An efficient method for scoring base pair interactions

Frederick R. Blattner

Professor of Genetics, University of Wisconsin, 445 Henry Mall, Madison, WI 53706, USA

Received 31 October 1983; Accepted 20 November 1983

ABSTRACT

A simple method is described for speeding up the computation of base pairing interactions. It is especially effective on microcomputers.

In comparing nucleic acid sequences for homology it is often desirable to assign a separate score to the strength of interaction between each of the possible base pairs. Usually this is done with a two dimensional matrix in which the i,j th element contains the score assigned to the interaction of the i th with the j th base. If the 4 standard bases plus one symbol for an unknown base are allowed in the sequence, a 5 by 5 symmetric matrix is required. When searching large sequences for homology, a large number of comparisons must be done and the lookup of scores in the matrix comprises a major part of the task. I therefore investigated ways of improving the efficiency of the process. I found that if the 5 bases are represented by the numbers 0, 1, 3, 7, and 12, a 24 element linear array, could be constructed in which the score for the ij th interaction is stored in the $(i + j)$ th element of the array. With this modification our sequence comparison program ran 6 times faster and required one less element of storage for the scoring array.

The key to the method is the selection of numbers to represent individual bases which have unique pairwise sums. The sets are constructed recursively. After obtaining such a set of n elements, the $(n + 1)$ -st set is found by adding, to the set of n elements, the next largest integer which results in unique pairwise sums. The first 40 such numbers beginning with zero are listed in Table 1. There appears to be no simple pattern to the series, although if it is started with a number different from zero the entire series is simply offset. As shown in the table, memory required for a linear scoring matrix begins to exceed that required for a square matrix beyond 5 elements, but not drastically so.

A Z80 based microcomputer was used for the speed test. In

such computers, multiplication is accomplished by an algorithm based on addition so the increase in speed is primarily due to the elimination of the multiplication needed to address a two dimensional array. If a computer with hardware multiplication were to be used the speed increase would be less significant. Thus, this method is of greatest utility for microcomputers, since these rarely are equipped with hardware multiplication.

TABLE
SETS WITH UNIQUE PAIRWISE SUMS

SET SIZE	SET ELEMENT	MEMORY REQUIRED	
		LINEAR ARRAY	SQUARE ARRAY
1	0	1	1
2	1	2	4
3	3	6	9
4	7	14	16
5	12	24	25
6	20	40	36
7	30	60	49
8	44	88	64
9	65	130	81
10	80	160	100
11	96	192	121
12	122	244	144
13	147	294	169
14	181	362	196
15	203	406	225
16	251	502	256
17	289	578	289
18	360	720	324
19	400	800	361
20	474	948	400
21	564	1128	441
22	592	1184	484
23	661	1322	529
24	774	1548	576
25	821	1642	625
26	915	1830	676
27	969	1938	729
28	1015	2030	784
29	1158	2316	841
30	1311	2622	900
31	1394	2788	961
32	1522	3044	1024
33	1571	3142	1089
34	1820	3640	1156
35	1895	3790	1225
36	2028	4056	1296
37	2253	4506	1369
38	2378	4756	1444
39	2509	5018	1521
40	2779	5558	1600