The Early Introduction of Dynamic Programming into Computational Biology

David Sankoff
Centre de recherches mathématiques, Université de Montréal,
CP 6128 succursale Centre-Ville, Montréal, Québec H3C 3J7.
sankoff@ere.umontreal.ca

Running head: Early Introduction of Dynamic Programming

Keywords: alignment, multiple alignment, phylogeny, secondary structure

This paper is dedicated to the memory of Mark Kac

1 Introduction

In 1994–1995, DIMACS sponsored a theme year on computational biology. Among the numerous seminars and workshops was one organized by Alberto Apostolico and Raffaele Giancarlo, recapitulated in their 1998 paper, on the history and motivations for sequence comparison. In my participation in this event, I was led to consider some of the early interactions, at the Centre de recherches mathématiques (CRM) and elsewhere, in the field now known as computational biology. A short time earlier, I had read a 1989 paper by Walter Goad on the impact of Stanislaw Ulam in this field. The penultimate sentence in this article was a quote from Ulam himself "I started all this", which puzzled me greatly. As far as I knew, Ulam had no impact in the early development of the field, and while he gave talks on it at least from 1971 (cf Ulam, 1971), the distance he defined was already published (Levenshtein, 1965) and the problem he proposed had already been solved, for all intents and purposes, in the molecular biology literature (Needleman and Wunsch, 1970) and elsewhere (Vintzyuk, 1968). I had also read a joint interview of Ulam and Mark Kac by Mitchell Feigenbaum (1982), and this led me to reflect on this misperception on the part of Ulam (and of Goad), and to crystallize the realization that ironically, Kac, his colleague of many years had played a crucial, if indirect, role in the earliest development of the field, especially that associated with the Centre de recherches mathématiques (CRM). Despite the fact that Kac had no personal research interest in the field, his encouragement of a number of junior mathematicians, and his role, intentional or not, in bringing researchers together, recur as important influences in several aspects of computational biology, and explain why I dedicate this article to his memory.

In presenting some of these thoughts to the DIMACS workshop, focusing on the early 1970s, my understanding of this period was broadened by comments from a number of participants, particularly Jerrold Griggs and Pavel Pevzner, and clarified by the presentation immediately following, by Michael Waterman, who concentrated on the era starting in the late 1970s.

In this article, I will draw on my recollections of the earliest phases of the field to describe how certain fundamental ideas found their ways into the vernacular of the computational biologist.

2 Dynamic Programming for Sequence Comparison

In the late summer of 1971, the sequence comparison problem was brought to my attention by my friend Robert Cedergren, a biochemist interested in the evolution of ribonucleic acid sequences. In discussions with him, we formulated this as the longest common subsequence problem, though at the time we called it maximum matching. I realized that the application of mathematical induction led to a simple recurrence relation for solving the problem. We consider two sequences of length m and n of terms from any alphabet. If the two sequences are $a(1), \ldots, a(m)$ and $b(1), \ldots, b(n)$, then using the notation a_i for the prefix sequence $a(1), \ldots, a(i)$ and $M(i, j)$ for the longest common subsequence of a_i and b_j,

$$M(i,j) = \text{Max} \begin{cases} M(i-1,j), \\ M(i,j-1), \\ M(i-1,j-1) + \Delta \end{cases}$$

where

$$\Delta = 1 \quad \text{if } a(i) = b(j) \quad \text{and} \quad \Delta = 0 \quad \text{if } a(i) \neq b(j),$$

under the initial conditions $M(i, 0) = M(0, j) = 0$. Then $M(m, n)$ is the length of the longest common subsequence, and all longest common subsequences may be found by a simple traceback routine on the matrix M: starting with matrix cell (m, n), we choose the predecessor cell (i, j) to be any one of the cells $(m-1, n)$, $(m, n-1)$, $(m-1, n-1)$ that satisfies

$$M(m,n) = M(m-1,n),$$

$$M(m,n) = M(m,n-1),$$
$$M(m,n) = M(m-1,n-1) + \Delta$$

respectively. We continue by finding a predecessor of (i,j), then its predecessor, and so on, until we reach some cell $(0,k)$ or $(k,0)$. Whenever during this procedure a predecessor to (r,s) is found that satisfies $M(r,s) = M(r-1,s-1)+1$, then $a(r) = b(s)$ forms part of the longest common subsequence.

I did not know at the time that this was a straightforward application of discrete dynamic programming – nor did any of its many other "discoverers", before or since!

Enter the Polish connection. A couple of weeks at most after I found this algorithm, a colleague of mine, S. K. Zaremba, hosted a colloquium on numerical computation of integrals in high-dimensional space, including Monte-Carlo methods—whose proceedings came the next year (Zaremba, 1972). Stanislaw Ulam, prominent among the promoters of Monte-Carlo methods, was invited. His talk covered a wide range of subjects, among them the problem of how to calculate the minimum edit distance between two finite binary sequences, based on substitutions, insertions and deletions of single terms. He stressed in the talk and in the published version (Ulam, 1972) that no algorithm was known for handling this problem. Needless to say, I was astonished since this sort of sequence comparison was uppermost in my mind, and I was in the middle of writing up my algorithm. Imagine being in an audience where Ulam formulates an important unsolved problem, which you have just cracked a few days earlier! After the talk I approached him and started to explain my result. He listened for about 30 seconds and then turned to speak to someone else, after saying my problem (maximum matching) and his (minimum edit distance) were different. I was astonished for the second time, partly because of his peremptory attitude, and partly because it was clear to me that the difference between the two problems lay only in their strategies for counting multi-term gaps (strings of terms in one sequence unmatched to any terms in the other, representing deletion or insertion of several terms in the edit calculation), but the essential nature of the two problems was the same, and that dynamic programming would work as well for one as for the other. I talked to Cedergren and he assured me that the edit distance would weight long gaps too heavily, and I promptly forgot about the incident. Recently, in the same collection as Goad's article, I read a controversial article by Gian-Carlo Rota (1989), in which he characterizes Ulam in this period as having a very limited attention span—maybe this accounts for his dismissive reply.

I do not think Mark Kac was at the CRM during Ulam's visit, though he had spent at least six months there during 1969–1970, and had visited frequently thereafter, being a member of the advisory board. I had gotten to know him somewhat and when I wrote my article on maximum matching, I asked him to communicate it to the *Proceedings of the National Academy of Sciences*. He gave it to a colleague at Rockefeller, Peter H. Sellers, to referee (perhaps on the basis of an indirect biological connection: Seller's interest in the combinatorics of chemical reaction systems). Not only did Sellers review my paper positively, he became very interested in the problem. In the meantime Cedergren had come across the article by Needleman and Wunsch (1969), published a year earlier, in which dynamic programming was first proposed for the comparison of protein sequences. Fortunately, I had made further progress on the problem, incorporating constraints on insertions and deletions, and refocused my 1972 *PNAS* article on this. Also, the computing time of the Needleman and Wunsch algorithm was a cubic function of sequence length, and my formulation of the unconstrained algorithm was quadratic—the number of steps required to compute the matrix M above is proportional to mn, or n^2 if $m = n$. Later I found out that at this time Donald Knuth knew the problem was quadratic, since he incorporated it in a technical report of open combinatorial problems that Vaclav Chvátal, D.A. Klarner and he put out (Knuth *et al.*, 1972). Pavel Pevzner suggests that Knuth knew this via his work on Young tableaux. Later I found out that speech recognition researchers had been using a quadratic dynamic programming approach for this general type of problem

since the late 1960s (Vintsyuk, 1968), as documented in the collection I put together with Joseph Kruskal (Sankoff and Kruskal, 1983). Walter Fitch once told me that when the Needleman and Wunsch article came out, he took the basic idea and programmed an algorithm so that it performed in quadratic time and did not realize that the original was cubic.

Shortly thereafter, I received a manuscript from Sellers in which he generalized the sequence comparison problem and my algorithm with constraints to the finding of optimal paths in partial orders. I made some suggestions and we worked on it together, publishing it in *Discrete Mathematics*, entitled in catchy dactylic heptameter: *Shortcuts, diversions and maximal chains in partially ordered sets* (Sankoff and Sellers, 1972). Some time later in early 1972 (Goad mistakenly cites 1974), Kac invited Ulam to talk at Rockefeller, where Sellers heard him pose the edit-distance version of the problem. Sellers devised a dynamic programming algorithm for it, based on maximum matching and reminiscent of my algorithm with insertion and deletion constraints, requiring cubic computing time, and sent me a copy of the relevant pages from his notebook. I pointed out how it could be done in quadratic time, as did Albert Nijenhuis, and the published version (Sellers, 1974a) incorporates our suggestions in a footnote. In the above notation, using $D(i,j)$ for the minimum number of steps, insertions, deletions or substitutions, to convert a_i into b_j,

$$D(i,j) = \text{Min} \begin{cases} D(i-1,j)+1, \\ D(i,j-1)+1, \\ D(i-1,j-1)+\delta \end{cases}$$

where

$$\delta = 0 \quad \text{if } a(i) = b(j) \quad \text{and} \quad \delta = 1 \quad \text{if } a(i) \neq b(j),$$

under the initial conditions $D(i,0) = D(0,i) = i$. Then $D(m,n)$ measures the edit distance between the two sequences, and all appropriate sets of edit steps may be found by the traceback routine in the matrix D.

The publication of this paper and similar work (Sellers, 1974b) had an immediate impact among mathematicians and biologists. They are frequently cited, though in recent years, perhaps because of the prevalence of local comparison, the tendency has been to ascribe importance to the dynamic programming and not the edit distance, for which credit is most generally given to Needleman and Wunsch.

If the distance approach is generalized to allow a different weight $s > 0$ on substitution compared to insertion and deletion:

$$D(i,j) = \text{Min} \begin{cases} \{D(i-1,j)+1, \\ D(i,j-1)+1, \\ D(i-1,j-1)+s\delta \end{cases}$$

then the longest common subsequence problem and the shortest edit distance problem become essentially identical when $s \geq 2$.

Soon after, Wagner and Fischer (1974) published this same algorithm in a computer science journal, oblivious of the previous computational biology work—to say nothing about the speech recognition work—and this had even greater impact. Not only was it frequently cited, but it interested a number of talented computer scientists in the problem area.

I have not mentioned many of the other independent discoveries of the same algorithm. It did not require great mathematical power or imagination to find a good solution to the sequence comparison problem. At least a half-dozen people who chanced upon the problem, in one field or another, quickly came up with the same solution (Kruskal, 1983). Ulam's characterization of it as an important unsolved problem was making a mountain out of a molehill.

Though the dynamic programming algorithm for the global alignment of two sequences was the inevitable result of a widespread need for this application, not all the early discoveries in computational biology necessarily would eventually have been made without the inspiration of their specific authors. For example, the optimal local alignment algorithm of Smith and Waterman (1981), which is more important for applications than the global comparison method, is based on an extremely simple but non-obvious adjustment to the basic dynamic programming algorithm. Local alignment was an important problem that defied a number of attempts at solution. Sellers (1980) did

find a valid algorithm, but this was rather unwieldy. The Smith-Waterman algorithm was ingenious, exactly what was needed, and has had wide impact. We define

$$L(i,j) = \text{Max} \begin{cases} 0, \\ L(i-1,j)-1, \\ L(i,j-1)-1, \\ L(i-1,j-1)+\sigma \end{cases}$$

where

$$\sigma \text{ is a constant} > 0 \quad \text{if } a(i) = b(j)$$

and

$$\sigma = -1 \quad \text{if } a(i) \neq b(j),$$

under the initial conditions $L(i,0) = L(0,i) = 0$. Then $\text{Max}_{i,j} L(i,j)$ measures the score of the optimal local alignment between the two sequences, and all appropriate sets of edit steps may be found by the traceback routine in the matrix L, starting at any cell containing this maximizing value and terminating when a cell containing zero is reached. If there are several regions of good local alignment between the two sequences, this algorithm finds all of them with a single pass of the dynamic programming followed by tracebacks from all cells containing locally maximal values of L.

3 Multiple Alignment and Phylogeny

Soon after our first applications of sequence comparison to pairs of small RNAs, Cedergren and I became interested in assessing the relative rates of the 12 possible substitution mutations among the four bases $\{A, C, G, U\}$, based on comparing small homologous RNA sequences from N species. The idea was to isolate each position in the ribonucleotide sequence, examine its evolutionary history, count the number of mutations of each kind, and then combine the data for all positions. Given the particular data available—one sequence each from a bacterium, a yeast, human, chicken and frog, there was no problem of determining the phylogenetic relationships among the species. In other words, the topology of the phylogeny was given. What was left was only the task of aligning corresponding positions in all the sequences and counting the mutations at all positions. After a little work, it was clear that dynamic programming would work here too, with one interesting difficulty. In edit distance terms:

$$D(\mathbf{i}) = \text{Min}_{\mathbf{e} \in \{\mathbf{0},\mathbf{1}\}^N} \{\mathbf{D}(\mathbf{i}-\mathbf{e}) + \mathbf{f}(\mathbf{e} \bullet \mathbf{a}(\mathbf{i}))\}$$

where $\mathbf{i}$ is an N-vector representing a cell in the N-way matrix of sequence positions, generalizing (i,j) in the above algorithms, $\mathbf{e}$ is a vector of 0's and 1's (excluding $\mathbf{0}$), the "product" $\mathbf{e} \bullet \mathbf{a}(\mathbf{i})$ is a vector containing the i_j-th term of sequence $\mathbf{a_j}$ if $e_j = 1$, but containing Φ if $e_j = 0$. (The appropriate initial conditions are not hard to work out.)

How is f calculated? Labeling the N terminal nodes of the given phylogenetic tree with the values of $\mathbf{e} \bullet \mathbf{a}(\mathbf{i})$, the question is how to label the internal, or ancestral, nodes of the tree so as to minimize the number of edges in the tree with two differently labeled end-points. After consulting with combinatorialist colleagues at the CRM, including A. Kotzig, it appeared to me that this problem was non-trivial. Indeed, Margaret Dayhoff (1969) was using an incorrect method at this time. But dynamic programming yields a solution. Some internal node on the tree is (through biological considerations, or arbitrarily) identified as the root ρ and all tree edges are directed away from the root. Then ρ alone is ancestral to all other tree nodes. Consider any non-terminal node ν whose immediate descendants are $\nu_1, \ldots, \nu_r$. Let A be the (finite) alphabet from which the terms of all the $\mathbf{a_j}$ are drawn. The problem is solved by applying the following recurrence to any node ν whose descendants are either terminal nodes or have already had the recurrence applied to them (this is always possible in a tree structure): For all $X \in A \cup \Phi$:

$$F_\nu(X) = \text{Min}_{\mathbf{Y} \in (\mathbf{A} \cup \mathbf{\Phi})^r} \sum_{i=1}^{r} \big[F_{\nu_i}(Y_i) + \delta(X, Y_i)\big],$$

where

$$\delta(X,Y) = 0 \quad \text{if } X = Y, \quad \delta(X,Y) = 1 \quad \text{if } X \neq Y,$$

with the initial conditions that $F_{\nu_i}(Y_i) = 0$ if ν_i is a terminal node and Y_i is the original label assigned to that node and $F_{\nu_i}(Y_i) = \infty$ otherwise. Then

$$f\big(\mathbf{e} \bullet \mathbf{a}(\mathbf{i})\big) = \mathrm{Min}_{\mathbf{Y} \in \mathbf{A} \cup \mathbf{\Phi}} \mathbf{F}_\rho(\mathbf{Y}).$$

The traceback part of the dynamic programming based on this algorithm reconstructs the inferred ancestral sequences at each of the nonterminal nodes. For fixed N this is still a polynomial algorithm, though the degree of the polynomial is N. Nevertheless, it can be implemented for small N and can serve as the basis for heuristic or locally optimal algorithms in the case of large N.

I rushed off a manuscript containing this algorithm to Mark Kac, again requesting him to communicate it to *PNAS*, but I did not receive a reply for many months. At the same time, I published a short paper with Cedergren and my student Cristiane Morel, where we applied the method to RNA sequences in order to calculate the mutation frequencies that were our original interest (Sankoff *et al.*, 1973). In retrospect, of course, it turns out that the significance of the paper was not the mutation frequencies, nor even the reconstruction of the ancestral sequences, but that this was the first formal algorithm for multiple sequence alignment. At the time, the eventual importance of multiple alignment was not realized.

During this time, a paper by John Hartigan (1973) came out in containing the essentials of the dynamic programming calculation for f. Indeed, this idea was first published by Walter Fitch (1970). Steve Farris (1971) had adapted it to optimizing trees in L_1 and some time later Pascale Rousseau and I showed how to apply it in an arbitrary metric space (Sankoff and Rousseau, 1975).

After waiting for perhaps six months for a reply from Kac about the paper (though in the interim I had seen him a few times) I finally summoned up enough courage to ask him about it. He told me that the referee thought that it did not represent enough of an advance over the $N = 2$ case, and that I should incorporate in the algorithm a way of simultaneously optimizing the tree topology! I was dismayed; even if complexity theory was not yet popular, I knew that finding a Steiner tree in Euclidean, L_1 or other well-known spaces was exceedingly hard and had been studied intensively for years by many prominent mathematicians. Doing it in the space of sequences under the edit distance metric was quite an order! I published my algorithm elsewhere (Sankoff, 1975).

4 Secondary Structure

Continuing his interest in RNA, Cedergren suggested we explore computer methods for analyzing the secondary structure of 5S RNA. In the same way that two entirely complementary DNA strands bind together through Watson-Crick pairing in forming a double helix, different regions of a single RNA molecule can bind to each other (C ribonucleotides with G ribonucleotides and A with U) to form a structure with single-stranded loops closed by double-stranded stems. In a secondary structure, or "folding" of the sequence $\mathbf{a}$, the two regions $a(i), \ldots, a(i+k)$ and $a(j), \ldots, a(j-k)$, where $k \geq 2$, can form a helical stem by pairing together if each $a(i+h)$ is complementary to $a(j-h)$, for $h = 0, \ldots, k$, if none of the terms is already paired to some other term, if $3 < (j-k-i-k)$ and if, in the simplest case, there are no "knots" in the structure. A knot exists if $a(r)$ is paired with $a(s)$ while $a(u)$ is paired with $a(v)$ and $r < u < s < v$.

Generally, for a given RNA molecule there are several foldings that satisfy all these conditions. A working hypothesis is that the folding that has a stable existence is the one that maximizes the free energy when it is formed. Stem formation releases energy in an amount which depends in a known way on k, the number of terms, and on which ribonucleotides are involved. Unpaired regions have the opposite effect and require energy to be formed, to an extent which may again depend on the number of terms and on which ribonucleotides are involved, but also depends on a structural parameter R. It can be shown

that every secondary structure satisfying the above conditions can be uniquely decomposed into a number of stems, at most two energetically neutral "dangling ends" containing the first or last terms of **a** if they are unpaired, and a number of R-loops, described as follows: $a(i_r), \ldots, a(k_r)$ are all unpaired, for $r = 1, \ldots, R$, $a(i_1 - 1)$ is paired to $a(k_R + 1)$, and the remaining $a(i_r - 1)$ are each paired to $a(k_{r-1} + 1)$. If $R = 1$ the loop is called a "hairpin", if $R = 2$ the loop is called an interior loop—except in the special case of a "bulge", with one null unpaired region so that $a(i_1 - 1)$ and $a(k_1 + 1)$ are paired to $a(i_2)$ and $a(i_2 - 1)$, respectively. For $R \geq 3$, the loop is called a multiple loop. Thanks to a series of experimental studies pioneered by Tinoco (1973), estimates are available for the energetic contribution of all possible stems and loops and these can be summed to estimate the free energy of a folding.

We obtained a copy of a program by Pipas and McMahon (1975) which essentially finds all possible stems that could be formed in a RNA sequence, and tries to fit them together so as to minimize the energy. With Annie Morin, we tried this on a sample of 5S RNA's from phylogenetically diverse organisms and found, as an early example of what is now known as "phylogenetic filtering", that only one structure recurred among the five energetically most stable solutions in almost all of the species (Sankoff *et al.*, 1978). This turns out to correspond to the folding which is now generally recognized for this molecule. Years later I formalized the concept of phylogenetic filtering by simultaneously solving the folding and multiple alignment problems using a weighted objective function (Sankoff, 1985).

While doing this research it became apparent that secondary structure stems disrupted only by bulges and other interior loops could be detected by the dynamic programming comparison of an RNA sequence with a reversed copy of itself, with negative costs for base pairing and positive costs for 2-loop formation. Trying to incorporate multiple loops into this methodology was somewhat trickier, and I devised an iterative algorithm which produced, on the first pass, only 0-branchings, namely stems alternating with 2-loops. On the second pass, it could add several such stems branching off a main stem, producing 1-branchings, and so on. Eventually, any secondary structure could be found. I gave a talk on this at the Numerical Taxonomy meetings in Kansas in the fall of 1976 and had a summer student, Nguyen Bach Hue, implement it in a computer program. The method turned out to be very dependent on the crude energy estimates available at the time and produced rather bizarre structures, so at the end of the summer I put the project on the shelf. A year or two later I came across Michael Waterman's papers (Waterman, 1978; Waterman and Smith, 1978) on this topic, in which, following a query of Charles Delisi, the combinatorics of folding and the same type of iterative algorithm I had be en experimenting with were already rigorously developed and indeed had been announced in a 1975 Los Alamos technical report.

During this time Mark Kac invited me to give a talk at Rockefeller University and I spoke about my approach to folding. George Pieczenik, a biochemist and former Rockefeller post-doc, was present and showed a great deal of interest in the subject. He had previously worked with Sydney Brenner at Cambridge on pattern recognition in sequences. I cannot recall the details of our discussion, but he may have been insisting on the possibility of a single-pass algorithm instead of the iterative approach. A year or so later the basic idea of a single-pass dynamic programming algorithm was published by Ruth Nussinov, Pieczenik, Jerrold Griggs, and Daniel Kleitman (Nussinov *et al.*, 1978). Nussinov was a graduate student working with Pieczenik; they had gotten in touch with Kleitman, and Mark Kac encouraged the resulting collaboration. I have since learned from Griggs, who was Kleitman's student, that experience with dynamic programming in another context, modeling shortest time turns of supersonic aircraft, led him to try it on the folding problem with multiple loops (Griggs, 1977). Certainly I had never thought of this solution, and did not think of it after my meeting Pieczenik. Nevertheless it is interesting that once again this interaction took place under the aegis of Mark Kac, and re-stimulated my interest in secondary structure (Sankoff *et al.*, 1983;

Zuker and Sankoff, 1984; Sankoff, 1985; Ferretti and Sankoff, 1988). Michael Zuker of the National Research Council of Canada wrote a very effective and widely disseminated program based on the Nussinov *et al.* principles (Zuker and Stiegler, 1985), later versions of which are still in use around the world.

The dynamic programming recurrence fundamental to folding may be represented as:

$$C(p,q) = \min_{R \geq 1} \min_{R-\text{loops } s} \left\{ E(s) + \sum_{r=1}^{R-1} C(k_r+1, i_{r+1}-1) \right\}$$

for the partial sequence $a(p), \ldots, a(q)$, if $a(p)$ and $a(q)$ can be paired, where $p = i_1 - 1, q = k_R + 1$ for any R-loop in the above notation. For unpairable $a(p)$ and $a(q)$, $C(p,q) = \infty$. Consecutive base-pairs in stems are considered to form 2-loops with two null unpaired regions in the same way as a bulge is a 2-loop with only one unpaired region. The energy terms $E(s)$ can be looked up in tables or derived by simple calculations provided by experimentalists. There are of course initial conditions; these are easily worked out.

The recurrence as it stands may require exponential computing time, but it elegantly expresses the idea of a single-pass algorithm. Mild restrictions on the form of $E(s)$ for multiple loops lead to polynomial algorithms (Sankoff *et al.*, 1983).

5 Conclusion

Returning to the puzzlement I described in the Introduction, it is now clear to me that though he certainly did not "start all this", Ulam *did* have some impact on the early use of dynamic programming. He did not invent the edit distance that Goad (1989) and Ronald Graham (1989) associate with his name—it was published by Levenshtein years earlier (1965)—and he exaggerated its difficulty as a mathematical problem and did not realize that dynamic programming, for example as already published by Needleman and Wunsch, represented a solution. He did, however, interest Peter Sellers, fresh from our dynamic programming collaboration, in this problem, and the resulting paper did have an impact. And he did influence a number of people at Los Alamos, including Michael Waterman, in the mathematics of sequences (cf Waterman, 1995). But it is clear that it was through refereeing for Mark Kac that Sellers originally learned about dynamic programming for sequence comparison and through Kac that he heard about the edit-distance problem from Ulam. Furthermore, Waterman reports that his own involvement with dynamic programming was triggered by Seller's paper. Independent of these developments, Mark Kac encouraged the Pieczenik-Kleitman collaboration leading to Griggs' algorithm. As far as my own work is concerned, I have described Mark Kac's encouragement in connection with my first paper on sequence comparison, the work on multiple alignment, and the research on RNA secondary structure. I doubt that Mark Kac counted me among his protégés; we scarcely ever discussed the details of a mathematical problem, and he did no work in the area. One area where he probably would have been of great help was the probabilistic analysis of matching random sequences which I undertook with Vaclav Chvátal (Chvátal and Sankoff, 1975), but I never took the opportunity to discuss it with him. (Indeed, he was interested in Waterman's early work in a related area, the statistics of local alignment.) Nevertheless, his friendly encouragement, his invitations to Rockefeller, his hospitality and (I suspect) his letters of reference, constituted a crucial impetus to my work at the CRM in the early years.

Acknowledgements

Thanks to the American Mathematical Society for permission to adapt a 1997 version of this article with the same title, published in Vinet,L. (ed.), *Advances in the Mathematical Sciences: CRM's 25 years.* CRM Proceedings and Lecture Notes, **11**, Providence, RI, AMS, pp. 403-413. Thanks as well to Alberto Apostolico, Martin Goldstein and Michael Waterman for reading and commenting on a draft of that version, to Jerrold Griggs, Jonothan Logan, George Pieczenik and Peter Sellers for generously responding to my in-

quiries, and to Erica Jen for key bibliographic leads. And to the late Robert Cedergren, who got me involved in the field, and whose collaboration over the years was a most rewarding and enjoyable experience.

References

Apostolico,A. and Giancarlo,R. (1998) Sequence alignment in molecular biology. *J. Comput. Biol.*, **5**, 173–196.

Chvátal,V., Klarner,D. A. and Knuth,D. E. (1972) Selected combinatorial research problems. *Stanford University Computer Science Technical Report*, **72–292**, 26.

Chvátal,V. and Sankoff,D. (1975) Longest common subsequences of two random sequences. *J. Appl. Probability*, **12**, 306–315.

Dayhoff,M. O. (1969) Computer analysis of protein evolution. *Sci. American*, **221**, 86–95.

Farris,J. S. (1970) Methods for computing Wagner trees. *Systematic Zoology*, **19**, 83–92.

Feigenbaum,M. (1982) Reflections of the Polish masters—an interview with Stanislaw Ulam and Mark Kac. *Los Alamos Science*, **3**, 54–65.

Ferretti,V.and Sankoff,D. (1989) A continuous analog for RNA folding. *Bull. Math. Biol.*, **51**, 167–171.

Fitch,W. M. (1971) Towards defining the course of evolution: Minimum change for a specific tree topology. *Systematic Zoology*, **20**, 406–416.

Goad,W. (1989) Sequence analysis—contributions by Ulam to molecular genetics. In Cooper, N. G. (ed.), *From Cardinals to Chaos. Reflections on the Life and Legacy of Stanislaw Ulam.*, Cambridge University Press, pp. 288–291.

Graham,R. L. (1989) A similarity measure for graphs—reflections on a theme of Ulam. In Cooper, N. G. (ed.), *From Cardinals to Chaos. Reflections on the Life and Legacy of Stanislaw Ulam.*, Cambridge University Press, pp. 114–121.

Griggs,J. R. (1977) *Symmetric chain orders, Sperner theorems, and loop matchings.* Ph.D. thesis. Massachusetts Institute of Technology.

Hartigan,J. A. (1973) Minimal mutation fits to a given tree. *Biometrics*, **29**, 53–65.

Kruskal,J. B. (1983) An overview of sequence comparison. In Sankoff,D. and Kruskal,J.B. (eds.), *Time Warps, String Edits and Macromolecules. The Theory and Practice of Sequence Comparison.* Addison-Wesley, Reading, Mass., pp. 1–33.

Levenshtein,V. I. (1965) Binary codes capable of correcting deletions, insertions and reversals. *Dokl. Akad. Nauk USSR*, **163**, 845–848 (Russian); Engl. Transl.: *Cybernetics and Control Theory*, **10**, 707–710 (1966)

Needleman,S. B. and Wunsch,C. D. (1970) A special method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Nussinov,R. Pieczenik,G., Griggs,J. R. and Kleitman,D. J. (1978) Algorithms for loop matchings. *SIAM J. Appl. Math.*, **35**, 68–82.

Pipas,J. M. and McMahon,J. E. (1975) Method for predicting RNA secondary structure. *Proc. Nat. Acad. Sci. U.S.A.*, **72**, 2017–2021.

Rota,G.-C. (1989) The lost café. In Cooper, N. G. (ed.), *From Cardinals to Chaos. Reflections on the Life and Legacy of Stanislaw Ulam.*, Cambridge University Press, pp. 23–32.

Sankoff,D. (1972) Matching sequences under deletion/insertion constraints. *Proc. Nat. Acad. Sci. U.S.A.*, **69**, 4–6.

Sankoff,D. (1975) Minimal mutation trees of sequences. *SIAM J. Appl. Math.*, **28**, 35–42.

Sankoff,D. (1985) Simultaneous solution of the RNA

folding, alignment and protosequence problems. *SIAM J. Appl. Math.*, **45**, 810–825.

Sankoff,D. and Kruskal,J. B. (eds.) (1983) *Time Warps, String Edits and Macromolecules. The Theory and Practice of Sequence Comparison.* Addison-Wesley, Reading, Mass.

Sankoff,D., Cedergren,R. J., Kruskal,J. B. and Mainville, S. (1983) Fast algorithms to predict RNA secondary structures containing multiple loops. In Sankoff,D. and Kruskal,J.B. (eds.), *Time Warps, String Edits and Macromolecules. The Theory and Practice of Sequence Comparison.* Addison-Wesley, Reading, Mass., pp. 93–120.

Sankoff,D., Morel,C. and Cedergren,R. J. (1973) Evolution of 5S RNA and the non-randomness of base replacement. *Nature New Biol.*, **245**, 232–234.

Sankoff,D., Morin,A.-M. and Cedergren,R. J. (1978) The evolution of 5S RNA secondary structures. *Canad. J. Biochem.*, **56**, 440–443.

Sankoff,D. and Rousseau,P. (1975) Locating the vertices of a Steiner tree in an arbitrary metric space. *Math. Programming*, **9**, 240–246.

Sankoff,D. and Sellers,P. H. (1973) Shortcuts, diversions and maximal chains in partially ordered sets. *Discrete Math.*, **4**, 287–293.

Sellers,P. H. (1974a) An algorithm for the distance between two finite sequences. *J. Combin. Theory Ser. A*, **16**, 253–258.

Sellers,P. H. (1974b) On the theory and computation of evolutionary distance. *SIAM J. Appl. Math.*, **26**, 787–793.

Sellers,P. H. (1980) The theory and computation of evolutionary distances: Pattern recognition. *J. Algorithms*, **1**, 359–373.

Smith,T. F. and Waterman,M. S. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Tinoco,I., Borer,P. N., Dengler,B., Levine,M. D., Uhlenbeck,O. C., Crothers,D. M. and Gralla,J. (1973) Improved estimation of secondary structure in ribonucleic acids. *Nature New Biol.*, **246**, 40–41.

Ulam,S. M. (1972) Some combinatorial problems studied experimentally on computing machines. In Zaremba, S.K. (ed.) *Applications of Number Theory to Numerical Analysis.* Academic Press, New York, pp. 1–10.

Vintsyuk,T. K. (1968) Speech discrimination by dynamic programming. *Kybernetika*, **4**, 81–88 (Russian); Engl. Transl.: *Cybernetics*, **4**, 52–57.

Wagner,R. A. and Fischer,M. J. (1974) The string-to-string correction problem. *J. Assoc. Comput. Mach.*, **21** 168–173.

Waterman,M. S. (1978) Secondary structure of single-stranded nucleic acids. *Studies in Foundations and Combinatorics, Adv. in Math. Suppl.*,**1**, 167–212.

Waterman,M. S. (1995) *Introduction to computational biology*, Chapman and Hall, London, xiv.

Waterman,M. S. and Smith,T. F. (1978) RNA secondary structure: A complete mathematical analysis. *Math. Biosci.*, **42**, 257–266.

Zaremba,S. K. (ed.) (1972) *Applications of number theory to numerical analysis.* Academic Press, New York.

Zuker,M. and Sankoff,D. (1984) RNA secondary structures and their prediction. *Bull. Math. Biol.*, **46**,591–621.

Zuker,M. and Stiegler,P. (1981) Optimal folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, **9** 133–148.