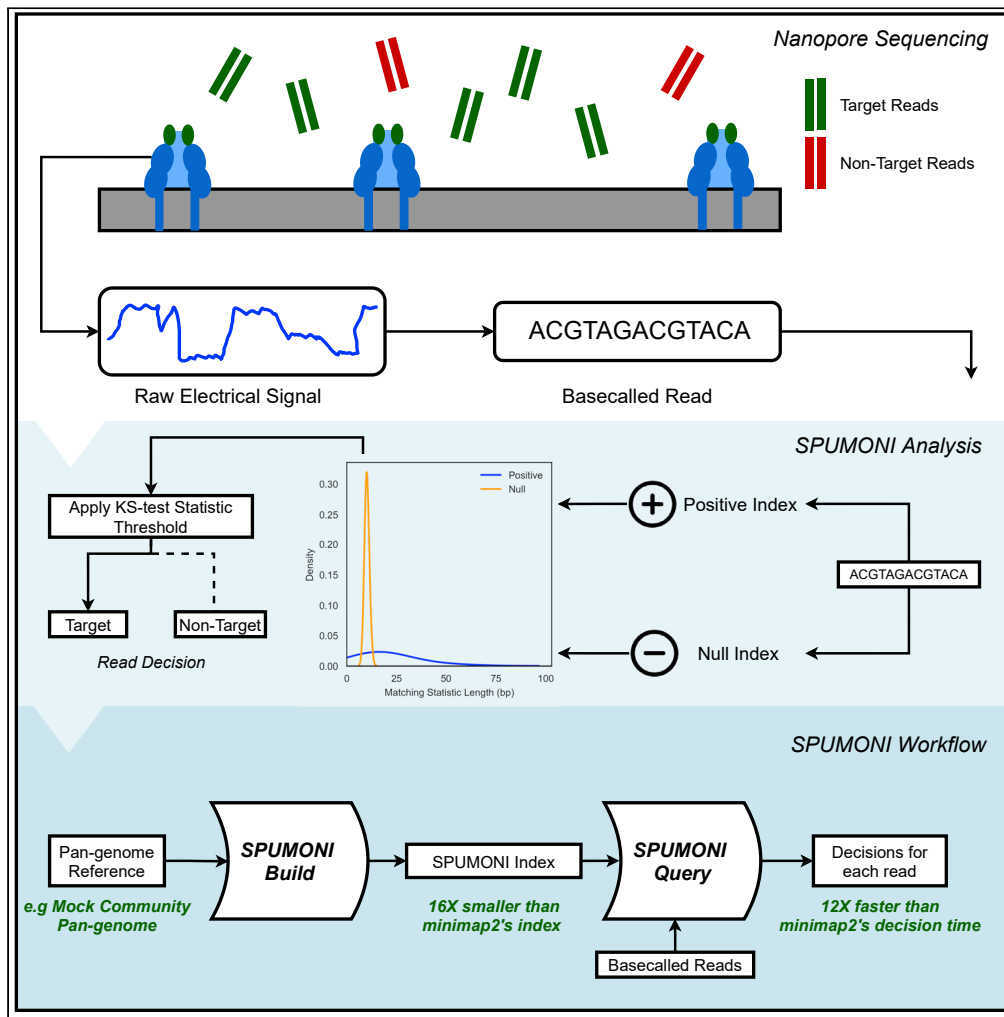## Article

Pan-genomic matching statistics for targeted nanopore sequencing

Omar Ahmed, Massimiliano Rossi, Sam Kovaka, Michael C. Schatz, Travis Gagie, Christina Boucher, Ben Langmead

oahmed6@jhu.edu (O.A.)
langmea@cs.jhu.edu (B.L.)

Highlights

SPUMONI uses an efficient pan-genome index to eject nontarget reads from the nanopore

Read classifications are highly accurate for typical nanopore sequencing error rates

For larger pan-genomes, SPUMONI is faster and uses less memory than minimap2

Enables analyses for strains that are missing or poorly represented in databases

Article

Pan-genomic matching statistics for targeted nanopore sequencing

Omar Ahmed,[1,*] Massimiliano Rossi,[2] Sam Kovaka,[1] Michael C. Schatz,[1] Travis Gagie,[3] Christina Boucher,[2] and Ben Langmead[1,4,*]

SUMMARY

Nanopore sequencing is an increasingly powerful tool for genomics. Recently, computational advances have allowed nanopores to sequence in a targeted fashion; as the sequencer emits data, software can analyze the data in real time and signal the sequencer to eject "nontarget" DNA molecules. We present a novel method called SPUMONI, which enables rapid and accurate targeted sequencing using efficient pan-genome indexes. SPUMONI uses a compressed index to rapidly generate exact or approximate matching statistics in a streaming fashion. When used to target a specific strain in a mock community, SPUMONI has similar accuracy as minimap2 when both are run against an index containing many strains per species. However SPUMONI is 12 times faster than minimap2. SPUMONI's index and peak memory footprint are also 16 to 4 times smaller than those of minimap2, respectively. This could enable accurate targeted sequencing even when the targeted strains have not necessarily been sequenced or assembled previously.

INTRODUCTION

Nanopore sequencing instruments have steadily improved in usability, speed, and accuracy. While it lags sequencing-by-synthesis instruments on base quality, quality has improved steadily, with recent data sets reaching and exceeding 90% accuracy (Wick et al. 2019). Nanopore sequencing is also convenient and flexible; nanopores are readily used outside of laboratories, for example, for analyzing biological species in a human or natural environment with the goal of detecting pathogens or contaminants. They can also be used for several assays, including DNA sequencing, direct RNA sequencing, and the detection of a variety of epigenetic modifications.

Recent computational approaches focus on the problem of allowing nanopores to sequence in a targeted fashion. Oxford Nanopore instruments provide the "Read Until" interface, enabling two-way communication between the sequencer and the control software. The sequencer reports batches of sequencing data, which software can analyze in real time. Importantly, nanopore sequencing has the unique capability where the control software can potentially signal to the sequencer that it should eject the DNA molecule currently in a pore. To eject, the sequencer reverses the voltage across the pore, causing the molecule to reverse direction and exit. The pore is then free to sequence a new molecule. Many such pores – up to 512 per MinION flowcell – are in simultaneous operation; the system can sequence in a targeted manner only as long as the software making ejection decisions can keep up with the aggregate rate of sequencing.

Recently, Payne et al. described the Readfish system (Payne et al., 2020) which combines an existing base caller with the minimap2 read aligner (Li 2018) to align reads to a reference genome in real time and make decisions on whether to eject. The UNCALLED method (Kovaka et al., 2020) is similar but capable of handling the nanopore current signal directly, without first using a base caller. Unlike Readfish, which generally uses a GPU for base calling, UNCALLED is designed to run on a general-purpose CPU. UNCALLED starts by processing the signal to find potential seeds, then maps them to a reference using an FM-index. Finally, it clusters the seeds to identify significant alignments. UNCALLED's performance degrades as the reference is repetitive, for example, if it is a collection of related strains.

Motivated by a need for faster methods which can classify reads against large, repetitive references, we developed SPUMONI. For example, in a typical metagenomics experiment, the exact strain or substrain

[1]Department of Computer Science, Johns Hopkins University, Baltimore, MD, USA

[2]Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL, USA

[3]Faculty of Computer Science, Dalhousie University, Halifax, NS, USA

[4]Lead contact

*Correspondence: oahmed6@jhu.edu (O.A.), langmea@cs.jhu.edu (B.L.)

of a microorganism is unknown before sequencing, and therefore, for optimal targeted sequencing, all strains and substrains need to be incorporated into the reference for identification. SPUMONI takes advantage of the overall repetitiveness of these references by building an *r*-index (Mun et al., 2020) and using the MONI algorithm to calculate matching statistics (MSs) (Rossi et al., 2021). The *r*-index enables efficient indexing of repetitive collections of reference genomes – for example, all of the strains of a bacterial species or several human genome assemblies – while still supporting efficient queries. Importantly, the space required by an *r*-index is proportional to the number of runs in the Burrows-Wheeler transform (BWT) of the reference genomes (defined as *r*) rather than the total length of the reference genomes. When the collection is highly repetitive, *r* grows sublinearly and far more slowly than the total length (Mun et al., 2020).

MONI augmented the *r*-index with an auxiliary data structure enabling more rapid calculation of MSs. An MS at position *i* of a query sequence *P* of length *m* equals the length of the longest prefix of $P[i..m]$ that exactly matches a sequence in the index. MONI efficiently calculates MSs at every position of a query *P*. The first insight of SPUMONI is that these statistics can be used to classify the query sequence; longer MSs indicate a better approximate match to the index.

SPUMONI extends MONI to improve its speed while also making it applicable to the problem of making fast ejection decisions. First, SPUMONI adds a "null index" together with a hypothesis testing framework to make principled ejection choices depending on whether the observed MS lengths are longer than what would be expected by random chance. Second, SPUMONI replaces MONI's "batch" MS-finding algorithm with a faster online algorithm that calculates a different quantity related to the MS, called the "pseudo-matching length (PML)," which we denote as PML (defined in Methods). (SPUMONI stands for Streaming PseUdo MONI.) This optimized PML-finding procedure makes SPUMONI about 3 times faster than MONI, while achieving similar (often greater) accuracy and allowing it to operate on streaming data.

Compared with a minimap2-based approach, SPUMONI can make ejection decisions with respect to a pangenome index more efficiently. When used to eject bacterial strains in a mock community scenario, SPUMONI has similar accuracy as minimap2 but is about 12 times faster. Moreover, its many-strain index is about one-sixteenth the size of minimap2's, and its memory footprint is less than one-fourth the size of minimap2's. When used to eject simulated human reads in a human microbiome scenario, SPUMONI is faster than minimap2 when both use an index consisting of 3 high-quality human reference genomes. In this scenario, SPUMONI's memory footprint and index size are higher, although the sublinear scaling of the *r*-index strategy underlying SPUMONI suggests it will benefit from indexes containing many human genomes.

RESULTS

Method overview

SPUMONI's core insight is that a read's MSs with respect to an index can reveal whether it has a "good" (i.e., long, high identity) approximate match to the index, without having to perform a more costly read alignment. To determine whether the MSs are long enough to indicate an approximate match, SPUMONI compares the observed distribution of MSs – calculated with respect to a "positive index" containing the target sequences – with those obtained from a "null index" containing the reverse (not the reverse complement) of the sequences from the positive index. The reverse sequences serve as a random sequence of the same length as the positive index but where nucleotide frequencies and simple repeat structures such as homopolymers are preserved. As soon as SPUMONI can confidently determine the distributions of MSs from the positive and null indexes are different – possibly having seen only a prefix of the read's full sequence – it can conclude that the read is among the targets in the positive index. SPUMONI uses a Kolmogorov-Smirnov statistic (KS-stat) threshold to make this decision.

By default, SPUMONI does not generate true matching statistics but instead generates an approximation thereof called PMLs. These are described in more detail in Methods. SPUMONI can also generate MSs, which it does in its SPUMONI-ms mode.

Experimental setup

During nanopore sequencing, electrical current data are transmitted from the sequencing instrument to the control software in "chunks," representing about 0.4 s of sequencing (the exact duration is user-defined parameter). As DNA translocates through the pore at about 450 bases per s, each chunk represents about 180 bases of data. Our experiments on both simulated and real reads mimic the situation where we are

processing the first 4 chunks of data delivered by the Read Until API. We chose this time interval as previous work showed it leads to most reads being mapped using minimap2 (Payne et al., 2020; Li 2018). We further assume that the data were already base-called, similar to a previous study (Payne et al., 2020). In practice, the Read Until API delivers batches of current signal, not bases; we address this further in the Discussion. We did not compare our method with UNCALLED (Kovaka et al., 2020) as it is reportedly slower than mini-map2 for large genomes and it starts by processing the current signal, where as we have assumed here that we are given base calls.

With each new batch, both SPUMONI and minimap2 (Li 2018) attempt to classify whether the read has an approximate match to a sequence in the positive index. Importantly, SPUMONI deals with new batches of data in an "online" fashion. That is, SPUMONI can easily suspend and resume its MS/PML computation as it awaits a new batch. This is in contrast to minimap2, which takes full reads as an input so as to perform full-read alignments. Because of this, our evaluation strategy was to run SPUMONI on each batch separately, allowing SPUMONI to possibly make an ejection decision at the end of each of the four batches. For mini-map2, we reran minimap2 on successively longer prefixes of the read as new 180-base batches arrived.

After processing a batch, SPUMONI and minimap2 each apply a threshold to determine if the read matches the positive index with high confidence. In practice, this leads to a decision about whether to eject the read. If the positive index contains depletion targets, a high-confidence match to the positive index indicates the read should be ejected. If the positive index contains enrichment targets, the absence of a high-confidence match after some prescribed period indicates the read should be ejected. For our experiments, the positive index always contains depletion (rather than enrichment) targets. Once a method has decided to eject the read, we cease delivering batches for that read; each method is benchmarked only on the read prefix up to the ejection decision, or up to 1.6 s (720 bases), whichever comes first.

For the minimap2-based approach, we used the standard ONT settings of minimap2 (Li 2018) to align the reads, which are the same settings used by Readfish (Payne et al., 2020). We used an MAPQ threshold to decide whether a read was confidently mapped or not. For nonrepetitive ("genomic") references, we used an MAPQ value of 30 or greater to determine if reads were uniquely mapped or not. For repetitive ("pan-genomic") references, we further checked whether all alignments were to the same species. For further details on the thresholds used, see Matching statistics with r-index.

For evaluation, an instance where a method ejected a read from a genome that was present in the positive index was called a true positive. An instance where a method ejected a read that was not in the positive index was called a false positive. An instance where a method failed to eject a read that was from a positive-index genome was called a false negative.

We performed all the experiments on a computer with a 2.0 GHz Intel Xeon(R) CPU (E7-4830 v4) with 1056 GB of memory. Each tool was run with a single thread, and we recorded the wall clock time and the peak Resident Set Size (RSS) reported by the individual tools. We compared these with the output from GNU time 1.7 program and found no discrepancies.

Evaluations with mock community

We considered a real data set consisting of Oxford Nanopore reads from the ZymoBIOMICS High-Molecular-Weight DNA Mock Microbial community (ZymoMC). We also used a simulated data set of Oxford-like reads derived from the same genomes, but with a software-controlled error rate. The ZymoMC consists of seven bacterial species – *Enterococcus faecalis*, *Listeria monocytogenes*, *Bacillus subtilis*, *Salmonella enterica*, *Escherichia coli*, *Staphylococcus aureus*, and *Pseudomonas aeruginosa* – as well as *Saccharomyces cerevisiae* (yeast). As in prior studies (Kovaka et al., 2020; Payne et al., 2020), we supposed that our goal was to deplete the bacterial reads, leading to proportionally more yeast reads sequenced.

Assessing genomic versus pan-genomic indexes

We hypothesized that a pan-genome index – consisting of many related strains – would allow us to both (a) target a particular strain for depletion or enrichment when that specific strain is not present in the index and (b) target a species as a whole by including many relevant strains or individuals from that species in the index. More specifically, we used the ZymoMC data and supposed that the seven bacterial strains were depletion targets (as in prior work [Kovaka et al., 2020; Payne et al., 2020]). We assessed the following

four strategies: (a) "One Genome w/o Zymo Mock Refs," a single random strain from each of the seven bacterial species in ZymoMC, not matching the particular strain targeted for depletion; (b) "One Genome with Zymo Mock Refs," the exact seven strains targeted for depletion; (c) "Pan-genome w/o Zymo Mock Refs," all RefSeq strains for each bacterial species in ZymoMC but excluding the depletion targets; and (d) "Pan-genome with Zymo Mock Refs," all RefSeq strains for each bacterial species in ZymoMC including the depletion targets.

Table 1 shows that using an index containing many strains but excluding the specific depletion target yields a similar F1-score (99.7% for SPUMONI and minimap2) compared with when we use an index consisting only of the depletion target (99.1% for minimap2, 99.8% for SPUMONI). The F1 score remained unchanged when the pan-genome index was used.

We conclude that a pan-genome index is a flexible tool for targeted sequencing, enabling targeting both at higher taxonomic levels and in situations where the particular target strain has not been assembled or is unknown. In subsequent experiments, we continued to assess both a single-strain index ("One Genome w/o Zymo Mock Refs") and a pan-genomic index ("Pan-Genome w/o Zymo Mock Refs"), focusing only on the indexes that exclude the target strain.

Simulated mock community: accuracy and efficiency

To assess these methods in the presence of sequencing error, we used PBSIM2 (Ono et al. 2020) to simulate Oxford-Nanopore-like reads (R9.4 chemistry) from ZymoMC references at varying levels of mean read accuracy (%): 85, 90, 95, and 98. We again supposed that our goal was to eject reads from the seven bacterial strains so as to obtain proportionally more reads from the yeast. The proportions of reads simulated from each genome were set to mimic those from the UNCALLED study (Kovaka et al., 2020) (Figure S1). Figure 1 shows that as the error rate decreases, the distribution of matching statistics from the positive index gains a heavier right tail; that is, the half-maximal exact matches become longer because they are interrupted less often by sequencing errors.

We next compared SPUMONI with a minimap2-based approach, using the reads' true simulated point of origin as the ground truth. As seen in Table 2, SPUMONI's F1 score – and several related measures – increase as read accuracy increases. For reads at 90% accuracy and greater, SPUMONI's pan-genome index achieved $\geq$ 99.7% F1, which was comparable with and sometimes greater than minimap2's pan-genome F1 scores. For both tools, the pan-genomic index substantially increased the F1 score, which is consistent with our results (Assessing genomic versus pan-genomic indexes).

Considering throughput as measured in base pairs processed per s (bp/sec), SPUMONI is on average about 19.3 times faster than minimap2 when using the pan-genomic index and about 1.8 times faster using the genomic index, and this is visualized in Figure S2. Furthermore, SPUMONI's pan-genomic index is about 16 times smaller than minimap2's, and SPUMONI's peak memory footprint is about 4 times lower.

Real mock community: accuracy and efficiency

Next, we applied our method to real nanopore reads from ZymoMC, obtained from SRA accession SRX7711546 (Kovaka et al., 2020). When we plotted the distribution of matching statistics obtained from reads from different species, we observed that the distributions were quite distinct for the bacterial reads, but overlapping for the yeast (Figure S3). This visualization shows how SPUMONI can distinguish between reads that it will try to eject and reads that it will let pass through the pore, and this difference can be statistically shown by differences in the KS-stat between the bacterial reads and the yeast reads (Figure S4).

We compared SPUMONI with minimap2, this time using a separately obtained minimap2 mapping as the gold standard. Specifically, we used minimap2 to map a suffix of the read, omitting the first 720 bases. To ensure the reads were long enough to enable an accurate mapping, we first filtered out reads that were shorter than 4,000 bp. We also trimmed the first 720 bases from each read before performing the gold-standard alignment because these bases are used for classification later. Gold-standard labels were given only to reads that minimap2 could uniquely map to a ZymoMC reference with an MAPQ of $\geq$ 30. For reads that had at least one secondary alignment, we required that the ratio of the secondary alignment's MAPQ to the primary alignment's MAPQ was $\leq$ 0.60.

Table 1. Assessing SPUMONI and minimap2 using both genomic and pan-genomic indexes

Accuracy on simulated mock community reads at 90% accuracy with indices of different size

Reference:	One genome w/o Zymo mock refs.			One genome with Zymo mocks refs.			Pan-genome w/o Zymo mock refs.			Pan-genome with Zymo mock refs.		
Reference size:	58 MB	58 MB	29 MB	56 MB	56 MB	28 MB	31 GB	31 GB	16 GB	31 GB	31 GB	16 GB
Approach:	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2
Accuracy	78.28	88.32	90.76	96.52	99.55	98.31	94.48	99.50	99.50	94.48	99.50	99.50
Precision	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Recall	87.15	87.97	90.48	96.42	99.54	98.26	94.32	99.49	99.49	94.32	99.49	99.49
Specificity	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
F1-score	93.13	93.13	95.00	98.18	99.77	99.12	97.08	99.74	99.74	97.08	99.74	99.74

"SPUMONI" refers to the mode that uses PMLs, while SPUMONI-ms refers to the mode that uses matching statistics instead.
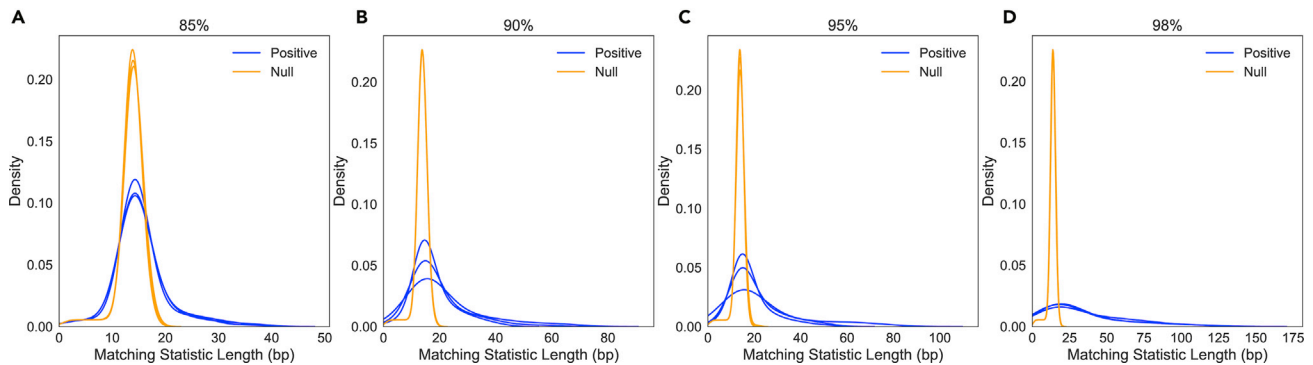
Figure 1. Distribution of matching statistics from positive and null indexes on simulated ZymoMC reads at accuracies of (A) 85%, (B) 90%, (C) 95%, and (D) 98%. Each plot contains the density curves for the first 720 bases (∼ 1.6 s) for three randomly chosen simulated _Escherichia coli_ reads.

Results in Table 3 show that SPUMONI achieved similar F1 score as minimap2. For the genomic ("One Genome") reference, SPUMONI achieved 92.79% F1 score, whereas minimap2 achieved 93.42% F1 score. Both tools achieved 100% precision and specificity in this case. For the pan-genomic ("Pan-genome") reference, SPUMONI achieved 97.94% F1 score, whereas minimap2 achieved 98.73%. In this case, SPUMONI achieved 100% precision and specificity, whereas minimap2 achieved 99.96% precision and 96.97% specificity.

When using the pan-genome reference, SPUMONI achieved a throughput about 11.9 times higher than that of minimap2. While when using the genomic index, SPUMONI achieved slightly higher throughput than minimap2 (902 kbp/s versus 852 kpb/s). When measuring peak RSS, we observed that SPUMONI's memory footprint was about one-fourth that of minimap2 and that its index was about 16 times smaller.

Human microbiome

Finally, we assessed our method on a human microbiome sequencing scenario with the goal of ejecting reads from the human host to enrich for any microbial species present. We constructed a data set consisting of a mixture of real reads from a recent human microbiome study that used Oxford Nanopore sequencing (Moss et al., 2020), as well as a set of simulated human nanopore-like reads with a mean read accuracy of 90%. Likely human reads were already filtered out of the former data set; therefore, we assumed that the only human reads in the final read set are the simulated ones. Because a human genome assembly is on the order of 3 billion nucleotides, an index containing one or more human assemblies presents a significantly larger but relevant challenge.

When we visualized the distribution of matching statistics for reads from different species (Figure 2), we saw the simulated human reads appeared to match the positive index (evidenced by the blue densities' thicker right tails), whereas reads from the microbiome study did not (indicated by the similarity of positive and null distributions).

We evaluated SPUMONI and minimap2 on this data set using two different indexes: (a) an index consisting only of the telomere-to-telomere consortium ("T2T") CHM13 (Miga et al., 2020) and (b) an index consisting of the T2T assembly together with the Ashkenazi (Zimin et al., 2020) and GRCh38 (Church et al., 2015) assemblies. Indexing multiple human genomes allows us to achieve similar benefits as we did for the mock-community pan-genomes, that is, coverage of a wider range of genetic variation, particularly structural variation. It also helps to reduce reference bias, which in our case would manifest as a tendency to find shorter matches in genomic regions with nonreference alleles.

When using the single-genome index, SPUMONI achieved somewhat higher F1 score (96.97%) than minimap2 (95.17%), and lower throughput (24.5 versus 35.7 kpb/s). When using the 3-genome index, minimap2 achieved higher F1 score (99.17%) than SPUMONI (97.08%), but SPUMONI had higher throughput (27.0 versus 13.5 kbp/src), which is shown in Table 4. As the reference became more repetitive – moving from one to 3 genomes – SPUMONI gained an index-size advantage, using 18 GB versus minimap2's 21 GB

Table 2. Comparing SPUMONI and minimap2 across various metrics on simulated ZymoMC reads of varying levels of accuracy

Accuracy, throughput, and index size on simulated mock community reads at various level of accuracy

Read accuracy (%):	85						90					
Reference:	One genome ref			Pan-genome ref			One genome ref			Pan-genome ref		
Reference size:	56 MB	56 MB	28 MB	31 GB	31 GB	16 GB	56 MB	56 MB	28 MB	31 GB	31 GB	16 GB
Approach:	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2
Accuracy	56.21	83.35	87.53	70.08	95.43	99.16	78.28	88.32	90.76	94.48	99.50	99.50
Precision	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
Recall	54.89	82.85	87.15	69.18	95.29	99.13	87.15	87.97	90.48	94.32	99.49	99.49
Specificity	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00	100.00
F1-score	70.88	90.62	93.13	81.78	97.59	99.56	93.13	93.13	95.00	97.08	99.74	99.74
Peak RSS (GB)	0.63	0.08	0.17	6.24	1.90	8.07	0.63	0.08	0.17	6.24	1.90	8.07
Index size (GB)[a]	0.68	0.09	0.10	6.20	1.90	31.00	0.68	0.09	0.10	6.20	1.90	31.00
Throughput (bp/s)	134,690	614,665	398,415	28,731	111,813	6,441	177,572	709,018	409,104	33,829	125,914	6,617
Read accuracy (%):	95						99					
Reference:	One genome ref			Pan-genome ref			One genome ref			Pan-genome ref		
Reference size:	56 MB	56 MB	28 MB	31 GB	31 GB	16 GB	56 MB	56 MB	28 MB	31 GB	31 GB	16 GB
Approach:	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2
Accuracy	86.48	91.00	90.86	99.60	99.65	99.40	89.76	92.20	91.30	99.65	99.60	99.50
Precision	100.00	99.94	100.00	100.00	99.95	100.00	100.00	100.00	100.00	100.00	99.95	100.00
Recall	86.07	90.78	90.58	99.59	99.69	99.39	89.45	91.96	91.04	99.64	99.64	99.49
Specificity	100.00	98.31	100.00	100.00	98.31	100.00	100.00	100.00	100.00	100.00	98.31	100.00
F1-score	92.52	95.14	95.06	99.80	99.82	99.69	94.43	95.81	95.31	99.82	99.80	99.74
Peak RSS (GB)	0.63	0.08	0.17	6.24	1.90	8.07	0.63	0.08	0.17	6.24	1.90	8.07
Index size (GB)[a]	0.68	0.09	0.10	6.20	1.90	31.00	0.68	0.09	0.10	6.20	1.90	31.00
Throughput (bp/s)	209,163	790,409	425,320	37,130	135,463	6,672	235,712	928,459	479,523	38,556	129,574	6,259

[a]The reported index size for SPUMONI-ms and SPUMONI includes only the positive index and not the null index because the null index can be used offline and deleted before the analysis.

Table 3. Comparing SPUMONI and minimap2 across various metrics on Real ZymoMC Reads

Accuracy, throughput and index size on real mock community reads

Reference:	One genome ref			Pan-genome ref		
Reference size:	56 MB	56 MB	28 MB	31 GB	31 GB	16 GB
Approach:	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2
Accuracy	81.64	86.72	87.82	94.62	96.02	97.52
Precision	100.00	100.00	100.00	100.00	100.00	99.96
Recall	81.39	86.54	87.66	94.55	95.97	97.53
Specificity	100.00	100.00	100.00	100.00	100.00	96.97
F1-score	89.74	92.79	93.42	97.20	97.94	98.73
Peak RSS (GB)	0.63	0.08	0.17	6.24	1.90	8.07
Index size (GB)[a]	0.68	0.09	0.10	6.20	1.90	31.00
Throughput (bp/s)	252,974	901,609	851,869	64,384	185,618	15,570

[a]The index sizes for SPUMONI-ms and SPUMONI are only for the positive index because the null index can be used offline and removed.

for the 3-genome index. While this comparison between SPUMONI and minimap2 is close, we expect that as we are able to index and align to more human references simultaneously – for example, as more assemblies from the Human Pangenome Reference Consortium (Human Pangenome Reference, 2021) and similar projects emerge — SPUMONI is well positioned for sublinear index growth and a greater throughput advantage. For instance, the *r*-index underlying SPUMONI was previously shown to be able to index up to 10 human genomes with sublinear growth in the index size (Mun et al., 2020).

DISCUSSION

SPUMONI is a streaming algorithm for targeted nanopore sequencing that uses matching statistics (and "PMLS") to classify reads in real time. SPUMONI's data structures – the *r*-index and MONI thresholds – allow it to handle repetitive pan-genome indexes more efficiently than competing approaches. SPUMONI's memory efficiency combines well with the flexibility afforded by nanopore sequencing, allowing SPUMONI to run on more portable hardware, like that associated with MinION and Flongle instruments. The ability to include a wide array of strains in a single index makes SPUMONI attractive for metagenomics applications where targets may not have already been cultured, assembled, and deposited in a resource like Refseq. As nanopore sequencing continues to improve, both base-calling accuracy and per-
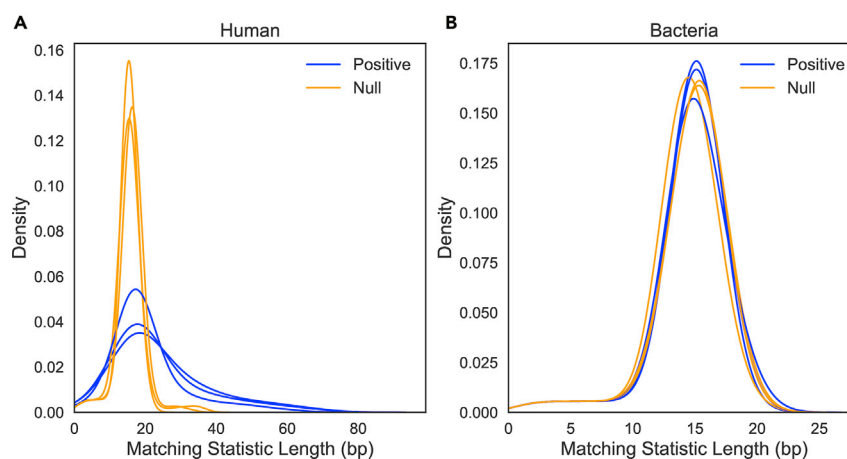
Figure 2. Distribution of matching statistics across three randomly chosen reads from (A) the human simulation and (B) the microbiome study (Moss et al. 2020). A single curve represents the first 720 bases (~ 1.6 Read Until seconds) of a read.

Table 4. Comparing SPUMONI and minimap2 on various metrics when processing the human microbiome reads

Accuracy, throughput and index size on human microbiome reads

Reference:	One human genome			Three human genomes		
Reference size:	5.8 GB	5.8 GB	2.9 GB	18.0 GB	18.0 GB	9.0 GB
Approach:	SPUMONI-ms	SPUMONI	minimap2	SPUMONI-ms	SPUMONI	minimap2
Accuracy	98.66	99.42	99.10	98.64	99.44	99.84
Precision	95.46	98.93	100.00	95.06	98.73	100.00
Recall	90.57	95.08	90.78	90.78	95.49	98.36
Specificity	99.54	99.89	100.00	99.49	99.87	100.00
F1-score	92.96	96.97	95.17	92.87	97.08	99.17
Peak RSS (GB)	57.29	15.23	7.85	62.80	18.06	9.70
Index size (GB)[a]	57.90	15.00	6.90	62.60	18.00	21.00
Throughput (bp/s)	7,518	24,476	35,742	6.860	27,024	13,549

[a]The index sizes for SPUMONI-ms and SPUMONI are only for the positive index because the null index can be used offline and removed.

instruction throughput will likely improve. SPUMONI is well positioned for these trends because it delivers its most advantageous combinations of speed and F1 score at higher base-calling accuracy.

SPUMONI operates on batches of already-called bases. In practice, the Read Until API delivers data in the form of raw current that must be base-called first. Because nanopore base callers have been steadily improving, it is possible that base calling will be integrated into onboard components of nanopore sequencers. Until then, users must run a separate base caller upstream of SPUMONI, as also required by Readfish (Payne et al., 2020). That said, the fact that SPUMONI's analysis is at the level of bases allows it to target other classification problems, such as metagenomics classification.

While SPUMONI's null index currently consists of the reverse of the sequences used in the positive index, this notion of "null" might be insufficient in some scenarios. For example, if there is substantial sequence similarity between depletion-target reads and reads that should not be targeted for depletion – for example, owing to conserved genes between species – the positive MSs within those sequences will be longer than what is expected by random chance for depletion-target reads. In these cases, we may need to augment the null model, perhaps by including the conserved sequences (not their reverses) in the null index.

Currently, we use the same KS-stat threshold for all the experiments which was optimized to perform well on real nanopore data sets. However, we expect that the optimal threshold will also be a function of the read accuracy and the reference used. In future work, we will investigate whether a simulation could be used to model the sequencing run to determine a threshold that is more tailored to a particular experiment.

Finally, we observed that SPUMONI can compress reads as it processes them: we can simply output each PML followed by the character in the read that did not match the corresponding character in the BWT; to decompress the read, we recover the characters that matched (and caused the PML to increment) using LF steps until we reach the mismatch character, at which point we jump to the previous or next occurrence of that character in the BWT, as we did while compressing the read. Because the compressing ratio of this scheme improves with larger PMLs, we may be able to use that compression ratio as an aggregate statistic when deciding whether to eject a read. Finally, we note that in some sense, this compression scheme works by predicting the characters in the reads and recording explicitly those characters it predicts incorrectly.

Limitations of study

One major limitation of the SPUMONI approach currently is the fact that it operates on sequences of bases opposed to the batches of signal that are delivered by the Read Until API. Future work on SPUMONI will be

aimed toward allowing it to accept electrical signal directly and interact with the Read Until API for it to be deployed as control software on a nanopore sequencer.

An additional limitation of our current approach is the limited performance improvement when SPUMONI focuses on human reads (Human microbiome) opposed to microbial reads (Evaluations with mock community). Our experiments show that SPUMONI's peak RSS for three human genomes is about 1.86X larger than the minimap2's peak RSS. SPUMONI's throughput when indexing those same three human genomes is only about 2X faster than minimap2's throughput. However, as the experiments in the Human microbiome section seem to indicate, we expect our throughput advantage to improve as the number of human genomes in the reference increases.

STAR★METHODS

Detailed methods are provided in the online version of this paper and include the following:

- KEY RESOURCES TABLE
- RESOURCE AVAILABILITY
  - Lead contact
  - Materials availability
  - Data and code availability
- METHOD DETAILS
  - Matching statistics with r-index
  - Pseudomatching lengths
  - Positive and null indexes
- QUANTIFICATION AND STATISTICAL ANALYSIS

SUPPLEMENTAL INFORMATION

Supplemental information can be found online at https://doi.org/10.1016/j.isci.2021.102696.

AUTHOR CONTRIBUTIONS

OA and BL conceived the SPUMONI method, MR and OA implemented the SPUMONI method, based on the MONI software. OA, MR, SK, TG, CB, and BL participated in the evaluation and refinement of the method. All authors contributed to the manuscript.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

Bannai, H., Gagie, T., and Tomohiro, I. (2020). Refining the *r*-index. Theor. Comput. Sci. *812*, 96–108. https://doi.org/10.1016/j.tcs.2019.08.005.

Burrows, M., and Wheeler, D. (1994). A block-sorting lossless data compression algorithm. Technical Report 124. *Digital SRC Research Report.*

Church, D.M., Schneider, V.A., Steinberg, K.M., Schatz, M.C., Quinlan, A.R., Chin, C.S., Kitts, P.A., Aken, B., Marth, G.T., Hoffman, M.M., et al. (2015). Extending reference assembly models.

Genome Biol. *16*, 13. https://doi.org/10.1186/s13059-015-0587-3.

Gagie, T., Tomohiro, I., Manzini, G., Navarro, G., Sakamoto, H., and Takabatake, Y. (2019). Rpair: Rescaling RePair with Rsync. Proc. SPIRE. https://doi.org/10.1007/978-3-030-32686-9_3.

Gagie, T., Navarro, G., and Prezza, N. (2020a). Fully functional suffix trees and optimal text searching in BWT-runs bounded space. J. ACM *67*, 2:1–2:54. https://doi.org/10.1145/3375890.

Gagie, T., Tomohiro, I., Manzini, G., Navarro, G., Sakamoto, H., Benkner, L.S., and Takabatake, Y. (2020b). Practical random access to SLP-compressed texts. Proc. SPIRE. https://doi.org/10.1007/978-3-030-59212-7_16.

Human Pangenome Reference (2021). Consortium. https://humanpangenome.org/.

Kovaka, S., Fan, Y., Ni, B., Timp, W., and Schatz, M.C. (2020). Targeted nanopore sequencing by realtime mapping of raw electrical signal with UNCALLED. Nat. Biotech. https://doi.org/10.1038/s41587-020-0731-9.

Li, H. (2018). Minimap2: pairwise alignment for nucleotide sequences. Bioinformatics *34*, 3094–3100. https://doi.org/10.1093/bioinformatics/bty191.

Miga, K.H., Koren, S., Rhie, A., Vollger, M.R., Gershman, A., Bzikadze, A., Brooks, S., Howe, E., Porubsky, D., Logsdon, G.A., et al. (2020). Telomere-to-telomere assembly of a complete human X chromosome. Nature *585*, 79–84. https://doi.org/10.1038/s41586-020-2547-7.

Moss, E.L., Maghini, D.G., and Bhatt, A.S. (2020). Complete, closed bacterial genomes from microbiomes using nanopore sequencing. Nat. Biotech. *38*, 701–707. https://doi.org/10.1038/s41587-020-0422-6.

Mun, T., Kuhnle, A., Boucher, C., Gagie, T., Langmead, B., and Manzini, G. (2020). Matching reads to many genomes with the r-index. J. Comput. Biol. *27*, 514–518. https://doi.org/10.1089/cmb.2019.0316.

Ono, Y., Asai, K., and Hamada, M. (2020). PBSIM2: a simulator for long read sequencers with a novel generative model of quality scores.

Bioinformatics. https://doi.org/10.1093/bioinformatics/btaa835.

Payne, A., Holmes, N., Clarke, T., Munro, R., Debebe, B.J., and Loose, M. (2020). Readfish enables targeted nanopore sequencing of gigabase-sized genomes. Nat. Biotech. https://doi.org/10.1038/s41587-020-00746-x.

Rossi, M., Oliva, M., Ben, L., Gagie, T., and Boucher, C. (2021). MONI: a pangenomics index for finding MEMs. *Proc. RECOMB.*

Wick, R.R., Judd, L.M., and Holt, K.E. (2019). Performance of neural network basecalling tools for Oxford Nanopore sequencing. Genome Biol. *20*, 129. https://doi.org/10.1186/s13059-019-1727-y.

Zimin, A.V., Shumate, A., Sherman, R.M., Puiu, D., Wagner, J.M., Olson, N.D., Pertea, M., Salit, M.L., Zook, J.M., and Salzberg, S.L. (2020). Assembly and annotation of Ashkenazi reference genome. BMC Bioinformatics *21*. https://doi.org/10.1186/s13059-020-02047-7.

STAR★METHODS

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Deposited data		
ZymoMC sequencing reads	Kovaka et al., 2020	https://www.ncbi.nlm.nih.gov/sra/SRX7711546[accn]
Human Bacterial Microbiome sequencing reads	Moss et al., 2020	https://www.ncbi.nlm.nih.gov/sra/SRX6602475[accn]
Telomere-to-Telemere Consortium CHM13 v1.0 assembly	Miga et al., 2020	https://github.com/marbl/CHM13
Ashkenazi assembly	Zimin et al., 2020	https://github.com/AshkenaziGenome/Assembly/
GRCh38 assembly	Church et al., 2015	https://www.ncbi.nlm.nih.gov/assembly/GCF_000001405.26/
Software and algorithms		
SPUMONI software	This paper	https://github.com/oma219/spumoni
PBSIM2	Ono et al., 2020	https://github.com/yukiteruono/pbsim2

RESOURCE AVAILABILITY

Lead contact
Requests for further information and resources should be directed to the Lead Contact, Ben Langmead (langmea@cs.jhu.edu).

Materials availability
This study did not generate any new materials.

Data and code availability
SPUMONI is an open-source software available at https://github.com/oma219/spumoni. The SPUMONI indexes used for each experiment can be obtained from: https://benlangmead.github.io/aws-indexes/spumoni.

For the "Pan-genome Reference" collection in the Assessing genomic versus pan-genomic indexes section, we used all available genomes for each bacterial species of the ZymoMC in the RefSeq Database. Accession numbers for the bacterial genomes can be downloaded at https://benlangmead.github.io/aws-indexes/spumoni. For the Real mock community: accuracy and efficiency section, we used the reads present in the SRA Project under Accession Number SRX7711546 (Kovaka et al., 2020).

For the human assemblies in the Human microbiome section, we used the telomere-to-telomere consortium CHM13 v1.0 assembly (Miga et al., 2020), the Ashkenazi assembly (Zimin et al., 2020), and GRCh38 (Church et al., 2015). For the read sets used in same section, the human reads were simulated from the telomere-to-telemere consortium CHM13 v1.0 assembly (Miga et al., 2020) at a mean read accuracy of 90% using PBSIM2 (Ono et al., 2020) and its model for the R9.4 chemistry. The bacterial microbiome reads were obtained SRA Accession SRX6602475 (Moss et al., 2020).

METHOD DETAILS

Matching statistics with r-index
Given a text $T[1..n]$ of length n, the Burrows-Wheeler transform (BWT) (Burrows and Wheeler 1994) is a reversible permutation of the T such that the character in position i is the character preceding the i-th lexicographic-sorted suffix of T. We use r to denote the number of maximal equal-letter runs of the BWT. The r-index Gagie et al. (2020a) is a self-index that stores a run-length encoded BWT, that is each run is encoded as a character together with the run length.

Given a text $T[1..n]$ of length n and a pattern $P[1..m]$ of length m, the *matching statistics* of P against T are defined as an array $MS[1..m]$ of length m, where each position $MS[i]$ stores the length of the longest prefix of

$P[i..m]$ that occurs in the study by *T.* Bannai et al. Bannai et al., 2020 introduced the *thresholds* which are $O(r)$ positions in the BWT marking a minimum of the longest common prefix array, between two equal-letter runs. They also proposed a two-pass algorithm to compute matching statistics using use these thresholds and the *r*-index. In the first pass, the algorithm steps backward along the pattern *P*. When it can, the algorithm uses the LF mapping to extend the match by one character. Where this is not possible, we "jump" either forward or back in the BWT to a position where the match can be extended. Whether we jump forward or back depends on which direction gives the longer common prefix with the match so far, which in turn is determined by the threshold's location. In the second pass, the algorithm uses a random-access data structure built over *T* to compute the lengths of the matching statistics.

Rossi et al. (Rossi et al., 2021) with MONI showed how to efficiently compute the thresholds for highly repetitive texts, and implemented the matching statistics algorithm. A MONI index consists of four main components, the run-length encoded BWT, suffix-array samples taken at run boundaries, the thresholds, and a grammar (Gagie et al. 2019, 2020b) that provides random access to *T*. These data structures allow computation of matching statistics in $O(m\log n)$ time and take $O(r + g)$ space where *g* is the size of a given straight-line program for *T*.

Pseudomatching lengths

SPUMONI modifies MONI by removing the second pass. As SPUMONI performs a backward LF-mapping search, it increments a length variable whenever the BWT character encountered matches the next character in *P*. If the character fails to match, the length variable is reset to 0, and we "jump" in the BWT as usual. The value of the length variable at each step gives the sequence of *pseudomatching lengths* (PMLs); these differ from matching statistics because we have ignored the possibility that a BWT jump can correspond to an extension of the current half-maximal match. PMLs will consistently be shorter than the true MSs. But long MSs – long enough to narrow the BWT range to the point where random matches are excluded – will generally yield long PMLs. Because the longest MSs are the ones with the most power to discriminate target from nontarget, we expect, and our results confirm, that PMLs are similarly useful for classification.

This simplification obviates the need to store either the SA samples or the random-access grammar for *T*; those were used only in MONI's second loop. Hence, a SPUMONI index consists only of the run-length encoded BWT and thresholds. This leads to improvements in the time and space complexity, where PMLs can be computed in $O(m\log\log n)$ time and take $O(r)$ space in worst case. Pseudocode highlighting differences between MONI and SPUMONI – and between MSs and PMLs – is given in Figure S5.

Positive and null indexes

In our approach, we generated matching statistics of the read with respect to both a positive and null index. The positive index consisted of both the forward and reverse complement of the sequence that we wanted to target, whether that be for depletion or enrichment. The null index simply consisted of the reverse of the positive index sequence, and the matching statistics generated with respect to the null index were meant to represent matching statistics you would get against random sequence. This would allow us to compare the distribution of matching statistic lengths with respect to the positive index to a baseline distribution, and if we see a clear difference, it is probably owing to the read matching significantly to sequence in the positive index.

QUANTIFICATION AND STATISTICAL ANALYSIS

To decide whether the positive and null distributions of matching statistics are different, we used the Kolmogorov-Smirnov test (KS-test), which compares the distributions' cumulative distribution functions (CDFs). We found that a Kolmogorov-Smirnov statistic (KS-stat) of 0.25 and 0.10 for matching statistics and PMLs, respectively, worked well across different nanopore data sets. We applied the KS-test to nonoverlapping regions of 90 bp which allows to us to compute the KS-stat as the Read Until API delivers new batches of data without having to revisit and use earlier batches of data in the computation.

In addition, before feeding in the matching statistics from the nonoverlapping regions into the KS-test, we applied a transformation function to the data. The function consisted of taking each matching statistic length and subtracting the mean of the null distribution, and replacing its value with 1 if it was less than 1. The intuition behind this function is that it compresses all of matching statistic lengths that are near-

random length into a matching statistic length of 1. This improves the accuracy using the KS-test because the KS-test is based on distances between CDFs so this transformation will tend to increase the KS-stat when the distributions are truly different.

Finally, to make a decision on the read level for whether the read should be classified as matching sequence in the positive index or not, we will perform the following. We gather all the KS-stats from the nonoverlapping regions and see if a simple majority of them are greater than the threshold.