Full Title

Demonstrating the utility of flexible sequence queries against indexed short reads with FlexTyper

Short Title

FlexTyper: Enabling flexible queries against indexed short read sequences

Phillip A. Richmond[1*], Alice M. Kaye[1*], Godfrain Jacques Kounkou[1*], Tamar V. Av-Shalom[1], Wyeth W Wasserman[1]

[1]Centre for Molecular Medicine and Therapeutics, BC Children's Hospital Research Institute, University of British Columbia, Vancouver, BC, Canada

*These authors contributed equally.

Corresponding Author

Wyeth W. Wasserman <wyeth@cmmt.ubc.ca>

Abstract

Across the life sciences, processing next generation sequencing data commonly relies upon a computationally expensive process where reads are mapped onto a reference sequence. Prior to such processing, however, there is a vast amount of information that can be ascertained from the reads, potentially obviating the need for processing, or allowing optimized mapping approaches to be deployed. Here, we present a method termed FlexTyper which facilitates a "reverse mapping" approach in which high throughput sequence queries, in the form of kmer searches, are run against indexed short-read datasets in order to extract useful information. This reverse mapping approach enables the rapid counting of target sequences of interest. We demonstrate FlexTyper's utility for recovering depth of coverage, and accurate genotyping of SNP sites across the human genome. We show that genotyping unmapped reads can correctly inform a sample's population, sex, and relatedness in a family setting, which can be used to inform optimized downstream analysis pipelines. Detection of pathogen sequences within RNA-seq data was sensitive and accurate, performing comparably to existing methods with increased flexibility. The long-term adoption of the "reverse mapping" approach represented by FlexTyper will be enabled by more efficient methods for FM-index generation and biology-informed collections of reference queries. In the long-term, selection of population-specific references or weighting of edges in pan-population reference genome graphs will be enabled by the FlexTyper reverse mapping approach. FlexTyper is available at https://github.com/wassermanlab/OpenFlexTyper.

Author Summary

In the past 15 years, next generation sequencing technology has revolutionized our capacity to process and analyze DNA sequencing data. From agriculture to medicine, this technology is enabling a deeper understanding of the blueprint of life. Next generation sequencing data is composed of short sequences of DNA, referred to as "reads", which are often shorter than 200 base pairs making them many orders of magnitude smaller than the entirety of a human genome. Gaining insights from this data has typically leveraged a reference-guided mapping approach, where the reads are aligned to a reference genome and then post-processed to gain actionable information such as presence or absence of genomic sequence, or variation between the reference genome and the sequenced sample. Many experts in the field of genomics have concluded that selecting a single linear reference genome for mapping reads against is limiting, and several current research endeavours are focused on exploring options for improved analysis methods to unlock the full utility of sequencing data. Among these improvements are the usage of sex-matched genomes, population-specific reference genomes, and emergent graph-based reference genomes. Data-driven approaches which inform these complex analysis pipelines are currently lacking. Here we develop a method termed FlexTyper, which creates a searchable index of the short read data and enables flexible, rapid, user-guided queries to provide valuable insights without the need for reference-guided mapping. We demonstrate the utility of our method by identifying sample ancestry and sex in human whole genome sequencing data, as well as detecting viral pathogen reads in RNA-seq data. We anticipate early adoption of FlexTyper within analysis pipelines as a pre-mapping component, and further envision the bioinformatics and genomics community will leverage the tool for creative uses of sequence queries from unmapped data.

Introduction

Short-read DNA sequencing enables diverse molecular investigations across life science applications spanning from medicine to agriculture. Obtaining useful information from sequencing datasets typically involves either performing *de novo* assembly, or mapping the data against one or more reference genomes. The process of mapping sequencing reads (short pieces of DNA read-outs from the DNA sequencer) against reference genomes, or a collection of reference genomes, is made computationally tractable by indexing the reference sequences, commonly performed with a Burrows Wheeler transform or FM-index. Several data analysis pipelines, whether they focus on quantification (e.g. observed gene expression in RNA sequencing data), or identifying sequence differences between a sample and a reference genome (e.g. genotyping), leverage reference genome mapping as a primary analysis component.

While the status quo has been to utilize linear representations of reference genomes, a transition away from a single haploid reference genome is inevitable (Yang et al. 2019; Ballouz, Dobin, and Gillis 2019). This transition is supported by several factors. A large amount of structural variation exists between human populations (Feuk, Carson, and Scherer 2006; MacDonald et al. 2014; Levy-Sakin et al. 2019). A recent study focusing on ~1000 individuals of African descent identified nearly 200 million bases missing from the most recent reference genome (Sherman et al. 2019). Static linear reference genomes which do not capture these large differences between populations impose challenges for accurate genotyping (Ballouz, Dobin, and Gillis 2019; Yang et al. 2019), with implications in medicine and association studies. An alternative to choosing from a collection of population-specific reference genomes is to use emerging graph genome approaches to unite the data (Dilthey et al. 2015). As highlighted in a review by (Paten et al. 2017), in either approach, a key challenge in the future will be to determine the most appropriate reference genome(s), or path(s) through a graph genome, to maximize genotyping performance. Knowledge of distributed single nucleotide polymorphisms (SNPs) genotypes across the genome can be used to guide such choices.

Currently, the primary approach for identifying SNP genotypes across the genome utilizes computationally expensive reference-based read mapping and variant calling strategies (Nielsen et al. 2011). Inferring ancestry from specific, population-discriminating SNPs can be performed rapidly with the recently published tool Peddy, which uses fewer than 25,000 SNPs to identify

ancestry through principal component analysis (Pedersen and Quinlan 2017). Previous work showed that it is possible to genotype predefined SNPs from unmapped sequence data, circumventing the read mapping and variant calling process (Dolle et al. 2017; Sun and Medvedev 2019; Shajii et al. 2016). Some approaches focus on kmer (short sequences of length k) hashing and matching to predefined target kmers to perform genotyping of known SNPs, as demonstrated in the VarGeno and LAVA frameworks (Sun and Medvedev 2019; Shajii et al. 2016). These approaches are fast, but rely upon indexes of kmers extracted from the reference genome and SNP databases, thus reducing their flexibility for kmers of different length and source. A separate approach is taken by Dolle et al., wherein the entire 1000 Genomes dataset is compressed into an FM-index and queried with kmers spanning polymorphic sites, thus demonstrating the utility of scanning unmapped reads for predefined kmers of interest. The "reverse mapping" highlighted in their approach was applied to aggregated data, but the concept can be extended to the analysis of individual genomes if implemented in a flexible way for diverse types of queries.

Within the paradigm of indexing reads and performing reverse mapping, other useful operations can be performed with increased utility, especially in cases with a diverse set of informative sequences. One example of this is within RNA sequencing (RNA-seq), where analysis of cancer RNA-seq datasets can reveal the presence of viral pathogens within patient data (Klijn et al. 2015). Several tools have been developed to specifically detect these viral pathogens from sequencing data including viGEN (Bhuvaneshwar et al. 2018) and VirTect (Xia et al. 2019). However, they are hampered by a computationally expensive iterative mapping procedure which first maps against the human reference genome and then subsequently maps against viral genome collections. Other methods, such as Centrifuge (Kim et al. 2016) and Kraken2 (Wood, Lu, and Langmead 2019), rely upon kmer searches against large viral and bacterial databases. Both of these methods are powerful, but come with drawbacks of flexibility and reliance upon phylogenetic relationships between target sequences. Specifically, they require re-indexing of search databases for different query lengths or when the target sequences change. Nevertheless, these tools are broadly used and thus serve as good comparators for efficacy, as they have both been demonstrated to have utility in detecting viral pathogens within cancer RNA-seq datasets by examining kmer content. (https://www.sevenbridges.com/centrifuge/).

Combining the current drive to decrease our reliance upon linear reference genomes, and the wealth of demonstrated utility of reverse mapping approaches, we developed FlexTyper.

FlexTyper is a computational framework which enables the flexible indexing and querying of raw next generation sequencing reads. We show example usage scenarios for FlexTyper by demonstrating the high accuracy of reference-free genotyping of SNPs in single samples, and the ability to identify foreign pathogen sequences within short-read datasets. We hope the flexibility afforded by the framework underpinning FlexTyper will fuel the emerging trend away from the necessity for a static reference genome that currently lay at the heart of the majority of genomic analysis tools.

Design and Implementation

Overview of FlexTyper

Usage can be broken down into three steps: 1) query generation, 2) indexing the raw reads, and 3) querying against the FM-index (Figure 1). For query generation, we allow for both custom user query generation, as well as pre-constructed queries from useful databases, such as CytoScanHD array probe queries. Custom queries designed to capture genomic loci can be generated by pairing a user-provided VCF (format v4.3) with a reference genome fasta file. For the capture of potential pathogen sequences, we also allow query generation from one or more fasta files. The files produced from query generation are used as input for subsequent index query operations. The second step is the production of an FM-index from a set of short-read sequences in fastq format. This process includes reverse-complementing the entire read file, and concatenating the transformed reads with the original set. This is done in order to prevent the need to scan for the reverse complement of the query kmers . The third step is the core FlexTyper search algorithm which takes the query input file, generates search kmers, and scans the FM-index for matches. This step creates an output with matching format to the input file, with appended counts of matching reads for each query. A detailed breakdown of these three components is described below.

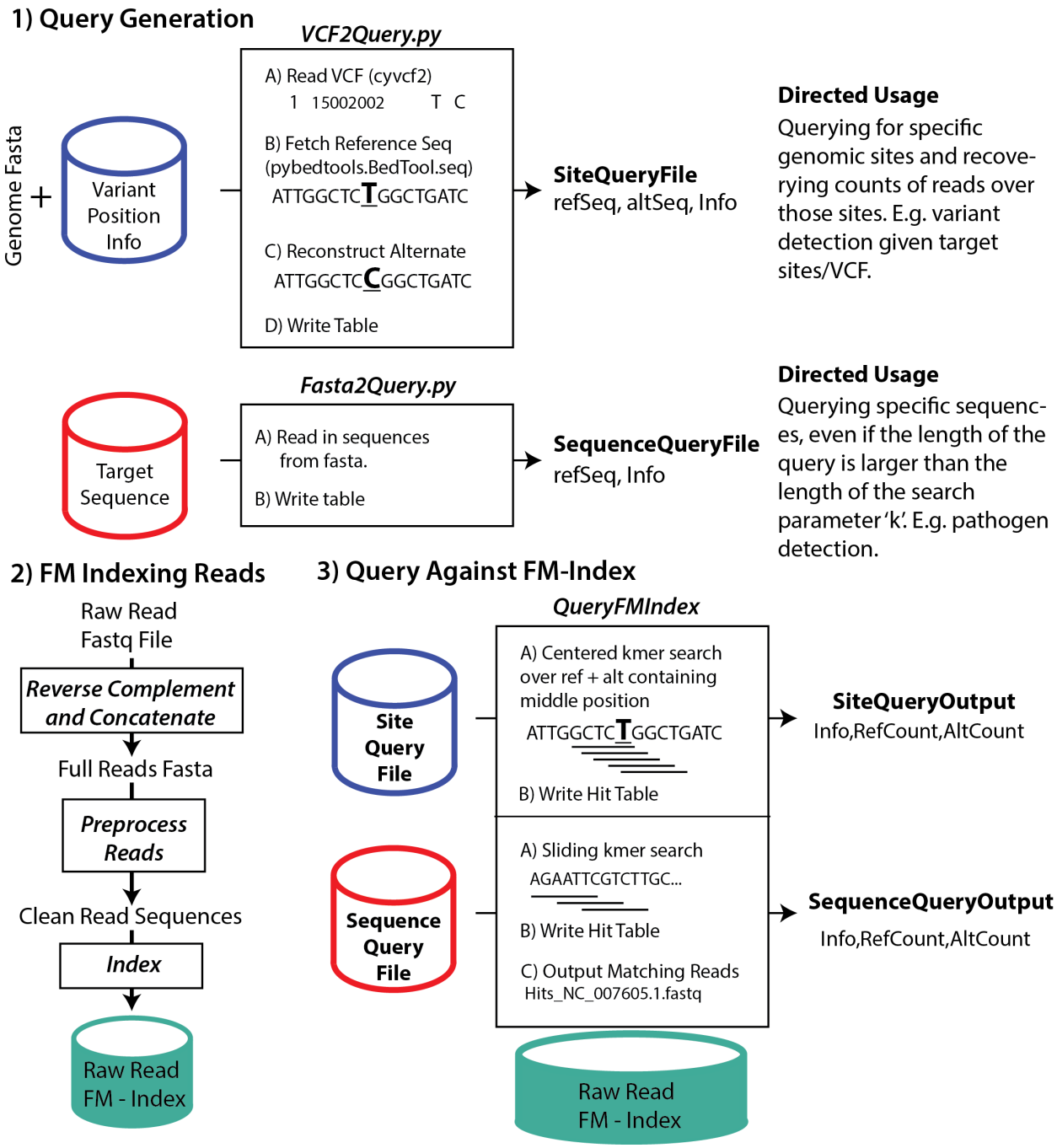


Fig 1 - Overview of FlexTyper

FlexTyper has three primary components: query generation, FM indexing reads, and querying against the FM-index. Query generation includes the capacity to translate VCF files into query files given a reference genome file (Genome Fasta), or to directly create queries from fasta sequences including pathogen genome sequences. Modules VCF2Query.py and Fasta2Query.py facilitate this process. The second component involves creating an FM-index of the raw reads, after reverse complementing and concatenating the read set and performing optional preprocessing steps. The third component executes the queries against the FM-index to produce output files with counts of reference and alternate sequences within the query files.

Query generation

FlexTyper supports flexible query generation giving users the capacity to query for any target sequence or allele within their read dataset. Query files can be generated from an input fasta and VCF file (VCF2Query.py), or directly from a fasta file (Fasta2Query.py). Potentially useful queries, including those presented here, are provided online and include all sites from the CytoScanHD chromosomal microarray, and ancestry discriminating sites (Pedersen and Quinlan 2017). These predefined query sets are available through git-lfs in the online FlexTyper github repository (https://github.com/wassermanlab/OpenFlexTyper). If users wish to directly query a short-read dataset with a set of predetermined kmers, they can provide the kmers as a fasta file and set the *k* parameter to the length of the kmers in the file.

FM-index creation

Generating the FM-index from short-read sequencing datafiles is performed in two steps; preprocessing and indexing. The focus of our work is not on the algorithms used to construct the FM-index, and hence we use two existing utilities to generate a compatible FM-index for FlexTyper. The toolkit Seqtk is used for reformatting compressed fastq files by removing quality scores and non-sequence information to create a sequence-only fasta format, and append this with the reverse complement of the reads. The output fasta file is then processed using the SDSL-Lite library to generate the FM-index. SDSL builds a suffix array that is used to generate the BWT of the input string, which is then compressed using a wavelet tree and subsampled. The resulting compressed suffix array is streamed to a binary index file. As the memory requirements for indexing large files can be burdensome, we support an option to split the input file and index each chunk of reads independently. Downstream search operations support the use of multiple indexes.

Query against FM-index

Querying the FM-index for user selected sequences can be conceptually divided into four steps: 1) kmer generation; 2) kmer filtering; 3) kmer searching; and 4) result collation (Figure 2).

There are two primary methods of kmer generation for a query; a centered search where the middle position of the query is included in all kmers, and a sliding search which starts at one end of the query and uses a sliding window approach to generate the kmers (Figure 2). Centered search can be used for genotyping or estimating coverage over a single position, and the sliding search can be used to count reads which match to any part of a query sequence. The *--ignore-duplicates* parameter filters query kmers by ignoring kmers that occur in multiple query sequences. After filtration, the kmers are searched for within the FM-index using C++ multithreading and asynchronous programming, using either a single thread on a single index, multiple threads on a single index, a single thread on multiple indexes, or multiple threads on multiple indexes (Figure 2). Importantly, asynchronous programming allows the number of threads used during searching to be increased beyond the number of available CPUs. The output from this search process is a collated results map containing the positions of each kmer within the FM-index. These positions are translated to read IDs, and finally collapsed into query counts using the kmer-to-query mapping. Importantly, if multiple kmers from the same query hit the same read, they are recorded as a single count at the query level. For cases of multiple indexes being searched in parallel, the kmer searching and assignment to the query count is performed independently and then merged to produce a final query count table.

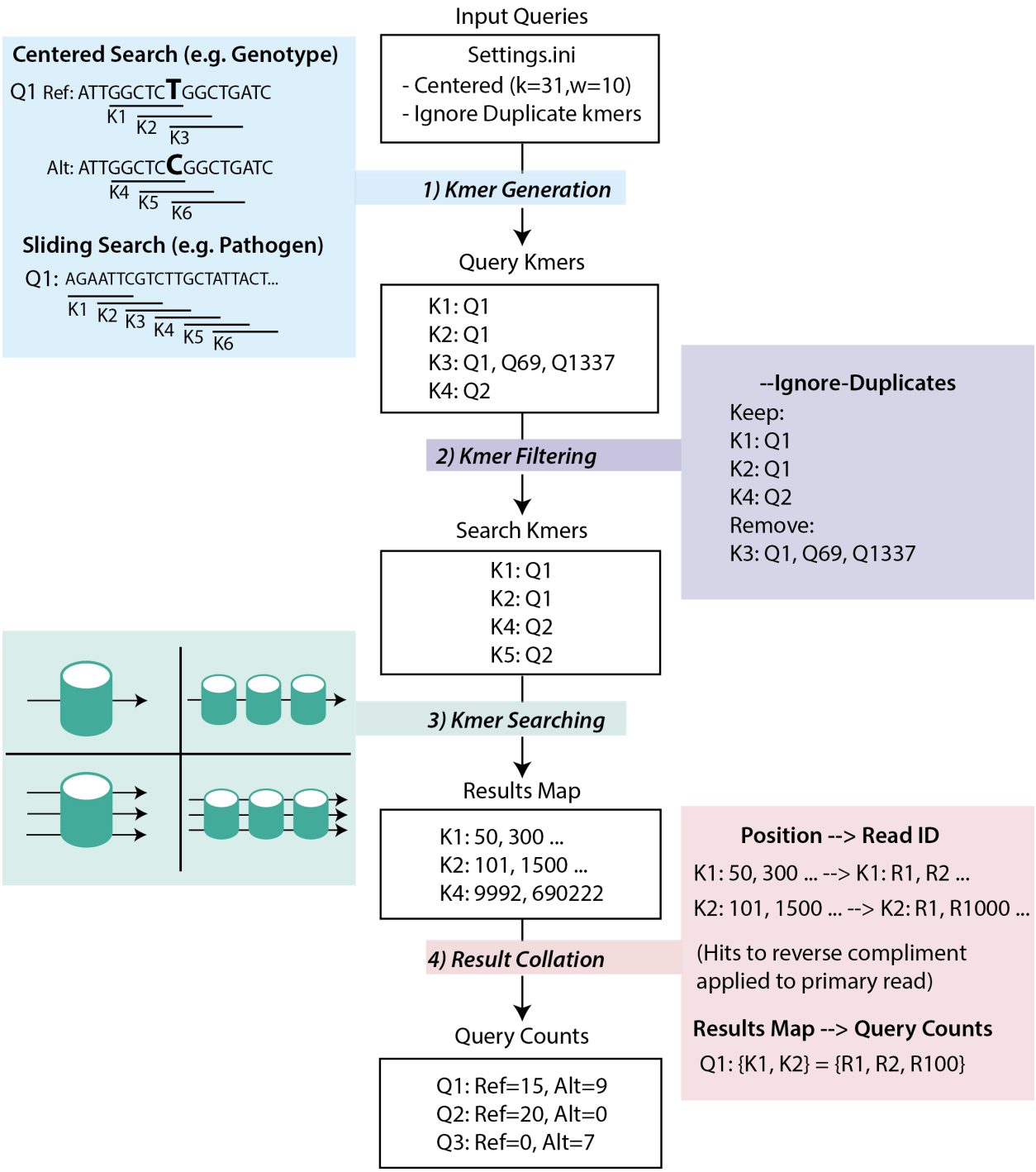


Figure 2 - Query Search Workflow

Workflow for query search against the FM-index, starting with input queries and settings defined in Settings.ini file. In this example, it sets a centered search with ignoring duplicate kmers enabled. 1) Kmer generation has two modes, centered search and sliding search. For a centered search, the position of interest lies in the middle of the query, and kmers are designed to overlap that central position with defined length (k) and step (w). 2) If the ignore-duplicates option is set, kmers

collated from the query set are filtered to remove any kmers which were found in multiple query sequences. 3) The filtered kmers are then searched for within the FM-index (left two panels) or multiple indexes (right two panels) of the read set. This can be done using single (top two panels) or multiple (bottom two panels) threads. 4) The results corresponding to a position within the FM-index are then translated back into reads, with hits on reverse complement reads assigned to the primary read, and collapsed into a set for each query. The final counts are reported per query.

Post-processing of results into downstream formats

The output tables from the search process for genotyping can be translated into useful formats for downstream analysis using the fmformatter scripts (https://github.com/wassermanlab/OpenFlexTyper/tree/master/fmformater). Currently, there is the capacity to output genotype calls in VCF, 23andMe, or Ancestry.com format. Genotype calls are derived here using a basic approach which assigns genotypes given a minimum read count parameter as follows:

Alt < minCount && Ref > minCount: Homozygous reference, 0/0

Alt > minCount && Ref > minCount: Heterozygous alternate, 0/1

Alt > minCount && Ref < minCount: Homozygous alternate, 1/1

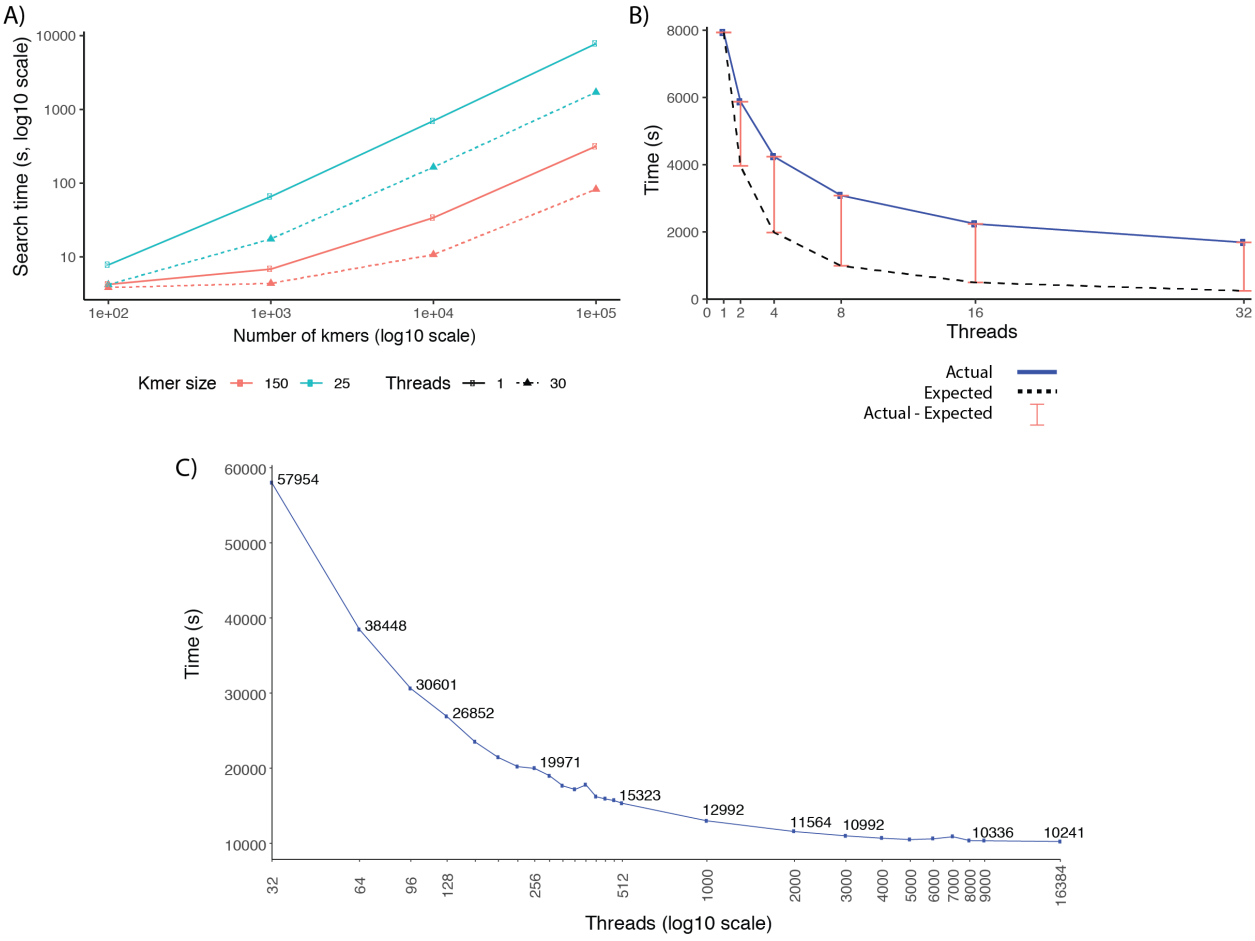
For searches which do not pertain to genotyping, the output tab-separated files can be used as count tables for observed query sequences.

Results

Performance metrics for indexing and querying

We used a human whole genome sequencing (WGS) sample to demonstrate the indexing and querying capacities of FlexTyper. Our indexing strategy utilizes open source tools to build the FM-index on a high memory CPU, with at least 1000GB of RAM. While index creation optimization was not the focus of this work, indexing is feasible on standard systems with ~256GB of RAM, as long as the input read dataset is smaller than ~20GB (Supplemental Table S1). Since our search function allows for multiple separate indexes, we incorporated the ability to sub-divide larger read sets into multiple smaller read sets that can be indexed in parallel. For querying, we generated a set of kmers designed from probes on the CytoScanHD Illumina genotyping microarray with a centered search process (Figure 2) for varying kmer lengths (Figure 3A). The CytoScanHD genotyping microarray, chosen for its broad usage in the field of human genetics, has probe sequences designed to uniquely detect well-characterized SNPs. There is a noticeable benefit from multithreading FlexTyper, which we demonstrated by isolating the kmer searching process across 1 to 32 threads (Figure 3A). As the number of threads increases, we observe a continuous decrease in search time, and by comparing between observed and expected performance, the performance advantage gained from additional threads does not plateau at 32 threads (Figure 3B). As the software is written using asynchronous programming, we tested the upper bound on allocated data threads given a fixed set of 32 CPUs on a single machine with 256GB of RAM. For this analysis, we used an extended set of queries from the CytoScanHD SNP set, for a total of ~6.4 million kmers. We increased the threads from 32 to 512 stepping by 32 and while we do see a decrease on the improvement in speed, there is still a benefit of additional threads (Figure 3C). To see where the benefit of increased threads plateaus, we increased threads from 1000 to 16,384 and witnessed little speed increase (<5 minutes) between 5000 and 16,384 threads (Figure 3C). Thus, we define the upper bound on data threads for a machine with 32 CPUs and 250GB of RAM to be ~5000 data threads, for a query set of ~6.4 million kmers. It is possible that higher thread counts may improve performance for larger query sets and more powerful computers. Lastly, to highlight the clear advantage over non-indexed methods, we compared

263 FlexTyper to popular non-indexed algorithms achieving a decrease in search time by roughly three
264 orders of magnitude when using FlexTyper (Supplemental Table S2).
265



266

267 Figure 3 - Search speeds for FlexTyper
268 A) FlexTyper search time with kmers of size 25 (blue) and 150 (red), increasing in number from
269 10-100,000, using one (solid) or 30 (dashed) threads. B) Increasing the number of threads from 1
270 to 32, for 100,000 kmers of length 25 (solid blue line). Expected values calculated by dividing
271 single thread time by the additional number of threads (dashed black line), with difference between
272 actual and expected plotted (red vertical bar). C) Hyperthreading results for the time (in seconds)
273 vs. thread counts from 32 up to 16,384 (log10 scaled x-axis)

274 Genomic coverage and genotype detection within human WGS data

275 Knowing whether a given kmer is present or absent from a human WGS datafile (in this
276 instance genome Illumina short-read, paired-end data) can have utility for estimating the depth of
277 coverage for a target region and genotyping SNPs. FlexTyper has the capacity to compute depth
278 of coverage or genotype SNPs from WGS data for both predefined and user-supplied loci. We

demonstrate this capacity for genomic sites using the probe sequences from the CytoScanHD microarray, as well as a subset of previously collated population discriminating SNPs (Pedersen and Quinlan 2017). Using these loci, we created query files with a reference and alternate query sequence centered on the biallelic site (Supplemental Methods).

We first sought to test the read recovery capacity of FlexTyper compared to an alignment based method which we call BamCoverage. The BamCoverage method involves mapping the reads to the reference genome, and then extracting per-base read coverage over a specific reference coordinate. BamCoverage utilizes the pysam package to extract read pileup over positions defined by the FlexTyper input query file (Supplemental methods). Using the CytoScanHD SNP set, we found a high concordance between the read counts from FlexTyper and the depth of coverage from aligned reads (Figure 4A). The vast majority, 780,178/797,653 or 97.8%, of sites differed by less than 10 between FlexTyper and BamCoverage (Figure 4B). This discrepancy is similar for both reference and alternate alleles, which is important since most genotyping models assume relative contributions of observed alleles for genotype calling. There were 16,282 sites with a delta, (Δ = FlexTyper - BamCoverage), greater than 10, and 4,256 sites with a delta greater than 100. We manually investigated a few of these sites which were overcounted by FlexTyper by more than 100 and found that they are being overcounted due to kmers mapping to multiple possible locations. Comparing these over-counted hits with delta greater than 100 to previously defined repeat regions shows that 4189/4256 or 98.4% of the overcounted sites overlapped with predefined repeats (Trost et al. 2018). The uniqueness of kmers is important for accurate read counting, thus it is recommended to filter such regions when using FlexTyper for genotyping or depth profiling. Lastly, by examining the recovery of reads across the chromosome between FlexTyper and the read alignment approach, it's clear that FlexTyper can accurately capture relative sequence abundance with relevance to copy number variant calling applications (Fig 4D).

Next, we investigated whether FlexTyper can accurately recover genotypes at the SNP sites profiled from the chromosomal microarray. The genotyping approach we use leverages a minimum count from the reference and alternate allele to assign heterozygous, homozygous alternate, or homozygous reference genotypes (Supplemental Methods). We applied this basic genotyping algorithm to both FlexTyper and BamCoverage counts to produce a VCF file. These genotypes were compared to an alternate pipeline which uses reference-based mapping and sophisticated variant calling using DeepVariant (Poplin et al. 2018). For the 797,653 SNPs on the CytoScanHD

microarray, all three methods agree on 99.2% (791,063/797,653) of the sites. For the sites where there was disagreement, we see an overlap with the repeat regions of 5004/5586 or 89.6%, affirming that these repeat regions are responsible for the majority of discordant genotypes. We further demonstrate the accuracy of these genotypes by indexing nine WGS samples from the Polaris project representing diverse populations including three African, three Southeast Asian, and three European individuals (S. Chen et al., n.d.). After indexing, we queried the samples for population discriminating sites and then genotyped the output table to produce a VCF file. The output VCFs were then used within the Peddy tool, and a principal component analysis was performed to predict the ancestry of the samples (Pedersen and Quinlan 2017). In all nine cases the population was correctly determined, as well as the relatedness inference for the three trios (Figure 4E, Figure S1). Interestingly, we observed a discrepancy between the listed sex for the child of the European trio, individual HG01683, and the inferred sex from FlexTyper and Peddy (Figure 4F). We followed up on this observation and revealed that the individual is not an XY male, but rather an XXY individual. Taken together, FlexTyper has the capacity to provide accurate counts of observed reads matching a query sequence, with relevant utilities such as copy number estimation, sample identification, ancestry typing, and sex identification.

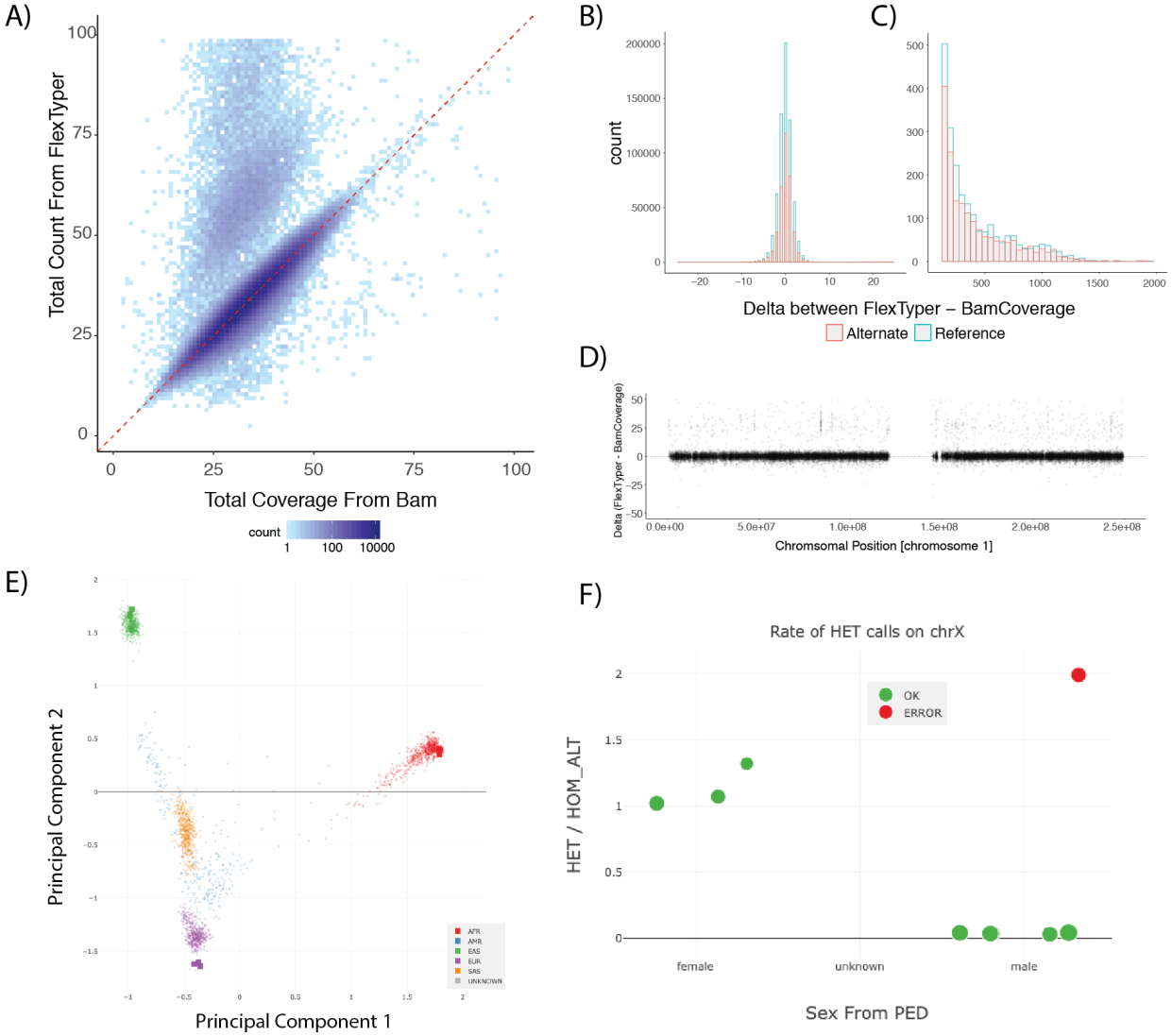


Figure 4 - WGS Genotyping using FlexTyper

A) FlexTyper read count compared to the total coverage from BAM file over SNP sites represented on the CytoScanHD microarray. B) Histogram showing the delta, (Δ = FlexTyper - BamCoverage), in read count for both the alternate (red) and reference (blue) alleles. C) Histogram of the same delta as B) but with an extended axis from 100-2000, showing the frequency of over-counting for sites using FlexTyper. D) Scatter plot showing the delta (Δ = FlexTyper - BamCoverage) on the y-axis, plotted across chromosome 1 on the x-axis. E) Principal component analysis showing projection of FlexTyper-derived SNP genotypes from nine individuals of Asian (green), African (red) and European (purple) ancestry. Squares denote FlexTyper genotypes, points denote existing data from the 1000 Genomes project provided by Peddy. F) Sex-typing for these Polaris samples showing the ratio of heterozygous to homozygous sites on the X chromosome (y-axis) for individuals for the defined sexes as male (right) and female (left). Each individual is labeled as green (correctly sex-labeled) or red (incorrectly labeled).

Testing for the presence of pathogen sequences in RNA-seq

To demonstrate the capacity of FlexTyper to detect pathogens from RNA-sequencing data, we generated synthetic reads from four relevant viral genomes including Epstein-Barr virus (EBV), Human Immunodeficiency virus type 1 (HIV-1), and two Human Papilloma virus strains 68b (HPV FR751039) and 70 (HPV U21941) (Supplemental Methods). We first examined the impact of various FlexTyper parameters on the recovery rate of pure, simulated read sets for each of the four viruses and one human blood RNA-seq dataset from the Genome England project (https://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-6523/samples/) (Table S3). Importantly, varying the parameters *k* (length of search substring) and *w* (step-size) change the specificity and sensitivity of read recovery. When *k* is set to 15 (a short kmer), there are roughly 1 million off-target hits to the viral genomes for the pure human RNA-seq file (Table S3). Next, we demonstrated that the kmer uniqueness setting only guarantees that identical kmers cannot appear across queries. Thus, if query specificity is a priority then setting the *w* parameter to 1 will produce results with the least amount of cross-query assignment. By exploring these parameters, we show that all simulated reads can be recovered with parameters of 30 and 5 for k and w respectively, with low off-target assignment.

Next, to simulate patients infected by one of the four viruses, we spiked-in simulated pathogen reads with the human RNA-seq dataset. Using the optimized parameters derived above, we are able to detect each virus in the patient sample even at low concentrations (Table S4). We further demonstrate the capacity for FlexTyper to discriminate between spiked-in virus samples by mixing the viruses at differing concentrations (read counts) within the human RNA-seq dataset (Table 1). FlexTyper was run with two settings by varying the k and w parameters for increased sensitivity (k=31, w=5), or increased speed (k=100, w=25). We compared these results with Centrifuge, a software tool that works with unmapped short-read sequencing data by performing read-length (k=150) kmer searches against a database of viral and bacterial genomes (Kim et al. 2016). For each of the datasets, FlexTyper is able to detect the contaminating pathogen sequences, even in a sample (Patient_5) where we only spiked in the equivalent of a 1x coverage of the viral genome, which equates to roughly 50-1150 reads depending on the size of the viral genome (Table 1). For comparison, Centrifuge results were manually combined for viruses of similar naming schema and presented as the sum of non-unique read hits (Table 1). For each of the samples, both Centrifuge and FlexTyper are capable of detecting the spiked-in pathogens. In all simulations,

consistent with its use of shorter kmers, FlexTyper is more sensitive in its detection capacity, recovering more reads than Centrifuge. In our collation of viral hits for the Centrifuge data, we observed the limitation that for HPV strains, Centrifuge utilizes a comprehensive genome database with hundreds of distinct strains. Thus, retrieving a combined count of HPV sequences within a sequencing dataset is nontrivial and requires collation over hundreds of viral genome hits. In contrast, FlexTyper is able to detect all of the spiked in reads for these viral genomes of interest. This is due to the increased flexibility of FlexTyper, which enables the user to define the relevant pathogens to search for without the need for reconstructing a complex bacterial or viral database, as is the case for Centrifuge. In summary, FlexTyper is more sensitive in its detection capacity than Centrifuge, and the flexibility to *ad hoc* define the pathogen search space could be beneficial in some applications, such as instances when the virus is a novel strain.

	EBV				HIV-1					
Sample	Expect	FT:30,5	FT:100,25	C	Expect	FT:30,5	FT:100,25	C		
Patient_1	1145	1146	861	538	610	610	462	284		
Patient_2	11450	11454	8290	5427	6100	6099	4359	2797		
Patient_3	114500	114502	83504	54352	61000	61000	43648	28206		
Patient_4	1145000	1144990	833955	543924	62	62	45	29		
Patient_5	1146	1146	861	538	62	62	45	29		
	Total HPV				U21941			FR751039		
Sample	Expect	FT:30,5	FT:100,25	C	Expect	FT:30,5	FT:100,25	Expect	FT:30,5	FT:100,25
Patient_1	57200	60812	40930	2650	5200	8475	3755	52000	52337	37175
Patient_2	52052	55371	37443	1470	52000	52005	37406	52	3366	37
Patient_3	572	615	422	25	52	92	38	520	523	384
Patient_4	5720	6084	4152	273	520	851	388	5200	5233	3764
Patient_5	104	112	75	3	5	57	38	52	55	37

Table 1 - Performance comparison for simulated spike-in pathogens.
Each of the samples, (Patient_1 - Patient_5), with expected (simulated known counts) vs. observed counts for Centrifuge (C) and FlexTyper with k=30/w=5 (FT:30,5) and k=100/w=25 (FT:100,25). Each quantified viral strain includes Epstein Barr Virus (EBV), Human Immunodeficiency Virus-1 (HIV-1), total Human Papillomavirus (HPV), and two strains of HPV (type 70 and 68b). The maxOcc parameter was set to limit the number of hits to one million and non-unique kmers were allowed. For centrifuge, sub-strain HPV counts were not feasible so counts were aggregated over all papilloma viral strains in the output report file per patient.

Discussion

Here we presented FlexTyper, a flexible tool which enables exploratory analysis of short read datasets without the need for alignment to a reference genome. Our framework allows for the custom generation of queries, giving the user total control to perform searches relevant to the problem at hand. We demonstrated three applications, including depth of coverage analysis, accurate SNP genotyping, and sensitive detection of pathogen sequences. FlexTyper is available for the creative use of genomics researchers.

The rapid and accurate recovery of read depth enables innovative usage of FlexTyper in the space of copy number variant profiling. We demonstrated that we can reproduce the depth of coverage of a genomic region without the need for reference-based mapping. As microarrays are replaced by genome sequencing assays, we envision that FlexTyper could be extended to reproduce microarray-style outputs. Further, we show that when genomic queries with counts higher than the expectation arise, these events correspond to repetitive genomic sequences. As such, FlexTyper may not only enable the recovery of read depth in an accurate manner, but it can also inform the quality of a sequence query as a "unique probe" for assessing genomic copy number.

The genotyping case study highlights how pre-alignment analysis of genome sequence data can provide rapid insights into the properties of a sample. SNP genotyping was accurate across the genome, allowing rapid identification of sample ancestry, sample relatedness in the trio setting, and sample sex typing using Peddy (Pedersen and Quinlan 2017). Interestingly, applying Peddy to the ouput of FlexTyper for open source trio data from the Polaris project revealed a mislabeling of the sex for individual HG01683, which was reported and subsequently ammended in the online data repository (https://github.com/Illumina/Polaris/wiki/HiSeqX-Kids-Cohort). Since ancestry and sex information can inform choices in downstream data processing, identifying these discrepancies between labeled sex and inferred sex in a data-driven manner is a critical step of pre-alignment informatics. For instance, mapping against the sample-matched sex chromosomes has been shown to improve performance (Webster et al. 2019; Olney et al., n.d.). As such, using FlexTyper, in combination with Peddy, on diverse datasets prior to reference-guided read alignment will lead to improved results from mapping-based pipelines.

The importance of pathogen identification is increasingly recognized. In both cancer profiling (Klijn et al. 2015) and public health studies (Gardy, Loman, and Rambaut 2015), rapid determination of the presence of pathogen sequences could obviate the need for full reference mapping. Some existing tools designed for viral detection in sequencing data rely upon pre-indexed databases of viral and bacterial sequences, sometimes including a phylogenetic relationship between genomes within the index (Xia et al. 2019; Wood, Lu, and Langmead 2019; Kim et al. 2016). One such approach, Centrifuge, has been applied to cancer genomes to confirm the presence of viral pathogens. We demonstrated that our approach compares favorably to Centrifuge, with a more sensitive detection level, due to the ability to search for kmers shorter than the read length and the advantage of fine-tuned control over the searchable database. Here we only searched for viral pathogens of interest, although other specific pathogen queries could be performed, such as the presence of antibiotic resistance genes within a patient RNA-seq sample.

We anticipate that the research community will identify diverse and creative uses for "reverse mapping" analysis with FlexTyper, but a few approaches are apparent to us. It is feasible to genotype complex structural variants by searching for sequences overlapping breakpoints, such as those observed in a subpopulation, or events recurrently found in cancer (Li et al. 2020; Sudmant et al. 2015). Within RNA-seq data, querying for exon-exon splice junctions in a rapid manner can allow isoform quantification, as has been previously demonstrated (Patro, Mount, and Kingsford 2014; Bray et al. 2016). Further, a recent report showed the utility of kmer-counting methods in resolving copy number variants within paralogous loci and genes (Shen, Shen, and Kidd 2020). Another group showed the advantage of examining depth of coverage at specific sites across the paralogous genes in Spinal Muscular Atrophy (X. Chen et al. 2020) As FlexTyper is well suited for specific sequence recovery operations, scanning with preselected query sequences such as defined by these studies can enable rapid detection (X. Chen et al. 2020). All of these proposed applications help tackle challenges which are currently a burden for traditional reference-based mapping approaches.

We focused this report on the utility of kmer searches against indexed read sets, but recognize that speed and computational resources are an important consideration for adoption of the method. One obvious (but transient) constraint on the utility of FlexTyper is the generation of the FM-index for the sequencing reads. As the FM-index is critical to many aspects of genome-scale sequence analyses, there are diverse efforts to develop novel indexing strategies, such as

optimizing FM-index construction using GPUs (Chacón et al. 2015) and creating efficient construction algorithms (N. Chen, Li, and Lu 2018; Labeit, Shun, and Blelloch 2017). Further, the nature of the mapping procedure holds promise with massive parallelization approaches, including those involving GPU acceleration (Hung et al. 2018). Moving forward, accelerations to the FM-index generation and reverse mapping approach will result in faster genomic analysis pipelines than is currently possible with alignment based methods.

Looking to the future, we see the kmer-searching approach of FlexTyper as having great utility when used in conjunction with emergent graph-based representations of the reference genome (Kehr et al. 2014; Paten et al. 2017; Kaye 2016). Whether users seek to select a population specific reference graph as the basis for read mapping, or to introduce Bayesian priors (edge weighting) within a pan-population reference graph, knowledge of population markers spanning chromosomes will be required to inform the processes. Furthermore, it is our expectation that graph-based mapping methods will ultimately use read-based FM-indices, as indexing the reference graph imposes restrictions on the graph structures that can be used and the types of variations that can be incorporated (Ghaffaari and Marschall 2019; Paten, Novak, and Haussler 2014). As the graph-based algorithms mature, approaches such as FlexTyper which enable reverse mapping of sequences against a set of indexed reads will be instrumental in the initial steps of genome analysis pipelines, and in the resolution of challenging regions of the genome.

References

Ballouz, Sara, Alexander Dobin, and Jesse A. Gillis. 2019. "Is It Time to Change the Reference Genome?" *Genome Biology* 20 (1): 159.

Bhuvaneshwar, Krithika, Lei Song, Subha Madhavan, and Yuriy Gusev. 2018. "viGEN: An Open Source Pipeline for the Detection and Quantification of Viral RNA in Human Tumors." *Frontiers in Microbiology* 9 (June): 1172.

Bray, Nicolas L., Harold Pimentel, Páll Melsted, and Lior Pachter. 2016. "Near-Optimal Probabilistic RNA-Seq Quantification." *Nature Biotechnology* 34 (5): 525–27.

Chacón, Alejandro, Santiago Marco-Sola, Antonio Espinosa, Paolo Ribeca, and Juan Carlos Moure. 2015. "Boosting the FM-Index on the GPU: Effective Techniques to Mitigate Random Memory Access." *IEEE/ACM Transactions on Computational Biology and Bioinformatics / IEEE, ACM* 12 (5): 1048–59.

Chen, N., Y. Li, and Y. Lu. 2018. "A Memory-Efficient FM-Index Constructor for Next-Generation Sequencing Applications on FPGAs." In *2018 IEEE International Symposium on Circuits and*

497 *Systems (ISCAS)*, 1–4.
498 Chen, Sai, Peter Krusche, Egor Dolzhenko, Rachel M. Sherman, Roman Petrovski, Felix Schlesinger,
499 Melanie Kirsche, et al. n.d. "Paragraph: A Graph-Based Structural Variant Genotyper for Short-Read
500 Sequence Data." https://doi.org/10.1101/635011.
501 Chen, Xiao, Alba Sanchis-Juan, Courtney E. French, Andrew J. Connell, Isabelle Delon, Zoya Kingsbury,
502 Aditi Chawla, et al. 2020. "Spinal Muscular Atrophy Diagnosis and Carrier Screening from Genome
503 Sequencing Data." *Genetics in Medicine: Official Journal of the American College of Medical*
504 *Genetics*, February. https://doi.org/10.1038/s41436-020-0754-0.
505 Dilthey, Alexander, Charles Cox, Zamin Iqbal, Matthew R. Nelson, and Gil McVean. 2015. "Improved
506 Genome Inference in the MHC Using a Population Reference Graph." *Nature Genetics* 47 (6): 682–
507 88.
508 Dolle, Dirk D., Zhicheng Liu, Matthew Cotten, Jared T. Simpson, Zamin Iqbal, Richard Durbin, Shane A.
509 McCarthy, and Thomas M. Keane. 2017. "Using Reference-Free Compressed Data Structures to
510 Analyze Sequencing Reads from Thousands of Human Genomes." *Genome Research* 27 (2): 300–
511 309.
512 Feuk, Lars, Andrew R. Carson, and Stephen W. Scherer. 2006. "Structural Variation in the Human
513 Genome." *Nature Reviews Genetics*. https://doi.org/10.1038/nrg1767.
514 Gardy, Jennifer, Nicholas J. Loman, and Andrew Rambaut. 2015. "Real-Time Digital Pathogen
515 Surveillance — the Time Is Now." *Genome Biology*. https://doi.org/10.1186/s13059-015-0726-x.
516 Ghaffaari, Ali, and Tobias Marschall. 2019. "Fully-Sensitive Seed Finding in Sequence Graphs Using a
517 Hybrid Index." *Bioinformatics* 35 (14): i81–89.
518 Hung, Che-Lun, Tzu-Hung Hsu, Hsiao-Hsi Wang, and Chun-Yuan Lin. 2018. "A GPU-Based Bit-Parallel
519 Multiple Pattern Matching Algorithm." *2018 IEEE 20th International Conference on High*
520 *Performance Computing and Communications; IEEE 16th International Conference on Smart City;*
521 *IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*.
522 https://doi.org/10.1109/hpcc/smartcity/dss.2018.00205.
523 Kaye, Alice. 2016. Methods for the graphical representation of genomic sequence data. USPTO
524 20160342737:A1. *US Patent*, filed May 20, 2016, and issued November 24, 2016.
525 https://patentimages.storage.googleapis.com/0e/37/3d/b2218ac644a833/US20160342737A1.pdf.
526 Kehr, Birte, Kathrin Trappe, Manuel Holtgrewe, and Knut Reinert. 2014. "Genome Alignment with
527 Graph Data Structures: A Comparison." *BMC Bioinformatics* 15 (1): 99.
528 Kim, Daehwan, Li Song, Florian P. Breitwieser, and Steven L. Salzberg. 2016. "Centrifuge: Rapid and
529 Sensitive Classification of Metagenomic Sequences." *Genome Research* 26 (12): 1721–29.
530 Klijn, Christiaan, Steffen Durinck, Eric W. Stawiski, Peter M. Haverty, Zhaoshi Jiang, Hanbin Liu,
531 Jeremiah Degenhardt, et al. 2015. "A Comprehensive Transcriptional Portrait of Human Cancer Cell
532 Lines." *Nature Biotechnology* 33 (3): 306–12.
533 Labeit, Julian, Julian Shun, and Guy E. Blelloch. 2017. "Parallel Lightweight Wavelet Tree, Suffix Array
534 and FM-Index Construction." *Journal of Discrete Algorithms* 43 (March): 2–17.
535 Levy-Sakin, Michal, Steven Pastor, Yulia Mostovoy, Le Li, Alden K. Y. Leung, Jennifer McCaffrey,
536 Eleanor Young, et al. 2019. "Genome Maps across 26 Human Populations Reveal Population-
537 Specific Patterns of Structural Variation." *Nature Communications* 10 (1): 1025.
538 Li, Yilong, Nicola D. Roberts, Jeremiah A. Wala, Ofer Shapira, Steven E. Schumacher, Kiran Kumar,
539 Ekta Khurana, et al. 2020. "Patterns of Somatic Structural Variation in Human Cancer Genomes."
540 *Nature* 578 (7793): 112–21.
541 MacDonald, Jeffrey R., Robert Ziman, Ryan K. C. Yuen, Lars Feuk, and Stephen W. Scherer. 2014. "The
542 Database of Genomic Variants: A Curated Collection of Structural Variation in the Human
543 Genome." *Nucleic Acids Research* 42 (Database issue): D986–92.
544 Nielsen, Rasmus, Joshua S. Paul, Anders Albrechtsen, and Yun S. Song. 2011. "Genotype and SNP
545 Calling from next-Generation Sequencing Data." *Nature Reviews Genetics*.
546 https://doi.org/10.1038/nrg2986.
547 Olney, Kimberly C., Sarah M. Brotman, Valeria Valverde-Vesling, Jocelyn Andrews, and Melissa A.

548 Wilson. n.d. "'Aligning RNA-Seq Reads to a Sex Chromosome Complement Informed Reference
549 Genome Increases Ability to Detect Sex Differences in Gene Expression.'"
550 https://doi.org/10.1101/668376.
551 Paten, Benedict, Adam Novak, and David Haussler. 2014. "Mapping to a Reference Genome Structure."
552 *ArXiv E-Prints*, 1–26.
553 Paten, Benedict, Adam M. Novak, Jordan M. Eizenga, and Erik Garrison. 2017. "Genome Graphs and the
554 Evolution of Genome Inference." *Genome Research* 27 (5): 665–76.
555 Patro, Rob, Stephen M. Mount, and Carl Kingsford. 2014. "Sailfish Enables Alignment-Free Isoform
556 Quantification from RNA-Seq Reads Using Lightweight Algorithms." *Nature Biotechnology* 32 (5):
557 462–64.
558 Pedersen, Brent S., and Aaron R. Quinlan. 2017. "Who's Who? Detecting and Resolving Sample
559 Anomalies in Human DNA Sequencing Studies with Peddy." *American Journal of Human Genetics*
560 100 (3): 406–13.
561 Poplin, Ryan, Pi-Chuan Chang, David Alexander, Scott Schwartz, Thomas Colthurst, Alexander Ku, Dan
562 Newburger, et al. 2018. "A Universal SNP and Small-Indel Variant Caller Using Deep Neural
563 Networks." *Nature Biotechnology* 36 (10): 983–87.
564 Shajii, Ariya, Deniz Yorukoglu, Yun William Yu, and Bonnie Berger. 2016. "Fast Genotyping of Known
565 SNPs through Approximate K-Mer Matching." *Bioinformatics* 32 (17): i538–44.
566 Shen, Shen, and Kidd. 2020. "Rapid, Paralog-Sensitive CNV Analysis of 2457 Human Genomes Using
567 QuicK-mer2." *Genes*. https://doi.org/10.3390/genes11020141.
568 Sherman, Rachel M., Juliet Forman, Valentin Antonescu, Daniela Puiu, Michelle Daya, Nicholas Rafaels,
569 Meher Preethi Boorgula, et al. 2019. "Assembly of a Pan-Genome from Deep Sequencing of 910
570 Humans of African Descent." *Nature Genetics* 51 (1): 30–35.
571 Sudmant, Peter H., Tobias Rausch, Eugene J. Gardner, Robert E. Handsaker, Alexej Abyzov, John
572 Huddleston, Yan Zhang, et al. 2015. "An Integrated Map of Structural Variation in 2,504 Human
573 Genomes." *Nature* 526 (7571): 75–81.
574 Sun, Chen, and Paul Medvedev. 2019. "Toward Fast and Accurate SNP Genotyping from Whole Genome
575 Sequencing Data for Bedside Diagnostics." *Bioinformatics* 35 (3): 415–20.
576 Trost, Brett, Susan Walker, Zhuozhi Wang, Bhooma Thiruvahindrapuram, Jeffrey R. MacDonald, Wilson
577 W. L. Sung, Sergio L. Pereira, et al. 2018. "A Comprehensive Workflow for Read Depth-Based
578 Identification of Copy-Number Variation from Whole-Genome Sequence Data." *American Journal*
579 *of Human Genetics* 102 (1): 142–55.
580 Webster, Timothy H., Madeline Couse, Bruno M. Grande, Eric Karlins, Tanya N. Phung, Phillip A.
581 Richmond, Whitney Whitford, and Melissa A. Wilson. 2019. "Identifying, Understanding, and
582 Correcting Technical Artifacts on the Sex Chromosomes in next-Generation Sequencing Data."
583 *GigaScience* 8 (7). https://doi.org/10.1093/gigascience/giz074.
584 Wood, Derrick E., Jennifer Lu, and Ben Langmead. 2019. "Improved Metagenomic Analysis with Kraken
585 2." *Genome Biology* 20 (1): 257.
586 Xia, Yuchao, Yun Liu, Minghua Deng, and Ruibin Xi. 2019. "Detecting Virus Integration Sites Based on
587 Multiple Related Sequencing Data by VirTect." *BMC Medical Genomics* 12 (Suppl 1): 19.
588 Yang, Xiaofei, Wan-Ping Lee, Kai Ye, and Charles Lee. 2019. "One Reference Genome Is Not Enough."
589 *Genome Biology* 20 (1): 104.

590

591

592