

# GPU 上の 16 ビット整数表現による Smith-Waterman アルゴリズムの高速化の検討

安井 一貴<sup>1</sup> 伊野 文彦<sup>1</sup> 萩原 兼一<sup>1</sup>

<sup>1</sup> 大阪大学大学院情報科学研究科

## 1 はじめに

ローカルアライメントとは、生体配列間の局所的な類似部位を特定する操作であり、生物の進化過程の推定などに用いる。一般に、配列間の類似度を表すスコアの算出によりローカルアライメントが求まる。

配列  $a$  および  $b$  が与えられたとする。配列  $a$  における  $i$  番目の文字を  $a_i$  で表すと、文字  $a_i$  および  $b_j$  の類似度  $S_{i,j}$  を以下のように定める。

$$S_{i,j} = \begin{cases} p, & a_i = b_j, \\ q, & \text{otherwise.} \end{cases} \quad (1)$$

ここで、 $p$  ( $\geq 0$ ) は文字  $a_i$  および  $b_j$  が一致する場合のスコアを表し、 $q$  ( $< p$ ) はそれらが異なる場合のスコアを表す。

ローカルアライメントの解法として SW (Smith-Waterman) アルゴリズム [1] が知られている。SW アルゴリズムは行列生成およびトレースバックで構成されている。前者は、動的計画法により類似度を表す行列を生成し、最大スコアおよびその位置を特定する。後者は、位置情報をもとに、解となる類似部位を特定する。入力配列の長さを  $m$  および  $n$  ( $\geq m$ ) とすると、 $O(mn)$  時間を必要とする前者が実行時間を支配する。そこで、数千個ものコアを内包し、高いメモリ帯域幅を持つ GPU (Graphics Processing Unit) による高速化が試みられてきた。

SW# [2] は  $O(n)$  の GPU スレッドを用い、 $O(m+n)$  ステップで行列を並列生成している。また、全スレッドの同期を伴うステップ数を削減するために、行列をブロックごとに生成する (図 1)。SW アルゴリズムの並列性は  $O(n)$  であるため、さらなる高速化を果たすためには、メモリ参照量の削減が必要である。そこで本研究は、SW アルゴリズムにおける行列生成を高速化するために、SW# を拡張し、16 ビット整数表現によってメモリ参照量を削減する手法を提案する。

## 2 提案手法

16 ビット整数が表現できる値の範囲は、32 ビットのそれよりも小さい。したがって、行列生成時にスコアのオーバーフローが生じうる。ゆえに、各ブロックの計算前に、そのブロック内におけるオーバーフローの可能性を判定する。オーバーフローする場合、従来通りの 32 ビット整数表現を使う。そうでない場合、以下の手順にしたがい、16 ビット整数表現を用いる。

1. スコアを 16 ビット整数のオフセット表現に変換
2. 16 ビット整数でブロックを計算

## 3. スコアをオフセット表現から絶対値に変換

あるブロックにおけるオーバーフローの可能性は、そのブロック内のスコアの上界  $U$  と下界  $L$  をもとに判定できる。 $U-L$  が 16 ビットで表現できれば、オーバーフローは生じない。すなわち、(2) が真であれば対象ブロックを 16 ビット整数で計算できる。

$$U - L < (16 \text{ ビット整数で表現できる最大の値}) \quad (2)$$

上界  $U$  および下界  $L$  は、対象ブロックに至るまでの最大スコア  $s_1$  および最小スコア  $s_2$  をもとに計算できる。

$$U = s_1 + p \cdot \min(h, w) \quad (3)$$

$$L = \max(0, s_2 - q \cdot \min(h, w)) \quad (4)$$

ここで、 $h$  および  $w$  はブロックの高さおよび幅である。

図 2 に、対象ブロックとその隣接領域 (オレンジ色の領域) を示す。入力配列の文字がすべて一致した場合にスコアが最大になる (図 3)。つまり、 $U$  は入力配列のうち  $\min(h, w)$  個の文字が一致した場合のスコアである。一方、 $L$  はこれまでの最小スコア  $s_2$  を基準として、ブロック内における入力配列のすべての文字が異なる場合のスコアである。

## 3 事前評価

提案手法の有用性を調べるために、その高速化の見込みおよび 16 ビット整数表現で計算できるブロックの分布を調べた。実験には NVIDIA GeForce GTX 980 を用いた。まず、行列生成を模倣するマイクロベンチマークとしてベクトル加算の CUDA 実装を用いた結果、16 ビット版は 32 ビット版に比べて 2 倍の速度向上を得た。次に、炭疽菌 (5.2Mbp × 5.2Mbp) で 95%、ヒト 21 番染色体 (33.3Mbp × 34.5Mbp) で 81% のブロックが 16 ビット整数表現で処理できた。これらの結果から、類似部位が短ければ、16 ビット整数で処理できるブロックの割合が増え、提案手法が有効となる。

## 参考文献

- [1] T.F. Smith and M.S. Waterman, Identification of common molecular subsequences, *J. Molecular Biology*, Vol.147, No.1, pp.195-197, March 1981.
- [2] Matija Korpar and Mile Šikić, SW#-GPU-enabled exact alignments on genome scale, *Bioinformatics*, Vol.29, No.19, pp.2494-2495, October 2013.

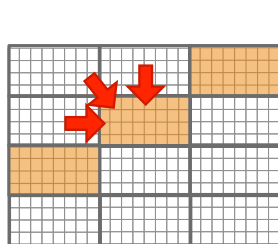


図 1: ブロックごとの行列生成

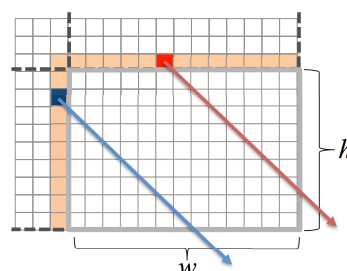


図 2: 上界および下界を与えるアライメントの例

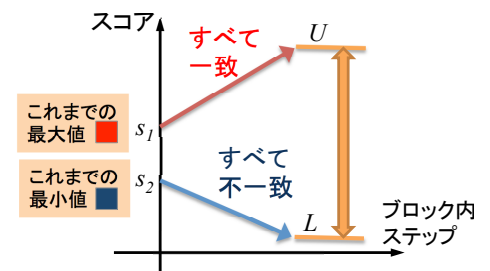


図 3: 最大および最小スコアが更新される様子