Analysis of Short-read Aligners using Genome Sequence Complexity

Quang Tran[1], Nam Sy Vo[2], Eric Hicks[1], Tin Nguyen[3], and Vinhthuy Phan[1,*]

[1]Department of Computer Science, University of Memphis, Memphis, TN 38152, USA
[2]Department of Computational Biomedicine, Vingroup Big Data Institute, Hanoi, Vietnam
[3]Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA
[*]Corresponding author: vphan@memphis.edu

***Abstract*—Next generation sequencing technologies have the capability to provide large numbers of short reads inexpensively and accurately. Researchers have proposed many different methods to align short reads to reference genomes. Nevertheless, long repeats, which are known to be abundant in eukaryotic genomes, have caused considerable difficulty for genome assembly methods that rely on short-read alignment. Although a few researchers have studied sequence complexity of genomes in terms of repeats, none have quantitatively related such complexity to the difficulty of short read alignment and assembly. In this paper, we investigate several measures of genome sequence complexity with the goal of quantifying the difficulty of short read alignment. Using genomic data from 17 different organisms and testing against 12 state-of-the-art short-read aligners, we found a very strong correlation between the performance of virtually all of these aligners and measures of genome sequence complexity. Further, we show how these measures might be used to analyze and predict the performance of aligners, and more importantly, select the best aligners for specific genomes.**

***Index Terms*—genome complexity, short-read alignment, genomic analysis**

I. Introduction

Modern sequencing technologies provide high volumes of raw data with increasingly high accuracy and longer read length. These advances have encouraged many different computational approaches to address the problem of short read alignment and assembly. To improve alignment speed, alignment algorithms typically index the genome or reads, and utilize advanced data structures such suffix trees, suffix arrays, FM-indexing, hash tables, and q-gram indexing [1].

Although there are many different approaches, the alignment and assembly problem remains challenging due to the presence of many long repeats in genomes [2]. This is particularly true of the human genome [3]. With current sequencing technologies, reads are typically much shorter than repeats, causing aligners to fail to align or misalign reads, resulting in large gapped/unaligned regions.

Becher [4] studied the complexity of general strings and derived several interesting properties. Whiteford [5] and Kurtz [6] formulated different ways to study the sequence complexity of genomes, especially in terms of repeat elements and unique k-mers. These works focus on visualizing and describing the complexity of genomes in general. Yu [7] evaluated the alignment performance of four aligners on a few repetitive regions selected from CpG islands, but did not provide deep analyses or conclusions beyond the generally known view that long repeats degrade alignment performance. A challenge for choosing the best aligner for a set of genomes of interest is the fact that an aligner can be superior for certain types of genomes and reads (with specific ranges of lengths from different technologies), but becomes inferior for other types of genomes or reads [8]. In practice, the adoption of aligners can be based on limited anecdotal evidence without firm knowledge of how accurately they will perform on new genomes. Even if one adopts a generally very good aligner, its alignment accuracy on new genomes is probably not well understood [9, 10, 11], potentially resulting in expensive experimental designs, e.g. requiring unnecessarily high numbers of reads aimed to cover the entire genome [12]. Thus, there is a practical need to thoroughly understand the accuracy of aligners.

Short-reads might not be exact substrings of the reference genome and their alignment to the reference might be obtained approximately. To align millions of reads efficiently [13, 14, 15], a common strategy taken by most recent aligners includes two main steps. In the first step, given a read, exact matches between a substring of the read (call a *seed*) and the reference genome are found. In the next step, seeds are extended to account for approximate matching between the read and regions of the genome. If a read matches identically to the genome and occurs only once in the genome, the alignment is easy. On the other hand, if a read matches approximately to a substring of the genome, and this substring occurs several times or matches approximately to many substrings of the genome, the alignment of the read is likely challenging. Although new and improved methods are constantly introduced, the problem of aligning short-reads to reference genomes remains intrinsically challenging due to the fact that genomic repeats are much longer than reads. For instance, Yu [7] evaluated alignment performance of several aligners on repetitive regions and concluded that long repeats seriously degraded alignment performance.

In this paper, we investigate three formulations of genome sequence complexity, defined in terms of distinct substrings and repeats, especially at specific read lengths. Our formulations are slightly different from the existing ones, because

978-1-7281-4510-5/20/$31.00 ©2020 IEEE

we aim to utilize these formulations to analyze the difficulty of short-read alignment and assembly. These formulations can be computed efficiently in linear time relative to the size of a genome. Through an extensive analysis of 12 state of the art aligners, we show that these formulations of genome sequence complexity correlate highly with the performance of short-read alignment. Consequently, genomes can be ranked and cataloged in terms of how hard it is to align short reads to them, and ultimately how hard it is to assemble them. Further, based on this result, we show how to predict the performance of alignment algorithms and how to select the best aligners for specific genomes. These results help deepen the understanding of genome sequence complexity and bring practical benefits to researchers who are interested in short-read alignment and genome assembly.

II. Method

A. Complexity of genomes

Given a sequence g, denote the number of occurrences of a string x in g as $f(x)$. $f(x) > 0$ if and only if x is a substring of g. If $f(x) > 1$, we call x a *repeat*. The lengths of g and x are denoted as $|g|$ and $|x|$, respectively. We define three measures of complexity for g as follows:

$$\begin{aligned} D_g &= \frac{2 \cdot |\{x : f(x) > 0\}|}{|g| \cdot (|g| + 1)} \\ D_g^k &= \frac{|\{x : f(x) > 0, |x| = k\}|}{|g| - k + 1} \\ R_g^k &= \frac{\sum_{f(x)>1} f(x)}{|g| - k + 1} \end{aligned}$$

We refer to D_g as the density of distinct substrings of g, D_g^k as the density of distinct k-substrings of g, and R_g^k as the density of k-repeats of g. Since the number of distinct substrings is maximally $|g|(|g| + 1)/2$ and the number of substrings of length k is maximally $|g|-k+1$. These densities are between 0 and 1. While D_g is not specific to a particular value k, D_g^k and R_g^k target specific values of read lengths. Both D_g and D_g^k are directly proportional, whereas R_g^k is inversely proportional to the number of repeats in g.

We contend that these densities capture different aspects of the complexity of g, especially in terms of how hard it is to align short reads to a genome. In particular, these densities are defined so that they are proportional to the probability of correctly aligning reads to genomes. D_g is similar to the notion of I-complexity introduced by [4]. The main difference is that I-complexity is defined using discrete derivative log functions, making this measure essentially exponentially smaller than D_g. The advantage of D_g is apparent later when it is linearly correlated with the performance of short-read aligners.

D_g^k is similar to the *k-mer occurrence ratio*, $\rho_{g,k}(1, |g| - k + 1)$, introduced by [6]. The difference is that while ρ is normalized (or divided) by $|D_g|$, D_g^k is normalized (or divided) by $|g| - k + 1$. Although this difference appears minute, it is actually an important distinction as D_g^k correlates better with the probability of correctly aligning short reads to genomes.

R_g^k is related to the function $C(k, r)$ defined by [5] to be the number of k-substrings that repeat exactly r times; i.e. $R_g^k = \sum_{r>1} C(k, r)$. The authors were mainly interested in using $C(k, r)$ to visualize genome sequence complexity, rather than relating complexity to the difficulty of aligning and assembling short reads, which is the goal of defining R_g^k.

B. Computation of repeat density and distinct substring density

The computation of D_g, D_g^k, and R_g^k can be done efficiently in linear time and space using suffix arrays and Longest Common Prefix (LCP) arrays. The suffix array S of g stores implicitly lexicographically sorted suffixes of g; i.e. for $i < j$, $g_{S[i]\cdots|g|}$, the suffix of g starting at index $S[i]$, is lexicographically smaller than $g_{S[j]\cdots|g|}$, the suffix of g starting at index $S[j]$.

$LCP[i]$ is defined to be the length of the longest common prefix of $g_{S[i-1]\cdots|g|}$ and $g_{S[i]\cdots|g|}$. The construction of S and LCP can be done in linear time. We now show that the numbers of distinct substrings, distinct k-substrings, and k-repeats can be computed in linear time by traversing the LCP array.

C. Correlation of sequence complexity of genomes and difficulty of alignment

When a genome has either low distinct substring densities (D_g and D_g^k) or high repeat density (R_g^k), the probability that a random substring will be mapped to multiple locations of the genome will be higher, making it difficult to identify correct locations of substrings even if other information is incorporated. Although reads are synthesized from unknown genomes and then aligned to reference genomes, these genomes are expected to be very similar as they belong to the same species. Thus, we expect that D_g, D_g^k, and R_g^k have a direct effect on how hard it is to align short reads (of unknown genomes) to reference genomes.

To quantify the effect of genome sequence complexity on the difficulty of aligning short reads to reference genomes, we may correlate the densities D_g, D_g^k, and R_g^k to the performance of different short-read alignment algorithms. The performance of an alignment algorithm can be described in terms of *precision* and *recall*:

$$Precision = \frac{tp}{tp + fp} \qquad Recall = \frac{tp}{tp + fn}$$

where tp is the number of correctly aligned reads, fp is the number of incorrectly aligned reads, and fn is the sum of incorrectly aligned reads and unaligned reads; in effect, $tp+fn$ is equal to the total number of reads.

As we shall show in Section III, D_g, D_g^k, and R_g^k are highly correlated with precision and/or recall of all aligners we tested. This validates their usefulness as measures of genome sequence complexity, especially in the context of alignment and assembly. Further, these correlations can help us construct linear models of alignment performance so that the best aligners can be adopted for specific genomes.

III. Experiments and Results

A. Data

We conducted our experiments on the genomic data of 17 prokaryotic & eukryotic species taken from the NCBI, EBI, and Ensembl databases. This data was selected based on its size and a diversity of complexity (Table I). Reads of lengths 100, 200, and 400 were simulated using the WGSIM program, part of the SAMtool package [16]. These lengths were selected as they represent the typical read lengths of current technologies. We applied the default setting of WGSIM with an error rate of 0.02 per base and 0.15 indel polymorphisms. We used 2x coverage across the datasets; coverage does not affect alignment quality as much as it affects assembly quality. Table II shows the values of D_g, D_g^k, R_g^k for these genomes. For D_g^k and R_g^k, the values were computed at k equal to read lengths 100, 200, and 400, respectively.

TABLE I: Genomic data from 17 species used as test data

gID	Genome	Accession #	Size (bp)
1	*Caenorhabditis elegans Bristol N2*	BX284601	251,136
2	*Canis lupus familiaris chr. 1*	CM000001	122,678,785
3	*Drosophila yakuba strain chr. 2L*	CM000157	22,324,452
4	*Bos taurus chr. 1*	CM000177	161,428,367
5	*Mus musculus chr. 1*	CM000209	199,526,509
6	*Equus caballus chr. 1*	CM000377	185,838,109
7	*Taeniopygia guttata chr. 1*	CM000515	118,548,696
8	*Zea mays chr. 1*	CM000777	301,354,135
9	*Acaryochloris marina MBIC11017*	CP000828	6,503,724
10	*Bacillus megaterium DSM319*	CP001982	87,884
11	*Achromobacter xylosoxidans A8*	CP002287	116,819
12	*Actinoplanes sp. SE50/110*	CP003170	9,239,851
13	*Arabidopsis thaliana chr. 1*	NC_003070.9	30,427,671
14	*Human herpesvirus 4 type 1*	NC_007605	171,823
15	*Populus trichocarpa linkage group I*	NC_008467.1	35,571,569
16	*Danio rerio Zv9.73 chr. 1*	GCA_000002035.2	60,348,388
17	*Gorilla gorilla gorGor3.1.73 chr. 1*	GCA_000151905.1	229,507,203

TABLE II: Values of R_g^k, D_g^k, D_g of 17 genomes at read lengths 100, 200, and 400.

gID	D_g	D_g^{100}	D_g^{200}	D_g^{400}	R_g^{100}	R_g^{200}	R_g^{400}
1	0.999997419	0.9879	0.9941	0.9972	0.0190	0.0099	0.0048
2	0.999997125	0.9912	0.9953	0.9969	0.0131	0.0079	0.0058
3	0.999998745	0.9928	0.9971	0.9992	0.0118	0.0053	0.0015
4	0.999999211	0.9749	0.9863	0.9889	0.0413	0.0264	0.0221
5	0.999999785	0.9844	0.9943	0.9984	0.0220	0.0086	0.0024
6	0.999999833	0.9972	0.9990	0.9996	0.0040	0.0015	0.0006
7	0.999999720	0.9884	0.9964	0.9996	0.0221	0.0070	0.0009
8	0.999999563	0.8628	0.9505	0.9827	0.1898	0.0774	0.0300
9	0.999992484	0.9775	0.9808	0.9857	0.0318	0.0273	0.0204
10	0.999993153	0.9921	0.9930	0.9944	0.0093	0.0083	0.0070
11	0.999995814	0.9981	0.9982	0.9984	0.0035	0.0032	0.0029
12	0.999996839	0.9962	0.9972	0.9981	0.0064	0.0046	0.0031
13	0.999993844	0.9816	0.9859	0.9885	0.0306	0.0245	0.0207
14	0.985835683	0.8500	0.8550	0.8620	0.1762	0.1691	0.1608
15	0.999998489	0.9813	0.9873	0.9922	0.0361	0.0249	0.0154
16	0.999997842	0.9663	0.9805	0.9877	0.0534	0.0318	0.0207
17	0.999999614	0.9841	0.9888	0.9915	0.0268	0.0196	0.0147

B. R_g^k, D_g^k, and D_g correlate highly with alignment quality

To avoid biases toward a specific algorithmic approach, we selected state-of-the-art research/commercial short-read alignment software packages that employ different algorithmic techniques and data structures including hash tables with gapped or contiguous seeds (SHRiMP2 [17], mrFAST [18], SeqAlto [19]), hash tables with seed-extension heuristics (GASSST [20], SRMapper [21], Novoalign* and Smalt†), FM-indexing (Bowtie2 [22], BWA-SW [23], SOAP2 [24]), and FM-indexing combined with comprehensive seed-extension heuristics (CUSHAW2 [25], Masai [26]).

TABLE III: Correlation coefficients of R_g^k, D_g^k, D_g and precision/recall at read lengths 100, 200, and 400.

	Precision			Recall		
$k = 100$	R_g^{100}	D_g^{100}	D_g	R_g^{100}	D_g^{100}	D_g
Bowtie2	-0.9829	0.9654	0.5177	-0.9832	0.9659	0.5211
Bwasw	-0.9915	0.9796	0.5686	-0.9864	0.9712	0.5434
Seqalto	-0.9962	0.9899	0.6124	-0.9964	0.9908	0.6201
Cushaw2	-0.9958	0.9878	0.6016	-0.9958	0.9878	0.6017
Shrimp	-0.9964	0.9898	0.6105	-0.9953	0.9879	0.6079
mrFAST	-0.9960	0.9890	0.6073	-0.9950	0.9915	0.6274
Masai	-0.9962	0.9889	0.6064	-0.9963	0.9893	0.6103
Smalt	-0.9960	0.9894	0.6106	-0.9963	0.7771	0.1306
Gassst	-0.9480	0.9143	0.3783	-0.8379	0.7921	0.3193
Soap2	-0.9939	0.9833	0.5785	-0.9883	0.9860	0.6516
Novoalign	-0.6585	0.5899	-0.1139	-0.9877	0.9680	0.5349
Srmapper	-0.6942	0.6200	-0.0238	-0.9662	0.9525	0.5578
$k = 200$	R_g^{200}	D_g^{200}	D_g	R_g^{200}	D_g^{200}	D_g
Bowtie2	-0.9794	0.9554	0.8128	-0.9794	0.9538	0.8105
Bwasw	0.9895	0.9895	0.8939	-0.9963	0.9894	0.8934
Seqalto	-0.9961	0.9909	0.8993	-0.9959	0.9901	0.8974
Cushaw2	-0.9963	0.9910	0.8983	-0.9962	0.9908	0.8982
Shrimp	-0.9969	0.9927	0.9038	-0.9819	0.9603	0.8212
mrFAST	-0.9959	0.9918	0.9029	-0.9897	0.9858	0.8989
Masai	-0.9961	0.9917	0.9029	-0.9674	0.9706	0.9075
Smalt	-0.9962	0.9924	0.9042	-0.9674	0.7254	0.6234
Gassst	-0.8880	0.8329	0.6152	-0.3156	0.2760	0.1934
Soap2	-0.9940	0.9857	0.8826	-0.8969	0.9211	0.8977
Novoalign	-0.2221	0.1306	-0.1401	-0.9843	0.9611	0.8312
Srmapper	-0.2504	0.1467	-0.1075	-0.9402	0.9116	0.7764
$k = 400$	R_g^{400}	D_g^{400}	D_g	R_g^{400}	D_g^{400}	D_g
Bowtie2	-0.9900	0.9818	0.9455	-0.9889	0.9804	0.9436
Bwasw	-0.9955	0.9946	0.9720	-0.9959	0.9951	0.9726
Seqalto	-0.9962	0.9951	0.9722	-0.9958	0.9946	0.9715
Cushaw2	-0.9958	0.9949	0.9723	-0.9957	0.9949	0.9722
Shrimp	-0.9963	0.9954	0.9725	-0.9944	0.9934	0.9708
mrFAST	-0.9962	0.9948	0.9712	-0.9912	0.9926	0.9726
Masai	-0.9898	0.9883	0.9621	-0.7671	0.7830	0.8048
Smalt	-0.9898	0.9936	0.9685	-0.7671	0.7830	0.8858
Gassst	-0.8827	0.8523	0.7785	-0.2869	0.2602	0.2378
Soap2	-0.9635	0.9493	0.8961	-0.2996	0.3013	0.9715
Novoalign	0.0329	-0.0580	-0.1030	-0.9871	0.9754	0.9362
Srmapper	-0.0291	-0.0430	-0.1583	-0.9269	0.9048	0.8529

Table III shows the Pearson correlation coefficients for precision and recall on the aligners at read length k and R_g^k, D_g^k, and D_g, for k equal to 100, 200, and 400, respectively. Correlation coefficients have values between -1 and 1. Values closer to 0 mean no correlation; values closer to 1 (or -1) mean the two quantities are highly correlated positively (or negatively). These observations can be made:

First, by and large, R_g^k, D_g^k, and D_g correlate very highly with the performance (precision and/or recall) of all aligners; correlation coefficients are nearly 1 in most cases. R_g^k correlates negatively, while D_g^k and D_g correlate positively with alignment performance. This makes sense as higher R_g^k values imply a higher ratio of repeats, whereas higher D_g^k or D_g values imply a higher ratio of distinct k-mers.

Second, R_g^k and D_g^k are similarly correlated (in magnitude) to performance, although R_g^k is a little bit more correlated in both precision and recall. We may conclude that given a

*Novocraft Technologies, www.novocraft.com

†Wellcome Sanger Institute, https://www.sanger.ac.uk/tool/smalt-0/

specific read length, R_g^k is the best measure of complexity.

Third, since D_g is by definition free of any specific lengths of substrings, it cannot take advantage of specific read lengths as the other two measures do. Consequently, D_g does not correlates as highly as the other two measures do. Nevertheless, the longer the reads become, the more D_g correlates with both precision and recall. In many cases at $k = 400$, D_g correlation with coefficients is nearly 1. This means, if we are not given specific information about read lengths *a priori*, D_g can still be a good measure of complexity.

Fourth, R_g^k, D_g^k, and D_g correlate well will most aligners in *both* precision and recall. At longer read lengths, the correlation is very high for only recall but not precision for two aligners (Novoalign and SRmaper). Interestingly, for GASSST, the reverse is true: correlation is high for precision, but not recall. We suspect that these aligners adopt algorithmic and heuristic strategies that yield consistent performance on either precision or recall, but not both.

In summary, across 12 state-of-the-art research and commercial short-read aligners that employ a diverse sets of strategies and data structures, performance is correlated very significantly to R_g^k, D_g^k, and D_g.

C. Analyzing aligners' performance

Although we expect that repeats and distinct substrings are closely related to alignment quality, the ability to *quantify* this relationship enables interesting genome analysis and performance prediction. For performance prediction, the correlations enable us to construct linear models of performance for each aligner (coefficients summarized in Table III). For example, the linear model for SeqAlto [19] is shown in Figure 1. Suppose hypothetically that we would like to know how well SeqAlto performs on an unknown genome, g', for reads of length 100. And further suppose that $R_{g'}^{100}$ is 0.1. Then, the linear model for SeqAlto predicts that precision would be $-0.8609 \cdot (0.1) + 1.0012 \approx 0.92$ and recall would be $-0.9211 \cdot (0.1) + 0.9959 \approx 0.90$. As R_g^{100} for SeqAlto is highly correlated to its performance (0.9962 for precision and 0.9964 for recall), we expect that this prediction is accurate.

D. Choosing the best aligners for specific genomes

As different aligners have different trade-offs and characteristics, it is often difficult to choose the best aligners for a specific need. Here, we are interested in finding the best aligners for specific genomes of interest. Given a list of highly correlated linear models of aligners, we can compute the complexity (e.g. R_g^k) of an unknown genome and use it to select the best aligners for that specific genome. Indeed, the linear models constitute a *concave-shaped optimal front* that divides the complexity space into intervals, each of which associates with an optimal aligner. Technically, the *optimal front* consists of an ordered list of models $[f_1, \cdots, f_m]$ and a list of numbers in increasing order: $[p_1, \cdots, p_{m-1}]$. The best aligner for a genome with complexity c is f_1 if $c \leq p_1$, f_m if $c > p_{m-1}$, and f_i if $p_{i-1} < c \leq p_i$.

As an example, Figure 2 shows the linear models (of recall), computed from our experiment, for mrFAST, GASSST, Novoalign: $y_1 = -0.8696x + 0.944$, $y_2 = -0.9187x + 0.9716$, and $y_3 = -1.1055x + 0.995$. The optimal front can be computed as $[y_3, y_2, y_1]$ and $[0.13, 0.56]$. For genomes whose repeat density (R_g^k) values are less than 0.13, Novoalign is the best. For values between 0.13 and 0.56, GASSST is the best. And for values larger than 0.56, mrFAST is the best.

The optimal front for the set of linear equations $H = \{h_1, \cdots, h_m\}$ can be found by first selecting the equation with the highest performance at 0, iteratively intersecting the last selected equation with the remaining ones, and then choosing the one with highest performance at the intersection points. Formally,

1: Find $h^* \in H$ such that $h^*(0) \geq h(0), \forall h \in H$
2: Let F be the queue $[h^*]$; P be an empty queue
3: $H \leftarrow H - h^*$
4: **while** H is not empty **do**
5: Let $I = \{(x_1, h_1(x_1)), \cdots, (x_{|H|}, h_{|H|}(x_{|H|}))\}$ be the intersection points of the last equation in F with equations in H
6: Find $(x^*, h^*(x^*)) \in I$ such that $h^*(x^*) \geq h_j(x_i), \forall (x_j, h_j(x_j)) \in I$.
7: $F.enqueue(h^*)$; $P.enqueue(x^*)$
8: $H \leftarrow H - h^*$
9: **return** (F, P)

IV. Conclusion

We investigated different measures of genome sequence complexity. These measures can be computed efficiently and were shown to strongly correlate to the difficulty of aligning short reads to reference genomes. This correlation enables us to build linear models of alignment performance to analyze performance characteristics of different algorithms. These linear models enable us to select the best aligners for specific genomes based on their complexity. These results should be useful for the study of genome sequence complexity as well as for the study of short-read alignment and assembly.

The ability to predict accuracy of short-read aligners without aligning reads has two benefits. First, it will save time and serve as an additional useful criterion to compare different alignment algorithms and to select the most accurate aligners for unknown genomes. Second, we can use linear regression models of aligners to build an *optimal front*, as shown conceptually in Figure 2 and select the most accurate aligners for a new genome based on its complexity value. Computing sequence complexity for genomes is much less computationally expensive than aligning millions of reads of different lengths and mutation rates to genomes.

Acknowledgment

The authors would like to thank the following for their contributions: Kevin O'Kello and Shanshan Gao for their invaluable help in running some of the short-read alignment tools; Eric Spangler for high performance computing support research.

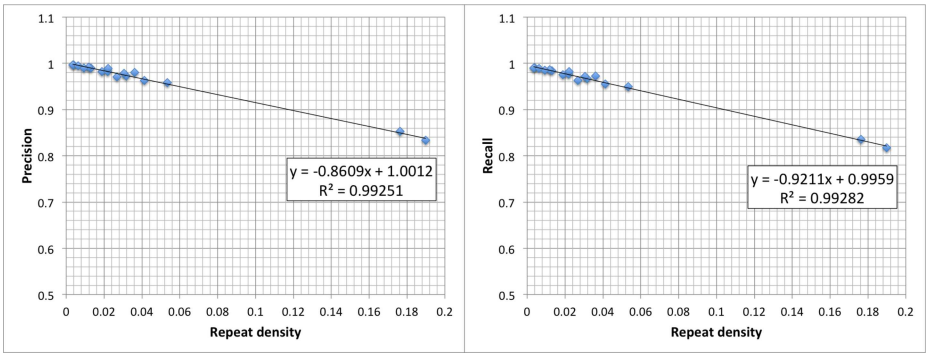

Fig. 1: Correlation of performance of SeqAlto [19] to R_g^{100} at read length 100.

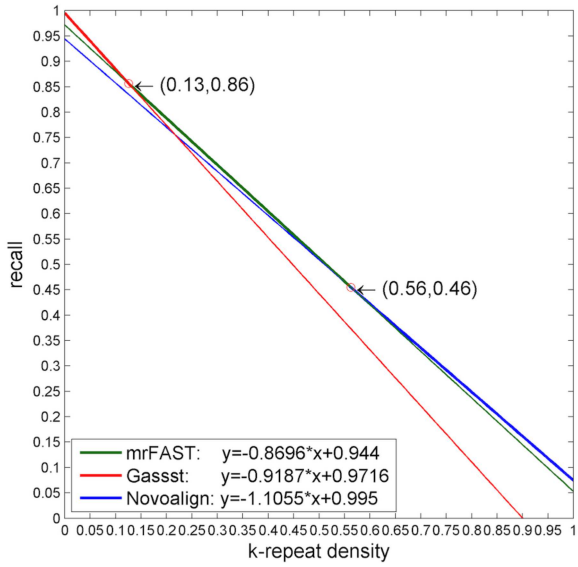

Fig. 2: Linear models for 3 different aligners are used to compute the *optimal front*, from which the best aligner can be chosen based on values of R_g^k. The optimal front is in bold color.

REFERENCES

[1] H. Li and N. Homer, "A survey of sequence alignment algorithms for next-generation sequencing," *Briefings in bioinformatics*, vol. 11, no. 5, pp. 473–483, 2010.

[2] T. Treangen and S. Salzberg, "Repetitive dna and next-generation sequencing: computational challenges and solutions," *Nat Rev Genet*, vol. 13, no. 1, pp. 36–46, 2012.

[3] J. de Koning, W. Gu, T. Castoe, M. Batzer, and D. Pollock, "Repetitive elements may comprise over two-thirds of the human genome," *PLoS Genet*, vol. 7, no. 12, p. e1002384, 2011.

[4] V. Becher and P. Heiber, "A linearly computable measure of string complexity," *Theoretical Computer Science*, vol. 438, pp. 62–73, 2012.

[5] N. E. Whiteford, N. J. Haslam, G. Weber, A. Prugel-Bennett, J. W. Essex, C. Neylon, *et al.*, "Visualizing the repeat structure of genomic sequences," *Complex Systems*, vol. 17, no. 4, pp. 381–398, 2008.

[6] S. Kurtz, A. Narechania, J. C. Stein, and D. Ware, "A new method to compute k-mer frequencies and its application to annotate large repetitive plant genomes," *BMC genomics*, vol. 9, no. 1, p. 517, 2008.

[7] X. Yu, K. Guda, J. Willis, M. Veigl, Z. Wang, M. D. Markowitz, *et al.*, "How do alignment programs perform on sequencing data with varying qualities and from repetitive regions?," *BioData mining*, vol. 5, p. 6, 2012.

[8] W. Wang, Z. Wei, T.-W. L. Lam, and J. Wang, "Next generation sequencing has lower sequence coverage and poorer snp-detection capability in the regulatory regions," *Scientific Reports*, vol. 1, no. 55, p. doi:10.1038/srep00055, 2011.

[9] Q. Tran, S. Gao, N. S. Vo, and V. Phan, "Repeat complexity of genomes as a means to predict the performance of short-read aligners," in *Proceedings of the 8th International Conference on Bioinformatics and Computational Biology (BiCOB)*, 2016.

[10] Q. Tran, S. Gao, and V. Phan, "Analysis of optimal alignments unfolds aligners bias in existing variant profiles," in *BMC bioinformatics*, vol. 17, p. 349, BioMed Central, 2016.

[11] S. Gao, Q. Tran, and V. Phan, "Understand effective coverage by mapped reads using genome repeat complexity," in *Proceedings of 11th International Conference on Bioinformatics and Computational Biology (BICOB)*, vol. 60, pp. 65–73, 2019.

[12] V. Phan, S. Gao, Q. Tran, and N. S. Vo, "How genome complexity can explain the difficulty of aligning reads to genomes," *BMC bioinformatics*, vol. 16, no. S17, p. S3, 2015.

[13] S. Canzar and S. L. Salzberg, "Short read mapping: An algorithmic tour," *Proceedings of the IEEE*, vol. 105, no. 3, pp. 436–458, 2015.

[14] N. S. Vo, Q. Tran, N. Niraula, and V. Phan, "A randomized algorithm for aligning dna sequences to reference genomes," in *2013 IEEE 3rd International Conference on Computational Advances in Bio and medical Sciences (ICCABS)*, pp. 1–2, IEEE, 2013.

[15] N. S. Vo, Q. Tran, N. Niraula, and V. Phan, "Randal:

a randomized approach to aligning dna sequences to reference genomes," *BMC genomics*, vol. 15, no. 5, p. S2, 2014.

[16] H. Li, B. Handsaker, A. Wysoker, T. Fennell, J. Ruan, N. Homer, G. Marth, G. Abecasis, R. Durbin, and . G. P. D. P. Subgroup, "The sequence alignment/map format and samtools," *Bioinformatics*, vol. 25, no. 16, pp. 2078–2079, 2009.

[17] M. David, M. Dzamba, D. Lister, L. Ilie, and M. Brudno, "Shrimp2: sensitive yet practical short read mapping," *Bioinformatics*, vol. 27, no. 7, pp. 1011–1012, 2011.

[18] C. Alkan, J. M. Kidd, T. Marques-Bonet, G. Aksay, F. Antonacci, *et al.*, "Personalized copy number and segmental duplication maps using next-generation sequencing," *Nature genetics*, vol. 41, no. 10, pp. 1061–1067, 2009.

[19] J. C. Mu, H. Jiang, A. Kiani, M. Mohiyuddin, N. B. Asadi, and W. H. Wong, "Fast and accurate read alignment for resequencing," *Bioinformatics*, vol. 28, no. 18, pp. 2366–2373, 2012.

[20] G. Rizk and D. Lavenier, "Gassst: global alignment short sequence search tool," *Bioinformatics*, vol. 26, no. 20, pp. 2534–2540, 2010.

[21] P. M. Gontarz, J. Berger, and C. F. Wong, "Srmapper: a fast and sensitive genome-hashing alignment tool," *Bioinformatics*, vol. 29, no. 3, pp. 316–321, 2013.

[22] B. Langmead and S. L. Salzberg, "Fast gapped-read alignment with bowtie 2," *Nature Methods*, vol. 9, no. 4, pp. 357–359, 2012.

[23] H. Li and R. Durbin, "Fast and accurate long-read alignment with burrows–wheeler transform," *Bioinformatics*, vol. 26, no. 5, pp. 589–595, 2010.

[24] R. Li, Y. Li, K. Kristiansen, and J. Wang, "Soap: short oligonucleotide alignment program," *Bioinformatics*, vol. 24, no. 5, pp. 713–714, 2008.

[25] Y. Liu and B. Schmidt, "Long read alignment based on maximal exact match seeds," *Bioinformatics*, vol. 28, no. 18, pp. i318–i324, 2012.

[26] E. Siragusa, D. Weese, and K. Reinert, "Fast and accurate read mapping with approximate seeds and multiple backtracking," *Nucl Acids Res*, vol. 41, no. 7, p. e78, 2013.