Volume 10 Number 1 1982 Nucleic Acids Research

Efficient algorithms for folding and comparing nucleic acid sequences

Jean-Pierre Dumas and Jacques Ninio

Biochimie de l'évolution, I.R.B.M., Tour 43, 2 Place Jussieu, 75251 Paris Cedex 05, France

Received 3 September 1981

SUMMARY

Fast algorithms for analysing sequence data are presented. An algorithm for strict homologies finds all common subsequences of length $\geqslant 6$ in two given sequences. With it, nucleic acid pieces five thousand nucleotides long can be compared in five seconds on CDC 6600. Secondary structure algorithms generate the N most stable secondary structures of an RNA molecule, taking into account all loop contributions, and the formation of all possible base-pairs in stems, including odd pairs (G.G., C.U., etc.). They allow a typical 100-nucleotide sequence to be analysed in 10 seconds. The homology and secondary structure programs are respectively illustrated with a comparison of two phage genomes, and a discussion of *Drosophila melanogaster* 5S RNA folding.

INTRODUCTION

With the increase in length and number of the determined nucleic acid sequences, there is a growing need for programs that can efficiently exploit the available data (1). Several programs were written around the world for editing or performing pattern analysis tasks on nucleic acid sequences (2-5, and J.-P. D., unpublished). Our main effort, the last years, has been in the design of reliable programs to search for secondary structures in nucleic acids. We describe here the main algorithmic ideas that helped us to gain on computing time without sacrificing the quality of the results. We also describe an algorithm for comparing sequences which is extremely efficient in finding common subsequences to two large sequences.

HOMOLOGY PROGRAMS

We shall describe, in its simplest form, an algorithm to search for strict repeats within a sequence. It adapts to the nucleic acid field some strategies that are classical in lexicographic search problems ('hash coding' and 'separate chaining'). With minor changes the algorithm searches for palindromes or inverted repeats or searches for homologies, inverted homologies or complementarities between two sequences.

We have a sequence N nucleotides long and want to know whether or not it contains repeated subsequences. The simplest strategy, used by Fitch (6) involves an explicit comparison between every nucleotide and every other nucleotide of the same sequence. Thus the number of calculation cycles is of the order of $N^2/2$. With genomes of five thousand or more nucleotides, the method becomes time-consuming. Korn's algortihm (4) involves in a first stage the determination and listing of all the subsequences of any length that are present only once in the complete sequence, then a stage of pairwise comparison of the sequences, facilitated by the fact that closely related subsequences occupy neighbouring positions in the list. We give now our central idea. Let us take a sequence :

1 10 20

T C G G A T T C G T A C G G T A C G G A T C

and consider that it is a string of overlapping n-mers. Written as a sequence of overlapping dinucleotides, the above sequence becomes :

1 10 20

TC,CG,GG,GA,AT,TT,TC,CG,GT,TA,AC,CG,GG,GT,TA,AC,CG,GG,GA,AT,TC

In practice, when dealing with sequences several thousand nucleotides long, we consider them as strings of overlapping hexanucleotides. For simplicity of exposition we conduct here the analysis at the dinucleotide level. Choosing a numbering system, we affect a different number to each possible dimer and rewrite the sequence as a string of numbers, thus for instance :

1 10 20

14, 4, 1, 9, 7, 6, 14, 4, 5, 10, 15, 4, 1, 5, 10, 15, 4, 1, 9, 7, 14

We now scan <u>once</u> the sequence from beginning to end, and fill two tables, T and M. Table T has sixteen places corresponding to the sixteen possible dinucleotides. In scanning the sequence, we find for instance that position 8 is occupied by dinucleotide 4 (CG). We then fill position 4 of T with number 8. But position 4 of T was containing a 2 since there is a CG in position 2 of the sequence. Before putting in the 8, we transfer the 2 into table M. Table M is, like the sequence, N positions long. We put the number 2 at the 8th position of M. Thus, M(8) = 2 indicates that a dimer of rank 8 in the sequence occured previously at position 2. At the end of the scanning the states of the various tables are as follows, S being the sequence written as a string of dimers :

	1				5					10					15					20	
(S)	14	4	1	9	7	6	14	4	5	10	15	4	1	5	10	15	4	1	9	7	14
(M)	0	0	0	0	0	0	1	2	0	0	0	8	3	9	10	11	12	13	4	5	7
(T)	18	0	0	17	14	6	20	0	19	15	0	0	0	21	16	0					

Now, we scan once table T. Looking at position 14, for instance, we learn that the dimer 14 (= TC) is present at position 21 of S, since T(14) = 21. Looking then at position 21 of table M, we learn, by M(21) = 7 that the previous occurence of TC was at position 7 of S, and by M(7) = 1, we learn that TC is present at position 1 of S too. We then refine the analysis by making pairwise comparisons and checking whether or not the discovered homologies extend beyond the initial dimers.

For example, we compared two complete genomes : M13, comprising 6407 nucleotides (7) and G4, containing 5577 nucleotides (8). Restricting the search to strict homologies at the decanucleotide level or beyond, and selecting one strand in each genome, we find 38 common decanucleotides in the two sequences, 15 common undecamers, and three dodecamers : TCTATTGTTGAT (position 5467 in G4, 3580 in M13), AAAGATGGCAAA (5530, 2415), and TGATATGGTTGG (2731, 4462). The total computation time for this task was less than two seconds on CDC 6600. The programs have been used to analyse ribosomal RNAs (9) and mitochondrial DNA (to be published elsewhere). A user's guide and the complete listing of the programs will be sent upon request.

SECONDARY STRUCTURE PROGRAMS

In a given sequence of N nucleotides, there are L stems that can be formed by associating pairs of complementary or nearly complementary segments. L grows roughly like N^2. Then, there are 2^L combinations of segments that may or may not correspond to feasible secondary structures. The problem of finding a fast algorithm to evaluate secondary structures is a challenging one, the requirement for speed being more important than for a homology program.

We started develloping secondary structure prorgams four years ago. One of them was used to exploit the information contained in tRNA sequences (see (10) for peripheral details on the functionning of the program and for an introduction to the problem of secondary structure evaluation. Here we restrict ourselves to the central ideas which allowed us to save on calculation time.

To begin with (we shall remove this constraint later) consider potential structures containing at most M stems. For simplicity, consider provisionally that the free energy of a structure is the sum of the free energies of its individual stems. Suppose that we have a partial structure containing p stems of combined free energy E_p. We may now complete the structure by adding at most M-p stems. Suppose that we have a way to know that, among the segments that may be added to the partial structure, the M-p most stable ones make up a total free energy E_{M-p}. Then, $E_{min} = E_p + E_{M-p}$ is a lower boundary to the free energies of all structures of M or less segments that contain the initial p segments.

While the program is exploring structures and evaluating their stabilities, it keeps in memory an assigned number of most stable structures. Let E_{best} be the free energy of the most stable structure in memory. If $E_{min} > E_{best}$, we know that no structure (of M or less segments) containing the p initial segments will be more stable than the most stable structure in memory. In practice, we give ourselves a "confidence free energy interval" e of plus three to eight kilocalories and consider that all structures that might have free energies $\leqslant E_{best} + e$ are worth being explored further. But if $E_{min} > E_{best} + e$, the program drops all branches from thereon. The exploration tree is, in our case, a depth first tree pruning, and the comparison between E_{min} and E_{best} is made every time a segment is added to a partial structure. Thus, one key idea is to calculate a lower boundary to the energies that can be obtained along given pathways and drop accordingly the unpromising branches. We explain now how E_{min} is computed.

The L segments are ranked in order of decreasing stabilities (increasing free energies). We have compatibility criteria to decide whether or not any two segments i and j are allowed to participate together to a structure. The usual criteria (11, 12) are that segments must not overlap and that they must be in a certain topological situation with respect to each other - satisfy the "principle of contiguity" (11, 12). For every segment i we construct its "compatibility set" which contains, as booleans, the results of the compatibility tests made with every other segment of the list. In a sense, this set is the list of all the segments that are compatible with segment i. Every result of a compatibility test occupies just one memory bit. During tree pruning, we also construct the "instantaneous compatibility set" of the partial solution, which is equivalent to the list of all the segments that are simultaneously compatible with all the p segments of the partial solution. When adding the next segment the new instantaneous compatibility set is obtained as the intersection of the compatibility set of this segment with the former instantaneous compatibility set. Since our tree pruning draws segments in a fully ordered manner, if the p^{th} segment of the partial solution occupies rank k of the list of segments, the only segments that may be added to the partial solution are those of rank $> k$ in the instantaneous compatibility set. Then, E_{M-p} may be taken as the sum of the free energies of the M-p first segments of rank $> k$ in this set.

Actually, the free energy of a structure is the sum of the free energies of the stems and the free energies of the single-stranded regions (the "topological" energy). In order to incorporate the topological contribution in the $E_{min} - E_{best}$ comparison, two measures were adopted.

First, instead of dealing with the true free energy e_s of a segment, we use a topological corrected energy $e_c = e_s + e_t$ where e_t is, in absolute value, the minimum topological contribution that can be expected from the inclusion of the considered segment anywhere in a structure. When a segment determines a loop too small to accomodate a further segment, e_t includes the topological contribution of the loop.

Let E_A and E_B be the true free energies of structures A and B, obtained by taking into account in a detailed manner all double- and single-stranded contributions. Let E'_A and E'_B be the approximate free energies computed by adding the topologically corrected free energies of the stems. Our extensive work with tRNA and 5S RNA shows that $|(E_A - E_B) - (E'_A - E'_B)|$ is generally less than 2 kcal and almost never exceeds 4 kcal. Our second measure then is to make e in the test $E'_{min} > E'_{best} + e$ sufficiently large that no solution can be lost. Of course, once a promising structure is found, its energy is computed in an exact manner, and compared to the true free energies of the structures stored in memory.

We now present a tree-pruning algorithm which obviates the need for a restriction on the maximum number of stems in a structure.

The idea is to separate the list of the segments into a number of disjoint subsets, the "incompatibility islets" such that within any islet, every segment is incompatible with all the segments of the islet. Assume that the first p segments have been partitionned into m incompatibility islets. If, in each of these islets, there is a segment compatible with segment p+1, then an $(n+1)^{th}$ islet is created, containing this segment. Otherwise, the segment p+1 is incorporated into the first islet not containing a compatible segment. Other modes of partitionning the list into islets have not been considered.

Every structure may contain at most one segment taken from any given islet. When a partial solution of p segments is constructed, we determine E'_p and the instantaneous compatibility set of the partial solution as before. We then determine for every islet that remains to be explored, the most stable segment that is compatible with the partial solution, and we add up the energies of those leading segments to make a total E'_L. We then compare as before $E'_{min} = E'_p + E'_L$ to $E'_{best} + e$. Our published work on tRNA (10) was run entirely on the "islet" program.

There are two main classes of non-heuristic algorithms for predicting secondary structures.

1. Recurrent algorithms in which, having determined the best partial solution for every subsequence of length $\leqslant$p, one determines the best partial solutions for the subsequences of length p+1, until p+1 equals the length of the complete sequence (13-16). Programs belonging to this class can be extremely rapid, especially so for large sequences. Unfortunately, they give by design only one structure so that the biologically important folding may be missed. Furthermore, difficulties are encountered when dealing with multi-branched loops (see (15) and (16)). The difficulty is a basic one for

these algorithms. Consider a subsequence from residue S_i to residue S_j and assume that the residues are unpaired. How to assign a free energy to this subsequence if we do not know whether S_i and S_j belong to internal, or to multi-branched loops ? The answer depends upon the arrangement of the molecule outside the section S_i - S_j. The Zuker-Stiegler program (16) turns around the difficulty by assigning equal values to internal and multi-branched loops. Thus, up to now, the programs of this kind work heuristically. Amendements are required in order to make them reliable, and capable of yielding more than one structure. The increase in complexity required by the implementation of such amendements, and hence the increase in calculation time are unknown.

2. Algorithms which, like ours, construct first a list of possible segments, then combine the segments to find the best structures (12, 17). In this case, the number of combinations to explore, and hence the calculation time are dependent upon a number of options : are knotted topologies allowed ? what is the maximum number of segments that is allowed in a structure ? what are the selection rules for constructing the list of the segments ?

- we allow the standard pairs G.C and A.U, the wobble pair G.U and optionally, all the odd pairs (G.G, C.C, U.C, C.A, A.A, G.A, C.U) with binding values as in (10).
- the shortening of a segment may provide the necessary room for the inclusion of another segment, giving a more stable conformation or permitting the formation of more favorable loops . Unless stated otherwise, we always include in our list the segments with their sub-segments.
- the list may be limited to segments having M base-pairs or more, to segments having an energy below a certain threshold, etc... (10).

In table 1, we give the calculation times obtained for three RNA sequences, under conditions of minimum complexity (no odd pairs, no sub-segments) and maximum complexity (sub-segments and odd pairs taken into account). Calculations are fifty time slower under MAX than under MIN conditions. This should be kept in mind when comparing the efficiencies of various programs.

At present the program is suitable for complete explorations of RNA pieces up to 120 to 150 nucleotides. Are the generated structures biologically significant ? The state of the art is illustrated in Fig. 1, with an analysis of the secondary structures in Drosophila melanogaster 5S RNA (18). The cross-linking of this molecule by trimethyl-psoralen (19) has brought forward reliable evidence on its structure in solution. Fig. 1a shows the structure proposed by Thompson, Wegnez and Hearst on the basis of their croos-linking studies (19). The third best structure generated by our programs, using the "statistical" set of values (10) and a minimum segment length of three base pairs agrees with the cross-linking results and differs from the previous model in two parts, shown in

TABLE 1

Examples of calculation times

	MIN			MAX		
	Number of segments	time (sec.)	set-up/ search	number of segments	time (sec.)	set-up/ search
mitochrondrial yeast tRNA^{Phe}	113	0.70	0.36 / 0.34	279	5.67	1.83 / 3.84
Anacystis Nidulans 5S RNA	191	2.27	0.91 / 1.36	736	67.2	12.1 / 55.1
Drosophila melanogaster 5S RNA	170	1.40	0.77 / 0.63	821	83.3	15.1 / 682

The exploration was made under two different conditions : MIN (no sub-segments, no odd pairs) and MAX (odd pairs and sub-segments taken into account). The number of segments indicated is that of the segments effectively used in the search. The total calculation time (on CDC 6600) is split into a "set-up" time during which the segments are generated and their compatibility sets constructed and a "search" time during which the tree of the solutions is explored recursively. Structures comprising 8 segments or less were examined. The confidence free energy interval e was set to 5.0 kcal and the 5 best solutions were kept in memory. The "empirical" binding model was used for tRNA, and the "statistical" model (10) was used for the 5S RNAs. A minimum segment length of 2 was chosen in the case of mitochondrial yeast tRNA^{Phe} (24) and a minimum length of 3 base-pairs was adopted for the 5S RNA sequences (18, 25). The unusual bulge loop in the TΨ stem of the mitochondrial tRNA is correctly predicted by our program (10) with the empirical, but not the statistical model. For comparison, the Nussinov-Jacobson program (15) deals with sequences 100 nucleotides long in about 15 sec. (on IBM 360) and gives, by design, only one structure.

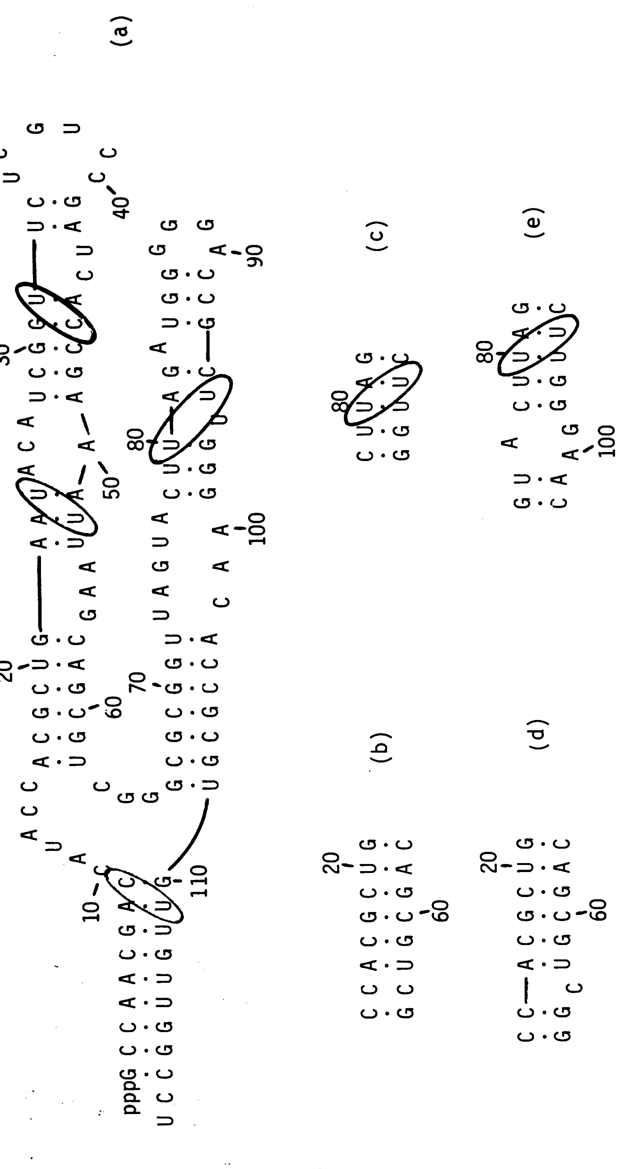

Figure 1.Drosophila melanogaster 5S RNA. (a) Model of Thompson,Wegnez and Hearst (19) based on cross-linking studies.The model was slightly rearranged to incorporate a correction in the sequence around residue 65.The cross-links are circled. (b) and (c) Modifications to this model suggested by the computer search,with a minimum segment length of three base-pairs.(d) and (e) Further modifications,making the model identical with that of Nishikawa and Takemura (20).

Fig. lb and lc. When a minimum segment length of two base pairs is used, the model agreeing with the cross-links regresses further. But if, in addition, the "statistical" set is replaced by the "empirical" set of values (10) the model comes out now on the first line, with the section shown in Fig. lc completed as shown in Fig. le. In Fig. ld we show another possible arrangement of the section in lb. With this modification, our computer generated model becomes identical to the Nishikawa-Takemura model for animal 5S RNA's (20). We must concede, after a close examination of coordinate base replacements in animal 5S RNA's that the Nishikawa-Takemura model emerges as the best rationalization, as also acknowledged by several authors (21, 22). Furthermore, homology arguments have led to a very similar model for prokaryotic 5S RNA's (23).

In model la, the cross-link between U80 and U95 is joining two residues belonging to two different stems. In model lc, the two cross-linked residues belong to the same stem, which contains an odd pair U.U adjoining a wobble pair U.G, thus :

```
5' - C U U A G
     G G U U C - 5'
```

In this case, both the homology arguments (21, 22) and our energetic calculations favour the same arrangement. This stresses the importance of having secondary structure programs capable of dealing properly with odd pairs.

Now if we compare the structures of Fig. lb and ld, we find, according to our energetic estimates, an advantage of 1.3 kcal in favour of lb, while the homology arguments point to the correctness of ld. Thus perhaps C.C pairs were overestimated in our calculations.

However, changing the evaluation of U.U's and C.C's by a fraction of a kcal will not modify fundamentally the situation. First, nothing guarantees that the solution structure of 5S RNA is identical to the functional structure on the ribosome, so that the model derived from homology considerations may rather apply to the latter. Next, the assumptions of additivity and independence of the various single and double stranded contributions in the energetic models are too crude anyhow to justify one structure to be singled out among a dozen competing structures nested within a 2 kcal interval. It is thus important to use algorithms that examine in a detailed manner all the potential structures, rather than algorithms that work fast but superficially. About twenty options are incorporated into our programs allowing a great flexibility of use. Like Zuker and Stiegler (16) wa can impose pairing in certain regions, or force regions of the molecule to remain single stranded. The programs were written in FORTRAN IV, for a CDC 6600 computer. Since they take advantage of the 60-bits capacity of the words and code the information on the compatibility between segments at the binary level, the programs would need rewriting if adapted on computers with differently-sized words.

The listings of the programs and a user's guide will be sent upon request.

REFERENCES

1. Gingeras, T.R. and Roberts, R.J. (1980) Science 209, 1322-1328.
2. Staden, R. (1977) Nucleic Acids Res. 4, 4037-4051.
3. Staden, R. (1980) Nucleic Acids Res. 8, 817-825.
4. Queen, C.L. and Korn, L.J. (1980) Methods in enzymol. 65, 595-609.
5. Sege, R. , Söll, D., Ruddle, F.H. and Queen, C. (1981) Nucleic Acids Res. 9, 437-444.
6. Fitch, W.M. (1966) J. Mol. Biol. 16, 9-16.
7. van Wezenbeek, P.M.G.F., Huselbos, T.J.M. and Schoenmakers, J.C.G. (1980) Gene 11, 129-148.
8. Godson, G.N., Barrell, B.G., Staden, R. and Fiddes, J.C. (1978) Nature 276, 236-247.
9. Jacq, B. (1981) Nucleic Acids Res. 9, 2913-2932.
10. Ninio, J. (1979) Biochimie 61, 1133-1150.
11. Ninio, J. (1971) Biochimie 53, 485-494.
12. Studnicka, G.M., Rahn, G.M., Cummings, I.W. and Salser, W.A. (1978) Nucleic Acids Res. 5, 3365-3387.
13. Lapidus, I.R., Rosen, B. and Hepperle, R. (1977) J. Theoret. Biol. 64, 587-595.
14. Waterman,M.S. and Smith, T.F. (1978) Math.Biosc. 42, 257-266.
15. Nussinov, R. and Jacobson, A.B. (1980) Proc. Nat. Acad. Sci. USA 77, 6309-6313.
16. Zuker, M. and Stiegler, P. (1981) Nucleic Acids Res. 9, 133-148.
17. Pipas, J.M. and Mc Mahon, J.E. (1975) Proc. Nat. Acad. Sci. USA 72, 2017-2021.
18. Benhamou, J., Jourdan, R. and Jordan, B.R. (1977) J. Mol. Evol. 9, 279-298.
19. Thompson, J.F., Wegnez, M.R. and Hearst, J.E. (1981) J. Mol. Biol., 147, 417-436.
20. Nishikawa, U. and Takemura, S. (1974) FEBS Lett. 40, 106-109.
21. Garrett, R.A., Douthwaite, S. and Noller, H.F. (1981) Trends Biochem. Sci. 6, 137-139.
22. MacKay, R.M. and Doolittle, W.F. (1981) Nucleic Acids Res. 9, 3321-3334.
23. Studnicka, G.M., Eiserling, F.A. and Lake, J.A. (1981) Nucleic Acids Res. 9, 1885-1904.
24. Martin, R., Sibler, A.-P., Schneller, J.-M., Keith, G., Stahl, A.J.C. and Dirheimer, G. (1978) Comptes-rendus Acad. Sci. Paris, D 287, 845-848.
25. Corry, M.J., Payne, P.I. and Dyer, T.A. (1974) FEBS Letters 46, 63-66.