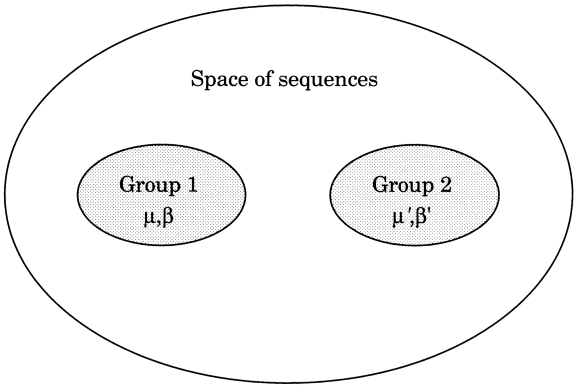

FIG. 3. Group characterized by some parameters.

stochastic particles walking on a lattice according to specified jump probabilities. With parallel processors, this type of approach should complement the traditional deterministic approaches to explore the neighborhood of an optimal alignment. RPA can provide much more statistical information. By choosing a very large β (depending how close to the optimal one desires to look into in order to find a biologically meaningful alignment), the probabilities are guaranteed to be sharply peaked around the optimal path(s). This is especially so if one wants to study the problem of random sequence alignments (i.e. the aligned sequence pair are regarded as random sequences) (Karlin & Altschul, 1990) which is essentially equivalent to the spin-glass problem in physics. Simulated annealing would be the only known efficient method to explore the complex structures.

(ii) RPA suggests a physical analogue such that sequence alignments could be done at the speed of light by an electro-optics system. We suggest that the lattice should be realized by a resistor network (each link is represented by an illuminable resistor), the resistance at each link is initialized according to the cost function. Let the electric current flow into the origin and drawn out off the end. Since the light intensity of an illuminated link reflects the passing electric current, the probability distribution function $P(\Gamma)$ can be immediately visualized from the brightness of the paths; the optimal path is represented by the brightest one. In cases where the brightest path is not obvious (i.e. many paths have similar brightness) by visual inspection, one may first eliminate the negative regions and then measure the total intensity (or, equivalently, total resistance) one path at a time (when all other paths are switched off by setting the corresponding resistance to infinity). If this process is automatable and the lighted paths are scored according to their intensity, the alignments ordered by their scores can be achieved fast.

(iii) With the help of the explicit structure of the solution, one could design better algorithms in parametric studies depending on the specific questions one would like to ask. If complete information of the distribution is required, "combinatorial explosion" (exponential growth of computational complexity with the sequence size) is unavoidable. If only a subset of the statistical information is desired, a more efficient computational scheme maybe achievable McCaskill (1990). If a particular symmetry of $E(\mathbf{a}, \mathbf{b})$ should arise or a more general statistical question is pursued, the RPA method and its algebraic solution can be a definite help. If one is only interested in the optimal path or $E_{\min}$, the standard DP algorithm is still the best choice.

We thank Michael Waterman for reading the manuscript and for bringing our attention to some references. We are grateful for support by NIH grant 1R01 HG00203–01A1 and DOE grant DE-FG02-91ER61190 to TGM, and partly by NIH grant 1K01 HG00010-01 to MQZ.

REFERENCES

ARRATIA, R., GORDON, L. & WATERMAN, M. S. (1985). An extreme value theory for sequence matching. *Ann. Stat.* **14,** 971.

BISHOP, M. J. & THOMPSON, E. A. (1986). Maximum likelihood alignment of DNA sequences. *J. molec. Biol.* **190,** 159.

BYERS, T. H., WATERMAN M. S. (1989). Determining all optimal and near-optimal solutions when solving shortest path problems by dynamic programming. *Oper. Res.* **32,** 1381.

FEYNMAN R. P. (1989). *Feynman Lectures on Physics*. Redwood City, CA: Addison-Wesley.

FICKETTT, J. W. (1984). Fast optimal alignment. *Nucleic Acids. Res.* **12,** 175.

FITCH, W. M. & SMITH, T. F. (1983). Optimal sequence alignments. *Proc. natn. Acad. Sci. U.S.A.* **80**, 1382.

GOTOH, O. (1982). An improved algorithm for matching biological sequences. *J. molec. Biol.*, **162,** 705.

HOWELL, J. A., SMITH, T.F. & WATERMAN M. S. (1980). Computation of generating functions for biological molecules. *SIAM J. appl. Math.* **39,** 119.

KARLIN, S. & ALTSCHUL, S. F. (1990). Methods for assessing the statistical significance of molecular sequence freatures by using general scoring schemes. *Proc. natn. Acad. Sci. U.S.A.* **87,** 2264.

LAQUER, H.T. (1981). Asymptotic limits fora two-dimensional recursion. *Stud. appl. Math.* **64,** 271.

MCCASKILL, J. S. (1990). The equilibrium partition function and base pair binding probabilities for RMNA secondary structure. *Biopolymers*, **29,** 1105.

MILLER, W. & MYERS, E. W. (1988) Sequence comparison with concave weighting functions. *Bull. math. Biol.* **50,** 97.

MYERS, G. W. (1991). "Sequence comparison: algorithms", lecture note given at a workshop of Cold Spring Harbor Laboratory, New York.

NEEDLEMAN, S. B. & WUNSCH, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequences of two proteins. *J. molec. biol* **48,** 444.

PEARSON, W. R. (1990). Rapid and sensitive sequence comparison with FASTP and FASTA. *Met. Enz.* **183,** 63.

SCHÖNIGER, M. & WATERMAN M. S. (1992). A local algorithm for DNA sequences alignment with inversions, *Bull. math. Biol.* **54,** 152.
SELLERS, P. H. (1974). Theory and computation of evolutionary distance. *SIAM J. appl. Math.* **26,** 787.
SMITH, T. F. & WATERMAN, M. S. (1981). Comparison of biosequences, *Adv. appl. Math.* **2,** 482.
SMITH, T. F., WATERMAN, M. S. & FITCH, W. M. (1981). Comparative biosequence metrics. *J. molec. Evol.* **18,** 38.
STANTON R. G. & COWAN, D. D. (1970). Note on a "square functional" equation, *SIAM Rev.* **12,** 277.
THORNE, J. L., KISHINO, H. & FELSENSTEIN, J. (1991). An evolutionary model for maximum likelihood alignment of DNA sequences. *J. molec. Evol.* **33,** 114.
THORNE, J. L., KISHINO, H. & FELSENSTEIN, J. (1992). Inching toward reality: an improved likelihood model of sequence evolution. *J. molec. Evol.* **34,** 3.
WATERMAN, M. S. (1984). Efficient sequence alignment algorithm. *J. theor. Biol.* **108,** 333.
WATERMAN, M. S. (1986). Multiple sequence alignment by consensus. *Nucleic Acids Res.* **14,** 9095.
WATERMAN, M. S., SMITH, T. F. & BEYER, W. A. (1976). Some biological sequence metrices. *Adv. Math.* **20,** 367.
WATERMAN, M. S. (1989). Sequence alignments. In: *Mathematical Methods for DNA Sequences* (Waterman, M. S. ed.) Boca Raton, FL., CRC Press, (1989).
WATERMAN, M. S. (1983). Sequence alignment in the neighborhood of the optimum with general applications to dynamic programming. *Proc. natn. Acad. Sci. USA* **80,** 3123.
WATERMAN, M. S., EGGERT, M. & LANDER E. (1992) *Proc. natn. Acad. Sci. U.S.A.* **89,** 6090.

APPENDIX

$$y_1 = \tfrac{1}{32}\ln(w_1 w_2^6 w_3^6 w_4 w_5^2 w_6^4 w_7^2 w_8^2 w_9^2 w_{10}^4 w_{11} w_{12})$$

$$y_2 = \tfrac{1}{32}\ln\frac{w_1 w_2^2 w_5^2 w_8^2 w_{10}^4 w_{11}}{w_3^2 w_4 w_7^2 w_9^2 w_{12}}$$

$$y_3 = \tfrac{1}{32}\ln\frac{w_1 w_2^2 w_5^2 w_7^2 w_{10}^4 w_{11}}{w_3^2 w_4 w_6^2 w_8^2 w_9^2 w_{12}}$$

$$y_4 = \tfrac{1}{32}\ln\frac{w_1 w_2^2 w_6^2 w_{12}}{w_3^2 w_4 w_{10}^2 w_{11}}$$

$$y_5 = \tfrac{1}{32}\ln\frac{w_1 w_2^2 w_7^2 w_9^2 w_{11}}{w_3^2 w_4 w_5^2 w_8^2 w_{12}}$$

$$y_6 = \tfrac{1}{32}\ln\frac{w_1 w_2^2 w_3^2 w_4^2}{w_6^2 w_{10}^2 w_{11} w_{12}}$$

$$y_7 = \tfrac{1}{32}\ln\frac{w_1 w_2^2 w_4 w_5^2 w_{11} w_{12}}{w_5^2 w_7^2 w_8^2 w_9^2}$$

$$y_8 = \tfrac{1}{32}\ln\frac{w_1 w_4 w_5^2 w_9^2 w_{11} w_{12}}{w_2^2 w_3^2 w_7^2 w_8^2}$$

$$y_9 = \tfrac{1}{32}\ln\frac{w_1 w_4 w_5^2 w_7^2 w_8^2 w_9^2 w_1 w_{12}}{w_2^2 w_3^2 w_6^4}$$

$$y_{10} = \tfrac{1}{32}\ln\frac{w_1 w_4 w_6^2 w_{10}^2}{w_2^2 w_3^2 w_{11} w_{12}}$$

$$y_{11} = -\tfrac{1}{32}\ln\frac{w_1 w_4 w_7^2 w_8^2 w_9^2 w_{11} w_{12}}{w_2^2 w_3^2 w_5^2 w_9^2}$$

$$y_{12} = \tfrac{1}{32}\ln\frac{w_1 w_3^2 w_{10}^2 w_{12}}{w_2^2 w_4 w_6^2 w_{11}}$$

$$y_{13} = \tfrac{1}{32}\ln\frac{w_1 w_3^2 w_8^2 w_9^2 w_{11}}{w_2^2 w_4 w_5^2 w_7^2 w_{12}}$$

$$y_{14} = \tfrac{1}{32}\ln\frac{w_1 w_3^6 w_5^2 w_6^4 w_7^2 w_{11}}{w_2^6 w_4 w_8^2 w_9^2 w_{10}^4 w_{12}},$$

where

$$w_1 = \sum_{k=0}^{3}\binom{3}{k}^2(1+\alpha+\gamma)^4$$

$$w_2 = 20+30\alpha+12\alpha^2+\alpha^3+10\gamma+8\alpha\gamma+\alpha^2\gamma-4\gamma^2 - \alpha\gamma^2-\gamma^3$$

$$w_3 = 20+30\alpha+12\alpha^2+\alpha^3-10\gamma-8\alpha\gamma-\alpha^2\gamma-4\gamma^2 - \alpha\gamma^2+\gamma^3$$

$$w_4 = \sum_{k=0}^{s}\binom{3}{k}^2(1+\alpha-\gamma)^k$$

$$w_5 = 20+30\alpha+12\alpha^2+\alpha^3+14\gamma+16\alpha\gamma+3\alpha^2\gamma + 4\gamma^2+3\alpha\gamma^2+\gamma^3$$

$$w_6 = 20+30\alpha+12\alpha^2+\alpha^3+2\gamma-\alpha^2\gamma-\alpha\gamma^2+\gamma^3$$

$$w_7 = 20+30\alpha+12\alpha^2+\alpha^3-6\gamma-8\alpha\gamma-\alpha^2\gamma - \alpha\gamma^2+\gamma^3$$

$$w_8 = 20+30\alpha+12\alpha^2+\alpha^3+6\gamma+8\alpha\gamma+\alpha^2\gamma - \alpha\gamma^2-\gamma^3$$

$$w_9 = 20+30\alpha+12\alpha^2+\alpha^3-14\gamma-16\alpha\gamma-3\alpha^2\gamma + 4\gamma^2+3\alpha\gamma^2-\gamma^3$$

$$w_{10} = 20+30\alpha+12\alpha^2+\alpha^3-2\gamma+\alpha^2\gamma-\alpha\gamma^2-\gamma^3$$

$$w_{11} = 20+30\alpha+12\alpha^2+\alpha^3+6\gamma+8\alpha\gamma+3\alpha^2\gamma+4\gamma^2 + 3\alpha\gamma^2+\gamma^3$$

$$w_{12} = 20+30\alpha+12\alpha^2+\alpha^3-6\alpha-8\alpha\gamma-3\alpha^2\gamma+4\gamma^2 + 3\alpha\gamma^2-\gamma^3.$$