An Eulerian Path Approach to Local Multiple Alignment of DNA Sequences

Yu Zhang[1], **Michael S. Waterman**[2]

Keywords: local multiple alignment, de Bruijn graph, repeat finding

1 Introduction.

Many available local alignment programs are limited by running time and accuracy when aligning large sequence sets. We present an Eulerian path approach to local multiple alignment for DNA sequences. The computational time and memory usage of this approach is almost linear to the total size of sequence set. By constructing a de Bruijn graph, most of the conserved segments are amplified as heavy paths in the graph, and the original patterns distributed in sequences are recovered even if they do not exist in any single sequence. This approach can detect both short (< 20bps) and long (> 300bps) conserved segments, as well as degenerate patterns (70% pairwise similarity). In addition, a Poisson heuristic [1] is applied to estimate the significance of local multiple alignments.

We demonstrate the performance of our method by an application in Alu repeat finding in the human genome. Five genomic sequences were tested, ranging from 22Kb to 1Mb in length. We compared the result to Alus marked by RepeatMasker[5], which uses detailed information about Alu repeats. Two programs are in good agreement.

2 Method.

The initial motivation for the method arises from the algorithm for DNA fragment assembly using the Eulerian superpath approach [2][4]. We first construct a de Bruijn graph using overlapping k-tuples from the given sequence set. Each k-tuple is represented by a directed edge in the graph, and two edges are joint by a node if their k-tuples overlap at (k-1) letters in any sequence. Identical k-tuples are represented by same edges. Under this construction, each sequence is mapped to a path traversing the graph. If a k-tuple appears in multiple sequences, the corresponding sequence paths will intersect at the edge representing the k-tuple. Based on this property, we define the multiplicity of an edge to be the number of sequence paths visiting the edge, i.e., the number of sequences containing the represented k-tuple. Using a probabilistic analysis, the larger the multiplicity is, the more likely the edge represents a conserved k-tuple. And vise versa, the conserved segments tend to be amplified in the graph by edges of large multiplicities. Therefore, we can extract a consensus of conserved regions by traversing the graph even if the original pattern do not exist in any sequence. We then apply constrained local alignment methods to find all segments similar to the consensus and output the result as a local multiple alignment. We use a Poisson heuristic to control the false positive rate and to estimate the significance of a local alignment as well.

[1]Department of Mathematics, University of Southern California, Los Angeles, CA 90089-1113. E-mail: yuzhang@usc.edu

[2]Department of Biological Sciences, University of Southern California, Los Angeles, CA 90089-1340. E-mail: msw@usc.edu

3 Results.

We generated 3 different patterns of various lengths ($| S |= 63$, $| M |= 42$, $| W |= 21$), and inserted mutated copies of each pattern into N random sequences of length L. The number of copies inserted in each sequence follows the Poisson distribution. Table 1 shows the result.

Sequences			Patterns Inserted				Patterns Found					
N	L	identity	Total	S	M	W	Total	S	M	W	FP	FN
10	2K	90%	27	10	10	7	26	10	10	6	0	1
30	2K	90%	77	33	19	25	76	33	19	24	0	1
10	20K	90%	33	10	9	14	29	10	9	10	0	4
30	20K	90%	82	26	24	32	80	26	24	30	0	2
10	2K	80%	33	8	8	17	31	8	8	15	0	2
30	2K	80%	92	29	28	35	87	29	28	30	0	5
10	20K	80%	22	6	5	11	19	5	5	9	0	3
30	20K	80%	81	38	20	23	76	38	20	18	0	5

Table 1: Simulation results.

To test on real data, we used our program to find Alu repeats in the human genome. Five sequences of various lengths were randomly selected from NCBI database. Our results agrees well with Alus found both by the authors who submitted the sequence and by RepeatMasker, a program which knows Alu consensuses in prior. Result is shown in Table 2. As another comparison, a repeat finding program REPuter [3] fails to find many Alu repeats presented and runs slower than our method does.

ID	L	RepM	Family	Len(avg:std)	Div(avg:std)	FP	FN	Time/sec
AF435921	22Kb	29	10	261 : 69	15.0% : 6.4%	0	0	11
Z15025	38Kb	53	13	245 : 85	15.7% : 5.7%	3	2	15
AC034110	167Kb	89	18	261 : 72	12.2% : 5.9%	0	4	65
AC010145	199Kb	120	13	277 : 55	15.0% : 5.6%	1	3	134
Chr22	1Mb	717	32	252 : 79	15.2% : 6.1%	5	107*	1095

Table 2: Alu finding in the human genome. 'RepM': number of Alus marked by RepeatMasker; 'Family': number of Alu subfamilies; 'Len': Alu lengths; 'Div': percentage divergence to Alu consensuses. * A second run of our program reduces the number of false negatives to 30.

References

[1] Aldous, D. 1989. *Probability approximations via the Poisson clumping heuristic.* Springer, New York.

[2] Idury, R. and Waterman, M.S. 1995. A new algorithm for DNA sequence assembly. *J. Comp. Biol.* **2**:291-306.

[3] Kurtz, S. and Schleiermacher, C. 1999. REPuter: fast computation of maximal repeats in complete genomes. *Bioinformatics*, **15**:426-427.

[4] Pevzner, P.A., Tang, H., and Waterman, M.S. 2001. An Eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci.* **98**:9748-9753.

[5] Smit, A.F.A. and Green, P. Unpublished results.