Long Reads Alignment

Giulia Guidi[1,2], Aydın Buluç[2]
{gguidi, abuluc}@lbl.gov
[1]Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Milan, Italy
[2]Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, USA

July 1, 2017

1 Problem Statement and General Overview

High-throughput sequencing technologies produce a large number of short and low-quality DNA sequences, called *reads*. These data are used as starting point to reconstruct the whole DNA sequence in a process called *de novo* genome assembly.

Recent advances in this field, such as the Pacific Biosciences Single Molecule Real-Time (SMRT) Sequencing technology, led to higher consensus accuracy, unbiased error distribution and longer reads. These improvements are crucial to assemble high-quality DNA sequences. On the other hand, the SMRT technology is prone to an higher error rate with respect to previous sequencing technology, about 15%. In general, high error rates lead to a significant computational effort in throwing away useless data. The presented work proposes a novel approach to select significant data from the input and recognize overlapping reads pairs to reconstruct the whole DNA sequence.

To give a general overview of our idea, let's consider a pair of reads that share a region of length k (k-mer), as shown in Figure 1. The probability that such region is correclty sequenced on both the reads is approximately $(1-e)^{2\cdot k}$ (summation of Bernoulli trials), where e is the error rate. Then, the probability $P(1,L)$ that at least one k-mer out of L consecutive k-mers being correct in both the sequences is:

$$P(1,L) = 1 - (1-(1-e)^{2\cdot k})^{L}$$

Considering an overlapping region L of 300 base-pairs (bp), with $k = 17$ and $e = 0.15$, we obtain that $P(1,300)$ is greater than 70%. Starting from these consideration, we decided to base our approach on shared k-mers between reads. However, we have to face the issue derived from having a high error rate. As a matter of fact, an error rate equal to 15% implies that every possible k-mer will be seen at least once, given enough depth (d, sequencer parameter). Consequently, we would need a k-mer look-up table of size 4^{k}. Furthermore, we have to consider at least a k value equal to 17 as, choosing a smaller k, k-mers are not even unique in large genomes. Using $k = 15$, we can only encode 1 Gbp possibilities whereas, for instance, the *Human* genome is 3 Gbp. To address this issue and identify overlapping regions, we take into account just a reliable set of k-mers.

2 Reliable k-mers

The first computed analysis regards the identification of a reliable set of k-mers (RKS). Our algorithm relies on detecting overlaps between long reads efficiently. We treat the k-mer occurrences in each long read as the feature vector of that read. However, due to high error rates, the number of distinct k-mers

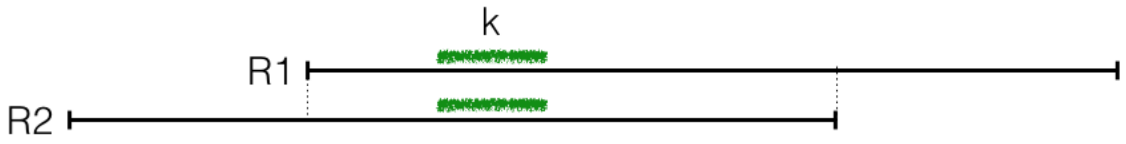


Figure 1: Pair of overlapping reads sharing a region of length k.

in a dataset can be orders of magnitude larger the actual correct k-mers. Keeping all the k-mers in our feature set would not only increase the computational costs and memory requirements, it would also lower our precision.

Ideally, we want k-mers that occur only once in the genome. Multiple occurences of the same k-mer in the genome correspond to repeat regions. If we kept non-unique k-mers in our feature set, they would increase the number of spurious alignments and hence increase the computational costs. Our rationale for ignoring non-unique k-mers comes from the observation that either (a) the repeated region is small enough compared to the length of the read that the unique part of the read can still be used to find overlaps of this read with other reads, or (b) that the repeated region is almost as long as the read itself, in which case there is no benefit in aligning this read to other reads because it does not increase our information about the final genome.

The histogram in Figure 2 shows the distribution of unique, non-genomic and repeated k-mers in the genome, given their frequency in the input set of reads. With *unique* k-mers we define k-mers that occur only once in the genome, while $non-genomic$ k-mers are those that not exist in the genome and *repeated* k-mers those that appear multiple times in the genome. To obtain the plot, we divide k-mers based on their occurrence in the reads and then, having the reference genome, for each occurrence we compute the amount of unique, non-genomic and repeated k-mers. We see by looking at the k-mer histogram that the majority of k-mers in the right tail either occur multiple times or do not occur in the genome at all. Furthermore, Figure 3 points out the percentage of unique k-mers over the total amount of k-mers belonging to a certain frequency; the number above each bar represents the absolute number of unique k-mers for a given frequency. Both Figure 2 and Figure 3 are obtained using the *Escherichia Coli* genome with $d = 30$, $k = 17$ and $e = 0.15$.

The probability of a k-mer being sequenced correctly is approximately $(1 - e)^k$ where e is the error rate. If the sequencing depth is d, then observing this k-mer in the input data d times is a very slim $(1 - e)^{dk}$. Our analysis here is only correct if no other distinct section of the genome has been morphed into this k-mer by error. We acknowledge that such morphing occurs in practice but the majority of the high-frequency k-mers in the input set are due to correct sequencing if the value of k is chosen appropriately. Take the $Human$ genome that is approximatyely 3 Gbp and for the sake of argument, only consider substitution errors. For $k = 17$, it encodes $4^{17} = 16$ billion different k-mers. Assuming independence, every possible k-mer exists in the genome with probability 3/16. The probability that a 17-mer we have seen being the result of off-by-1 error in sequencing is $(3/16) \cdot 17 \cdot e \cdot (1 - e)^{17-1} \approx 3e(1 - e)^{16}$, whereas the probability of sequencing a 17-mer correctly is $(1 - e)^{17}$.

The probability that observing a k-mer that corresponds to a unique (non-repetitive) region $d - 1$ times in the input would be approximately $P(d - 1) = d(1 - e)^{(d-1)k}(1 - (1 - e)^k)$. To generalize, the probability to observe $d - t$ times is:

$$P(d-t) = \binom{d}{t}(1-e)^{(d-t)k}(1-(1-e)^k)^t$$

Figure 4 shows the probability that a k-mer with input occurrence $d - t$ corresponds to a unique region in the genome given $d = 60$, $k = 17$ and $e = 0.15$.

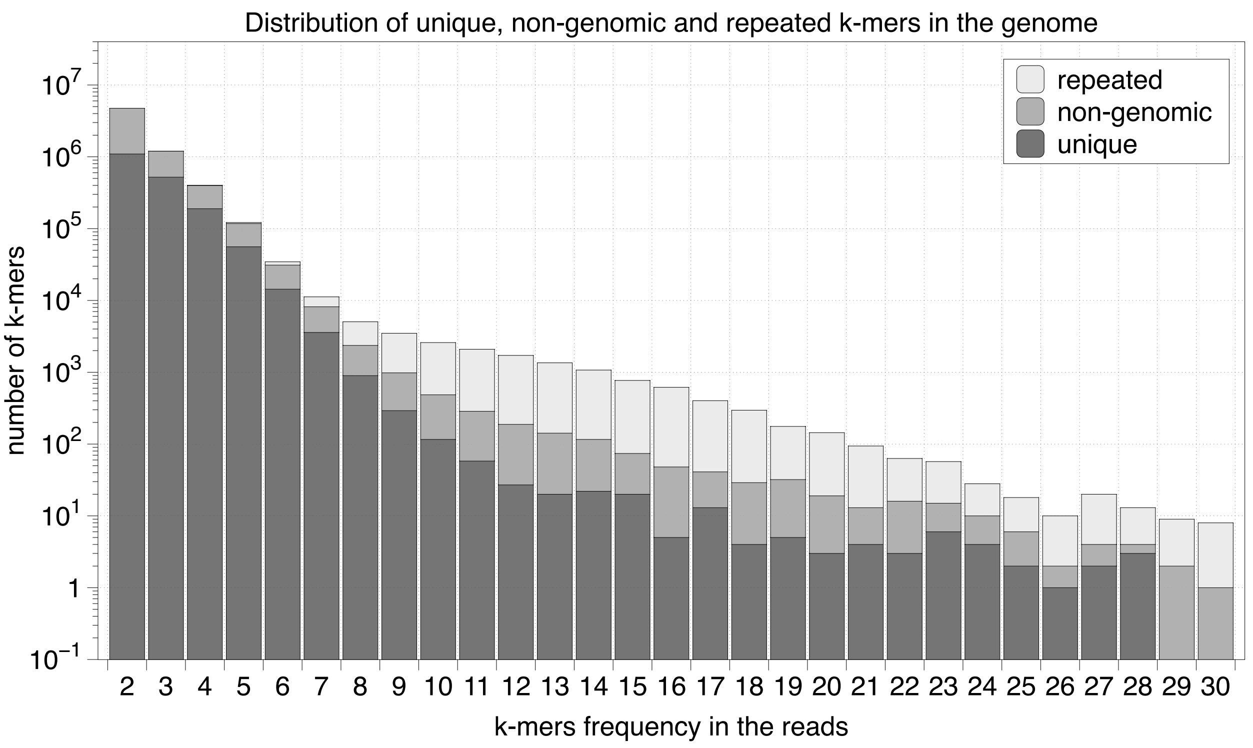


Figure 2: Distribution of unique, non-genomic and repeated k-mers over the total amount of k-mers belonging to a certain frequency.

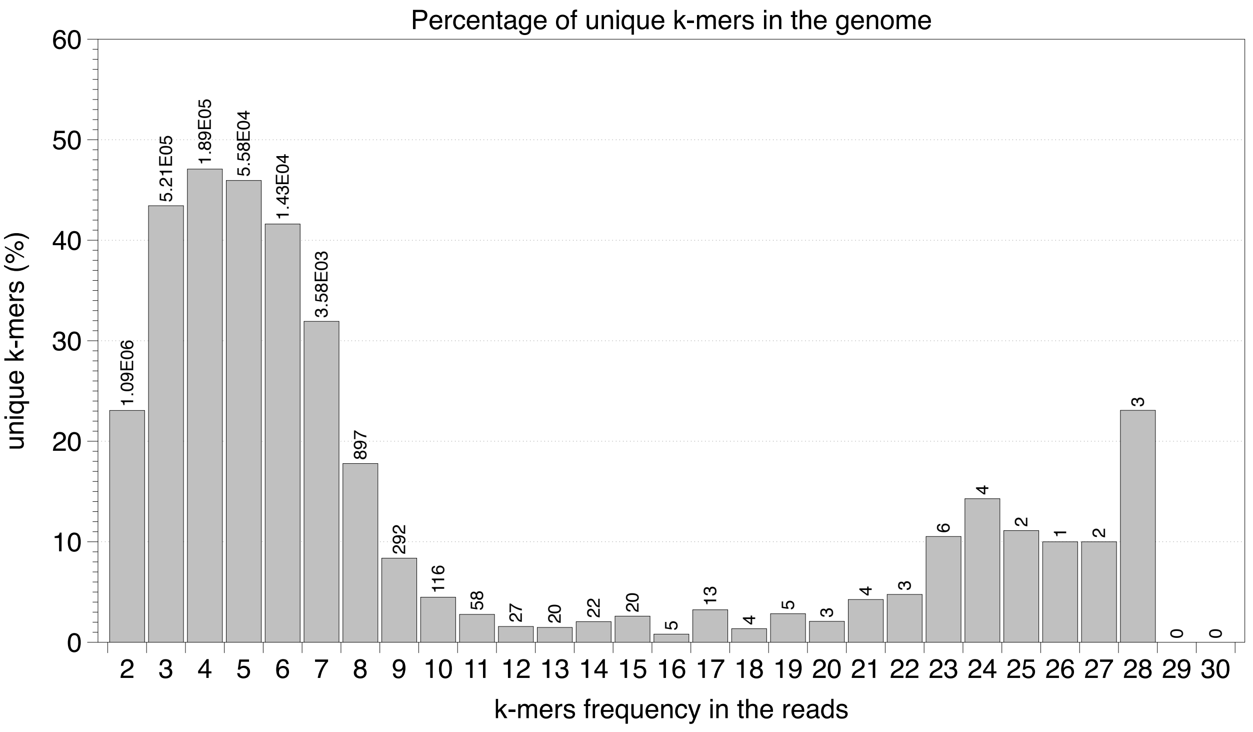


Figure 3: Percentage of unique k-mers over the total amount of k-mers belonging to a certain frequency. The number above each bar represents the absolute number of unique k-mers for a given frequency.

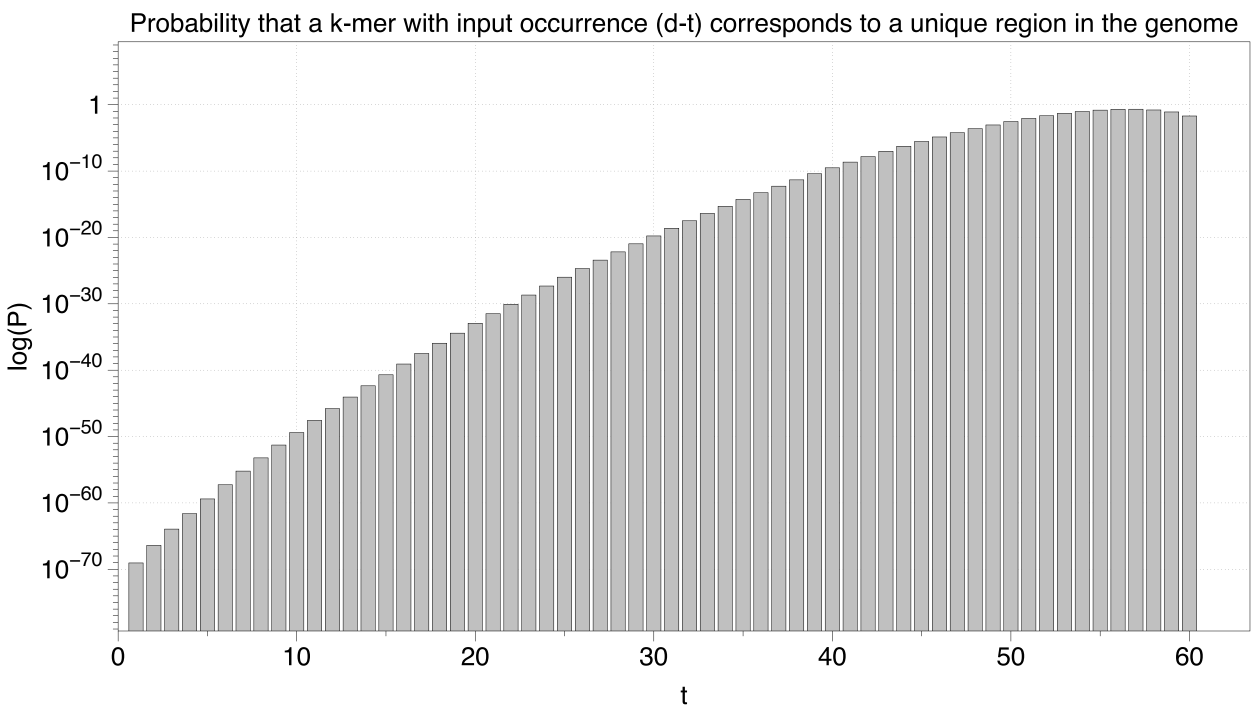


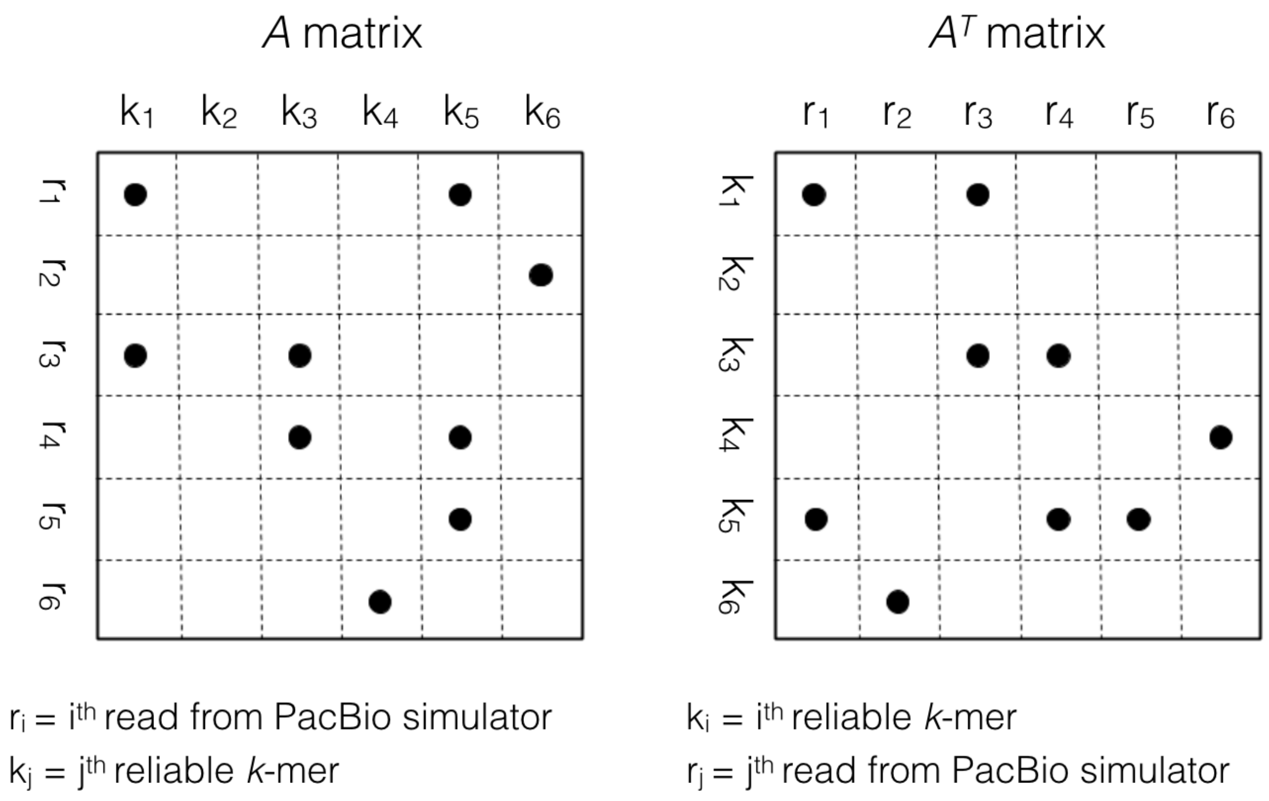
Figure 4: Probability that observing a k-mer that corresponds to a non-repetitive region $d-1$ times in the input.

3 Proposed algorithm

To identify the overlapping reads pairs, the proposed approach exploits sparse matrices. This because sparse matrices express the data access patterns in a concise and clear manner, allowing better organization of computation and generality. The column-by-column algorithm we use is equivalent to the k-mer index table concept in HipMer and many other assembler. The first step consists in the creation of a sparse matrix where row are represented by the reads and columns by the set of reliable k-mers. The cell (i, j) contains the k-mer id and the positions of that k-mer in the considered read. This matrix is used as starting point for the construction of a feature vector in finding alignments among the reads. Consequently, we create the transpose of the first one obtaining a $kmers-by-reads$ matrix, as shown in Figure 5.

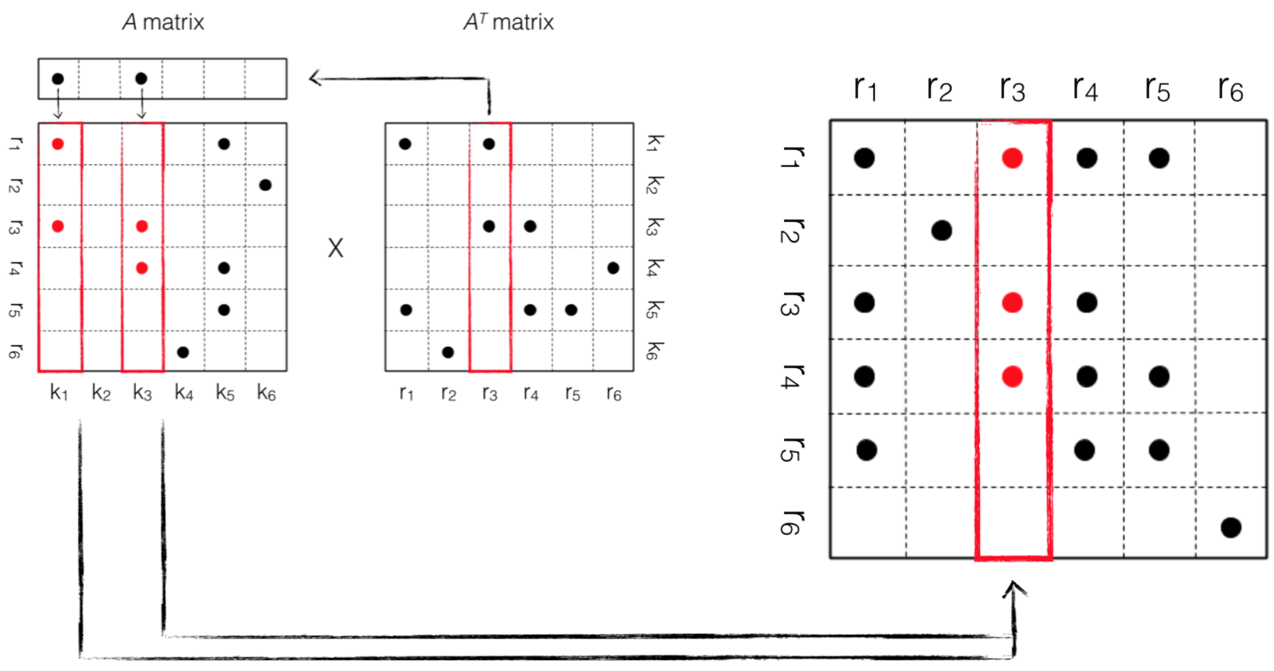
After that, exploiting the column-by-column algorithm we multiply these two sparse matrices obtaining a resulted matrix, as illustrated in Figure 6. The final sparse matrix allows us to keep track of all the read pairs that share some k-mers and the corresponding positions on both the reads. Here, the cell (i, j) consists in a vector containing the k-mers ids are and the corrisponding positions in the read i and read j. The generality of this approach comes from being able to use the same algorithm to also keep track of distances between shared k-mers. In fact, we are interested in keeping track of the distance between shared k-mers as it can provide more evidence of overlap.

To detect which reads pairs show some evidence of potential overlap, we implemented the following method, insert figure. For each read pairs, (a) we compute the distance between each pair of consecutive k-mers both on read i and read j. For each k-mers pairs, (b) we select the shortest distance between the one on read i and on read j and then, given the probabilities of insertion and deletion (that are given by the sequencer), (c) we compute the maximum length of the shortest one (dividing with $1-p_{deletion}$) and the minimum length of the longest one (dividing with $1+p_{insertion}$). If the former is greater than the latter, we consider that k-mers pair as evidence of potential overlap. In order to be as conservative



A matrix
A^T matrix
k_1 k_2 k_3 k_4 k_5 k_6
r_1 r_2 r_3 r_4 r_5 r_6
r_i = ith read from PacBio simulator
k_j = jth reliable *k*-mer
k_i = ith reliable *k*-mer
r_j = jth read from PacBio simulator

Figure 5: Sparse matrix $reads-by-kmers$ on the left and its transpose on the right.



A matrix
A^T matrix
X
k_1 k_2 k_3 k_4 k_5 k_6
r_1 r_2 r_3 r_4 r_5 r_6

Figure 6: Sparse matrix multiplication exploiting a $column-by-column$ algorithm cite.

as possible and be sure to remove just those reads pairs that certainly not overlap, we decide to keep into account all the reads pairs that present at least one evidence of potential overlap among their k-mers pairs.

Add references.

3.1 Light version

4 Evaluation