HammerIT: homopolymer-space Hamming clustering for IonTorrent read error correction

Anton Korobeynikov, Artem Tarasov
St. Petersburg State University, Russia

Introduction

FILLME

Notation

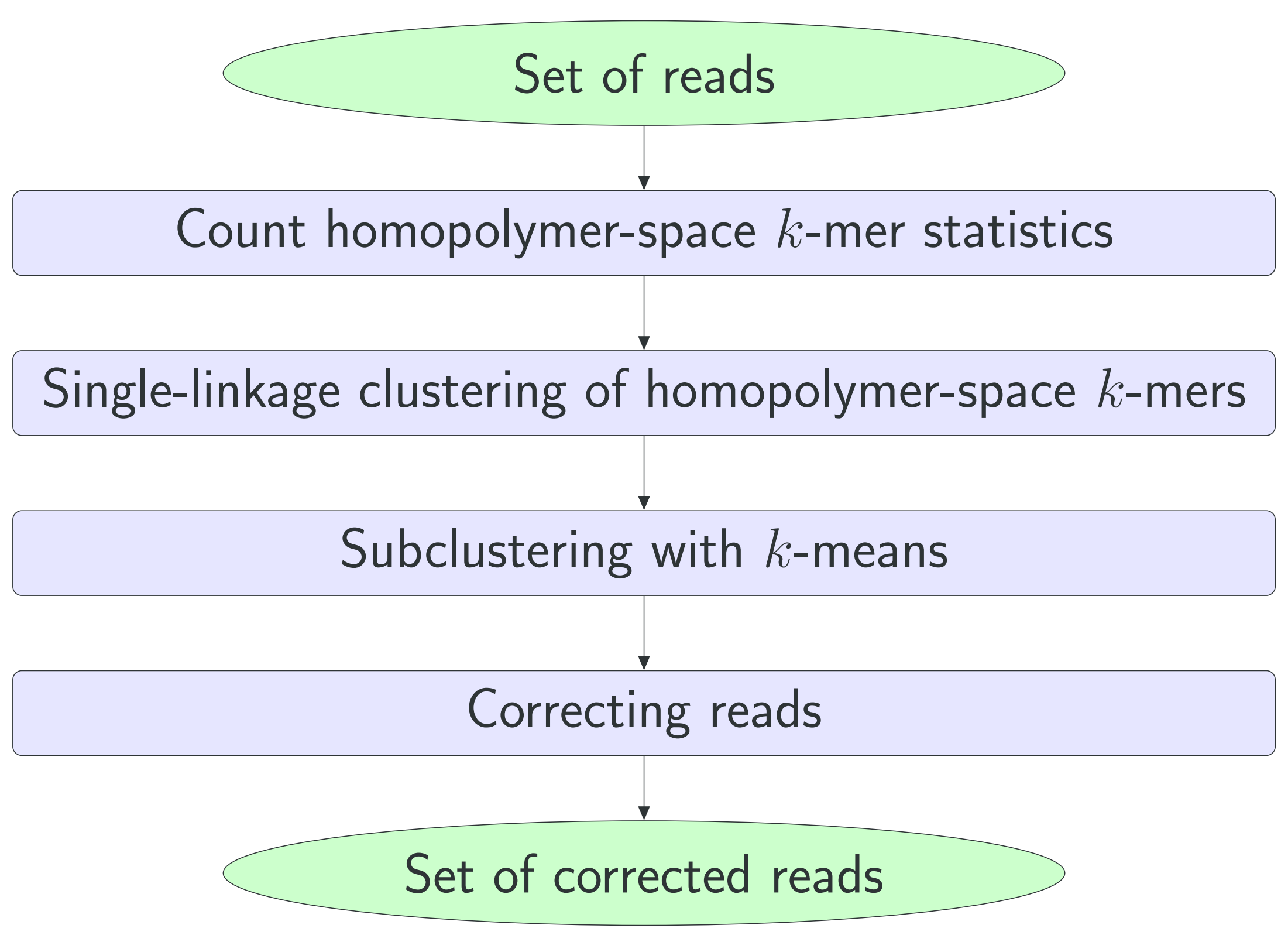
Let $\mathcal{N}$ denote the nucleotide alphabet $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$.
By definition, put $\mathcal{H} = \mathcal{N} \times \mathbb{N}^{+}$.

We call an element of the alphabet $\mathcal{H}$ a *homopolymer run*, and an element of $\mathcal{H}^k$ a *homopolymer-space k-mer*.

We use $x[k]$ to denote k-th element of a sequence, using zero-based indexing; $x[k\ ..\ l]$ to denote a subsequence $x[k]x[k+1]\ldots x[l]$; finally, length of x is denoted by $|x|$.

Distance between $i,\ j \in \mathcal{H}^k$ is defined as the minimum number of 1-base insertions/deletions/mismatches needed to align common part of i and j. It is denoted by $\mathrm{dist}(i, j)$.

HammerIT workflow


Set of reads
Count homopolymer-space k-mer statistics
Single-linkage clustering of homopolymer-space k-mers
Subclustering with k-means
Correcting reads
Set of corrected reads

IonTorrent error profile

Corrected flow signal intensities are available in BAM files produced by versions of Ion Torrent Suite prior to 3.4. Called homopolymer length is obtained as corrected flow signal intensity rounded to the nearest integer.

We have studied flow signal intensity distributions around insertion/deletion sites. File B7-295.bam, downloaded from Ion Community website, contained 4.6M insertions, 5.0M deletions, and 1.5M mismatches.

Overwhelming majority of errors turned out to be insertions/deletions of length 1, occuring when flow signal intensity is approximately halfway between two adjacent integers.

Figure 1: Flow signal intensities at insertion sites

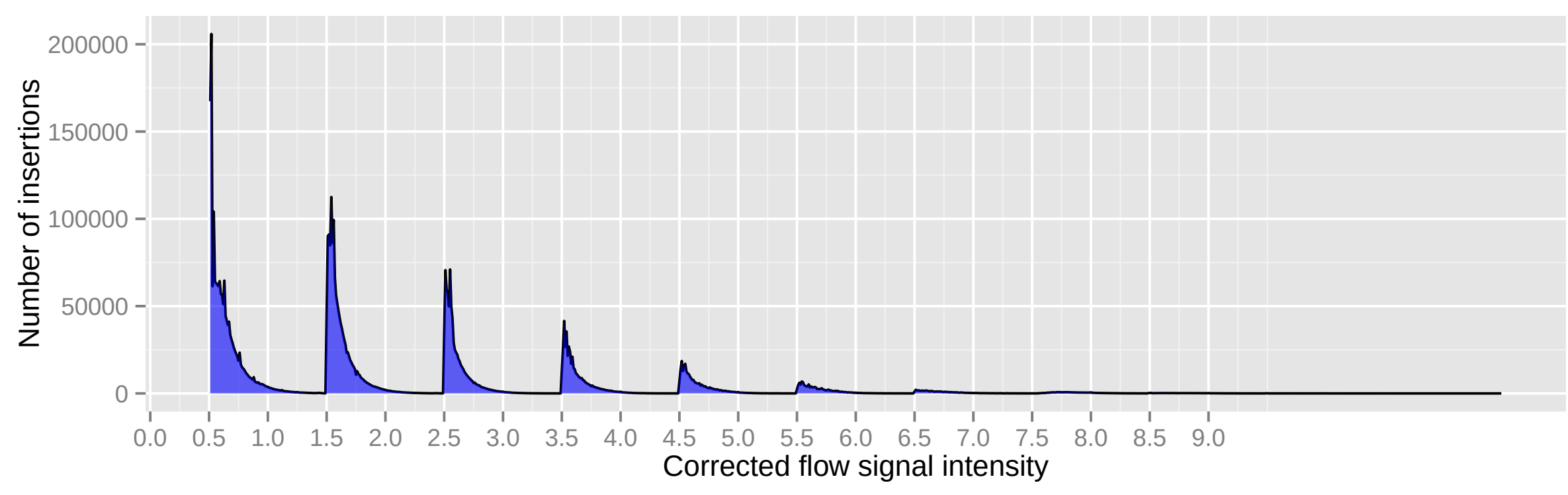
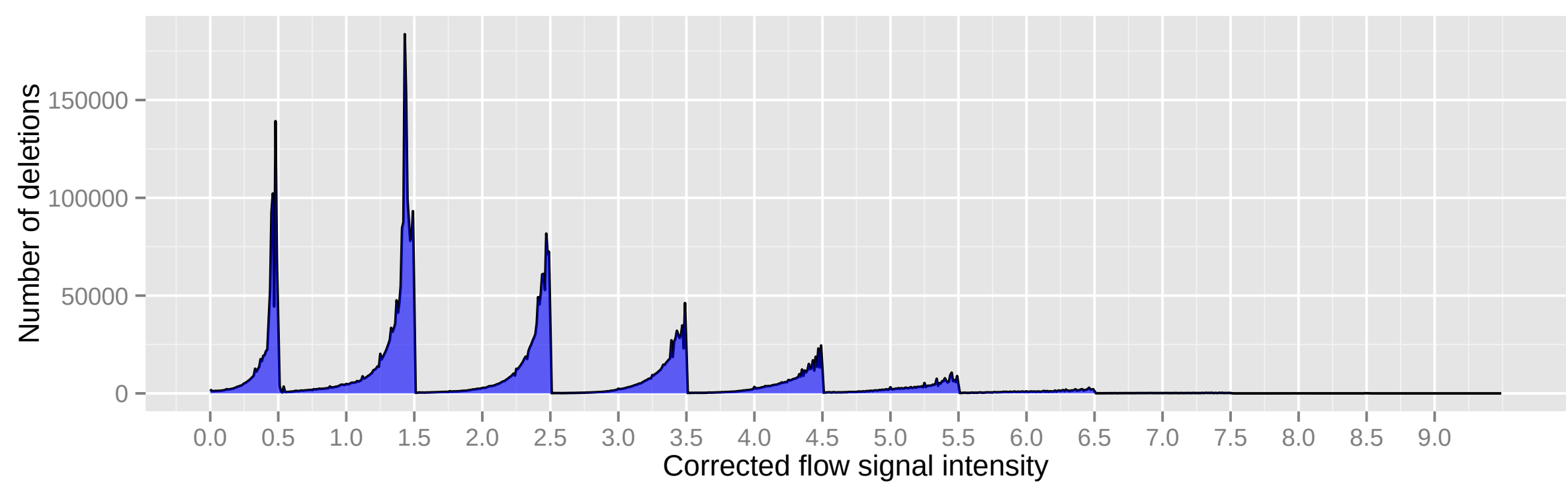


Figure 2: Flow signal intensities at deletion sites



More detailed analysis of errors in IonTorrent data can be found in the article "Shining a Light on Dark Sequencing: Characterising Errors in Ion Torrent PGM Data" *(PLoS Comput Biol 9(4))*

Error reduction results

We evaluated HammerIT on 6 publicly available datasets, using the same pipeline as the authors of the recently published article "Updating benchtop sequencing performance comparison" *(Nature Biotechnology, v. 31, no. 4)*. In that article, error rate in four Ion Torrent datasets has been assessed. We used the same data plus two extra datasets from 314v2 chip, which recently became available on Ion Community Portal.

Indel/mismatch error rates were calculated for uniquely mapped reads before and after correction. For each dataset, correction was done in two ways. In the first setup, trimming was done for read ends that couldn't be corrected due to lack of good k-mers, while in the second one such read ends were preserved in the output. Relative change in read coverage after correction stayed within 0.4% in all cases.

Figure 3: Error rates before and after correction

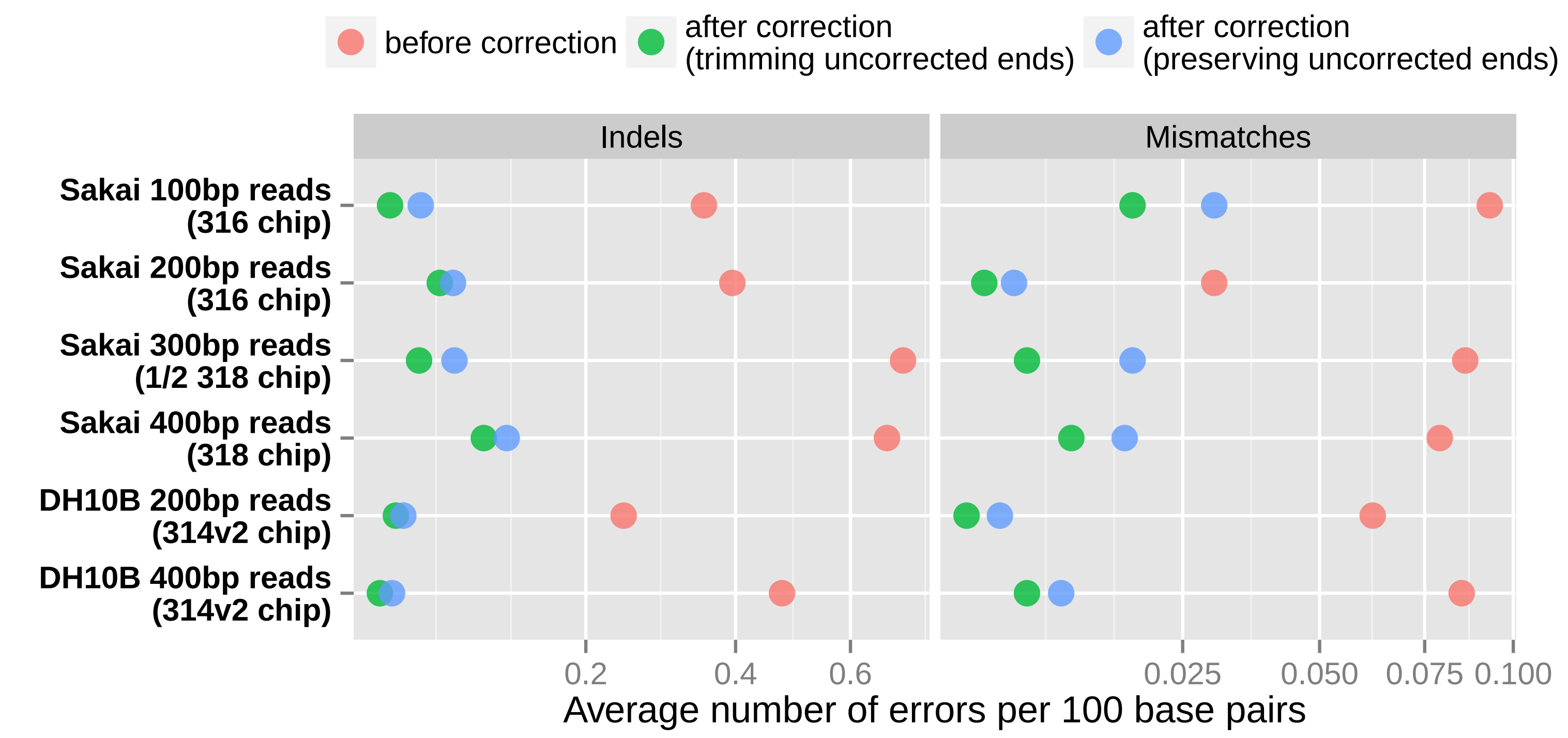
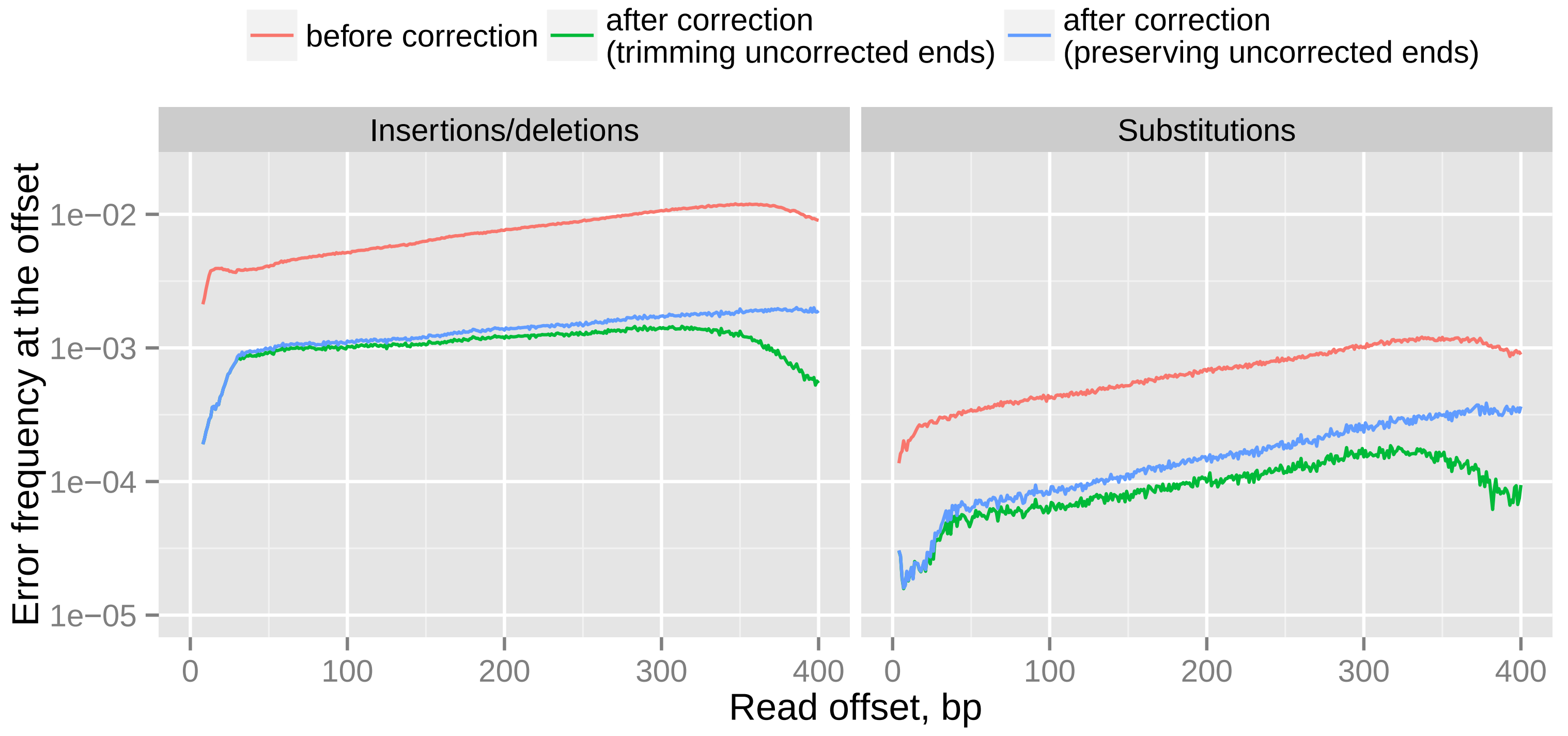


Figure 4: Error reduction by read position for Sakai 400bp reads



Pairwise distance calculation

We use 5-base lookahead to compute distance between k-mers in homopolymer-space. Helper table stores precomputed values of

$$H_k : \mathcal{N}^k \times \mathcal{N}^k \to \{\mathrm{Insertion}, \mathrm{Deletion}, \mathrm{Mismatch}\}\,,\quad k = 1, 2, 3, 4, 5\,.$$

The chosen value of 5 is a trade-off between accuracy and speed.

(Mapped reads from B7-295.bam dataset were used for training.)

G A G T A C A C T G T C G T C G
G T G T A C A T G T C G A T G C

$$H_5(\mathrm{AGTAC}, \mathrm{TGTAC}) = \mathrm{Mismatch}$$
$$H_5(\mathrm{CTGTC}, \mathrm{TGTCG}) = \mathrm{Deletion}$$
$$H_3(\mathrm{TCG}, \mathrm{ATG}) = \mathrm{Mismatch}$$
$$H_2(\mathrm{CG}, \mathrm{TG}) = \mathrm{Mismatch}$$

Algorithm

Input: $x, y \in \mathcal{N}^{+}$ — homopolymer-space k-mers in nucleotide alphabet
Output: $dist$ — approximate distance between x and y

```
$pos.x \leftarrow 0$; $pos.y \leftarrow 0$; $dist \leftarrow 0$;
while $pos.x < |x|$ and $pos.y < |y|$ do
    if $x[pos.x] = y[pos.y]$ then
        $pos.x \leftarrow pos.x + 1$; $pos.y \leftarrow pos.y + 1$;
    else
        $k \leftarrow \min(5, |x| - pos.x, |y| - pos.y)$;
        adjust $pos.x$ and $pos.y$ according to
            $H_k(x[pos.x\ ..\ pos.x + k - 1], y[pos.y\ ..\ pos.y + k - 1])$;
        $dist \leftarrow dist + 1$;
```

Homopolymer-space k-mer clustering

The starting point is single-linkage clustering of homopolymer-space k-mers. Two homopolymer-space k-mers belong to the same cluster if the distance between their nucleotide representations does not exceed one.

We reduce quadratic time requirements of the naive algorithm by noticing that if distance between two k-mers is less or equal to one, they share a common substring of length at least $\lfloor k/2 \rfloor$. This allows us to group k-mers into smaller blocks sharing a substring, and then use the quadratic algorithm for each block.

In order to detect all pairs of connected k-mers with such grouping, ranges $(0\ ..\ \lfloor k/2 \rfloor - 1)$, $(1\ ..\ \lfloor k/2 \rfloor)$, $\ldots$, $(\lfloor (k+1)/2 \rfloor\ ..\ k-1)$ are to be examined. Despite of this, we use only the first and the last of the ranges, to speed up the clustering step. This increases the number of *singletons* — clusters consisting of only one k-mer, usually erroneous — but has little impact on correction performance because of adjacent read k-mers making much larger contribution into the consensus scores (the number of singleton occurrences is usually low).

$$\mathit{partition}(\mathcal{K},\ i,\ j) = \left\{ B_s : \bigcup_{s \in \mathcal{H}^{j-i+1}} B_s = \mathcal{K},\ \forall x \in B_s\ x[i\ ..\ j] = s \right\}$$

Algorithm

```
$kmers \leftarrow$ homopolymer-space $k$-mers seen in the data
$components \leftarrow \{\{k\} : k \in kmers\}$
$blocksL \leftarrow partition(kmers,\ 0,\ \lfloor k/2 \rfloor - 1)$
$blocksR \leftarrow partition(kmers,\ \lfloor (k+1)/2 \rfloor,\ k - 1)$
for each $block$ in $blocksL$, $blocksR$ do
    for each $i \in block$ do
        for each $j \in block$ do
            if $\mathrm{dist}(i, j) \le 1$ then
                join the components to which $i$ and $j$ belong
```

Subclustering

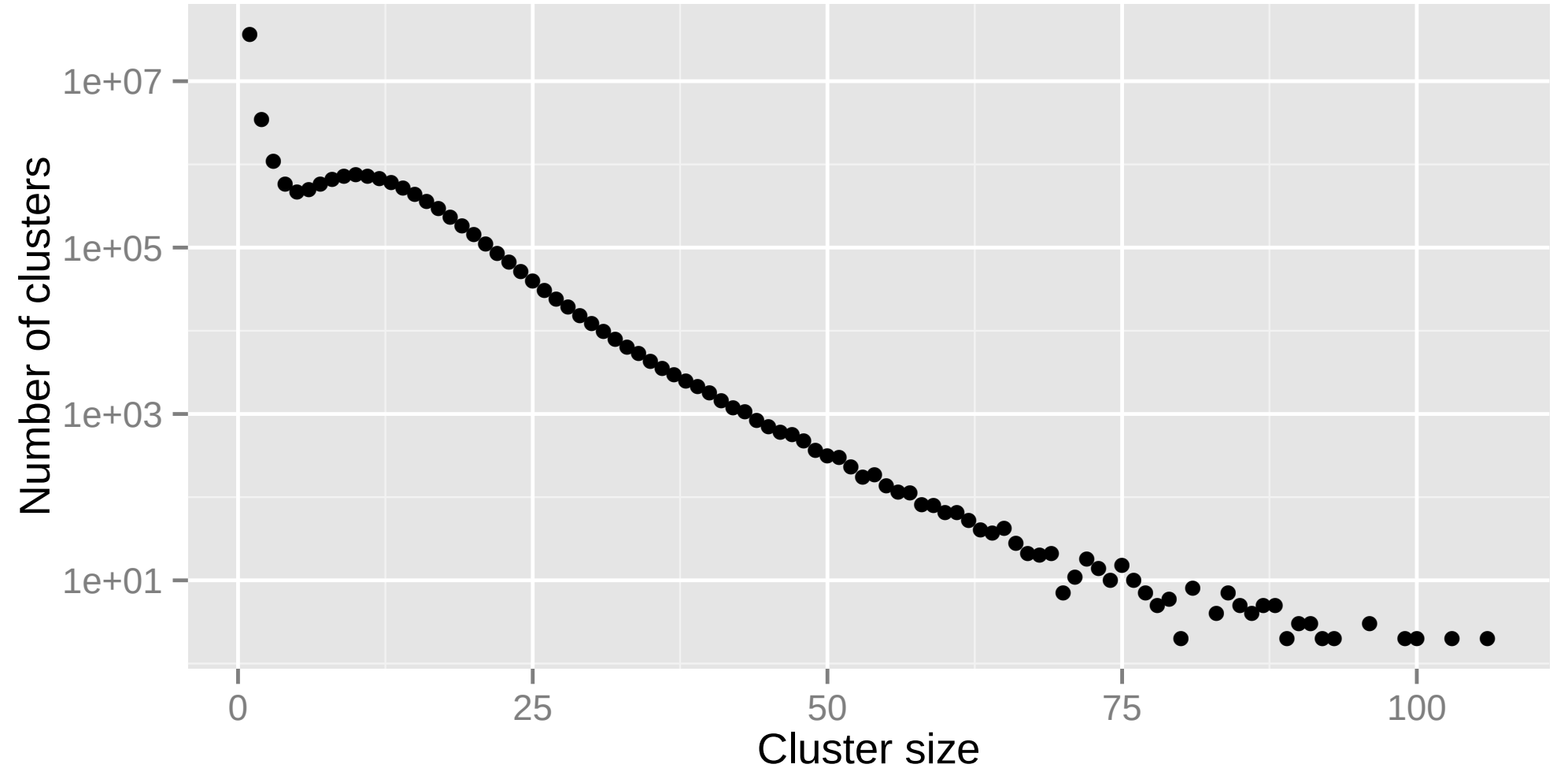
A cluster obtained from the initial process may contain $m \geq 2$ homopolymer-space k-mers from the genome. In this case, we split the cluster into m subclusters by running k-means algorithm on it.

The subtle question is how to determine m. Currently we just set it to be the number of the cluster elements with quality within machine epsilon of 1, where *quality* of $x \in \mathcal{H}^k$ is defined as $\Pr(x \text{ is genomic})$. For the set of reads $\mathcal{R} \subset \mathcal{N}^{+}$ the quality of $x \in \mathcal{H}^k$ is computed as

$$\prod_{r \in \mathcal{R}} \prod_{\substack{0 \le m \le |r| - |x'|,\\ r[m\ ..\ m + |x'| - 1] = x'}} \prod_{m \le n < m + |x'|} \Pr(r[n] \text{ is correct})\,,$$

where $x' \in \mathcal{N}^{+}$ is the sequence of nucleotides in x.

Cluster size distribution



Typical cluster

homopolymer-space 16-mer	n	qual.
CCCGTTTGT-GCGCCC-GG--CATTT-GAT	562	1.00
CCCGTTTGT-GCGCCC-GG--CATT--GAT	13	1.00
CCCGTT-GT-GCGCCC-GG--CATTT-GAT	10	1.00
CCCGTTTGT-GCGCCCCGG--CATTT-GAT	6	1.00
CCCGTTTGT-GCGCCC-GG--CATTTTGAT	5	0.99
CCGTTTGT-GCGCCC-GG--CATTT-GAT	4	1.00
CCCCGTTTGT-GCGCCC-GG--CATTT-GAT	4	0.98
CGTTTGT-GCGCCC-GG--CATTT-GAT	3	1.00
CCCGTTTGT-GCGCC--GG--CATTT-GAT	3	0.98
CCCGTTTGT-GCGCCC-GG--CATTT-GAAT	3	0.98
CCCGTTTGT-GCGCCC-GGA-CATTT-GA	3	0.98
CCCGTTTGT-GCGCCC-GG--C-TTT-GATG	3	0.97
CCCGTT-GTTGCGCCC-GG--CATTT-GAT	2	0.94
CCCGTTTGTTGCGCCC-GG--CATTT-GAT	2	0.94
CCCGTTTGGTGCGCCC-GG--CATTT-GAT	2	0.93
CCCGTTTGT-GCGCCC-GG--CATTT-GAG	2	0.89
CCCGTTTGT-GCGCCC-GG--CATTTGGAT	2	0.74
CCCGTTTGT-GCGCTC-GG--CATTT-G	1	0.97
CCCGTTTGTG-CGCCC-GG--CACTTTGA	1	0.87
CCCGTTTGTG-CGCCC-GG-CCATTT-GAT	1	0.85
CCCGTTTGTGC-GCCC-G---CATTT-GAT	1	0.83
CCCGTTTGTGC-GCCC-GG--CATTT-GGTG	1	0.74
CCGTTTGTGCCGCCC-GG--CATTT-GAT	1	0.71
CCCGTTTTGTGCGCCC-GG--CATTT-GAT	1	0.69
CCCGTTTGT-GCGCCC-GGTACATTT-G	1	0.68
CCCGTTTGT-GCGCCC-GG--CATTTCGA	1	0.68
CCCGTTTGTGCCGCCC-GG--CATTT-GAT	1	0.66
CCGTT-GT-GCGCC--GG--CATTT-GAT	1	0.66
CCCGTTTGTG-CGCCC-GGG-CATTT-GAT	1	0.61
CCCGTTTGT-GCGCCC-GGACCATTT-GA	1	0.61
CCCGTTTGTGC-GCCC-GG--CATTTTGTA	1	0.58
CCCGTTTGTGGCGCCC-GG--CATTT-GAT	1	0.57
CCCGTTTGTGC-GCCC-GGA-CATTT-AG	1	0.56
CCCGTTTGT-GCGCCC-GG--CATTT-AGT	1	0.55
CCGTTTGTGC-GCC--GG--CATTT-GAT	1	0.54
CCCGTT-GTGC-GCCC-GGG-CATTT-GAT	1	0.44
CCCGTTTGTGCCGCCCTGG--CATTT-GA	1	0.43
CCCGTTTTGTGCGCCC-GG--CATTTTGAT	1	0.36
CCCGTTTGGTGCGCCCCGG--CATTT-GAT	1	0.22

Error correction algorithm

```
for each $read$ from the dataset do
    (Producing contiguous corrected parts of read)
    for each homopolymer-space $kmer$ from the read do
        $center \leftarrow$ center of the cluster to which $kmer$ belongs;
        if $center$ quality is more than user-specified threshold and
          $center$ bases agree with the previous "good" center then
            include $center$ into consensus score calculation;
        else
            yield new corrected part from current consensus;
            trim homopolymer runs with low consensus score from ends;
            reset consensus table and start a new part;
    (Combining corrected parts)
    while there are two or more parts do
        $curr \leftarrow$ first part; $next \leftarrow$ second part;
        align last 8 homopolymer runs of $curr$ against the read;
        align first 8 homopolymer runs of $next$ against the read;
        if there is a gap on the read between the two parts then
            copy read homopolymer runs as is;
        else
            select homopolymer runs with higher consensus score
                from the intersection of the two parts;
        replace $curr$ and $next$ with the combined part;
    (Optionally, attaching uncorrected end)
    align last 8 runs of the last chunk against the read sequence;
    append read homopolymer runs after the last aligned run.
```