Streaming approaches to error detection and trimming in the analysis of short sequencing reads

TBD

December 21, 2014

1 Introduction

K-mer spectral analysis is a powerful approach to error detection and correction in shotgun sequencing data that uses k-mer abundances to determine likely errors (cite Pevzner, Quake, khmer-counting paper). Approaches derived from spectral analysis can be very accurate: Zhang et al. (2014) show that spectral analysis is considerably more effective at finding errors than quality-based approaches (cite). However, spectral analysis is also very compute intensive; most implementations must count k-mers across entire sequencing data sets, which can be memory- or I/O-intensive for large data sets.

Streaming algorithms can offer improved algorithmic and computational efficiency in the analysis of large data sets (cite). Streaming algorithms typically examine the data only once, and scale in memory usage sublinearly with respect to the input data. Streaming algorithms have not been applied to k-mer spectral analysis of reads, although XXX (melsted).

Brown et al. (2012) introduced a streaming algorithm for normalizing k-mer abundance spectra, termed "digital normalization" (abbreviated as "diginorm"). This procedure estimates the k-mer coverage of each read in an online algorithm, by calculating the median k-mer abundance of the read given all previous reads; reads above a certain estimated coverage are set aside and their k-mers are not tracked. This algorithm is both online and *streaming* because it only collects k-mers in reads with a low estimated coverage; for the same reason, it is sublinear in memory for high coverage data sets. The net effect of diginorm is to reduce the data set size that must be considered for downstream processing, such as *de novo* assembly (cite trinity, elijah, etc.)

Here we develop a streaming algorithm for k-mer spectral analysis, based on digital normalization, that can detect and remove errors in sequencing reads. This algorithm operates in sublinear memory, and examines the data at most twice. The approach offers a general framework for making use of locus-specific graph saturation and could potentially be used for error correction, variant calling, and a streaming implementation of assembly. Moreover, it can be applied to data sets with variable coverage such as transcriptomes, metagenomes, and amplified genomic DNA.

Name	Origin	Number of reads	Description
simple genome	10,000	No repeats	
E. coli MG1655	5,000,000	Subset of NCXXXXX	
simple metagenome	2,347	Dynamic range of YYY	
simple transcriptome	568	Dynamic range of ZZZ; multiple exons	

Table 1: Data sets used

2 Results

2.1 Coverage-normalized data can be used to locate and correct errors in high-coverage shotgun sequencing data

Our initial question was whether we could apply spectral error analysis to genomic short read data after digital normalization. We tested this on a synthetic data set and an *E. coli* data set. We then compared the performance of Quake on the raw and digitally normalized counts from the *E. coli data*.

Simulated data: We first applied digital normalization to a simulated data set with known errors. We generated the synthetic data set from a simulated low-complexity genome ("simple genome"; see Methods for generation and Table 1 for data set details). We then applied digital normalization to these synthetic reads, normalizing to a median 20-mer coverage of 20 (k=20, C=20).

The k-mer spectrum before and after digital normalization is shown in Figure 1. While the total number of k-mers decreased in the digitally normalized data set, the separation between the high count k-mers and the low-count k-mers remains clear. The key concept underlying k-mer spectral error analysis is that in a high-coverage data set, these high count k-mers will represent *correct* k-mers, while the low count k-mers are produced by errors in the reads. Simple classification methods suffice to identify and trim or correct these low-count k-mers.

We next used k-mer counts from the downsampled read set to detect errors in the original read set. The algorithm is straightforward: we look for bases at the beginning or ends of low-abundance runs of k-mers in each read. We used a "trusted k-mer" cutoff of $C_0 = 3$ as our abundance cutoff, below which we assumed k-mers were erroneous (see Methods). The results are presented in Table 2. Of the 531 simulated reads from the simple genome containing one or more errors, predicted errors matched the known truth exactly for 443 of them (true positives), and 466 reads were correctly predicted to contain no errors (true negatives). 0 reads were falsely predicted to have no errors (false negatives). The errors in 88 reads were miscalled – while the reads each had one or more errors, the positions were not correctly called – and three reads were incorrectly predicted to contain errors, leading to a total of 91 false positives. Using the above definitions, we calculated the prediction sensitivity to be 100%

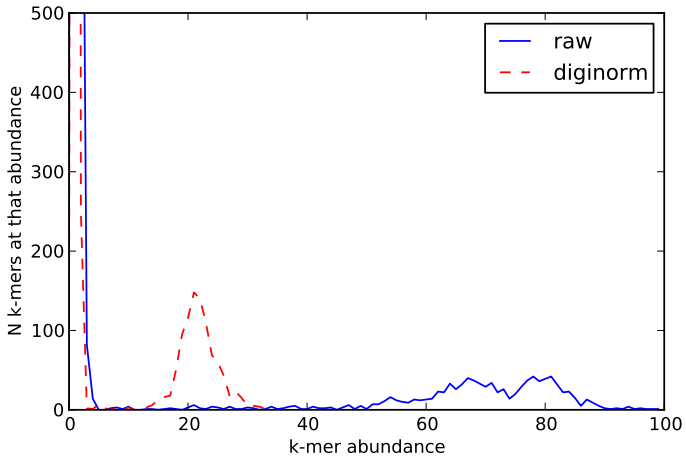

Figure 1: **K-mer spectrum of a simple artificial data set, before and after digital normalization. The peaks at 1 represents erroneous k-mers resulting from (simulated) error; the peaks centered at 80 (raw) and 20 (diginorm) represent k-mers truly present in the genome which are present in many reads.**

and the prediction specificity to be 83.0%.

When we applied spectral error detection to the unnormalized reads, we saw similar results: 440 TP, 455 TN, 105 FP, and 0 FN, for a sensitivity of 100% and a specificity of 80.7% (Table 2). The only difference in parameters for this analysis was that we used a cutoff of $C_0 = 10$, to account for the shifted k-mer spectrum from diginorm.

***E. coli* reads:** We next applied digital normalization and k-mer spectral error detection to an Illumina data set from *E. coli* MG1655 (cite Chitsaz). In real reads, we do not know the location of errors; to calculate them, we mapped 5m untrimmed reads to the known *E. coli* MG1655 genome with bowtie1 (cite) and recorded mismatches between the reads and the genome. These mismatches were taken as errors in the reads. We found 8.0m errors in 2.2m reads, for an overall error rate of 1.60%.

We then compared the results of k-mer spectral error detection with and without digital normalization. We used the same parameters as on the simulated genome. The results are presented in Table 3. Using the raw counts, the sensitivities were close – using the raw counts, we achieved a sensitivity of 72.0%, versus 71.4% using the counts from the digitally normalized reads. The specificities were also comparable – 43.5% using the raw counts, and 42.9% using the digitally normalized counts.

(Simple genome)	Raw counts	Diginorm counts
Perfect detection (TP)	474	485
No errors (TN)	355	366
Miscalled errors (FP)	159	148
Mispredicted errors (FP)	12	1
Missed errors (FN)	0	0
Sensitivity	100%	100%
Specificity	73.5%	76.5%

Table 2: Spectral error detection on 1000 synthetic reads from a simulated 10kb genome, using raw and digitally normalized counts. The counts are the number of reads where all errors were detected perfectly (TP), no errors were present (TN), one or more errors were miscalled (one type of FP), errors were mistakenly called in an error-free read (the other type of FP), and errors present in a read were missed (FN).

(E. coli)	Raw counts	Diginorm counts
Distinct k-mers	39,677,503	26,296,651 (66%)
Perfect detection (TP)	1,042,325	1,030,787
No errors (TN)	2,782,265	2,782,413
Miscalled errors (FP)	1,133,049	1,140,148
Mispredicted errors (FP)	474	326
Missed errors (FN)	2,135	6,574
Sensitivity	99.8%	99.4%
Specificity	47.9%	47.5%

Table 3: Spectral error detection on 5m *E. coli* reads, using raw and digitally normalized counts.

Sample	raw read M	normalized read M
E. coli		

Table 4: Unique k-mer counts (M) for raw and normalized data sets using a k-mer size of 20 and the specified coverage cutoff. Digital normalization reduces the total number of k-mers being tracked.

Sample	raw read M	normalized read M
E. coli		

Table 5: Unique k-mer counts for raw and normalized data sets using a k-mer size of 20 and the specified coverage cutoff. Digital normalization reduces the total number of reads considered informative.

***E. coli* error correction with Quake:** While the results above suggest that simple spectral error detection works equally well both before and after digital normalization, we wanted to see if we could successfully perform a full error *correction* after digital normalization. To evaluate this, we applied the Quake error corrector to the data. Because Quake permits the use of a different k-mer counting table from the data being corrected, we separately used the counts from the raw reads and the digitally normalized reads to correct the raw reads, and compared the results (see Methods for details).

The results of running Quake on the raw data using counts from the raw and digitally normalized data are shown in Table **??**. The performance was essentially the same: Quake brought the overall error rate in the data set from 1.60% (8.0m errors) to 0.005%-0.006% (23,000 - 29,000 errors).

These results demonstrate that digitally normalized data retains all of the information necessary for effective error correction with Quake, despite having many fewer distinct k-mers.

2.2 Coverage-normalized data can be used to locate errors in variable coverage shotgun sequencing data

Digital normalization works on both genomic data, with even coverage, and variable coverage data such as transcriptome and metagenome data (cite diginorm, elijah). However, one of the drawbacks of spectral abundance analysis is that it cannot be directly applied to data with uneven coverage. For example, metagenomic or transcriptomic data sets typically contain reads from both high-abundance and low-abundance molecules. This in turn leads to high coverage and low coverage reads. Variability in coverage confounds naive spectral analysis for two reasons: first, erroneous k-mers from very high abundance regions can accumulate and increase in abundance over the threshold for trusted k-mers, thus appearing to be correct; and second, correct reads from low coverage regions yield k-mers below the trusted k-mer threshold that appear to be incorrect. In practice, therefore, error analysis for metagenomic and transcriptome data uses other approaches than direct spectral error analysis.

Using digital normalization, we developed a general approach that enables spectral error analysis to be used for variable coverage data. We then applied this to two synthetic data sets as well as two real data sets, a mock shotgun metagenome and mRNAseq data from mouse.

Coverage-normalized spectral error analysis: Using digital normalization, we should be able to address both of these problems. First, by applying digital normalization to variable coverage data and then working only with the k-mer counts from the normalized reads, we can avoid counting high abundance errors. And second, by ignoring reads with a low estimated coverage, we can avoid misclassifying true low-abundance k-mers. The process is shown in Figure 2.

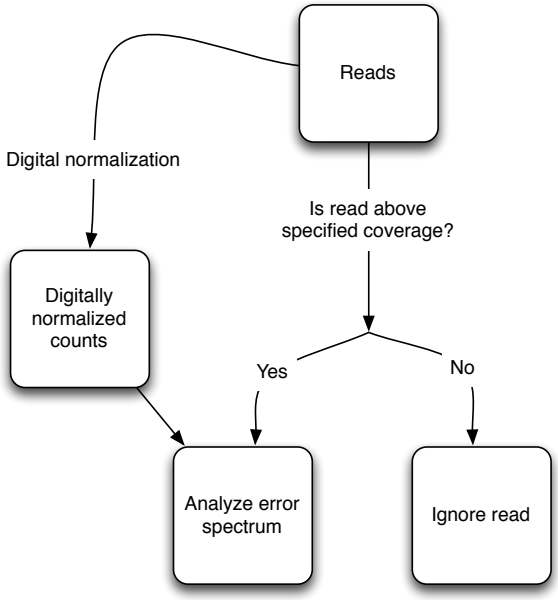

Figure 2: Coverage-normalized spectral error analysis. Reads are normalized, and high-coverage reads are subjected to spectral error analysis with the normalized counts, while low-coverage reads are ignored.

Simulated data: To test this approach, we generated two more synthetic data sets, "simple metagenome" and "simple mRNAseq," which contain both high- and low-abundance species (see Table 1 for data set details). After generating synthetic reads with a 1% error rate and applying digital normalization (k=20/C=20), we again did spectral error detection using the normalized counts. However, this time we modified the algorithm so that only reads with a median k-mer abundance of 20 or greater were examined (see Methods).

The results of running error detection on the synthetic metagenome and mRNAseq data sets are shown in Table 6. For the simple metagenome data set, 2254 of 2347 reads (96.0%) met the coverage criterion. Of the 2254 reads analyzed, the errors in 978 erroneous reads were called perfectly (TP) and 1098 of the reads with no errors were correctly called as error-free (TN). 2 reads were incorrectly determined to be error free (FN). Of the remaining 176 reads, 170 were miscalled (errors existed but were not exactly called) and 6 were incorrectly called as erroneous when they were in fact correct. We calculated the prediction sensitivity to be 99.9% and the prediction specificity to be 84.7%. For the simple mRNAseq data set, 524 of 568 reads (92.3%) met the coverage criterion, with 228 true positives, 235 true negatives, 0 false negatives, and 61 false positives, for a prediction sensitivity of 100.0% and prediction specificity of 78.9%. In both cases low-coverage reads were not included in the statistics.

The results of the variable coverage error detection approach on synthetic

	simple metagenome	simple mRNAseq
K-mer coverage threshold	20	20
Total reads	2347	568
High coverage reads	2254 (96.0%)	524 (92.3%)
Perfect detection (TP)	978	228
No errors (TN)	1098	235
Miscalled errors (FP)	170	52
Mispredicted errors (FP)	6	9
Missed errors (FN)	2	0
Sensitivity	99.8%	100%
Specificity	84.7%	78.9%

Table 6: Variable coverage spectral error detection on two synthetic data sets, a simple metagenome and a simple mRNAseq data set. Per-read coverage was estimated by median k-mer abundance within the read, and only the reads with estimated coverage at or above the specified threshold were analyzed. Digitally normalized counts were used for the spectral error analysis.

metagenomes and mRNAseq are comparable to the results on the synthetic genome (100.0% sensitivity and 83.0% specificity with the same parameters; Table 2).

Mock metagenome data: To evaluate variable coverage spectral error analysis on real data (albeit from a synthetic community), we applied our approach to 5m reads from a diverse mock community data set (Shakya et al., 2013). We used a coverage threshold of 10 for digital normalization, and found that of the 5m reads, 404,896 were at or above this coverage threshold. After calling errors in the reads by mapping them back to the known genomes, we used spectral analysis to identify putative errors. The results are shown in Table 7, second column. We achieve 90.5% sensitivity and 41.4% specificity on the high coverage reads.

mRNAseq data: We next applied variable coverage spectral error analysis to 1m mouse mRNAseq reads (cite Trinity), with the same parameters as above. Here errors were again calculated by mapping the reads to the known reference and finding mismatches. The results are shown in Table 7, third column. We achieve 92.9% sensitivity and 80.7% specificity on the 340,000 high coverage reads in this data set.

Error correcting variable coverage data with Quake: There are many sophisticated error correction algorithms implemented for shotgun genome data, but relatively few work directly on variable coverage data such as mRNAseq. It should be possible to apply *any* genomic error correction algorithm to the high coverage components of data sets after digital normalization.

	Mock metagenome	mouse mRNAseq
K-mer coverage threshold	10	20
Total reads	4,667,491	791,687
High coverage reads	404,896 (8.7%)	260,641 (32.9%)
Perfect detection (TP)	11,743	90,954
No errors (TN)	366,253	141,009
Miscalled errors (FP)	3,911	20,547
Mispredicted errors (FP)	21,750	1173
Missed errors (FN)	1239	6931
Sensitivity	90.5%	92.9%
Specificity	31.4%	80.7%

Table 7: The results of variable coverage spectral error detection on two real variable coverage data sets, a mock shotgun metagenome and a mouse mRNAseq data set. Per-read coverage was estimated by median k-mer abundance within the read, and only the reads with estimated coverage at or above the specified threshold were analyzed. Digitally normalized counts were used for the spectral error analysis.

To evaluate this, we tried using Quake, a genomic error corrector, to correct the high coverage mRNAseq reads. We first extracted the 260,641 reads with estimated coverage greater than or equal to 20 from the mouse mRNAseq data set, and then digitally normalized the data. We next applied the Quake error corrector to the unnormalized high-coverage reads using the k-mer counts from the normalized reads, as with the *E. coli* data set. Quake discarded 23,606 reads and corrected the remainder.

We evaluated the corrected reads by mapping them against the mouse transcriptome, and found that, for the reads that Quake did not discard, Quake had reduced the overall error rate from 1.28% (230,125 mismatches in 18,012,456 bp total) to 0.43% (74,797 in 17,254,585 total). As with the *E. coli* correction above, this suggests that sufficient information remains in the digitally normalized data to do error correction.

2.3 A streaming algorithm for error analysis

The spectral error detection approach outlined above is a 2-pass offline algorithm for any given data set - the first pass normalizes the read set and records the k-mer abundances, while the second pass analyzes the reads for low-abundance k-mers. Even with digital normalization reducing the number of k-mers under consideration, this 2-pass approach is time consuming on large data sets. We therefore developed a general streaming approach that considers many of the reads only once, by making use of the redundancy of shotgun sequencing.

Streaming analysis of coverage-saturated regions: Shotgun sequencing oversamples most regions – for example, for a 100x coverage genomic data set,

we would expect 50% or more of the genome to be represented by more than 100 reads. This is a consequence of the Poisson-random sampling that underlies shotgun sequencing (cite... Waterman?) This oversampling provides an opportunity, however: if we regard the read data set as a stream of incoming data, some portions of the reference will more highly sampled earlier in the stream than others. For example, in mRNAseq, highly expressed transcripts will on average be highly sampled much earlier than low-expressed transcripts.

We can adapt the same approaches used in previous sections to do error analysis in a streaming fashion, by detecting high-coverage reads *during* the first pass. The conceptual idea is presented in Figure 3. On the first pass, low-coverage reads would be incorporated into the k-mer database and set aside for later analysis, while high-coverage reads would be analyzed for errors. On the second pass, the set aside reads would be checked for coverage again, and either ignored or analyzed for errors. Crucially, this second pass involves *at most* another full pass across the data, but only when the entire data set is below the coverage threshold; the larger the high coverage component of the data, the smaller the fraction of the data that is examined twice.

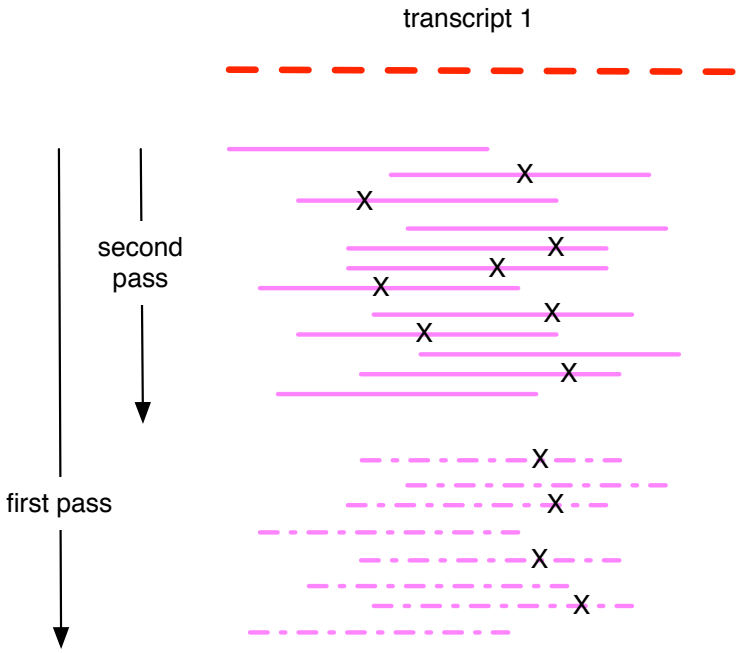

Figure 3: **Diagram of streaming error detection. In a first pass over the read data, reads are loaded in until the graph locus to which they belong is saturated. From that point on, reads are examined for errors and not loaded into the graph. In a second pass, only the subset of reads loaded into the graph are examined for errors.**

In Figure 4, we show diginorm-generated coverage saturation curves for both real and error-free simulated reads from *E. coli* MG1655. In both cases, we see that by about 1m reads, the majority of reads comes from loci with an estimated sequencing depth of 20 or higher, and hence can be used for error analysis on the first pass through the data. Only those reads collected on the first pass need to be examined again.

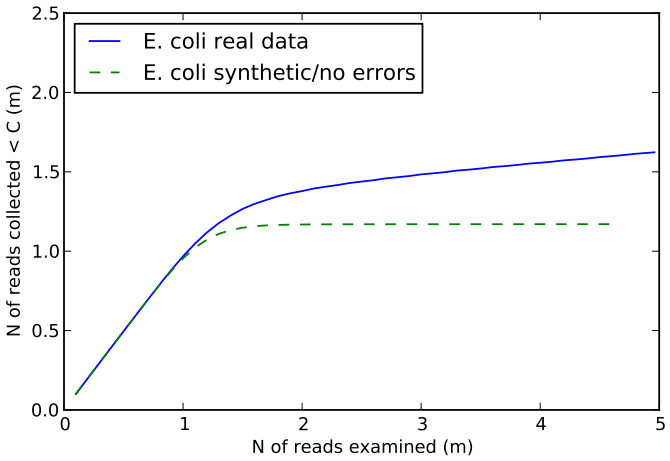

Figure 4: **Saturation curve of a real and a simulated *E. coli* read data set. Reads are collected when they have an estimated coverage of less than 20; in the early phase (< 1m reads), almost all reads are collected, but by 2m reads into the data set, the majority of reads come from loci with an estimated sequencing depth of > 20 and are rejected.**

We first apply this streaming error detection approach to the three synthetic data sets used earlier, and then to the three real data sets.

Streaming error analysis of synthetic data: When we apply this streaming approach to the simulated reads from the "simple genome", we obtain nearly identical numbers to the full two-pass approach: 485 TP, 365 TN, 150 FP, and 0 FN, for a sensitivity of 100% and a specificity of 76.3% (Table 8. However, only 320 of the 1000 reads are examined twice - hence, it is a 1.32x pass algorithm on this data set. Likewise, for the "simple metagenome" and "simple mRNAseq" data sets, we obtain nearly identical and identical results, respectively, with the streaming approach; due to differences in the order in which reads are examined, the simple metagenome fails to detect one true positive and erroneously finds errors in three extra reads. On the metagenome, 380 of 2347 (16.2%) of the reads are examined twice, and on the mRNAseq data set, 33.1% of the reads are examined twice.

	simple genome	simple metagenome	simple mRNAseq
Number of passes	1.32	1.16	1.33
Perfect detection (TP)	485	977 (-1)	228
No errors (TN)	365 (-1)	1095 (-3)	235
Miscalled errors (FP)	148	171 (+1)	52
Mispredicted errors (FP)	2 (+1)	9 (+3)	9
Missed errors (FN)	0	2	0
Sensitivity	100.0%	99.9%	100.0%
Specificity	76.4%	84.4%	78.9%

Table 8: Results from applying streaming error detection to the same synthetic data sets as in Table 2 and Table 6. Number of passes is the average number of times each read in the data set was examined; numbers in parentheses give the difference between these numbers and the previous results.

	E. coli	mock metagenome	mouse mRNAseq
Number of passes	xx	yy	zz
Perfect detection (TP)	1,033,261	11,726	91,133
No errors (TN)	2,781,961	364,627	140,461
Miscalled errors (FP)	1,138,940	3,951	20,742
Mispredicted errors (FP)	778	23,376	1721
Missed errors (FN)	5308	1216	6584
Sensitivity	99.5%	90.6%	93.3%
Specificity	47.6%	30.0%	80.2%

Table 9: Results from applying streaming error detection to the same real data sets as in Table 3 and Table 7. Number of passes is the average number of times each read in the data set was examined; unless noted in parentheses, numbers were within 1% of non-streaming results.

Streaming error analysis of real data: We also get similar quality results on the real data sets when comparing two-pass error detection with streaming error detection (Table 9). For *E. coli*, with streaming error detection we obtain a sensitivity of 99.5% and a specificity of 47.6%, compared to 99.8% and 47.9% with the two-pass approach (Table 3). For the mock metagenome, we get a sensitivity of 90.6% with streaming, vs 90.5% with the two-pass approach; and a specificity of 30.0% with streaming, vs 31.4% with the two pass approach (compare Table 9 and Table 7). And for the mRNAseq data set, we see a sensitivity of 93.3% with streaming vs 92.9% with two-pass, and a specificity of 80.2% vs 80.7% for streaming vs two-pass, respectively (Table 9 and Table 7). However, the streaming approach examined the *E. coli* reads only XX times, the metagenome reads YY times, and the mRNAseq reads ZZ times on average – a significant speedup for the high-coverage data sets.

Data set	pre-trim error	% bp trim	% reads trim	post-trim error
E. coli				
Mock metagenome	0.32%			0.30%
mouse mRNAseq	2.07%			1.84%
(high coverage only)	1.61%			0.66%

Table 10: A summary of trimming statistics for streaming error trimming. Error rates before and after trimming was estimated by mapping mismatches.

2.4 A streaming algorithm for error trimming

Once errors can be *detected* with a streaming algorithm, errors can also be *removed* reads by trimming the read at the first detected error. (While it is possible to split reads around errors rather than truncating them, this introduces complications in downstream read processing.)

Streaming error trimming on synthetic data: On the "simple genome" with counts from the digitally normalized reads, this trimming approach eliminates 149 reads entirely due to a starting low-abundance k-mer, and truncates another 392 reads. Of the 100,000 bp in the simulated reads, 31,910 (31.9%) were removed by the trimming process. In exchange, trimming eliminated *all* of the errors, bringing the overall error rate from 0.63% to 0.00%.

For the simple metagenome we used the variable abundance approach described above and only trimmed reads with estimated coverage of 20 or higher. Here, of 2347 reads containing 234,700 bp, 314 reads (13.4%) were removed and 851 reads (36.3%) were trimmed, discarding a total of 74,321 bases (31.7%). Of 1451 errors total, all but 61 were eliminated, bringing the overall per-base error rate from 0.62% to 0.04%. The simple mRNAseq data set showed similar improvement: 83 of 568 reads were removed, and 208 were trimmed, removing 19,507 of 56,800 bases (34.34%). The initial error rate was 0.65% and the final error rate was 0.07%.

Streaming error trimming on real data: Applying the streaming error trimming to the E. coli MG1655 data set used in section XXX, we trimmed 2.0m reads and removed 98,903 reads entirely. Of 8.0m errors, all but 203,345 were removed, bringing the error rate from 1.60% to 0.05%. Trimming discarded 61 Mbp of the original 500 Mbp (12.4%).

On the mouse mRNAseq data set, streaming error trimming removed 28,219 reads and trimmed 70,260 reads, removing 5.52% of the total bases, bringing the overall error rate from 2.1% to 1.8%. When we measured only the error rates in the high-coverage reads that were subject to trimming, trimming brought the error rate from 1.61% to 0.66%. On the mock metagenome data set, 5823 reads were removed and 21,231 reads were trimmed, removing 0.26% of bases; this low percentage is because of the very low coverage of most of the reads in this data set. XXX

2.5 Illumina error rates and error profiles can be determined from a small sample of sequencing data

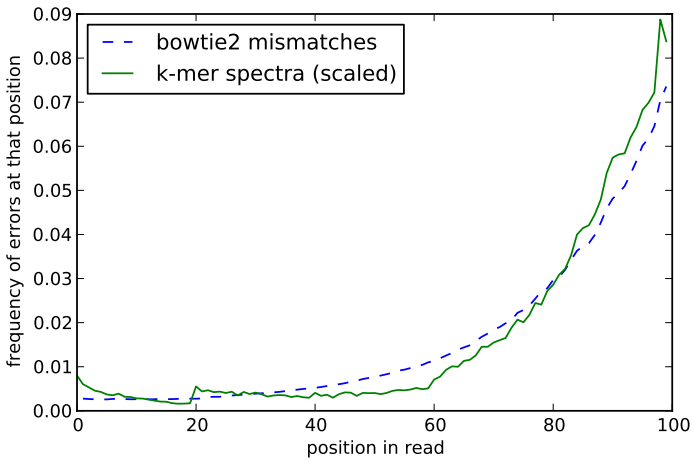

Figure 5: Error spectrum of reads in the *E. coli* data set. The k-mer spectrum based analysis is calculated based on saturation of a fraction of the data set, while bowtie2 mismatches are based on all mapped reads. The y values for the k-mer spectral analysis are scaled by a factor of four for ease of comparison.

With Illumina sequencing, average and per-position error rates may vary between sequencing runs, but are typically systematic within a run (cite?). Per-position error rates are caused by fluidics etc.

We can use the approaches described above to calculate systematic error profiles for shotgun sequencing data across entire data sets, but the variable abundance approach developed above can also be applied to *subsets* of Illumina data. The essential idea is to consume reads until sufficient data has been collected to calculate error rates, and then to calculate those error rates for the new reads based on the k-mer abundances from the old reads. This can aso be done in one pass for data sets with sufficiently high coverage data: as shown above (Figure XXX), more than half of the reads will have sufficient coverage to call errors by the time 10% of the data set has been consumed.

We first simulated a set of reads from the simple genome with errors only at even positions (0, 2, 4, etc.), and called errors in these reads using the single-pass algorithm described above (see Methods for implementation details). We also separately called errors by mapping with bowtie and examining mismatches. The results are shown in Figure YYY; the difference in predicted vs actual mismatch profiles has an average of 0.04%, with a variance of 0.0002% across positions. The predicted per-base error rate is 0.57%, while the true error rate is 0.65%; this underestimate may be due to ignoring reads with multiple errors

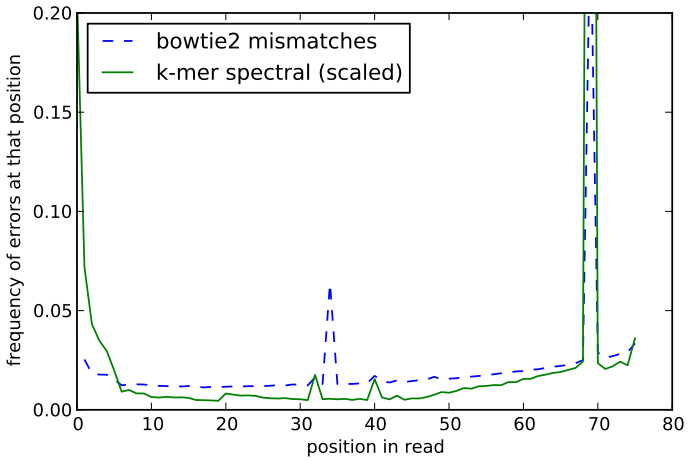

Figure 6: Error spectrum of reads in the mouse RNAseq data set. The k-mer spectrum based analysis is calculated based on saturation of a fraction of the data set, while bowtie2 mismatches are based on all mapped reads. The peak of errors at position 34 in the bowtie2 mapping reflects Ns that are ignored by the spectral error analysis; the peak at 69 is a mixture of Ns and random errors, so is detected by both mismatch analysis and spectral error analysis. See text for details. Note, the y values for the k-mer spectral analysis are scaled by a factor of four for ease of comparison.

in them that present as too low-coverage to assess.

@@show figure with shaded color indicating difference

We next applied to ecoli data.@@

@@error rates? apply regression?

3 Discussion

3.1 Digital normalization enables k-mer spectral error detection in variable abundance shotgun data

Reference-free error detection and correction in shotgun data from metagenomic, transcriptomic, and amplified samples typically requires specialized approaches (cite, discuss). Here we demonstrate that after digital normalization, k-mer abundance spectral approaches can be applied to reads with high estimated coverage. This should generalize to any approaches that rely on k-mer abundance, including error detection, trimming, correction.

One significant disadvantage of the variable abundance approach is that low-coverage reads cannot be trimmed because we do not know whether they are

from low-abundance molecules or are highly erroneous. This must be taken into account for downstream analysis; for example, while assemblers must already ignore or correct these erroneous reads, quantitation approaches using e.g. mapping should be not be applied to the trimmed data.

Another disadvantage of the variable abundance approach is that we cannot distinguish between low abundance and highly erroneous reads because the median k-mer abundance is an underestimate of coverage for highly erroneous reads. (In k-mer spectral error analysis of genomes, low abundance reads can simply be assumed to be highly erroneous.)

(@This is where applying Quake to diginorm reads would be a useful demonstration.)

3.2 A streaming algorithm for read analysis

We introduce a few-pass algorithm ($\leq 2N$) for detecting and analyzing reads that belong to high-coverage regions of the De Bruijn graph. As with digital normalization, the algorithm is sublinear in its memory use because only k-mer counts in the retained reads are tracked. The algorithm looks at every read once, and some reads twice; the exact number is dictated by biological features of the data set as well as the error rate, but for several real data sets is below 30% (Table 5).

This algorithm can potentially be applied to any shotgun sequencing data set, including genomic, transcriptomic, metagenomic, and whole-genome amplified data.

If better methods for read coverage estimation come along, they can be used in place of median k-mer abundance.

Potentially opens up streaming error correction and variant calling.

3.3 Empirical error profiles can be calculated on subsets of Illumina shotgun data

XXXX

Useful for cores etc: quick evaluation of sequencing quality, regardless of origin.

Repeats could be pulled off first.

3.4 Concluding thoughts

Here we have demonstrated that a general streaming algorithm based on detection of graph-local saturation can be used to detect and remove errors in high-coverage shotgun sequencing data. The algorithm is few-pass, in that some fraction of the data must be examined twice, although the precise fraction depends on biological features of the data set. In addition, because the algorithm relies on abundance-normalized sequencing data, it can be applied to metagenomic and transcriptomic data as well as genomic. And, finally, the approach may enable some sublinear-time analyses as well.

From a pragmatic perspective, the error trimming approaches described here should only be used on metagenomes and transcriptomes prior to assembly, and should not be used prior to mapping. Because they ignore low-abundance reads, those reads will not be trimmed, and may map poorly, in which case downstream analyses such as counting may be biased against low-abundance reads.

Time- and memory-efficient error profile calculation is practically useful for large sequencing cores.

Dealing with variants is outside the scope of this work, but 50/50 variants should not be trimmed. What about repeats?

Streaming methods for error correction and variant calling, more generally.

Theory needed for thresholding, but, in theory, thresholds should be static. (We can demonstrate that there is little variation.) Interesting to examine theory, real parameters for data sets... for the future.