Vol.2022-BIO-69 No.8
2022/3/10

結合エネルギーを考慮した ゲノムワイドな高速短鎖核酸配列検索手法の開発

山崎 眞拓[1]　伊澤 和輝[1]　平田 稜[2]　柳澤 渓甫[1]　大上 雅史[1]　秋山 泰[1,a)]

概要：核酸配列間における結合エネルギーは核酸同士の結合安定性を考慮する上で重要な要素である．さまざまな手法により結合エネルギーの計算が行われているが，一方の配列が短鎖であった場合に効率的かつ網羅的な計算を試みる手法は存在していない．本研究ではクエリ配列をアンチセンス (ASO) 医薬品のギャップ領域として利用される長さである 10 塩基程度の短いものと想定し，結合エネルギーが強くなる配列を列挙しターゲット配列から検索する手法の提案，実装，性能比較を行なった．結果，クエリ配列長が 10–14 塩基のとき既存手法と比較して 60–90 倍の高速化に成功した．また，メモリ使用量についても 1.4–116.6MB となり，サーバ等の大規模な計算環境でなくても本手法が十分に実行可能であることを示した．

キーワード：結合エネルギー計算，アンチセンス核酸 (ASO) 医薬品，最近接塩基対法．

Development of a Genome-wide Fast Short Nucleotide Sequence Search Method Considering Binding Energy

Yamazaki Mahiro[1]　Izawa Kazuki[1]　Hirata Ryo[2]　Yanagisawa Keisuke[1]　Ohue Masahito[1]
Akiyama Yutaka[1,a)]

Abstract: The binding energy between nucleic acid sequences is an important factor in considering the binding stability of nucleic acids. Although various methods exist, no method is dedicated for short sequences. The binding energy estimation of short sequences is needed to develop antisense oligonucleotide (ASO) drugs, which has a gap region of approximately 10 bases. In this study, we propose a method for short sequences that enumerate sequences and search through target sequences. We compared to RIsearch, a well-known tool, and the proposed method is 60-90 times faster than the previous tool when the query sequence length is 10-14 nucleotides. In addition, the amount of memory usage was 1.4-116.6MB, indicating that the proposed method can be implemented with a personal computer.

Keywords: Binding Energy, Antisense Oligonucleotide Drugs, Nearest Neighbor Method.

1. はじめに

1.1 アンチセンス核酸 (Antisense Oligonucleotide; ASO)

核酸医薬品は，化学合成された DNA あるいは RNA の骨格を有する一本鎖や二本鎖のオリゴヌクレオチドであり [1]，がんや遺伝性疾患などに対する革新的な医薬品とし

[1] 東京工業大学 情報理工学院 情報工学系
Department of Computer Science, School of Computing, Tokyo Institute of Technology
[2] 株式会社情報数理バイオ
IMSBIO Co., Ltd.
[a)] akiyama@c.titech.ac.jp

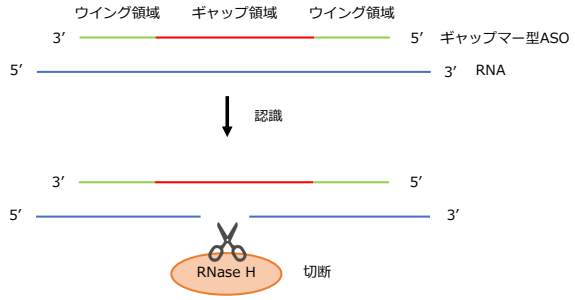

図 1 ギャップマー型 ASO による RNA の切断

ての発展が期待されている [2]．核酸医薬品の一つにアンチセンス核酸 (Antisense Oligonucleotide; ASO) を用いたものが存在しており，これは 13-25 塩基程度と比較的短い配列で設計される [3]．

図 1 のように，中央にギャップ領域 (gap) と呼ばれる 10 塩基ほどの DNA と，両端にウイング領域 (wings) と呼ばれる 5 塩基ほどの化学修飾された RNA を持ったギャップマー (Gapmer) 型 ASO が開発されている．ギャップ領域が目的となる RNA と結合し，細胞内の RNA 分解酵素である RNase H によって認識され，結合した相補鎖 RNA が切断される [2]（図 1）．このようにしてギャップマー型 ASO はターゲットとなる RNA を切断し，発現を阻害する．

ギャップマー型 ASO が標的領域と正しく結合し，RNA を切断するためには，ASO の分子内・分子間相互作用や RNA の二次構造，標的 RNA との結合部位の個数などといった様々な要素が影響する．井澤らはこれらの要素と発現抑制率の関係を解析した結果として，ASO による発現抑制率と RNA–ASO 間の結合エネルギーの相関が最も強い（ピアソンの相関係数 $R = -0.426$）ことを明らかにしており [4]，結合エネルギーの計算は ASO を設計する上で重要である．

ギャップマー型 ASO が標的 RNA と同一あるいは類似した相補結合領域を持つ別の RNA と結合してしまい，想定外の RNA が切断されてしまうことがあり，これにより本来意図しなかった遺伝子の発現を抑制してしまうことがある．これをオフターゲット効果という [5]．ギャップマー型 ASO が標的領域（オンターゲット）と正しく結合し，オフターゲット効果を発生させないためには，ギャップマー型 ASO がオンターゲットに対し特異的に結合することが重要になり，オンターゲット以外の領域で結合エネルギーが強くなる領域が存在しないことが重要になる．

また，オフターゲットによる毒性を考慮した ASO 設計を行う際に，まずは非臨床試験としてマウスやラットといった動物から実験を行い，動物に対する毒性の調査を行なってから最終的に臨床試験としてヒトを対象とすることが一般的である．その際に，マウスやラット，ヒトといった複数の生物種のゲノム配列での解析が必須となる．

1.2 結合エネルギー計算の必要性

結合エネルギーは結合安定性を示す重要なパラメータである．結合安定性には，塩基同士で結合した部分を開く開放エネルギーや RNA の二次構造といった要素が影響を与えている [4]．開放エネルギーと RNA の二次構造予測の計算を精密に実行することは困難であり，精度を落とした推定値を用いる必要がある．一方，結合エネルギーの計算は最近接塩基対法 [6] と呼ばれる手法により比較的容易に行うことができるため，結合エネルギーを計算機により予測することは DNA や RNA 研究の分野において重要になる．

1.3 最近接塩基対による結合エネルギー予測

RNA–RNA 間あるいは DNA–RNA 間の結合エネルギーを計算する手法として最近接塩基対法 [6] が知られている．これはターゲット配列とクエリ配列の対応する部分から 2 塩基ずつ抜きだしその組み合わせに相当する結合エネルギーを与え，その後 1 塩基ずつずらして同様にして与えた結合エネルギーを足し合わせていく動作を繰り返すことにより結合エネルギーを計算するという手法である．RNA–RNA 間において，A（アデニン）–U（ウラシル），G（グアニン）–C（シトシン）が相補的に結合し，また DNA–RNA 間において A–U，T（チミン）–A，G–C，C–G が相補的に結合することは一般的に知られているが，結合エネルギーを計算する際には，GU ウォブルペア (GU wobble pair) [7] というペアを許容するのが一般的である．これは GU 間において塩基対がプロトン化されることによって安定的に結合するため [7]，G-U も相補的塩基対として認めるというものである．したがって，G は C とだけでなく U とも結合し，U も A とだけでなく G と結合する．また，DNA–RNA 間においては GU ペアだけでなく GT ペアも同様に相補的塩基対として扱う．

1.4 本研究に類似した既存の手法

本研究以前にも，クエリ配列を入力しターゲット配列から類似した部分配列を検索するための手法はいくつか存在している．RIsearch[8] は 2012 年に公開された高速な核酸相互作用検索ツールであり，RNA–RNA 間の結合エネルギーだけでなく，DNA–RNA 間の結合エネルギーの計算を行うことができる．また，RIsearch はバルジループを考慮して結合エネルギーを計算することが可能である．結合エネルギー計算の際に，Smith-Waterman-Gotoh アルゴリズム [9], [10] を用いており，クエリ配列とターゲット配列を与えたとき，ターゲット配列の先頭から順にクエリ配列との結合エネルギー計算を行なっていくことによりターゲット配列から結合エネルギーが強い部分領域を検索する．その際，比較回数のオーダーはターゲット配列の長さ m とクエリ配列の長さ n を用いて $\mathcal{O}(mn)$ と表すことができる．

1.5 本研究の目的

ASO 創薬の研究では，ターゲット配列に対し，結合するクエリ配列は 10 塩基程度とかなり短いものとなっている．本研究では，ターゲット配列を特定の生物種の全ゲノム配列，クエリ配列を 10–20 塩基程度の DNA と限定し，ターゲット配列が転写された RNA とクエリ配列間における結合エネルギーが強い部分領域をターゲット配列から抽出する．本研究の目的はターゲット配列とクエリ配列を限定することにより，高速なゲノム配列に対する短鎖核酸配列検索手法を提案することである．

2. 手法

2.1 提案手法の概要

ターゲット配列からクエリ配列が強く結合する部分配列を検索する際に，本研究では既存手法と大きく異なる考え方を用いる．既存手法は，ターゲット配列とクエリ配列を照らし合わせて結合エネルギーの計算を行い，計算結果を保持し，その後，計算結果を用いてターゲット配列から結合エネルギーが強い部分領域を出力する．一方，本研究における手法では，まず与えられたクエリ配列に対して，クエリ配列と同じ長さの塩基配列のうち，結合エネルギーが閾値よりも低い（結合安定性が高い）配列を網羅的に探索，列挙する．その際に，相手となる塩基配列はターゲット配列内に存在するかは考慮せず，クエリ配列と同じ長さの塩基配列で条件を満たす全てのパターンを用意する．条件については後述する．その後，クエリ配列と結合エネルギーが強い塩基配列について，その塩基配列がターゲット配列内に存在するかどうかを検索し，存在すれば結果として出力する．

2.2 提案する核酸配列検索手法

本節では本研究が提案する検索手法について詳述する．以下に示す流れによって結合エネルギー計算と配列検索を行う．（図 3）

(1) 検索を行う塩基配列（DNA または RNA）と結合エネルギーの閾値を入力．
(2) 入力された塩基配列に対し，長さが入力された塩基配列と等しく，連続相補結合条件を満たす塩基配列について，ターゲット配列 (RNA) 内における存在に関係なく全てのパターン列挙し，順にクエリ配列との結合エネルギー計算を実行．
(3) 2. の結果から，与えられた閾値を満たした塩基配列のみを記録．
(4) 3. の結果，残った配列をエネルギーの強い順にソート．
(5) 4. で与えられた塩基配列に対してターゲット配列上での位置を検索．

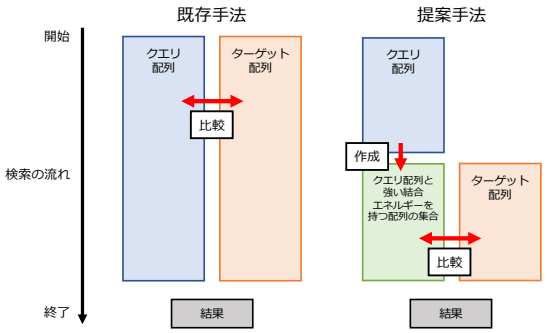

図 2 既存手法と提案手法の違い．既存手法が検索の際にクエリ配列とターゲット配列を比較するため，開始から終了までクエリ配列とターゲット配列を使用し続けるのに対し，提案手法ではまずクエリ配列のみを用いて結合エネルギーが強くなる配列の集合を作成し，その集合に含まれる配列のターゲット配列上における位置を走査することにより検索を行う．

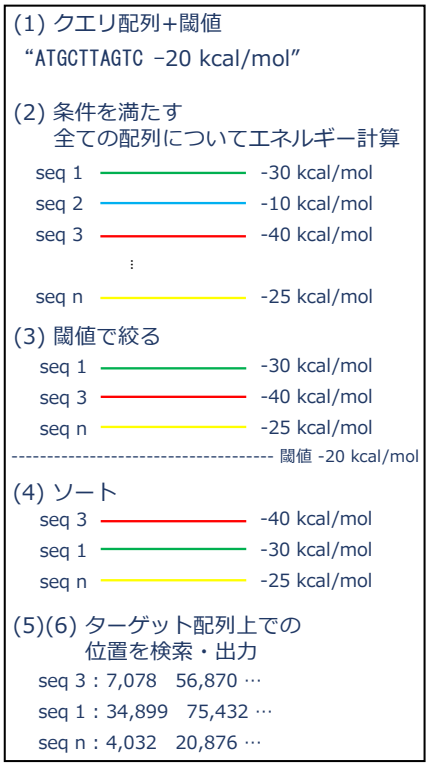

図 3 結合エネルギー計算の流れ

(6) 結果（塩基配列，結合エネルギーの値，ターゲット配列での位置）を出力．

2.2.1 条件を満たす配列の列挙と結合エネルギー計算 (図 3–(2)

本手法ではクエリ配列と l 塩基以上連続で相補的に結合する配列を列挙することで長さ n の塩基配列の全パターンである 4^n 個列挙するときと比較して計算コストの削減を

図っている．また，GU ウォブルペアを考慮するため，クエリ配列の長さ n と連続相補一致数 l が一定であってもクエリ配列を構成する塩基によって配列のパターン数は異なり，n と l を用いて最小と最大のパターン数を示すと式 1 のようになる．また，n と l はユーザーによって変更することが可能とした．

$$\begin{aligned} &\text{最小}: 4^{n-l-1}(3n-3l+4) \\ &\text{最大}: 2^{2n-l-1}(n-l+2) \end{aligned} \tag{1}$$

述べた方法により条件を満たした配列を列挙した後，最近接塩基対法により結合エネルギーを計算する．本研究の手法では DNA–RNA 間と RNA–RNA 間のスコアリングマトリクスを用いることで，クエリ配列が DNA と RNA のどちらであっても結合エネルギーの計算が可能である．また，RIsearch と同一のスコアリングマトリクスを用いることができるようにした [11], [12], [13].

2.2.2 閾値による選別とソート (図 3–(3)(4)

列挙された配列とクエリ配列間における結合エネルギーを計算した後，閾値による選別とソートを行う．閾値はユーザーによる設定が可能である．

2.2.3 ターゲット配列上における位置の検索 (図 3–(5)

これまでの過程で残った塩基配列に対し，ターゲット配列上における位置の検索を行う．その際に，k-mer（k 塩基）逆引きテーブルを用いる．これはターゲット配列を前処理することによって作成され，具体的には，長さ k の塩基配列（4^k 種類）を全て用意し，それぞれの配列がターゲット配列上のどこにあるかをそれぞれの塩基配列の名前がついた csv ファイルに記録することで作られる．例えば，ヒトゲノム配列の 6-mer 逆引きテーブルは，$4^6 = 4{,}096$ 個の csv ファイル (AAAAAA–TTTTTT) が作られ，それぞれのファイル内にファイル名となっている塩基配列がゲノム配列上のどこに存在しているかを記録することによって作成される（図 4）．k-mer 逆引きテーブルにはヒト・マウス・ラット・カニクイザル・マーモセットの 5 種類の動物のゲノム配列が用意されている．このうち，ヒト以外の 4 生物種は，創薬において薬効や副作用の解析に頻繁に用いられるモデル動物である．逆引きテーブルを用いることによりクエリ配列のゲノム配列上における位置を検索することが可能であり，また，クエリ配列が $k+1$ 塩基以上の長さであっても k-mer のテーブルを複数回用いることによってゲノム上での位置を特定することができる．

3. 実験

本研究で提案した配列検索手法を実装したプログラムの計算速度とメモリ使用量を評価する実験を行なった．この章では以下の 3 つの実験を行い，提案手法の実行速度やメモリ使用量の評価を行う．

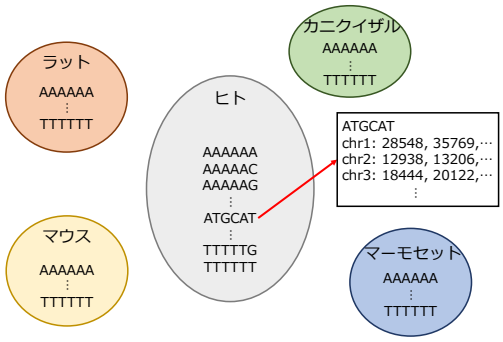

図 4 6-mer テーブルの構成

実験 1.**検索速度比較**
同一のクエリ配列，ターゲット配列に対して提案手法および既存手法を用いて実験を行い，それぞれの実行時間を計測する．

実験 2.**メモリ使用量の測定**
この実験は提案手法に対してのみ行う．同一のターゲット配列に対して，クエリ配列と検索条件を変化させた際にメモリ使用量がどのように変化するかを測定する．

実験 3.**GU ウォブルペアが与える影響の検証**
この実験も提案手法に対してのみ行う．同一のターゲット配列に対して，GU ウォブルペアが存在するクエリ配列と存在しないクエリ配列を用い実行時間がどのように変化するかを測定する．

3.1 実験手順

3.1.1 比較対象プログラム

比較対象プログラムとして RIsearch[8] を利用した．これは前述のように Smith-Waterman-Gotoh アルゴリズムに基づいた手法である．2021 年 6 月 16 日に発表された最新版を用いており，バージョンは 1.2 である．

3.1.2 使用したデータ

ターゲット配列には，NCBI（アメリカ国立生物工学情報センター）が公開しているヒトゲノム配列である GRCh38.p13 を利用する．これはヒトの全ゲノム配列を含んでおり，長さは約 30 億塩基である．また，このヒトゲノム配列を RIsearch でターゲット配列として使用する際には，DNA 配列を転写した RNA 配列を用いている．DNA 配列を RNA 配列に変換する際に FASTA ファイル操作ツールである SeqKit[14] を使用しており，オプションは `-pr --dna2rna` を用いた．

クエリ配列には，実験のために用意した仮想的な塩基配列を複数用いる．実験 1 および 2 に用いた具体的な塩基配列を表 1，実験 3 に用いた具体的な塩基配列を表 2 に示す．実験 1 では表 1 に示した全てのものを，実験 2 では表 1 の

表 1 使用したクエリ配列（実験 1 および 2）．実験 1 では表中全ての配列，実験 2 では表中で塩基配列長が 6–16 塩基のものを用いる．

塩基配列	塩基配列長
ATGCAT	6 塩基
ATGCATGC	8 塩基
ATGCATGCAT	10 塩基
ATGCATGCATGC	12 塩基
ATGCATGCATGCAT	14 塩基
ATGCATGCATGCATGC	16 塩基
ATGCATGCATGCATGCAT	18 塩基

表 2 使用したクエリ配列（実験 3）．それぞれの塩基配列長についてパターン数（式 1）が最小と最大となるような塩基配列を用いる．

塩基配列	塩基配列長	パターン数
ACACACACACAC	12 塩基	22,528
GTGTGTGTGTGT	12 塩基	1,048,576
ACACACACACACAC	14 塩基	458,752
GTGTGTGTGTGTGT	14 塩基	20,971,520

表 3 計算機環境

CPU	2 コア 4 スレッド Intel Core i5-5350U 1.8 GHz（ターボ時 2.9 GHz）
OS	macOS Big Sur 11.2.3
メモリ	8 GB
コンパイラ	gcc ver 9.3.0
オプション	`-o -O2`

表 4 RIsearch の実行オプション

プログラム	オプション
RIsearch	`-Q -t -m su95 -p2`

表 5 提案手法の実行オプション

	連続相補一致塩基長 l	表示件数上限	閾値 [kcal/mol]
実験 1	6	10	−10
実験 2	6	10	−15, −10, −5, 0
実験 3	6	10	−10

うち塩基配列長が 6–16 塩基のものをクエリ配列として用いる．

3.1.3 実験設定

実験はノートパソコン上で行った．具体的な計算機環境は表 3 に示す．計算サーバを保持しない研究者が利用する環境でも使用できる速度になることに注意し，実験を行なった．また，RIsearch 実行時に用いたオプションと提案手法の細かい設定についてはそれぞれ表 4 と表 5 に示す．

3.1.4 計測方法

実行時間とメモリ使用量の計測は time コマンドを用いて行なった．オプションには`-l` を使用した．各実験についてそれぞれ 5 回ずつ同様のコマンドを行い，得られた値の中央値を実験結果とした．以降，提案手法全体のうち 2.2 節において示した検索手法の中で，(4) までの部分を「配列エネルギー計算」，それ以降の 6-mer 逆引きテーブルを用いたターゲット配列上の検索部分を「ターゲット配列検索」と呼ぶ．また，実験結果として示す表中において配列エネルギー計算を「前半部分」，ターゲット配列検索を「後半部分」と表記する．

表 6 クエリ配列長による実行時間の比較．表中 “< 0.1” は計算を行なっているが小数点以下第 1 位までで 0.0 となっていることを示す．

クエリ配列長	提案手法 [s] 前半部分（候補列挙）	後半部分（配列検索）	合計	RIsearch [s]
6 塩基	< 0.1	< 0.1	< 0.1	311.2
8 塩基	< 0.1	< 0.1	< 0.1	393.2
10 塩基	< 0.1	5.0	5.0	476.7
12 塩基	< 0.1	7.2	7.2	557.4
14 塩基	0.7	10.1	10.8	641.2
16 塩基	31.6	9.9	41.5	717.0
18 塩基	594.3	9.4	603.7	803.5

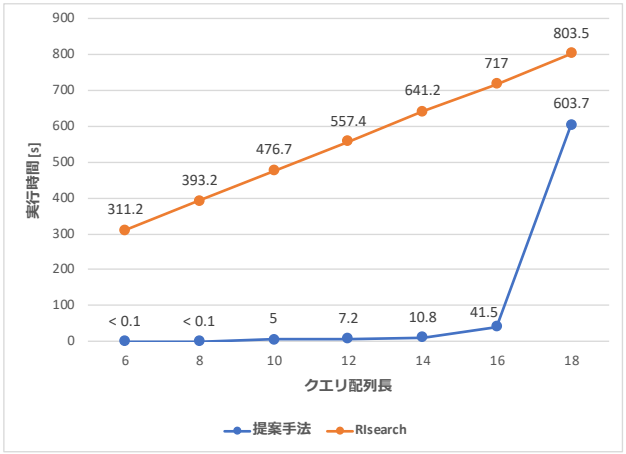

図 5 クエリ配列長による実行時間の比較．縦軸は実行時間 [s]，横軸はクエリ配列長を示す．

4. 結果

4.1 実験 1．検索速度比較

実験 1 において提案手法と既存手法では結果として出力する塩基配列は同じであったが，推定される結合エネルギーの値が異なった．原因は RIsearch が用いているスコアの一部を提案手法では用いていないためである．

クエリ配列の長さを変化させることにより実行時間がどのように変化するかについて検証したところ，提案手法と既存手法では検索時間の変化の様子が大きく異なった．結果を表 6 に示す．また，表 6 の提案手法の合計実行時間と RIsearch の実行時間を折れ線グラフにしたものを図 5 に示す．なお，クエリ配列長が 6–8 塩基の場合は提案手法において閾値を満たす（結合エネルギーが −10 kcal/mol 以下）配列が存在しなかった．

表 7 クエリ配列長によるメモリ使用量の変化．表中括弧で囲まれている値は条件を満たした配列が存在しなかったことを示す．

	閾値 [kcal/mol]			
クエリ配列長	−15	−10	−5	0
6 塩基	(1.4 MB)	(1.4 MB)	33.5 MB	54.4 MB
8 塩基	(1.4 MB)	(1.4 MB)	36.6 MB	36.7 MB
10 塩基	(1.4 MB)	40.3 MB	41.9 MB	41.8 MB
12 塩基	34.3 MB	37.6 MB	36.6 MB	37.1 MB
14 塩基	39.6 MB	38.4 MB	38.4 MB	116.6 MB
16 塩基	68.8 MB	79.0 MB	240.9 MB	1,972.3 MB

表 8 GU ウォブルペアによる実行時間の変化．表中 “< 0.1” は計算を行なっているが小数点以下第 1 位までででは 0.0 となっていることを示す．

塩基配列長	パターン数	実行時間 [s]		
		前半部分（候補列挙）	後半部分（配列検索）	合計
12 塩基	22,528	< 0.1	14.3	14.3
12 塩基	1,048,576	0.2	11.4	11.6
14 塩基	458,752	0.1	22.8	22.9
14 塩基	20,971,520	3.7	18.3	22.0

提案手法が全体を通して既存手法より高速に検索を行うことができるという結果になった．特に，クエリ配列長が 10–14 塩基程度の場合は，検索速度が既存手法に比べて 60–90 倍ほど高速となった．提案手法の結果を詳しく見ると検索時間が急速に増えているのは前半部分にあたる配列エネルギー計算であり，後半部分のターゲット配列検索にかかる実行時間は 10 s 程度で安定するという結果になった．

4.2 実験 2．メモリ使用量の測定

実験 2 においてクエリ配列と検索条件を変化させた際に，提案手法のメモリ使用量がどのように変化するか検証した．結果を表 7 に示す．

全体を通して表の左から右（閾値による制限が緩くなる），上から下（配列長が長くなる）に向けてメモリ使用量が増加している．クエリ配列が長くなる，あるいは，閾値が正に近づくほど閾値を満たした配列の数が増加するのでメモリ使用量が増加する．クエリ配列長が 14 塩基以下の場合はほとんど 40 MB 以下で安定しているが，クエリ配列長が 16 塩基になると 40 MB を越えて，閾値が正に近づくほど急速に増加する．しかし今回の実験においていずれの場合も，PC 上で実施可能な程度のメモリ容量に収まっている．

4.3 実験 3．GU ウォブルペアが与える影響の検証

実験 3 において GU ウォブルペアが存在するクエリ配列と存在しないクエリ配列を用い実行時間がどのように変化するかを測定することで，GU ウォブルペアによる配列数増加が与える影響を検証した．結果を表 8 に示す．

塩基配列長が 12 塩基，14 塩基の両方の場合において，列挙し結合エネルギーを計算する配列のパターン数が少ない塩基配列である A と C で構成されているものの方が全体の実行時間は長くなった．一方，前半部分の配列エネルギー計算の実行時間はパターン数が多い塩基配列である G と T で構成されているものの方が長いという結果になった．

5. 考察

5.1 実行時間

実験 1 の結果より，提案手法が既存手法よりも計算速度の点において優れているという結果が得られた．特に，クエリ配列長が 10–14 塩基程度の場合において計算速度が 60–90 倍と圧倒的に速くなった．ASO はギャップ領域が 10 塩基程度であり，ASO 設計において結合エネルギーが強くなる部分を検索したい場合には提案手法を用いることで，計算時間の大幅な短縮が期待できる．また，既存手法である RIsearch はクエリ配列が長くなってもかかる時間は線形にしか増加しないので，安定した時間で検索を行うことができるということが明らかになった．クエリ配列が大きい場合は RIsearch の方が検索手法として適していると言えるが，ASO 設計に応用を絞った場合に対象の配列長が急に長くなることは考えにくいため，提案手法がより適している．

5.2 実行時間増加率

既存手法である RIsearch はクエリ配列とターゲット配列を先頭から順に比較していくことによって結合エネルギーが強くなる部分を検索する．したがって比較回数のオーダーはターゲット配列長 m とクエリ配列長 n を用いて $\mathcal{O}(mn)$ と表すことができる．今回行なった実験では m が固定であるため，n が線形に増加すると実行時間も線形に増加することが予想され，実際の実験結果からもそれが読み取れる．一方，提案手法は表 6 から，前半部分にあたる配列エネルギー計算の実行時間が全体の実行時間に大きな影響を与えていることがわかる．塩基配列を構成する塩基は 4 種類あり，また，GU ウォブルペアを考慮する必要もあるので，クエリ配列長が n 塩基から $n+1$ 塩基になったとき列挙する配列数は 4 倍以上になる．配列エネルギー計算にかかる実行時間は n が線形に増加すると 4^n 以上の速さで増加することが予想され，今回の実験の場合は 2 塩基ずつクエリ配列を長くしているので 16 倍以上の倍率で増加することになる．表 9 に実験 1 のクエリ配列が 12–18 塩基の場合において提案手法の配列エネルギー計算の実行

表 9 配列エネルギー計算の実行時間の増加倍率．表中 “0.03” は実験 1 において “< 0.1 ” となっていたが倍率を計算することができないため，値を丸める前の “0.03” を実験値として用いた．

クエリ配列長	12	14	16	18
前半部分実行時間 [s]	0.03	0.7	31.6	594.3
前の配列長からの増加倍率	–	×21.7	×45.1	×18.8

時間が何倍増えているかについて示す．表より，バラつきがあるものの実際に 16 倍以上の速さで増加していることがわかる．

5.3 メモリ使用量

実験 2 の結果より，検索条件を満たした配列が存在し，クエリ配列長 12 塩基以下，あるいは，クエリ配列長 14 塩基以下で閾値が −5 kcal/mol 以下のときメモリ使用量はおおよそ 40 MB となった．これはターゲット配列検索にあたる 6-mer 逆引きテーブルを使用したターゲット配列上における位置の検索にかかるメモリが 40 MB 程度であることを示している．一方，クエリ配列長が 16 塩基の場合における結果から，クエリ配列が長くなる，あるいは，閾値を大きくしすぎると，配列エネルギー計算のメモリ使用量が，40 MB を越えてしまい，メモリ使用量が急速に増加してしまうことを示している．

5.4 GU ウォブルペア

実験 3 の結果より，パターン数が多い方が，列挙しクエリ配列間との結合エネルギーを計算する配列の数が多いので，配列エネルギー計算にかかる時間が長くなることがわかった．一方，全体の実行時間はなぜかパターン数が少ない場合の方がパターン数が多い場合より長くなった．6-mer 逆引きテーブルを使用したターゲット配列上における検索にかかる時間に差が生まれたことによりこのような結果となっている．テーブル内それぞれのファイルサイズなどに原因があると考え調査したものの原因は不明である．

6. 結論

6.1 本研究の結論

本研究では，ゲノム配列のような長いターゲット配列と 10 塩基程度の短いクエリ配列が与えられたとき，結合エネルギーが強くなる部分を高速に検索する手法を提案し，その実装と評価実験を行なった．評価実験では，いくつかのクエリ配列を用意し，既存手法である RIsearch との速度比較を行い，また，提案手法のメモリ使用量についてもパラメータを変化させて評価を行なった．結果として，提案手法は既存手法に対し，クエリ配列長が 10–14 程度の場合において 60–90 倍ほどの大幅な実行時間の短縮に成功した（表 10）．また，同条件においてメモリ使用量についても，クエリ配列長が 18 塩基までであれば，サーバ等の大規模な計算環境がなくても十分に実行可能であることを示した．

表 10 実行時間比較

クエリ配列長	提案手法 [s]	RIsearch [s]
10 塩基	5.0	476.7
12 塩基	7.2	557.4
14 塩基	10.8	641.2
16 塩基	41.5	717.0

6.2 今後の課題

今後の課題や改善点としては，今回実装しなかった機能を実装することが挙げられる．例えば，バルジループを考慮できるようにする点がある．ただ，計算速度が著しく落ちてしまう可能性もあるので，ユーザー側でバルジループの許容の有無を決められるようにするとより好ましい．また，提案手法において結合エネルギー計算の際に RIsearch が用いているスコアリングマトリクスを用いたが，RIsearch ではさらに別のスコアをクエリ配列とクエリ配列が結合する部分配列に応じて与えていた．このスコアを導入している理由について深く調査し実装することも今後の課題である．

謝辞 貴重な時間を割いて相談に乗ってくださり，生物学的観点から助言をいただいた東京工業大学生命理工学院正木慶昭助教に感謝の意を表します．

参考文献

[1] T. C. Roberts, *et al.* Advances in oligonucleotide drug delivery. Nature Reviews Drug Discovery, Vol. 19, No. 10, 673–694, 2020.
[2] 横田隆徳, 仁科一隆, 桑原宏哉, 核酸医薬を用いた遺伝子治療の展望, 神経治療, Vol. 33, No.3, 303–306, 2016.
[3] M. Hendling, and I. Barišić. In-silico Design of DNA Oligonucleotides: Challenges and Approaches. Computational and Structural Biotechnology Journal, Vol. 17, 1056–1065, 2019.
[4] 井澤和也，柳澤渓甫，大上雅史，秋山泰，標的配列との結合・開放エネルギー推定に基づくアンチセンス核酸の阻害活性モデルの研究，研究報告バイオ情報学，Vol. 2021–BIO–65，No. 7，1–7，2021.
[5] 木下潔，中澤隆弘，荒戸照世，三井田宏明，平林容子，真木一茂，吉田徳幸，井上貴雄，既承認核酸医薬品の審査報告書を読み解く，医薬品医療機器レギュラトリーサイエンス，Vol. 51，No. 2，70–82，2020.
[6] I. Tinoco Jr, *et al.* Improved estimation of secondary structure in ribonucleic acids. Nature New Biology, Vol. 246, 40–41, 1973.
[7] T. Brown, G. A. Leonard, *et al.* Crystal structure and stability of a DNA duplex containing A(anti)・G(syn) base-pairs. Journal of Molecular Biology, Vol. 207, Issue 2, 455–457, 1989.
[8] A. Wenzel, `et al.` RIsearch: fast RNA–RNA interaction search using a simplified nearest-neighbor energy model. Bioinformatics, Vol. 28, No. 21, 2738–2746, 2012.
[9] T. F. Smith, and M. S. Waterman. Identification of Common Molecular Subsequences. Journal of Molecular Biology, Vol. 147, Issue 1, 195–197, 1981.

[10] O. Gotoh. An improved algorithm for matching biological sequences. Journal of Molecular Biology, Vol. 162, 705–708, 1982.
[11] N. Sugimoto, et al. Thermodynamic Parameters To Predict Stability of RNA/DNA Hybrid Duplexes. Biochemistry, Vol. 34, Issue 35, 11211–11216, 1995.
[12] N. Sugimoto, *et al.* Thermodynamics-Structure Relationship of Single Mismatches in RNA/DNA Duplexes. Biochemistry, Vol. 39, Issue 37, 11270–11281, 2000.
[13] D. H. Turner, and D. H. Mathews. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Research, Vol. 38, Issue suppl_1, D280–D282, 2010.
[14] SeqKit - a cross-platform and ultrafast toolkit for FASTA/Q file manipulation. https://bioinf.shenwei.me/seqkit/