Local sequence alignment for an associative model of parallel computation

Shannon I. Steinfadt
Department of Computer Science, Kent State University, Kent, Ohio

Introduction

The goal of sequence alignment is to find similarity between the different strings of genetic information.

similar characters → similar structure → similar function

Ancestral Relationships
Gene Functionality
Aid in Drug Discovery

Local Sequence Alignment

Optimization Problem

- Maximize sub-sequence similarity (**local alignment)**

Smith Waterman algorithm

- Exact algorithm[1], finds highest scoring local alignment
- Slow, uses dynamic programming method (DP) to get exact answer
- Heuristic algorithms such as BLAST are faster, but not always the best alignments

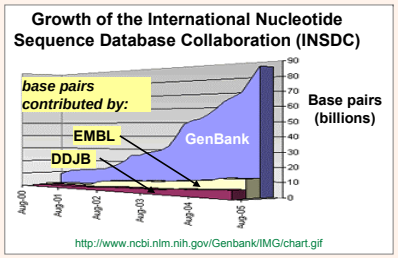


Fig. 1. Exponential growth of sequence data means more to align with; speed as well as quality of information is vital.

Parallel Alignment Goals

Produce More Information

- Return top *k*-ranked alignments
- The *k* alignments are non-overlapping / non-intersecting

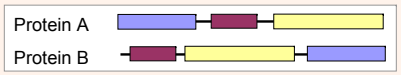


Fig. 2. Proteins with three local non-intersecting alignments.

Fast Local Alignments

- Find all *k* alignments in the same time it takes to find a single alignment

Parallel Model

The ASsociative (ASC) and Multiple ASsociative Computing (MASC) models[2] are SIMDs with an associative property and some additional hardware features.

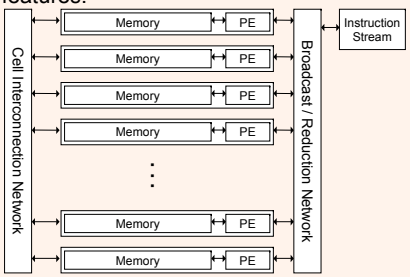


Fig. 3. A high-level view of the ASC model. The MASC model includes more than one instruction stream.

ASC / MASC features

- Fast processing
- Constant time search/respond ops
- Constant time global reduction ops: max/min of processing elements (PEs)
- A base of existing algorithms[3]
- Existing programming language and emulator

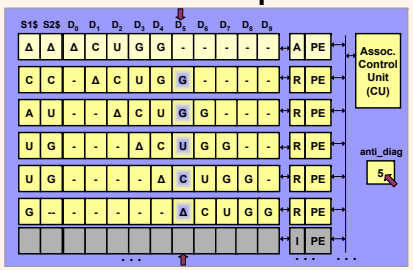
Associative Adaptation

Weights for the following examples:
Gap Insert: 3; Gap Extend: 0
$d(S1_i, S2_j) = 10$ when $S1_i = S2_j$
$d(S1_i, S2_j) = 10$ when $S1_i \neq S2_j$

PE i[$] index / S1	PE j[$] index / S2	0 Δ	1 C	2 U	3 G	4 G
0	Δ	0	0	0	0	0
1	C	0	10	7	7	7
2	A	0	7	7	4	4
3	U	0	7	17	14	14
4	U	0	7	17	14	14
5	G	0	7	14	27	24

Fig. 4. Traditional Smith-Waterman DP table. The dependency free anti-diagonal that the ASC algorithm executes in parallel is highlighted.

Associative Example



S1: CAUUG Alignment: CAUUG
S2: CUGG C- - -UGG

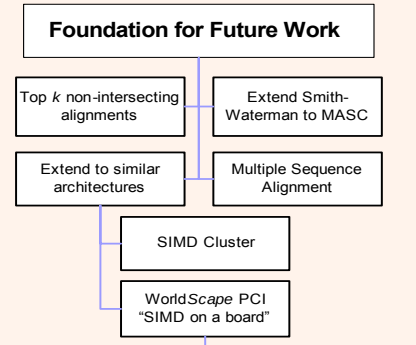
Fig. 5. Mapping the Smith-Waterman algorithm on ASC.

Active PEs hold one character of S1 and S2. D is a parallel array of size |S1| + |S2|. D's subscript represents a particular anti-diagonal.

S2 is stored in a systolic fashion to allow parallel computation of the values along the anti-diagonal.

Active PEs compare their S1 value with each particular D_j value to compute the function *d* listed above (not shown).

Foundation for Future Work



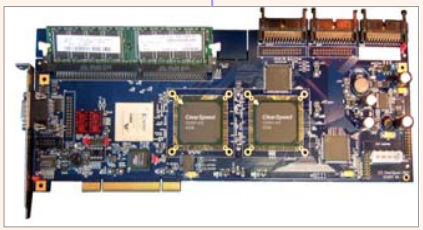


Fig. 6. World*Scape* Dual 64 PCI SIMD Board with 50 GFLOPS performance.

Intent

- Provide fast, accurate, more detailed alignments that aid in bioinformatics
- Work towards identifying regulatory regions in genes and response elements

References

[1] Goto, O. "An Improved Algorithm for Matching Biological Sequences." *J. of Molecular Biology* 162, 705-708, 1982.
[2] Potter, J., J. Baker, A. Bansal, S. Scott, C. Leangsuksun, and C. Asthagiri. "ASC: An Associative Computing Paradigm." *IEEE Computer*, 27(11): 19-25, November, 1994.
[3] Esenwein, M., J. Baker, "VLCD String Matching for Associative Computing and Multiple Broadcast Mesh", *Proc. of 9th IASTED International Conf. on Parallel and Distributed Computing Systems*, 69-74, 1997.

For further information

Please contact ssteinfa@cs.kent.edu. More information on this and related projects can be obtained at http://www.cs.kent.edu/~parallel.