Accurate k-mer Classification Using Read Profiles

Yoshihiko Suzuki[1]
Okinawa Institute of Science and Technology Graduate University, Okinawa, Japan

Gene Myers[2]
Okinawa Institute of Science and Technology Graduate University, Okinawa, Japan
Max Planck Institute of Molecular Cell Biology and Genetics, Dresden, Germany
Center for Systems Biology Dresden, Dresden, Germany

Abstract

Contiguous strings of length k, called k-mers, are a fundamental element in many bioinformatics tasks. The number of occurrences of a k-mer in a given set of DNA sequencing reads, its k-mer count, has often been used to roughly estimate the copy number of a k-mer in the genome from which the reads were sampled. The problem of estimating copy numbers, called here the *k-mer classification* problem, has been based on simply analyzing the histogram of counts of all the k-mers in a data set, thus ignoring the positional context and dependency between multiple k-mers that appear nearby in the underlying genome. Here we present an efficient and significantly more accurate method for classifying k-mers by analyzing the sequence of k-mer counts along each sequencing read, called a *read profile*. By analyzing read profiles, we explicitly incorporate into the model the dependencies between the positionally adjacent k-mers and the sequence context-dependent error rates estimated from the given dataset. For long sequencing reads produced with the accurate high-fidelity (HiFi) sequencing technology, an implementation of our method, **ClassPro**, outperforms the conventional, histogram-based method in every simulation dataset of fruit fly and human with various realistic values of sequencing coverage and heterozygosity. Within only a few minutes, ClassPro achieves an average accuracy of $> 99.99\%$ across reads without repetitive k-mers and $> 99.5\%$ across all reads, in a typical fruit fly simulation data set with a $40\times$ coverage. The resulting, more accurate k-mer classifications by ClassPro are in principle expected to improve any k-mer-based downstream analyses for sequenced reads such as read mapping and overlap, spectral alignment and error correction, haplotype phasing, and trio binning to name but a few. ClassPro is available at `https://github.com/yoshihikosuzuki/ClassPro`.

2012 ACM Subject Classification Applied computing → Molecular sequence analysis

Keywords and phrases K-mer, K-mer count, K-mer classification, HiFi sequencing

Digital Object Identifier 10.4230/LIPIcs.WABI.2022.10

Acknowledgements We wish to thank Shinichi Morishita, Yuta Suzuki, Bansho Masutani, Ryo Nakabayashi, Charles Plessy, and Michael Mansfield for their feedback and stimulating works. We also thank the Scientific Computing and Data Analysis section of Research Support Division and Communication and Public Relations Division at OIST for providing HPC resources and for proofreading of the manuscript, respectively.

1 Introduction

Long read DNA sequencing technologies are enabling the *de novo* reconstruction of reference quality genomes providing the impetus for projects such as the Vertebrate Genome Project [26], the Darwin Tree of Life Project [30] and the Human Pangenome Project [33], whose goals are to build reference atlases of entire phyla, eco-systems of living creatures, or worldwide

[1] Current affiliation: Department of Computational Biology and Medical Sciences, The University of Tokyo, Japan
[2] Corresponding author

human populations. In addition to dramatic gains in read length, the most recent arrival of long reads with an error rate of only 0.2%, as for example realized by PacBio's HiFi protocol [34], permits haplotype phasing and the resolution of many complex repetitive regions [4, 5, 7, 8, 21] because there is almost always a modest level of heterogeneity between the haplotypes or repeat elements in wild-type animals. However, the goal of a perfect telomere to telomere, phased reconstruction of a multiploid genome is as yet unrealized [11], requiring either better or more data, or better assembly algorithms. In this paper, an initial analysis of a high fidelity shotgun data set delivers precise information about phasing and repetitiveness, that should in principle improve the performance of any downstream assembly method.

In a typical high fidelity data set of a genome G, a collection of reads R is collected so that the genome is covered $c = \sum_{S \in R} |S|/|G|$ times where c is typically 10-30 and the read length $|S|$ averages from 10Kbp to 25Kbp (with current technologies). Note carefully, that c is the **haploid** coverage and G is the phased genome, i.e. it is not a consensus haplotype but the set of all distinct haplotype sequences. For example, for a human genome, G is 6Gbp in length and consists of 46 sequences corresponding to the full diploid set of chromosomes. In the treatment that follows, only R is known, and G is a hypothetical used for definitional purposes.

A k-mer is a string of length k defined over the DNA nucleotides. For any set of DNA sequences X, a k-mer counter such as Meryl [32], Jellyfish [17], KMC [13], or FastK [19], calculates the number of occurrences of each distinct k-mer in X. For a k-mer, α, let $\#_X(\alpha)$ denote the number of times α occurs in X. Reference to "*the* count of α" implicitly refers to $\#_R(\alpha)$, the count in the read data set R. A naive expectation for a shotgun data set R and implied genome G is that $\#_R(\alpha) \sim c \cdot \#_G(\alpha)$ subject to stochastic fluctuations in the arrival of sequencing errors and the read sampling process. This in turn implies that the histogram, $\mathcal{H}_R$, of $\#_R$ (often called the k-mer spectrum of R) will typically have discernible peaks at $0, c, 2c, \cdots$ (Fig. 1a) where the k-mers about 0 are considered to be due to errors in the reads as it is most likely that $\#_G = 0$ for said; i.e. they are not in G. This basic observation has lead to a number of k-mer-based analysis tools. For example, GenomeScope [25] estimates the size, ploidy, heterozygosity, and repeat fraction of G and the error rate of R by fitting a negative binomial mixture model to the histogram $\mathcal{H}_R$. As another example, KAT [16], Merqury [27], and Merfin [6] evaluate the completeness of an assembly of R using a complete table of $\#_R$ and the histogram $\mathcal{H}_R$.

In this paper, we consider the ***k*-mer classification problem** to be that of inferring $\#_G(\alpha)$ for every k-mer α in R. In all previous work of which we are aware, this classification is based solely on the count of α in the context of the histogram $\mathcal{H}_R$; e.g. a k-mer α is deemed an error (i.e. $\#_G(\alpha) = 0$) if the count of α is less than some fixed threshold based on an examination of $\mathcal{H}_R$. Let $\mathcal{H}_{R|v}$ be the histogram or distribution of $\{\ \#_R(\alpha)\ :\ \#_G(\alpha) = v\ \}$, that is, the counts of the k-mers with classification v. Then because of sequencing error and the stochasticity of the Poisson sampling process, the distributions $\mathcal{H}_{R|0}$, $\mathcal{H}_{R|1}$, $\mathcal{H}_{R|2}$, $\cdots$ can and typically do overlap significantly, increasingly so as a function of v and the sequencing error rate. This inseparability implies that the many assemblers (e.g [23, 2, 12, 29, 14, 3, 1]) using k-mers for tasks such as seeding alignments, spectral error correction, or haplotype phasing, are working with classifications (or probabilities of classifications) that are often incorrect as much as 5–10% of the time. So clearly, having highly accurate classifications would improve the performance of all of these systems.

Here we present a method of k-mer classification over a high-fidelity read data set that has a typically accuracy of $> 99.9\%$. We do so by exploiting the contextual information between positionally close k-mer counts along each read $S \in R$. We term the sequence

counts for consecutive k-mers along a read S, a **count profile** (Fig. 1a). In a count profile, neighboring k-mer counts are dependent on each other, providing much stronger statistical leverage than found in the histogram $\mathcal{H}_R$. For example, if a k-mer is an error (classification 0), then typically $O(k)$ neighbors about this k-mer in the profile are also errors. An even more significant observation is that if two consecutive k-mers in the profile have the same classification then their counts will only vary if (a) there has been a read arrival or departure in the underlying sampling of reads or (b) some number of reads that have an error in said k-mer changes. In contrast, if they have different classifications, then there will be a difference on the order of c or more in their counts. As a concrete example, consider a 20× HiFi data set of 15Kbp reads (40× of a diploid genome). A read arrives on average every 375bp and an error occurs every 500bp assuming a 0.2% error rate. One thus expects typically a change of 0, 1 or 2 counts between successive k-mers in the same class and a change on the order of 20 or so counts if the classification changes.

While the read sampling process for PacBio data is to first order Poisson, the likelihood of an error at a given point in a sequence is known to vary widely depending on context. For example, the most common errors that account for $\sim 80\%$ of all the errors in the HiFi sequencing are homopolymer indels [34, 22], followed by copy number errors in dinucleotide satellites, and thereafter those of trinucleotide satellites. It is therefore important to account for this as it can considerably affect the probability of a count transition being due to a classification change versus due to error and read sampling. In other work, the dominance of homopolymer errors was effectively by-passed with homopolymer compression of the reads as in HiCanu [22] and LJA [1]. This approach however does not account for elevated error rates around di- and tri-nucleotide satellites, so in this work we develop a data-driven model of sequence-dependent error.

In this paper, we describe an algorithm and software implementation, ClassPro, that for each read count profile of a diploid genome, classifies every k-mer α in the profile into one of the four types: **error** ($\#_G(\alpha) = 0$), **haploid** ($\#_G(\alpha) = 1$), **diploid** ($\#_G(\alpha) = 2$), and **repeat** ($\#_G(\alpha) \geq 3$). The concept of a count profile has been sporadically seen in previous work [36, 24, 18, 16], but we leverage both stochastic and deterministic properties of a profile to improve k-mer classification. We show empirically that the resulting classifications are highly accurate and so using these in previously studied contexts like error correction, read overlap detection, haplotype phasing, and trio binning should result in significant performance improvements.

2 Preliminaries

2.1 Problem definition and terminology

Throughout the paper, we focus on classifying the k-mers of a single read $S \in R$ of length N. Let the sequence of S, $Seq(S) = s_{-k+1}s_{-k+2}\cdots s_{-1}s_0s_1\cdots s_{N-k}$, and let the $N-k+1$ k-mers of S, $Kmer(S) = \alpha_0\alpha_1\cdots\alpha_{N-k}$, where $\alpha_i = s_{i-k+1}\cdots s_i$. Note that this unusual definition of $Seq(S)$ prefixed by $k-1$ negative indices is for a definitional purpose so that the first k-mer affected by any change to nucleotide s_j is α_j, which will be used in Methods. Lastly, let the k-mer count profile of S, $Count(S) = c_0c_1\cdots c_{N-k}$, where $c_i = \#_R(\alpha_i)$.

For a k-mer α, we call it a haplo-mer iff $\#_G(\alpha) = 1$ and a diplo-mer iff $\#_G(\alpha) = 2$. Note that almost all of the diplo-mers are the homozygous k-mers shared among both alleles, but a very small fraction of them typically consists of paralogous copies of k-mers, especially in repetitive regions. Similarly we call α an error-mer iff $\#_G(\alpha) = 0$ and a repeat-mer iff $\#_G(\alpha) > 2$. (NB: a sequencing error in a read does not necessarily imply that every

k-mer spanning it is an error-mer as it may occur elsewhere in the genome, however this happens rarely, increasingly so as k increases.) Let $\mathcal{T}$ be the set of the four k-mer types, i.e. {E(rror), H(aplo), D(iplo), R(epeat)}. For each k-mer α_i in $Kmer(S)$, let $\tau_i \in \mathcal{T}$ be the true type of α_i, which is unknown, and $t_i \in \mathcal{T}$ be the inferred assignment of a type of α_i by some method. Then, our ultimate objective is to find the best sequence of assignments, $Class(S) = t_0 \cdots t_{N-k}$, such that the number of wrong classifications, i.e. $|\{\ \alpha_i \mid t_i \neq \tau_i\ \}|$, is minimized.

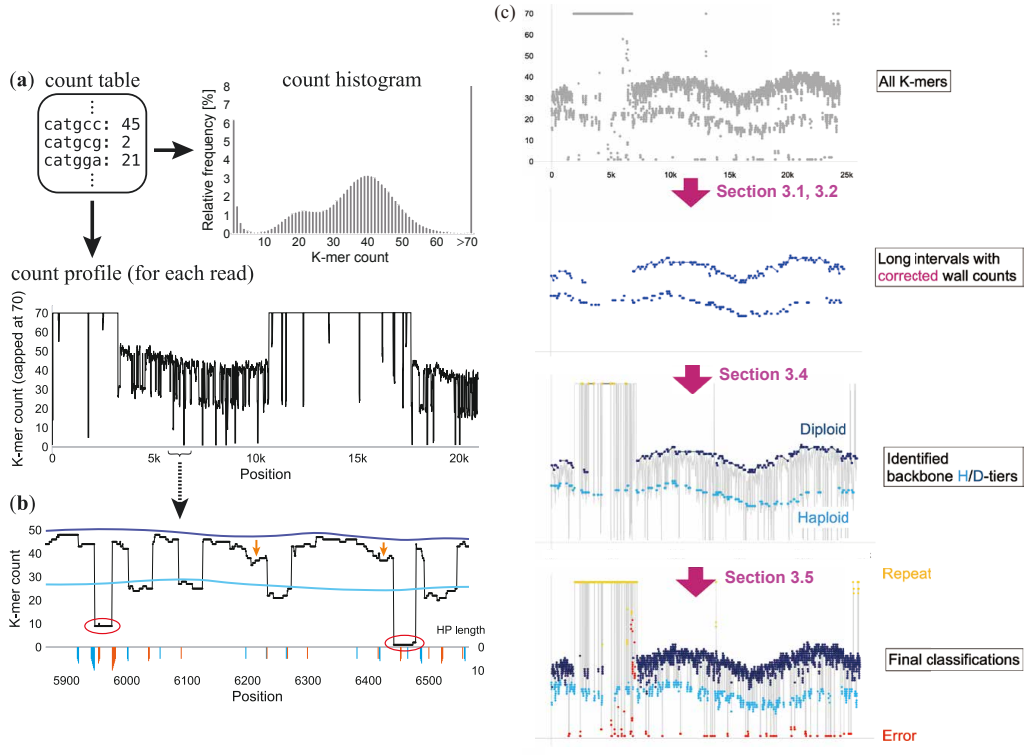

Figure 1 **(a)** Examples of a histogram $\mathcal{H}_R$ and a profile $Count(S)$ of k-mer counts (real HiFi data, $c = 20$ and $k = 40$). In the profile, consecutive k-mer counts are connected with solid lines. **(b)** While a count of 30 is hard to classify based only on $\mathcal{H}_R$, in this region we can conclude it is likely to be haploid rather than diploid because we clearly see two "tiers" of haploid (light blue curve) and diploid (dark blue) segments fluctuating smoothly. The sharp and large count drops (red circles) are caused by sequencing errors occurring in S. In contrast, errors in other reads result in small drops from a tier (orange arrows). The bars at the bottom indicate the homopolymer length for count drop (blue, $\overrightarrow{l}_i^{\mathrm{HP}}$ in Methods) and gain (orange, $\overleftarrow{l}_{i-k+1}^{\mathrm{HP}}$) where only $\geq$ 5bp are depicted. The drop to count 9 at position ~5,950 exemplifies an elevated number of co-occurrences of a homopolymer error. **(c)** The flow of ClassPro. The second and third subplots show reliable intervals instead of k-mers, and the background profile is depicted in gray in the third and last subplots for clarity.

2.2 Anatomy of a k-mer count profile

The intuition that read profiles are effective for the k-mer classification problem is based on two key observations that hold for a typical HiFi dataset: the **coherence principle** and the ***k*-knockout principle**. In brief, the coherence principle is that "adjacent k-mers with the same copy number have similar counts." In other words, the count change between two consecutive k-mers with the same copy number $\#_G$ is relatively much smaller than the

count change due to a copy number change. The k-knockout principle is that a transition from a higher copy number to a lower copy number in the event of a sequencing error or allelic variant lasts for roughly k-or-more k-mers, because $\sim k$ consecutive k-mers share the nucleotide(s) of the event. These two principles create local dependencies among adjacent k-mers that cannot be captured with a histogram.

To elaborate these principles, we consider what a count profile $Count(S)$ should look like from a generative perspective (Fig. 1b). First suppose a read has no errors and is sampled from a region in a diploid genome that is completely homozygous and non-repetitive. Then all the k-mers are diplo-mers and every change between two consecutive counts, c_{i-1} and c_i, is fully explained by the read sampling process. That is, c_i get $+1$ compared to c_{i-1} for every read starting at i and -1 for every read ending at $i-1$. Since the average number of read arrivals per position is $2c/N$ when c is the haploid coverage and N is the read length, for long reads, where $2c/N \ll 1$, the count profile without sequencing errors is very smooth. However, do note that over a large number of bases the counts can drift up and down significantly based on the underlying undulation in the Poisson sampling process, i.e. $|c_i - c_j|$ can be large when $|i-j|$ is large.

Next suppose the read now comes from a heterozygous region where the haplotype it was sampled from varies from its mate in a number of places. Then the k-mers spanning the variant sites will be haplo-mers and those not will be diplo-mers, effectively partitioning the profile into diplo-mer segments and haplo-mer segments. By coherence these segments will be smooth with $O(c)$ jumps between segments. Basically the diplo-mer and haplo-mer segments will create two layers, one roughly twice the height of the other while undulating under the Poisson sampling process. Furthermore, by the knockout principle the haplotype segments are $O(k)$ or longer (in the event two variant sites are less than k bases apart).

Finally consider the case where there are errors both in the read S under consideration and the set O of all the other reads that were sampled from the same region. When the error is in S, the profile of the k-mers containing the error drops to 1 or nearly so, happens roughly once every 500bp for a HiFi data set with a 0.2% error rate, and the profile count stays very low for $O(k)$ counts by the knockout principle. Errors in the other reads O create -1 drops like a read end event but in this case they last for only $O(k)$ consecutive counts before popping back up $+1$ and these fluctuations are much more frequent, occurring every $0.2c$ bases for haplo segments and $0.4c$ for diplo segments, e.g. every 25bp and every 12.5bp when $c = 20\times$.

In summary, if c is not too small and error is not too high, then the transition of counts in a single profile is caused by a combination of the following four factors in order of their possible effect size: (a) copy number changes, (b) sequencing errors in S, (c) sequencing errors in others O, and (d) read sampling fluctuation. The coherence principle implies that the effect sizes of (c) and (d) are generally much smaller than (a) and (b), and the k-knockout principle can be applied to (a), (b) and (c). However, we will only use the knockout principle for the special case of errors, as changes in the underlying repetitiveness of the genome can negate the length of a haplo-mer, diplo-mer, triplo-mer, $\cdots$ run but not so for an error-mer segment.

2.3 The approach

While we have introduced the fundamental components causing the movements in a count profile $Count(S)$, a full statistical model, i.e. $\Pr\{Seq(S), Count(S), Class(S)\}$, that incorporates all of these stochastic factors is very complicated and thus impractical. Our solution

to this challenge is to divide the classification problem into two heuristic parts: we first resolve local dependencies between k-mer counts due to errors using the k-knockout principle, and then identify haplo-/diplo-profiles using the coherence principle (Fig. 1c).

In Section 3.1, we describe how the "knockout length" of an error is precisely determined based on the sequence context and the type of the error, and present how to compute the probabilities of both errors in S and errors in O for each position. Using these error probabilities, we identify the change-points of k-mer classes in $Class(S)$ when the coherence principle breaks. We call these **walls** and split the profile into a set of contiguous segments partitioned by the walls wherein all the k-mers in an **interval** should belong to the same class. The classification by ClassPro is performed on the intervals (instead of the k-mers), and the inferred class ($\in \mathcal{T}$) of an interval is assigned to all the k-mers in the interval at the end. In Section 3.2 we introduce a criterion for selecting potential haplo-/diplo-intervals (called **reliable intervals**) from all the intervals, and estimate their error-free counts at each boundary wall, i.e. the counts that would occur if there were no errors in O (Fig. 1c, second plot). This partition into reliable intervals with corrected wall counts allows us to accurately approximate the complicated transition among all the k-mer counts in $Count(S)$, by only analyzing the count changes at the walls and between the walls.

In the latter part of the divided problem, we first classify only the reliable intervals (Section 3.4; Fig. 1c, third plot) and then do the rest of the intervals while fixing the classification results of the reliable intervals (Section 3.5; Fig. 1c, forth plot). Since the wall counts are corrected for the reliable intervals, at this point the transition between the wall counts of the reliable intervals should be only due to 1) read sampling fluctuation for those having the same class of H or D (i.e. the coherence principle), or 2) copy number changes for those having different classes. Although the optimal classifications for the reliable intervals can be obtained via dynamic programming (D.P.) if all the reliable intervals are haploid or diploid, repeats and errors cannot be handled in the same manner because of the lack of the coherence principle for them. Nevertheless, we employ a heuristics that combines two pseudo-D.P. sweeps in the forward and backward directions and empirically show that it achieves very accurate classifications over various simulation datasets.

3 Methods

3.1 Wall detection: How errors affect the profile

We term a position i in a profile a *wall* if and only if there is a "significant" change between the two counts c_{i-1} and c_i due to a state change (i.e. $t_{i-1} \neq t_i$). We also call a segment $[b..e)$ partitioned by two adjacent walls at b and e an *interval*. The start of an error state in either S or O causes a count drop and the end does a count gain, and below we describe how to determine walls by finding pairs of count drops and gains due to errors while considering positionally variable sequencing error rates due to low-complexity sequences. Another objective here is to evaluate how likely each interval is a product of errors in S.

Let F be a set of the types of sequence features that alter the sequencing error rate. For HiFi reads, we consider three types of low-complexity sequences: the homopolymers (HP; e.g. `aaaa`), the dinucleotide satellites (DS; `ctctct`), and the trinucleotide satellites (TS; `ctgctg`), denoted by $F = \{\text{HP, DS, TS}\}$. For each $f \in F$, let $\overrightarrow{l}_i^f$ and $\overleftarrow{l}_i^f$ be the maximal length of f on $Seq(S)$ up to position $i-1$ and that from i, respectively. For example, if $s_0 \cdots s_8 = \text{agggctcta}$, then $\overrightarrow{l}_4^{\text{HP}} = 3$ (`ggg`) and $\overleftarrow{l}_4^{\text{DS}} = 4$ (`ctct`). For each feature f, let $err^f(l)$ denote the average indel error rate right after f of length l. We estimate $err^f(l)$ directly from a given dataset with HIsim [20], which efficiently and comprehensively

computes the frequency of each error type using a k-mer count table and a user-specified count threshold between erroneous k-mers and normal k-mers. Since accurate estimation of error rates is difficult for large l due to the relatively small number of observations of such long low-complexity sequences in a dataset, we extrapolate the error rates for $l > l_{\max} = 5$ by fitting a quadratic function $err^f(l) = a_2 l^2 + a_1 l + a_0$ to the average estimated error rates for feature lengths up to $l_{\max}$.

Since the first k-mer affected by the start of a sequencing error in either S or O at position i is α_i (see Section 2.1), a count drop event at i, i.e. $c_{i-1} > c_i$, due to an error should depend on the nucleotide sequence up to $i-1$. Likewise, a count gain at i, i.e. $c_{i-1} < c_i$, due to the end of an error depends on the sequence context from $i-k+1$. Therefore, for each position i the sequence context-dependent error rate ε_i is represented as follows for each type of count change, i.e. drop $\searrow$ and gain $\nearrow$:

$$\varepsilon_i^{\searrow}(f) = err^f\left(\overrightarrow{l}_i^f\right)$$
$$\varepsilon_i^{\nearrow}(f) = err^f\left(\overleftarrow{l}_{i-k+1}^f\right)$$

In other words, $\varepsilon_i^{\searrow}$ and $\varepsilon_i^{\nearrow}$ represent the potential error rate that causes a count drop and a gain, respectively, between $i-1$ and i due to a low-complexity indel error of type $f \in F$. Regarding the other "high-complexity" errors other than F, we use $\bar{\varepsilon} = err^{\mathrm{HP}}(1)$ as the error rate of a single event of insertion, deletion, or substitution for both count drop and gain. We consider up to 5 bases for a single high-complexity error. We denote the set of all the possible error types above by Ω.

For each position i in S, let $\Delta_i \in \{\searrow, \nearrow\}$ denote the direction of the count transition between $i-1$ and i. Given a potential wall at i, let c_{in} and c_{out} be the count just inside and outside of the wall, respectively. That is, $(c_{\mathrm{in}}, c_{\mathrm{out}}) = (\min\{c_{i-1}, c_i\}, \max\{c_{i-1}, c_i\})$, which is equal to (c_i, c_{i-1}) if $\Delta_i = \searrow$ and (c_{i-1}, c_i) if $\Delta_i = \nearrow$. We approximate the "error-free" count (i.e. the count if errors do not exist) of c_{in} by c_{out} for each potential wall. Since the sequencing errors should occur independently among the error-free count, the count change between the two consecutive positions, $i-1$ and i, due to an error in S is modeled by a binomial distribution:

$$c_{\mathrm{in}} \sim \mathrm{Binomial}\left(c_{\mathrm{out}}, \varepsilon_i(\omega)\right)$$

where $\varepsilon_i(\omega)$ is the error rate given the type of the error $\omega \in \Omega$. That is, $\varepsilon_i(\omega) = \varepsilon_i^{\Delta_i}(\omega)$ if $\omega \in F$ (i.e. low-complexity errors) and otherwise $\varepsilon_i(\omega) = \bar{\varepsilon}$ for high-complexity errors. With this model, we can compute how likely a count change occurred by the sequencing errors, or how common it is. We define the probability $p_i^S(\omega)$ that a count change at i is caused due to an error in S whose type is ω by using the p-value of the one-sided binomial test:

$$\begin{aligned} p_i^S(\omega) &= \Pr\left\{X \geq c_{\mathrm{in}} \mid c_{\mathrm{out}}, \varepsilon_i(\omega)\right\} \\ &= \mathrm{BinomialTest}\left(c_{\mathrm{in}} \mid c_{\mathrm{out}}, \varepsilon_i(\omega)\right) \end{aligned}$$

We also define the probability $p_i^O(\omega)$ that a count change at i is caused due to the sequencing errors occurring in a subset of O as follows:

$$\begin{aligned} p_i^O(\omega) &= \Pr\left\{X \geq c_{\mathrm{out}} - c_{\mathrm{in}} \mid c_{\mathrm{out}}, \varepsilon_i(\omega)\right\} \\ &= \mathrm{BinomialTest}\left(c_{\mathrm{out}} - c_{\mathrm{in}} \mid c_{\mathrm{out}}, \varepsilon_i(\omega)\right) \end{aligned}$$

An error event makes a pair of count changes, and thus we wish to define the error probabilities for a pair of walls instead of a single count change. The length of an error-interval generated by a single contiguous error event is not always k bp. In HiFi reads, it is

usually smaller than k bp due to the low-complexity indel errors (Fig. 2a). More precisely, given the location i and the type of an error, the length of an error-interval is exactly given by $k+n-m-1$ bp where m and n are specified as follows. First, $m = \overrightarrow{l}^{f}_{i} > 0$ holds for the low-complexity errors of type f and $m = 0$ for the others, because that the error state in a profile can quickly return to the normal state due to the arbitrariness of low-complexity bases in terms of k-mers. Next, n indicates the number of bases in S that are affected by the error. That is, $n = 0$ holds if the error is a deletion in S or an insertion in O, and otherwise, $n(>0)$ is the number of the bases inserted in S, deleted in O, or substituted in S or O. For low-complexity errors of type f, n is the maximal length of the low-complexity sequence from i, i.e. $\overleftarrow{l}^{f}_{i}$. The essential point here is that for a given error type ω, the length of the error-interval caused by a single error event is uniquely determined. Therefore, for each position i, we calculate the probability that the count changes at both of the pair of i and its corresponding position are due to an error in S as the maximum product of probabilities among all possible error types Ω:

$$p_i^S = \begin{cases} \max_{\omega\in\Omega}\left\{p_i^S(\omega)\cdot p_{i+\pi}^S(\omega) \;\middle|\; \Delta_{i+\pi} = \nearrow\right\} & \text{if } \Delta_i = \searrow \\ \max_{\omega\in\Omega}\left\{p_{i-\pi}^S(\omega)\cdot p_i^S(\omega) \;\middle|\; \Delta_{i-\pi} = \searrow\right\} & \text{if } \Delta_i = \nearrow \end{cases}$$

where $\pi = k+n-m-1$. There is only one possible combination of m and n for each of the low-complexity errors and are $\eta+1$ for the high-complexity errors up to η bases (because $m=0$ always holds and $n = 0, \cdots, \eta$). Since $|F| = 3$ and $\eta = 5$ here, the total number of cases inspected per position is always a constant of 9. In the same manner, p_i^O is also computed using $p_i^O(\omega)$.

A set of walls is then determined by finding count changes that i) can be explained by errors in S or ii) cannot be explained by errors in O, that is, $\{\ i\ :\ p_i^S > \theta^S$ or $p_i^O < \theta^O\ \}$, where θ^S and θ^O are user-defined parameters (the default value is 10^{-5} for both). In practice, most of the positions in a HiFi read are not walls, and we do not perform the pairing for any position that already does not satisfy the condition. Note that the pairing of a count drop and gain above is applicable only to a single error event. For a long error-interval containing multiple error events in S, it is generally impossible to determine the exact locations and types of the errors within it. To handle this case, for every pair of two consecutive walls at position i and j that cannot be explained by O, as another possible p_i^S $(= p_j^S)$ we also calculate $p_i^S = \max_\omega\left\{p_i^S(\omega)\right\}\max_\omega\left\{p_j^S(\omega)\right\}$. We do not need to consider the multi-error cases for p_i^O because the same set of multiple errors rarely co-occurs among different reads in O. We take unions of overlapping error-intervals, and for each interval I we define the error probability p_I^{E} as the largest p_i^S among the overlapping error-intervals. The number of resulting walls per read is typically on the order of 10–100 (depending on the heterozygosity and repetitiveness), which is much smaller than N.

3.2 Collecting reliable intervals and correcting the wall counts

We now have a set of non-overlapping intervals split by walls, each of which should be comprised only of a single type of k-mer class in $\mathcal{T}$. While we no longer need to consider the dependencies between k-mers due to errors in S across multiple intervals, the dependencies due to errors in O, i.e. a pair of count drop and gain due to errors in O, still exist across multiple intervals. We resolve this by canceling out the decrease in counts due to errors in O at the walls although not all wall counts can be corrected as described below.

From the set of intervals, we extract "reliable" intervals where we can confidently correct the counts at walls and thus use for the determination of a backbone haplo-profile and diplo-profile in the next step (Fig. 2b). Specifically, an interval $I = [b..e)$ is called *reliable*

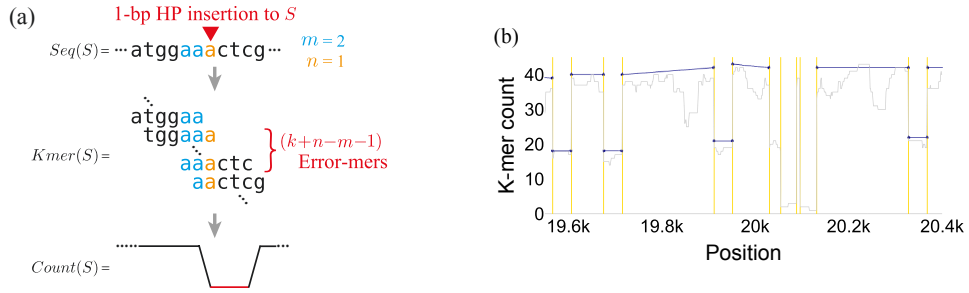

Figure 2 (a) An example of a pair of walls induced by a 1-bp homopolymer insertion event (orange) that occurred in S. Note carefully that the last k-mer, `aactcg`, has a normal count although it contains the inserted base, because of the arbitrariness of the insertion location of low-complexity bases in terms of the k-mer strings. **(b)** Examples of walls (yellow) and reliable intervals (navy) in a small region of a read profile (gray). Each reliable interval is represented and shown by a pair of corrected counts at the walls and a line connecting them.

if and only if i) the error probability p_I^{E} defined above is smaller than the threshold θ^S, ii) the counts at walls are not obviously repetitive, i.e. $\max\{c_b, c_{e-1}\} < \theta^R$, where θ^R is some (loose) threshold for repetitive count, and iii) the length of the interval $e - b$ is at least k. As for the repeat count threshold θ^R, we used the 6σ value of the global diploid coverage while assuming a Poisson distribution of sequencing coverage; that is, $\theta^R = d + 6\sqrt{d}$ where d is the diploid coverage and $\sqrt{d}$ is the standard deviation of Poisson(d). In the last condition we exclude short intervals that can be fully contained within a pair of count drop and gain due to errors in O and thus cannot provide sufficient information for count correction. In contrast, for an interval longer than or equal to k bp, the number of counts decreased at a wall due to errors in O can be estimated because one of the drop-gain pair (i.e. start or end position) of the error state in O that exists across the wall is expected to be contained in the interval in most cases. Another reason for the value of k is because a single SNV results in an interval of length k and this requirement keeps a haplo-interval caused by an SNV as a reliable interval, which avoids over-filtering of intervals.

The counts at both ends of each reliable interval, c_b and c_e, are corrected into $\hat{c}_b$ and $\hat{c}_e$, respectively, using the count changes among $\sim k$ k-mers from b and $\sim k$ k-mers up to e, respectively, based on a logic similar to that in the wall detection:

$$\hat{c}_b \leftarrow c_b + \sum_{i=b+1}^{b+k-1} \max\{c_i - c_{i-1}, 0\} - \sum_{i=b+1}^{b+\max_{f\in F}\left\{\overleftarrow{l}^{f}_{b+k-1}\right\}} \max\{c_{i-1} - c_i, 0\}$$

$$\hat{c}_e \leftarrow c_e + \sum_{i=e-k+1}^{e-1} \max\{c_i - c_{i+1}, 0\} - \sum_{i=e-\max_{f\in F}\left\{\overrightarrow{l}^{f}_{e-k+1}\right\}}^{e-1} \max\{c_{i+1} - c_i, 0\}$$

In both formulae, the first term represents the number of gains/drops within the k bases just after the start position and just before the end position. The second term is the number of gains/drops that actually have a corresponding drop/gain within k bases or less (i.e. drop-gain pairs that are actually contained in the interval) owing to the low-complexity errors.

3.3 Modeling count transition due to the read sampling fluctuation

In addition to sequencing errors, we define a probability of the count change between two positions with some distance due to read sampling fluctuation, i.e. arrivals and exits of the other reads on S. This represents the degree of coherence between k-mers in the same class and is crucial in the next step.

Let u and v ($u < v$) be the locations of two k-mers having the same (non-zero) copy number, i.e. $\#_G(\alpha_u) = \#_G(\alpha_v)(> 0)$. Let $\tilde{c}_u$ and $\tilde{c}_v$ be the error-free counts of α_u and α_v, respectively. For the read length N and the sequencing coverage of the class that the k-mers belong to, i.e. $C = \#_G(\alpha_u) \cdot c$ (c is the global haploid coverage), the distribution of the number of reads starting within a segment $[u..v]$ asymptotically follows Poisson(λ) where $\lambda = (v-u)C/N$ [15]. If we assume that the read departure process is independent from the arrival process, then the distribution of the number of reads ending within $[u..v]$ also follows Poisson(λ). Under the independence, the difference between counts $\tilde{c}_u$ and $\tilde{c}_v$ is modeled by the Skellam distribution [10], which represents the distribution of the difference between two variables independently following a Poisson distribution:

$$\tilde{c}_v - \tilde{c}_u \sim \text{Skellam}\,(\lambda, \lambda)$$

where the shape of the distribution is symmetrical with the mean of 0 given the two variables follow the same Poisson distribution. In practice, this probability is defined when the classes of the two k-mers are deemed identical, i.e. $t_u = t_v$. We thus denote the probability of read sampling fluctuation by $p^{\text{sample}}\,(\tilde{c}_u \to \tilde{c}_v \mid u, v, c^t)$ where c^t is the global coverage of the class $t = t_u (= t_v)$. Using this, we calculate the probability of the transition between haploid intervals and that between diploid intervals, although we cannot use it for repeat intervals because the copy numbers can be different in general between two repeat-mers.

3.4 Classification of the reliable intervals: Finding the backbone haplo- and diplo-profiles

Given a set of reliable intervals, we classify each reliable interval into one of the states $\mathcal{T}$. The main purpose of this step is to detect the backbone haplo- and diplo-profiles using only the corrected counts at the walls for the subsequent classification of the rest of the k-mers (see Fig. 1c). Let $I_i = [b_i..e_i]$ and $T_i \in \mathcal{T}$ denote the i-th interval and its assignment, respectively. Here assigning a specific class to I_i, i.e. $T_i = t'$, means that all the k-mers in I_i are classified as t' in the original profile. The corrected k-mer counts at the two walls, b_i and e_i, of I_i are $\hat{c}_{b_i}$ and $\hat{c}_{e_i}$, respectively.

First, suppose that every reliable interval is either a haplo-interval or a diplo-interval, i.e. $\mathcal{T} = \{\text{H,D}\}$. We assume that for an interval I_i we can estimate the local haploid-coverage and diploid-coverage at e_i given the assignment T_i, and let $\text{cov}[i][s][t]$ be the estimated coverage of class t ($\in \{\text{H,D}\}$) at e_i given $T_i = s$. We also assume that the transition from $T_i = s$ to $T_{i+1} = t$ is determined only by the count transition from $\text{cov}[i][s][t]$ to $\hat{c}_{b_{i+1}}$, i.e. count transition between walls in the sub-profile of class t. Then, (backtracking of) the following dynamic programming using likelihoods of initial classes and transitions gives the classifications of reliable intervals with the maximum likelihood:

$$\begin{aligned} \text{dp}[0][t] &= \Pr\{I_0 \mid T_0 = t\} \\ \text{dp}[i+1][t] &= \max_{s \in \{\text{H,D}\}} \{\text{dp}[i][s] + \Pr\{I_{i+1} \mid T_i = s, T_{i+1} = t\}\} \end{aligned}$$

where

$$\Pr\{I_0 \mid T_0 = t\} = \text{Poisson}\left(\hat{c}_{b_0} \mid c^t\right)$$
$$\Pr\{I_{i+1} \mid T_i = s, T_{i+1} = t\} = p^{\text{sample}}\left(\text{cov}[i][s][t] \to \hat{c}_{b_{i+1}} \mid e_i, b_{i+1}, c^t\right)$$

and c^t is the global coverage of the class t. In practice, $\text{cov}[i][s][t]$ is estimated as follows: i) for $t = s$, then $\hat{c}_{e_i}$ is directly set to $\text{cov}[i][s][t]$, and ii) for $t \neq s$, it is estimated using a linear interpolation using corrected wall counts of the haplo-intervals and diplo-intervals in the best path up to I_i given $T_i = s$.

There actually can exist some repeat-intervals and a small number of error-intervals that are not excluded as unreliable intervals, while the classification categories in the D.P. above cannot be directly extended to $\mathcal{T} = \{\text{E}, \text{H}, \text{D}, \text{R}\}$ because the k-mer counts of different error-intervals and repeat-intervals are generally independent of each other. We thus employ heuristic likelihoods for those intervals as follows. While we cannot use p^{sample} for repeat-intervals, we set the diploid coverage at I_i, $\text{cov}[i][s][\text{D}]$, plus its 2σ (under the assumption of Poisson distribution) as $\text{cov}[i][s][\text{R}]$ and define the likelihood of $T_{i+1} = \text{R}$ given $T_i = s$ as follows:

$$\Pr\{I_{i+1} \mid T_i = s, T_{i+1} = \text{R}\} = \begin{cases} 1 & \text{if } \hat{c}_{b_{i+1}} > \text{cov}[i][s][\text{R}] \\ \text{Binomial}\left(\hat{c}_{b_{i+1}} \mid \text{cov}[i][s][\text{R}], 1 - \bar{\varepsilon}\right) & \text{Otherwise} \end{cases}$$

where $\bar{\varepsilon}$ is the average sequencing error rate. For the probability of $T_{i+1} = \text{E}$ we reuse $p^{\text{E}}_{I_{i+1}}$ that was already computed in Section 3.1.

The classification result can be different in the forward direction and backward direction (where transition from b_{i+1} to e_i is considered instead of transition from e_i to b_{i+1} above) because of the independency of E and R and the estimation of the local coverages. However, we practically obtain accurate classifications by combining the classification result of the pseudo-D.P. in the forward direction and that in the backward direction. Specifically, we find the combined classifications with the maximum likelihood whose prefix is taken from the backward result and suffix is from the forward result, because the coverage estimation tends to become more accurate as the update of the D.P. proceeds.

3.5 Classification of the rest

We finally classify each of the remaining intervals while fixing the assignments of the reliable intervals that are classified as haploid or diploid. Given a focal interval $I = [b..e]$, let $\mathcal{I}_{-I}$ denote the intervals except I, and let $\mathcal{T}_{-I}$ be the assignments of the intervals except T_I, where assignments are initially given to only the reliable intervals. We assign to T_I the class that gives the maximum likelihood computed using the classification results of the other intervals:

$$T_I = \arg\max_t \Pr\{I \mid T_I = t, \mathcal{I}_{-I}, \mathcal{T}_{-I}\}$$

For the case of $I = \text{E}$ we reuse the error probability p^{E}_I, and for the other classes we decompose the probability above into the upstream transition p^+_I (in the direction toward b) and the downstream transition p^-_I (toward e), i.e. $\Pr\{I \mid T_I = t, \mathcal{I}_{-I}, \mathcal{T}_{-I}\} = p^+_I(t) \cdot p^-_I(t)$. For both p^+_I and p^-_I, we consider only the count changes at walls just like the classification of the reliable intervals. As possible events at walls, we consider both read sampling fluctuation and errors in O for $t \in \{\text{H},\text{D}\}$ and only errors in O for $t = \text{R}$. The probability of transition by read sampling fluctuation is calculated between I and the nearest interval J satisfying

$T_I = T_J$ ($\in$ {H,D}) using p^{sample}. The probability by errors in O is computed using the binomial distribution given an estimated coverage and sequencing error rate at a wall of I just like the wall detection (for $t \in$ {H,D}) and the reliable interval classification (for $t =$ R).

3.6 Simulation and real datasets

We adopted two model organisms, fruit fly and human. For the main simulation experiment, we downloaded the latest reference haploid genome sequence (GenBank accession numbers: GCF_000001215.4 [9] for fruit fly and GCA_009914755.3 [21] for human) and a publicly available real HiFi read dataset (BioProject accession numbers: PRJNA573706 for fruit fly and PRJNA586863 [21] for human). For each species we simulated a ground-truth diploid genome sequence from the reference haploid genome sequence using HIsim [20] and then generated synthetic reads using two long-read simulators, Badread [35] and HIsim, both of which build a sequencing error model from a given dataset using short ($\sim$ 10bp) k-mers.

As a baseline of the k-mer classification compared to ClassPro, we performed a histogram-based k-mer classification using GenomeScope with the `-fitted_hist` option [25], where the global thresholds between the four classes (i.e. E,H,D,R) are determined by finding change points of the class that gives the maximum probability according to the GenomeScope inference. We used $k = 40$ unless the value of k is explicitly stated below.

The average overall accuracy of the classifications for a dataset is defined as the number of k-mers with correct classifications divided by the total number of k-mers in the dataset (while regarding multiple k-mers of the same string on different reads or different positions as different k-mers), i.e. $|\{\ \alpha_i \mid t_i = \tau_i\ \}| \ /\ |\{\ \alpha_i \mid \alpha_i \in S, S \in R\ \}|$, and for each combination of parameters we took a harmonic mean of five datasets with different random seeds. In addition, to examine the detailed behavior of the two classification methods, we calculated the average local accuracy of the classifications in each 2Kbp non-overlapping window in the reads as well as the average overall accuracy.

Beyond confirming the expected average behavior, we also explored more practical and realistic cases. First we prepared a 40× simulation dataset of the human major histocompatibility complex (MHC) region by using a publicly available, high-quality diploid assembly of the MHC region [5] as the ground-truth diploid genome and mixing 20× synthetic reads generated from each of the two MHC haplotypes using HIsim. In addition, we downloaded a real 55× HiFi dataset of a diploid human sample HG002/NA24385 (BioProject: PRJNA586863) [37, 21] where an accurate, trio-based diploid assembly (GenBank: GCA_021950905.1 and GCA_021951015.1) [11] is available and can be used as a surrogate of the ground-truth diploid genome, although any missing sequences and false duplicated sequences in the assembly affect the accuracy estimation and thus the "accuracy" should not be perfectly accurate. To try to minimize the effect of the false positive/negative sequences on the accuracy, we ignored a read from accuracy calculation if more than 20% of the k-mers in the read are error-mers or more than 80% are repeat-mers.

4 Results

First we generated HiFi read datasets simulated from a synthetic diploid genome of fruit fly with a small genome size of $\sim$ 160Mbp to deeply investigate ClassPro's performance under various values of sequencing coverage of the reads and heterozygosity of the genome. We used 20×, 25×, 30×, 40×, and 50× as the diploid sequencing coverage, i.e. $2c$. As for the heterozygosity, we specified 0.05%, 0.1%, 0.3%, and 0.5% as the value of the `-p` option of HIsim, which correspond to GenomeScope's estimated heterozygosity of $\sim$ 0.11%, $\sim$ 0.22%, $\sim$

0.66%, and $\sim$ 1.1%, respectively (Fig. S1). We simulated the HiFi reads using two long-read simulators, Badread and HIsim. For each dataset we classified the k-mers in the reads using ClassPro (CP) and GenomeScope (GS; histogram-based method). ClassPro is efficiently parallelized since each read is handled independently, and for example, the degree of the speed up was constantly about 6× when using 8 threads, regardless of the coverage (Fig. S2). Assuming k is sufficiently small compared to the read length N and both $|\Omega|$ and $|\mathcal{T}|$ (which are always 9 and 4, respectively) are constants, the classification algorithm itself also runs fast in $O(N)$, and thus it takes only $\sim$ 100 seconds wall time to classify a whole 40× fruit fly dataset using 8 threads on an AMD Epyc 7702 CPU and SSD Lustre system, given a precomputed k-mer count table.

The average overall accuracy of CP was superior than GS in every combination of sequencing coverage and heterozygosity for both Badread datasets and HIsim datasets (Fig. 3, Fig. S3). For example, given a heterozygosity of 0.66% that is close to the estimated heterozygosity value of a real fruit fly dataset [22], in the Badread datasets the overall accuracy of CP exceeds 99.9% (GS=95.1%) when the global coverage per haploid c is 15× (which is typically called a 30× dataset) and does 99.99% (GS=96.8%) when $c = 20\times$ (i.e. a 40× dataset) for normal reads. This indicates that with a typical sequencing coverage ClassPro achieves almost perfect classifications in the most fundamental case where the distinguishment between erroneous, haploid, and diploid k-mers is the only problem. The identification of haploid/diploid k-mers is more difficult for repetitive reads than normal reads without repetitive k-mers in general because the distribution of haploid/diploid k-mers becomes sparse in repetitive regions and thus the number of k-mers that follow the coherence principle decreases. Nevertheless, the overall accuracy of CP for all the reads including repetitive reads (e.g. 99.7% when $c = 20\times$ in the fruit fly dataset above) is still higher than that of GS for only normal reads in every parameter combination. When we calculate the accuracy only with highly repetitive reads in each of which more than 80% of the k-mers are repeat-mers, for example, in a dataset with $c = 20\times$ and 0.22% heterozygosity the accuracy of CP is 98.1% (GS=96.1%), and the false-negative rate of error-mers in such reads is 0.3% (GS=1.0%). This implies that another advantage of CP especially for highly repetitive regions such as centromeres is that we can exclude more false error-mers, which should help singly unique nucleotide k-mers (SUNKs)-based methods (e.g. [3]).

The advantage of CP over GS becomes greater as the heterozygosity gets higher, because the overlap between the distribution of the haploid k-mers and that of the diploid k-mers in the k-mer count histogram (i.e. $\mathcal{H}_{R|1}$ and $\mathcal{H}_{R|2}$ in Introduction) becomes larger in a dataset with a higher genomic diversity. Fig. 3 clearly demonstrates that CP is more robust than GS against haplotype divergence or genome mutation and captures those signals better. We confirmed that CP is robust against the choice of the value of k compared to GS and that the value of k can be either even or odd because we use canonical k-mers in the k-mer counting (Fig. S4).

We further investigated the detailed behavior of the classification in each dataset by calculating the local sequencing coverage and accuracy for every 2Kbp window of the reads in the dataset (Fig. 4). In every parameter combination, the local accuracy of GS drops more quickly than CP when the window coverage deviates from the "sweet spot" that depends on the global coverage c. Thus, the local accuracy of GS classifications can be very low compared to the average overall accuracy not only when the local coverage is considerably lower than c but also when higher. In contrast, CP classifications are consistently better and more robust against changes in coverage than GS especially when the local coverage is higher than c. For example, the window accuracy of CP and GS for normal reads are 99.95%

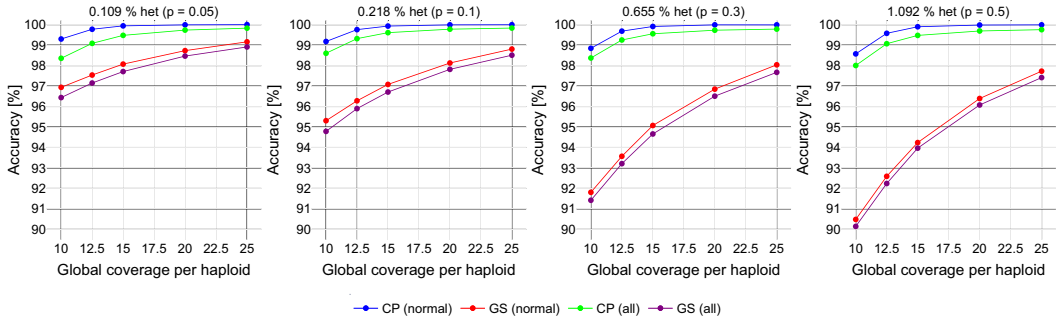

Figure 3 Average overall accuracy between ClassPro and GenomeScope across normal reads without repetitive k-mers and across all reads in the fruit fly simulation datasets by Badread.

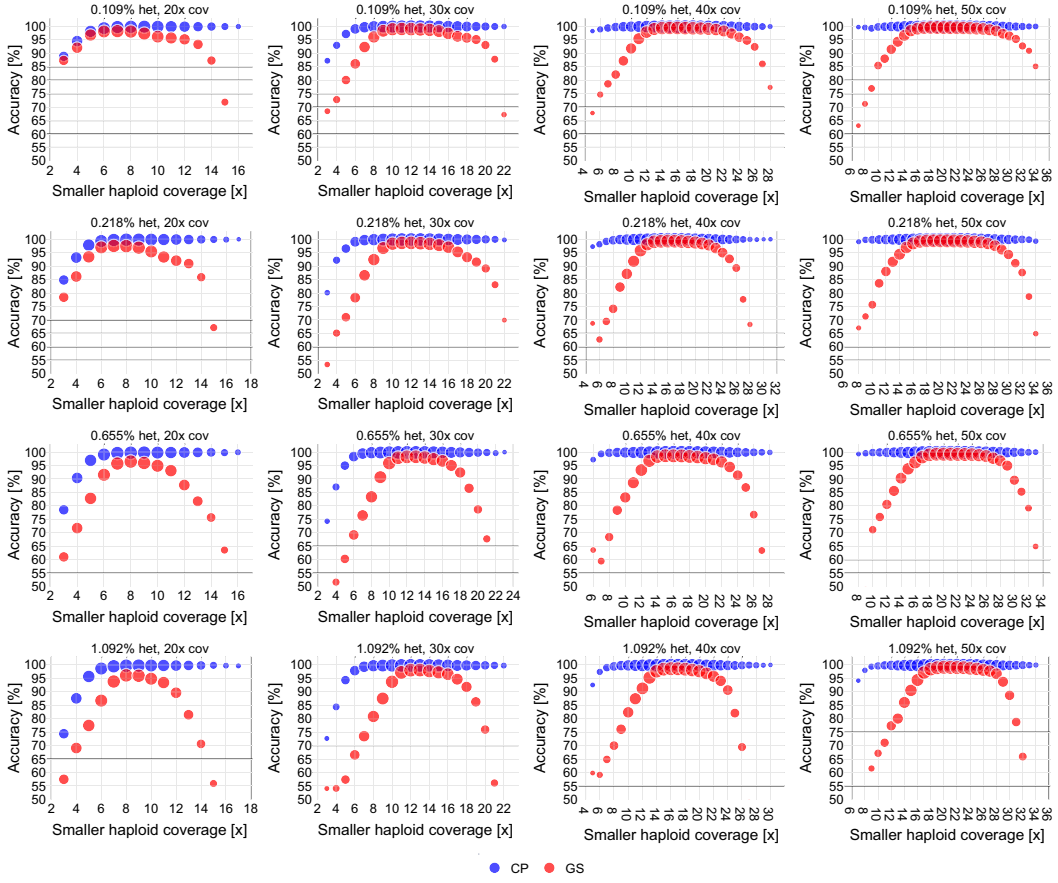

Figure 4 Relationship between local coverage and average accuracy at the resolution of 2Kbp windows across normal reads in the fruit fly simulation datasets. The size of the dots is $\log_2$(# of windows), and only dots of size larger than 100 are drawn. Note that several dots of GS whose accuracy is smaller than 50% are omitted in the plot.

and 25.14%, respectively, when the smaller average haploid coverage is 34× in a dataset with $2c = 50\times$ and $\sim 1.1\%$ heterozygosity. For both CP and GS, the local accuracy with a very small coverage such as 3× becomes worse as the heterozygosity increases because a higher heterozygosity makes the read profile more like a mosaic of haploid k-mers and diploid k-mers and thus requires a more sensitive discrimination between them, which is challenging given 3× per haplotype. The average local accuracy for all reads has the same tendency as normal reads, although there are some fluctuation due to repetitiveness (Fig. S5).

We also evaluated the performance of CP using both simulated and real human HiFi read datasets. We first confirmed that it works with human simulation datasets as well using various sequencing coverages given a typical heterozygosity of $\sim 0.2\%$ (Fig. S6). The accuracy was largely the same as that of the fruit fly dataset with the same heterozygosity: e.g. 99.9% by CP and 97.0% by GS for normal reads in 30× datasets. We then inspected a particular genomic region of interest to researchers. The human MHC region is a highly divergent and repetitive region and thus known to be difficult to accurately assemble [5]. With a simulated 40× dataset generated from a pair of real MHC haplotypes, we confirmed that CP performs well in a difficult-to-assemble region with an accuracy of 99.43% (GS=97.51%). Lastly we applied CP to a real 55× diploid human HiFi dataset while using a high-quality diploid assembly of the same sample as a substitute of the ground-truth genome, and the accuracy of CP was estimated as 99.09% (GS=97.56%). Note carefully that a small amount of remaining missing sequences and false duplicated sequences in the assembly would affect and slightly lower the accuracy estimation.

Note that CP can output different classification results for the same k-mer because it classifies each read independently, while the histogram-based approaches such as GS always classifies the same k-mer into the same class. Nevertheless, by virtue of the high accuracy of CP, the overall consistency of the CP classifications was, for example, over 99.9% in a 40× fruit fly dataset, implying that almost all of the k-mers are consistently classified and the classification results can be used as they are in most applications.

5 Discussion

We developed a novel approach to the k-mer classification problem using k-mer count read profiles, and confirmed that its software implementation, ClassPro, outperforms the conventional, most widely used method based on the k-mer count histogram in every combination of realistic parameter values of sequencing coverage and heterozygosity for two model organisms. The k-mer classification is a fundamental task and used in many sequence analysis programs including error correction, sequence alignment, and genome assembly, and thus the more accurate and robust k-mer classifications by ClassPro promise to help any of such applications boost their performance and accuracy.

The read profiles for ClassPro can be computed by the FastK k-mer counter. It uniquely and very efficiently delivers read profiles as a direct output, whereas with other k-mer counters one is forced to build each profile via a sequence of (relatively more expensive) k-mer table look ups. ClassPro outputs a FastQ-like file of the reads where the QV sequence for each read is replaced with a sequence over the alphabet {E,H,D,R} corresponding to the classification result of the k-mer at each position.

As a demonstration of the power of the improved k-mer classifications by ClassPro, our next target is to incorporate the k-mer classifications into sequence alignment and genome assembly. In the de Bruijn graph approach of genome assembly, ClassPro should offer a better elimination of error-mers for a higher space efficiency and also a more informative guide

for a graph touring. On the other hand, the string graph approach requires the sequence alignments between reads, and the seed selection step is necessary for practical sequence alignment methods. We are currently working on a better seed selection method using the k-mer classification result. Besides that, a more accurate detection and removal of the erroneous k-mers alone would be helpful for trio binning and so on.

Our approach utilizes the positional dependencies between the k-mers. However, it performs the k-mer classification for each read independently. Therefore, it would be natural to think of employing a more complicated data structure capable of handling the k-mer counts of all reads along with their positions simultaneously. The positional de Bruijn graph [28, 31] is apparently the most plausible one of such a representation, although the practical algorithm including the cycle handling due to repeats is not trivial.

The current implementation of ClassPro assumes as input only HiFi reads with an average error rate of $\sim 0.1\%$ (QV30). One direction for future research is to make the method more robust against noisy reads such as Oxford Nanopore reads. Given a sequencing error rate of $\varepsilon\%$, at most $\varepsilon K\%$ bases are expected to be error-mers in each read profile. Therefore, even using the recent Q20+ chemistry with a mean alignment accuracy of QV20 (i.e. 1% error), at most $1\% \times 40 = 40\%$ of the k-mers can be error-mers for a typical Nanopore read given $k = 40$, making a read profile look very erroneous compared to HiFi ($0.1\% \times 40 = 4\%$ error-mers). Moreover, the fluctuation of the haploid and diploid counts by errors in other reads increases as well, i.e. the coherence principle is weakened, making the classification more difficult.

Another possible research target is a utilization of a (not exact but) approximate k-mer counting method for generation and classification of read profiles. Although FastK and ClassPro currently handle only the exact k-mer counting, the whole computation process could be even faster if they can be replaced with an approximate k-mer counting system.

References

1 Anton Bankevich, Andrey V. Bzikadze, Mikhail Kolmogorov, Dmitry Antipov, and Pavel A. Pevzner. Multiplex de Bruijn graphs enable genome assembly from long, high-fidelity reads. *Nature Biotechnology*, 2022. doi:10.1038/s41587-022-01220-6.

2 Jonathan Butler *et al.* ALLPATHS: De novo assembly of whole-genome shotgun microreads. *Genome Research*, 18(5):810–820, 2008.

3 Andrey V. Bzikadze and Pavel A. Pevzner. Automated assembly of centromeres from ultra-long error-prone reads. *Nature Biotechnology*, 38(11):1309–1316, 2020.

4 Haoyu Cheng, Gregory T. Concepcion, Xiaowen Feng, Haowen Zhang, and Heng Li. Haplotype-resolved de novo assembly using phased assembly graphs with hifiasm. *Nature Methods*, 18(2):170–175, 2021.

5 Chen-Shan Chin *et al.* A diploid assembly-based benchmark for variants in the major histocompatibility complex. *Nature Communications*, 11(1):4794, 2020.

6 Giulio Formenti *et al.* Merfin: improved variant filtering and polishing via k-mer validation. *bioRxiv*, 2021. doi:10.1101/2021.07.16.452324.

7 Shilpa Garg. Computational methods for chromosome-scale haplotype reconstruction. *Genome Biology*, 22:101, 2021. doi:10.1186/s13059-021-02328-9.

8 David Heller, Martin Vingron, George Church, Heng Li, and Shilpa Garg. SDip: A novel graph-based approach to haplotype-aware assembly based structural variant calling in targeted segmental duplications sequencing. *bioRxiv*, 2020. doi:10.1101/2020.02.25.964445.

9 Roger A. Hoskins *et al.* The Release 6 reference sequence of the Drosophila melanogaster genome. *Genome Research*, 25(3):445–458, 2015.

10 J. O. Irwin. The frequency distribution of the difference between two independent variates following the same Poisson distribution. *Journal of the Royal Statistical Society*, 100(3):415–416, 1937.

11 Erich D. Jarvis *et al.* Automated assembly of high-quality diploid human reference genomes. *bioRxiv*, 2022. doi:10.1101/2022.03.06.483034.

12 David R. Kelley, Michael C. Schatz, and Steven L. Salzberg. Quake: quality-aware detection and correction of sequencing errors. *Genome Biology*, 11:R116, 2010. doi:10.1186/gb-2010-11-11-r116.

13 Marek Kokot, Maciej Długosz, and Sebastian Deorowicz. KMC 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17):2759–2761, May 2017.

14 Sergey Koren *et al.* *De novo* assembly of haplotype-resolved genomes with trio binning. *Nature Biotechnology*, 36(12):1174–1182, 2018.

15 Eric S. Lander and Michael S. Waterman. Genomic mapping by fingerprinting random clones: A mathematical analysis. *Genomics*, 2(3):231–239, 1988.

16 Daniel Mapleson, Gonzalo Garcia Accinelli, George Kettleborough, Jonathan Wright, and Bernardo J Clavijo. KAT: a K-mer analysis toolkit to quality control NGS datasets and genome assemblies. *Bioinformatics*, 33(4):574–576, November 2016.

17 Guillaume Marçais and Carl Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 2011.

18 Eric Marinier, Daniel G. Brown, and Brendan J. McConkey. Pollux: platform independent error correction of single and mixed genomes. *BMC Bioinformatics*, 16(1):10, 2015.

19 E. W. Myers. FastK. https://github.com/thegenemyers/FASTK, Accessed on 24/06/2022.

20 E. W. Myers. HIsim. https://github.com/thegenemyers/HI.SIM, Accessed on 24/06/2022.

21 Sergey Nurk *et al.* The complete sequence of a human genome. *Science*, 376(6588):44–53, 2022. doi:10.1126/science.abj6987.

22 Sergey Nurk *et al.* HiCanu: accurate assembly of segmental duplications, satellites, and allelic variants from high-fidelity long reads. *Genome Research*, 30(9):1291–1305, 2020.

23 Pavel A. Pevzner, Haixu Tang, and Michael S. Waterman. An Eulerian path approach to DNA fragment assembly. *Proceedings of the National Academy of Sciences*, 98(17):9748–9753, 2001.

24 Nicolas Philippe, Mikaël Salson, Thérèse Commes, and Eric Rivals. CRAC: an integrated approach to the analysis of RNA-seq reads. *Genome Biology*, 14(3):R30, 2013.

25 T. Rhyker Ranallo-Benavidez, Kamil S. Jaron, and Michael C. Schatz. GenomeScope 2.0 and Smudgeplot for reference-free profiling of polyploid genomes. *Nature Communications*, 11(1):1432, 2020.

26 Arang Rhie *et al.* Towards complete and error-free genome assemblies of all vertebrate species. *Nature*, 592(7856):737–746, 2021.

27 Arang Rhie, Brian P. Walenz, Sergey Koren, and Adam M. Phillippy. Merqury: reference-free quality, completeness, and phasing assessment for genome assemblies. *Genome Biology*, 21(1):245, 2020.

28 Roy Ronen, Christina Boucher, Hamidreza Chitsaz, and Pavel Pevzner. SEQuel: improving the accuracy of genome assemblies. *Bioinformatics*, 28(12):i188–i196, 2012.

29 Jared T. Simpson. Exploring genome characteristics and sequence quality without a reference. *Bioinformatics*, 30(9):1228–1235, 2014.

30 The Darwin Tree of Life Project Consortium. Sequence locally, think globally: The Darwin Tree of Life Project. *Proceedings of the National Academy of Sciences*, 119(4):e2115642118, 2022.

31 German Tischler and Eugene W. Myers. Non hybrid long read consensus using local de Bruijn graph assembly. *bioRxiv*, 2017. doi:10.1101/106252.

32 Brian Walenz et al. Meryl. https://github.com/marbl/meryl, Accessed on 24/06/2022.

33 Ting Wang *et al.* The Human Pangenome Project: a global resource to map genomic diversity. *Nature*, 604(7906):437–446, 2022.

34 Aaron M. Wenger *et al.* Accurate circular consensus long-read sequencing improves variant detection and assembly of a human genome. *Nature Biotechnology*, 37(10):1155–1162, 2019.

35 Ryan R. Wick. Badread: simulation of error-prone long reads. *Journal of Open Source Software*, 4(36):1316, 2019.

36 Xiaohong Zhao *et al.* EDAR: An efficient error detection and removal algorithm for next generation sequencing data. *Journal of Computational Biology*, 17(11):1549–1560, 2010.

37 Justin M. Zook *et al.* Extensive sequencing of seven human genomes to characterize benchmark reference materials. *Scientific Data*, 3(1):160025, 2016.

A Supplementary Figures

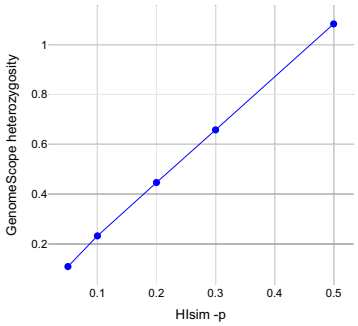

Figure S1 Relationship between HIsim's heterozygosity parameter (`-pX,X` is specified for a value of the x-axis) and the heterozygosity estimated by GenomeScope from the generated fruit fly simulation dataset with 50× coverage.

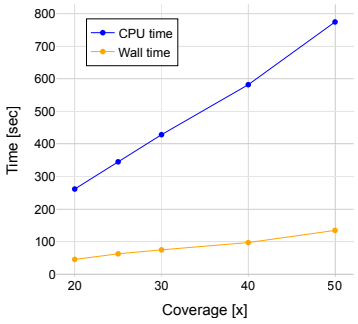

Figure S2 Average computation time of ClassPro for the fruit fly simulation datasets using 8 threads (`-T8` option). Both CPU time (which is the sum of user CPU time and system CPU time) and wall clock time scale linearly with respect to the coverage of the dataset, i.e. the number of k-mers in the dataset, with a consistent speed up of about 6× in wall clock time by the parallelization.

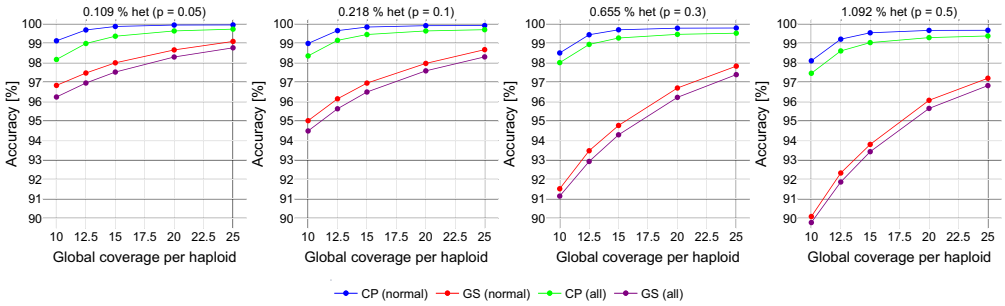

Figure S3 Average overall accuracy between ClassPro and GenomeScope across normal reads without repetitive k-mers and across all reads in the fruit fly simulation datasets by HIsim.

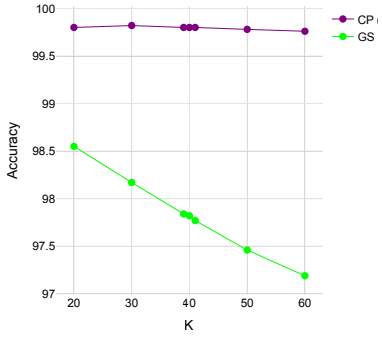

Figure S4 Average overall accuracy with various values of k (=20,30,39,40,41,50,60).

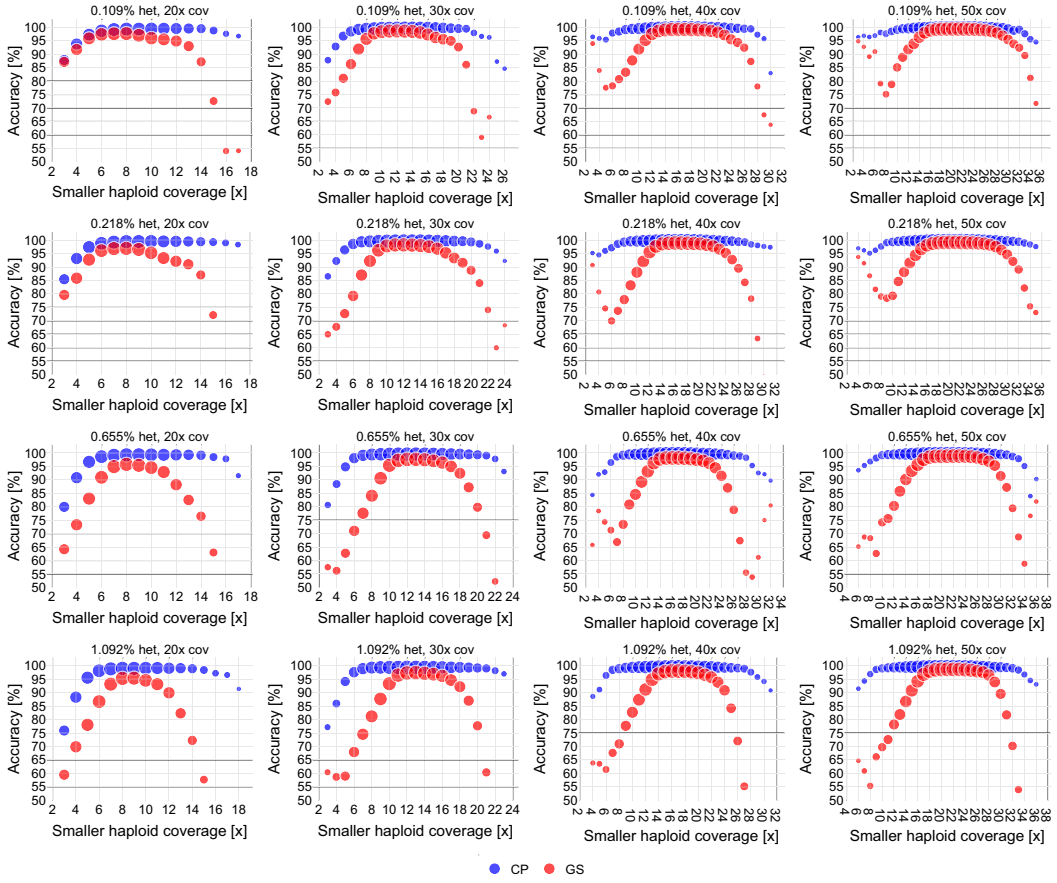

Figure S5 Relationship between local coverage and average accuracy at the resolution of 2Kbp windows across all reads in the fruit fly simulation datasets by Badread.

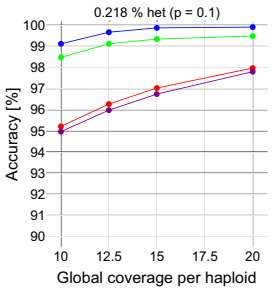

Figure S6 Average overall accuracy between ClassPro and GenomeScope across normal reads without repetitive k-mers and across all reads in the human simulation datasets generated by HIsim using a typical heterozygosity of $\sim 0.2\%$.