2016

Probabilistic insertion, deletion and substitution error correction using Markov inference in next generation sequencing reads

Vahid Noroozi
Iowa State University

Probabilistic insertion, deletion and substitution error correction using Markov inference in next generation sequencing reads

by

Vahid Noroozi

A thesis submitted to the graduate faculty
in partial fulfillment of the requirements for the degree of
MASTER OF SCIENCE

Major: Electrical Engineering

Program of Study Committee:
Aditya Ramamoorthy, Major Professor
Karin Dorman
Namrata Vaswani

Iowa State University
Ames, Iowa
2016

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to take this opportunity to express my thanks to those who helped me with various aspects of conducting research and the writing of this thesis. First and foremost, Dr. Aditya Ramamoorthy for his guidance, patience and support throughout this research and the writing of this thesis. I would also like to thank Dr. Karin Dorman and my colleague Xin Yin for their efforts and invaluable contributions to this work. I would additionally like to thank Dr. Namrata Vaswani for her guidance throughout the initial stages of my graduate studies.

ABSTRACT

Error correction of noisy reads obtained from high-throughput DNA sequencers is an important problem since read quality significantly affects downstream analyses such as detection of genetic variation and the complexity and success of sequence assembly. Most of the current error correction algorithms are only capable of recovering substitution errors. In this work, Pindel, an algorithm that simultaneously corrects insertion, deletion and substitution errors in reads from next generation DNA sequencing platforms is presented. Pindel corrects insertion, deletion and substitution errors by modelling the sequencer output as emissions of an appropriately defined Hidden Markov Model (HMM). Reads are corrected to the corresponding maximum likelihood paths using an appropriately modified Viterbi algorithm. When compared with Karect and Fiona, the top two current algorithms capable of correcting insertion, deletion and substitution errors, Pindel exhibits superior accuracy across a range of datasets.

CHAPTER 1. OVERVIEW

Next generation DNA sequencing is a rapidly evolving technology that enables the low cost and fast determination of the genomic sequences of organisms ranging from viruses to humans. It is widely used to understand microbial populations and may facilitate technologies such as personalized medicine. However, current sequencing technologies suffer from one major issue: they produce relatively short reads with a significant fraction of errors [Jünemann et al. (2013); Shendure and Ji (2008)]. The correction of sequencing errors is a crucial task in bioinformatics since the presence of errors significantly interferes with many downstream analyses, including detection of ultra-rare mutations [Schmitt et al. (2012)], genetic heterogeneity detection [Lou et al. (2013)], and *de novo* genome assembly [Nagarajan and Pop (2013); Gordon and Green (2013); Schulz et al. (2012)].

DNA sequencing operates by randomly breaking many copies of a genome, whose length may range from thousands to billions of nucleotides, into fragments which are roughly a few hundred nucleotides long. The starting position of a given fragment is random. Thus, while the sequencer produces a large number of reads such that each position is read multiple times, there is no alignment information to indicate which reads cover a given nucleotide. The dominant error pattern in these reads varies across the different sequencing platforms. The Illumina platform is known to primarily exhibit substitution errors [Jünemann et al. (2013); Shendure and Ji (2008)], while platforms such as 454 pyrosequencing, Ion Torrent PGM and PacBio real-time sequencing exhibit a large number of insertion, deletion (collectively referred to as indels) and substitution errors [Jünemann et al. (2013); Bragg et al. (2013); Yang et al. (2013); Loman et al. (2012); Laehnemann et al. (2015)].

It should be emphasized that error correction is different from basecalling[Merriman et al. (2012); Kao et al. (2009)] in which the decisions are made only by examining the sequenced

nucleotides in a given read. Significant research work addresses the design of good basecallers [Bragg et al. (2013); Merriman et al. (2012); Rothberg et al. (2011)]. In contrast, error correction aims at correcting the reads that are produced by the basecaller and critically relies on processing the information from all the reads simultaneously. While it is conceivable that having access to the raw sequencer output can improve error correction, in this work the basecalls and corresponding quality scores are modeled instead. This approach provides greater flexibility in modelling reads originating from different platforms. In addition, the raw sequencer output is often unavailable or discarded (to conserve space). Therefore, in this work the sequencer and basecaller pair is called "the sequencer".

The problem of error correction has received significant attention in recent years (see [Yang et al. (2013); Laehnemann et al. (2015)]). However, most of the proposed algorithms only deal with substitution errors. There are only a few methods such as Karect [Allam et al. (2015)], Fiona [Schulz et al. (2014)], Coral [Salmela and Schröder (2011)] and HSHREC [Salmela (2010)], that are capable of correcting indels and substitutions. On the other hand, the Illumina platform *does* produce indels [Schirmer et al. (2015)], and some popular platforms, *e.g.* Ion Torrent, produce reads with significant numbers of indels as well as substitutions.

Finally, third generation sequencing technologies that promise long reads extending to thousands of nucleotides also have the highest error rates, including substantial indel rates [Wang et al. (2015); Ip et al. (2015)]. In summary, the development of high performance error correction algorithms that deal with indels and substitutions is an important problem [Laehnemann et al. (2015)].

Main Contributions of this work

In this work Pindel, a flexible, probabilistic method that addresses the problem of correcting insertion, deletion and substitution errors in noisy reads is presented. The approach in this work builds on the basic framework proposed in [Yin et al. (2013b)] for error correction only in the presence of substitution errors. These are three main technical contributions of this work.

1. An appropriate Hidden Markov Model (HMM) for the emission of reads from the sequencer is defined. The problem of correcting substitution errors in a probabilistic set-

ting is already challenging, as standard implementations of the HMM result in model estimation problems and poor error correction performance [Yin et al. (2013b)]. Hence, the consideration of insertion and deletion errors adds significant challenges. In this work indel errors are modeled by expanding the state space, which was previously restricted to the $k$mer-spectrum [Yin et al. (2013b,a)]. The specification of appropriate state transitions allows the insertion and deletion errors to be effectively modeled.

2. The results on real, publicly available Ion Torrent sequencing datasets demonstrate 7.5% average improvement (ranging from 1% to 26% on different datasets) in error correction rates (gains) over Karect [Allam et al. (2015)], the current state of the art error correction technique, and 32.7% average improvement over Fiona [Schulz et al. (2014)]. However, it needs to be emphasized that the techniques presented here are not specific to Ion Torrent and are applicable to any sequencing technology.

In the next section, a review of the relevant background and related work is presented followed by detailing the proposed HMM in section 3.1. Parameter estimation, modeling choices and the final error correction step are discussed in section 3.2. Subsequently the proposed method is compared with Karect and Fiona in section 4.1. Some details about the model, run parameters, and parameter estimation are discussed in the Appendix.

CHAPTER 2. REVIEW OF LITERATURE

DNA consists of two directed strands bound in an antiparallel duplex. Let $\mathcal{G}$ denote the first strand of the DNA, which is a quaternary sequence of length $|\mathcal{G}|$ over the alphabet $\mathcal{B}_1 = \{A, C, G, T\}$. The sequencer produces "reads" from short fragments of the genome, either moving along strand $\mathcal{G}$ or in the reverse direction on its reverse complement, $\overline{\mathcal{G}}$. Thus, the i-th read is an estimate of a substring $\mathbf{s}_i$ that starts at a random position in either $\mathcal{G}$ or $\overline{\mathcal{G}}$, but the read contains neither information about the source strand nor starting position.

The full dataset, denoted by $\mathcal{R}$ where $|\mathcal{R}| = r$, is the entire set of sequence reads and quality scores produced by a sequencing experiment. The combined length of all reads is $L = \sum_{i=1}^{r} l_i$. A read is the sequencer's best estimate of a contiguous set of nucleotide bases on one strand but may contain insertion, deletion or substitution errors relative to the true genome sequence. The i-th read is the tuple $(\mathbf{x_i}, \mathbf{y_i})$, in which $\mathbf{x}_i$ is the sequence of bases, called "base calls" in $\mathcal{B}_1$, and $\mathbf{y}_i$ is its corresponding sequence of quality scores that indicate the sequencer's confidence in the basecall. Both $\mathbf{x}_i$ and $\mathbf{y}_i$ are of length l_i.

Error correction of reads is only possible because each base in the genome is typically covered by multiple reads. If the starting positions of the reads are uniformly distributed along the genome, each base should be covered on average by $L/(2|\mathcal{G}|)$ reads. This is referred to as the *coverage level* of the sequencing experiment. High coverage provides good redundancy for error correction, but the lack of positional information makes the problem challenging.

There are three main approaches for correcting errors in the sequenced reads. Methods such as Karect [Allam et al. (2015)], [Coral Salmela and Schröder (2011)] and ECHO [Kao et al. (2011)] use $k$mers, which are substrings of length k, as "seeds" to form multiple sequence alignment (MSA) on overlapping reads. Subsequently by creating a consensus, error correction is performed.

A large class of methods [Yang et al. (2010); Liu et al. (2013); Kelley et al. (2010); Medvedev et al. (2011); Heo et al. (2014)] are based on extracting the $k$mer-spectrum of the observed read set, *i.e.* the set of all observed $k$mers in all the reads. If k is chosen large enough, most of the true $k$mers appear in unique locations in the genome (except the ones in repeated portions of the genome); this is usually referred to as the $k$mer-uniqueness assumption. Under this assumption, $k$mers with small observed counts can be identified as errors and corrections can be attempted. The methods differ significantly in how they identify erroneous $k$mers and how exactly they make decisions about the corresponding correction. It needs to be emphasized at this point that all these algorithms [Yang et al. (2010); Liu et al. (2013); Kelley et al. (2010); Medvedev et al. (2011); Heo et al. (2014)] only deal with substitution errors and cannot handle indels. The proposed algorithm here, Pindel, also belongs to the class of $k$mer-based methods. However, as discussed in §3.1 and §3.2.2 there are several novel aspects of the presented approach that allow for superior performance in the presence of indels.

Finally, suffix tree/array based methods [Schröder et al. (2009); Ilie et al. (2011); Schulz et al. (2014)] can be considered as generalized variable length $k$mer based methods. Suffix trees from the read set, help to identify and correct erroneous $k$mers using tree nodes with low weights for a range of $k$mer length.

CHAPTER 3. METHODS AND PROCEDURES

3.1 Hidden Markov Modeling of Error Correction

A short overview of how Pindel, the HMM based error correction algorithm, corrects different types of errors is provided in the following. Let $\mathcal{S}$ be the set of true sequences that generates the read set, $\mathcal{R}$. In particular, $\mathbf{s}_i \in \mathcal{S}$ generates the i-th read $(\mathbf{x_i}, \mathbf{y_i})$, where $\mathbf{x}_i$ and $\mathbf{y}_i$ denote the basecalls and the quality scores. Let $\mathbf{s}_i[j]$ denote the j-th nucleotide of $\mathbf{s}_i$ and $\mathbf{s}_i[j...t]$ denote the substring of $\mathbf{s}_i$ from position j to position t (inclusive). Similarly substrings, $\mathbf{x}_i[j...t]$ or $\mathbf{y}_i[j...t]$, can be extracted from the observed read or quality scores, though because of insertions and deletions, the exact correspondence between positions in the hidden $\mathbf{s}_i$ and the observed $\mathbf{x}_i$ or $\mathbf{y}_i$ is not known. Substrings of length k or $k+1$ bases are called $k$mers and $(k+1)$mers, respectively. Let $\mathbf{x}_{i,t} = \mathbf{x}_i[t-k+1...t]$ be the t-th observed $k$mer of read $\mathbf{x}_i$; therefore $\mathbf{x}_{i,t}[j]$ will be the j-th nucleotide of the t-th $k$mer, $\mathbf{x}_{i,t}$.

Each read is modeled as an independent emission from the HMM. It is assumed that the underlying Markov chain starts at state $\mathbf{s}_{i,k}$ with probability governed by an initial state distribution. Given the initial state, the sequencer emits k base calls, $\mathbf{x}_{i,k}$, as well as the quality scores $\mathbf{y}_{i,k}$. Then, the sequencer transitions to a new state, $\mathbf{s}_{i,k+1}$ which is either a $k$mer or a $(k+1)$mer. When the sequencer is at state $\mathbf{s}_{i,t}$ for $t > k$, it emits $(\mathbf{x}_i[t], \mathbf{y}_i[t])$. The hidden state $\mathbf{s}_{i,t}$ is always responsible for the emission of t-th observed nucleotide $\mathbf{x}_i[t]$, but may not correspond to position t of $\mathbf{s}_i$ because of indels. Transitions between a $k$mer and a $(k+1)$mer model deletion errors, transitions from a $k$mer to itself model insertion errors and the emission distribution models the substitution errors. The following example demonstrates the modeling of the different error types.

Example 3.1.1. Fig. 3.1 shows a situation where $k = 4$ and the true underlying sequence is

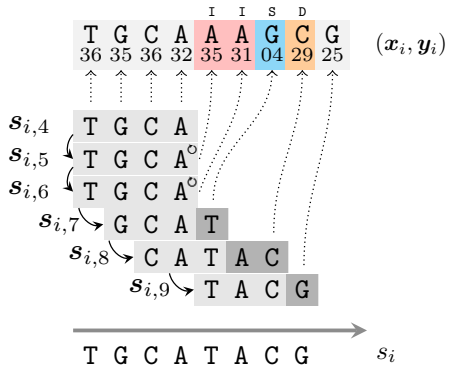

Figure 3.1: Simple example demonstrating how different types of errors are modeled, where the true underlying sequence is TGC**ATA**CG, and the erroneous read is TGCA**AAG**CG.

TGCATACG. The sequencer transitions through the states $\mathbf{s}_{i,4}, \mathbf{s}_{i,5}, \ldots, \mathbf{s}_{i,9}$. There are two insertion errors at positions five and six, leading to $\mathbf{s}_{i,4} = \mathbf{s}_{i,5} = \mathbf{s}_{i,6} = \mathbf{TGCA}$. The next transition to $\mathbf{s}_{i,7} = \mathbf{GCAT}$ produces a substitution error at the seventh position when base **T** is miscalled as **G**. Next, $\mathbf{s}_{i,7}$ transitions to the $(k+1)$mer $\mathbf{s}_{i,8}$ and consequently the true base $\mathbf{s}_{i[6]} =$**A** is deleted. Finally, the sequencer transitions to $\mathbf{s}_{i,9}$ (a $k$mer) and emits base **G** without any error.

The Markov model is well motivated because in a finite genome and with large $k$mer size, most $k$mers are unique, leading to strong local dependence between $k$mers. Since the genome is not observed directly during sequencing, the Markov states are latent variables [Rabiner (1989)]. The strategy in this work is to first fit the parameters of the HMM based on the observed reads. Then the maximum likelihood state sequence $\hat{\mathbf{s}}_i$ that best explains each observed read $(\mathbf{x_i}, \mathbf{y_i})$ is determined and declared as the corrected read. In particular, in the example in Fig. 3.1, the goal is to recover the maximum likelihood state sequence $\hat{\mathbf{s}}_{i,4}, \ldots, \hat{\mathbf{s}}_{i,9}$.

There are critical modeling choices that make the presented approach work, and all revolve around the central notion of a $k$mer. Ignoring self transitions (insertions) and $k$mer to $(k+1)$mer transitions (deletions), the remaining *genomic* transitions model the observed $k$mer to $k$mer transitions in the genome. The genome $\mathcal{G}$ is finite, and each error-free $k$mer occurs 0 or some finite number of times in $\mathcal{G}$. If k is small, the genomic transition probabilities of the HMM reflect the signal from multiple genomic locations and are not useful for separating genomic variation from error. Thus, k needs to be large enough to guarantee $k$mer-uniqueness for most $k$mers, but the k required to guarantee *all* $k$mers are unique will typically be larger than

the read length or result in insufficient coverage to distinguish error and true transitions. In this work, k is chosen such that it balances the $k$mer-uniqueness requirement while retaining sufficient coverage (see §3.2.3). In addition, the following three ideas which result in a solution with excellent performance are used.

- Only $k$mers and $(k+1)$mers that have been observed in the reads are included in the state space.
- The resulting state space includes many erroneous $k$mers, and hence the model is overparameterized in genomic transition probabilities, so a ℓ_0-like penalty is imposed on these parameters to enforce the belief that most error-free $k$mers are unique.
- Finally, the coverage is effectively doubled by combining information from both the forward and reverse complement strands of the genome.

In the remainder of this section, the components of the HMM are described in detail.

3.1.1 State Space

The state space of HMM in this work, denoted by $\mathcal{K}$, consists of both $k$mers and $(k+1)$mers. Let us define $\mathcal{K}_1$ as the set of all observed $k$mers in $\mathcal{R}$, as well as their reverse complements. Similarly, define $\mathcal{K}_2$ as the set of observed $(k+1)$mers in $\mathcal{R}$, plus their reverse complements. Each $(k+1)$mer $\boldsymbol{\omega} \in \mathcal{K}_2$ is a deletion state, where the penultimate nucleotide, $\boldsymbol{\omega}_{[k]}$, is deleted during sequencing. To model the insertion errors, a specialized insertion copy of $\boldsymbol{\omega}$, denoted by $\boldsymbol{\omega}^{\circlearrowleft}$ is introduced. By defining $\mathcal{K}_1^{\circlearrowleft} = \{\boldsymbol{\omega}^{\circlearrowleft} : \boldsymbol{\omega} \in \mathcal{K}_1\}$, the final state space is $\mathcal{K} = \mathcal{K}_1 \cup \mathcal{K}_2 \cup \mathcal{K}_1^{\circlearrowleft}$.

For any k, the underlying true genome $\mathcal{G}$ has at most $\min\{2(|\mathcal{G}|-k+1), 4^k\}$ $k$mers. Restricting $\mathcal{K}_1$ and $\mathcal{K}_2$ to the observed $k$mers and $(k+1)$mers in $\mathcal{R}$ guarantees that $|\mathcal{K}_1 \cup \mathcal{K}_2| \ll 4^k + 4^{k+1}$. By including the reverse complements of the observed $k/(k+1)$-mers, the risk of excluding valid oligomers in $\mathcal{G}$ is reduced, which is a growing possibility when $\mathcal{G}$ is sequenced with low or non-uniform coverage.

The following notation is used in the subsequent discussion. Let $\mathcal{B}_1 = \{A, C, G, T\}$ (single

bases), $\mathcal{B}_2 = \{AA, \ldots, TT\}$ (all pairs of bases) and $\{\circlearrowleft\}$ (the self transition), then

$$\boldsymbol{\omega} \blacktriangleright \beta = \begin{cases} \boldsymbol{\omega}_{[2+\varepsilon(\boldsymbol{\omega})\ldots k+\varepsilon(\boldsymbol{\omega})]} \oplus \beta & \beta \in \mathcal{B}_1 \cup \mathcal{B}_2 \\ \boldsymbol{\omega}^{\circlearrowleft} & \boldsymbol{\omega} \in \mathcal{K}_1, \beta = \circlearrowleft \end{cases} \quad \text{and} \quad \boldsymbol{\omega}^{\circlearrowleft} \blacktriangleright \beta = \begin{cases} \boldsymbol{\omega}_{[2\ldots k]} \oplus \beta & \beta \in \mathcal{B}_1 \cup \mathcal{B}_2 \\ \boldsymbol{\omega}^{\circlearrowleft} & \beta = \circlearrowleft, \end{cases}$$

where $\oplus$ is the string concatenation operator and $\varepsilon(\boldsymbol{\omega}) = \mathbb{1}\{\boldsymbol{\omega} \in \mathcal{K}_2\}$ serves as a $(k+1)$mer indicator function. Thus, if $\beta \in \mathcal{B}_1$, $\boldsymbol{\omega} \blacktriangleright \beta$ is a $k$mer, and if $\beta \in \mathcal{B}_2$, $\boldsymbol{\omega} \blacktriangleright \beta$ is a $(k+1)$mer.

3.1.2 Emission Distribution

Let $D(\boldsymbol{\omega}_1, \boldsymbol{\omega}_2)$ denote the edit distance between states $\boldsymbol{\omega}_1, \boldsymbol{\omega}_2 \in \mathcal{K}$ with equal costs of one for insertions, deletions and substitutions. This distance is used to limit the computational complexity of the proposed algorithm. If $\mathbf{s}_{i,t}$ is the true state (either a $k$mer or a $(k+1)$mer) emitting the t-th observed base of the i-th read, $\mathbf{x}_{i,t-1}$ is the $(t-1)$-th observed $k$mer, and $\beta \in \mathcal{B}_1$, then for $t > k$, the emission distribution is

$$f(\mathbf{x}_{i,t[k]} = \beta, \mathbf{y}_{i,t[k]} \mid \mathbf{s}_{i,t}, \mathbf{x}_{i,t-1}) = \underbrace{\varrho(\mathbf{y}_{i,t[k]} \mid \mathbf{s}_{i,t}, \mathbf{x}_{i,t})}_{\text{quality score model}} \underbrace{g(\mathbf{x}_{i,t[k]} = \beta \mid \mathbf{s}_{i,t}, \mathbf{x}_{i,t-1})}_{\text{base emission model}},$$

in which the base emission model is given by

$$g(\mathbf{x}_{i,t[k]} = \beta \mid \mathbf{s}_{i,t}, \mathbf{x}_{i,t-1}) \propto \mathbb{1}\{D(\mathbf{s}_{i,t}, \mathbf{x}_{i,t-1} \blacktriangleright \beta) \leq d\} \cdot g_0(\beta \mid \mathbf{s}_{i,t[k+\varepsilon(\mathbf{s}_{i,t})]}) \tag{3.1}$$

with the constraint $\sum_{\beta \in \mathcal{B}_1} g_0(\beta \mid \mathbf{s}_{i,t[k+\varepsilon(\mathbf{s}_{i,t})]}) = 1$, for $\beta \in \mathcal{B}_1$. The set of possible hidden states $\mathbf{s}_{i,t}$ is limited to those within a maximal edit distance to the last observed $k$mer.

Quality scores potentially inform on the error state of the current base call. The emission of a quality score is modelled by four different probability mass functions (pmfs) supported on the integers $\{q_{\min}, \ldots, q_{\max}\}$, where $q_{\min}$ and $q_{\max}$ are the minimum and maximum quality score values reported by the sequencer. The datasets which were used in the experiments within this work use Phred+33 quality scores, which consist of about 40 distinct quality scores. In particular, the first pmf ϱ_1 models quality scores for bases emitted without error, ϱ_2 models quality scores accompanying substitution errors, ϱ_3 models quality scores for bases emitted after a deletion error, and ϱ_4 is for quality scores of inserted nucleotides. Specifically, the

quality emission distribution is

$$\varrho(\mathbf{y}_{i,t[k]} \mid \mathbf{s}_{i,t}, \mathbf{x}_{i,t}) = \begin{cases} \varrho_1(\mathbf{y}_{i,t[k]}) & \text{if } \mathbf{x}_{i,t[k]} = \mathbf{s}_{i,t[k]}, \mathbf{s}_{i,t} \in \mathcal{K}_1, \varphi(\mathbf{s}_{i,t}) = 0 \\ \varrho_2(\mathbf{y}_{i,t[k]}) & \text{if } \mathbf{x}_{i,t[k]} \neq \mathbf{s}_{i,t[k]}, \mathbf{s}_{i,t} \in \mathcal{K}_1, \varphi(\mathbf{s}_{i,t}) = 0 \\ \varrho_3(\mathbf{y}_{i,t[k]}) & \text{if } \mathbf{s}_{i,t} \in \mathcal{K}_2, \text{ and} \\ \varrho_4(\mathbf{y}_{i,t[k]}) & \text{if } \mathbf{s}_{i,t} \in \mathcal{K}_1, \varphi(\mathbf{s}_{i,t}) = 1, \end{cases}$$

where $\varphi(\mathbf{s}_{i,t}) = \mathbb{1}\{\mathbf{s}_{i,t} \in \mathcal{K}_1^{\circlearrowleft}\}$ is an indicator of insertion copy $k$mers.

The emission of the first k bases and quality scores of a read is handled differently. Here, the fact that error rates tend to be low at the beginning of reads is exploited and no indel errors are assumed to exist among the first k emitted bases and the first k quality scores are not modeled. Furthermore, it is assumed that $\mathbf{s}_{i,k} \in \mathcal{K}_1$ is a $k$mer. Then,

$$f(\mathbf{x}_{i,k} \mid \mathbf{s}_{i,k}) \propto \mathbb{1}\{D_H(\mathbf{s}_{i,k}, \mathbf{x}_{i,k}) \leq d_k\},$$

where $D_H(\cdot, \cdot)$ is the Hamming distance and parameter $d_k < d$. The values of d and d_k determine the computational complexity of the parameter estimation and sequence decoding. Their choices are discussed in §3.2.3.

3.1.3 Transition Probability Distribution

The transition between state $\mathbf{s}_{i,t}$ and adjacent state $\mathbf{s}_{i,t+1}$ is governed by probability distribution $p(\boldsymbol{\omega} \blacktriangleright \beta \mid \boldsymbol{\omega})$ for $\boldsymbol{\omega} \in \mathcal{K}, \beta \in \mathcal{B} = \mathcal{B}_1 \cup \mathcal{B}_2$. Throughout this work the transition from $\boldsymbol{\omega}$ to β is interchangeably refered to as $\boldsymbol{\omega} \to \beta$ or $\boldsymbol{\omega} \to \boldsymbol{\nu}$, where $\boldsymbol{\nu} = \boldsymbol{\omega} \blacktriangleright \beta$.

The transition probability is defined in terms of $k$mer-to-$k$mer transition probabilities $q(\beta \mid \boldsymbol{\omega})$ defined for $\boldsymbol{\omega} \in \mathcal{K}_1 \cup \mathcal{K}_1^{\circlearrowleft}$ and $\beta \in \mathcal{B}_1$ ($\boldsymbol{\omega}^{\circlearrowleft} \in \mathcal{K}_1^{\circlearrowleft}$ shares the same parameters $q(\cdot|\boldsymbol{\omega})$ with its non-insertion copy $\boldsymbol{\omega}$). Nonzero $q(\cdot \mid \cdot)$ represent true transitions in the genome. To induce additional sparseness in transitions and assuming reasonable coverage, all transitions $\boldsymbol{\omega} \to \beta$ are required to be observed at least once. For this purpose, $\mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})$ is defined for all $\boldsymbol{\omega} \in \mathcal{K}_1$, as the subset of $\beta \in \mathcal{B}_1$ such that either $\boldsymbol{\omega} \to \beta$ or $\overline{\boldsymbol{\omega} \blacktriangleright \beta} \to \overline{\boldsymbol{\omega}[1]}$ ($\overline{\boldsymbol{\omega}}$ denotes the reverse complement of $\boldsymbol{\omega}$) is observed in at least one read in $\mathcal{R}$. Then,

$$\sum_{\beta \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})} q(\beta \mid \boldsymbol{\omega}) = 1 \tag{3.2}$$

and $q(\beta \mid \boldsymbol{\omega}) = 0$ for $\beta \notin \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})$. Given $q(\cdot \mid \cdot)$ and defining $\boldsymbol{\omega}_{1+\varepsilon..} = \boldsymbol{\omega}[1+\varepsilon(\boldsymbol{\omega})..k+\varepsilon(\boldsymbol{\omega})]$ as the k-suffix of $\boldsymbol{\omega}$, the transition probability distribution is

$$p(\boldsymbol{\omega} \blacktriangleright \beta \mid \boldsymbol{\omega}) = \begin{cases} \left(1 - p_d - p_i \cdot \mathbb{1}_{\{\boldsymbol{\omega} \in \mathcal{K}_1\}}\right) \cdot q(\boldsymbol{\omega} \blacktriangleright \beta \mid \boldsymbol{\omega}_{1+\varepsilon..}) & \text{if } \beta \in \mathcal{B}_1 \\ p_d \cdot q(\boldsymbol{\omega} \blacktriangleright \beta_{[1]} | \boldsymbol{\omega}) \cdot \; q\left((\boldsymbol{\omega} \blacktriangleright \beta_{[1]}) \blacktriangleright \beta_{[2]} \mid \boldsymbol{\omega}_{1+\varepsilon..} \blacktriangleright \beta_{[1]}\right) & \text{if } \beta \in \mathcal{B}_2 \\ p_i \cdot \mathbb{1}_{\{\boldsymbol{\omega} \in \mathcal{K}_1\}} & \text{if } \beta = \circlearrowleft \end{cases} \tag{3.3}$$

where p_d is the deletion error probability and p_i is the probability of $k$mer self transition (insertion error). While transition probability $p(\cdot \mid \cdot)$ estimates a mix of signal from the genome and indel errors of the sequencer, the $q(\cdot \mid \cdot)$ represent pure, genomic transition probabilities. The properties of these transitions are discussed in what follows.

3.1.4 Modeling dependence between strands.

Both forward $\mathcal{G}$ and reverse strands $\overline{\mathcal{G}}$ of the genome are sequenced. If the strands are sequenced with equal coverage, the probability of observing $\boldsymbol{\omega} \to \boldsymbol{\nu}$ on one strand must equal the probability of observing $\overline{\boldsymbol{\nu}} \to \overline{\boldsymbol{\omega}}$ on the reverse strand. This observation allows us to halve the total number of state transition parameters and initial state distribution parameters in the model.

Let $\pi(\boldsymbol{\omega})$ denote the probability of starting a read in state $\boldsymbol{\omega}$. Here, it was assumed there are no indels in the first k bases of a read, so only states in $\mathcal{K}_1$ have nontrivial initial state probabilities. Under the assumption of equal coverage on both strands the following equations hold

$$\pi(\boldsymbol{\omega}) = \pi(\overline{\boldsymbol{\omega}}) \qquad \text{and} \qquad \pi(\boldsymbol{\omega}) \cdot q_1(\boldsymbol{\nu}_{[k]} \mid \boldsymbol{\omega}) = \pi(\overline{\boldsymbol{\nu}}) \cdot q_2(\overline{\boldsymbol{\omega}}_{[k]} \mid \overline{\boldsymbol{\nu}}) \tag{3.4}$$

for all $\boldsymbol{\omega}, \boldsymbol{\nu} \in \mathcal{K}_1$. Transition parameters are subscripted by 1 for the first strand and 2 for the reverse complement strand. Although the strands of a $k$mer are unknown, these parameter relationships can be beneficial. Let $\widetilde{\boldsymbol{\omega}}$ represent the lexically ordered pair $\boldsymbol{\omega}$ and its reverse complement $\overline{\boldsymbol{\omega}}$ and let $\pi(\widetilde{\boldsymbol{\omega}})$ unambiguously identify the initial state probability $\pi(\boldsymbol{\omega}) = \pi(\overline{\boldsymbol{\omega}})$. In addition, a consistent method is required to label transitions such that when $\boldsymbol{\omega} \to \boldsymbol{\nu}$ is labeled 1, $\overline{\boldsymbol{\nu}} \to \overline{\boldsymbol{\omega}}$ is labeled 2. Then, transitions on strand 2 are functions of the transitions on

strand 1 as

$$q_2(\overline{\boldsymbol{\omega}}_{[k]} \mid \overline{\boldsymbol{\nu}}) = \frac{\pi(\widetilde{\boldsymbol{\omega}}) \cdot q_1(\boldsymbol{\nu}_{[k]} \mid \boldsymbol{\omega})}{\pi(\widetilde{\boldsymbol{\nu}})}. \tag{3.5}$$

A labeling procedure is described in [Yin (2016)] where all outgoing transitions from the same $k$mer share the same label, a crucial choice that leads to a tractable M-step during HMM parameter estimation. Thus, $k$mers rather than transitions are labeled, and $\mathcal{L}(\boldsymbol{\omega})$ can be defined as the label of $\boldsymbol{\omega}$. Specifically, $\mathcal{L}(\boldsymbol{\omega}) = 1$ implies that all transitions $q(\beta \mid \boldsymbol{\omega})$ for $\beta \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})$ carry label 1. Furthermore, the aforementioned labeling algorithm also requires the value of k to be even (see [Yin (2016)]).

There is more than one way to accomplish such a labeling and in fact different labelings will result in different error correction performances. Suppose that $k = 4$ and the true transition (CCAA $\rightarrow$ CAAG) is observed 50 times and the erroneous transition (CCAA $\rightarrow$ CAAC) only once on the forward strand. The corresponding transitions on the reverse complement strand are such that $k$mers GTTG and CTTG only have unique transition into the kmer TTGG. In this case, by assigning $\mathcal{L}(\text{CCAA}) = 1$, the model parameters will be $q_1(\text{G} \mid \text{CCAA})$ and $q_1(\text{C} \mid \text{CCAA})$ and the second (erroneous) transition should be driven to zero in the penalized estimation procedure. On the other hand, by setting $\mathcal{L}(\text{CCAA}) = 2$, then $\mathcal{L}(\text{CTTG}) = \mathcal{L}(\text{GTTG}) = 1$ and the erroneous $k$mer GTTG on the reverse strand transitions only to $k$mer TTGG. The penalty fails to drive $q_1(\text{G} \mid \text{GGTG})$ to zero since transition probabilities $q_1(\cdot \mid \text{GGTG})$ must sum to one. This example demonstrates that label assignments have the potential to impact error correction. This issue is discussed further in [Yin (2016)] and the labeling algorithm is explained.

3.2 Parameter Estimation, Viterbi Decoding and Model Choices

Pindel was implemented in C/C++. The major components of Pindel, including the construction of the k spectrum, the EM algorithm, and the Viterbi decoding are parallelized using OpenMP for shared memory computers. Some details of the major steps are discussed in the following.

3.2.1 HMM parameter estimation

For large k, most $k$mers $\boldsymbol{\omega}$ become unique in the genome, and the genomic transition distributions $q_1(\cdot \mid \boldsymbol{\omega})$ should become degenerate, *i.e.*, have only one non-zero transition probability with value 1. However, since there is no independent method to distinguish true transitions from error transitions, the model is formulated with many non-zero transitions that do not exist in the genome. Fortunately, if errors are rare and k is large, most error transitions will be observed relatively fewer times than true transitions. This signal from the data can be capitalized to eliminate erroneous transitions and sparsify $q_1(\cdot \mid \cdot)$.

Subsuming all emission and transition parameters in vector $\boldsymbol{\theta}$, one would normally maximize the log likelihood, $l(\boldsymbol{\theta} \mid \mathcal{R})$, of observing the read set $\mathcal{R}$ given $\boldsymbol{\theta}$ to produce parameter estimates $\hat{\boldsymbol{\theta}}$. To enforce the belief that most $k$mers are unique in the genome, an ℓ_0-like penalty is incorporated on the genomic transition probabilities. Subsequently, a penalized log-likelihood function $l(\boldsymbol{\theta} \mid \mathcal{R}) - \rho\mathcal{J}(\boldsymbol{\theta})$ is maximized over $\boldsymbol{\theta}$, where the term $\mathcal{J}(\boldsymbol{\theta})$ is

$$\mathcal{J}(\boldsymbol{\theta}) = \sum_{\substack{\boldsymbol{\omega}:\, \boldsymbol{\omega}\in\mathcal{K}_1 \\ \mathcal{L}(\boldsymbol{\omega})=1}} \sum_{\beta\in\mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})} \log[1 + q_1(\boldsymbol{\omega} \blacktriangleright \beta|\boldsymbol{\omega})/\gamma], \tag{3.6}$$

and constants γ and ρ are chosen to achieve a desired level of sparsity in the transition probabilities [Alexander and Lange (2011)]. Briefly, when γ is tiny, the parameter ρ defines a threshold such that when the expected number of transitions $\boldsymbol{\omega} \to \beta$ exceeds ρ, then parameter $q_1(\beta \mid \boldsymbol{\omega})$ is approximately set equal to the corresponding maximum likelihood estimator. On the other hand, if the expected number of transitions is less than ρ, then $q_1(\beta \mid \boldsymbol{\omega})$ is pushed to 0 (see [Yin (2016)]).

3.2.2 Viterbi Decoding and Error Correction

Once the parameters $\hat{\boldsymbol{\theta}}$ are estimated from the data, error correction is achieved by running the Viterbi algorithm. It was noticed that a straightforward application of the usual Viterbi algorithm fails to identify many insertions in the data sets used in this work. Therefore a modified Viterbi is described in this section.

Ion Torrent sequencers occasionally produce reads that contain significantly long bursts of consecutive insertion errors. For instance, in datasets $\mathcal{D}_3, \mathcal{D}_4, \mathcal{D}_5, \mathcal{D}_6$ from Table 4.1, about

9% of insertion errors appear as insertion blocks over 10 nucleotides long. Modeling such long insertions is computationally expensive because it strikingly increases the number of plausible hidden pathways. The distance constraint in Eq. (3.1), which insures computational tractability, specifically disallows such long insertions. Fortunately, while long insertions account for many individual insertion errors, they are infrequent, so their exclusion during parameter estimation has negligible effect on parameter estimation. For instance, only about 0.26% of reads in $\mathcal{D}_2$, have consecutive insertion errors longer than 10. Yet recovering these errors is essential for performance, so a modified Viterbi algorithm that can correct long insertions is proposed. In particular, the edit distance constraint imposed in Eq. (3.1) is modified during the Viterbi decoding as follows.

$$f(\mathbf{x}_{i,t[k]} = \beta, \mathbf{y}_{i,t[k]} \mid \mathbf{s}_{i,t}, \mathbf{x}_{i,t-1}) = \\ \mathbb{1}\Big\{\{\varphi(\mathbf{s}_{i,t})\} \cup \{\mathbf{s}_{i,t[k]} = \beta \text{ and } \beta \in \mathcal{B}_1\} \cup \{D(\mathbf{s}_{i,t}, \mathbf{x}_{i,t-1} \blacktriangleright \beta) \leq d\}\Big\} \times \\ g_0(\beta \mid \mathbf{s}_{i,t[k+\varepsilon(\mathbf{s}_{i,t})]})\varrho(\mathbf{y}_{i,t[k]} \mid \mathbf{s}_{i,t}, \mathbf{x}_{i,t}). \tag{3.7}$$

The modified indicator function takes the value 1 if $\mathbf{s}_{i,t}$ is an insertion copy $k$mer or if there is a match between $\mathbf{s}_{i,t[k]}$ and the emitted base β or if the original constraint in Eq. (3.1) is met. Now, during Viterbi decoding, state sequences with repeated self-transitions are allowed, regardless of the edit distance. Once such a long insertion is decoded, it is likely that $D(\mathbf{s}_{i,t}, \mathbf{x}_{i,t-1} \blacktriangleright \beta) \geq d$. If this happens, to return to a valid pathway it is required that the $k$mer $\mathbf{s}_{i,t}$ to emit the observed base β; otherwise, there may not exist a valid pathway to explain the read.

To address the limited capability of error-correction within the first $k$mer, which is assumed to contain no indel errors, the Viterbi algorithm is run in both directions to estimate the "true sequence" $\mathbf{s}_i$ given the read pair $(\mathbf{x_i}, \mathbf{y_i})$. In the first run, $\hat{\mathbf{s}}'_i$ is decoded given the read pair $(\mathbf{x_i}, \mathbf{y_i})$. Next, it runs on read pair $\left(\overline{\hat{\mathbf{s}}'_i}, \text{rev}(\mathbf{y_i})\right)$, where $\overline{\hat{\mathbf{s}}'_i}$ is the reverse complement of the first estimate of the true sequence and $\text{rev}(\mathbf{y_i})$ are the quality scores in reverse order. The decoded sequence after the second decoding, $\overline{\hat{\mathbf{s}}_i}$, produces the final estimated sequence of true states $\hat{\mathbf{s}}_i$.

3.2.3 Parameter Choices

The most important parameter to choose for Pindel is the $k$mer length, k. As discussed earlier, the choice of k should encourage $k$mer uniqueness while retaining sufficient $k$mer coverage.

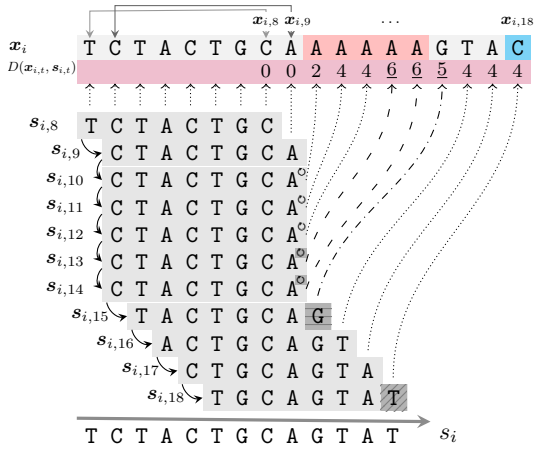

Figure 3.2: Example with $k = 8, d = 4$ where there are insertion errors at positions 10 - 14, but no errors prior to position 10. A possible state sequence encountered in the decoding is shown. States $\mathbf{s}_{i,10} - \mathbf{s}_{i,12}$ are self-transitions with $D(\mathbf{s}_{i,12}, \mathbf{x}_{i,12}) = 4$. However, $D(\mathbf{s}_{i,13}, \mathbf{x}_{i,13}) = 6$ and such a state would be disallowed under Eq. (3.1). However, it is allowed under the modified Eq. (3.7). Similarly, after decoding the insertion burst, $D(\mathbf{s}_{i,14}, \mathbf{x}_{i,14}) = 6$, the next few transitions have to be such that $\mathbf{s}_{i,t}, 15 \leq t \leq 17$ emits the observed base. A substitution error is allowed at $t = 18$ as $D(\mathbf{s}_{i,18}, \mathbf{x}_{i,18}) = 4$.

Another constraint on k comes from the fact that the labeling algorithm that marks the free and dependent transition parameters requires k to be even. Finally, k is limited to a maximum value of $k = 30$ on 64-bit machines. To satisfy the uniqueness assumption, k is chosen according to the heuristic $\frac{|2\mathcal{G}|}{4^k} \approx 10^{-5}$ inspired by [Kelley et al. (2010); Heo et al. (2014)].

The next set of parameters to choose are ρ and γ. Specifically, in the EM algorithm if the expected count of transition $\boldsymbol{\omega} \to \boldsymbol{\nu}$ is above ρ, then it will be retained, but if it is below ρ, the penalty function $\mathcal{J}(\boldsymbol{\theta})$ will likely drive it zero. The parameter γ determines the severity of the penalty, *i.e.*, how close it is to a ℓ_0-penalty (see [Yin (2016)]). $\gamma = 1 \times 10^{-20}$ was used for the experiments in this work. For each dataset, ρ was chosen such that it equals the first valley in the $k$mer counts histogram [Liu et al. (2013)].

The parameter d in Eq. (3.1) is chosen based on the computational complexity the proposed algorithm can handle. It was set to $d = 10$ for the experiments presented in this work, *i.e.*, half of the $k$mer length and $d_k = 2$ was used. The EM is terminated when the relative change in penalized log likelihood is less than 10^{-4}. The parameters used for initializing the EM algorithm are discussed in the Appendix (§A2.3).

CHAPTER 4. EXPERIMENTAL RESULTS

4.1 Experimental Results

The proposed method was evaluated on six Ion Torrent sequencing datasets listed in Table 4.1. For all six datasets, the reference genome is known a priori, allowing us to directly compare the performance of the different algorithms. The "ground truth" errors are determined by aligning the reads to the reference genome using the `bwa mem` algorithm provided by the BWA aligner (v0.7.12) [Li and Durbin (2010)]. The `bwa mem` algorithm, with its default parameters, tends to clip the low-quality ends of the reads, resulting in an underestimation of the total number of errors.

Therefore, to have an accurate comparison, especially in reads with high error rates, this clipping behavior was suppressed by specifying a large clipping penalty, with option `-L 100,100`, and all other settings were left at default. All reads with a unique match to the reference genome (total length L_a) were retained and the number of true errors (e) was tallied as the mismatches between the selected reads and the reference sequence; adjacent insertions and deletions in the reads were counted as separate errors. The respective error rates, e/L_a, for $\mathcal{D}_1 \ldots \mathcal{D}_6$, are available in Table 4.1.

Using the rules discussed in §3.2.3, $k = 20$, $\rho = 1$ were used for $\mathcal{D}_1$ and $\mathcal{D}_6$, $\rho = 2$ for $\mathcal{D}_4$ and $\mathcal{D}_5$ and $\rho = 4$ for $\mathcal{D}_2$ and $\mathcal{D}_3$. These values of ρ are appropriate considering the corresponding coverage levels (Table 4.1).

The performance of the proposed algorithm, Pindel, is compared to Karect, the current state of the art error correction algorithm, and Fiona which is the next top-performer among algorithms able to correct indels [Allam et al. (2015)], on these six datasets. For Karect, its latest version on GitHub was used (commit `ba3ad54`). Karect was run with its default

Table 4.1: The Ion Torrent sequencing datasets used in the performance analysis

Dataset	Coverage	Error rate	Read Length (Average)	$\lvert\mathcal{R}\rvert$	Source
$\mathcal{D}_1$	7.68×	1.48%	16 - 107 (92)	390 976	ERR039477*
$\mathcal{D}_2$	33.9×	0.94%	12 - 636 (324)	494 921	B22-730†
$\mathcal{D}_3$	30×	0.95%	25 - 629 (367)	385 452	
$\mathcal{D}_4$	10×	0.95%	25 - 588 (366)	128 484	Ion 520 Chip
$\mathcal{D}_5$	10×	0.95%	25 - 501 (367)	128 484	*E. coli* 400bp Run‡
$\mathcal{D}_6$	5×	0.95%	25 - 509 (366)	64 242	

For all datasets, the reference genome is ***E. coli* DH10B**, which is of length **4 686 137** nucleotides.
*: accession number on Sequence Read Archive (http://www.ncbi.nlm.nih.gov/sra).
†: available from the Ion Torrent website (http://ioncommunity.lifetechnologies.com/welcome).
‡: $\mathcal{D}_3 - \mathcal{D}_6$ are randomly subsampled versions of a high-coverage 584.1× dataset, available from the Ion Torrent website.

parameters, except setting `-matchtype=edit, -celltype=haploid`, which are appropriate for the six datasets. To run Fiona (v0.2), the genome length and error rates in Table 4.1 were provided using the `-g` and `-e` options, while leaving other settings at default.

Table 4.2: Performance comparison of Pindel, Karect and Fiona on Ion Torrent datasets

Dataset	Method	Gain	Gain (I)	Gain (D)	Gain (S)	Error-free Reads%	
						Before	After
	Pindel	**0.8806**	**0.9373**	**0.7640**	**0.9245**		**87.10%**
$\mathcal{D}_1$	Karect	0.6211	0.7199	0.6335	0.5041	54.01%	85.38%
	Fiona	0.3759	0.4124	0.4647	0.2572		78.91%
	Pindel	**0.9489**	**0.9710**	**0.9078**	**0.9548**		86.89%
$\mathcal{D}_2$	Karect	0.9297	0.9706	0.9070	0.8731	20.28%	**90.13%**
	Fiona	0.6930	0.7391	0.7435	0.5333		74.65%
	Pindel	**0.9771**	**0.9875**	**0.9700**	**0.9712**		**98.96%**
$\mathcal{D}_3$	Karect	0.9675	0.9879	0.9683	0.9206	19.82%	95.57%
	Fiona	0.7425	0.7773	0.7756	0.5881		77.46%
	Pindel	**0.9700**	**0.9874**	**0.9622**	**0.9558**		**89.90%**
$\mathcal{D}_4$	Karect	0.9334	0.9738	0.9177	0.8815	19.84%	89.22%
	Fiona	0.6760	0.7114	0.7046	0.5310		72.61%
	Pindel	**0.9689**	**0.9860**	**0.9578**	**0.9570**		**89.19%**
$\mathcal{D}_5$	Karect	0.9336	0.9752	0.9127	0.8908	19.87%	88.54%
	Fiona	0.6743	0.7079	0.7031	0.5321		72.39%
	Pindel	**0.9166**	**0.9427**	**0.9095**	**0.8759**		**78.59%**
$\mathcal{D}_6$	Karect	0.8265	0.8906	0.8004	0.7477	19.93%	76.85%
	Fiona	0.5343	0.5714	0.5407	0.4381		57.90%

Gain (I)/(D)/(S) are the gain metrics regarding insertion, deletion and substitution errors respectively.

CHAPTER 5. SUMMARY AND DISCUSSION

It can be observed that Pindel corrects more errors than both Karect and Fiona on all the datasets, as measured both by gains and percentages of error-free reads after error correction, with the sole exception on $\mathcal{D}_2$, where Karect outperforms Pindel in terms of the latter metric. The margin of improvement is largest for $\mathcal{D}_1$, where the average read length is shorter than the other datasets. Note that having shorter read lengths deteriorates the performance for all methods, however Pindel is less sensitive to decreasing read lengths than alignment-based methods which rely on the read overlap sizes. All methods display increased performances as the coverage increases. The hypothesis is that Pindel is more resilient to low coverage because the performance gap increases as the coverage decreases.

Pindel has better gains than competing methods for all categories of errors (Table 4.2). Despite the fact that Pindel only models non-consecutive deletion errors, it corrects more deletions than Karect and Fiona on all six datasets. The reason is that the majority of consecutive deletion errors occur in homopolymers. As long as there are no more than $\lceil \frac{l_h}{2} \rceil$ deletion errors in a homopolymer, where l_h is the homopolymer length in the true genomic sequence, Pindel can address such deletion errors by reinterpreting the consecutive deletions as isolated, single deletion errors. In §3.2.2 the necessity of a modified Viterbi algorithm in order to handle long insertion errors was discussed. This modified Viterbi algorithm increased the overall gain for Pindel by 6.5% on average.

Finally, the sensitivity of Pindel's performance to the choice of k on various datasets is discussed (see §A1). Overall, Pindel's performance is highly robust with respect to the $k$mer length, especially when the average read length is considerably larger than k.

APPENDIX A. ADDITIONAL MATERIAL

Appendix

A1 Robustness analysis

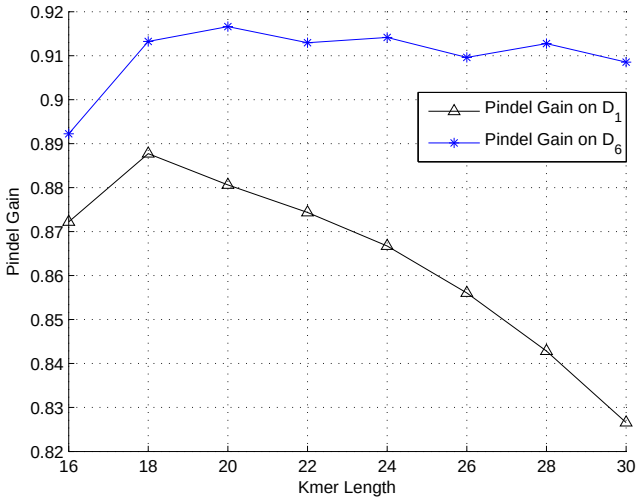

Figure A1: Pindel's gains on datasets $\mathcal{D}_1$, $\mathcal{D}_6$, for $k$mer length $= \{16, 18, \ldots, 30\}$, and fixed $\frac{k}{D} = 2$. The plot shows Pindel outperforms Karect's gains of 0.6211 and 0.8265 on $\mathcal{D}_1$ and $\mathcal{D}_6$ for all $k = \{16, 18, \ldots, 30\}$.

In §3.2.3, a heuristic is proposed to choose an appropriate k that induces $k$mer-uniqueness, while retaining sufficient redundancy. To study the fluctuations of performance around the choice of k, a robustness analysis of Pindel's performance with respect to k was performed on datasets $\mathcal{D}_1$ and $\mathcal{D}_6$. Pindel ran on both datasets, with even k ranging from 16 to 30, with $d = \frac{k}{2}$ and all the other tuning parameters fixed to the values specified in §3.2.3.

As can be observed from Fig. A1, Pindel's gain fluctuates from about 0.82 to 0.87 for $\mathcal{D}_1$, and from about 0.89 to 0.92 for $\mathcal{D}_6$, with the peak performances obtained at $k = 18$ and $k = 20$ for $\mathcal{D}_1$ and $\mathcal{D}_6$ respectively. While these variations are noticeable, for all evaluated k values, Pindel outperforms Karect with sizeable margins on both datasets, indicating the overall robustness of performance with regard to the choice of k.

Pindel's performance is fairly stable for k between 18 and 30 on dataset $\mathcal{D}_6$, whereas it gradually declines on dataset $\mathcal{D}_1$. It can be hypothesized that the two contrasting performance trajectories are largely attributed to the difference in the average read length, $\overline{l_i}$ between the two datasets. That is, when $\overline{l_i}$ is more comparable to the typical values of k, the increment in k results in more drastic decline in the redundancy. For instance, increasing k from 18 to 30 results in 16% less $k$mers per read when $\overline{l_i} = 92$ ($\mathcal{D}_1$), but only 3.4% less $k$mers when $\overline{l_i} = 366$ ($\mathcal{D}_6$).

A2 EM Derivations

In this section, an expectation-maximization algorithm that iteratively maximizes the penalized log-likelihood function is derived.

A2.1 E step

Let $\boldsymbol{\theta}$ denote the vector all model parameters, and let $\mathcal{S}$ denote the set of true genomic sequences that generate $\mathcal{R}$. Then, define $\ell_c(\boldsymbol{\theta} \mid \mathcal{R}, \mathcal{S})$ as the complete-data log-likelihood.

To evaluate the conditional expectation of the complete-data log-likelihood, the following is defined.

$$\begin{aligned} \mathcal{N}_H^{d_k}(\mathbf{x}_{i,k}) =& \{\boldsymbol{\omega} : \boldsymbol{\omega} \in \mathcal{K}_1, D_H(\boldsymbol{\omega}, \mathbf{x}_{i,k}) \leq d_k\}, \\ \mathcal{N}^{d}(\mathbf{x}_{i,t}) =& \{\boldsymbol{\omega} : \boldsymbol{\omega} \in \mathcal{K}, D(\boldsymbol{\omega}, \mathbf{x}_{i,t}) \leq d\}, \end{aligned} \tag{A1}$$

where $\mathcal{N}_{\mathcal{H}}{}^{d_k}(\mathbf{x}_{i,k})$ and $\mathcal{N}^d(\mathbf{x}_{i,t})$ are respectively referred to as the Hamming and edit distance neighborhoods of the observed $k$mer $\mathbf{x}_{i,k}$ and $\mathbf{x}_{i,t}$.

For the E-step, it is required to evaluate the conditional expectation of the above complete-data log-likelihood, $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$, which is,

$$
\begin{aligned}
Q(\boldsymbol{\theta}, \boldsymbol{\theta}^*) &= Q_I(\boldsymbol{\theta}, \boldsymbol{\theta}^*) + Q_T(\boldsymbol{\theta}, \boldsymbol{\theta}^*) + Q_E(\boldsymbol{\theta}, \boldsymbol{\theta}^*) \\
&= \sum_{i=1}^{r} \sum_{\boldsymbol{\omega} \in \mathcal{N}_H^{d_k}(\mathbf{x}_{i,k})} \log \pi(\widetilde{\boldsymbol{\omega}}) \cdot \zeta_{i,k}(\boldsymbol{\omega}) \\
&\quad + \sum_{i=1}^{r} \sum_{t=k}^{l_i - 1} \sum_{\boldsymbol{\omega} \in \mathcal{N}^d(\mathbf{x}_{i,t}),\ \beta \in \mathcal{B}} \log p(\boldsymbol{\omega} \blacktriangleright \beta \mid \boldsymbol{\omega}) \cdot \xi_{i,t}(\boldsymbol{\omega}, \boldsymbol{\omega} \blacktriangleright \beta) \\
&\quad + \sum_{i=1}^{r} \sum_{t=k+1}^{l_i} \sum_{\boldsymbol{\omega} \in \mathcal{N}^d(\mathbf{x}_{i,t})} \log f(\mathbf{x}_{i,t[k]}, \mathbf{y}_{i,t[k]} \mid \boldsymbol{\omega}, \mathbf{x}_{i,t-1}) \cdot \zeta_{i,t}(\boldsymbol{\omega}), \qquad \text{(A2)}
\end{aligned}
$$

where $Q(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$ has been partitioned into three components corresponding to the contributions from the initial state distribution, the transition distribution and the emission distribution. In Eq. (A2), the quantities $\zeta_{i,t}(\boldsymbol{\omega})$ and $\xi_{i,t}(\boldsymbol{\omega}, \boldsymbol{\omega} \blacktriangleright \beta)$ are for the probabilities of the hidden states at each position in each read,

$$
\begin{aligned}
\xi_{i,t}(\boldsymbol{\omega}, \boldsymbol{\nu}) &= P\left(\mathbf{s}_{i,t} = \boldsymbol{\omega}, \mathbf{s}_{i,t+1} = \boldsymbol{\nu} \mid \mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}^*\right), \\
\zeta_{i,t}(\boldsymbol{\omega}) &= P\left(\mathbf{s}_{i,t} = \boldsymbol{\omega} \mid \mathbf{x}_i, \mathbf{y}_i, \boldsymbol{\theta}^*\right),
\end{aligned}
$$

given the current parameter vector $\boldsymbol{\theta}^*$.

A2.2 M step

The objective function to be maximized in the M-step is $\widetilde{Q}(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = Q(\boldsymbol{\theta}, \boldsymbol{\theta}^*) - \rho \mathcal{J}(\boldsymbol{\theta}) - \sum_{\boldsymbol{\omega} \in \mathcal{K}_1 : \mathcal{L}(\boldsymbol{\omega}) = 1} \lambda_{\boldsymbol{\omega}} \left(\sum_{\beta \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})} q_1(\boldsymbol{\omega} \blacktriangleright \beta \mid \boldsymbol{\omega}) - 1 \right)$. Now let us discuss the estimation of the individual parameters.

MLEs of p_d and p_i.

Since the ℓ_0-penalty $\rho \mathcal{J}(\boldsymbol{\theta})$ and the equality constraint do not involve p_d and p_i, the objective function to maximize, with regard to p_d and p_i is simply $Q_T(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$. And $Q_T(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$ can be expanded into,

$$
\begin{aligned}
&Q_T(\theta,\theta^*) = \\
&\sum_{i=1}^{r}\sum_{t=k}^{l_i-1}\sum_{\substack{\boldsymbol{\omega}\in\mathcal{N}^d(\mathbf{x}_{i,t}),\\ \beta_1\in\mathcal{B}_1}} [\log(1-p_d-p_i)+\log q(\boldsymbol{\omega}\blacktriangleright\beta_1 \mid \boldsymbol{\omega}_{1+\varepsilon..})]\cdot\xi_{i,t}(\boldsymbol{\omega},\boldsymbol{\omega}\blacktriangleright\beta_1) \\
&+\sum_{i=1}^{r}\sum_{t=k}^{l_i-1}\sum_{\substack{\boldsymbol{\omega}\in\mathcal{N}^d(\mathbf{x}_{i,t}),\\ \beta_2\in\mathcal{B}_2}} \left[\log p_d+\log q(\boldsymbol{\omega}\blacktriangleright\beta_{2[1]} \mid \boldsymbol{\omega}_{1+\varepsilon..})+\log q\Big((\boldsymbol{\omega}\blacktriangleright\beta_{2[1]})\blacktriangleright\beta_{2[2]} \mid \boldsymbol{\omega}\blacktriangleright\beta_{2[1]}\Big)\right]\cdot\xi_{i,t}(\boldsymbol{\omega},\boldsymbol{\omega}\blacktriangleright\beta_2) \\
&+\sum_{i=1}^{r}\sum_{t=k}^{l_i-1}\sum_{\boldsymbol{\omega}\in(\mathcal{K}_1\cup\mathcal{K}_1^{\circlearrowleft})\cap\mathcal{N}^d(\mathbf{x}_{i,t})} \log p_i\cdot\xi_{i,t}(\boldsymbol{\omega},\boldsymbol{\omega}^{\circlearrowleft}), \qquad \text{(A3)}
\end{aligned}
$$

where $q(\cdot|\boldsymbol{\omega})$ denotes $q_1(\cdot|\boldsymbol{\omega})$ if $\mathcal{L}(\boldsymbol{\omega})=1$, and $q_2(\cdot|\boldsymbol{\omega})$ if $\mathcal{L}(\boldsymbol{\omega})=2$. For completeness, $(\boldsymbol{\omega}^{\circlearrowleft})^{\circlearrowleft}=\boldsymbol{\omega}^{\circlearrowleft}$ is also defined.

Taking the partial derivative of $Q(\theta,\theta^*)$ with respect to p_d, p_i and setting them to 0, we get the following.

$$
\widehat{p_i} \propto \sum_{i=1}^{r}\sum_{t=k}^{l_i-1}\sum_{\boldsymbol{\omega}(\mathcal{K}_1\cup\mathcal{K}_1^{\circlearrowleft})\cap\mathcal{N}^d(\mathbf{x}_{i,t})} \xi_{i,t}(\boldsymbol{\omega},\boldsymbol{\omega}^{\circlearrowleft}) \qquad \text{(A4)}
$$

$$
\widehat{p_d} \propto \sum_{i=1}^{r}\sum_{t=k}^{l_i-1}\sum_{\boldsymbol{\omega}\in\mathcal{N}^d(\mathbf{x}_{i,t}),\beta_2\in\mathcal{B}_2} \xi_{i,t}(\boldsymbol{\omega},\boldsymbol{\omega}\blacktriangleright\beta_2) \qquad \text{(A5)}
$$

MLEs of $g_0(.,.)$ and $\varrho(.)$.

To estimate the emission parameters, which are only involved in $Q_E(\boldsymbol{\theta},\boldsymbol{\theta}^*)$, it can be seen that,

$$
\begin{aligned}
Q_E(\theta,\theta^*) &= \sum_{i=1}^{r}\sum_{t=k+1}^{l_i}\sum_{\boldsymbol{\omega}\in\mathcal{N}^d(\mathbf{x}_{i,t})} \log f(\mathbf{x}_{i,t[k]},\mathbf{y}_{i,t[k]} \mid \boldsymbol{\omega},\mathbf{x}_{i,t-1})\cdot\zeta_{i,t}(\boldsymbol{\omega}) \\
&= \sum_{i=1}^{r}\sum_{t=k+1}^{l_i}\sum_{\boldsymbol{\omega}\in\mathcal{N}^d(\mathbf{x}_{i,t})} \log g_0(\mathbf{x}_{i,t[k]} \mid \boldsymbol{\omega}_{[k+\varepsilon(\boldsymbol{\omega})]})\cdot\zeta_{i,t}(\boldsymbol{\omega}) \\
&\quad+\sum_{i=1}^{r}\sum_{t=k+1}^{l_i}\sum_{\boldsymbol{\omega}\in\mathcal{N}^d(\mathbf{x}_{i,t})} \log\varrho_{\cdot}(\mathbf{y}_{i,t[k]} \mid \boldsymbol{\omega},\mathbf{x}_{i,t})\cdot\zeta_{i,t}(\boldsymbol{\omega}). \qquad \text{(A6)}
\end{aligned}
$$

It follows that by taking the partial derivative of the above equation with respect to $g_0(.,.)$ and $\varrho_{\cdot}(.)$, and setting them to zero, the following MLEs of the emission parameters can be derived

$$
\widehat{g_0}(\beta|\alpha) \propto \sum_{i=1}^{r}\sum_{t=k+1}^{l_i}\sum_{\boldsymbol{\omega}\in\mathcal{N}^d(\mathbf{x}_{i,t})} \mathbb{1}\{\boldsymbol{\omega}[k+\varepsilon(\boldsymbol{\omega})]=\alpha,\mathbf{x}_{i,t}[k]=\beta\}\cdot\zeta_{i,t}(\boldsymbol{\omega}), \qquad \text{(A7)}
$$

$$\widehat{\varrho}_1(\tau) \propto \sum_{i=1}^{r} \sum_{t=k+1}^{l_i} \sum_{\boldsymbol{\omega} \in \mathcal{N}^d(\mathbf{x}_{i,t})} \mathbb{1}\{\mathbf{x}_{i,t}[k] = \boldsymbol{\omega}[k], \boldsymbol{\omega} \in \mathcal{K}_1, \mathbf{y}_{i,t}[k] = \tau\} \cdot \zeta_{i,t}(\boldsymbol{\omega}), \tag{A8}$$

$$\widehat{\varrho}_2(\tau) \propto \sum_{i=1}^{r} \sum_{t=k+1}^{l_i} \sum_{\boldsymbol{\omega} \in \mathcal{N}^d(\mathbf{x}_{i,t})} \mathbb{1}\{\mathbf{x}_{i,t}[k] \neq \boldsymbol{\omega}[k], \boldsymbol{\omega} \in \mathcal{K}_1, \mathbf{y}_{i,t}[k] = \tau\} \cdot \zeta_{i,t}(\boldsymbol{\omega}), \tag{A9}$$

$$\widehat{\varrho}_3(\tau) \propto \sum_{i=1}^{r} \sum_{t=k+1}^{l_i} \sum_{\boldsymbol{\omega} \in \mathcal{N}^d(\mathbf{x}_{i,t})} \mathbb{1}\{\boldsymbol{\omega} \in \mathcal{K}_2, \mathbf{y}_{i,t}[k] = \tau\} \cdot \zeta_{i,t}(\boldsymbol{\omega}), \tag{A10}$$

$$\widehat{\varrho}_4(\tau) \propto \sum_{i=1}^{r} \sum_{t=k+1}^{l_i} \sum_{\boldsymbol{\omega} \in \mathcal{N}^d(\mathbf{x}_{i,t})} \mathbb{1}\{\boldsymbol{\omega} \in \mathcal{K}_1^{\circlearrowleft}, \mathbf{y}_{i,t}[k] = \tau\} \cdot \zeta_{i,t}(\boldsymbol{\omega}). \tag{A11}$$

MPLE of $q_1(\boldsymbol{\nu} \mid \boldsymbol{\omega})$.

In order to estimate the transition probabilities $q_1(\boldsymbol{\nu} \mid \boldsymbol{\omega})$, the objective function needs to be maximized

$$\widetilde{Q}_T(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = Q_T(\boldsymbol{\theta}, \boldsymbol{\theta}^*) - \rho \mathcal{J}(\boldsymbol{\theta}) - \sum_{\boldsymbol{\omega} \in \mathcal{K}_1 : \mathcal{L}(\boldsymbol{\omega}) = 1} \lambda_{\boldsymbol{\omega}} \left(\sum_{\beta \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})} q_1(\boldsymbol{\omega} \blacktriangleright \beta \mid \boldsymbol{\omega}) - 1 \right).$$

For simplicity of notation, let us define, for $t > k$,

$$\mathcal{N}^d_{\otimes,1}(\mathbf{x}_{i,t}) = \left\{ (\boldsymbol{\omega}, \boldsymbol{\nu}) \in \mathcal{N}^d(\mathbf{x}_{i,t}) \times \mathcal{N}^d(\mathbf{x}_{i,t+1}) : \varepsilon(\boldsymbol{\nu}) = 0, \boldsymbol{\nu} = \boldsymbol{\omega} \blacktriangleright \beta,\ \beta \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega}) \right\}, \tag{A12}$$

which contains all pairs $(\boldsymbol{\omega}, \boldsymbol{\nu})$ such that there exists a $k$mer-to-$k$mer transition $\boldsymbol{\omega}_{1+\varepsilon..} \to \boldsymbol{\nu}$. Similarly, for $t > k$, define

$$\mathcal{N}^d_{\otimes,2}(\mathbf{x}_{i,t}) = \Bigg\{ (\boldsymbol{\omega}, \boldsymbol{\nu}, \boldsymbol{\mu}) : \quad (\boldsymbol{\omega}, \boldsymbol{\omega}') \in \mathcal{N}^d(\mathbf{x}_{i,t}) \times \left\{ \boldsymbol{\omega}^* \in \mathcal{N}^d(\mathbf{x}_{i,t+1}) : \varepsilon(\boldsymbol{\omega}^*) = 1 \right\}, \\ \boldsymbol{\nu} = \boldsymbol{\omega} \blacktriangleright \boldsymbol{\omega}'[k], \boldsymbol{\omega}'[k] \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega}_{1+\varepsilon..}), \boldsymbol{\mu} = \boldsymbol{\nu} \blacktriangleright \boldsymbol{\omega}'[k+1], \boldsymbol{\omega}'[k+1] \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\nu}) \Bigg\}, \tag{A13}$$

which considers all $k/(k+1)$mer-to-$(k+1)$mer transition pairs factored into two consecutive $k$mer transitions.

Using the above notation, we have the following,

$$\tilde{Q}_T(\theta,\theta^*) =$$

$$\sum_{i=1}^{r}\sum_{t=k}^{l_i-1}\sum_{(\boldsymbol{\omega},\boldsymbol{\nu})\in\mathcal{N}^d_{\otimes,1}(\mathbf{x}_{i,t})}\Bigg[\log(1-p_d-p_i)+\mathbb{1}\{\mathcal{L}(\boldsymbol{\omega})=1\}\cdot\log q_1(\boldsymbol{\nu}\mid\boldsymbol{\omega}_{1+\varepsilon..})$$
$$+\mathbb{1}\{\mathcal{L}(\boldsymbol{\omega})=2)\}\cdot\left(\log q_1(\overline{\boldsymbol{\omega}_{1+\varepsilon..}}\mid\overline{\boldsymbol{\nu}})-\log\pi(\widetilde{\boldsymbol{\omega}_{1+\varepsilon..}})+\log\pi(\widetilde{\boldsymbol{\nu}})\right)\Bigg]\cdot\xi_{i,t}(\boldsymbol{\omega},\boldsymbol{\nu}) \tag{A14}$$

$$+\sum_{i=1}^{r}\sum_{t=k}^{l_i-1}\sum_{(\boldsymbol{\omega},\boldsymbol{\nu},\boldsymbol{\mu})\in\mathcal{N}^d_{\otimes,2}(\mathbf{x}_{i,t})}\Bigg[\log p_d$$
$$+\mathbb{1}\{\mathcal{L}(\boldsymbol{\omega})=1\}\log q_1(\boldsymbol{\nu}\mid\boldsymbol{\omega}_{1+\varepsilon..})+\mathbb{1}\{\mathcal{L}(\boldsymbol{\nu})=1\}\log q_1(\boldsymbol{\mu}\mid\boldsymbol{\nu})$$

$$+\mathbb{1}\{\mathcal{L}(\boldsymbol{\omega})=2)\}\left(\log q_1(\overline{\boldsymbol{\omega}_{1+\varepsilon..}}\mid\overline{\boldsymbol{\nu}})-\log\pi(\widetilde{\boldsymbol{\omega}_{1+\varepsilon..}})+\log\pi(\widetilde{\boldsymbol{\nu}})\right)$$
$$+\mathbb{1}\{\mathcal{L}(\boldsymbol{\nu})=2\}\left(\log q_2(\overline{\boldsymbol{\nu}}\mid\overline{\boldsymbol{\mu}})-\log\pi(\widetilde{\boldsymbol{\nu}})+\log\pi(\widetilde{\boldsymbol{\mu}})\right)\Bigg]\cdot\xi_{i,t}(\boldsymbol{\omega},\boldsymbol{\omega}\blacktriangleright\boldsymbol{\mu}_{[k-1\ldots k]}) \tag{A15}$$

$$+c$$

$$-\rho\sum_{\substack{\boldsymbol{\omega}\in\mathcal{K}_1:\\ \mathcal{L}(\boldsymbol{\omega})=1}}\sum_{\beta_1\in\mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})}\log[1+q_1(\boldsymbol{\omega}\blacktriangleright\beta_1|\boldsymbol{\omega})/\gamma]-\sum_{\substack{\boldsymbol{\omega}\in\mathcal{K}_1:\\ \mathcal{L}(\boldsymbol{\omega})=1}}\lambda_{\boldsymbol{\omega}}\left(\sum_{\beta_1\in\mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})}q_1(\boldsymbol{\omega}\blacktriangleright\beta_1\mid\boldsymbol{\omega})-1\right), \tag{A16}$$

where c are some constant (related to self transition) that does not involve $q_1(\cdot\mid\cdot)$. In the above equation, the two components in (A14) and (A15) respectively correspond to the $k/(k+1)$mer-to-$k$mer transitions, and the $k/(k+1)$mer-to-$(k+1)$mer transitions.

Therefore,

$$
\begin{aligned}
&\frac{\partial}{\partial q_1(\boldsymbol{\nu} \mid \boldsymbol{\omega})} \tilde{Q}_T(\theta, \theta^*) = \\
&\frac{1}{q_1(\boldsymbol{\nu} \mid \boldsymbol{\omega})} \Bigg[\sum_{i=1}^{r} \sum_{t=k}^{l_i-1} \xi_{i,t}(\boldsymbol{\omega}, \boldsymbol{\nu}) + \xi_{i,t}(\overline{\boldsymbol{\nu}}, \overline{\boldsymbol{\omega}}) \\
&\quad + \sum_{i=1}^{r} \sum_{t=k}^{l_i-1} \sum_{\beta_2 \in \mathcal{B}_2} \xi_{i,t}(\boldsymbol{\omega}, \boldsymbol{\omega} \blacktriangleright \beta_2) \cdot \mathbb{1}\{\boldsymbol{\omega} \blacktriangleright \beta_{2[1]} = \boldsymbol{\nu}, \beta_{2[1]} \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})\} \\
&\quad + \sum_{i=1}^{r} \sum_{t=k}^{l_i-1} \sum_{\beta_2 \in \mathcal{B}_2} \xi_{i,t}(\overline{\boldsymbol{\nu}}, \overline{\boldsymbol{\nu}} \blacktriangleright \beta_2) \cdot \mathbb{1}\{\overline{\boldsymbol{\nu}} \blacktriangleright \beta_{2[1]} = \overline{\boldsymbol{\omega}}, \beta_{2[1]} \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\nu})\} \\
&\quad + \sum_{i=1}^{r} \sum_{t=k}^{l_i-1} \sum_{\substack{\beta_2 \in \mathcal{B}_2, \\ \boldsymbol{\mu} \in \mathcal{N}^d(\mathbf{x}_{i,t})}} \xi_{i,t}(\boldsymbol{\mu}, \boldsymbol{\mu} \blacktriangleright \beta_2) \cdot \mathbb{1}\{\boldsymbol{\mu} \blacktriangleright \beta_{2[1]} = \boldsymbol{\omega}\} \cdot \mathbb{1}\{(\boldsymbol{\mu} \blacktriangleright \beta_{2[1]}) \blacktriangleright \beta_{2[2]} = \boldsymbol{\nu}\} \\
&\qquad \cdot \mathbb{1}\{\beta_{2[1]} \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\mu}), \beta_{2[2]} \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})\} \\
&\quad + \sum_{i=1}^{r} \sum_{t=k}^{l_i-1} \sum_{\substack{\beta_2 \in \mathcal{B}_2, \\ \boldsymbol{\mu} \in \mathcal{N}^d(\mathbf{x}_{i,t})}} \xi_{i,t}(\boldsymbol{\mu}, \boldsymbol{\mu} \blacktriangleright \beta_2) \cdot \mathbb{1}\{\boldsymbol{\mu} \blacktriangleright \beta_{2[1]} = \overline{\boldsymbol{\nu}}\} \cdot \mathbb{1}\{(\boldsymbol{\mu} \blacktriangleright \beta_{2[1]}) \blacktriangleright \beta_{2[2]} = \overline{\boldsymbol{\omega}}\} \\
&\qquad \cdot \mathbb{1}\{\beta_{2[1]} \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\mu}), \beta_{2[2]} \in \mathcal{T}_{\mathcal{K}_1}(\overline{\boldsymbol{\nu}})\} \Bigg] \\
&\quad - \rho \frac{1}{\gamma + q_1(\boldsymbol{\nu} \mid \boldsymbol{\omega})} - \lambda_{\boldsymbol{\omega}} \\
&= \frac{1}{q_1(\boldsymbol{\nu} \mid \boldsymbol{\omega})} \widetilde{\xi}(\boldsymbol{\omega}, \boldsymbol{\nu}) - \rho \frac{1}{\gamma + q_1(\boldsymbol{\nu} \mid \boldsymbol{\omega})} - \lambda_{\boldsymbol{\omega}}. \qquad \text{(A17)}
\end{aligned}
$$

Subsequently, the maximum penalized likelihood estimator (MPLE) for $q_1(\boldsymbol{\nu} \mid \boldsymbol{\omega})$ is derived as the following.

$$
\widehat{q_1(\boldsymbol{\nu} \mid \boldsymbol{\omega})} = \frac{\widetilde{\xi}(\boldsymbol{\omega}, \boldsymbol{\nu}) - \gamma\lambda_{\boldsymbol{\omega}} - \rho}{2\lambda_{\boldsymbol{\omega}}} \pm \frac{\sqrt{\left(\gamma\lambda_{\boldsymbol{\omega}} + \rho - \widetilde{\xi}(\boldsymbol{\omega}, \boldsymbol{\nu})\right)^2 + 4\gamma\lambda_{\boldsymbol{\omega}} \left(\widetilde{\xi}(\boldsymbol{\omega}, \boldsymbol{\nu})\right)}}{2\lambda_{\boldsymbol{\omega}}},
$$

where $\widetilde{\xi}(\boldsymbol{\omega}, \boldsymbol{\nu})$ is the summation term in the square brackets of Eq. (A17), and the value of the Lagrange multiplier $\lambda_{\boldsymbol{\omega}}$ is determined by numerically solving the equation

$$
\sum_{\beta' \in \mathcal{B}_1} \widehat{q_1(\boldsymbol{\omega} \blacktriangleright \beta' | \boldsymbol{\omega})} = 1. \qquad \text{(A18)}
$$

For tractability of the M-step, which involves solving the equation (A18) numerically, all outgoing transitions of the same $k$mer need to have the same label, which explains why $k$mers were labeled instead of individual transitions.

A2.3 Initialization of the EM

Provided that errors are scarce, the $k$mer-to-$k$mer transition probabilities $q_1(\cdot \mid \cdot)$ are initialized based on the observed information in $\mathcal{R}$. Specifically, for every $k$mer $\boldsymbol{\omega} \in \mathcal{K}_1$ with $\mathcal{L}(\boldsymbol{\omega}) = 1$, the incidence $n(\beta \mid \boldsymbol{\omega})$ of transition $\boldsymbol{\omega} \to \beta$ in $\mathcal{R}$, for $\beta \in \mathcal{T}_{\mathcal{K}_1}(\boldsymbol{\omega})$ is counted. By defining $\widetilde{n}(\beta|\boldsymbol{\omega}) = n(\beta \mid \boldsymbol{\omega}) + n(\overline{\boldsymbol{\omega}[1]} \mid \overline{\boldsymbol{\omega} \blacktriangleright \beta})$, $\widetilde{n}(\beta|\boldsymbol{\omega})$ is the observed number of times $\boldsymbol{\omega} \to \beta$ as witnessed by both strands. $q_1(\beta \mid \boldsymbol{\omega})$ is initialized using the M-step update (see §A2.2), plugging-in the observed counts $\widetilde{n}(\beta \mid \boldsymbol{\omega})$ as if they were the expected counts computed in the E-step, to induce sparsity in the parameter space. To initialze the initial state distribution, the same observed counts were utilized. For $\boldsymbol{\omega} \in \mathcal{K}_1$, if $\widetilde{\boldsymbol{\omega}}$ is the lexically smaller $k$mer between $(\boldsymbol{\omega}, \overline{\boldsymbol{\omega}})$, then

$$\pi^{(0)}(\widetilde{\boldsymbol{\omega}}) \propto \sum_{\beta \in \mathcal{T}_{\mathcal{K}_1}(\widetilde{\omega})} n(\beta \mid \widetilde{\boldsymbol{\omega}}) + \sum_{\beta \in \mathcal{T}_{\mathcal{K}_1}(\overline{\widetilde{\omega}})} n(\beta \mid \overline{\widetilde{\boldsymbol{\omega}}}).$$

It follows that the dependent $k$mer-to-$k$mer transition probabilities $q_2(\cdot \mid \cdot)$ can be initialized using Eq.(3.5). Finally, to finish the initialization of $p(\boldsymbol{\omega} \blacktriangleright \beta \mid \boldsymbol{\omega})$, $p_i^{(0)} = p_d^{(0)} = \frac{1}{3}$ were used.

For the emission parameters, the initialization was as follows,

$$g_0^{(0)}(\beta \mid \beta') = \mathbb{1}\{\beta = \beta'\} \cdot 0.99 + \mathbb{1}\{\beta \neq \beta'\} \cdot \frac{0.01}{3}, \quad \beta, \beta' \in \mathcal{B}_1,$$
$$\varrho_j^{(0)}(y) = \frac{1}{N_q}, \quad q_{\min} \leq y \leq q_{\max},\ j = 1, \ldots, 4,$$

where N_q is number of distinct quality scores observed in $\mathcal{R}$.

BIBLIOGRAPHY

Alexander, D. H. and Lange, K. (2011). Enhancements to the ADMIXTURE algorithm for individual ancestry estimation. BMC Bioinformatics, 12(1):246.

Allam, A., Kalnis, P., and Solovyev, V. (2015). Karect: accurate correction of substitution, insertion and deletion errors for next-generation sequencing data. Bioinformatics, 31(21):3421–3428.

Bragg, L. M., Stone, G., Butler, M. K., Hugenholtz, P., and Tyson, G. W. (2013). Shining a light on dark sequencing: Characterising errors in Ion Torrent PGM data. PLoS Computational Biology, 9(4).

Gordon, D. and Green, P. (2013). Consed: a graphical editor for next-generation sequencing. Bioinformatics, 29(22):2936–2937.

Heo, Y., Wu, X.-L., Chen, D., Ma, J., and Hwu, W.-M. (2014). BLESS: Bloom filter-based error correction solution for high-throughput sequencing reads. Bioinformatics, 30(10):1354–1362.

Ilie, L., Fazayeli, F., and Ilie, S. (2011). HiTEC: accurate error correction in high-throughput sequencing data. Bioinformatics, 27(3):295–302.

Ip, C., Loose, M., Tyson, J., de Cesare, M., Brown, B., Jain, M., Leggett, R., Eccles, D., Zalunin, V., Urban, J., Piazza, P., Bowden, R., Paten, B., Mwaigwisya, S., Batty, E., Simpson, J., Snutch, T., Birney, E., Buck, D., Goodwin, S., Jansen, H., O'Grady, J., and Olsen, H. (2015). MinION analysis and reference consortium: Phase 1 data release and analysis [version 1; referees: awaiting peer review]. F1000Research, 4(1075).

Jünemann, S., Sedlazeck, F. J., Prior, K., Albersmeier, A., John, U., Kalinowski, J., Mellmann, A., Goesmann, A., von Haeseler, A., Stoye, J., and Dag, H. (2013). Updating benchtop sequencing performance comparison. Nature Biotechnology, 31(4):294–296.

Kao, W.-C., Chan, A. H., and Song, Y. S. (2011). Echo: a reference-free short-read error correction algorithm. Genome Research, 21(7):1181–1192.

Kao, W.-C., Stevens, K., and Song, Y. S. (2009). Bayescall: A model-based base-calling algorithm for high-throughput short-read sequencing. Genome Research, 19(10):1884–1895.

Kelley, D. R., Schatz, M. C., and Salzberg, S. L. (2010). Quake: quality-aware detection and correction of sequencing errors. Genome Biol, 11(11):R116.

Laehnemann, D., Borkhardt, A., and McHardy, A. C. (2015). Denoising DNA deep sequencing data-high-throughput sequencing errors and their correction. Briefings in Bioinformatics.

Li, H. and Durbin, R. (2010). Fast and accurate long-read alignment with Burrows-Wheeler transform. Bioinformatics, 26(5):589–595.

Liu, Y., Schröder, J., and Schmidt, B. (2013). Musket: a multistage k-mer spectrum-based error corrector for Illumina sequence data. Bioinformatics, 29(3):308–315.

Loman, N. J., Misra, R. V., Dallman, T. J., Constantinidou, C., Gharbia, S. E., Wain, J., and Pallen, M. J. (2012). Performance comparison of benchtop high-throughput sequencing platforms. Nature Biotechnology, 30(5):434–439.

Lou, D. I., Hussmann, J. A., McBee, R. M., Acevedo, A., Andino, R., Press, W. H., and Sawyer, S. L. (2013). High-throughput DNA sequencing errors are reduced by orders of magnitude using circle sequencing. Proceedings of the National Academy of Sciences, 110(49):19872–19877.

Medvedev, P., Scott, E., Kakaradov, B., and Pevzner, P. (2011). Error correction of high-throughput sequencing datasets with non-uniform coverage. Bioinformatics, 27(13):i137–i141.

Merriman, B., Ion Torrent R&D, and Rothberg, J. (2012). Progress in Ion Torrent semiconductor chip based sequencing. Electrophoresis, 33(23):3397.

Nagarajan, N. and Pop, M. (2013). Sequence assembly demystified. Nature Review Genetics, 14(3):157–167.

Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2):257–286.

Rothberg, J. M., Hinz, W., Rearick, T. M., Schultz, J., Mileski, W., Davey, M., Leamon, J. H., Johnson, K., Milgrew, M. J., Edwards, M., et al. (2011). An integrated semiconductor device enabling non-optical genome sequencing. Nature, 475(7356):348–352.

Salmela, L. (2010). Correction of sequencing errors in a mixed set of reads. Bioinformatics, 26(10):1284–1290.

Salmela, L. and Schröder, J. (2011). Correcting errors in short reads by multiple alignments. Bioinformatics, 27(11):1455–1461.

Schirmer, M., Ijaz, U. Z., D'Amore, R., Hall, N., Sloan, W. T., and Quince, C. (2015). Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform. Nucleic Acids Research, 43(6):e37.

Schmitt, M. W., Kennedy, S. R., Salk, J. J., Fox, E. J., Hiatt, J. B., and Loeb, L. A. (2012). Detection of ultra-rare mutations by next-generation sequencing. Proceedings of the National Academy of Sciences, 109(36):14508–14513.

Schröder, J., Schröder, H., Puglisi, S. J., Sinha, R., and Schmidt, B. (2009). SHREC: a short-read error correction method. Bioinformatics, 25(17):2157–2163.

Schulz, M. H., Weese, D., Holtgrewe, M., Dimitrova, V., Niu, S., Reinert, K., and Richard, H. (2014). Fiona: a parallel and automatic strategy for read error correction. Bioinformatics, 30(17):i356–i363.

Schulz, M. H., Zerbino, D. R., Vingron, M., and Birney, E. (2012). Oases: Robust de novo RNA-seq assembly across the dynamic range of expression levels. Bioinformatics, 28(8):1086–1092.

Shendure, J. and Ji, H. (2008). Next-generation DNA sequencing. Nature Biotechnology, 26(10):1135–1145.

Wang, Y., Yang, Q., and Wang, Z. (2015). The Evolution of Nanopore Sequencing. Frontiers in Genetics, 5.

Yang, X., Chockalingam, S. P., and Aluru, S. (2013). A survey of error-correction methods for next-generation sequencing. Briefings in Bioinformatics, 14(1):56–66.

Yang, X., Dorman, K. S., and Aluru, S. (2010). Reptile: representative tiling for short read error correction. Bioinformatics, 26(20):2526–2533.

Yin, X. (2016). PhD Dissertation in Preparation. Iowa State University.

Yin, X., Song, Z., Dorman, K., and Ramamoorthy, A. (2013a). PREMIER Turbo: PRobabilistic Error-correction using Markov Inference in Errored Reads using the Turbo principle. In IEEE Global Conference on Signal and Image Processing (GlobalSIP).

Yin, X., Song, Z., Dorman, K. S., and Ramamoorthy, A. (2013b). PREMIER–PRobabilistic Error-correction using Markov Inference in Errored Reads. In IEEE International Symposium on Information Theory Proceedings (ISIT), pages 1626–1630.