1 Results

1.1 Coverage-normalized data can be used to locate errors in high-coverage shotgun sequencing data

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). While approaches derived from spectral analysis can be very accurate – Zhang et al. (2014) suggest that spectral analysis is considerably more effective at finding errors than quality-based approaches – it is also very compute intensive: most implementations count k-mers across entire sequencing data sets, which can be memory- or I/O-intensive for large data sets.

Brown et al. (2012) introduced a single-pass 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.)

To determine if digital normalization could be applied prior to k-mer spectral error detection, we first generated a synthetic data set from a simulated low-complexity genome ("simple genome"; see Methods for generation and Table XXX for data set details). We then applied digital normalization to a median 20-mer coverage of 20 (k=20, C=20) and used k-mer counts from the downsampled read set to detect errors by looking for bases at the beginning or ends of low-abundance regions 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).

Of the 531 reads in the simple genome with one or more errors, predicted errors matched exactly for 317 of them (true positives), and 466 reads were correctly predicted to contain no errors (true negatives). 90 reads were falsely predicted to have no errors (false negatives). The errors in 124 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 127 false positives. Using the above definitions, we calculated the prediction sensitivity to be 77.9the prediction specificity to be 71.4%.

@@put in picture of k-mer abundance spectrum before and after?

@list in a table: matched exactly, etc. rundown.

@next, E. coli?

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

One of the drawbacks of spectral abundance analysis is that it cannot be applied to metagenomic or transcriptomic data sets, which frequently contain reads from both high-abundance and low-abundance molecules. This variability in coverage confounds naive spectral analysis for two reasons: first, errors in very high abundance regions can accumulate and increase 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 and appear to be incorrect. In practice, therefore, error correction for metagenomic and transcriptome data uses more complex approaches than a single threshold.

If we apply diginorm to variable coverage data, we address the problem of accumulated errors from high-abundance regions by eliminating the majority of the reads from these regions; this is a form of error correction, detailed in the original diginorm paper (cite diginorm). However, we can also address the issue of low-abundance regions by using the median k-mer abundance to estimate which reads are too low-abundance to analyze; in effect, we can ignore reads that are outside of a desired coverage rage (Figure XXX).

@graph with read abundance spectrum ,showing which reads we will call errors in.

We generated two more synthetic data sets, "simple metagenome" and "simple mRNAseq," which contain both high- and low-abundance species (see Table XXX for data set details). After generating synthetic reads with a 1% error rate and applying digital normalization to k=20/C=20, we again applied our spectral error detection approach using the normalized counts, but with the modification that only reads with a median k-mer abundance of 20 or greater were examined. For the simple metagenome data set, 2254 of 2347 reads (96.0%) met the coverage criterion. Of the 2347 reads total, the errors in 658 erroneous reads were called perfectly (TP) and 1126 of the reads with no errors were correctly called as error-free (TN). 297 reads were incorrectly determined to be error free (FN; including the reads that were too low coverage to be considered). Of the remaining 266 reads, 261 were miscalled (errors existed but were not exactly called) and 5 were incorrectly called as erroneous when they were in fact correct. We calculated the prediction sensitivity to be 68.9% and the prediction specificity to be 71.2%. For the simple mRNAseq data set, 524 of 568 reads (92.3%) met the coverage criterion, with 153 true positives, 255 true negatives, 87 false negatives (including the low coverage reads not examined), and 73 false positives, for a prediction sensitivity of 63.8% and prediction specificity of 67.7%.

@ real data / measurements

1.3 A streaming algorithm can be used to detect errors based on read coverage

Even with digital normalization, 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 examines the reads. For large data sets, streaming algorithms can be considerably more efficient.

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 in 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.

If highly sampled regions could be detected in advance of assembly, twe could apply the same approaches used above to do error detection in a streaming fashion. Digital normalization provides such a method: by measuring coverage for each read against an online De Bruijn graph, reads that reach a specified coverage threshold can be examined for errors immediately. Those reads that had not yet reached high coverage could be set aside and re-examined later. This could result in significant runtime savings: for a genomic data set with 100x coverage, no more than 20% of the reads would need to be examined twice.

In Figure XXX, we show diginorm-generated coverage saturation curves for error-free simulated reads from synthetic and a real (*E. coli*) genome, as well as simulated reads with error from a synthetic genome, and real reads from the *E. coli* genome. In all cases, we see that more than BLAH % of reads reach an estimated coverage of 20 no more than ZAH % of the way through the data set.

@ discuss streaming error detection results.

1.4 A streaming algorithm for error trimming

We can now implement a streaming algorithm for error *trimming* of any shotgun data. This follows directly from streaming error detection: instead of identifying the locations of all errors, we truncate the read at the first (5') low abundance k-mer. (While it is possible to split reads around errors, this introduces complications in downstream read processing.)

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 truncated 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 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%.

@@@ real data

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

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 .04%, with a variance of .0002% across positions.

@@show figure with shaded color indicating difference

We next applied to ecoli data.@@