# GPU による DNA 断片配列の高速マッピング

鈴木 脩司† 石田 貴士‡ 秋山 泰†‡

## 1. はじめに

近年, 次世代シーケンサーと呼ばれるハイスループットなゲノム解読装置が登場した. この次世代シーケンサーの特徴としては数十残基程度の短い断片配列を大量に読め, 1~2 週間程度で数十億塩基が解読可能であり, 従来のシーケンサーと比べると数百倍から数千倍のスループットであることが挙げられる.

この次世代シーケンサーによって得られるデータは短い DNA 断片情報であるため, これらのデータを利用するためには配列マッピングと呼ぶ処理が必要となる. これはシーケンサーが読み取った短い DNA 断片が元の配列のどの部分のものであったかを同定する処理である. 配列マッピングを行うツールとして RMAP[1]や Bowtie[2]等がある. 単一の生物に対するゲノム解読ではシーケンサーの読み取った DNA 断片と元の配列とでは一致率が高いと考えられるため, これらのツールでは文字列の完全一致, または数残基の違いしか許容しないという特徴がある.

一方, 次世代シーケンサーを用いた他の解析にメタゲノム解析というものがある. これは土壌や腸内等に生息する複数の微生物の DNA を分離・培養を経ずに直接シーケンサーで読み取ることで, 単一の生物ではなく, ある環境における遺伝子の分布を解析しようというものである. 環境中に含まれる微生物全てのゲノム配列が既知であることは稀であるため, このメタゲノム解析では遠縁のゲノム配列しか参照できない場合が多く, 通常の単一の生物に対する配列マッピングに比べ, 配列マッピングの際に多くのミスマッチとギャップを許容する必要があり, 困難なものとなっている. このためさらに, メタゲノム解析では感度を増すために DNA 配列をタンパク質配列に変換してから解析する場合がある. DNA 配列は A, T, G, C の 4 文字で表現されるのに対し, タンパク質配列は 20 種の標準アミノ酸からなっており, 20 文字で表現される. さらに, DNA 配列の比較は一致か不一致との 2 状態しか区別しないが, タンパク質配列ではアミノ酸種間でその性質の類似性に差があるため, アミノ酸種毎に置換スコアが付されているスコア行列が用いられる. このため, ミスマッチの場合でもペナルティはアミノ酸種に依存し, これは配列マッピングをさらに困難なものとしている. このような曖昧さを許容した複雑な配列間の比較には, もっとも正確なアラインメントが可能である動的計画法を用いた Smith-Waterman アルゴリズム[3]の実装である SSEARCH[4]等を利用することが望ましいが, これは低速であるという問題がある. このため現在では, BLAST[5]という高速に配列比較が可能な近似的手法が利用されている.

しかし, 次世代シーケンサーは一度に大量の DNA 断片を出力するため, すべてをタンパク質配列に変換してから配列マッピングを行うには BLAST を用いてもまだ長い計算時間が必要となる. このため, 本研究ではメタゲノム解析にも対応可能となるように曖昧さを許容し, また, GPU を用いることで高速に実行可能となる配列マッピングのアルゴリズムを提案し, その実装を行った. これは, GPU 上で実行し易い配列マッピングのアルゴリズムを提案したものであり, 既存のツールを単に GPU 上で実装したものではない. その結果, 従来の BLAST と比較して 1 GPU を使用した場合で約 50 倍, 4 GPU を使用した場合で約 200 倍の高速化を達成した.

## 2. 手法

### 2.1 提案する配列マッピングアルゴリズム

本研究のアルゴリズムでは, まずインデックスを用いて DB に対してマッピング位置の候補を探索

† 東京工業大学 工学部 情報工学科
Department of Computer Science, School of Engineering, Tokyo Institute of Technology
‡ 東京工業大学 大学院情報理工学研究科 計算工学専攻
Graduate School of Information Science and Engineering, Tokyo Institute of Technology

し，その後，その周辺を動的計画法により詳細に探索して最適アラインメントとそのスコアを計算する．以下に詳細を示す．

#### 2.1.1 前処理

マッピングの対象となる既知のタンパク質配列の DB の中には多数のタンパク質配列が含まれている．しかし，複数の配列よりも単一の配列の方が扱い易いため，これらの配列を区切り文字を入れて連結しておく．次に連結した配列を 1 文字ずらしで K-mer(長さ K の文字列)に区切って key を生成していく．そして，key がどこの位置にあったかを key 毎にまとめて，インデックスとし，マッピング候補探索の際に利用する．

また，シーケンサーが読み取った DNA 断片をタンパク質配列に翻訳しておく．

#### 2.1.2 マッピング候補探索

まず，クエリ(問い合わせ配列)を s 文字ずらしで K-mer に区切り，これを key とする．

ここでクエリでの key の出現位置を縦軸，DB での key の出現位置を横軸として，クエリと DB 間で，同じ key が出現する箇所に印をつけていく．クエリと DB 間で高い一致率がある領域ではこの印がいくつか同一斜線上付近に並ぶ．これを利用して DB を r という大きさの領域で区切り，同領域と右隣の領域内の key 数の和がある閾値 t 以上となった領域をマッピング候補とする．図 1に閾値 t が 2 の場合を示す．

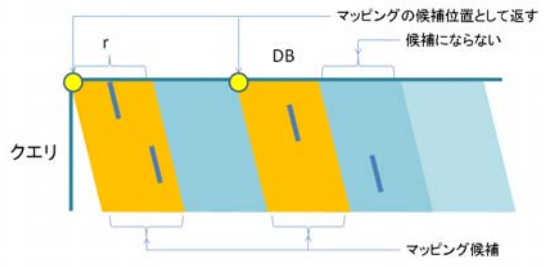


図 1 閾値が 2 の場合の候補探索

#### 2.1.3 アラインメントによるスコア計算

マッピング候補探索によって得られたマッピング候補位置の周辺を Smith-Waterman アルゴリズムによって最適なアラインメントとそのスコアを計算し，マッピング候補位置のスコアとする．

マッピング候補探索によってマッピング位置が大まかに分かっているため，アラインメントを計算する領域の幅は図 2のように m+2r+2e とした．

ここで m はクエリの長さ，r は探索のときに用いた r の値，e はアラインメントの計算を行う領域の幅を拡張する長さである．こうすることで少なくとも r+e 個ギャップがあったとしても正しいアラインメントが計算できる．

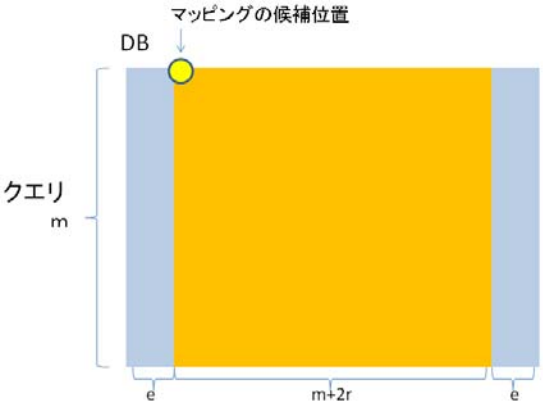


図 2 アラインメントを計算する領域

### 2.2 GPU による実装

前節のアルゴリズムを CPU だけで実行した場合，マッピング候補探索とアラインメントによるスコア計算の箇所で多くの計算時間が必要となることが判った．このため，本研究ではこの 2 か所を GPU によって処理することとした．

### 2.3 マッピング候補探索の GPU 化

マッピング候補探索では予めどの程度のマッピング候補位置が出力されるかわからない．このため，一度ですべてのマッピング候補位置を計算しようとするとマッピング候補位置の個数が GPU の global memory の大きさの限界を超える可能性がある．

このため，本研究では以下のように 2 段階に分けてマッピング候補探索を行うようにした．

1. クエリ毎にマッピング候補の数をカウント
2. GPU の global memory を超えない数ずつマッピング候補位置を計算し，すべての候補位置を計算するまで繰り返す

このようにすることで GPU 上でも問題なくマッピング候補探索を実行することができる．

### 2.4 アラインメントによるスコア計算の GPU 化

Smith-Waterman アルゴリズムの GPU 化は既に SW-CUDA[6]等が提案されている．しかし，ほとんどの場合，ある程度長い配列同士のアラインメントを複数の thread を同期させながら計算していくという手法である．しかし，今回のように短い配列のアラインメントでは同期の時間の割合が増えてしまって計算速度が出ない．

このため，本研究ではマッピング候補位置毎に 1 つの thread を割り当てて，計算を実行するようにした．また，スコア行列はすべての thread が何度もアクセスするため本研究では texture memory に配置した．こうすることで cache の効果により global memory に配置するよりも高速にアクセスが可能となっている．

## 3. 実験結果

本研究で提案した DNA 断片配列を DB 内のタンパク質配列にマッピングする新システムの実験を行った．以下に実験の手順と結果について述べる．

### 3.1 計算機環境

本研究で構築した新システムを実行した計算機環境を表 1に示す．

表 1 計算機環境

| | |
|---|---|
| CPU | Opteron 2224 SE (3.2GHz) |
| Memory | 16GB |
| GPU | Tesla S1070(1.44 GHz) |
| CUDA の version | 2.2 |

### 3.2 使用データ

本研究では既知タンパク質配列の DB としては京都大学 KEGG[7]の Web サイト[8]から 2009 年 4 月 30 日にダウンロードした genes.pep というタンパク質配列データを使用した．この DB の中の配列の本数は約 423 万本，全配列の合計長は約 15 億残基である．また，クエリとなる DNA 断片配列は東京工業大学 大学院生命理工学研究科 生命情報専攻の黒川顕教授より 60 塩基の配列 10 万本を頂き，速度比較用としてこれをそのまま使用した．また既存法(BLAST)との精度比較用として 10 万本の中からランダムに選んだ 2,000 本を使用した．

### 3.3 新システムで使用した値

今回の実験で使用した変数(2章参照)の値を示す．

- K-mer の K の値は 4
- クエリを K-mer で区切る際に何文字ずらしに区切っていくかを指定する s の値は 2
- マッピング候補探索の際に，いくつ key があれば候補にするかを指定する閾値 t の値は 2
- マッピング候補探索の際に，DB を区切る領域の大きさ r の値は 8
- アラインメントのスコア計算の際，アラインメント領域の拡張幅 e の値は 2

また，アラインメントのときのスコア行列は PAM30，ギャップのペナルティは-8 を使用した．

### 3.4 既存法(BLAST)のオプション

今回使用した BLAST のオプションは以下の通りであり，ギャップペナルティ等は新システムと同じものを用いている．

%blastall –p blastx –G 8 –E 8 –g T –F F –e 50 –M PAM30

### 3.5 マッピング精度の比較

マッピング精度を比較するために，既存の近似計算手法(BLAST)と，提案手法との間で，クエリ毎にマッピング位置とアラインメントの正確さを評価した．正解には，厳密な動的計画法を行う SSEARCH の結果を用いて，両手法の結果に SSEARCH が与える正解が含まれる割合を比較した．図 3に結果を示す．ここで変化させた E-value とは，現在の DB において，偶然同じスコアになる配列の数の期待値を示している．つまり，E-value が小さいほど偶然には起こりえないことを示している．結果としては BLAST と新システムとでは新システムの方が若干精度が落ちるものの，E-value が十分に低い領域では BLAST とほぼ同等の精度が得られた．

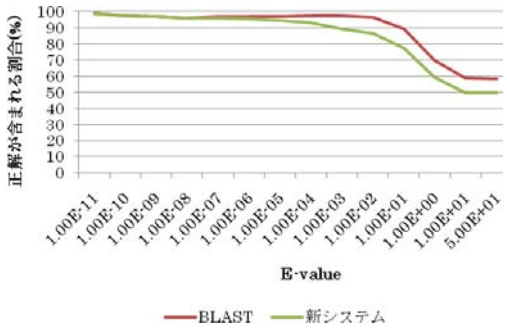


図 3 E-value を変化させた場合の正解が含まれる割合(%)

### 3.6 計算速度の比較

速度比較には60塩基のDNA配列10万本を用い，計算時間を測定した．測定の対象は，新システムのCPUのみを使用した場合，1 GPUの場合，4 GPUの場合，既存法(BLAST)の 1 thread の場合と 4 thread の場合の計 5 つとし，それら全ての計算時間を測定した．

BLASTの1 threadの場合の計算速度を1とした場合の速度向上比を図 4に示す．結果としては，1 GPUではBLASTの1 threadと比べると約50倍，4 GPU では約 200 倍の高速化が達成できた．

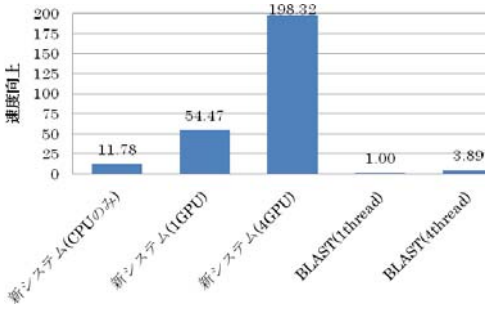


**図 4　計算速度比較**
**（BLAST 1 thread 版からの速度向上比）**

## 4.　考察

E-value が高いヒットを許容する条件で用いると，新システムの精度はBLASTよりも落ちているが，実際はE-valueが高いヒットは偶然得られる可能性があるため，大量データ解析の場合にはあまり使われない．このため，新システムは実用上，十分な精度が出ていると考えられる．

また，計算速度については，BLAST の 1 thread と比較して 1 GPU の場合で約 50 倍，4 GPU で約 200 倍高速になった．ここで 1 GPU と 4 GPU とを比較すると 4 GPU は 1 GPU よりも約 3.6 倍高速になっている．4 倍にならない理由としては，GPU で計算した後，結果をファイルへと書き込む際の I/O が衝突しているためであると考えている．

## 5.　結論

### 5.1 本研究の成果

本研究ではメタゲノム解析にも対応可能な曖昧さを許容し，GPU を用いることで高速に実行可能となる配列マッピングのアルゴリズムを提案し，Tesla 上でこの実装を行った．

実装した新システムではマッピングの精度は既存手法の BLAST よりも若干落ちるものの，E-value の低い領域ではほぼ同等の精度が得られ，実用上，十分な精度を持っていることが示された．計算速度においてはBLASTの 1 thread 版と比較すると 1 GPU の場合は約 50 倍，4 GPU の場合は約 200 倍の高速化を達成できた．

### 5.2 今後の課題

今後，DNA シーケンサーから出力されるデータは技術の発展により，さらに増大すると考えられ，また，読み取られる DNA 断片の長さも長くなっていくと考えられる．このため今よりもさらにマッピングを高速化する必要がある．また，現在の新システムではシーケンサーが読み取った短い断片配列を対象としているが，これをさらに長い断片配列でも同様に高速に計算できるようにすることが将来的には求められる．

## 参　考　文　献


[1] Andrew D Smith,Zhenyu Xuan,Michael Q Zhang: "Using quality scores and longer reads improves accuracy of Soleza read mapping", BMC Bionfomartics 9,128, (2008) .

[2] Langmead B, Trapnell C, Pop M, Salzberg SL: "Ultrafast and memory-efficient alignment of short DNA sequences to the human genome", Genome biology, 10,25, (2009) .

[3] Smith TF, Waterman MS: "Identification of Common Molecular Subsequence", Journal of Molecular Biology, 147: 195-197, (1981).

[4] Pearson WR., Searching protein sequence libraries: comparison of the sensitivity and selectivity of the Smith-Waterman and FASTA algorithms. Genomics. 3:635-650,(1991) .

[5] Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: "Basic local alignment search tool", Journal of Molecular Biology, 215: 403-410, (1990) .

[6] Manavski SA, Valle G: "CUDA compatible GPU cards as efficient hardware accelerators for Smith-Waterman sequence alignment", BMC Bioinformatics, 9:26, (2008) .

[7] Kanehisa, M., Goto, S., Furumichi, M., Tanabe, M., and Hirakawa, M.; KEGG for representation and analysis of molecular networks involving diseases and drugs. Nucleic Acids Res. 38, 355-360 (2010).

[8] KEGG, http://www.kegg.jp/