

学士論文

DNAストレージにおけるナノ
題目 ポアシーケンサーの読み出し特
性を考慮した誤り訂正手法

指導教員 佐藤 高史 教授

京都大学工学部 電気電子工学科

氏名 岩瀬 真太郎

令和8年2月10日

Error Correction Method Considering the Reading Characteristics of Nanopore Sequencers in DNA Storage

Shintaro Iwase

Abstract

This document briefly explains the usage of the L^AT_EX style file for KUEE bacheler thesis and master thesis.

目 次

第 1 章 序論	1
第 2 章 DNA ストレージの符号化・読み出しプロセス	3
2.1 ナノポアシーケンサーと Basecaller	3
2.2 HEDGES 符号化	4
2.3 HEDGES 復号	7
2.4 既存手法の課題と研究目的	8
第 3 章 提案手法	10
3.1 復号の探索効率化	10
3.1.1 ナノポアシーケンサーの読み出し特性のモデル化	10
3.1.2 CTC 出力の扱い	11
3.1.3 報酬とペナルティの最適化	12
3.2 ロングリードに向けた符号化アルゴリズムの改良	13
第 4 章 シミュレーション	14
4.1 復号の探索効率化	14
4.1.1 実験方法	14
4.1.2 実験条件	17
4.1.3 結果と考察	18
4.2 ロングリードに向けた符号化アルゴリズムの改良	19
4.2.1 実験手法	19
4.2.2 実験条件	20
4.2.3 結果と考察	20
第 5 章 結論	21
謝 辞	22

第1章 序論

情報化社会が進展した現代では、日々大量のデジタルデータが生成されており、世界中のデータ量は爆発的に増加し続けている。2025年の報告では全世界のデジタルデータ量は173.4ゼタバイトにのぼると推定され、2029年には527.5ゼタバイトに達すると予測されている[1]。これらの大量のデータのうち、約60%はアクセス頻度の極めて低いコールドデータであることが知られている[2]。コールドデータの中には、法令に基づき長期間保存が義務付けられているものや、将来的に解析される可能性のある研究データなどが含まれており、これらを消去することは困難である。そのため、コールドデータを安価で長期間保存できるアーカイブストレージ技術が求められている[3, 2]。

近年、次世代のアーカイブストレージ技術としてDNAストレージが注目されている。DNAストレージとは、デジタルデータをDNAの4種類の塩基配列(A, C, G, T)に符号化し、人工的に合成したDNA分子にデータを保存する技術である。DNAストレージは、適切な保存環境下であれば数千年以上にわたりデータを保存することが可能な高い耐久性を持つことに加えて、体積あたりに保存可能な情報量が非常に大きいため高い情報密度を実現できることが見込まれており、コールドデータの保存に適した新たなストレージ技術として期待されている[4, 5, 2]。

DNAストレージにおけるデータの読み出しでは、DNAシーケンサーを用いてDNA分子の塩基配列を解析する[5]。DNAシーケンサーは数十から数百bp^{*}の比較的短いDNAを対象としたショートリードシーケンサー[6]と、数百から数万bpの比較的長いDNAを対象としたロングリードシーケンサー[7]に大別されるが、DNAストレージに関する現在までの多くの研究では、SBS(Sequence By Synthesis)法[8]と呼ばれる技術を用いたショートリードシーケンサーが主に用いられてきた。SBS法は同時に大量のDNA断片を高精度で解析することができるという利点があるが、一度に解析できるDNAの塩基長が数百bp程度と短いことや、解析速度が遅いという欠点がある[8]。一方、ロングリードシーケンサーの一つであるナノポアシーケンサーは一度に数十万bp程度の長いDNAを高速に解析するこ

* base pair(塩基対)の略。2本鎖DNAの塩基配列長を表す単位。1本鎖の場合はnucleotidesからntという単位を用いる。

とが出来るため、DNAストレージの読み出し用のシーケンサーとして実用的であると考えられる[7]。そこで、本研究ではナノポアシーケンサーを用いたDNAストレージに着目する。

ナノポアシーケンサーは高速で長い塩基長のDNAを解析できる一方で、SBS法に比べて塩基配列の決定精度が低いことが知られており、特に挿入・消失エラーが多いことが課題となっている[7, 9]。そのため、DNAストレージにおいてナノポアシーケンサーを用いる場合、置換・挿入・消失エラーに対応した誤り訂正符号の検討が重要である。

先行研究では、同一塩基の一定以上の連続(ホモポリマー)の禁止やグアニン・シトシン(GC)の含有率に関する制限といったシーケンス制約を満たしつつ、置換・挿入・消失エラーに対応可能なDNAストレージ向けの誤り訂正符号を提案し、シミュレーションと実際に合成DNAを用いた実験により、最大10%程度の誤りを含む塩基配列からペタバイト規模のデータを正確に復元できる可能性を示した[10]。しかし、先行研究はDNAストレージの読み出しにはショートリードシーケンサーを用いることを前提としており、ナノポアシーケンサーの読み出し特性を考慮した誤り訂正符号の検討は十分に行われていない。また、先行研究の提案手法では長い塩基長では復号精度が低下することが知られており、ナノポアシーケンサーのようなロングリードシーケンサーを用いて長い塩基長を扱う場合の復号精度の向上も課題である。

本研究では、DNAストレージの読み出しにおいてナノポアシーケンサーを用いることを前提とした誤り訂正手法を提案する。既存の誤り訂正手法における復号アルゴリズムにおいて、ナノポアシーケンサーの読み出し特性を考慮した最適化を行い、復号処理の効率化と精度の向上を図る。また、符号化プロセスにおいても長い塩基長のDNAを扱う際の精度の低下を抑制するための手法を提案する。提案手法に対してバイナリデータの符号化からナノポアシーケンサーを用いた読み出し、復号までの一連のプロセスに関するシミュレーションを行い、その性能を評価する。

第2章 DNAストレージの符号化・読み出しプロセス

2.1 ナノポアシーケンサーとBasecaller

ナノポアシーケンサーはDNAが溶液中で帯電していることを利用して、電気泳動現象によりDNAをナノポアと呼ばれるタンパク質の細孔に通過させることでDNAの塩基配列を解析するシーケンサーである[7, 9]。ナノポアシーケンサーのフローセル内の膜の上には複数のナノポアタンパク質が配置されており、その細孔の直径は約1nmで、1本のDNAのみが通過できる大きさになっている。フローセルは塩化カリウムを含んだ電解質で満たされており、溶液中でDNAは負に帯電しているため、電圧をかけることで電気泳動により電場の方向に移動する。これによりDNAの進行方向を制御し、ナノポアを通過させる。ナノポアを通過するDNAはヘリカーゼと呼ばれる酵素によって2本鎖DNAから1本鎖DNAに解かれつつ、一定の速度で通過するように制御されている。このとき、ナノポアを通過するDNAがナノポア内のイオンの流れを阻害するため、イオン電流の大きさが変化する。イオン電流の変化は、ナノポアを通過中の塩基の並びによって異なるため、電流の変化を解析することでDNAの塩基配列を決定することができる。

電流信号から塩基配列を決定するプロセスはBasecallと呼ばれ、Basecallerと呼ばれる機械学習モデルによって行われる。Basecallerはナノポアシーケンサーを開発しているOxford Nanopore Technologies(ONT)社からいくつかのバージョンが公開されているが、本研究で使用するBonitoと呼ばれるBasecallerのアーキテクチャの概要を図2.2に示す[11]。Bonitoでは、まず入力された電流信号に対して畳み込みニューラルネットワーク(CNN)を適用し、特徴量を抽出する。次に、LSTMを用いて電流信号の長距離の依存関係を捉え、時系列データとしての特徴を抽出する。その後、CTC出力層でLSTM出力に線形層を適用し、blankと4種類の塩基からなるラベル集合(N, A, C, G, T)に対するスコア系列を出力する。このスコア系列に対してビームサーチを用いて最終的な塩基配列を決定する。このと

き，ラベル列に対して Collapse と呼ばれる以下の処理を適用する。

1. 連続する同一ラベルを1つにまとめる。
2. blank ラベル (N) を削除する。

これにより，例えばラベル列 “AANNCCGNGGGTT” は “ACGGT” に変換される。

DNAストレージの読み出しにナノポアシーケンサーを採用する利点としては以下のような点が挙げられる [7, 9, 8]。

- SBS法では一度に解析できるDNAの塩基長が数十bpから数百bpと短く，長いDNAは複数の断片に分割して解析する必要があるのに対し，ナノポアシーケンサーは一度に数百bpから数百万bpの長いDNAを直接的に解析することが出来る。
- SBS法では解析に数日から数週間を要するのに対し，ナノポアシーケンサーは400bp/sでの高速な解析が可能である。
- ナノポアシーケンサーはリアルタイムでの解析が可能であり，解析するサンプルを柔軟に制御できるため，必要なデータを効率的に取り出すことが出来る。
- 小型で低コストな装置であるため，設置場所の制約が少なく，運用コストが低い。

一方で，ナノポアシーケンサーはヘリカーゼを用いたDNAの進行速度の制御が難しく，Basecallerでの塩基配列の推論が正確に行われない場合がある。そのため，ナノポアシーケンサーはSBS法に比べて読み取り精度が低いことが知られており，ナノポアシーケンサーを用いたDNAストレージでは，誤り訂正符号の検討が特に重要である [7, 9]。

2.2 HEDGES 符号化

DNAストレージにおける符号化では，DNA合成時のエラーの抑制や構造安定性の確保のためにシーケンス制約が課される。シーケンス制約には一定以上の同一塩基の連続(ホモポリマー)の禁止やグアニン・シトシン(GC)含有率が一定範

囲内に収まるようにする制約がある [12]. また, 復号の際に発生するエラーの訂正を考慮した冗長性を付加することに加えて, シーケンスの際に発生するエラーに対しての耐性を持たせる必要がある. シーケンス時に発生するエラーには以下の 3 種類がある.

- 置換エラー (substitution): 読み出された塩基が本来の塩基とは異なる場合.
- 挿入エラー (insertion): 本来の塩基配列に存在しない塩基が読み出された場合.
- 消失エラー (deletion): 本来の塩基配列に存在する塩基が読み出されなかつた場合.

HEDGES(Hash Encoded, Decoded by Greedy Exhaustive Search) は, シーケンス制約を満たしつつ, 置換・挿入・消失エラーに対応可能な誤り訂正手法である [10]. HEDGES ではデータをパケット単位で扱う. 各パケットは 255 本の固定長 DNA ストランドからなり, ストランドの塩基長は自由に設定できるが, 通常は数百 nt 程度が選ばれる. 符号化の際はまずバイナリデータをパケット単位に分割し, 各パケットのビット列に対してリード・ソロモン (RS) 符号化を適用する. この RS 符号は外部符号と呼び, 塩基配列からビット列への復号の際に発生したエラーを最終的に訂正する役割を持つ. 次に, RS 符号化を適用したビット列に対して内部符号である HEDGES 符号化を適用し, 塩基配列を生成する. HEDGES はビット列から塩基配列への変換において, 符号化率が可変であるという特徴を持つ. DNA の塩基は 4 種類あるため, 1 塩基あたり最大 2 ビットまで割り当てることが出来るが, この場合, 符号化率は $r = 1$ であり冗長性がないため, 誤り訂正能力を持たない. 符号化率を下げることで冗長性を付加し, 誤り訂正能力を持たせることが出来る.

ビット列から塩基配列の変換においては, ここでは符号化率 $r = 0.5$, すなわち 1 ビットを 1 塩基に対応させる場合の符号化手法について説明し, その後に他の符号化率への拡張方法について述べる.

メッセージのビット列 $\{b_i\}$ が以下のように与えられたとする.

$$b_i, \quad i = 0, 1, 2, \dots, M, \quad b_i \in \{0, 1\} \quad (2.1)$$

符号化率 $r = 0.5$ では, 各ビット b_i に対して塩基 C_i を割り当てればよい.

$$C_i, \quad i = 0, 1, 2, \dots, M, \quad C_i \in \{C_i^*\} \quad (2.2)$$

ここで、 $\{C_i^*\}$ はシーケンス制約を満たすために C_i として許される塩基の候補集合である。 $\{C_i^*\}$ の各塩基には 0 から順に整数値が割り当てられているとする。このとき、 i 番目の塩基 C_i は以下の式で与えられる。

$$K_i = F(S_i, I_i, B_i) \quad (2.3)$$

$$C_i = K_i + b_i \pmod{\#C_i^*} \quad (2.4)$$

ここで、 F はハッシュ関数であり、 S_i はストランド ID に基づくソルト、 I_i はビット位置のインデックス、 B_i は直前の 12 ビット分の値である。また、 $\#C_i^*$ は $\{C_i^*\}$ の要素数を表す。すなわち、各ビットに対応する塩基は、ストランド ID、ビット位置、直前のビット列に基づくハッシュ関数の値を目的のビットに加算し、出力塩基の候補数で剰余をとることにより、擬似ランダムかつ一意に決定される。

$r = 0.5$ 以外の符号化率を用いる場合も基本的な考え方は同様である。 $r = 0.5$ の場合は 1 塩基に 1 ビットを対応させたが、他の符号化率を実現するには、各塩基へ割り当てるビット数を 0 ビットから 2 ビットの範囲で規則的に変化させる。つまり、一般には出力塩基 C_i に対応するものはビット b_i ではなく、0 から 2 の長さを持つビット列 v_i となる。以下に、符号化率 $r = 0.750$ の場合と $r = 0.333$ の場合の v_i の構成例を示す。

$$r = 0.750 : v_0 = b_0b_1, v_1 = b_2, v_2 = b_3b_4, v_3 = b_5, v_4 = b_6b_7, \dots \quad (2.5)$$

$$r = 0.333 : v_0 = b_0, v_1 = b_1, v_2 = 0, v_3 = b_2, v_4 = b_3, v_5 = 0, \dots \quad (2.6)$$

ただし、式 (2.5) において隣接するビットは 2 ビットの値を表している。 $r = 0.750$ では塩基へビット割り当て数は 2 ビットと 1 ビットが交互に繰り返され、 $r = 0.333$ では、塩基へのビット割り当て数は 1 ビット、1 ビット、0 ビットの順に繰り返される。したがって、各ビット列 v_i に対応する出力塩基を決定する式は、式 (2.4) の b_i を v_i に置き換えたものとなり、以下のように表される。

$$C_i = K_i + v_i = F(S_i, I_i, B_i) + v_i \pmod{\#C_i^*} \quad (2.7)$$

HEDGES でエンコードするビット列の先頭はストランド ID を表すビット列であり、ID 部分の符号化では $S_i = 0$ とされる。ビット列を符号化して得られた塩基配列の両端にはプライマーと呼ばれる決められた塩基配列を付加され、最終的な DNA ストランドの出力となる。

2.3 HEDGES 復号

HEDGES では置換・挿入、消失エラーを含む塩基配列から元のビット列を復元するために、Greedy Exhaustive Search(貪欲探索法)に基づく復号アルゴリズムを用いる [13]。復号アルゴリズムでは、まず入力された塩基 C'_i に対応するビット列 v_i の各候補について仮説を立てる。各仮説において式 (2.7) を用いて予測される塩基 C_i と、実際に入力された塩基 C'_i を比較し、一致していればスコアに報酬(負の値)を加算し、不一致であればペナルティ(正の値)を加算する。仮説は各ステップにおいてスコアが最小となるものを選択し、これを親仮説として次の仮説を開していくことで、ヒープ構造を構築する。さらに、挿入・消失エラーを考慮するために、各仮説では v_i だけではなく、 $\Delta \in \{-1, 0, 1\}$ で表される skew と呼ばれるパラメータも同時に仮定する。skew は参照塩基を現在の位置からどれだけずらすかを表し、skew の値によってステップごとのペナルティ ΔP は、報酬 P_{ok} 、置換ペナルティ P_{sub} 、挿入ペナルティ P_{ins} 、消去ペナルティ P_{del} を用いて以下のように計算される。

- $\Delta = 0$ の場合:

位置ずれではなく、一致または置換エラーを表す。

$$\Delta P = \begin{cases} P_{ok} & (\text{一致の場合}) \\ P_{sub} & (\text{不一致の場合}) \end{cases} \quad (2.8)$$

- $\Delta = -1$ の場合:

消失エラーを仮定するため、入力塩基の参照位置を一つ戻す。この場合は参考する塩基がないので比較は行われない。

$$\Delta P = P_{del} \quad (2.9)$$

- $\Delta = 1$ の場合:

1 塩基の挿入エラーを仮定するため、入力塩基の参照位置を一つ進める。

$$\Delta P = \begin{cases} P_{ins} + P_{ok} & (\text{一致の場合}) \\ P_{ins} + P_{sub} & (\text{不一致の場合}) \end{cases} \quad (2.10)$$

各仮説のスコアは、親仮説のスコアに ΔP を加算することで蓄積されていく。そのため、仮説に誤りがある場合、後続の仮説のペナルティは急速に蓄積される特

性がある。これにより、正しい復号経路は他の経路に比べて顕著に低いスコアを持つようになり、効率的に正しい復号経路を見つけることができる。ただし、メッセージの末尾の数バイトでは、誤った仮説に対するペナルティが蓄積されるのに十分な後続の塩基がないため、復号精度が低下する。これに対処するため、符号化時にメッセージ末尾に2バイト分の runout と呼ばれるゼロパディングを追加し、復号時にはこれを除去する処理が行われる。

復号におけるヒープ探索処理では、計算の複雑性が爆発的に増加することを防ぐために、ヒープサイズに上限を設け、上限に達した時点でデコードの失敗を宣言して探索を中止し、ストランドの残りの部分をビット消失としてマークする。復号失敗によるビット消失は外部符号である RS 符号によって最終的に訂正される。

2.4 既存手法の課題と研究目的

[10] の研究における課題点としては、以下の点が挙げられる。

- 既存の HEDGES 復号アルゴリズムにおいて、置換・挿入・消失エラーに対するペナルティ P_{sub} , P_{ins} , P_{del} の値は概念的には各エラーの発生確率の負の対数であるとされているが、先行研究のシミュレーション評価においては、各エラーが等しい確率で発生する場合を想定して $P_{sub} = P_{ins} = P_{del} = 1$ と設定した上で、いくつかの符号化率において最適な報酬 P_{ok} の値を経験的に決定するにとどめられており、実際にナノポアシーケンサーを用いて塩基の読み出しを行う場合の詳細な出力特性に基づいた報酬とペナルティの最適化については検討されていない。
- 既存の HEDGES 復号アルゴリズムにおける入力は塩基配列であるため、各時点における塩基の出力確率分布を考慮した復号は行われない。
- 既存の HEDGES 符号化・復号手法では、ヒープサイズの上限に達した場合に復号失敗としてその後のビット列をすべて消失として扱うため、長い塩基配列では復号失敗の発生率が高くなり、復号精度が低下する。

本研究では、DNA ストレージの読み出しにおいてナノポアシーケンサーを用いることを前提とした誤り訂正手法を提案する。HEDGES 復号アルゴリズムにおける報酬とペナルティに関して、ナノポアシーケンサーで塩基の読み出しを行う場合

の各塩基のエラーの発生確率や遷移確率といった統計的な読み出し特性を考慮した最適化を行う。また、Basecaller によって出力される各塩基の確率分布も報酬とペナルティに反映されることにより、復号における探索数の削減と精度の向上を図る。さらに、符号化プロセスにおいても長い塩基長の DNA を扱う際の精度の低下を抑制するための手法を提案する。これらの提案手法に対してシミュレーションを実施し、その性能を評価する。

第3章 提案手法

3.1 復号の探索効率化

本節では、復号における探索の効率化を図るために、ナノポアシーケンサーの統計的な読み出し特性と Basecaller 出力における各時点での塩基の確率分布を考慮した報酬とペナルティの最適化手法を提案する。まず、ナノポアシーケンサーの読み出し特性のモデル化と、Basecaller における CTC 出力の扱いについて説明し、その後にこれらを報酬とペナルティに反映させる手法について述べる。

3.1.1 ナノポアシーケンサーの読み出し特性のモデル化

塩基配列の読み出しにおいて、読み出し後の塩基配列と本来の塩基配列が分かっているとき、シーケンスアライメントと呼ばれる手法を用いてこの 2 つの塩基配列を比較し、一致部分や各種エラーの発生箇所を特定することができる [14]。アライメントされた 2 つの塩基配列において、対応する塩基のペアが一致している場合は正しい読み出しを表し、不一致の場合は置換エラーを表す。また、一方の配列の塩基が他方の配列の空白に対応している場合は挿入・消失エラーを表す。このようにして得られるアライメント情報から、DNA 読み出しにおける正しい読み出しと置換・挿入・消失エラーの発生という各事象は 4 種類の塩基に空白 (N) を含めた $\{N, A, C, G, T\}$ の 5 種類のラベル間の遷移として表すことができる。

これを用いてナノポアシーケンサーの統計的な読み出し特性を以下の行列 \mathbf{T} で表す。

$$\mathbf{T} = \begin{bmatrix} 0 & p_{AN} & p_{CN} & p_{GN} & p_{TN} \\ p_{NA} & p_{AA} & p_{CA} & p_{GA} & p_{TA} \\ p_{NC} & p_{AC} & p_{CC} & p_{GC} & p_{TC} \\ p_{NG} & p_{AG} & p_{CG} & p_{GG} & p_{TG} \\ p_{NT} & p_{AT} & p_{CT} & p_{GT} & p_{TT} \end{bmatrix} \quad (3.1)$$

この行列の各要素は、4 種類の塩基が等しい割合で含まれる十分に長い DNA 配列をナノポアシーケンサーで読み出す場合における、正しい読み出し、置換・挿入・消失エラーの発生確率を表し、 $X, Y \in \{A, C, G, T\}$ に対して

- p_{XX} : 塩基 X が正しく読み出される確率
- p_{XY} : 塩基 X が塩基 Y に置換される確率
- p_{XN} : 塩基 X が消失する確率
- p_{NY} : 塩基 Y が挿入される確率

を表す。 \mathbf{T} の全要素の和は 1 になる。この行列 \mathbf{T} の第 1 行第 1 列以外の各要素に対して自然対数をとって -1 を乗じた行列を \mathbf{P}_{STATS} とおく、後述の手法により報酬とペナルティの算出に利用する。

3.1.2 CTC 出力の扱い

2.1 節で説明したように、Basecaller は通常、CTC 出力に対してビームサーチを用いて最終的に塩基配列を出力するが、本研究の提案手法では Basecaller では塩基配列の決定までは行わず、CTC 出力のスコア系列を直接出力する。また、2.3 節で説明したように、既存の HEDGES 復号アルゴリズムでは塩基配列を入力とするが、提案手法では CTC 出力のスコア系列を入力として利用する(図 3.1)。これにより、各時点における出力塩基の確率分布を復号アルゴリズムに反映させることができる。

CTC 出力が $N \times 5$ 行列 \mathbf{S} で与えられるとする。ここで、 N は入力電流信号の長さに依存する時系列長であり、各行は各時点における 5 種類のラベル $\{N, A, C, G, T\}$ に対する出力確率を表す。 \mathbf{S} に対して、2.1 節で説明した Collapse に相当する以下の処理を適用して \mathbf{S}' を得る。

1. \mathbf{S} において出力確率が最大となるラベルが連續している複数行に対し、各ラベルの確率の平均をとり、1 行にまとめた行列を \mathbf{S}' とする。
2. \mathbf{S}' から、blank ラベル (N) に対する確率が最大となっている行を削除する。

これにより得られた行列 \mathbf{S}' の各行で確率が最大となる塩基を選択することで得られる塩基配列は、通常の Basecaller の最終層でのビームサーチにおいてビーム幅を 1 とした場合の Basecall 結果に相当する。 \mathbf{S}' の各要素に対して自然対数をとって -1 を乗じた行列を \mathbf{P}_{CTC} とおく。 \mathbf{P}_{CTC} の各列の要素は、各時点において各塩基または blank が出力される確率に基づくスコアを表す。2.3 節で説明したよ

うに、既存の HEDGES 復号アルゴリズムでは、仮説から式 (2.7) を用いて予測される塩基と対応する入力塩基を比較して、一致・不一致により報酬やペナルティを決定するが、提案手法では、仮説から予測される塩基に対する出力確率に基づくスコアを \mathbf{P}_{CTC} から取得し、これを用いて報酬やペナルティを決定する。

3.1.3 報酬とペナルティの最適化

既存手法では、報酬 P_{ok} と置換・挿入・消失エラーに対するペナルティ P_{sub} , P_{ins} , P_{del} は一定値を用いるが、本研究では 3.1.1 節および 3.1.2 節で述べたナノポアーケンサーの読み出し特性と Basecaller の CTC 出力を考慮した新しい報酬 P'_{ok} とペナルティ P'_{sub} , P'_{ins} , P'_{del} を提案し、以下のように定義する。

$$P'_{ok} = P_{ok} + \alpha_{ok}(\mathbf{P}_{STATS}[x_{hypo}, x_{max}[i]] - \mu_{STATS_ok}) + \beta_{ok}(\mathbf{P}_{CTC}[i, x_{hypo}] - \mu_{CTC_ok}) \quad (3.2)$$

$$P'_{sub} = P_{sub} + \alpha_{sub}(\mathbf{P}_{STATS}[x_{hypo}, x_{max}[i]] - \mu_{STATS_sub}) + \beta_{sub}(\mathbf{P}_{CTC}[i, x_{hypo}] - \mu_{CTC_sub}) \quad (3.3)$$

$$P'_{ins} = P_{ins} + \alpha_{ins}(\mathbf{P}_{STATS}[N, x_{max}[i-1]] - \mu_{STATS_ins}) + \beta_{ins}(\mathbf{P}_{CTC}[i-1, x_{hypo}] - \mu_{CTC_ins}) \quad (3.4)$$

$$P'_{del} = P_{del} + \alpha_{del}(\mathbf{P}_{STATS}[x_{hypo}, N] - \mu_{STATS_del}) \quad (3.5)$$

ここで、 x_{hypo} は仮説のビット列から出力される塩基を表し、 i は仮説が参照する \mathbf{P}_{CTC} の行番号を表す。また、 $x_{max}[i]$ は \mathbf{P}_{CTC} の i 行目において出力確率が最大となる塩基を表す。 $\mathbf{P}_{STATS}[X, Y]$ ($X, Y \in \{N, A, C, G, T\}$) は \mathbf{P}_{STATS} におけるラベル X からラベル Y への遷移に対応するスコアを表す。 $\mathbf{P}_{CTC}[i, X]$ は、 \mathbf{P}_{CTC} の i 行目におけるラベル X のスコアを表す。 μ_{STATS_ok} , μ_{STATS_sub} , μ_{STATS_ins} , μ_{STATS_del} はそれぞれ、正しい読み出し、置換エラー、挿入エラー、消失エラー発生したときの \mathbf{P}_{STATS} のスコアの期待値を表し、 μ_{CTC_ok} , μ_{CTC_sub} , μ_{CTC_ins} はそれぞれ、 \mathbf{P}_{CTC} において正しい読み出し、置換エラー、挿入エラーに対応するスコアの統計的な期待値を表す。報酬と各ペナルティは、固定の値 P_{ok} , P_{sub} , P_{ins} , P_{del} に対して、期待値が 0 になるように正規化された補正項を加算することで計算される。各補正項には、調整用のパラメータ α_{ok} , α_{sub} , α_{ins} , α_{del} , β_{ok} , β_{sub} , β_{ins} が乗じられており、シミュレーションにより固定のペナルティと各補正項の重み付け

を最適化する。各定数の設定方法やパラメータの最適化手法については 4 で詳しく述べる。

3.2 ロングリードに向けた符号化アルゴリズムの改良

2.4 節で述べたように、既存の HEDGES 符号化・復号手法ではストランドを長くすると復号精度が低下することが知られている。これは、復号アルゴリズムにおいて仮説数がヒープサイズの上限に達した場合に復号失敗としてその後のビット列をすべて消失として扱うため、塩基長が大きくなるほど復号失敗の発生率が高くなることに起因している。本節では、長い塩基長の DNA を扱う際の精度の低下を抑制するための符号化・復号アルゴリズムの改良手法を提案する。

HEDGES の符号化アルゴリズムでは、各ビット列 v_i に対応する出力塩基 C_i は式 (2.7) により、ストランド ID, ビット位置, 直前のビット列に基づくハッシュ関数を用いて連鎖的に決定される。そのため、例えばストランド内のある位置でバースト誤りが発生した場合、後続の塩基配列の復号に対しても連鎖的に影響を及ぼし、正しい復号経路の探索が困難になる。そこで、提案手法では図 3.2 のように長い塩基配列を数百 nt 程度の複数の短いセグメントに分割し、各セグメントの先頭で、ハッシュ関数の入力として用いるビット位置 I_i と直前のビット列情報 B_i を 0 にリセットして符号化する。セグメントの区切り位置を特定するために、隣接するセグメント間には特定のパターンを持つ 10 塩基程度のセパレータを挿入する。復号時にはまず、セパレータの位置として予想される位置から前後 30 塩基程度の範囲を探査し、セパレータのパターンに一致または類似する塩基配列を検出する。セパレータの位置が特定できたら、各セグメントごとに 2.3 節で説明した復号アルゴリズムを適用し、各セグメントの先頭では I_i と B_i を 0 にリセットした上でヒープを初期化して復号を行う。これにより、ヒープサイズが上限を超えることによる復号失敗を抑制し、復号失敗が発生した場合でも、その影響をセグメント内だけに限定することができ、次のセグメントから復号を再開することができる。

第4章 シミュレーション

4.1 復号の探索効率化

4.1.1 実験方法

3.1節で提案した報酬とペナルティの最適化手法の性能をシミュレーションにより評価するために、まずはナノポアシーケンサーの読み出し特性を評価し、行列 \mathbf{P}_{STATS} を得る必要がある。そのため、ランダムなバイナリデータを HEDGES で符号化した複数の DNA ストランドに対し、ナノポアシーケンサーによる電流波形を Squigulator と呼ばれるシミュレータを用いてシミュレーション行った [15]。次に、シミュレーションされた電流波形に対して Bonito Basecaller を用いて CTC 出力を取得し、3.1.2節で述べた方法で \mathbf{P}_{CTC} を得た。ここで、HEDGES 符号化結果における n 番目の DNA ストランドの i 番目の塩基を $C_i^{(n)}$ と表し、この塩基配列を $\{C_i^{(n)}\}$ とする。また、 $\{C_i^{(n)}\}$ の Basecallにおいて得られた \mathbf{P}_{CTC} を $\mathbf{P}_{CTC}^{(n)}$ と表すこととし、 $\mathbf{P}_{CTC}^{(n)}$ の各行において、出力確率が最大となる塩基を選択することで得られる塩基配列を $\{C'_i^{(n)}\}$ と表す。この $\{C'_i^{(n)}\}$ と $\{C_i^{(n)}\}$ に対してグローバルアライメントを行い、各塩基ごとに正しい読み出し、置換・挿入・消失エラーの発生数を集計した [14]。これを全ての n について行い、集計することで各事象の発生確率を計算し、行列 \mathbf{P}_{STATS} を得た。

μ_{STATS_ok} , μ_{STATS_sub} , μ_{STATS_ins} , μ_{STATS_del} はそれぞれ、正しい読み出し、置換エラー、挿入エラー、消失エラー発生したときの \mathbf{P}_{STATS} のスコアの期待値で

あるから、

$$\mu_{STATS_ok} = \frac{\sum_{X \in \{A,C,G,T\}} p_{XX} \mathbf{P}_{STATS}[X, X]}{\sum_{X \in \{A,C,G,T\}} p_{XX}} \quad (4.1)$$

$$\mu_{STATS_sub} = \frac{\sum_{X,Y \in \{A,C,G,T\}, X \neq Y} p_{XY} \mathbf{P}_{STATS}[X, Y]}{\sum_{X,Y \in \{A,C,G,T\}, X \neq Y} p_{XY}} \quad (4.2)$$

$$\mu_{STATS_ins} = \frac{\sum_{Y \in \{A,C,G,T\}} p_{NY} \mathbf{P}_{STATS}[N, Y]}{\sum_{Y \in \{A,C,G,T\}} p_{NY}} \quad (4.3)$$

$$\mu_{STATS_del} = \frac{\sum_{X \in \{A,C,G,T\}} p_{XN} \mathbf{P}_{STATS}[X, N]}{\sum_{X \in \{A,C,G,T\}} p_{XN}} \quad (4.4)$$

により計算される。 μ_{CTC_ok} , μ_{CTC_sub} , μ_{CTC_ins} はそれぞれ、 \mathbf{P}_{CTC} において正しい読み出し、置換エラー、挿入エラーに対応するスコアの期待値であるから、

$$\mu_{CTC_ok} = \frac{1}{N} \sum_n \sum_i \mathbf{P}_{CTC}^{(n)}[i, C_i'^{(n)}] \quad (4.5)$$

$$\mu_{CTC_sub} = \frac{1}{3N} \sum_n \sum_i \sum_{X \in \{A,C,G,T\}, X \neq C_i'^{(n)}} \mathbf{P}_{CTC}^{(n)}[i, X] \quad (4.6)$$

$$\mu_{CTC_ins} = \frac{1}{N} \sum_n \sum_i \mathbf{P}_{CTC}^{(n)}[i, N] \quad (4.7)$$

により計算される。ただし、 N は Basecall された全てのストランドにおける塩基数の総和を表す。

次に、式 (3.2) から (3.5) に示した固定の報酬とペナルティ P_{ok} , P_{sub} , P_{ins} , P_{del} 及び各補正項の重み付けパラメータ α_{ok} , α_{sub} , α_{ins} , α_{del} , β_{ok} , β_{sub} , β_{ins} の最適な値を決定する必要がある。ここで、 P_{ok} には [10] の研究で用いられた値を各符号化率ごとに使用し、 P_{ok} 以外のパラメータを最適化した。各パラメータの最適化は、復号におけるヒープの探索数を目的関数とした、数値微分を用いた勾配降下法により行った。具体的には、まずランダムに生成したバイナリデータを HEDGES で符号

化し，ナノポアシーケンサーでの読み出しをシミュレーションした上でその電流波形に対して Basecaller を適用し，CTC 出力を得た. 勾配降下法によるパラメータの最適化は，この CTC 出力の復号を対象として，以下の手順で行った.

1. 最適化対象の各パラメータを適當な初期値で初期化する.
2. 提案手法を用いて復号を実施し，復号が完了するまでのヒープの探索数を計測する.
3. 各パラメータごとに微小な値 Δ を加えた場合の復号におけるヒープの探索数を計測する.
4. これらの値を用いて各パラメータに関する目的関数の勾配を計算する.
5. 各パラメータを学習率 η を用いて以下のように更新する.

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{\partial J}{\partial \boldsymbol{\theta}} \quad (4.8)$$

ここで， $\boldsymbol{\theta}$ は最適化対象の各パラメータを表し， J は目的関数を表す.

6. 2 から 5 を一定回数繰り返す.

パラメータの最適化については，ナノポアシーケンサーの読み出し特性に基づく補正項と CTC 出力に基づく補正項の両方を用いる場合と，それぞれ一方のみを用いる場合について効果を検証するため， $P_{sub}, P_{ins}, P_{del}, \alpha_{ok}, \alpha_{sub}, \alpha_{ins}, \alpha_{del}, \beta_{ok}, \beta_{sub}, \beta_{ins}$ の 10 個のパラメータに対して行うのに加えて， $\alpha_{ok} = \alpha_{sub} = \alpha_{ins} = \alpha_{del} = 0$ として他の 6 個のパラメータに対してのみ行う場合と $\beta_{ok} = \beta_{sub} = \beta_{ins} = 0$ として他の 7 個のパラメータに対してのみ行う場合についても実施した. なお，これらの最適化は符号化率 0.75, 0.60, 0.50, 0.33 についてそれぞれ独立に行った.

以上の手順により得られた各パラメータを用いて，提案手法の復号性能を評価する. まず，テストに用いるバイナリデータを HEDGES で符号化し，得られた DNA ストランドに対してナノポアシーケンサーでの読み出しを Squigulator でシミュレーションする. 次に，シミュレーションされた電流波形に対して，Bonito Basecaller で Basecall を行う. このとき，最終的な塩基配列だけでなく，CTC デコード前の CTC スコア行列も出力する. この Basecall 結果を用いて，既存手法および提案手法による復号を行い，ヒープの探索数，復号失敗による消失ビット数，RS 復号後の最終的なビット誤り率を計測する. 既存手法を用いた復号では，

Basecaller が output する塩基配列を input とする。提案手法を用いた復号では、CTC スコア行列を input とし、ナノポアシーケンサーの読み出しに特性に基づく補正項のみ用いる場合・CTC 出力に基づく補正項のみ用いる場合・両方の補正項を用いる場合の 3 通りについて評価を行う。このバイナリデータの符号化から復号までの一連のシミュレーションを、符号化率 0.75, 0.60, 0.50, 0.33 についてそれぞれ行う。

4.1.2 実験条件

行列 \mathbf{P}_{STATS} を得るためのシミュレーションでは、ランダムに生成した 400KB のバイナリデータを HEDGES で符号化率 0.50 で符号化した。各ストランドの塩基長は 400nt とし、合計 12240 本のストランドを生成した。Squigulator を用いたナノポアシーケンサーの電流値シミュレーションでは、MinION R9.4.1 フローセルを想定したパラメータ設定を用いた [15]。Bonito Basecaller のモデルには、既存の学習済みモデルである dna_r9.4.1@v2 を HEDGES で符号化した DNA の読み出しに向けてファインチューニングしたものを使用した [11]。ファインチューニング用の学習データには、ランダムに生成した 200KB のバイナリデータを HEDGES で符号化率 0.50、塩基長 400nt で符号化して得られた 6120 本の DNA ストランドと、これらのストランドに対して Squigulator でシミュレーションした電流波形を用いた。dna_r9.4.1@v2 の重みを初期値として、バッチサイズを 64 として 1 エポックあたり 75 ステップの学習を行い、エポック数は 50 としてファインチューニングを行った。Squigulator のシミュレーション条件と Basecaller に使用するモデルは以降の全てのシミュレーションで同様に使用した。 $\{C_i^{(n)}\}$ と $\{C_i'^{(n)}\}$ に対するアライメントには Biopython の Gotoh アルゴリズムを用いたグローバルアライメントを使用した [16, 14]。

パラメータの最適化では、ランダムに生成した 400KB のバイナリデータを符号化率 0.75, 0.60, 0.50, 0.33 で塩基長を 400nt として HEDGES 符号化したものに対して電流波形シミュレーション・Basecall を行って得られた CTC 出力を用いた。 P_{ok} は各符号化率ごとに表 4.1 に示す固定の値を使用した。各パラメータの初期値は $P_{sub} = P_{ins} = P_{del} = 1.0$, $\alpha_{ok} = \alpha_{sub} = \alpha_{ins} = \alpha_{del} = 0.0$, $\beta_{ok} = \beta_{sub} = \beta_{ins} = 0.0$ とした。学習率は符号化率ごとに表 4.2 に示す値を使用し、パラメータの更新を 300 回繰り返した。なお、 \mathbf{P}_{STATS} を得るためのシミュレーション、Bonito Basecaller のファインチューニング、パラメータの最適化に用いたランダムなバイナリデータは全て独立に生成したものである。

既存手法と提案手法の復号性能評価では、テスト用のバイナリデータとして「ころ」(夏目漱石著)の全文のテキストデータ(UTF-8形式)を青空文庫から取得し、ルビ部分(《…》で囲まれた部分)を正規表現により除去したものを使用した[17]。このデータのサイズは481158バイトである。既存手法及び提案手法による復号において、ヒープの上限サイズは1000000とした。

4.1.3 結果と考察

ナノポアシーケンサーの読み出し特性の評価において、ラベル $\{N, A, C, G, T\}$ 間の遷移確率は表4.2に示すように得られた。この結果では、正しい読み取りは全体のうち99.1%を占めており、置換エラーは0.421%，挿入エラーは0.133%，消失エラーは0.302%であった。また、置換エラーはA,G間とC,T間で発生しやすい傾向が見られ、特に、最も確率の低いA,T間の置換と比べてA,G間の置換は約23倍の確率で発生することが分かった。塩基ごとの置換エラー発生率の偏りは、プリン塩基(A,G)とピリミジン塩基(C,T)の化学構造の類似性に起因すると考えられている[18]。NからA,C,G,Tへの遷移で表わされる挿入エラーは、各塩基に対して同程度の確率で発生していた。A,C,G,TからNへの遷移で表わされる消失エラーは、A,GにおいてC,Tよりも比較的高い確率で発生していた。この結果から得られた行列 P_{STATS} は付録の式(??)，各補正項の期待値は表??に示す。各符号化率ごとに最適化されたペナルティと補正項の重み付けパラメータは付録の表??および表??に示す。

既存手法と提案手法の復号性能評価における、ストランド1本あたりの平均ヒープ探索数、復号失敗によるビット消失率、RS復号後の最終的なビット誤り率を表4.3, 4.4, 4.5に示す。ただし、各表において、「Default」は既存手法によるBasecallと復号を表し、「STATS」はナノポアシーケンサーの読み出し特性に基づく補正項のみを適用した場合を表し、「CTC」はCTC出力に基づく補正項のみを適用した場合を表し、「CTC+STATS」は両方の補正項を適用した場合を表す。実験結果では、提案手法を用いることで、各符号化率において既存手法と比べてヒープ探索数を削減し、復号失敗率を低減することができた。ナノポアシーケンサーの読み出し特性に基づく補正項のみ適用した場合、ヒープ探索数が43%から61%削減され、ビット消失率は60%から77%低減した。CTC出力に基づく補正項のみ適用した場合、ヒープ探索数が56%から65%削減され、ビット消失率は68%から83%低減した。また、これら両方の補正項を適用した場合には、ヒープ探索数は

64%から 68%削減され，ビット消失率は 76%から 83%低減し，もっとも良好な結果が得られた．さらに，RS 復号後の最終的なビット誤り率については，符号化率 0.75において，両方の補正項を適用した場合，既存手法と比べて 77%の誤り率低減が達成された．以上の結果から，提案手法により復号の探索が効率化されると同時に，復号失敗によるビット消失率が低減されるため，最終的な復号精度の向上にも寄与することが示された．

4.2 ロングリードに向けた符号化アルゴリズムの改良

4.2.1 実験手法

3.2 節で提案した符号化・復号アルゴリズムの改良手法の性能をシミュレーションにより評価するために，まず既存の HEDGES 符号化アルゴリズムにより，テスト用のバイナリデータを符号化した．得られた DNA ストランドに対して 4.1 節の実験と同様の手法で電流波形のシミュレーションと Basecall を行った．この Basecall 結果に対して 3.1 節の提案手法により復号を行い，各パラメータは 4.1 節の実験で得られたものを使用した．これを DNA ストランド 1 本の塩基長を 500nt から 15000nt の範囲で変化させながら行い，塩基長ごとのビット誤り率を計測した．

次に，同じバイナリデータを用いて，提案手法によるセグメント化を適用した符号化を行い，同様に電流波形のシミュレーションと Basecall を行った．復号においては，まずセパレータの位置を検出する必要があるため，CTC 出力に対して 3.1.2 節で説明した方法で得られた \mathbf{P}_{CTC} に対して以下の手順を用いてセパレータの位置を検出した．

1. $n = l_p + l_{seg}$ とする．ただし l_p はプライマー長， l_{seg} はセグメント長を表す．
2. $n - 30 < k < n + 30$ を満たす整数 k 対して $S_{match}(k)$ を以下の式で計算し， $S_{match}(k)$ を最小とする k をセパレータの位置として選択し，これを k' とする．

$$S_{match}(k) = \sum_{j=0}^{l_{seg}-1} \mathbf{P}_{CTC}[k+j, S[j]] \quad (4.9)$$

ただし $S[j]$ はセパレータ配列の j 番目の塩基を表し， l_{sep} はセパレータ配列の長さを表す．

3. $n = k' + l_{sep} + l_{seg}$ とする.
4. 2 から 3 を繰り返し, 全てのセグメントのセパレータ位置を検出する.

セパレータを検出後, セグメントごとに 3.1 節の提案手法を用いて復号を行った. これをセグメント長は一定として, セグメント数を変化させることで DNA ストランド全体の塩基長を 500nt 程度から 15000nt 程度の範囲で変化させながら行い, 塩基長ごとのビット誤り率を計測した.

これらのシミュレーション結果を比較することで提案手法による符号化・復号アルゴリズムの性能を評価した.

4.2.2 実験条件

テスト用に用いたバイナリデータは 4.1 節の実験で使用したものと同じである. また, Squigulator による電流値シミュレーションのパラメータ設定, Basecaller のモデルについても 4.1 節の実験と同様に行った. 既存手法及び提案手法による符号化における符号化率は 0.60 とした. 提案手法におけるセグメント化では, セグメント長は 341nt とし, セパレータは 10nt の塩基配列”GTACTGCATG”とした. 復号におけるヒープ探索では既存手法, 提案手法ともにヒープの上限サイズは 1000000 とした.

4.2.3 結果と考察

既存手法と提案手法における, DNA ストランドの塩基長とビット誤り率の関係を図 4.1 に示す. 実験結果では, 既存手法では [10] の研究の報告と同様に, 塩基長が大きくなるほどビット誤り率が直線的に増加することを確認した. 一方, 提案手法においては塩基長が長くなてもビット誤り率はほぼ変化しなかった. 既存手法における復号精度の低下は, 復号におけるヒープ探索数の増加に伴う復号失敗率の増加が主な要因であり, 提案手法ではセグメントごとにハッシュ関数の入力とヒープをリセットすることで, ヒープサイズを抑えつつ, 復号失敗が発生した場合においても次のセグメントからの復号を継続できるため, 復号精度の低下を抑制できたと考えられる.

第5章 結論

本研究では、DNAストレージの読み出しにおいてナノポアシーケンサーを用いることを前提とした誤り訂正手法を提案し、その性能をシミュレーションにより評価した。既存手法である HEDGES は、DNAストレージにおけるシーケンス制約を満たしつつ挿入・消失・置換エラーへの耐性を持つ誤り訂正符号であるが、その復号アルゴリズムにおいて、ナノポアシーケンサーでの DNA 読み出しを前提とした最適化が行われていないという課題と、DNAストランドの塩基長が長い場合に復号精度が低下するという課題が存在した。

本研究では、これらの課題を解決するために、まず HEDGES の復号アルゴリズムにおいて、ナノポアシーケンサーの統計的な読み出し特性と Basecaller の CTC 出力を考慮した探索手法を提案した。実験結果より、提案手法では既存手法と比較して複号アルゴリズムにおける探索数を 64% から 68% 削減し、復号失敗によるビット消失率を 76% から 83% 低減できることを示した。さらに、ロングリードに向けた符号化・復号アルゴリズムの改良を提案し、セグメント化による手法を用いることで長い塩基長の DNA 配列を扱う場合の復号精度の低下を抑制できることを示した。

今後の課題としては、以下の点が挙げられる。第一に、ナノポアシーケンサーの出力特性をより詳細にモデル化することである。本研究では、ラベル間の遷移確率のみを考慮した単純なモデルを用いたが、塩基配列のコンテキスト依存性や、バースト的なエラー発生など、より複雑な特性を考慮することで、復号における探索をより最適化できる可能性がある。第二に、Basecaller の最新アーキテクチャに対応することである。本研究で使用した Bonito は CNN-LSTM-CTC 構成であるが、最新の Basecaller では CRF を用いたアーキテクチャなど、より先進的な機械学習手法が採用されており、これらに対応することで復号精度のさらなる向上が期待できる。第三に、実際の DNA を用いた実験的検証である。本研究はシミュレーションのみによる評価であるため、実際に合成 DNA を用いたナノポアシーケンサーでの読み出し実験を行うことで、提案手法の実用性を検証する必要がある。

謝 辞

本研究を進めるにあたり、ご指導・ご助言を賜りました佐藤高史教授に深く感謝申し上げます。研究会やミーティングを通じて多くの貴重なご指摘・ご助言をいただきました粟野皓光准教授、橋本昌宜教授、上野嶺准教授、白井僚助教、新津葵一教授、劉昆洋助教に深く感謝致します。日々の研究において様々な助言をいただきました小池健文氏に深く感謝いたします。研究室生活において様々な面から支えて頂いた佐藤高史研究室支援職員の西山修子氏、上西香織氏に感謝致します。最後に、日頃から様々なご支援、ご協力を頂きました佐藤研究室、橋本研究室、新津研究室の皆様に感謝致します。

参 考 文 献

- [1] Petroc Taylor, Data generation volume worldwide 2010–2029, (2025), <https://www.statista.com/statistics/871513/worldwide-data-created/>, Accessed: 2026-01-26.
- [2] DNA Data Storage Alliance, Preserving our digital legacy: An introduction to dna data storage, Technical report, (DNA Data Storage Alliance, San Diego, CA, 2021).
- [3] David Reinsel, John Gantz, and John Rydning, Data age 2025: The evolution of data to life-critical — don't focus on big data; focus on the data that's big, Idc white paper, sponsored by seagate, (International Data Corporation (IDC), Framingham, MA, 2017).
- [4] A. Akash, E. Bencurova, and T. Dandekar, How to make DNA data storage more applicable, *Trends in Biotechnology*, Vol. 42, No. 1, (2024), pp. 17–30.
- [5] Luis Ceze, Jeff Nivala, and Karin Strauss, Molecular digital data storage using DNA, *Nature Reviews Genetics*, Vol. 20, No. 8, (2019), pp. 456–466.
- [6] S. A. Simon, J. Zhai, R. S. Nandety, K. P. McCormick, J. Zeng, D. Mejia, and B. C. Meyers, Short-read sequencing technologies for transcriptional analyses, *Annual Review of Plant Biology*, Vol. 60, (2009), pp. 305–333.
- [7] S. L. Amarasinghe, S. Su, X. Dong, L. Zappia, M. E. Ritchie, and Q. Gouil, Opportunities and challenges in long-read sequencing data analysis, *Genome Biology*, Vol. 21, No. 1, (2020), p. 30.
- [8] M. Uhlen and S. R. Quake, Sequential sequencing by synthesis and the next-generation sequencing revolution, *Trends in Biotechnology*, Vol. 41, No. 12, (2023), pp. 1565–1572, Epub 2023 Jul 21; PMID: 37482467.

- [9] Adam Dorey and Stefan Howorka, Nanopore DNA sequencing technologies and their applications towards single-molecule proteomics, *Nature Chemistry*, Vol. 16, No. 3, (2024), pp. 314–334.
- [10] William H. Press, John A. Hawkins, Stephen K. Jones, Jeffrey M. Schaub, and Ilya J. Finkelstein, Hedges error-correcting code for dna storage corrects indels and allows sequence constraints, *Proceedings of the National Academy of Sciences*, Vol. 117, No. 31, (2020), pp. 18489–18496.
- [11] Chris Seymour, Oxford Nanopore Technologies Ltd., Bonito: A pytorch base-caller for oxford nanopore reads, (2019), Oxford Nanopore Technologies, Ltd. Public License, v. 1.0.
- [12] Joao Gervasio, Henrique Oliveira, André Costa Martins, Joao Pesquero, Bruno Verona, and Natalia Cerize, How close are we to storing data in dna?, *Trends in Biotechnology*, Vol. 42, (2023).
- [13] Peter E. Hart, Nils J. Nilsson, and Bertram Raphael, A formal basis for the heuristic determination of minimum cost paths, *IEEE Transactions on Systems Science and Cybernetics*, Vol. 4, No. 2, (1968), pp. 100–107.
- [14] Osamu Gotoh, An improved algorithm for matching biological sequences, *Journal of Molecular Biology*, Vol. 162, No. 3, (1982), pp. 705–708.
- [15] Hasindu Gamaarachchi, James M Ferguson, Hiruna Samarakoon, Kisaru Liyanage, and Ira W Deveson, Squigulator: simulation of nanopore sequencing signal data with tunable noise parameters, *bioRxiv*, (2023), pp. 2023–05.
- [16] Peter J. A. Cock, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg, Thomas Hamelryck, Frank Kauff, Bartek Wilczynski, and Michiel J. L. de Hoon, Biopython: freely available python tools for computational molecular biology and bioinformatics, *Bioinformatics*, Vol. 25, No. 11, (2009), pp. 1422–1423.
- [17] 夏目漱石, こころ, (1914), <https://www.aozora.gr.jp/cards/000148/card773.html>.

- [18] Clara Delahaye and Jacques Nicolas, Sequencing dna with nanopores: Troubles and biases, *PLOS ONE*, Vol. 16, No. 10, (2021), pp. 1–29.

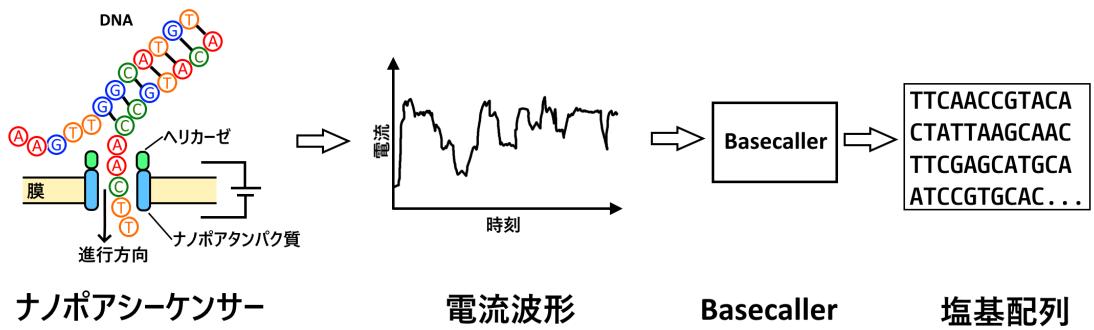


図 2.1: ナノポアシーケンサーの原理と手順

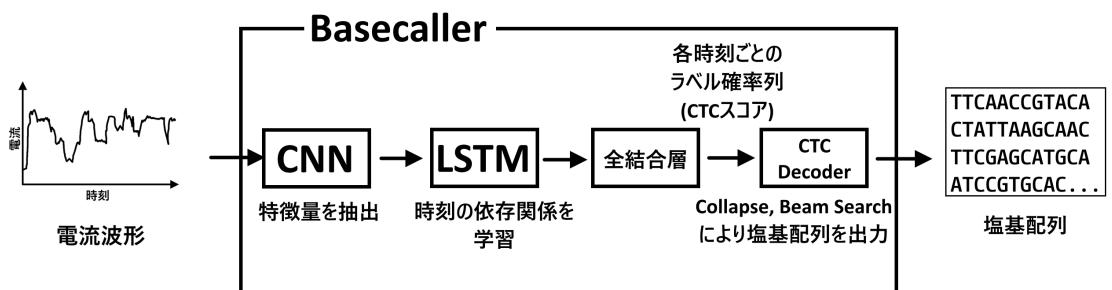


図 2.2: Bonito Basecaller のアーキテクチャ概要

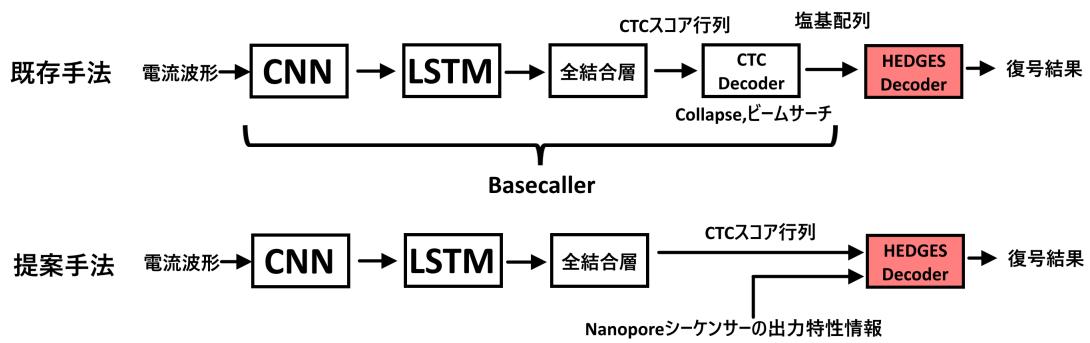


図 3.1: 提案手法の概要

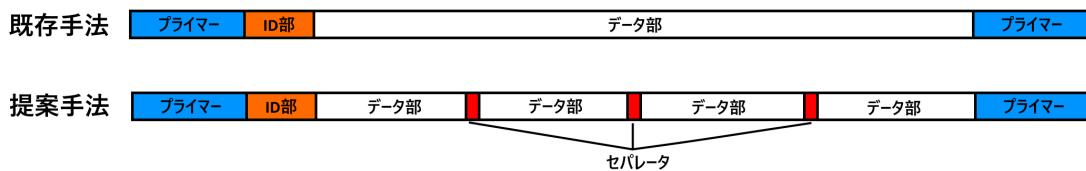


図 3.2: 提案手法による符号化方式

表 4.1: 符号化率ごとの報酬値 P_{ok}

符号化率	P_{ok}
0.75	-0.035
0.60	-0.082
0.50	-0.127
0.33	-0.229

表 4.2: 符号化率ごとの学習率 η

符号化率	η
0.75	1.3×10^{-7}
0.60	4.1×10^{-7}
0.50	1.0×10^{-6}
0.33	3.8×10^{-6}

表 4.2: ラベル間の遷移確率

入力 \ 出力	N	A	C	G	T
N	0.000	3.11×10^{-4}	3.11×10^{-4}	3.18×10^{-4}	3.86×10^{-4}
A	9.28×10^{-4}	0.232	5.03×10^{-4}	1.01×10^{-3}	3.94×10^{-5}
C	5.95×10^{-4}	3.10×10^{-4}	0.244	3.56×10^{-4}	1.27×10^{-4}
G	8.27×10^{-4}	9.00×10^{-4}	6.60×10^{-4}	0.269	5.13×10^{-5}
T	6.73×10^{-4}	4.36×10^{-5}	1.55×10^{-4}	5.85×10^{-5}	0.247

表 4.3: ストランド 1 本あたりの平均ヒープ探索数

code rate	0.75	0.6	0.5	0.33
Default	135550	45219	21291	12290
STATS	77281	23707	10893	4785
CTC	53098	17966	9353	4354
CTC+STATS	48507	14972	7625	3894

表 4.4: ビット消失率

code rate	0.75	0.6	0.5	0.33
Default	0.03083	0.00949	0.00341	0.00328
STATS	0.00741	0.00376	0.00121	0.00076
CTC	0.00789	0.00265	0.00111	0.00055
CTC+STATS	0.00693	0.00229	0.00069	0.00056

表 4.5: RS 復号後のビット誤り率

code rate	0.75	0.6	0.5	0.33
Default	0.022981	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$
STATS	0.059977	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$
CTC	0.008228	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$
CTC+STATS	0.005173	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$

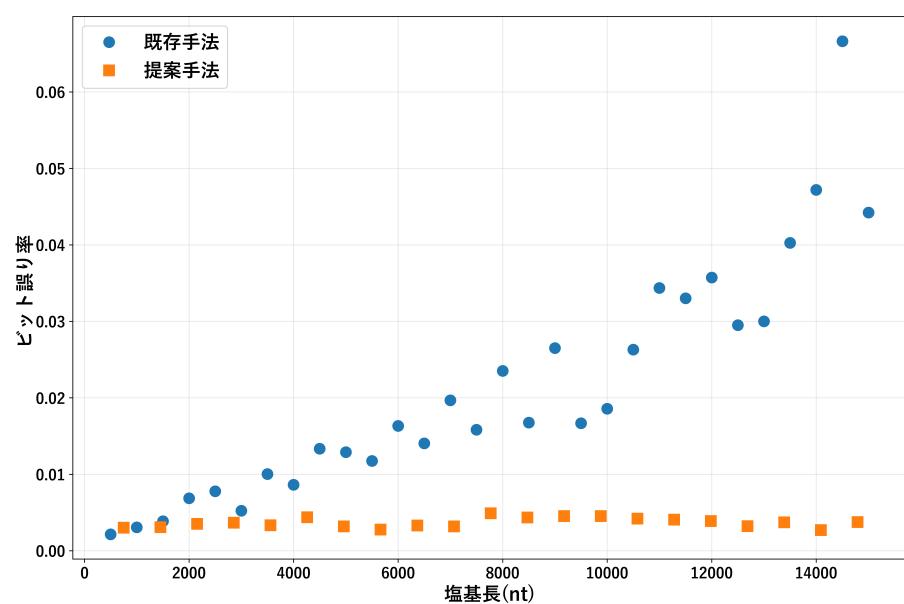


図 4.1: 塩基長とビット誤り率の関係