MinION re-sequencing of *Giardia* genomes and *de novo* assembly of a new *Giardia* isolate

Stephen M. J. Pollo[1], Sarah J. Reiling[2], Janneke Wit[3], Matthew Workentine[1], Rebecca A. Guy[4], G. William Batoff[5], Janet Yee[5], Brent R. Dixon[2], and James D. Wasmuth[1*]

[1] Department of Ecosystem and Public Health, Faculty of Veterinary Medicine, University of Calgary, Calgary, Alberta Canada.

[2] Bureau of Microbial Hazards, Food Directorate, Health Canada, Ottawa, Ontario, Canada.

[3] Department of Comparative Biology and Experimental Medicine, Faculty of Veterinary Medicine, University of Calgary, Calgary, Alberta Canada.

[4] Division of Enteric Diseases, National Microbiology Laboratory, Public Health Agency of Canada, Guelph, Ontario, Canada.

[5] Department of Biology, Biochemistry and Molecular Biology Program, Trent University, Peterborough, Ontario, Canada.

* Corresponding author

Email: jwasmuth@ucalgary.ca

Running title: MinION assembly of *Giardia*.

Key words: long read sequencing, nanopore

Subject Category: Genomics

Abstract

The cost and portability of the Oxford Nanopore Technologies MinION make it a good candidate for detection of food and waterborne parasites. As a step toward developing the MinION as a tool for detection of food and waterborne parasites, we have evaluated the accuracy of genome assemblies produced from MinION sequencing data on a food- and waterborne parasite – *Giardia duodenalis*. Two strains of *G. duodenalis* that have reference genomes available in the literature (*G. duodenalis* Assemblage A isolate WB and *G. duodenalis* Assemblage B isolate GS) were re-sequenced on the MinION. *De novo* genome assemblies were performed using combinations of 1D or 1Dsq reads produced by sequencing, pooling data from sequencing runs for the same organism, and using different long read assemblers (Canu, Abruijn, or SMARTdenovo). The resulting assemblies then underwent up to eight rounds of genome polishing. The 207 draft assemblies were then compared against the reference genomes and evaluated on their average percent identity, proportion of mismatching bases, number of insertions and deletions per 1000 aligned bases, average size of insertions and deletions, and proportion of the reference genome that they covered between zero-and-four times. The assemblies were also evaluated on their overall size, number of contigs, and the number of known genes each was found to contain. The optimal assembly pipeline for *Giardia* sequences generated on the MinION was found to be 1D reads assembled with SMARTdenovo followed by four or five rounds of genome polishing with the program Nanopolish.

Introduction

Nucleic acid sequencing is a powerful tool used to study a variety of biological questions from mutations involved in leukemia (Minervini *et al.*, 2017), to population diversity among yeasts (Istace *et al.*, 2017), to the nature of Antarctic extremophiles (Johnson *et al.*, 2017). Currently, whole genomes are sequenced using second generation technologies, third generation technologies, or strategies involving combinations of technologies. Second generation sequencing of whole genomes involves fragmenting the genomic DNA, sequencing the short fragments (resulting in short sequencing "reads") and reassembling the sequences together computationally (Alekseyenko *et al.*, 2013). Second generation sequencing platforms produce high quality reads with low error rates (0.1% for Illumina HiSeq) but short lengths (mean length <250 bp for Illumina HiSeq), which pose challenges for assembly programs resulting in more fragmented assemblies (Rhoads and Au, 2015). In contrast, third generation sequencing platforms produce much longer reads (mean length <10 000 bp for PacBio and MinION) but have higher error rates (10-15% for PacBio and >10% for MinION depending on the chemistry) (Tyson *et al.*, 2017; Rhoads and Au, 2015; Lu *et al.*, 2016). These longer reads have the potential to resolve many genomic areas that are problematic for second generation data, such as repetitive and/or duplicated regions (Lu *et al.*, 2016). Importantly, eukaryotic genomes have many such repetitive and duplicated regions (as much as two thirds of the human genome may be repetitive elements (de Koning *et al.*, 2011)), making eukaryotic genomes especially good candidates for sequencing with third generation technologies.

The Oxford Nanopore Technologies (ONT) MinION is a recently released third generation sequencing platform based on nanopore technology (Lu *et al.*, 2016; Feng *et al.*, 2015). Briefly, the nucleic acids to be sequenced are driven through small pores in a membrane by an electrical current which causes fluctuations in the current in the pore (Lu *et al.*, 2016). Sensors measure these fluctuations, sending the data to a connected computer for processing and storage (Lu *et al.*, 2016). With maximum read lengths at least as long as 171 kbp (Johnson *et al.*, 2017), applications of MinION sequencing range from *de novo* sequencing of whole genomes to transcript isoform detection to metagenomics and amplicon sequencing. Moreover, the small size of the MinION makes it the most portable sequencing platform to date, which enables sequencing in remote areas, sequencing in the field, and real-time disease monitoring at the site of outbreak. To assemble genomes *de novo* from MinION data the following steps need to be performed: basecalling of the squiggle files produced by the MinION during sequencing, assembly of the long reads into draft genomes, and polishing of the assemblies.

The ONT website links to over 150 research articles that involve use of the MinION and a search of pubmed for articles that involve use of the MinION yields over 300 results. Of particular note, while only scratching the surface of this rapidly growing body of literature, are four studies. The first is a study that used the MinION as a diagnostic tool for characterizing *Mycobacterium tuberculosis* in patient samples (Votintseva *et al.*, 2017). By comparing to Illumina data the authors concluded that full strain identification and drug susceptibility could be discerned after as little as six hours of MinION sequencing. A second study used the MinION to detect arbovirus in RNA from a single mosquito (Batovska *et al.*, 2017). The authors found that the MinION sequencing data, of which only 0.28% mapped to the virus (229 reads = 10X coverage), was able to generate the viral genome with > 98% accuracy. Another study used the current changes generated by the MinION to map methylated bases in the DNA of *Escherichia coli* (Rand *et al.*, 2017). The authors were able to correctly map the methylation of 96% of cytosines in *E. coli* DNA and 86% of adenines in pUC19 plasmid DNA. Finally, a genomic survey of yeast isolates was conducted using the MinION (Istace *et al.*, 2017). The authors first re-sequenced the genome of the reference *Saccharomyces cerevisiae* strain S288C, then sequenced and assembled *de novo* genomes of 21 additional *S. cerevisiae* isolates to analyze genetic diversity in the species. The comparatively small genomes of these eukaryotes, which are contained within 17 chromosomes, were ultimately resolved into a few dozen contigs per genome (18 – 105 contigs depending on the isolate).

Giardia duodenalis (syn. *Giardia lamblia* or *Giardia intestinalis*) is a single-celled, eukaryotic, food and waterborne intestinal parasite that infects roughly 200 million people worldwide (Certad *et al.*, 2017). Infections can cause nausea, vomiting, diarrhea, and impaired growth and cognitive development (Certad *et al.*, 2017). Strains of *G. duodenalis* are categorized into eight subtypes, named Assemblages A through H, two of which are known to infect humans (A and B) (Certad *et al.*, 2017). The cells have two diploid nuclei each containing five chromosome pairs (Morrison *et al.*, 2007). The haploid genome size is ~12.8 MB (Aurrecoechea *et al.*, 2009). Genome comparisons between assemblages of *G. duodenalis* found only 77% nucleotide and 78% amino acid identity in coding regions, probably reflecting the unclear taxonomy of this group (Franzén *et al.*, 2009). Currently six strains of *G. duodenalis* have reference genomes available (Aurrecoechea *et al.*, 2009). Here we have generated MinION sequence data for *G. duodenalis* Assemblage A isolate WB (hereafter referred to as *Giardia* AWB), *G. duodenalis* Assemblage B isolate GS (hereafter referred to as *Giardia* BGS), and *G. duodenalis* isolated from a beaver (hereafter referred to as *Giardia* beaver).

Methods

Giardia strains

G. intestinalis assemblage A strain WB (ATCC 50803), and *G. intestinalis* assemblage B strain GS (ATCC 50580) were obtained from the American Tissue Culture Collection, *G. intestinalis* Beaver isolate was a gift from Dr. Gaetan Faubert from McGill University. *Giardia* trophozoites were growth in TYI-S-33 medium within 16-mL screw capped glass tubes incubated at 37°C

DNA extraction

Ten 16-mL culture tubes of each *Giardia* isolate (WB, GS and Beaver) grown to late logarithm stage (~5 - 8 x 10^5 cells/mL) were used for genomic DNA isolation. The culture tubes were chilled on ice for 5 min and the cells were collected by centrifugation at 1,100 x *g* for 15 minutes at 4°C. Genomic DNA was extracted with DNAzol Reagent (ThermoFisher Scientific) by following the manufacturer's instructions. Briefly, each cell pellet was resuspended and lyzed in DNAzol Reagent by gentle pipetting followed by a freeze (30 min at 80°C) and thaw (10 min at room temperature) step. The lysate was then centrifuged at 10,000 x g for 10 min at 4°C to remove insoluble cell debris. The supernatant was transferred to a new tube and the DNA was recovered by centrifugation of the supernatant at 4,000 x g for 5 min at 4°C. The DNA pellet was washed twice with 75% ethanol and then air-dried. The DNA was resuspended initially in 8 mM NaOH and then neutralized by additional of HEPES to a final concentration of 9 mM.

RNA was removed from the DNA sample by the addition of 1 - 2 μL of 20 ug/uL RNaseA (BioShop) followed by incubation at 65°C for 10 min. The degraded RNA was precipitated by the addition of ammonium acetate, incubation at 4°C for 20 min, and centrifugation at 12,000 xg for 30 min. at 4°C. The supernatant was transferred to a new tube and the DNA was precipitated by the addition of 95% ethanol, incubation at room temperature for 5 min, and centrifugation at 12,000 xg for 20 min. at 4° C. The DNA pellet was washed once with 0.01M ammonium acetate in 75% ethanol and once with 75% ethanol alone. The DNA pellet was air-dried before resuspension in TE buffer (10 mM Tris-HCl pH 8.0, 1 mM EDTA)

MinION sequencing

The 1Dsq library preparation kit SQK-LSK308 was used as recommended by the manufacturer (Oxford Nanopore Technologies, Oxford, United Kingdom). Approximately 200 ng of prepared library was loaded onto a FLO-MIN107 (R9.5) flow cell. Data collection was carried out with live basecalling for 48 h, or until no more strands were being sequenced.

Basecalling, de novo *assembly, and genome polishing*

Basecalling of all MinION output files was performed with the program Albacore (version 2.0.2) (Vera, 2017) using the full_1dsq_basecaller.py method to basecall both 1D and 1Dsq reads. The flowcell and kit parameters were FLO-MIN107 and SQK-LSK308 respectively. The general command used to run Albacore was: `full_1dsq_basecaller.py --flowcell FLO-MIN107 --kit SQK-LSK308 --input PATH/TO/FAST5/FILES --save_path ./ --worker_threads 38`

De novo assemblies were performed using the programs Abruijn (version 2.1b) (Lin *et al.*, 2016), Canu (version 1.6) (Koren *et al.*, 2017), and SMARTdenovo (version 1.11 running under Perl version 5.22.0) (Ruan, 2017). Abruijn assemblies were conducted using the nanopore platform setting, coverage estimates calculated as the number of bases in the input reads divided by the reference genome size (Table 1) all rounded to the nearest integer, and all other default settings (one polishing iteration, automatic detection of kmer size, minimum required overlap between reads of 5000 bp, automatic detection of minimum required kmer coverage, automatic detection of maximum allowed kmer coverage). Canu assemblies were performed using Canu's settings for uncorrected nanopore reads (-nanopore-raw), genome sizes estimated from the reference genome sizes (Table 1), and setting gnuplotTested=true to bypass html output report construction. SMARTdenovo assemblies were conducted using default settings (kmer length for overlapping of 16 and minimum required read length of 5000 bases). The general commands used to run each of the assemblers, with variable parameters written in upper case, were:

Abruijn: `abruijn PATH/TO/READS out_nano COVERAGE_ESTIMATE --platform nano --threads 56`

Canu: `canu -p UNIQUE_NAME genomeSize=12.8m -nanopore-raw PATH/TO/READS gnuplotTested=true`

SMARTdenovo: `smartdenovo.pl -p UNIQUE_NAME PATH/TO/READS > UNIQUE_NAME.mak` , followed by the command: `make -f UNIQUE_NAME.mak`

Genome polishing is an error correction step performed on assemblies generated from third-generation data to compensate for the high error rate of the reads (Lu *et al.*, 2016). It involves re-evaluating the base calls from the MinION squiggle files together with the read overlap information from the assembly to improve base accuracy and correct small insertions and deletions (Loman *et al.*, 2015). Here polishing was performed with the program Nanopolish (version 0.8.5) following the directions for "computing a new consensus sequence for a draft assembly" (Simpson, 2017). Briefly, the draft genome was first indexed using BWA (version 0.7.15-r1140) (Li and Durbin, 2010) and the basecalled reads were aligned to the draft genome using BWA. SAMtools (version 1.6 using htslib 1.6) (Cock *et al.*, 2015) was then used to sort and index the alignment. Nanopolish then computed the new consensus sequence in 50kb blocks in parallel, which were then merged into the polished assembly. The general commands used to run Nanopolish were:

```
nanopolish index -d PATH/TO/FAST5/FILES PATH/TO/READS

bwa index PATH/TO/ASSEMBLY/TO/POLISH

bwa mem -x ont2d -t 8 PATH/TO/ASSEMBLY/TO/POLISH PATH/TO/READS |
samtools sort -o reads.sorted.bam -T reads.tmp

samtools index reads.sorted.bam

python ~/nanopolish/scripts/nanopolish_makerange.py
PATH/TO/ASSEMBLY/TO/POLISH | parallel --results
nanopolish.results -P 14 nanopolish variants --consensus
UNIQUE_NAME_polished_x${POLISHING_ITERATION}.{1}.fa -w {1} -r
PATH/TO/READS -b reads.sorted.bam -g PATH/TO/ASSEMBLY/TO/POLISH
-t 4 --min-candidate-frequency 0.1
```

```
python ~/nanopolish/scripts/nanopolish_merge.py
UNIQUE_NAME_polished_x${POLISHING_ITERATION}.*.fa >
UNIQUE_NAME_polished_x${POLISHING_ITERATION}_genome.fa
```

Read Error Profile Analysis

Read error profiles were examined for the six *Giardia* AWB and *Giardia* BGS runs using the program NanoOK (version v1.31) (Leggett *et al.*, 2016). NanoOK extracts fasta sequences from the fast5 files produced by the MinION and aligns them to the reference genome using the LAST aligner (version 876) (Kielbasa *et al.*, 2011). It then calculates error profiles for each set of reads that aligned to each contig in the reference. To obtain overall values for all reads in the sequencing run, for each error metric the value for each contig was extracted from the .tex file produced by NanoOK and multiplied by the proportion of the total reads mapping to that contig. These values were then summed to yield the metric value with respect to all reads in the sequencing run. The sums were also scaled according to the proportion of the total reads that were included in the metric calculation - those that were mapped to the contigs - to yield the metric value for all reads used in the analysis.

Optimal Assembly Pipeline Determination

The effects on final assembly quality were evaluated for the following parameters: 1D vs 1Dsq input reads, pooling reads for the same organism from multiple runs, assembly program, and number of genome polishing iterations. Firstly, 13 distinct input combinations, that represent all permutations of pooling runs for the same organism for both 1D and 1Dsq reads, were used for *de novo* assemblies: AWB_0157 1D reads, AWB_0157 1Dsq reads, AWB_0150_0157 1D reads, AWB_0150_0157 1Dsq reads, AWB_2338 1D reads, AWB_2338 1Dsq reads, AWB_2331_2338 1D reads, AWB_0150_0157_2331_2338 1D reads, AWB_0150_0157_2338 1Dsq reads, BGS_2244 1D reads, BGS_2244 1Dsq reads, BGS_2237_2244 1D reads, and BGS_2237_2244 1Dsq reads (Table 1). Each of these input combinations was used to perform a *de novo* assembly with each of the three assemblers used: Abruijn, Canu, and SMARTdenovo. All of the resulting assemblies that produced contiguous sequences were polished with Nanopolish. Eight rounds of Nanopolish polishing were performed on the Canu and SMARTdenovo assemblies and seven rounds were performed on the Abruijn assemblies (which get polished once by Abruijn).

All assemblies and polished versions of the assemblies were aligned to the corresponding reference genome using the LAST aligner (version 876) (Kielbasa *et al.*, 2011) following the example for human-ape alignments (Mcfrith, 2017). Briefly, the reference genome was indexed using LAST, then substitution and gap frequencies were determined using the last-train method (Hamada *et al.*, 2017). Finally, alignments were performed using the lastal method and the determined substitution and gap frequencies. The resulting alignments were then filtered to retain only those alignments with an error probability < $1e^{-5}$. Preliminary inspection of the *Giardia* BGS assemblies suggested the assemblies could be an improvement to the reference sequence, so they were excluded from further analysis. *Giardia* AWB assemblies were aligned to only the contigs from the reference genome labelled GLCHR01, GLCHR02, GLCHR03, GLCHR04, and GLCHR05 (representing the five chromosomes of *Giardia duodenalis*). Filtered alignments were converted to other file formats (for metric calculation) using the maf-convert method in the LAST aligner.

Average percent identity was calculated from alignments in blasttab format by taking the sum of the percent identity multiplied by the alignment length for each aligned portion and dividing that sum by the total alignment length. Proportion of mismatching bases was calculated from alignments in psl format by taking the sum of mismatching bases for all aligned portions divided by the total alignment length. Total number of indels per 1000 aligned bases was calculated from alignments in psl format by taking the sum of the number of insertions in the query and the number of insertions in the target for all aligned portions, dividing that sum by the total alignment length and multiplying by 1000. Average size of indels was calculated from alignments in psl format by taking the sum of the number of bases inserted in the query and the number of bases inserted in the target for all aligned portions and dividing that sum by the total number of indels. The proportions of the reference covered 0, 1, 2, 3, or 4 times were calculated using BEDtools (version v2.27.1) (Quinlan and Hall, 2010). Alignments were first converted to SAM format and SAMtools was used to sort the alignment and convert it to a bam file. The genomecov function of BEDtools was then used to analyze the coverage of every base in the reference genome in the alignment. The proportion of bases in the reference genome with 0, 1, 2, 3, and 4 fold coverage in the assembly were retrieved.

The assembly evaluation metrics Number of Contigs and Genome Size were calculated for each assembly from the assembly fasta file. Finally the metric Spaln Value was calculated using the program spaln (Iwata and Gotoh, 2012), which aligns *Giardia* AWB proteins against the assembly to determine how many genes (out of 8157) are found in the assembly. Default parameters were used in the search. The values are reported as the proportion of the 8157 genes that were found in each assembly.

Average and standard deviation values for the groupings presented in the tables and figures for each metric were calculated in R (R, 2013). R was also used to construct the plots for the figures.

Results

Read basecalling and error analysis

The MinION sequencing runs used here produced several hundred thousand reads each with the exception of Run2, which was a second run conducted on a previously used flow cell (Table 2). In addition to producing fewer reads, re-using the flow cell also resulted in lower proportions of reads passing the quality threshold during basecalling with 64% and 81% of 1D reads passing in Run2 compared to 90 – 98% of 1D reads passing in Runs 1, 3, and 4 and 0% and 10% of 1Dsq reads passing in Run2 compared to 39 – 58% of 1Dsq reads passing in Runs 1, 3, and 4 (Table 2). NanoOK (Leggett *et al.*, 2016) analysis of read error profiles showed that reads from Run2 have lower aligned base identity and higher substitutions per 100 bases compared to the other runs (Table 3, scaled values).

NanoOK analysis of 1D read error profiles for all runs indicated a 9 – 17% error rate in read regions that aligned to the reference genome (Table 3, aligned base identity – scaled) and a 24 – 46% error rate across the entirety of reads that aligned to the reference genome (Table 3, overall base identity – scaled). The analysis also showed deletions are slightly more likely than insertions in the reads, though both are 1 to 2 bases in length on average (Table 3). Average and maximum read lengths for all runs are presented in Table 4. Notably, the maximum 1D read length generated in the sequencing runs analyzed here was 1,132,445 bases, though this read did not align to any *Giardia* reference genome nor did it have significant BLAST hits longer than ~45 bp in the nr database (data not shown).

De novo *assembly and genome polishing*

Of the 39 *de novo* assemblies performed, five did not have sufficient numbers of reads to generate any contigs (AWB_2338_1D_smartdenovo, AWB_2338_1Dsq for all three assemblers, and AWB_2331_2338_1D_smartdenovo). Additionally, three of the assemblies could not be polished the full eight times (BGS_2244_1D_smartdenovo, BGS_2237_2244_1D_canu, and BGS_2237_2244_1D_smartdenovo). The remaining assemblies were all polished eight times and the evaluation metrics were calculated for the nine resulting draft assemblies from each *Giardia* AWB input/assembler combination for a total of 207 assemblies (Supplementary Table 1).

Evaluation of assemblies and optimal pipeline analysis

The *Giardia* AWB assemblies that performed the best in each evaluation metric are listed in Table 5. No assembly ranked first in more than two of the metrics. To further examine the effects of 1D vs 1Dsq input reads, pooling reads for the same organism from multiple runs, assembly program, and number of genome polishing iterations, for each metric the values for all the assemblies were plotted (Figs 1 – 5 and S1 – S5 in supplementary material). The average value and standard deviation for each group were also calculated (Tables 6 – 9).

Discussion

Upon initial inspection, the averages and standard deviations of the evaluation metrics for the assemblies generated from 1D vs 1Dsq input reads would suggest no difference between the two (Table 6). The standard deviations also indicate performances of the 1D assemblies are more variable (Table 6). However, plotting the values (Fig. 1 and S1), suggests that when the 1D assemblies perform well they often out-perform the 1Dsq assemblies, but they are also much more variable. It is also worth noting that most of the 1D assemblies with poor performance in any of the evaluation metrics come from using only Run2 data (AWB_2331 and/or AWB_2338) and may be performing poorly due to insufficient sequencing depth (Supplementary Table 1). Since every assembly constructed from 1Dsq input reads has a corresponding assembly constructed from 1D input reads (Supplementary Table 2), to further examine the relationship between using 1D vs 1Dsq input reads, the 1D vs 1Dsq input pairs were plotted together (Fig. 2 and S2) and the average and standard deviation for every metric was calculated from only these assemblies (and not the additional 1D assemblies with no corresponding 1Dsq assembly) (Table 7). The new values and plots show that while the 1D assemblies are often more variable than the 1Dsq assemblies, the 1D assemblies generally out-perform the 1Dsq assemblies, especially in the average percent identity, number of indels per 1000 aligned bases, spaln value, number of contigs, and genome size metrics (Table 7, Fig. 2 and S2).

When examining the effects of pooling or not pooling runs for the same organism, the most obvious difference is between assemblies generated from solely Run2 data (AWB_2331 and AWB_2338) and assemblies that include Run1 data (AWB_0150 and AWB_0157) (Table 8, Fig. 3 and S3). Since this difference may be caused by the much smaller number of reads in Run2, informative comparisons of the effects of pooling or not pooling runs are AWB_0157 assemblies compared to AWB_0150_0157 and AWB_0150_0157_2331_2338 assemblies or AWB_2338 assemblies compared to AWB_2331_2338 assemblies. Among the assemblies that used Run1 data, no input combination produced values significantly different from the others for any metric examined (Table 8) and examination of the plotted values shows no clear patterns for any metric (Fig. 3 and S3), suggesting pooling or not pooling the input data had no effect. Similarly, the pooled and non-pooled Run2 assemblies did not have significantly different values for any metric (Table 8), nor did any clear pattern emerge when plotting the values (Fig. 3 and S3). Taken together these results suggest pooling or not pooling input data for the same organism has no significant effect on the final assembly once adequate genome coverage is achieved (though the exact cut-off for “adequate coverage” was not determined here). For runs with low read counts however, pooling runs can improve the final assembly, as was the case here for AWB_0150_0157_2331_2338 assemblies compared to AWB_2331_2338 assemblies.

Among the three assemblers tested, the SMARTdenovo assemblies showed the lowest variability in all metrics except average indel size (Table 9). Moreover, the SMARTdenovo assemblies had the highest average values for average percent identity, spaln value, and proportion of reference covered 1X (where higher values indicate better performance) (Table 9). They also had the lowest average value for proportion of reference not covered and an average genome size value that is closest to the size of the reference genome (Table 9). Additionally, four of the top performing assemblies in Table 5 are SMARTdenovo assemblies (for spaln value, number of contigs, proportion of reference not covered, and proportion of reference covered 1X). Plotting the SMARTdenovo assembly values for each metric also showed consistently strong performance in all metrics except average indel size (Fig. 4 and S4). The Abruijn assemblies show the greatest variability in all metrics except average indel size, number of contigs, and genome size (Table 9). They had the lowest average indel size and the lowest variability in average indel size (Table 9). Despite eight of the top performing assemblies in Table 5 being Abruijn assemblies, plotting the Abruijn assembly values for each metric showed highly variable performance consistent with the averages and standard deviations in Table 9 (Fig. 4 and S4). Finally, the Canu assemblies generally performed somewhere between the SMARTdenovo and Abruijn assemblies, except in the number of indels per 1000 aligned bases where Canu had the strongest performance of the assemblers (Table 9). Notably, all of the assemblies with poor performance in the number of contigs metric were Canu assemblies (Fig. 4 and S4) and these were all generated from Run2 data only (Supplementary Table 1), suggesting that Canu is particularly sensitive to low coverage compared to the other assemblers.

The effects of genome polishing on each of the assembly evaluation metrics are shown in Fig. 5 and S5. For all metrics the biggest changes occur after the first round of polishing. The values for most of the metrics remain relatively consistent after the first round of polishing with the exception of average percent identity, proportion of mismatching bases, average indel size, and spaln value. For average percent identity, unpolished genomes that perform well show no significant change with polishing. For the unpolished genomes that do not perform well, polishing improves average percent identity up to about four or five rounds of polishing, after which average percent identity levels off (Fig. 5 and S5). The same trend can be seen for the proportion of mismatching bases metric where genomes that show an improvement with more than one round of polishing level off after around four or five rounds of polishing. Interestingly, the genomes that showed the biggest changes in the average indel size metric showed a decrease in performance with increasing polishing, though this decrease levelled off after around two or three rounds of polishing (Fig. 5 and S5). Finally, most improvements to the spaln value metric were seen after two rounds of polishing, after which values levelled off or occasionally decreased as polishing rounds reached six or higher (Fig. 5 and S5). Overall these results suggest four to five rounds of polishing will increase or not affect the performance of assemblies in all the metrics used here except average indel size.

Analysis of the 207 *Giardia* AWB assemblies produced through different combinations of 1D vs 1Dsq input reads, pooling different sequencing runs for the same organism, assembly program, and number of rounds of genome polishing indicates that the optimal assembly pipeline for MinION sequenced *Giardia* sp. is a SMARTdenovo assembly from 1D reads (either pooled or non-pooled input to reach sufficient genome coverage) followed by four or five rounds of polishing with Nanopolish.

An optimal assembly pipeline for MinION data can change with each release of new programs specializing in handling long error prone reads. Since all of the programs used here are under active development, all future comparisons and evaluations of an assembly pipeline for MinION sequenced *Giardia* sp. should include assemblies made from updated versions of these programs. Already having the scripts to calculate the evaluation metrics used here would make such re-evaluations easier to perform and enable an evaluation of assembler performance that is current with each new program or version release. The typical publication process, from numerous drafts of a manuscript and peer-review, can be time-consuming and not conducive to keeping such an analysis current. Therefore, we recommend that future work be presented in more of a blog or community forum similar to an analysis on github of MinION basecalling programs (Wick, 2017). These media may also make it easier to discuss issues surrounding installation of these programs and running them in various computing environments. For example, some of the programs used here took up to a month to get installed and running properly. Having a current analysis of available long read assemblers would therefore also allow researchers to determine which programs are worth the time to get working and when it may be a better use of time to go with programs that need less configuration (like Canu which worked immediately) but will still perform adequately for the intended purpose.

The data generated here can also be used to investigate the level of heterozygosity in the tetraploid *G. duodenalis*. While the amount of the reference genome covered by a particular assembly zero times or one time can be used as a metric to evaluate the completeness of the assembly, the amounts of two, three, and four times coverage may be more indicative of the level of variation between the chromosomes (since the reference sequence contains a single sequence for each of the four copies of each chromosome). A more comprehensive analysis of the levels of heterozygosity in *Giardia* sp., both within isolates and between isolates, perhaps by including more accurate second-generation sequencing reads, may provide insight into the biology and pathogenicity of these organisms.

Acknowledgements

This work was supported the Ontario Ministry of Agriculture, Food and Rural Affairs (OMAFRA).

Table 1. MinION sequencing run metadata. Sequencing run names, isolate name, and reference genome size for all sequencing runs are presented.

Run Name	**Run ID**	**Isolate**	**Reference Genome Size (bp)**	**Name Used in this Document**
SRRun1	20170720_0150_GiardiaWB_20170719	Giardia AWB	12,827,416	AWB_0150
SRRun1	20170720_0157_GiardiaWB_20170719	Giardia AWB	12,827,416	AWB_0157
SRRun2	20170721_2331_GiardiaWB_20170721	Giardia AWB	12,827,416	AWB_2331
SRRun2	20170721_2338_GiardiaWB_20170721	Giardia AWB	12,827,416	AWB_2338
SRRun3	20170726_2302_GiardiaBeaver_20170726	Giardia Beaver	N/A	Beaver_2302
SRRun3	20170726_2309_GiardiaBeaver_20170726	Giardia Beaver	N/A	Beaver_2309
SRRun4	20170731_2237_GiardiaGS_20170731	Giardia BGS	11,001,532	BGS_2237
SRRun4	20170731_2244_GiardiaGS_20170731	Giardia BGS	11,001,532	BGS_2244

Table 2. Albacore (Vera, 2017) basecalling results for all MinION sequencing runs. Both 1D and 1Dsq basecalling were performed. “Pass” and “Fail” refer to reads that met or did not meet the quality threshold respectively.

Name	Total Number of 1D Reads	Number of 1D Reads Pass	Number of 1D Reads Fail	Total Number of IDsq Reads	Number of 1Dsq Reads Pass	Number of 1Dsq Reads Fail
AWB_0150	1,225	1,207	18	172	68	104
AWB_0157	329,039	304,219	24,820	60,156	25,755	34,401
AWB_2331	237	152	85	16	0	16
AWB_2338	19,531	15,842	3,689	1,904	192	1,712
Beaver_2302	1,668	1,603	65	146	69	77
Beaver_2309	382,740	354,581	28,159	53,553	29,349	24,204
BGS_2237	1,508	1,449	59	212	124	88
BGS_2244	885,046	804,942	80,104	143,371	62,452	80,919

Table 3. Read error profiles for *Giardia* AWB and *Giardia* BGS MinION sequencing runs. Using NanoOK (Leggett *et al.*, 2016), reads were aligned to the corresponding reference genome and the error profiles of aligned reads were evaluated. NanoOK outputs read error profiles for each reference contig. To get overall error profiles for all reads, the values for each contig were multiplied by the proportion of total reads that aligned to that contig. The sum of these values for each error metric are denoted as (WS) for "weighted sum". The proportion of total sequencing reads that were used for NanoOK's analysis is presented and the weighted sum values scaled according to this proportion are also presented as (S) for "scaled".

Error Type	AWB_01 50 Reads	AWB_01 57 Reads	AWB_23 31 Reads	AWB_23 38 Reads	BGS_22 37 Reads	BGS_22 44 Reads
Overall Base Identity (%) (WS)	67.33243	62.31616	15.22368	34.67214	7.351816	43.87562
Aligned Base Identity (%) (WS)	79.25533	74.66262	23.29439	44.14773	11.53835	69.68747
Identical Bases per 100 (WS)	70.41662	65.45936	19.9151	37.667	10.20389	61.07305
Inserted Bases per 100 (WS)	4.632477	3.242652	2.190203	2.676066	0.438305	3.469315
Deleted Bases per 100 (WS)	5.130817	7.061156	1.894861	5.046213	1.02281	6.109137
Substitutions per 100 (WS)	7.367688	7.799397	4.039511	7.220712	0.955251	6.818345
Mean Insertion (WS)	1.434478	1.221896	0.491964	0.778653	0.187017	1.185356
Mean Deletion (WS)	1.419001	1.493229	0.446071	0.940473	0.233182	1.470522
Proportion of Reads Counted (%)	87.55	83.56	28.04	52.61	12.62	77.47
Overall Base Identity (%) (S)	76.9074	74.57654	54.29272	65.90409	58.25528	56.63563
Aligned Base Identity (%) (S)	90.52579	89.3521	83.07557	83.91508	91.4291	89.95414
Identical Bases per 100 (S)	80.43017	78.33816	71.02389	71.59665	80.85492	78.83445
Inserted Bases per 100 (S)	5.291236	3.880627	7.810995	5.086611	3.473098	4.478269
Deleted Bases per 100 (S)	5.860442	8.450402	6.757707	9.591737	8.104675	7.88581
Substitutions per 100 (S)	8.415406	9.333888	14.40624	13.72498	7.569342	8.801271
Mean Insertion (S)	1.638467	1.462298	1.754508	1.480048	1.48191	1.530084
Mean Deletion (S)	1.620789	1.787014	1.590838	1.787632	1.847718	1.898183

Table 4. Read statistics for all MinION sequencing runs for both 1D and 1Dsq reads.

Name	Average Length of 1D Reads	Longest 1D Read	Average Length of 1Dsq Reads	Longest 1Dsq Read
AWB_0150	5066.15	42781	5335.22	18489
AWB_0157	7195.29	470735	7685.61	43102
AWB_2331	3450.08	32138	2853.62	6523
AWB_2338	6484	330795	7344.74	32705
Beaver_2302	5113	37229	5273.86	22740
Beaver_2309	8270.88	1132445	8472.84	59564
BGS_2237	6534.03	56642	5529.57	25876
BGS_2244	9417.6	485807	9829.82	66185

Table 5. Top performing *Giardia* AWB assemblies for each metric. See Supplementary Table 1 for the full dataset. The cells highlighted in yellow represent the best values for the metric among the genome assemblies that were large enough to contain the entire *Giardia douodenalis* genome sequence.

Name	**0150_0157_2331_2338 abruijn_1d polished_x8**	**0150_0157 abruijn_1d polished_x7**	**0157 abruijn_1dsq polished_x2**	**0157 abruijn_1d unpolished**	**0157 smartdenovo_1dsq polished_x1**	**0157 smartdenovo_1d polished_x6**	**0150_0157 abruijn_1dsq unpolished**	**0157 smartdenovo_1d polished_x8**
Average Percent Identity	99.8828	99.8785	99.6546	90.8326	99.4366	99.4821	97.0083	99.5268
Proportion Mismatching Bases	0.0004	0.0004	0.0002	0.0131	0.0006	0.0007	0.0021	0.0008
Indels per 1000 Aligned Bases	0.7368	0.7120	2.5067	48.3308	4.0095	1.2200	17.6432	1.1358
Average size of indels	1.0937	1.1982	1.2853	1.6252	1.2566	3.6727	1.5793	3.5063
genome_0	0.1568	0.1539	0.1700	0.9644	0.1438	0.1429	0.1581	0.1431
genome_1	0.8106	0.8196	0.8149	0.0347	0.8273	0.8114	0.7387	0.8111
genome_2	0.0244	0.0186	0.0137	0.0001	0.0221	0.0362	0.0747	0.0364
genome_3	0.0071	0.0067	0.0009	0.0002	0.0035	0.0049	0.0133	0.0049
genome_4	0.0005	0.0006	0.0002	0.0004	0.0012	0.0018	0.0068	0.0018
Spaln Value	0.9326	0.9378	0.8157	0.0636	0.7770	0.9496	0.3347	0.9497
Number of Contigs	57	51	72	97	29	38	95	38
Genome Size	11105834	11064011	10587062	14043058	11266530	11732229	12853839	11731454

Table 6. Summary statistics for all assemblies generated from 1D and 1Dsq reads for all metrics examined. The average and standard deviation for each group is presented. All values are plotted in Fig. 1 and S1.

Metric	Average 1D	Stdev 1D	Average 1Dsq	Stdev 1Dsq
Average Percent Identity	95.3332	10.2696	99.3950	0.4842
Proportion Mismatching Bases	0.0093	0.0146	0.0005	0.0003
Indels per 1000 Aligned Bases	16.6999	23.8932	4.2978	2.8039
Average size of indels	2.0664	0.8524	1.2542	0.0798
Spaln Value	0.5662	0.4407	0.7558	0.0891
Number of Contigs	38.7540	21.1942	66.9259	28.3376
Genome Size	7661519	5346589	11175628	469973
Proportion of Reference Not Covered	0.4475	0.3945	0.1563	0.0093
Proportion of Reference Covered 1X	0.5193	0.3774	0.8105	0.0192
Proportion of Reference Covered 2X	0.0188	0.0152	0.0248	0.0126
Proportion of Reference Covered 3X	0.0035	0.0027	0.0051	0.0038
Proportion of Reference Covered 4X	0.0015	0.0013	0.0016	0.0014

Table 7. Summary statistics for all assemblies generated from 1Dsq reads and the corresponding assemblies generated from 1D reads. The remaining 1D read assemblies are not included. The average and standard deviation for each group is presented. All values are plotted in Fig. 2 and S2.

Metric	Average 1D	Stdev 1D	Average 1Dsq	Stdev 1Dsq
Average Percent Identity	98.1439	11.1382	99.3950	0.4842
Proportion Mismatching Bases	0.0009	0.0021	0.0005	0.0003
Indels per 1000 Aligned Bases	2.7493	7.6205	4.2978	2.8039
Average size of indels	2.3181	0.9537	1.2542	0.0798
Spaln Value	0.8761	0.1767	0.7558	0.0891
Number of Contigs	48.9506	15.7994	66.9259	28.3376
Genome Size	11614325	497815	11175628	469973
Proportion of Reference Not Covered	0.1602	0.0938	0.1563	0.0093
Proportion of Reference Covered 1X	0.7891	0.1255	0.8105	0.0192
Proportion of Reference Covered 2X	0.0287	0.0089	0.0248	0.0126
Proportion of Reference Covered 3X	0.0052	0.0017	0.0051	0.0038
Proportion of Reference Covered 4X	0.0021	0.0012	0.0016	0.0014

Table 8. Summary statistics for all assemblies generated from pooling sets of reads or not pooling reads. The average and standard deviation for each group is presented. All values are plotted in Fig. 3 and S3.

Metric	Average WB_0157	Stdev WB_0157	Average WB_0150_0157_2331_2338	Stdev WB_0150_0157_2331_2338	Average WB_0150_0157	Stdev WB_0150_0157	Average WB_2331_2338	Stdev WB_2331_2338	Average WB_2338	Stdev WB_2338
Average Percent Identity	99.3185	1.2361	97.6735	13.5483	99.3165	1.3745	91.3444	5.9112	89.5601	5.6771
Proportion Mismatching Bases	0.0008	0.0017	0.0005	0.0002	0.0008	0.0020	0.0224	0.0160	0.0257	0.0151
Indels per 1000 Aligned Bases	3.8700	6.6986	2.8725	2.7108	3.8282	6.9632	37.3378	23.1210	44.7932	22.2634
Average Indel Size	1.8596	0.8610	1.6940	0.8190	1.8049	0.9060	1.5835	0.2623	1.6332	0.3101
Spaln Value	0.8100	0.1513	0.8193	0.1521	0.8184	0.1553	0.0067	0.0068	0.0096	0.0069
Number of Contigs	60.9074	26.6786	57.1296	23.6779	55.7778	23.4309	14.7778	14.9570	24.1482	17.3731
Genome Size	11346737	565863	11428786	543416	11409407	485056	410638	424826	637020	485480
Proportion of Reference Not Covered	0.1723	0.1106	0.1489	0.0222	0.1537	0.0197	0.9724	0.0265	0.9594	0.0280
Proportion of Reference Covered 1X	0.7927	0.1063	0.7962	0.1116	0.8105	0.0275	0.0256	0.0264	0.0388	0.0281
Proportion of Reference Covered 2X	0.0266	0.0103	0.0274	0.0123	0.0264	0.0107	0.0009	0.0019	0.0008	0.0019
Proportion of Reference Covered 3X	0.0046	0.0028	0.0054	0.0031	0.0055	0.0029	0.0006	0.0004	0.0002	0.0004
Proportion of Reference Covered 4X	0.0019	0.0014	0.0018	0.0013	0.0018	0.0014	0.0005	0.0005	0.0007	0.0009

Table 9. Summary statistics for all assemblies generated by each of the three assembly programs used. The average and standard deviation for each group is presented. All values are plotted in Fig. 4 and S4.

Metric	Average Abruijn	Stdev Abruijn	Average Canu	Stdev Canu	Average SMARTdenovo	Stdev SMARTdenovo
Average Percent Identity	93.5031	12.4071	98.8829	1.3752	99.4380	0.0901
Proportion Mismatching Bases	0.0119	0.0173	0.0029	0.0047	0.0007	0.0002
Indels per 1000 Aligned Bases	22.9040	27.2869	6.1908	6.8009	2.8029	1.3765
Average Indel Size	1.5028	0.3351	1.6114	0.5498	2.3002	1.1553
Spaln Value	0.5273	0.4089	0.6093	0.3646	0.8515	0.0824
Number of Contigs	58.7161	21.9757	51.6667	37.4373	33.8519	4.2400
Genome Size	7742127	4916432	8641157	5016929	11505586	232703
Proportion of Reference Not Covered	0.4313	0.3750	0.3653	0.3680	0.1446	0.0024
Proportion of Reference Covered 1X	0.5355	0.3633	0.6052	0.3520	0.8171	0.0083
Proportion of Reference Covered 2X	0.0165	0.0176	0.0196	0.0116	0.0302	0.0073
Proportion of Reference Covered 3X	0.0031	0.0039	0.0052	0.0034	0.0042	0.0008
Proportion of Reference Covered 4X	0.0006	0.0013	0.0027	0.0009	0.0016	0.0004

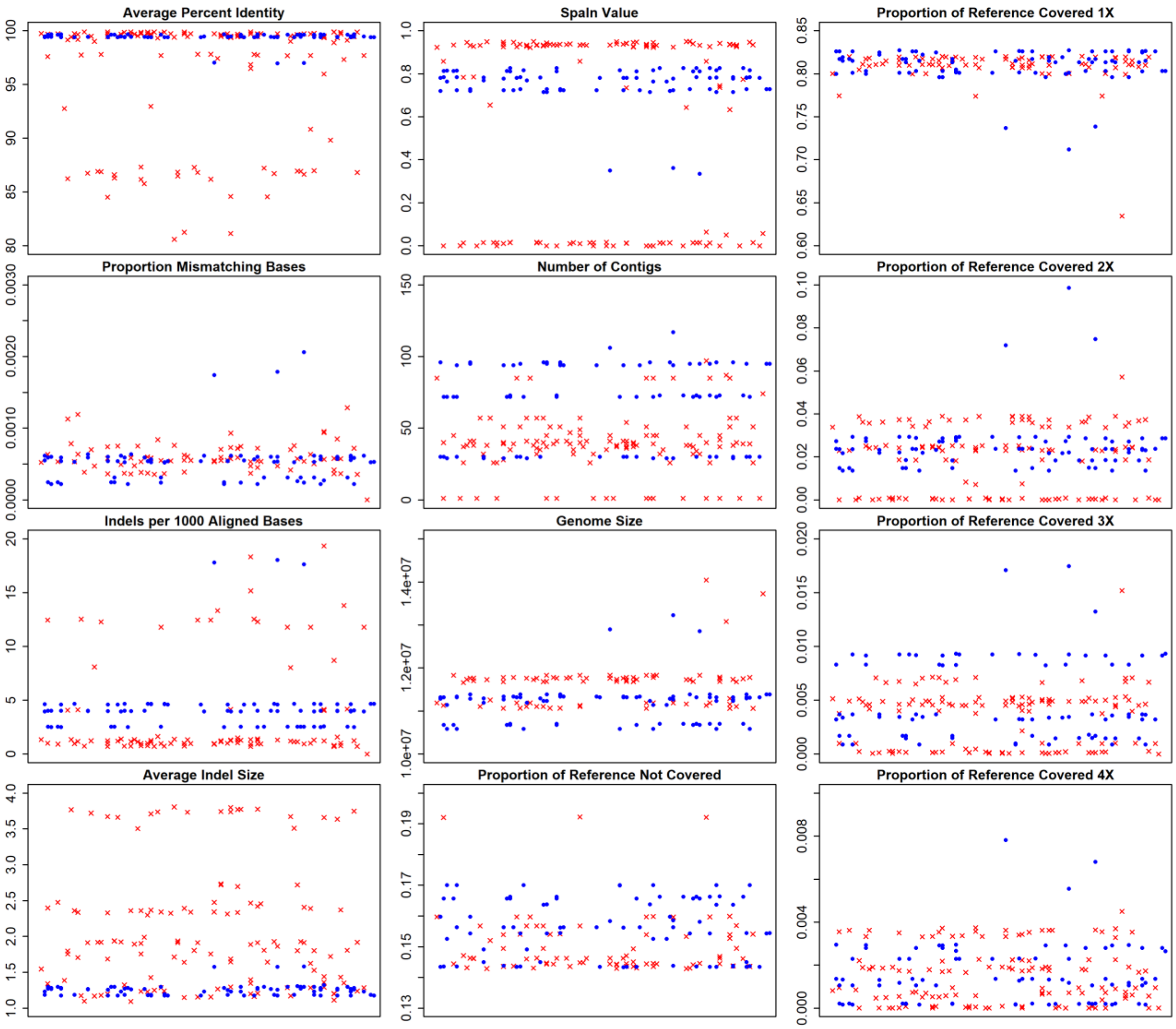

Figure 1. Performance metrics for all 1D and 1Dsq *Giardia* AWB assemblies. Red X's denote all assemblies created from 1D reads and blue circles denote all assemblies created from 1Dsq reads. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis has no units because the x-values are randomly assigned to spread out the data points for visualization purposes. Alignments of *Giardia* AWB draft genome assemblies to the *Giardia* AWB reference genome were used to calculate average percent identity (identical bases between the assembly and reference genome in aligned regions), proportion of mismatching bases (nonidentical bases between aligned regions of the assembly and reference genome), number of indels per 1000 aligned bases (insertions and deletions in assembly), and average indel size (number of base pairs per indel in the assembly). Spaln value indicates the proportion of *Giardia* AWB protein sequences that were mapped onto each assembly compared to the reference. Number of Contigs and Genome Size (in base pairs) were calculated from each assembly. Proportions of the reference genome covered denote the total proportion of bases in the reference genome that were found in each assembly 0, 1, 2, 3, or 4 times.

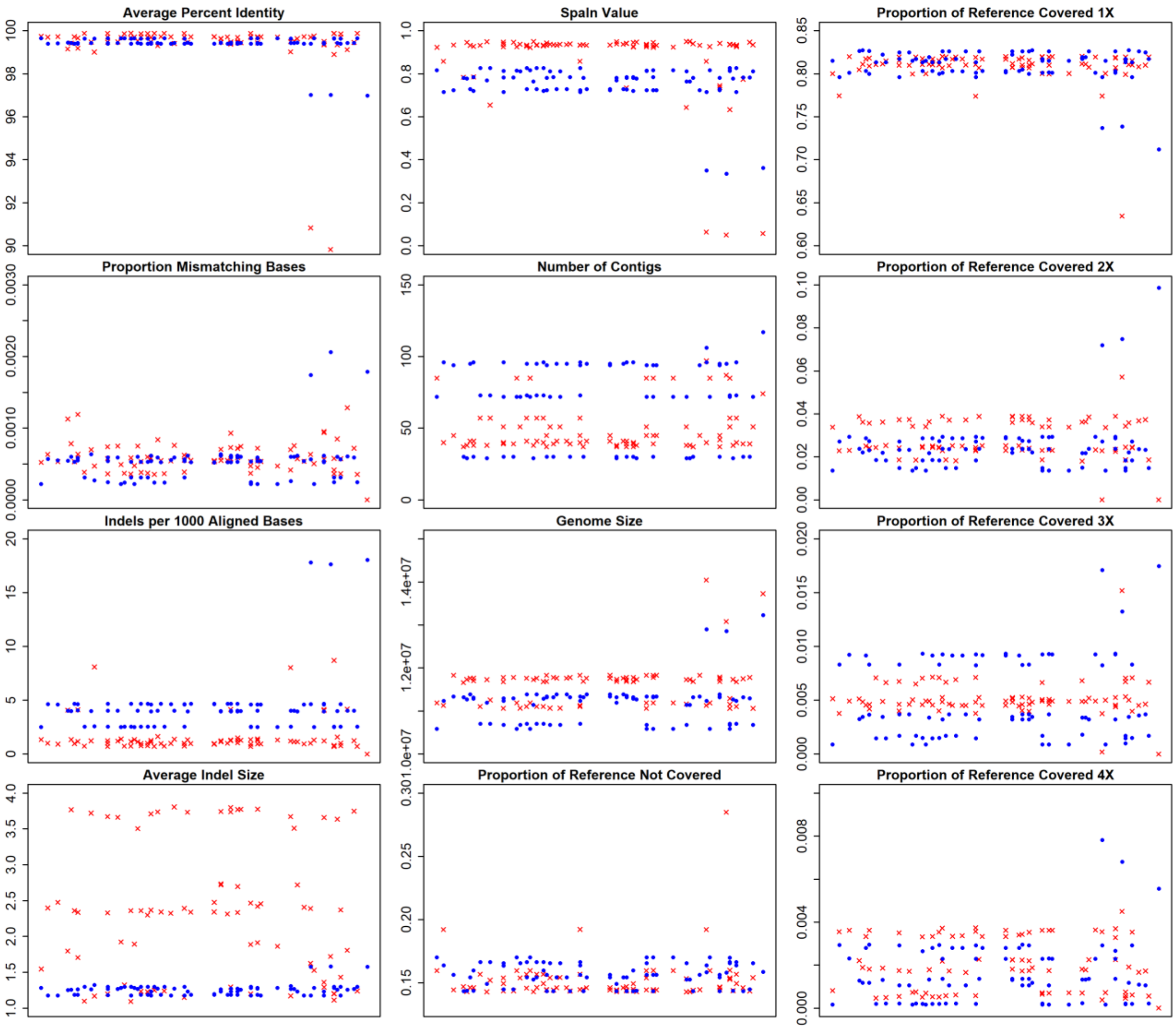

Figure 2. Performance metrics for corresponding 1D and 1Dsq *Giardia* AWB assembly pairs. Each pair was assigned the same random x-value so the points would stack on top of each other. Additional 1D assemblies are not shown. Red X's denote all assemblies created from 1D reads and blue circles denote all assemblies created from 1Dsq reads. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis has no units because x-values are random to spread out the data points for visualization. Alignments of *Giardia* AWB draft genome assemblies to the *Giardia* AWB reference genome were used to calculate average percent identity (identical bases between assembly and reference genome in aligned regions), proportion of mismatching bases (nonidentical bases between assembly and reference genome in aligned regions), number of indels per 1000 aligned bases (insertions and deletions in assembly), and average indel size (number of base pairs per indel in assembly). Spaln value indicates the proportion of *Giardia* AWB protein sequences that were mapped onto each assembly compared to the reference. Number of Contigs and Genome Size (in base pairs) were calculated from each assembly. Proportions of the reference genome covered denote the total proportion of bases in the reference genome that were found in each assembly 0, 1, 2, 3, or 4 times.

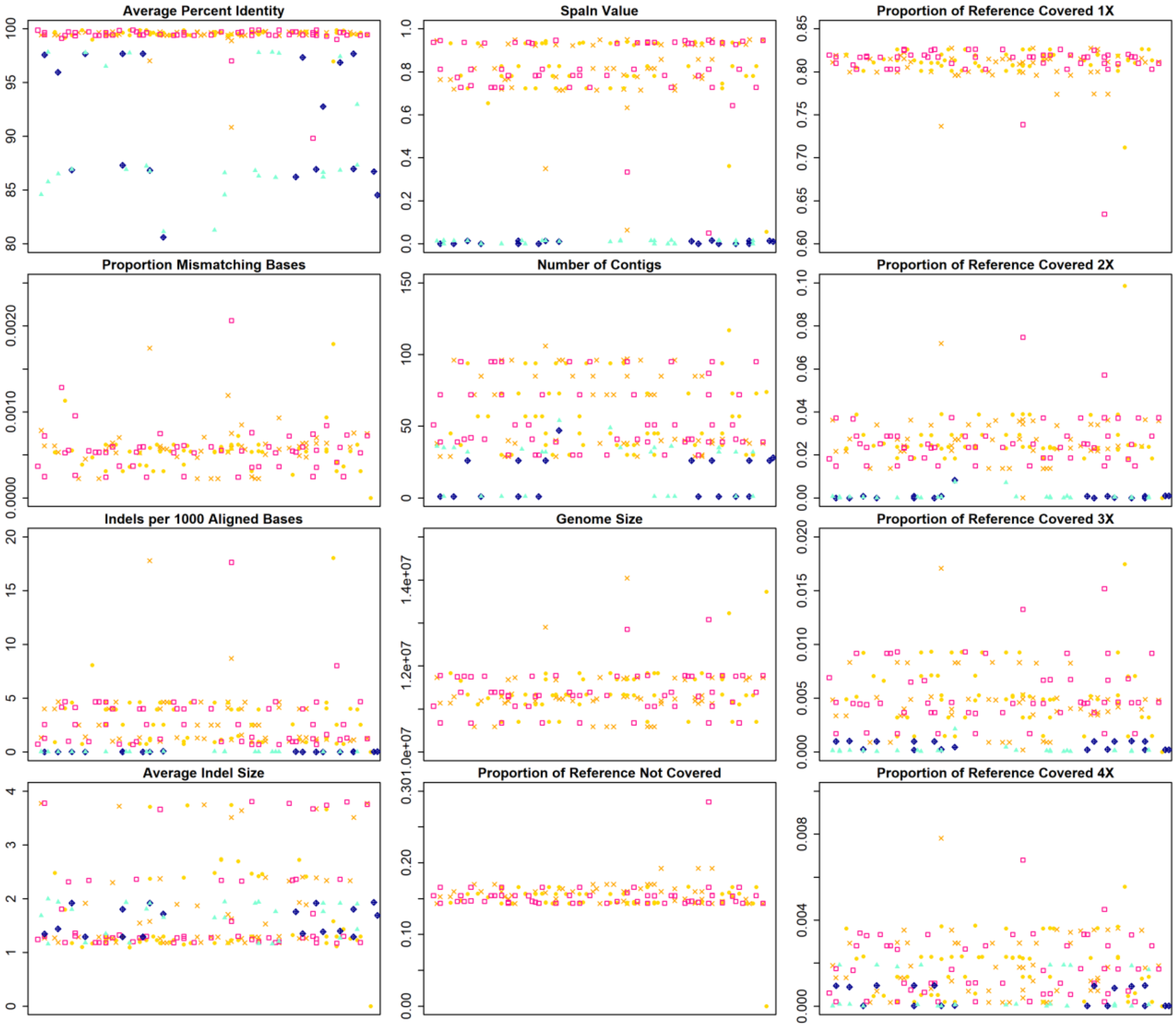

Figure 3. Performance metrics for pooled and non-pooled input *Giardia* AWB assemblies. Orange X's are non-pooled assemblies from Run1_0157 reads, gold circles are pooled assemblies from Run1 and Run2 reads (0150, 0157, 2331, and 2338 reads), pink squares are pooled assemblies from Run1 (0150 and 0157 reads), dark blue diamonds are pooled assemblies from Run2 (2331 and 2338 reads), and aquamarine triangles are non-pooled assemblies from Run2_2338 reads. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis has no units because x-values are random to spread out the data points for visualization. Alignments of *Giardia* AWB draft genome assemblies to the *Giardia* AWB reference genome were used to calculate average percent identity (identical bases between the assembly and reference genome in aligned regions), proportion of mismatching bases (nonidentical bases between aligned regions of the assembly and reference genome), number of indels per 1000 aligned bases (insertions and deletions in assembly), and average indel size (number of base pairs per indel in the assembly). Spaln value indicates the proportion of *Giardia* AWB protein sequences that were mapped onto each assembly compared to the reference. Number of Contigs and Genome Size (in base pairs) were calculated from each assembly. Proportions of the reference genome covered denote the total proportion of bases in the reference genome that were found in each assembly 0, 1, 2, 3, or 4 times.

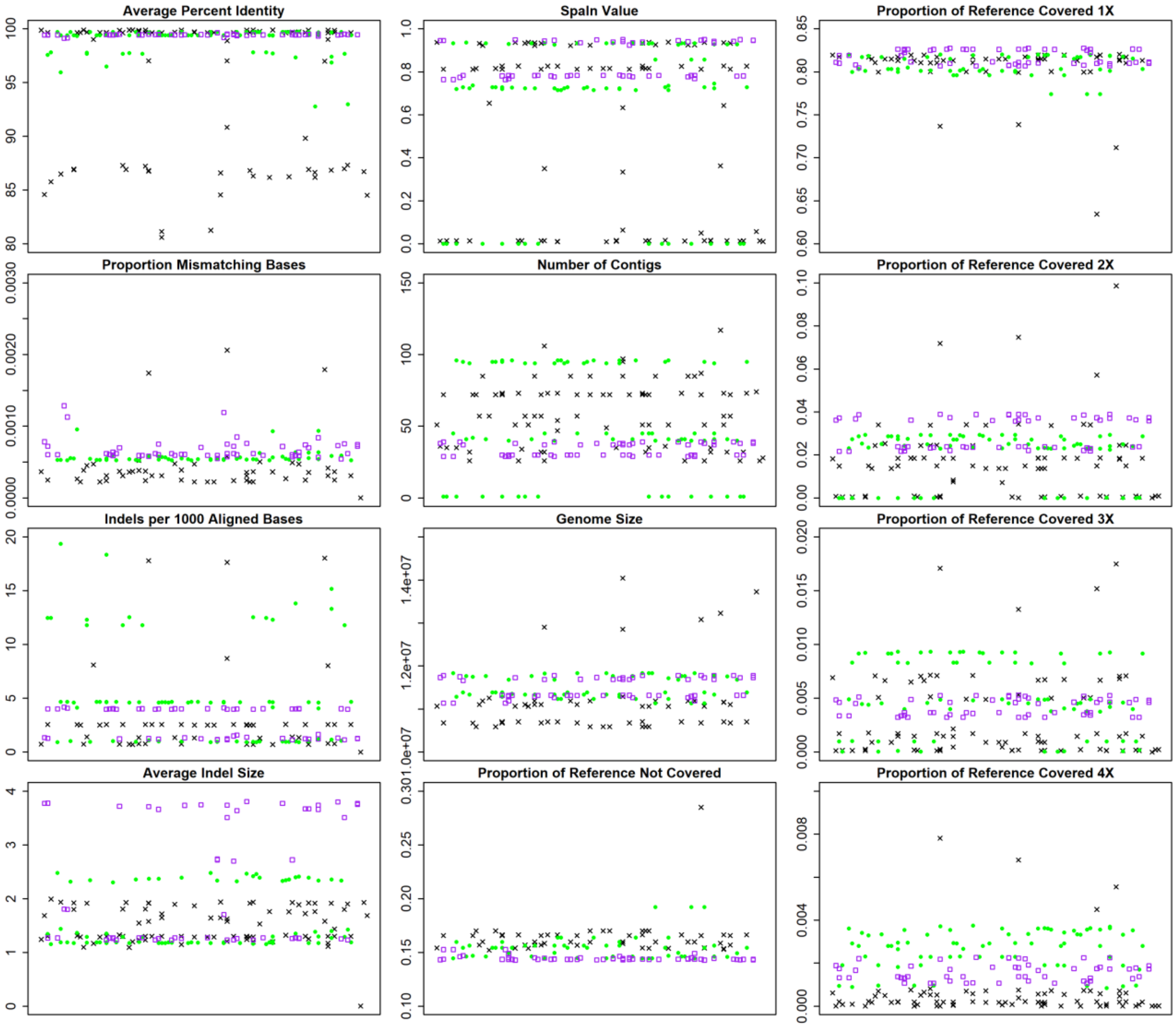

Figure 4. Performance metrics for all *Giardia* AWB assemblies, separated by assembly program. Black X's denote all Abruijn assemblies, green circles denote all Canu assemblies, and purple squares denote all SMARTdenovo assemblies. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis has no units because the x-values are randomly assigned to spread out the data points for visualization purposes. Alignments of *Giardia* AWB draft genome assemblies to the *Giardia* AWB reference genome were used to calculate average percent identity (identical bases between the assembly and reference genome in aligned regions), proportion of mismatching bases (nonidentical bases between aligned regions of the assembly and reference genome), number of indels per 1000 aligned bases (insertions and deletions in assembly), and average indel size (number of base pairs per indel in the assembly). Spaln value indicates the proportion of *Giardia* AWB protein sequences that were mapped onto each assembly compared to the reference. Number of Contigs and Genome Size (in base pairs) were calculated from each assembly. Proportions of the reference genome covered denote the total proportion of bases in the reference genome that were found in each assembly 0, 1, 2, 3, or 4 times.

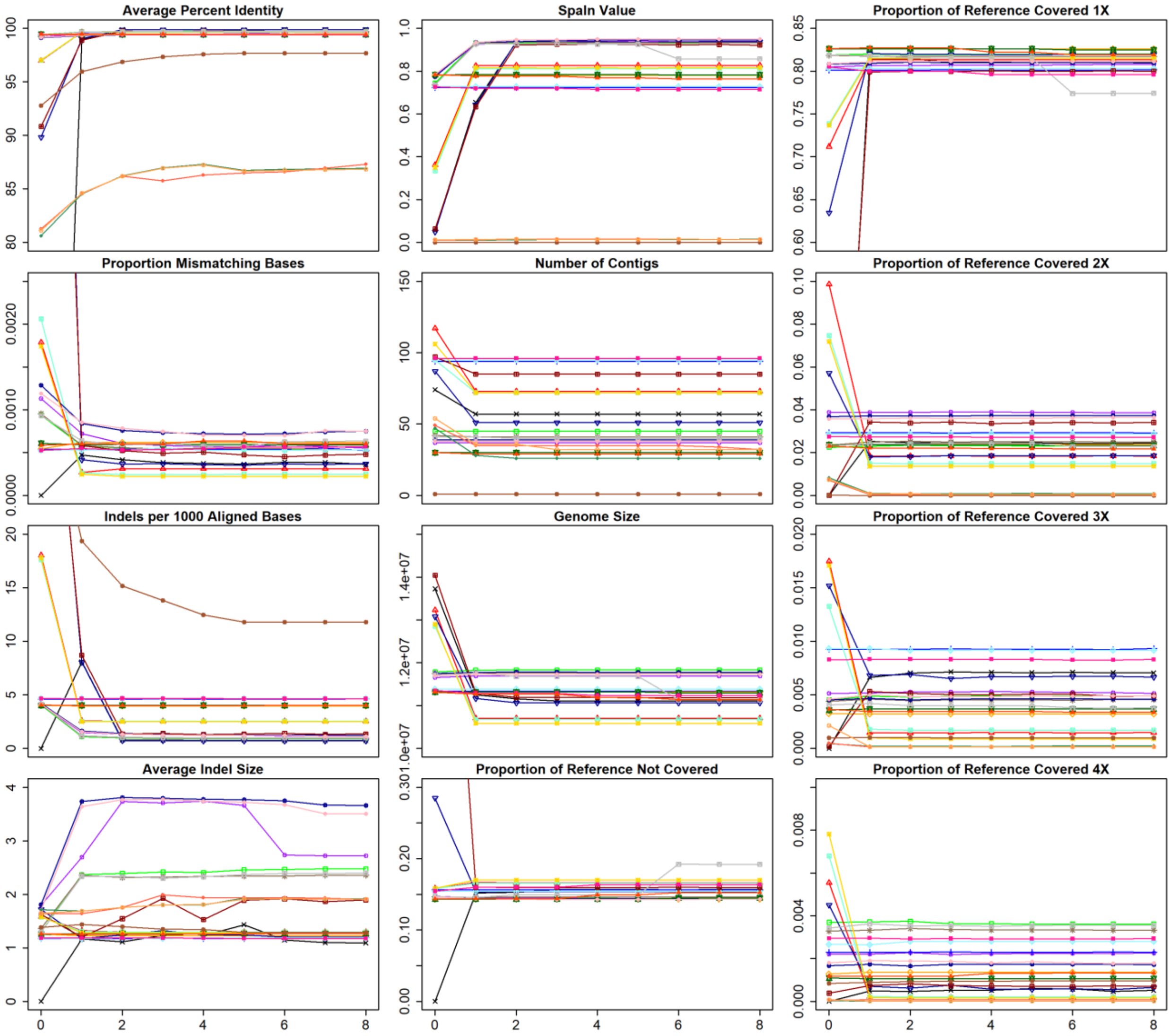

Figure 5. Performance metrics for polished sets of *Giardia* AWB assemblies. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis denotes how many times the draft assembly has been polished. Alignments of *Giardia* AWB draft genome assemblies to the *Giardia* AWB reference genome were used to calculate average percent identity (identical bases between the assembly and reference genome in aligned regions), proportion of mismatching bases (nonidentical bases between aligned regions of the assembly and reference genome), number of indels per 1000 aligned bases (insertions and deletions in assembly), and average indel size (number of base pairs per indel in the assembly). Spaln value indicates the proportion of *Giardia* AWB protein sequences that were mapped onto each assembly compared to the reference. Number of Contigs and Genome Size (in base pairs) were calculated from each assembly. Proportions of the reference genome covered denote the total proportion of bases in the reference genome that were found in each assembly 0, 1, 2, 3, or 4 times.

References

Alekseyenko,A. *et al.* (2013) Next-generation DNA sequencing informatics Cold Spring Harbor Laboratory Press, Cold Spring Harbor, NY.

Aurrecoechea,C. *et al.* (2009) GiardiaDB and TrichDB: Integrated genomic resources for the eukaryotic protist pathogens Giardia lamblia and Trichomonas vaginalis. *Nucleic Acids Res.*, **37**, 526–530.

Batovska,J. *et al.* (2017) Metagenomic arbovirus detection using MinION nanopore sequencing. *J. Virol. Methods*, **249**, 79–84.

Certad,G. *et al.* (2017) Pathogenic mechanisms of *Cryptosporidium* and *Giardia*. *Trends Parasitol.*, **33**, 561–576.

Chin,C. *et al.* (2016) Phased diploid genome assembly with single molecule real-time sequencing. *Nat. Methods*, **13**, 1050–1054.

Cock,P.J.A. *et al.* (2015) SAM/BAM format v1.5 extensions for *de novo* assemblies. *bioRxiv*, **0**, 1–3.

Feng,Y. *et al.* (2015) Nanopore-based fourth-generation DNA sequencing technology. *Genomics. Proteomics Bioinformatics*, **13**, 4–16.

Franzén,O. *et al.* (2009) Draft genome sequencing of *Giardia intestinalis* Assemblage B isolate GS: Is human giardiasis caused by two different species? *Plos Pathog.*, **5**, e1000560.

Hamada,M. *et al.* (2017) Training alignment parameters for arbitrary sequencers with LAST-TRAIN. *Bioinformatics*, **33**, 926–928.

Istace,B. *et al.* (2017) *de novo* assembly and population genomic survey of natural yeast isolates with the Oxford Nanopore MinION sequencer. *Gigascience*, **6**, 1–13.

Iwata,H. and Gotoh,O. (2012) Benchmarking spliced alignment programs including Spaln2, an extended version of Spaln that incorporates additional species-specific features. *Nucleic Acids Res.*, **40**, e161.

Johnson,S.S. *et al.* (2017) Real-time DNA sequencing in the antarctic dry valleys using the Oxford nanopore sequencer. *J. Biomol. Tech.*, **28**, 2–7.

Kielbasa,S.M. *et al.* (2011) Adaptive seeds tame genomic sequence comparison. *Genome Res.*, **21**, 487–493.

de Koning,A.P.J. *et al.* (2011) Repetitive elements may comprise over two-thirds of the human genome. *PLoS Genet.*, **7**, e1002384.

Koren,S. *et al.* (2017) Canu: scalable and accurate long-read assembly via adaptive *k*-mer weighting and repeat separation. *Genome Res.*, **27**, 722–736.

Leggett,R.M. *et al.* (2016) NanoOK: Multi-reference alignment analysis of nanopore sequencing data, quality and error profiles. *Bioinformatics*, **32**, 142–144.

Li,H. (2016) Minimap and miniasm: Fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, **32**, 2103–2110.

Li,H. and Durbin,R. (2010) Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*, **26**, 589–595.

Lin,Y. *et al.* (2016) Assembly of long error-prone reads using de Bruijn graphs. *Proc. Natl. Acad. Sci.*, **113**, E8396–E8405.

Loman,N.J. *et al.* (2015) A complete bacterial genome assembled *de novo* using only nanopore sequencing data. *Nat. Methods*, **12**, 733–736.

Lu,H. *et al.* (2016) Oxford nanopore minION sequencing and genome assembly. *Genomics. Proteomics Bioinformatics*, **14**, 265–279.

Mcfrith (2017) last-genome-alignments. https://github.com/mcfrith/last-genome-alignments.

Minervini,C.F. *et al.* (2017) Mutational analysis in BCR-ABL1 positive leukemia by deep sequencing based on nanopore MinION technology. *Exp. Mol. Pathol.*, **103**, 33–37.

Morrison,H.G. *et al.* (2007) Genomic minimalism in the early diverging intestinal parasite Giardia lamblia. *Science (80-.).*, **317**, 1921–1926.

Quinlan,A.R. and Hall,I.M. (2010) BEDTools: A flexible suite of utilities for comparing genomic features. *Bioinformatics*, **26**, 841–842.

R,C.T. (2013) R: A language and environment for statistical computing.

Rand,A.C. *et al.* (2017) Mapping DNA methylation with high-throughput nanopore sequencing. *Nat. Methods*, **14**, 411–413.

Rhoads,A. and Au,K.F. (2015) PacBio sequencing and its applications. *Genomics. Proteomics Bioinformatics*, **13**, 278–289.

Ruan,J. (2017) Ultra-fast de novo assembler using long noisy reads. https://github.com/ruanjue/smartdenovo.

Simpson,J. (2017) Signal-level algorithms for MinION data. https://github.com/jts/nanopolish.

Tyson,J.R. *et al.* (2017) Whole genome sequencing and assembly of a Caenorhabditis elegans genome with complex genomic rearrangements using the MinION sequencing device. *bioRxiv*.

Vera,D. (2017) Dockerfile for the Albacore basecaller from Oxford Nanopore. https://github.com/dvera/albacore.

Votintseva,A.A. *et al.* (2017) Same-day diagnostic and surveillance data for Tuberculosis via whole-genome sequencing of direct respiratory samples. *J. Clin. Microbiol.*, **55**, 1285–1298.

Wick,R. (2017) A comparison of different Oxford Nanopore basecallers. https://github.com/rrwick/Basecalling-comparison#m.

Supplementary Data

The full dataset of values calculated for each alignment/assembly can be found at:

https://github.com/stweebs/Supplementary_Data_GI

Supplementary Figures

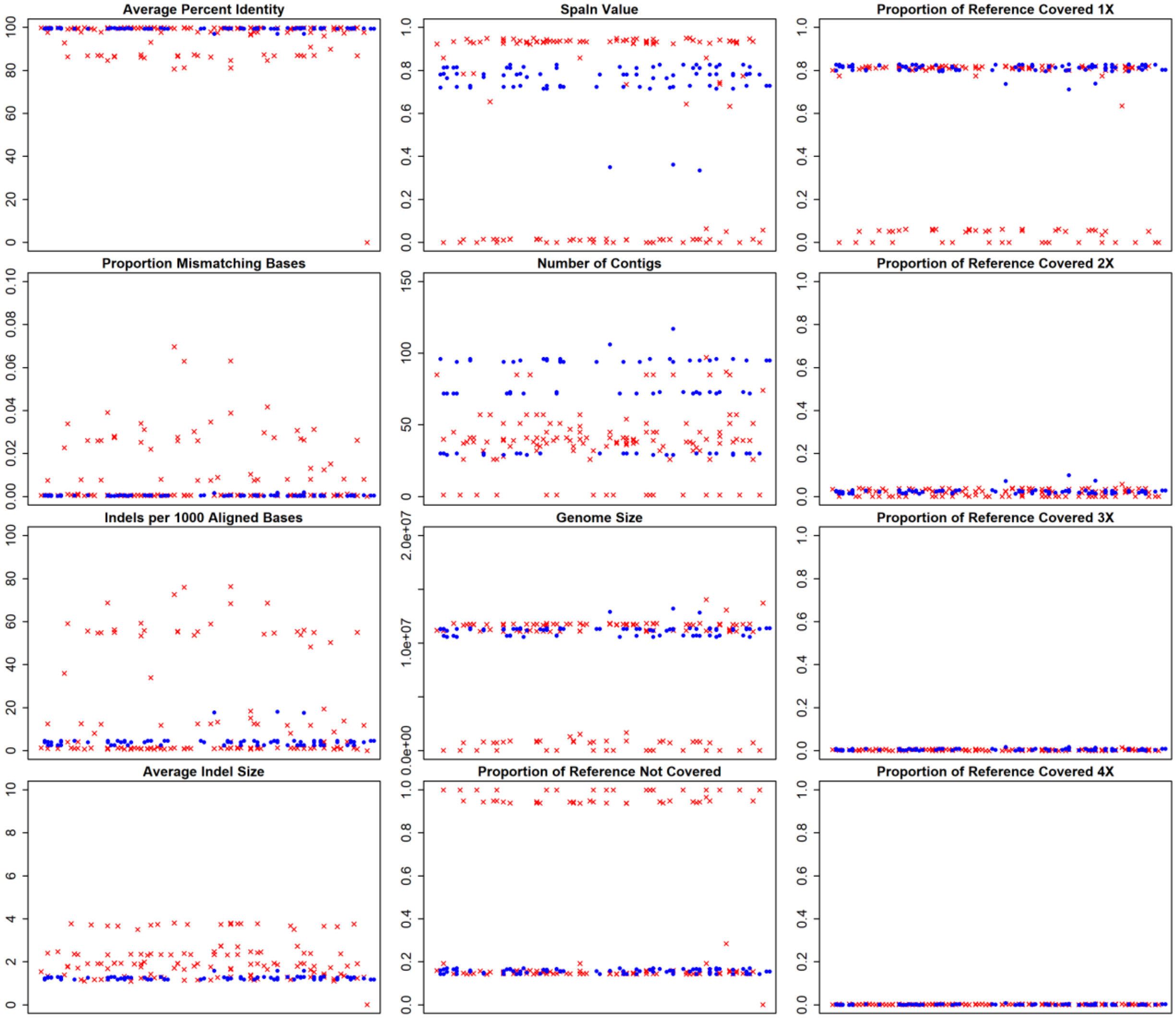

Figure S1. Alternate view of performance metrics for all 1D and 1Dsq *Giardia* AWB assemblies. Red X's denote all assemblies created from 1D reads and blue circles denote all assemblies created from 1Dsq reads. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis has no units because the x-values are randomly assigned to spread out the data points for visualization purposes. The units from Fig. 1 are extended here to show all outlier data. All metrics were calculated as described in the Fig. 1 legend and the main text Methods.

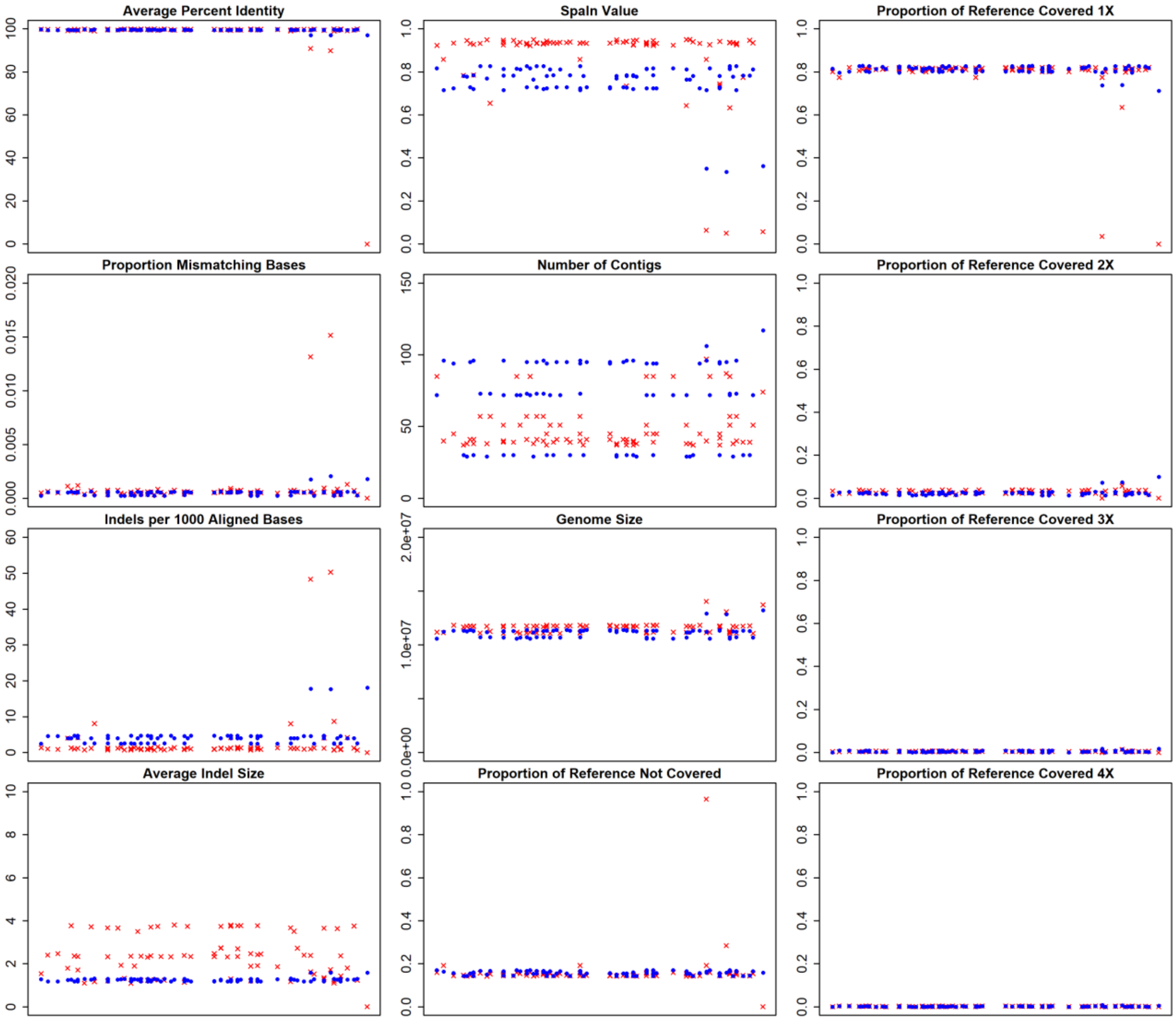

Figure S2. Alternate view of performance metrics for corresponding 1D and 1Dsq *Giardia* AWB assembly pairs. All assemblies made from 1Dsq reads had a corresponding assembly made from 1D reads from the same inputs. Each pair was assigned the same random x-value so that corresponding 1D and 1Dsq assemblies would stack on top of each other in each plot. The additional assemblies from 1D reads are not shown. Red X's denote all assemblies created from 1D reads and blue circles denote all assemblies created from 1Dsq reads. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis has no units because the x-values are randomly assigned to spread out the data points for visualization purposes. The units from Fig. 2 are extended here to show all outlier data. All metrics were calculated as described in the Fig. 2 legend and the main text Methods.

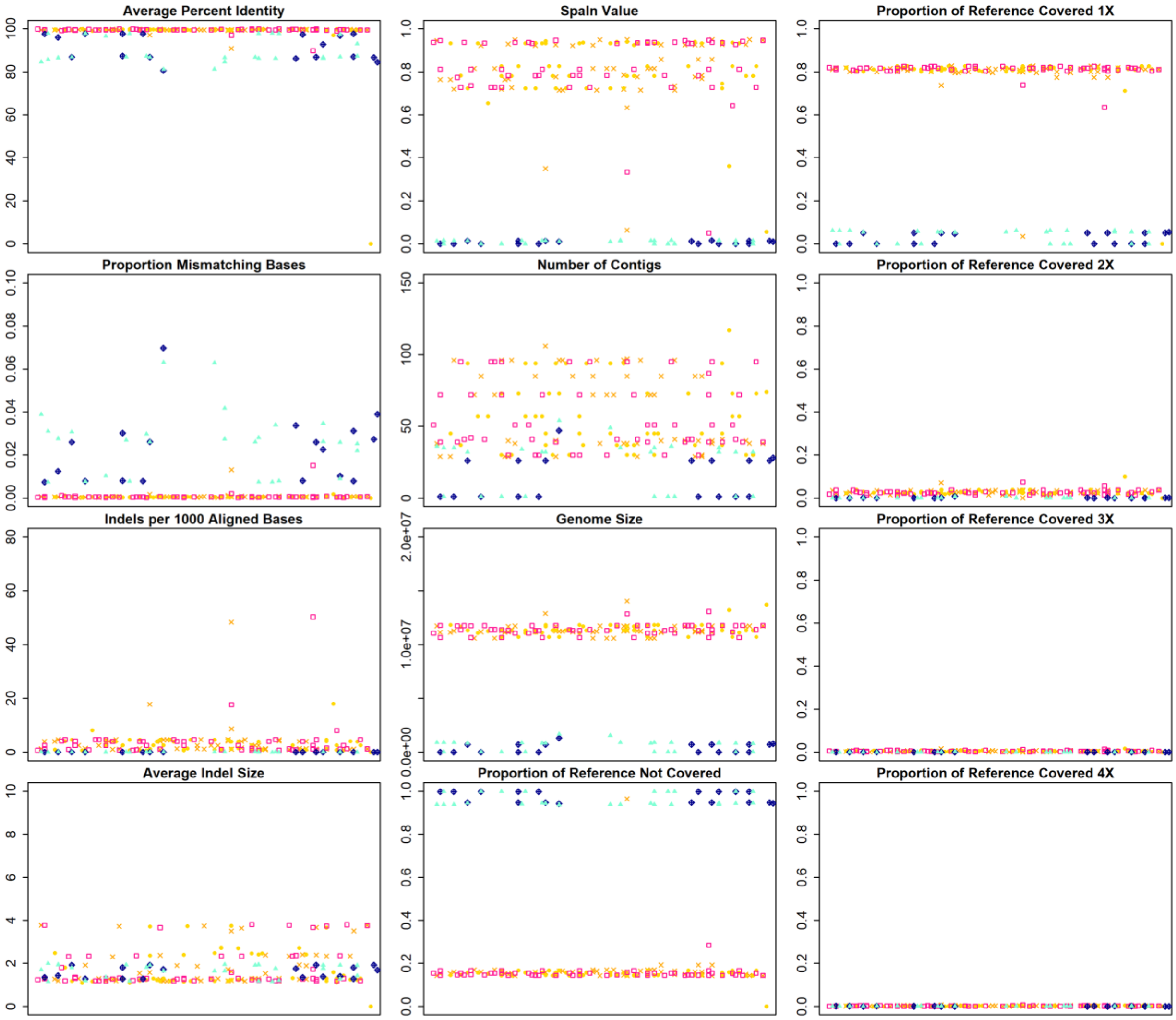

Figure S3. Alternate view of performance metrics for pooled input and non-pooled input *Giardia* AWB assemblies. Orange X's denote the non-pooled assemblies made from Run1_0157 reads, gold circles denote the pooled assemblies made from all reads from both Run1 and Run2 (0150, 0157, 2331, and 2338 reads), pink squares denote the pooled assemblies made from Run1 data only (0150 and 0157 reads), dark blue diamonds denote the pooled assemblies made from Run2 data only (2331 and 2338 reads), and aquamarine triangles denote the non-pooled assemblies made from Run2_2338 reads. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis has no units because the x-values are randomly assigned to spread out the data points for visualization purposes. The units from Fig. 3 are extended here to show all outlier data. All metrics were calculated as described in the Fig. 3 legend and the main text Methods.

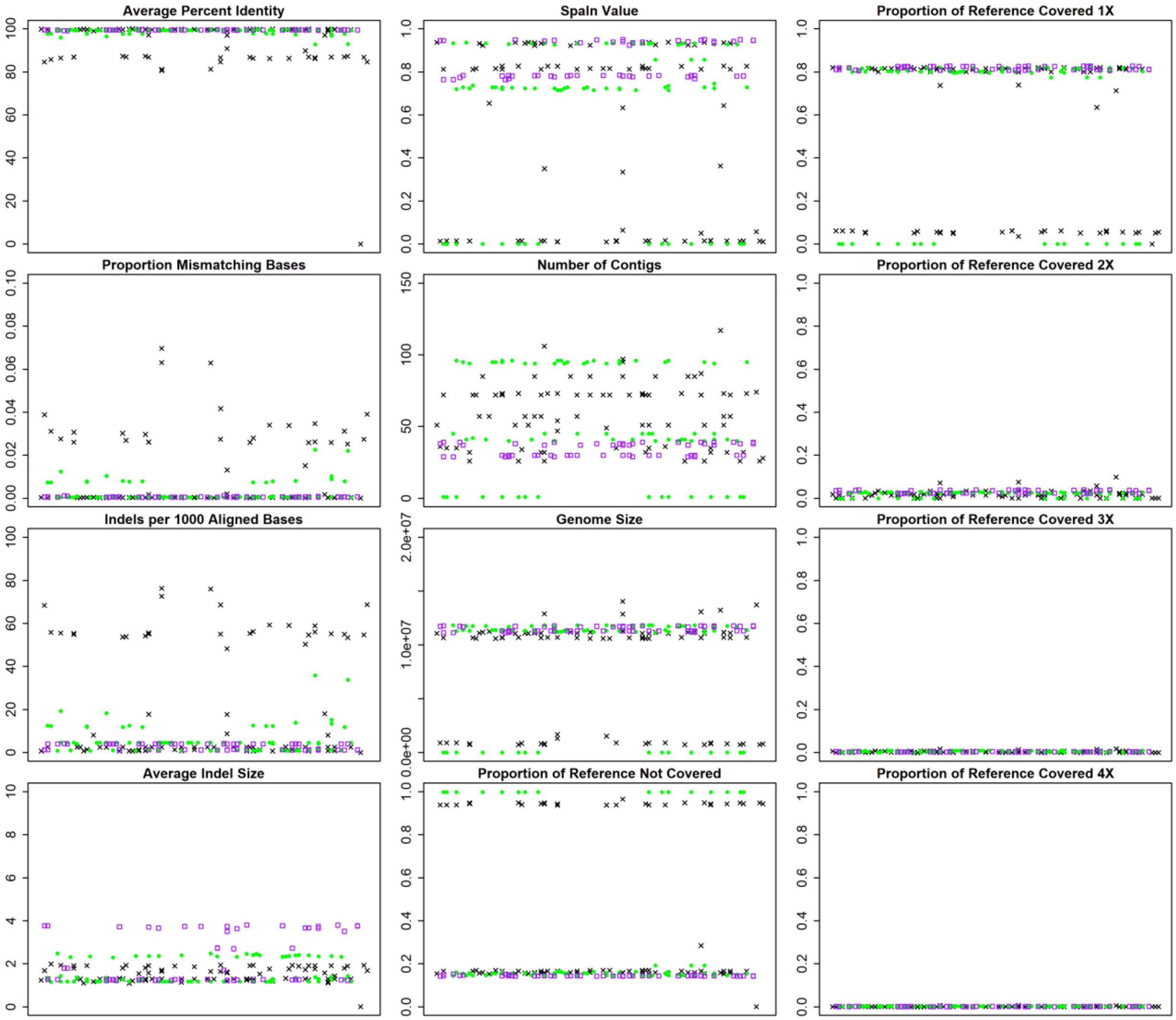

Figure S4. Alternate view of performance metrics for all *Giardia* AWB assemblies, separated by assembly program. Black X's denote all Abruijn assemblies, green circles denote all Canu assemblies, and purple squares denote all SMARTdenovo assemblies. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis has no units because the x-values are randomly assigned to spread out the data points for visualization purposes. The units from Fig. 4 are extended here to show all outlier data. All metrics were calculated as described in the Fig. 4 legend and the main text Methods.

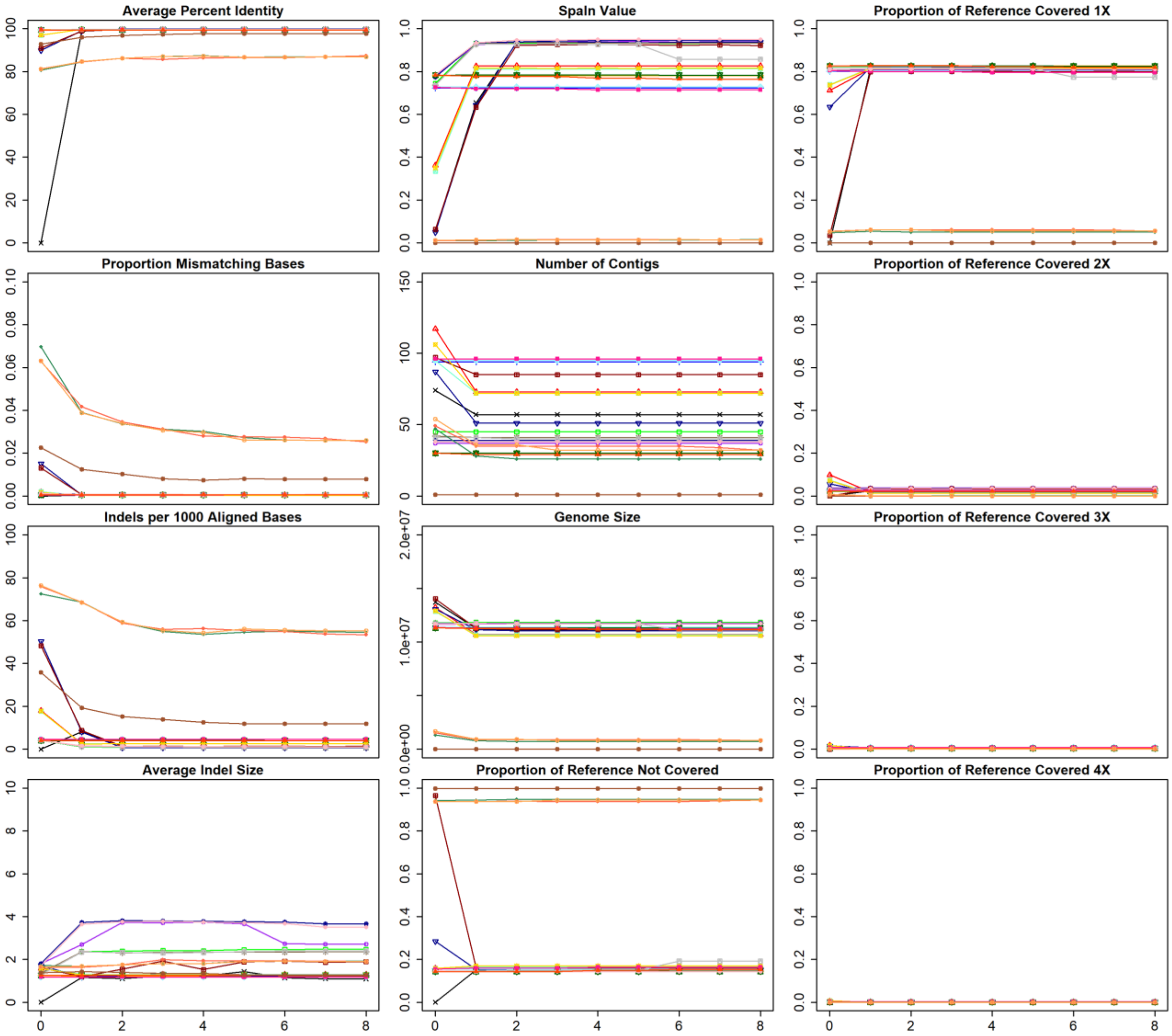

Figure S5. Alternate view of performance metrics for polished sets of *Giardia* AWB assemblies. The title above each scatterplot denotes the metric being plotted on the y-axis. The x-axis denotes how many times the draft assembly has been polished. The units from Fig. 5 are extended here to show all outlier data. All metrics were calculated as described in the Fig. 5 legend and the main text Methods.