

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

In today's information society, large amounts of digital data are generated every day, and the amount of data worldwide continues to grow explosively. DNA storage has emerged as a promising solution for long-term archival storage due to its high information density and durability. While nanopore sequencers offer advantages in reading long DNA sequences at high speeds, they suffer from lower accuracy compared to short-read sequencers, particularly with insertion and deletion errors. Although existing error-correcting codes have been proposed for DNA storage with capability to correct these errors, their decoding algorithms have not been optimized for the specific characteristics of nanopore sequencers.

This study aims to optimize existing error correction methods specifically for DNA storage systems using nanopore sequencers. The proposed method incorporates two key improvements: (1) optimization of reward and penalty parameters based on statistical transition matrices that model nanopore reading characteristics, and (2) integration of confidence scores from the basecaller's CTC (Connectionist Temporal Classification) output into the decoding process. Additionally, to address accuracy degradation with longer DNA sequences, we propose a segmentation-based encoding scheme that divides sequences into short segments with separators and reinitializes the hash function at segment boundaries.

Simulation results demonstrate that the proposed method improves both decoding efficiency and accuracy. The optimized decoding algorithm reduces heap search operations by 64-68% and bit deletion rates by 76-83% compared to the existing method. The segmentation scheme maintains stable decoding accuracy even for DNA sequences up to 15,000nt without significantly increasing computational complexity.

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 DNA ストレージの概要	3
2.2 HEDGES	4
2.2.1 符号化	4
2.2.2 復号	6
2.3 ナノポアシーケンサーとベースコーラー	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	15
4.1 復号の探索効率化	15
4.1.1 実験方法	15
4.1.2 実験条件	16
4.1.3 結果と考察	16
4.2 ロングリードに向けた符号化アルゴリズムの改良	17
4.2.1 実験手法	17
4.2.2 実験条件	18
4.2.3 結果と考察	18
第 5 章 結論	19

謝 辞	20
参 考 文 献	21
付 錄 A 復号の探索効率化における予備実験	24
A.1 実験方法	24
A.2 実験条件	26
A.3 結果	27

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、デジタルデータをアデニン(A), シトシン(C), グアニン(G), チミン(T)の4種類の塩基配列に符号化し、人工的に合成したDNA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^{*}の短い塩基長の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の長い塩基長の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法は大量のDNA断片を並列に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

* 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。

るナノポアシーケンサーは一度に数十万 bp 程度の長い塩基長の DNA を高速に解析することができるため、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による読み出しを前提とした 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、連続した同一塩基（ホモポリマー）の禁止や 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最適化は行われていない。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と述べられており、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 DNAストレージの概要

DNAストレージは、デジタルデータをDNAの塩基配列に符号化し、人工的に合成したDNAを保存媒体としてデータを保存する技術である。DNAストレージは以下の手順でデータの保存と読み出しを行う。

1. 符号化

まずバイナリデータをDNAの塩基配列に符号化する。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ホモポリマーの禁止やGC含有率が一定範囲内に収まるようにする制約がある[11]。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する耐性を持たせる必要がある。

2. 保存

符号化によって得られた塩基配列に基づいて人工的にDNAを合成する。合成されたDNAは保存密度と安定性を高めるため、乾燥状態で保管する方法が一般的である[12]。これにより数千年にわたるデータの長期保存が可能であると考えられている[13]。

3. データ読み出し

保存されたDNAからデータを読み出すにはまずDNAを取り出し、SBS法や2.3節で詳しく述べるナノポアシーケンサーなどのDNAシーケンサーを用いて塩基配列を解析する。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った場合.

これらのエラーを含む塩基配列から、誤り訂正を行いつつ元のバイナリデータを復元する.

2.2 HEDGES

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].

2.2.1 符号化

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 鎖からなり, DNA 鎖の塩基長は自由に設定できる. 符号化はビット列から誤り訂正能力をもたせたビット列への変換を行うリードソロモン (RS) 符号化と, そのビット列からさらに冗長性を付加させた塩基配列に変換する HEDGES 符号化の 2 段階で行う. 図 2.1 に HEDGES 符号化の概要を示す. はじめに, 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 RS 符号化を適用し, 1 パケット分のビット列を得る. 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. 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できる.

以下に, HEDGES における具体的な符号化手順を示す. まず, 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.

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

任意の符号化率を実現するためには, 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. そのため, ビット列 $\{b_i\}$ を 0 から 2 の長

さを持つビット列 v_i に分割する。以下に、符号化率 $r = 0.750$ の場合と $r = 0.500$ の場合と $r = 0.333$ の場合の v_i の構成例を示す。

$$r = 0.750 : \quad v_0 = b_0b_1, \quad v_1 = b_2, \quad v_2 = b_3b_4, \quad v_3 = b_5, \quad v_4 = b_6b_7 \dots \quad (2.2)$$

$$r = 0.500 : \quad v_0 = b_0, \quad v_1 = b_1, \quad v_2 = b_2, \quad v_3 = b_3 \dots \quad (2.3)$$

$$r = 0.333 : \quad v_0 = b_0, \quad v_1 = b_1, \quad v_2 = 0, \quad v_3 = b_2, \quad v_4 = b_3, \quad v_5 = 0 \dots \quad (2.4)$$

ただし、式(2.2)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る。 $r = 0.500$ では、1塩基に1ビットを割り当てる。 $r = 0.333$ 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。各ビット列 v_i に対して塩基 C_i を割り当てるとする。

$$C_i, \quad i = 0, 1, 2, \dots, N, \quad C_i \in \{C_i^*\} \quad (2.5)$$

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り、直前の数塩基の出力に基づいて決定される。例えば、ホモポリマーの最大長に制約を設ける場合、直前の塩基出力が最大長のホモポリマーとなつていれば、次の出力候補から同一の塩基は除外される。また、GC 含有率の制約を設ける場合は、直前数塩基の GC 含有率に基づいて A と T または C と G を出力候補から除外する。したがって $\{C_i^*\}$ の要素数 $\#C_i^*$ は $0 < \#C_i^* \leq 4$ の範囲で変化する。 $\{C_i^*\}$ 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。

$$K_i = F(S_i, I_i, V_i) \quad (2.6)$$

$$C_i = K_i + v_i \pmod{\#C_i^*} \quad (2.7)$$

ここで、 F はハッシュ関数であり、 S_i は DNA 鎖の ID に基づくソルト、 I_i はビット位置のインデックス、 V_i は直前の 12 ビット分の値である。すなわち、各ビットに対応する塩基は、DNA 鎖の 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。

HEDGES でエンコードするビット列の先頭は DNA 鎖の 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 鎖の出力となる。

2.2.2 復号

HEDGES で符号化された塩基配列から元のデータを復元するには、まず HEDGES 復号により置換・挿入・消失エラーを訂正しながら塩基配列からビット列への復元を行い、次に外部符号である RS 符号の復号を行うことで、HEDGES 復号で復元に失敗したビット列の訂正を行う（図 2.1）。HEDGES 復号では Greedy Exhaustive Search(貪欲探索法) に基づく復号アルゴリズムを用いる [14]。このアルゴリズムは図 2.2 に示すようなビット列の候補に基づくヒープ構造を探索することで、入力された塩基配列に対して尤もらしいビット列を選定する手法である。各要素には候補のビット列と、挿入と消失を仮定する skew と呼ばれるパラメータが存在する。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、 $\Delta \in \{-1, 0, 1\}$ で表される。復号プロセスとしては、まずビット列 v_i と $\text{skew}\Delta$ の全ての組み合わせに対して候補を生成し、ヒープに追加する。各候補において、その v_i から式 (2.7) を用いて予測される塩基 C_i と、候補に対応する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を加算し、不一致であればペナルティを加算する。各ステップにおいてスコアが最小となる候補を選択し、これを親として次の候補を展開していくことで、ヒープ構造を構築する。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。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、DNA 鎖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。

2.3 ナノポアシーケンサーとベースコーラー

ナノポアシーケンサーは DNA が溶液中で帶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]。図 2.3 にナノポアシーケンサーによる DNA 解析の概要を示す。ナノポアシーケンサーのフローセルと呼ばれるセンサー基板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 は負に帶電しているため、電圧を印加す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、ナノポアへ誘導する。誘導された 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、一定の速度でナノポアを通過するように制御される。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ベースコールと呼ばれ、ベースコーラーと呼ばれる機械学習モデルによって行われる。本研究で使用する Bonito と呼ばれるベースコーラーのアーキテクチャの概要を図 2.4 に示す [15]。Bonito では、まず入力された電流信号に対して 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、空白 (N) 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. 空白ラベル (N) を削除する。

これにより、例えばラベル列 “AANNCCGNGGGT” 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一度に数百 bp から数百万 bp の塩基長の長い DNA を直接的に解析できることや、高速な解析が可能であることなどが挙げられる。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、ベースコーラー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]。

2.4 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。

- 既存研究では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統計的な出力特性に基づいた報酬とペナルティの最適化は行われていない。
- 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、ベースコーラーが出力する各時点における各塩基の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、長い塩基配列では探索数が爆発し、ヒープの上限に達しやすく、復号失敗が起こりやすいため、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、ベースコーラー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ベースコーラー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、ベースコーラーにおける 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

ナノポアシーケンサーの読み出し特性をモデル化するために、事前に統計的なエラー率を表現する遷移行列を作成する。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、シーケンスアライン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 [16]。アラインメントされた 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ンメント情報から、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

2.3 節で説明したように、ベースコーラーは通常、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ベースコーラー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、2.2.2 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 $L \times 5$ 行列 \mathbf{S} で与えられるとする。ここで、 L 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、2.3 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続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、空白ラベル (N) に対する確率が最大となっている行を削除する。

\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空白が出力される確率に基づくスコ

アを表す。2.2.2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節で述べたナノボアシーケンサーの読み出し特性とベースコーラーの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。

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

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

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

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

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付録で詳しく述べる。

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 DNA の塩基長を長くすると復号精度が低下する。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、DNA 鎖の 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 DNA 鎖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 400nt の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 に初期化して符号化する。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。セパレータのパターンとしては、ホモポリマーがなく、GC 含量が 50% 程度となるような塩基配列を選択する。本手法では”GTACTGCATG”という塩基配列をセパレータとして用いる。

復号時にはまず、以下の手順でセパレータの位置を検出する。

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。

$$S_{match}(k) = \sum_{j=0}^{l_{sep}-1} \mathbf{P}_{CTC}[k+j, S[j]] \quad (3.6)$$

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.

セパレータの位置が特定できたら, 各セグメントごとに 2.2.2 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 に初期化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

本章では、シミュレーションを用いて提案手法の性能を評価する。4.1節では3.1節で提案した報酬とペナルティの最適化手法を用いる場合の復号時の探索数と復号精度を計測し、既存手法を用いる場合と比較してその有効性を評価する。4.2節では3.2節で提案したロングリードに向けた符号化・復号アルゴリズムの改良手法を用いた場合の復号精度を計測し、既存手法を用いる場合と比較してその有効性を評価する。

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の実験手順の概要を図4.1に示す。予備実験として、シミュレーションによりナノポアシーケンサーの読み出し特性を評価し、行列 P_{STATS} を得た。次に、ランダムバイナリを用いた符号化・復号シミュレーションにおいて、復号時のヒープ探索数を目的関数とする勾配降下法により、式(3.2)から(3.5)における各補正項のパラメータを最適化した。パラメータの最適化は複数の符号化率に対して独立に行い、ナノポアシーケンサーの出力特性に基づく補正項のみを用いる場合と、CTC出力に基づく補正項のみを用いる場合、両方の補正項を用いる場合の3通りについて行った。以上の予備実験の具体的な手順や条件については付録で詳しく述べる。

予備実験で得られた最適なパラメータを用いて、提案手法の復号性能を評価した。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鎖に対し、ナノポアシーケンサーによる電流波形をSquigulator[17]と呼ばれるシミュレータを用いてシミュレーションした。次に、シミュレーションされた電流波形に対して、Bonitoでベースコールを行った。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ようにした。このベースコール結果を用いて、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した. 既存手法を用いた復号では, ベースコーラーが出力する塩基配列を入力とする. 提案手法を用いた復号では, CTC スコア行列を入力とし, 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った. 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, 複数の符号化率について実施した.

4.1.2 実験条件

テスト用のバイナリデータには, 「こころ」(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 [18]. このデータのサイズは 481158 バイトである. このバイナリデータを HEDGES で符号化は, 符号化率 0.75, 0.60, 0.50, 0.33 で行い, 塩基長は 400nt とした. 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 [17]. Bonito のモデルには, 既存の学習済みモデルを 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 [15]. 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

既存手法と提案手法の復号性能評価における, DNA 鎖 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 4.1, 4.2, 4.3 に示す. ただし, 各表において, 「STATS のみ」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 のみ」は CTC 出力に基づく補正項のみを適用した場合を表し, 「STATS+CTC」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、セグメント化を行わない既存手法と、提案手法によりセグメント化を行う場合で符号化・復号のシミュレーションを行い、復号における探索数、復号精度を比較した。

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 鎖に対して 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ベースコールを行った。このベースコール結果に対して 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 鎖 1 本の塩基長を 500nt から 15000nt の範囲で変化させながら行い、HEDGES 復号におけるビット誤り率を計測した。また、既存手法による符号化は、塩基長を変えてもヒープ上限は一定値とする場合と、ヒープ上限を塩基長に比例して増加させる場合の 2 通りについて評価を行った。

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ベースコールを行った。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 P_{CTC} に対して 3.2 節の方法でセパレータの検出を行った後、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 鎖全体の塩基長を 500nt 程度から 15000nt 程度の範囲で変化させながら行い、HEDGES 復号におけるビット誤り率を計測した。

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、ベースコーラー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400ntとした。復号におけるヒープ探索では、既存手法では塩基長に関わらずヒープサイズの上限を1000000とした場合と、塩基長が400ntのときのヒープサイズ上限を1000000とし、これを基準としてヒープサイズ上限を塩基長に比例して増加させる場合の2通りについて評価を行った。提案手法におけるセグメントごとの復号では、ヒープサイズの上限を各セグメントごとで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鎖の塩基長とビット誤り率の関係を図4.2に、塩基長とDNA鎖1本あたりの平均ヒープ探索数の関係を図4.3に示す。ここで、既存手法(1)はヒープサイズの上限を塩基長に関わらず一定とした場合、既存手法(2)はヒープサイズの上限を塩基長に比例して増加させる場合を表す。実験結果では、既存手法(1)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。また、既存手法(2)では塩基長に比例してヒープサイズの上限を増加させることで塩基長に対してビット誤り率を一定に抑えることができたが、ヒープ探索数は3つの手法の中で最も大きくなつた。一方、提案手法においては塩基長が長くなってもビット誤り率はほぼ変化しなかつた。提案手法におけるヒープ探索数の上限は、各セグメントごとに1000000としているため、DNA鎖全体での上限を1000000とした既存手法(1)と比較すると24%ほど増加したが、既存手法(2)と比較すると21%ほど削減できた。

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初期化することで、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鎖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ベースコーラーの 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復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 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、ベースコーラー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、最新のベースコーラーでは 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] 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).
- [12] Dave Landsman and Karin Strauss, The dna data storage model, *IEEE Computer*, Vol. 56, No. 7, (2023), pp. 78–85.
- [13] Robert N. Grass, Reinhard Heckel, Michela Puddu, Daniela Paunescu, and Wendelin J. Stark, Robust chemical preservation of digital information on dna in silica with error-correcting codes, *Angewandte Chemie International Edition*, Vol. 54, No. 8, (2015), pp. 2552–2555.
- [14] 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.
- [15] 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.
- [16] Osamu Gotoh, An improved algorithm for matching biological sequences, *Journal of Molecular Biology*, Vol. 162, No. 3, (1982), pp. 705–708.
- [17] 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.
- [18] 夏目漱石, こころ, (1914), <https://www.aozora.gr.jp/cards/000148/card773.html>.

- [19] 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.
- [20] Clara Delahaye and Jacques Nicolas, Sequencing dna with nanopores: Troubles and biases, *PLOS ONE*, Vol. 16, No. 10, (2021), pp. 1–29.

付 錄 A 復号の探索効率化における予備実験

A.1 実験方法

まず、ナノポアシーケンサーの出力特性を表す行列 \mathbf{P}_{STATS} を得るため、ランダムなバイナリデータを HEDGES で符号化した複数の DNA 鎖に対し、ナノポアシーケンサーによる電流波形を Squigulator を用いてシミュレーションした。次に、シミュレーションされた電流波形に対して Bonito 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 j 番目の DNA 鎖の i 番目の塩基を $C_i^{(j)}$ と表し、この塩基配列を $\{C_i^{(j)}\}$ とする。また、 $\{C_i^{(j)}\}$ のベースコールにおいて得られた \mathbf{P}_{CTC} を $\mathbf{P}_{CTC}^{(j)}$ と表すこととし、 $\mathbf{P}_{CTC}^{(j)}$ 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 $\{C_i'^{(j)}\}$ と表す。この $\{C_i'^{(j)}\}$ と $\{C_i^{(j)}\}$ に対してグローバルアラインメントを行い、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 [16]。これを全ての j 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、

$$\mu_{STATS_ok} = \frac{\sum_{X \in \{A,C,G,T\}} p_{XX} \mathbf{P}_{STATS}(X, X)}{\sum_{X \in \{A,C,G,T\}} p_{XX}} \quad (\text{A.1})$$

$$\mu_{STATS_sub} = \frac{\sum_{\substack{X,Y \in \{A,C,G,T\}, X \neq Y}} p_{XY} \mathbf{P}_{STATS}(X, Y)}{\sum_{\substack{X,Y \in \{A,C,G,T\}, X \neq Y}} p_{XY}} \quad (\text{A.2})$$

$$\mu_{STATS_ins} = \frac{\sum_{\substack{Y \in \{A,C,G,T\}}} p_{NY} \mathbf{P}_{STATS}(N, Y)}{\sum_{Y \in \{A,C,G,T\}} p_{NY}} \quad (\text{A.3})$$

$$\mu_{STATS_del} = \frac{\sum_{\substack{X \in \{A,C,G,T\}}} p_{XN} \mathbf{P}_{STATS}(X, N)}{\sum_{X \in \{A,C,G,T\}} p_{XN}} \quad (\text{A.4})$$

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、

$$\mu_{CTC_ok} = \frac{1}{M} \sum_j \sum_i \mathbf{P}_{CTC}^{(j)}(i, C_i'^{(j)}) \quad (\text{A.5})$$

$$\mu_{CTC_sub} = \frac{1}{3M} \sum_j \sum_i \sum_{\substack{X \in \{A,C,G,T\}, X \neq C_i'^{(j)}}} \mathbf{P}_{CTC}^{(j)}(i, X) \quad (\text{A.6})$$

$$\mu_{CTC_ins} = \frac{1}{M} \sum_j \sum_i \mathbf{P}_{CTC}^{(j)}(i, N) \quad (\text{A.7})$$

により計算される。ただし、 M はベースコールされた全ての DNA 鎖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。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ベースコーラーを適用し, 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当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.

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \eta \frac{\partial J}{\partial \boldsymbol{\theta}} \quad (\text{A.8})$$

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.

A.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DNA鎖の塩基長は 400nt とし, 合計 12240 本のDNA鎖を生成した. Squigulator による電流値シミュレーションのパラメータ設定, ベースコーラーのモデルについては4.1節の実験と同様である.

ベースコーラーのファインチューニングには、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 鎖と、これらの DNA 鎖に対して Squigulator でシミュレーションした電流波形を学習データとして用いた。既存の学習済みモデルである 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。

$\{C_i^{(n)}\}$ と $\{C'_i^{(n)}\}$ に対するアラインメントには Biopython の Gotoh アルゴリズムを用いたグローバルアラインメントを使用した [19, 16]。

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・ベースコールを行って得られた CTC 出力を用いた。 P_{ok} は各符号化率ごとに表 A.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 A.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、ベースコーラー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。

A.3 結果

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 A.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 [20]。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 \mathbf{P}_{STATS} は式 (A.9) のようになった。各補正項の期待値は表 A.3 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

表 A.4 および表 A.5 に示す。

$$\boldsymbol{P}_{STATS} = \begin{bmatrix} -1.20 & -0.650 & -0.650 & -0.643 & -0.613 \\ -0.580 & 0.130 & -0.670 & -0.590 & -1.25 \\ -0.620 & -0.650 & 0.110 & -0.640 & -1.09 \\ -0.550 & -0.520 & -0.590 & 0.160 & -1.53 \\ -0.590 & -1.53 & -1.09 & -1.25 & 0.120 \end{bmatrix} \quad (\text{A.9})$$

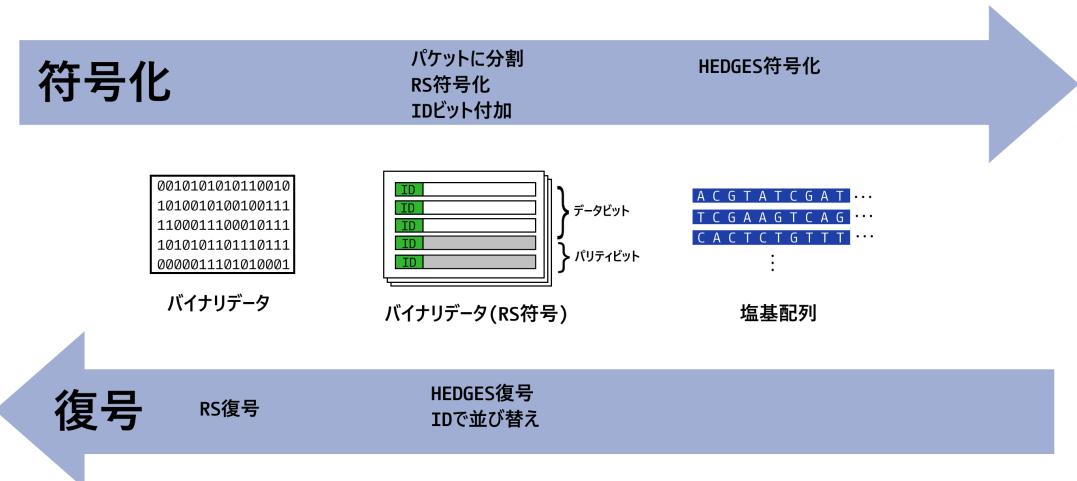


図 2.1: HEDGES 符号化の概要

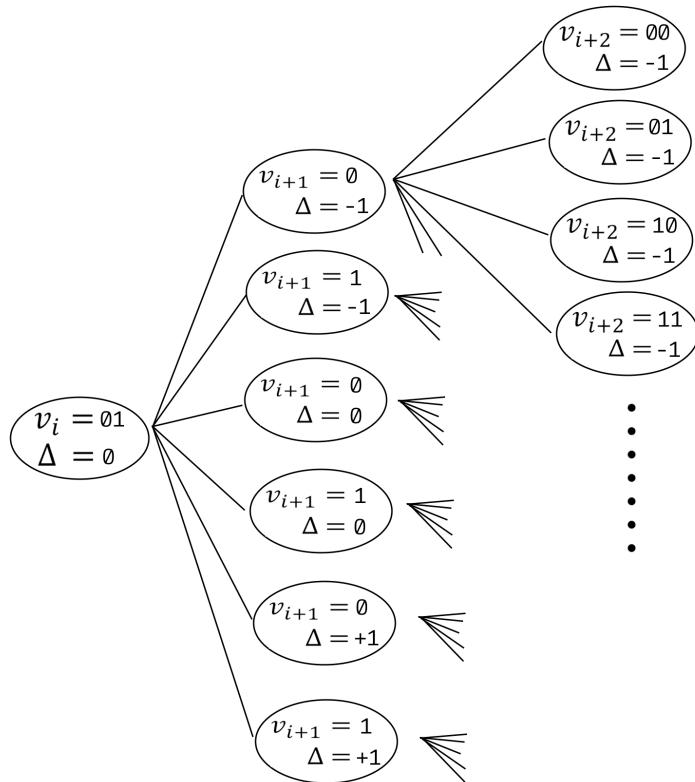


図 2.2: HEDGES 復号におけるヒープ構造

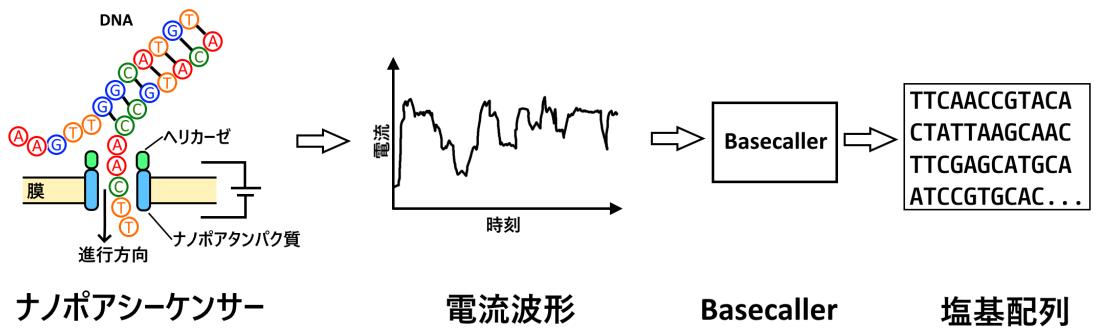


図 2.3: ナノポアシーケンサーの原理と手順

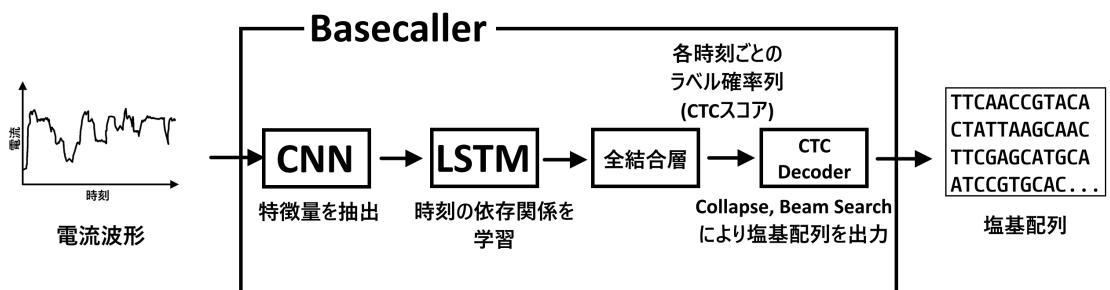


図 2.4: Bonito アーキテクチャ概要

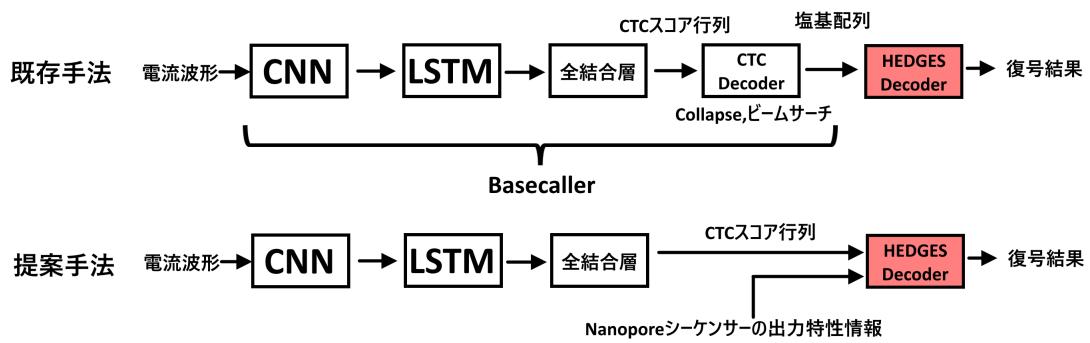


図 3.1: 提案手法の概要

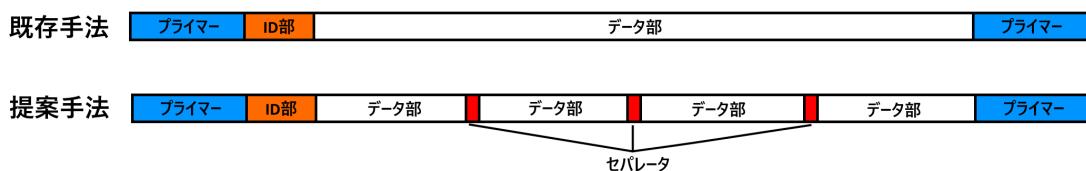


図 3.2: 提案手法による符号化方式

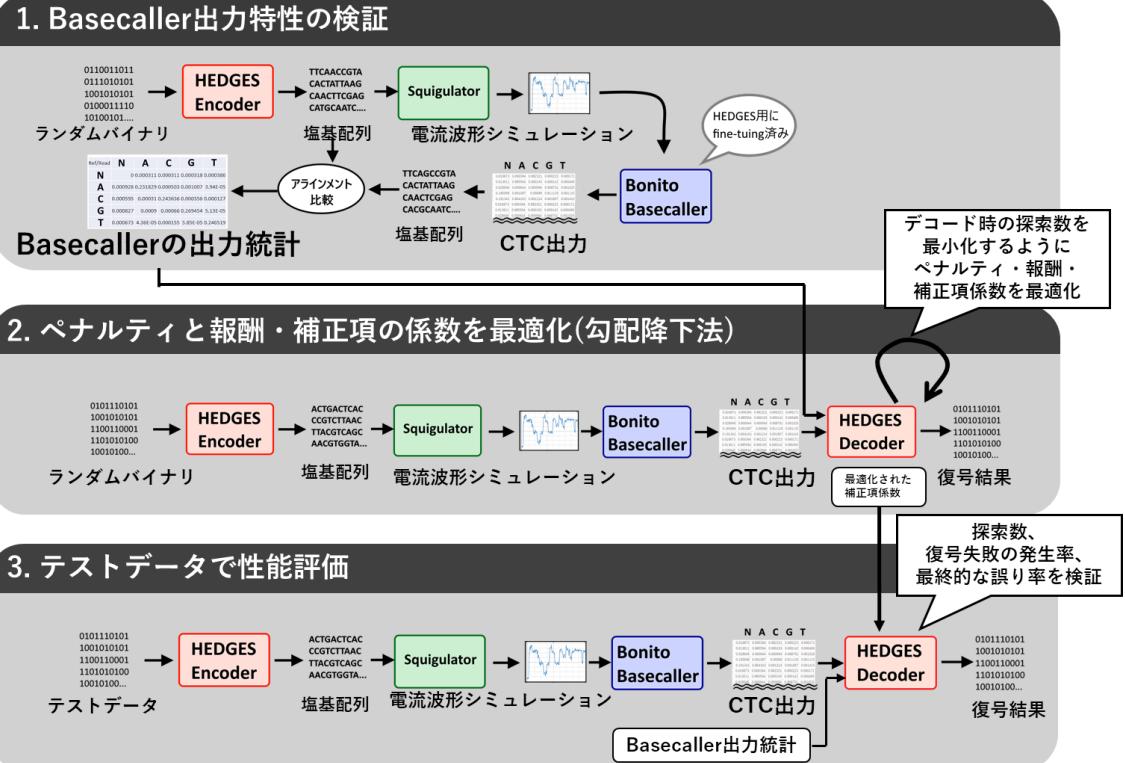


図 4.1: シミュレーション手順概要

表 4.1: DNA 鎖 1 本あたりの平均ヒープ探索数

符号化率	0.75	0.6	0.5	0.33
既存手法	135550	45219	21291	12290
提案手法	STATS のみ	77281	23707	10893
	CTC のみ	53098	17966	9353
	STATS+CTC	48507	14972	7625

表 4.2: ビット消失率

符号化率	0.75	0.6	0.5	0.33
既存手法	0.03083	0.00949	0.00341	0.00328
提案手法	STATS のみ	0.00741	0.00376	0.00121
	CTC のみ	0.00789	0.00265	0.00111
	STATS+CTC	0.00693	0.00229	0.00069

表 4.3: RS 復号後のビット誤り率

符号化率	0.75	0.6	0.5	0.33
既存手法	0.022981	$< 10^{-6}$	$< 10^{-6}$	$< 10^{-6}$
提案手法	STATS のみ	0.059977	$< 10^{-6}$	$< 10^{-6}$
	CTC のみ	0.008228	$< 10^{-6}$	$< 10^{-6}$
	STATS+CTC	0.005173	$< 10^{-6}$	$< 10^{-6}$

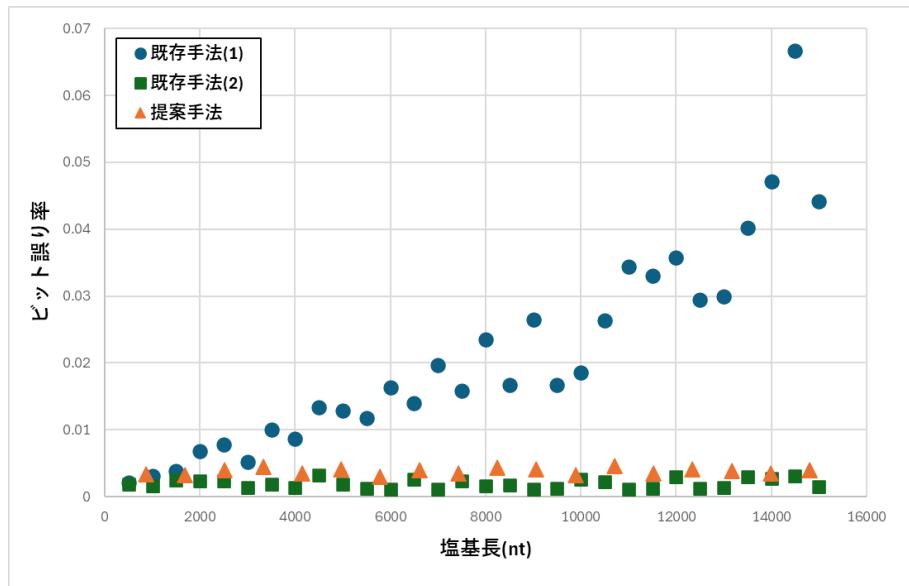


図 4.2: 塩基長とビット誤り率の関係

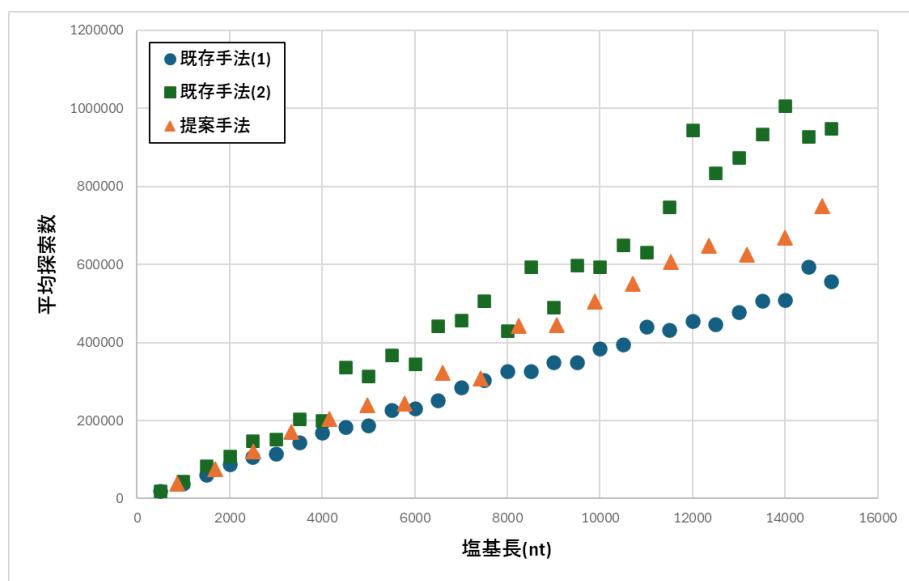


図 4.3: 塩基長と DNA 鎖 1 本あたりの平均ヒープ探索数の関係

表 A.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 A.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 A.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 A.3: 得られた各期待値

パラメータ	値
μ_{STATS_ok}	1.38
μ_{STATS_sub}	7.52
μ_{STATS_ins}	8.01
μ_{STATS_del}	7.17
μ_{CTC_ok}	0.111
μ_{CTC_sub}	6.85
μ_{CTC_ins}	2.78

表 A.4: 符号化率ごとに最適化したペナルティ

符号化率	手法	P_{sub}	P_{ins}	P_{del}
0.75	STATS のみ *	0.630	0.650	0.687
	CTC のみ **	0.837	0.912	0.765
	STATS+CTC***	0.875	0.911	0.778
0.60	STATS のみ *	0.883	0.965	0.995
	CTC のみ **	0.856	1.06	0.890
	STATS+CTC***	1.08	1.17	1.02
0.50	STATS のみ *	0.988	1.17	1.25
	CTC のみ **	0.953	1.11	1.04
	STATS+CTC***	1.15	1.23	1.19
0.33	STATS のみ *	1.07	1.25	1.39
	CTC のみ **	1.06	1.22	1.21
	STATS+CTC***	1.29	1.35	1.34

*STATS のみ: 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使用

**CTC のみ: CTC 出力に基づく補正項のみを使用

***STATS+CTC: ナノポアシーケンサーの読み出し特性および CTC 出力の両方にに基づく補正項を使用

表 A.5: 符号化率ごとに最適化した補正項の重み付けパラメータ

符号化率	手法	α_{ok}	α_{sub}	α_{del}	α_{ins}	β_{ok}	β_{sub}	β_{ins}
0.75	STATS のみ	0.0605	0.116	0.0929	0.0952	0.00	0.00	0.00
	CTC のみ	0.00	0.00	0.00	0.00	-0.192	0.211	0.347
	STATS+CTC	0.00668	0.130	-0.0105	0.544	-0.150	0.181	0.301
0.60	STATS のみ	0.00513	0.256	0.200	0.0969	0.00	0.00	0.00
	CTC のみ	0.00	0.00	0.00	0.00	-0.223	0.190	0.419
	STATS+CTC	0.110	0.243	0.236	0.219	-0.115	0.184	0.330
0.50	STATS のみ	0.0548	0.294	0.0800	0.0871	0.00	0.00	0.00
	CTC のみ	0.00	0.00	0.00	0.00	-0.108	0.193	0.411
	STATS+CTC	0.131	0.292	0.234	0.176	-0.136	0.188	0.467
0.33	STATS のみ	-0.00773	0.282	0.167	0.246	0.00	0.00	0.00
	CTC のみ	0.00	0.00	0.00	0.00	0.0272	0.145	0.422
	STATS+CTC	0.0260	0.328	0.177	0.230	0.0417	0.134	0.460