Haplotype and Repeat Separation in Long Reads

German Tischler[(1)]

(1) Myers Lab, Max Planck Institute of Molecular Cell Biology and Genetics,
Pfotenhauerstraße 108, Dresden, Germany, tischler@mpi-cbg.de

Keywords: long reads, repeats, haplotypes.

Abstract. Resolving the correct structure and succession of highly similar sequence stretches is one of the main open problems in genome assembly. For non haploid genomes this includes determining the sequences of the different haplotypes. For all but the smallest genomes it also involves separating different repeat instances. In this paper we discuss methods for resolving such problems in third generation long reads by classifying alignments between long reads according to whether they represent true or false read overlaps. The main problem in this context is the high error rate found in such reads, which greatly exceeds the amount of difference between the similar regions we want to separate. Our methods can separate read classes stemming from regions with as little as 1% difference.

1 Scientific Background

Third generation sequencing reads like those produced by Pacific BioSciences (PacBIO) and Oxford Nanopore Technologies (ONT) sequencers are very long in comparison with the ones produced by second generation sequencers. The average read length for PacBIO is often 10k base pairs (bp) and for ONT 7-9kbp have been reported. For PacBIO more than half of the sequenced bases can be in reads of length 20kbp and above. This is much longer than the reads produced by second generation sequencers, which often yield reads as short as 150bp, so third generation sequencers allow a much better repeat resolution for assembly because more repeats are spanned by single reads. This increased read length however comes at the price of a much higher average base error rate (about 13% for PacBIO and even higher for ONT). This poses major algorithmic challenges in the areas of sequence alignment, comparison and signal detection. Read versus read comparison (see e.g. [1]) operates at a correlation of 70% and less. This makes it very hard to detect small differences between regions reads were sampled from. Reads stemming from sufficiently similar regions in an underlying genome, like instances of a repeat or different haplotypes, will often align within the parameters used, as the difference between the two sources is small in comparison with the read error rate. Being able to segregate read alignments into classes according to whether or not an alignment between a read pair designates a real overlap in the underlying genome to the degree possible is however important for multiple applications like genome assembly and variant detection. In genome assembly for instance the quality of any consensus sequence produced rises and falls with the ability to select the correct reads as input (cf. [2]). Linking up different haplotypes during the assembly of a non haploid organism results in patchwork like output, in particular an assembly process yielding output contigs which are in this way not contained in the genome to be reconstructed. Haplotype assembly designates the problem of separating reads into haplotype classes by first mapping them to a given reference sequence and then splitting the reads into groups using the information obtained. Several papers have presented methods for haplotype assembly (or read phasing) in the diploid setting (cf. [3, 4, 5, 6, 7, 8]). Canu (see [9]) performs repeat separation using a sequence of error correction, residual error estimation and classification of the error corrected reads. The authors report being able to separate repeat instances with 3% difference and above.

2 Materials and Methods

In this paper we discuss methods for repeat and haplotype separation in long reads. We consider the setting of de novo assembly, in particular we do not presume or require the existence of a reference sequence or known variation sites. Instead of read to reference alignments we use read to read consensus alignments, i.e. we align reads to error corrected reads. In addition we do not limit our attention to a scenario requiring there to be at most two versions of a sequence, like it is the case for the haplotype assembly of diploid genomes.

We consider two basic principles for splitting a set of reads. The first one is based on the trivial observation that reads in the same class should agree on most positions, particularly including those for which a very rough analysis shows a potential for disagreement in the read set. As the reads we consider are not error free we cannot expect the reads inside one class to agree on all positions. This approach has it's merits when the number of versions a sequence appears in is low but as we will see below, it becomes unsuitable as the number of versions grows. The second principle is based on observing sets of reads (more or less) consistently disagreeing on certain positions. This scales to higher version numbers but is computationally much more expensive.

2.1 Preliminaries

Let $G = \{S_1, S_2, \ldots, S_k\}$ denote a genome containing sequences S_i for $i = 1, \ldots, k$, i.e. strings over the alphabet $\Sigma = \{A, C, G, T\}$. Further let $R = \{R_1, R_2, \ldots R_r\}$ be a set of reads sampled from G (randomly of the forward and reverse complement strand) such that the strings in R have length L on average and the error rate (errors per length on G) between the reads and the intervals on G they were drawn from is p_e on average. For PacBIO the length distribution in R would follow a log normal distribution with average length 10kb and p_e would be in the order of 0.13. We denote a local alignment between sequences U_i and U_j by a tuple $(i, j, ib, ie, jb, je, c)$ where ib and ie mark the start and end of the alignment on U_i, jb and je the start and end on U_j and c is a Boolean value marking whether U_j or the reverse complement of U_j was used ($c =$ true for reverse complement). In practice long reads often contain stretches of very low quality, so even for two reads sharing a true overlap we find a sequence of local alignments instead of a single suffix/prefix or containment type alignment. Our methods can easily be generalised to this case, however for the sake of simplicity of exposition we assume that alignments between reads are contiguous below. Let A denote the set of all (local) alignments between pairs of reads in R s.t. the correlation between the two reads inside the alignment is at least $1 - 2p_e$ and the alignment covers at least ℓ bases on both reads involved for some length ℓ. In practice we commonly use $\ell = 1k$ for third generation long reads. For a given read R_k we call the subset of A s.t. the first component of the tuples is k the alignment pile for R_k. If G represents a non haploid genome or contains sufficiently long repeating regions then not all the alignments in A may refer to true read overlaps on G. We can use an alignment pile of a read or a subset thereof (for instance by choosing the top k best aligning other reads for some k) to compute a preliminary consensus or error corrected version of the read, e.g. using the algorithm proposed in [2]. We denote a preliminary consensus obtained for a read R_i in this way by $\hat{R}_i$. The alignment pile for R_i can be transformed into an alignment pile for $\hat{R}_i$ by aligning the reads in the pile for R_i to $\hat{R}_i$ while taking the positions of the original alignments on R_i into account and transforming those to positions on $\hat{R}_i$ using an alignment of R_i and $\hat{R}_i$. In addition to the original alignments in the pile of R_i we also insert an alignment between $\hat{R}_i$ and R_i into the pile of $\hat{R}_i$. An alignment pile for some $\hat{R}_k$ can be transformed into a matrix where the columns represent positions on or

(100,0)	(101,0)	(102,-1)	(102,0)
A	C	A	T
A	C		
A	T	A	T
	C	−	G

Table 1: Excerpt from a matrix given by an alignment pile

before the bases of $\hat{R}_k$ (before for base insertions into $\hat{R}_k$) and the rows are the reads in the alignment pile for $\hat{R}_k$. An alignment $(k, j, kb, ke, jb, je, c)$ between $\hat{R}_k$ and a read j is active from base kb to ke on k. The cells of a matrix row corresponding to read R_j are set as follows. Columns the respective alignment is inactive on remain empty. In the active region of the alignment a cell is filled with the base from R_j if the alignment features a match, mismatch or insertion operation for the respective position and a dash $(-)$ otherwise. As a convention we always have the alignment between $\hat{R}_k$ and R_k as the first row of the matrix. Table 1 shows an example. The excerpt shows positions 100 to 102 on the read. Some bases have been inserted before position 102, which is marked by the position identifier $(102, -1)$. The alignment corresponding to the second row ends at position 101, the one for the last row starts at position 101.

2.2 Agreement Based Splitting

Let d denote the average sequencing depth of the read set R. We assume the arrival rate of reads on the genome follows a Poisson distribution with mean d, i.e. we have a probability of $P_d(i) = \frac{d^i}{i!}e^{-d}$ to see a depth of i at a given position. The probability to see d' correctly sequenced bases for any position is thus

$$P_c(d') = \sum_{i=d'}^{\infty} P_d(i) \binom{i}{d'} (1 - p_e)^{d'} p_e^{i-d'} \tag{1}$$

We want to detect variation sites inside a given read R_i. One very simple way to do this is to scan the matrix constructed for $\hat{R}_i$ for columns in which more than one symbol appears with a frequency above a given threshold. Assuming the alignments used to construct the matrix are suitable we would see such a variation with probability $\sum_{j=d'}^{\infty} P_c(j)$ if we chose a threshold of d'. For $d = 20$ and $p_e = 0.15$ we obtain $d' = 8$ if we ask the probability to be at least 99%, i.e. we are 99% sure not to miss a relevant site if we look for columns containing at least two symbols with 8 or more instances. There is however the chance of calling variation sites because of unsuitable alignments in the pile for $\hat{R}_i$ or a sufficiently high number of wrongly sequenced bases (this is a problem especially in the presence of a high number of sequence versions as this increases the total number of reads involved in the pile). Consider a given position q in the genome G and two reads R_i and R_j covering this position. Then we have a probability of $(1 - p_e)^2$ for having the base at position q sequenced correctly in both R_i and R_j. Let $A_{ij} = (i, j, ib, ie, jb, je, c)$ denote an alignment between read R_i and R_j and assume we called n variants on R_i inside the index interval $[ib, ie]$. If R_i and R_j overlap as designated by A_{ij} in the underlying genome, then we have a probability of

$$P_s(j) = \sum_{j'=j}^{n} \binom{n}{j'} (1 - p_e)^{2j'} (1 - (1 - p_e)^2)^{n-j'} \tag{2}$$

to see R_i and R_j agree on at least j of the n disagreement points in the matrix for the alignment pile of $\hat{R}_i$. If there are two underlying versions, e.g. a repeat with two copies or haplotypes in a diploid genome, then we would expect to see reads coming from different versions disagree on most of the variant locations. In this case we have a strong signal for separating the two versions. It becomes weaker in the presence of more versions when some of the versions agree with others in a large fraction of the variant locations. In this case we cannot reliably tell the difference between two reads stemming from different versions with a relatively low number of sequencing errors and two reads stemming from the same version but agreeing on a lower number of variant locations due to a higher number of sequencing errors. For experiments we choose the number m of disagreement points two reads need to agree on so we consider them as from the same class as the smallest number s.t. $P_s(m) \geq 0.995$.

2.3 Disagreement Based Splitting

One of the main problems with agreement based splitting is suboptimal performance when reads from different classes agree on a large number of the detected variant locations. Splitting based on the differences between genomic regions does not suffer from this effect. Every attempt via directly comparing two long reads is however bound to fail as the high sequencing error rate drowns any slight difference between the two underlying real sequences. At a single base error rate of $p_e = 15\%$ the probability to see a correct pair of corresponding bases in two reads is $(1 - p_e)^2 = 72.25\%$, i.e. 27.75% of the pairs are wrong and most of these wrong pairs lead to a false disagreement between reads which should agree. When we compare bases for discovering disagreements between reads, we need to make reasonably sure that the bases compared are correct representations of their class for a given position. Consider some position on k reads stemming from the same class. Then we have a probability of $1 - p_e^k$ to see the correct base in at least one of these k reads. For $k = 2$ and $p_e = 0.15$ we have a probability of 97.75%, still a probability of more than 2% for all the bases to be wrong, for $k = 3$ we reach 99.6625%. In consequence, if three reads from the same class agree on a base, then this is most likely a correctly reported base. We use this observation by instead of comparing single read bases to single read bases comparing three tuples of bases to three tuples of bases. Given a read R_j we first build the matrix corresponding to the alignment pile of $\hat{R}_j$. We then scan the matrix column for column. In each column c we extract all 6 tuples $(r_1, r_2, r_3, r_4, r_5, r_6)$ s.t. r_i for $1, 2, \ldots, 6$ are row identifiers marking non empty cells in column c, the cells for row r_1, r_2 and r_3 all contain the same symbol a, the cells for row r_4, r_5 and r_6 all contain the same symbol b, $a \neq b$, $1 = r_1 < r_2 < r_3$ and $r_4 < r_5 < r_6$. Remember row 1 in the matrix refers to the alignment between $\hat{R}_j$ and R_j. There are $O(\binom{q}{2}\binom{q}{3}) = O(q^5)$ distinct such tuples in the worst case if q is the maximum number of active alignments in any column of the matrix. For each distinct tuple T we count the number $Y(T)$ of times it appears summed up over all columns. The support $Z(T)$ of a tuple T is the intersection of the active intervals of the alignments it is based on. If we want to split read sets down to a difference rate of δ, then we expect $\Delta = \delta|Z(T)|$ differences to exist inside $Z(T)$. Assuming suitable alignments comprising $\hat{R}_j$'s matrix, the probability to see each single of these differences is $p_6 = (1 - p_e)^6$ which is about 37.7% for $p_e = 15\%$. The probability to see at least m of these differences is $\eta(i) = \sum_{i=m}^{\Delta} \binom{\Delta}{i} p_6^i (1 - p_6)^{\Delta - i}$. We choose the smallest i s.t. $\eta(i)$ is at least 99.5% as a threshold. For each i we count the number $H(i)$ of tuples satisfying their threshold in which i appears as r_4, r_5 or r_6. Given P_d (Poisson distribution) as defined above we can determine a depth threshold d_t which is reached for most bases on the genome. Using the average sequence depth we can also estimate the likelihood of having a certain number v of sequence variants in the pile observed. Reads i with a count $H(i)$ close to or exceeding $h_t = \binom{d_t}{2}\binom{(v-1)d_t}{2}$ (we have fixed r_1 to 1 and one of r_4, r_5 or r_6 to r) are most likely not in the same class at R_j. Reads i in the same class as R_j should have a $H(i)$ equal or close to zero.

	Agreement based			Disagreement based		
copies	Precision	Recall	F_1 score	Precision	Recall	F_1 score
1	1	0.910	0.952	1	1	1
2	0.986	0.820	0.896	0.998	1	0.999
3	0.908	0.862	0.885	0.999	1	1
4	0.567	0.862	0.717	0.999	1	0.999
5	0.225	0.998	0.367	0.998	1	0.999
6	0.139	1	0.243	0.998	1	0.999
7	0.120	1	0.214	0.997	1	0.998
8	0.107	1	0.193	0.996	1	0.998

Table 2: Performance of splitting on 190kbp stretch of E. coli with $1 - 8$ copies added at 1% difference to original

3 Results

We have implemented both splitting methods. They are freely available as the programs split_agr and split_dis in the daccord package (see `https://github.com/gt1/daccord`). The daccord program (see [2]) in this package was also used to compute preliminary consensus sequences for the splitting. Read versus read alignments were computed using DALIGNER (cf. [1]). We performed two types of performance tests, both of which are based on simulated reads to ensure we can properly check whether and to what degree splittings computed are accurate.

In the first test we took a 190kb piece of the E. coli genome, duplicated it k times for $k = 1, 2, \ldots 8$ and spiked in 1% difference between the duplicated versions and the original. The differences are single bp insertions, deletions and substitutions with equal probability. We generated reads of average length 15kbp with an error rate of 15% to evenly cover the sequences at depth $d = 20$. For the splitting we only considered read overlaps of 5kbp and more to reduce noise in the underlying statistics. Table 2 shows the performance of agreement and disagreement based splitting in this scenario. We provide precision (which fraction of the alignments kept is true), recall (which fraction of the true alignments is kept) and F_1 (harmonic mean of precision and recall) score measures to quantify the performance of the read classification methods. All scores given are rounded to 3 significant decimals. While agreement based splitting has good precision for one and two modified copies, the performance quickly drops up to the point where essentially most wrong alignments are kept. The disagreement based splitting works close to perfectly in this setting. For computing the threshold h_t we have provided the correct value for the number of variants v to the program, as it does not yet support estimating it from the input data. The δ parameter was set to 1%.

As the first test is highly synthetic, we have chosen a somewhat more realistic scenario for the second one. We have extracted regions containing the genes FCGR1(A |B|CP), FCGR2(A|B|C) and FCGR3(A|B) plus 100kbp to the left and right of these regions from chromosome 1 of the human reference genome (GRCh38). These regions are highly repetitive with repeating stretches of length up to 46kbp with a difference of merely 1% and one repeat of length 26kbp with 0.4% difference between the copies. We generated reads and alignments using the same parameters as for the other test. Table 3 shows the performance of the splitting approaches we measured. While the region considered is repetitive in it's entirety, we do not have many cases of stretches appearing more than twice in total, i.e. most repeats have only two instances. As this is the setting in which agreement based splitting mostly works, we see a decent performance for this method as reflected in the table. For the disagreement based splitting we provide two lines, one for the default value of $h_t = 441$ which is computed as described above, the other one for $h_t = 7$, the setting which maximises the F_1 score in this scenario. As

Agreement based			Disagreement based			
Precision	Recall	F_1 score	h_t	Precision	Recall	F_1 score
0.953	0.934	0.943	441	0.899	1	0.947
			7	0.937	0.989	0.963

Table 3: Performance of splitting on FCGR regions of human chromosome 1

above we used $\delta = 1\%$. The recall value is good for both choices of h_t. We lose hardly any true alignments. The precision value for the default h_t of 441 is worse than the one of the agreement based method. A closer look reveals that the average difference between the true sequences we fail to separate (which lead to the false positive alignments we keep) is 0.467% which is way below our setting for δ, so the failure in separation is not surprising. Just reducing the parameter δ below 1% however does not markedly improve the splitting, as this also greatly increases noise (disagreement tuples observed although they are not real). The solution to this may be to require longer ($>$ 5kbp) overlaps between reads. As for PacBIO 50% of the sequenced bases is found in reads longer than 20kbp, this may be feasible. When we reduce h_t, then we are able to rule out more false positives, as there are some tuples for which most of the disagreements are observed and not just the small fraction we assume as a lower bound in our statistical considerations. This however also reduces the recall.

4 Conclusion

We have shown that repeat and haplotype separation in long reads with current read length and error rates is possible down to a difference of 1% and possibly less. This improves on the current state of the art of 3% set by Canu. The methods proposed also work if there are more than two underlying sequence versions. We hope these new insights can help to significantly improve the assembly of repetitive regions in genomes.

Acknowledgments

We thank Gene Myers for interesting algorithmical discussions related to this paper.

References

[1] G. Myers. ”Efficient Local Alignment Discovery amongst Noisy Long Reads”. *Proceedings WABI*, 2014, *LNCS*, vol.8701, pp. 52–67.

[2] G. Tischler and E. W. Myers. ”Non Hybrid Long Read Consensus Using Local De Bruijn Graph Assembly”. *bioRxiv*, 2017, `https://doi.org/10.1101/106252`.

[3] M. Martin et al. ”WhatsHap: fast and accurate read-based phasing”. *bioRxiv*, 2016, `https://doi.org/10.1101/085050`.

[4] V. Bansal, A. L. Halpern, N. Axelrod and Vineet Bafna. ”An MCMC algorithm for haplotype assembly from whole-genome sequencing data”. *Genome Research*, vol.18, pp. 1336–1346, 2008.

[5] V. Bansal and V. Bafna. ”HapCUT: an efficient and accurate algorithm for the haplotype assembly problem”. *Bioinformatics*, vol.24, pp. i153–i159, 2008.

[6] S. Mazrouee and W. Wang. ”FastHap: fast and accurate single individual haplotype reconstruction using fuzzy conflict graphs”. *Bioinformatics*, vol.30, pp. i371–i378, 2014.

[7] F. Deng, W. Cui and L. Wang. ”A highly accurate heuristic algorithm for the haplotype assembly problem”. *BMC Genomics*, vol.14S2, 2013.

[8] Chen-Shan Chin et al. ”Phased diploid genome assembly with single-molecule real-time sequencing”. *Nature Methods*, vol.13, pp. 1050–1054, 2016.

[9] S. Koren et al. ”Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation”. *Genome Research*, vol.27, pp. 722–736, 2017.