Vol.2012-AL-138 No.3
2012/1/28

ゲノムシーケンスにおける反復部位識別アルゴリズム

伊佐英寿[†1]　岡崎威生[†2]　名嘉村盛和[†2]

バイオインフォマティクスにおける DNA シークエンシングとは、塩基配列を決定することであり、遺伝情報を解析するための基本手段である。塩基の読み取り長を短くし高速な読み取り処理が可能となった反面、塩基配列中の反復部分が、読み取られた塩基配列を結合する過程（DNA アセンブル）で元の配列を復元することが困難になるという問題が生じた。反復部分を識別する手法として k-mer が利用されているが、k の値に識別の精度が依存している。そこで本報告では、複数の k-mer を利用した反復部分識別精度向上アルゴリズムを提案する。

Identification algorithm of repetitive sites for genome sequence

HIDEHISA ISA,[†1] TAKEO OKAZAKI[†2]
and MORIKAZU NAKAMURA[†2]

In bioinformatics, DNA sequencing is nucleotide sequence and determining a major method of analysis for gene code. With development of second generation sequencer technology, it has become possible to get massive short-read sequences speedly. The repetitive sites occur misassembly because of short lentgth sequences. Thus, it is important to identify repetitive sites for sensitive assenbly. k-mer is used to identify, but the results of identification depend on k value. In this regard, we proposed a robust identification method with a way of multiple k-mer processing for accuracy improvement.

†1 Information Engineering Course, Graduate School of Engineering and Science, University of the Ryukyus
†2 Faculty of Engineering, University of the Ryukyus

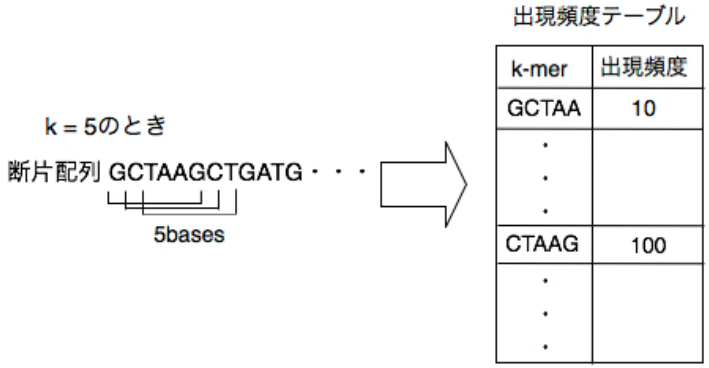

図 1　$k = 5$ の場合の k-mer の例。断片配列の始端から 5base 配列を断片配列終端の 5base まで 1base づつずらし k-mer の頻度情報をテーブルとして格納する。

1. 研究背景

DNA アセンブルとは、シークエンサーにより得られた断片配列同士を結合し元の配列を得ることである。アセンブルアルゴリズムとして代表的な Velvet[1] アルゴリズムでは、シークエンサーから得られた断片配列それぞれを k-mer で分割し、k-mer をノードとしたグラフを生成する。ここで、k-mer とは定数長の配列のことであり、読み取り配列長より短く設定される。k-mer の利用により、シークエンサーの引き起こす読み取りミスに対応することが可能となった。図 1 に読み取り配列に対して k-mer を 1base ずつずらした配列と読み取り配列内での出現頻度情報を格納したテーブル作成の例を示す。

生成したグラフからオイラーパスを探索し、巡回順に配列を結合し、元の配列を復元する。Velvet アルゴリズムでは、グラフでループする箇所の情報と、k-mer の頻度情報を用いて反復部分を識別している。しかし、反復部分を含む断片配列が多く存在する場合には、アセンブルが困難になってしまう。このことより、反復部分を含んだ部分配列を識別し取り除くことが重要となる。

ここで、反復配列には大きく分けて縦列反復配列と散在性反復配列の二種類に分類される。縦列反復配列は、同じパターンの配列が連続してゲノム上で反復している配列に対して、散在性反復配列はゲノム上に散在して反復している配列である。

縦列反復配列では、反復領域のみが正しく構築されないアセンブルへの影響が考えられ

る。一方で、散在性反復配列はあるパターンの配列が別の領域に存在するために、同種の散在性反復配列の一部を含んでいる別領域の断片配列まで正しく構築されない影響が考えられる。元の配列を構築する再現性という観点では、散在性反復配列はアセンブルへの影響が縦列反復配列よりも大きいと考えられる。

本研究では、散在性反復配列を識別することを目的とした反復部位識別アルゴリズムを提案する。

2. 反復部分識別における k-mer の役割と問題点

一般的な反復部分識別の方法として Repeatmasker[2] がある。Repeatmasker は反復部分の事前情報と入力された配列とを照らし合わせ反復部分をマスクするツールである。しかし、未知の生物には反復部分の情報が無いため、Repeatmasker では適応できないという問題があった。その一方で、k-mer を利用した識別方法である WindowMasker[3] WindowMasker は配列の相同性検索を行う際に、生物学的に意味の無い大量のマッチングが発生してしまう問題を解決するために開発された。

WindowMasker では入力された配列を基に k-mer の頻度と累積頻度分布を用いてしきい値を決定し反復部分の識別を行う。WindowMasker は２つの処理を用いて反復配列識別を行う。それぞれの処理を first pass、second pass とし以下にその詳細を示す。

[1] first pass 配列中の k-mer の頻度、反復部分を識別するためのしきい値を計算する。

(1) L をすべての配列の長さの和とし、$\frac{L}{4^k} < 5$ を満たす最大の k を決定する。

(2) 配列を読み取り配列内で出現する k-mer s の頻度 $freq(s)$ を求める。

(3) カットオフ値 $T_{threshold}$ 、 T_{extend} 、 T_{low} 、 T_{high} を計算する。ここで全 k-mer の数を C とすると、$T_{threshold}$ は以下の式を満たす最大の値となる。

$$size(\{S|freq(S) \leq T\}) \leq 0.995 \times C \quad (1)$$

つまり、全 k-mer の 99.5% が $T_{threshold}$ 以下の頻度となるように値を設定する。同様に $T_{extend}, T_{low}, T_{high}$ は 99.0%,90.0%,99.8%で求める。

以上より、 k-mer s のスコア $k{-}mer\_score(s)$ を以下のようにして決定する。

$$k{-}mer\_score(s) = \begin{cases} T_{high} & \text{if } freq(s) \geq T_{high}; \\ \lceil T_{low}/2 \rceil & \text{if } freq(s) \leq T_{low}; \\ freq(s) & \text{otherwise.} \end{cases}$$

k-mer の頻度 $freq(s)$ が T_{high} より高い値を取った場合は $k{-}mer\_score(s)$ の値は T_{high} とする。これは、ある k-mer の出現頻度が他の k-mer よりも著しく高い場合の影響を抑えることを意味している。k-mer の頻度 $freq(s)$ が T_{low} を下回った場合は $\lceil T_{low}/2 \rceil$ とする。これは、k-mer の頻度が低い値をまとめて計算コストを抑えるためである。

[2] second pass 全ての配列に対してマスクする領域を決定する。

(1) 断片配列内のサイズ $k+4$ のウィンドウ $W_i = a_l \cdots a_{l+k+4}$ を左から順次読み取り、以下の式を用いて各ウィンドウのスコア $win\_score(W)$ を計算する。

$$win\_score(W) = \sum_{i=l}^{l+4} k{-}mer\_score(a_j \cdots a_{i+k-1})/5 \quad (2)$$

(2) 求めた $win\_score(W)$ と first pass で求めたしきい値を比較して、$T_{threshold}$ よりも高いウィンドウをマスクする。

(3) ２つの連続するウィンドウ間について、$win\_score(W)$ を計算し、その値が T_{extend} より高い値の場合、ウィンドウ間の領域をマスクする。

以上が WindowMasker の反復部分識別の手続きである。

ここで、事前に与える k の値が識別精度に及ぼす影響を調べるために、k の値を変え WindowMasker を実行した。データは、ゴリラの DNA 配列の一部へ反復部分を挿入し、擬似的に断片配列を生成して用いた。性能評価には $sensitivity$ 式 (3) と $specificity$ 式 (4) を用いた。

$$sensitivity = \frac{TruePositive}{TruePositive + FalseNegative} \quad (3)$$

$$specificity = \frac{TrueNegative}{TrueNegative + FalsePositive} \quad (4)$$

図 2 に結果を示す。図 2 より k の値が反復部分識別の精度に大きく影響しているとわかる。式 (3) 中 $TruePositive$ とは、反復部分を含んだ断片配列を反復部分を含んだ配列と正識別した配列数である。$FalseNegative$ とは、反復部分を含んだ断片配列を反復部分を含んでいない断片配列と誤識別した配列数である。式 (4) 中 $TrueNegative$ とは、反復部分を含んでない断片配列を反復部分を含んでいない配列と正識別した配列数である。$FalsePositive$ とは、反復部分を含んでない断片配列を反復部分を含んだ配列と誤識別した配列数である。$sensitivity$ と $specificity$ の両方の値が大きいことが望ましいが、結果より、$sensitivity$ の値が高くなる場合、反復部分を含んだ断片配列の識別精度が高くなるが、反復部分を含んでいない断片配列の識別精度は低くなることがわかる。一方で、$specificity$ の値が高くな

Vol.2012-AL-138 No.3
2012/1/28

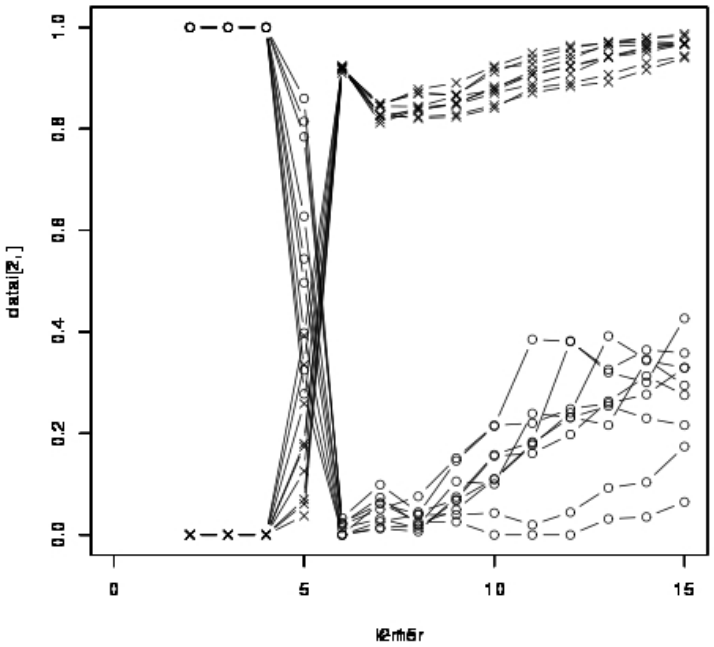

図 2 k の値の変化が反復部位識別の精度に与える影響を調べた結果を示す。
x 軸は k の値を、 y 軸は $sensitivity$(x)、$specificity$(○) である。WindowMasker では $k \leq 15$ で反復配列をマスクする手法であるため、k の値を 2 から 15 まで変化させ性能評価を行った。

る場合、反復部分を含んだ断片配列の識別精度が低くなるが、反復部分を含んでいない断片配列の識別精度は高くなる。このように $sensitivity$ $specificity$ の間にはトレードオフの傾向が見られる。図 2 において、高い k において $sensitivitiy$ の値は高いが $specificity$ の値はばらついている。これは、 k の値の選び方によっては、反復と判定された配列に非反復列が含まれてしまうことを示している。また k が 5 以下の場合は $sensitivity$ が 1 に近い値で $specificity$ が 0 に近い値となっている。すなわち、配列を反復配列と識別している。一方で各 k の値は各生物種に依存するとされていおり、生物種が未知の場合は精度の高い識別を行うことは難しい。そこで本研究では、生物種の違いに影響されない反復配列識別を行うために、それぞれの k の値によって得られる識別結果の特徴を用いて、識別の信頼性を向上させるためのアルゴリズムを提案する。

3. 複数の k を用いた反復部位識別

WindowMasker を用いた反復部位識別では、適切な k の値が事前にわからないため k の値によっては反復部位を誤識別してしまう可能性がある。一方である特定の k の値では正しく識別される可能性もある。多くの k を適用し、一つでも反復部位と識別した結果を取ることで高い検出力を獲得できると考えられるが、この方法では非反復部位を反復部位と識別する危険性が想定される。そこで、反復部位として識別された結果の個数を識別の信頼度として使用することで反復部位識別精度を向上させることを試みる。

断片配列集合に対して $k_1, k_2, \cdots, k_N$ の値を用いて反復部位を識別する。このとき、WindowMasker は反復部位の位置情報を配列へ付加して識別結果としている。これを用いて、WindowMasker によって反復部位の位置情報が付加されている断片配列に 1 を、そうでない場合に 0 を割り当てる。

$$d_{k_i}(s_j) = \begin{cases} 1 \text{ 配列 } s_j \text{が WindowMasker}(k_i\text{-mer}) \text{ により反復部位を含むと判定されたとき} \\ 0 \text{ その他} \end{cases}$$

各 k_i での結果を要素とするベクトルにより各断片配列のスコアとする。

$$D(S_j) = \begin{pmatrix} dk_1(s_j) \\ dk_2(s_j) \\ \vdots \\ dk_N(s_j) \end{pmatrix} \tag{5}$$

反復部位と識別した結果の数を信頼度として用いるために、このスコアの 1-ノルムを計算する。計算した 1-ノルムが信頼度を反映させた値 d_0 を上回る値の場合に反復部位を含むと判断することとする。

以上のことをまとめると以下の手順になる。

入力：シークエンサーに読み取られた断片配列グループ $\{s_j\}$

出力：反復部位を含む断片配列群

(1) N 個の k-mer それぞれで WindowMasker を利用した反復部位識別を実行する。

(2)　各 k-mer の結果に基づきスコアベクトル $D(s_j)$ を生成する。
(3)　$||D(s_j)||_1 \geq d_0$ となる断片配列を抽出し出力とする

4. 提案法の検証実験

識別に用いる d_0 の値の変化が、識別の精度と抽出される反復部位にどのような影響を与えるかを検証するための実験を行った。利用した配列は Gen Bank から引用した Homo sapiense 、Musmusculus　、Bison bonasus mitochondrion の 3 種である。これらに対して以下の手順により実験データを生成した。
(1)　反復配列を最低 100b の間隔を開けランダムに DNA 配列へ挿入する。
(2)　配列始端から 70b 毎に 35b の断片配列のランダムサンプリングを 10 回行う。
(3)　反復配列の挿入回数を 2 回から 10 まで (1) ,(2) の手順を繰り返す。
以上を繰り返し適用し、 各 10 セットのデータを生成した。挿入した反復配列は SINE(100-300)、LINE(1500-2000)、LTR(両端に反復配列) の 3 種である。WindowMasker での反復部位を識別するしきい値を決定するためのパラメータを 99.5% とした。

各データに対して、反復部位と識別された配列数と、正解率を計算し、それらの生物種ごとの平均値を評価値として用いた。表 1、表 2、表 3 に Homo sapiense 、Bison bonasus mitochondrion 、Musmusculus での結果を示す。

表 1 と表 2 より、Homo sapiense と Bison bonasus mitochondrion は d_0 の変化に対する正解率の挙動は類似している。正解率は $d_0 = 7, 8$ 付近で最大となる。しかし表 2 を見てみると、信頼度として用いられている d_0 が高い値のときの正解率が低下していることがわかる。これはには、断片配列同士のオーバーラップ部分の頻度が反復部分の頻度より多いことで識別結果の精度を下げている可能性が考えられる。WindowMasker では、反復部位の頻度情報を基に識別をしている。断片配列同士のオーバーラップ部分が多いということは、共通する並びの配列が多いということである。このことによって、本来反復部位とは無関係な並びの配列が反復部位の頻度情報よりも頻度が大きくなった状況が原因と考えられる。抽出配列数は各生物種で類似しており、 $d_0 = 3$ のとき最大となる。得られる配列数が少なくても精度が高い結果が望ましい場合においては $d_0 = 7$ 又は 8 を使い、より多くのマスキングを行いたい場合は $d_0 = 3$ を使うといった使い分けが考えられる。

5. ま と め

本研究では生物種、特定の生物種に依存しない反復部位識別を行うために、複数の k の値を用いた識別アルゴリズムを提案した。提案手法を実際のデータに対して適用し、識別に用いるしきい値の変化に対する識別精度の挙動を確かめた。非反復部位の識別に関しては当日報告する。

d_0	2	3	4	5	6	7	8	9	10
正解率	0.81093	0.86287	0.86371	0.86356	0.86321	0.86278	0.86017	0.85788	0.85832
抽出配列本数	105439	268477	255659	244148	231107	215284	185684	109489	29771

表 1　Homo sapiense での実行結果

d_0	2	3	4	5	6	7	8	9	10
正解率	0.65160	0.85784	0.86156	0.86250	0.86240	0.86214	0.85896	0.85251	0.84113
抽出配列本数	104537	266677	253505	241032	227463	211786	180563	95821	27409

表 2　Bison bonasus mitochondrion での実行結果

d_0	2	3	4	5	6	7	8	9	10
正解率	0.19005	0.55839	0.61204	0.66870	0.72496	0.76075	0.76037	0.62803	0.35432
抽出配列本数	110922	381875	368420	355040	338454	310632	238455	87674	19218

表 3　Mus musculus での実験結果

参 考 文 献

1) ”Daniel R. Zerbino and Ewan Birney ” Velvet:Algorithms for de novo short read assembly using de Bruijn graphs, Genome Research, vol.18, pp.821- 829, (2008)
2) ”Smit,A.F.A. and Green,P. ” Repeatmasker, (1996).
3) ”Aleksandr Morgulis, E. Michael Gertz, Alejandro A. Schaffer and Richa Agarwala ” WindorMasker:window-based masker for sequenced genomes, vol. 22, pp.134-141, (2006)