6K-08

高速に類似文字列ペアを発見するビット並列フィルター

山田太樹† 清水佳奈†

†早稲田大学基幹理工学部

1 序論

編集距離の近い文字列ペアの発見は，ゲノム情報解析や時系列情報の解析等，様々な分野において重要な課題である．例えばゲノム情報解析では，ゲノム配列のクラスタリングやアセンブリ（ゲノム配列の断片をつなげて一つの配列を復元するタスク）を行う際に，大量の編集距離を計算する必要がある．このような背景から本研究では，長さがそれぞれ n, m（ただし，$n \leq m$ とする）の文字列が与えられた場合に，編集距離が閾値 θ 以内であるか否かを高速に判定する手法の開発を目的とする．編集距離そのものを高速に計算する従来研究では，Myers によるワード長 w のとき $O(nm/w)$ を達成するビット並列アルゴリズムが知られている [1]．一方で，文字列のセットが与えられた場合にそれらの中から類似するペアを発見する手法としては SlideSort 法 [2] や ED-Join 法 [3] が知られている．SlideSort や ED-Join では，実際に用いられるデータセットが疎であることに着目し，類似性の低いペアを計算量の低い方法によって予め除外し，編集距離の計算回数を減らすことで全体の計算量を大幅に向上させている．これらの手法が対象としている全ペア列挙問題の他にも，特定のパターンとの編集距離が近いデータの探索などでは，類似性の低いペアを評価する回数が多くなりがちであるため，類似文字列ペア発見の問題に対して直接的に編集距離を計算する Myers の手法が適しているとは言い難い．そこで本研究では，ED-Join 法や SlideSort 法で用いられている鳩ノ巣原理を応用して，編集距離が遠いペアを計算量 $O(\theta n/w)$ で判定することのできるビット並列フィルタ F を構築し，F が受理したペアに対してのみ編集距離を計算する手法を提案する．

2 提案手法

文字列 s の i 番目の文字を $s[i]$ と表す．s と t の編集距離 $Editdist(s,t)$ について，以下の定理が成り立つことが知られている [2].

定理. 等しい長さ l の文字列 s と t について，$Editdist(s,t) \leq \theta$ であるなるならば，

$$s[p] = t[q],\ q \in \{p - \lfloor \theta/2 \rfloor, \ldots, p + \lfloor \theta/2 \rfloor\}$$

を満たす q が存在するような p が少なくとも $l - \theta$ 個存在する．

上記定理で示される必要条件を満たさないペアの編集距離は必ず θ よりも大きくなるため，編集距離の計算を行わなくとも類似性の低いペアと判断することができる．説明のため，$\theta = 2$ における具体例を**図 1** に示す．例では，$\theta = 2$ より，少なくとも $5 - 2 = 3$ 箇所において s がオフセット幅 1 以内で t と一致すればよい．左の s と t の組では，4 つの一致が存在するため必要条件を満たす．一方，右の組では 2 つの一致しか存在しないため必要条件を満たさない．よって，右の組については編集計算の計算を行わずとも，非類似ペアと判断することができる．このように，上記定理は類似ペアを発見する際のフィルターとして利用することが可能である．本研究では，上述の必要条件を高速に計算するビット並列アルゴリズムを提案し，多数のペアを評価する際のならし計算量を削減することを目指す．

図 1: $\theta = 2$ の時の s と t の一致する文字の組の例

2.1 フィルタ F の基本構築方法

まず，オフセット $j \in \{-\lfloor \theta/2 \rfloor, \ldots, \lfloor \theta/2 \rfloor\}$ について，s と t の一致を示すビットベクトル

$$F_j[i] = \begin{cases} 1 & (s[i] \neq t[i+j]) \\ 0 & (s[i] = t[i+j]) \end{cases}$$

を求める．これは，以下により計算できる．

$$F_j = (s \oplus (t >> j))$$

ただし，$\oplus$ は排他的論理和，$t >> j$ は配列 t を j の幅で右シフトする演算を示す．F_i の論理積

$$M = F_{-\lfloor \theta/2 \rfloor} \wedge \ldots \wedge F_{\lfloor \theta/2 \rfloor}$$

における各ビットは，s と t がオフセット幅 $\lfloor \theta/2 \rfloor$ 以内で一致するか否かを示す．このため，M に含まれる 1 のビットの数が θ 以下であれば，定理を満たす．具体例を**図 2** に示す．例では s の 2 文字目と 4 文字目がそれぞれオフセット幅 1 以内で t に一致するので, M の対応するビットは 0，それ以外は 1 となる．

2.2 ビット列のブロック化による性能の改良

2.1 の手法では，文字の種類が少ない場合に偶然の一致が多くなり，編集距離が遠いのにもかかわらずフィルターに受理されてしまうペアが増加する問題がある．そこで，一文字の一致ではなく，長さ k の文字列ブロッ

Efficient bit-parallel filtering algorithm for finding similar string pair in edit distance

Taiki YAMADA†and Kana SHIMIZU†

†School of Fundamental Science and Engineering, Waseda University

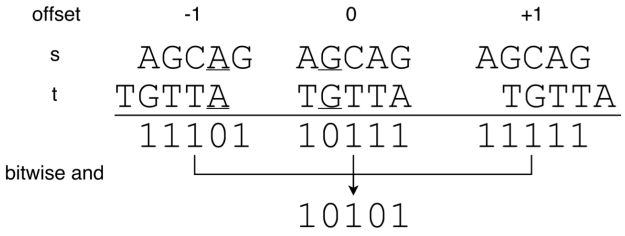

図 2: $\theta = 2$ におけるビットベクトル M の計算の例

クが一致する個数を数える工夫により，フィルターの性能を向上させる．それぞれの j に対して，

$$F'_j = F_j \vee (F_j << 1) \ldots \vee (F_j << (k-1))$$

を計算すると，位置 p において連続する k 文字が一致する場合は 0, 一致しない場合は 1 となる．ここで，先ほどと同様にすべての F'_j の論理積をとったビット列 M' を得ると，それぞれのビットは連続する k 文字がオフセット幅 $\lfloor \theta/2 \rfloor$ 以内で s と t が一致するか否かを示す．ここでブロックの区切り方を表現するビットベクトル

$$E_b[i] = \begin{cases} 1 & (i\%k = b) \\ 0 & (i\%k \neq b) \end{cases}$$

を用意すると，条件

$$max(CountTrue(M' \wedge E[b])) \leq \theta$$

によって，$Editdist(s,t) \leq \theta$ となるペアの候補を抽出できる．ただし $CountTrue(s)$ はビット列 s に含まれる 1 の個数を数える演算とする．これは，ブロックの区切り方の違う k 種類のフィルターを同時に評価していることに相当することに注意されたい．フィルターの計算はビット並列に行うことが可能であるので，時間計算量はブロック長を k として $O(k\theta l/w)$ となる．以上をふまえたフィルター計算の擬似コードを **Algorithm 1** に示す．

Algorithm 1 フィルタ計算の擬似コード

```
M' ← 1^l
for j = −⌊d/2⌋ to ⌊d/2⌋ do
  F_j ← (s ⊕ (t >> j))   F'_j ← 0^l
  for b = 0 to k − 1 do
    F'_j ← F'_j ∨ (F_j << b)
  end for
  M' ← M' ∧ F'_j
end for
result ← 0
for b = 0 to k − 1 do
  result ← max(result, CountTrue(M' ∧ E_b))
end for
```

3 実験

長さ 75 文字の 10000 個のゲノム配列から編集距離 3 以下のペアを列挙する実験を行い，以下の 3 種類の手法の性能を比較した．

- 提案手法（フィルター F を計算し，F を通過したペアのみに対して Myers の手法で編集距離を計算する手法）
- $Na\ddot{i}ve$（全ペアについて $O(nm)$ の動的計画法により編集距離を計算する手法）
- $Myers$（全ペアについて Myers の手法により編集距離を計算する手法）

提案手法のブロック長 k の値は $k = 3$ とした．プログラムは C++で実装し，Intel Core i5(2.7GHz)，8GB RAM を搭載した PC 上で実行した．それぞれのプログラムについて，編集距離 (EditDist) の計算を行った回数，編集距離計算にかかった CPUTime[sec]，全体の CPUTime[sec] を計測した．実験データセットのうちで，実際に編集距離が 3 以下であったペアの数は 14212 個であった．結果を**表 1** にまとめる．提案手法は，単純に動的計画法を計算する場合と比較しておよそ 120 倍， Myers の手法により計算する場合と比較しておよそ 20 倍高速であった．また，提案手法のフィルターにより，編集距離の計算回数が，およそ 0.3% になっていることが確認された．

表 1: 実験結果

プログラム	ED 計算回数	ED 計算時間	総計算時間
$Na\ddot{i}ve$	49995000	705.542	705.542
$Myers$	49995000	118.928	118.928
提案手法	150448	0.368	5.780

4 結論

本研究では，類似ペアの発見を行う際に効率よく遠いペアを取り除くことのできるビット並列アルゴリズムを提案した．実験結果より，編集距離を直接的に計算するビット並列アルゴリズムと比較して，大幅にならし計算量を削減することが確認できた．今回の実装ではワード長を 64bit としたが，SIMD 命令を用いて実装することにより，長い文字列に対する並列性能を高めることができる．また，提案手法は文字列長が異なるケースにも拡張することができるため，様々な問題への応用が期待できる．

参考文献

[1] Myers, Gene. "A fast bit-vector algorithm for approximate string matching based on dynamic programming." Journal of the ACM (JACM) 46.3 (1999): 395-415.

[2] Shimizu, Kana, and Koji Tsuda. "SlideSort: all pairs similarity search for short reads." Bioinformatics 27.4 (2010): 464-470.

[3] Xiao, Chuan, Wei Wang, and Xuemin Lin. "Ed-join: an efficient algorithm for similarity joins with edit distance constraints." Proceedings of the VLDB Endowment 1.1 (2008): 933-944.