Error correction and assembly complexity of single molecule sequencing reads: How long is long enough?

Hayan Lee[1,2], James Gurtowski[1,3], Shinjae Yoo[4], Shoshana Marcus[1], W. Richard McCombie[5] and Michael C. Schatz[1]

[1]Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, [2] Department of Computer Science, Stony Brook University.
[3]Department of Computer Science, Columbia University. [4]Computational Science Center, Brookhaven National Laboratory. [5]Stanley Center for Cognitive Genomics, Cold Spring Harbor Laboratory.

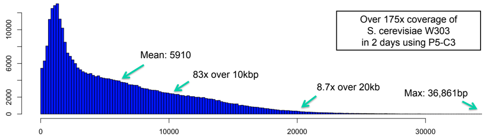
http://github.com/jgurtowski/ectools
http://qb.cshl.edu/asm_model/predict.html

Abstract

Third generation single molecule sequencing technology from Pacific Biosciences, Moleculo, Oxford Nanopore, and other companies are revolutionizing genomics by enabling the sequencing of long, individual molecules of DNA and RNA. One major advantage of these technologies over current short read sequencing is the ability to sequence much longer molecules, thousands or tens of thousands of nucleotides instead of mere hundreds. This capacity gives researchers substantially greater power to probe into microbial, plant, and animal genomes, but it remains unknown on how to best use these data. To answer this, we systematically evaluated the human genome and 25 other important genomes across the tree of life ranging in size from 1Mbp to 3Gbp in an attempt to answer how long the reads need to be and how much coverage is necessary to completely assemble their chromosomes with single molecule sequencing. We also present a novel error correction and assembly algorithm using a combination of PacBio and pre-assembled Illumina sequencing. This new algorithm greatly outperforms other published hybrid algorithms.

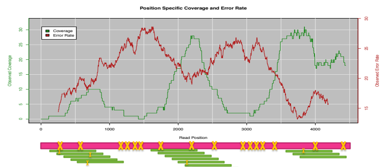
Background

Current long read technologies differ from short read technologies in that they sequence a single molecule, rather than a clonal population of DNA fragments. The current leading long read technology from Pacific Biosciences, the RS, records the incorporation of nucleotides by a stationary polymerase in real time. The resulting read lengths are thus not uniform; instead tending toward an exponential distribution as seen below.



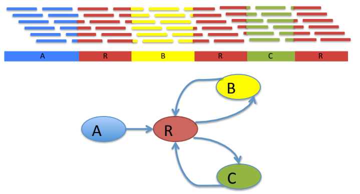
This is one of the few technologies that can produce reads greater than 20kb, but at the cost of reduced per-base accuracy. Currently, no assembler is designed to handle reads with an error rate as high as those produced by the RS making error correction very important for de novo assembly of this data.

Current hybrid error correction techniques align short high-identity reads to low-identity long reads and compute a consensus sequence. This approach is successful if the long-read error rate remains below a certain threshold. However, in certain circumstances, portions of reads may have more errors making the alignment of short reads to these regions difficult.



Above, a striking negative correlation appears between the increased per-base error rate and short read coverage. The alignment algorithm has difficulty placing reads in regions of high error rate. Regions that do not have short-read coverage will either remain uncorrected or cause the entire read to be split by the correction algorithm. When a read is split, vital information is lost and the ability to span repeats is severely diminished.

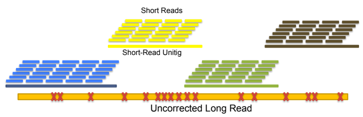
Below is an illustration of how repeats make assembly difficult. If repeats are longer than the read length, it is not possible to conclusively determine the ordering of unique contigs (unitigs).



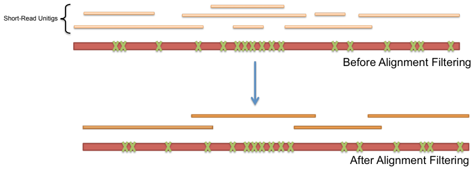
The above assembly graph is ambiguous because it is unclear whether R should be followed by B or C and how many times these contigs occur in succession.

ECTools: Enhanced Hybrid Error Correction

In all aspects of assembly long reads are preferred over short reads because they contain more contiguous information about the nucleotide sequence. Error correction is no exception. Rather than align raw short reads to a long-read backbone, the short reads are first preassembled into unitigs. These unitigs give more contiguous information and can span regions of high error rate. In order to align unitigs a very sensitive aligner is needed. Nucmer from the Mummer suite was chosen for its sensitivity and flexibility.



Above, unitigs composed of short reads are aligned to a raw long read. Because the short-read unitigs are the assembler's attempt at creating a unique representation of the genome, it is no longer appropriate to try to make a consensus sequence. Instead, we want to find the layout of the unitigs along the long read scaffold that maximizes long read coverage while minimizing short-read unitig overlap. This functionality is implemented in the delta-filter program which is also a part of the Mummer suite of tools. The functionality of delta-filter is depicted below where a large set of alignments are filtered down to a candidate set of unitigs that most likely come from the same region as the long read.



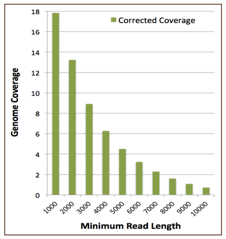
Using a variation of the longest increasing subsequence dynamic programming algorithm, delta-filter is able to filter unitig alignments to find the subset that are most likely to be from the same region as the long read. Once the set of unitigs is determined, the show-snps program is used to determine the difference between the unitig set and the raw long read. A simple python script uses this information to "correct" the long read, incorporating all of the information provided by the scaffold of unitigs.

Once the layout is determined and bases are corrected, and a conservative trimming algorithm removes regions of the long reads that did not have short-read unitig alignments. Depending on the context, the user may want to choose between shorter, well corrected reads and longer, lower identity reads. The user can specify a minimum identity and the splitting algorithm will make an effort to only output reads with a per-base identity greater than this number. It assumes all bases with a short-read unitig alignment can be corrected to within 99% identity while regions without coverage remain at the uncorrected 85% identity. Taking the average identity along the read and recursively splitting out sections without coverage eventually produces subreads with an identity that meets the user's requirements.

O. sativa pv Nipponbare

Genome size: 370 Mb
Chromosome N50: 29.7 Mbp
19x PacBio C2XL sequencing at CSHL from Summer 2012

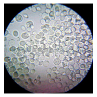
Assembly	Contig NG50
MiSeq Fragments 23x 459bp 8x 2x251bp @ 450	6,332
"ALLPATHS-recipe" 50x 2x100bp @ 180 36x 2x50bp @ 2100 51x 2x50bp @ 4800	18,248
PacBioToCA 19x @ 3500 ** MiSeq for correction	50,995
ECTools 19x @ 3500 ** MiSeq for correction	155,695

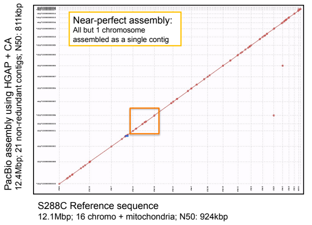


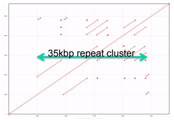
The table above shows the results of various rice assemblies. Assemblies using the long read data produced the best results. Our enhanced error correction pipeline was able to boost the n50 metric by nearly 3 fold over the correction with the existing PacBioToCA pipeline bundled with the Celera assembler.

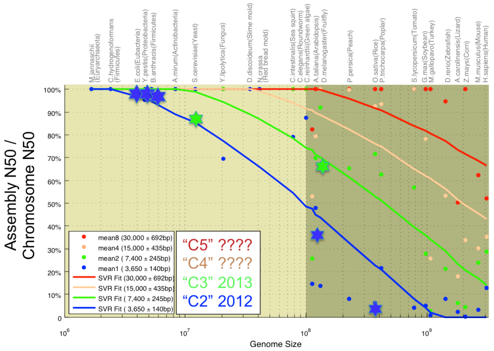
Assembly Modeling

We recently sequenced long size-selected templates from a BluePippin (Sage Sciences) procedure with a new sequencing chemistry from Pacific Biosciences to generate very long reads (greater than 80x coverage of reads exceeding 10kbp as shown on left). Using these data we assembled the Saccharomyces W303 genome using HGAP and the Celera Assembler. The resulting contig N50 length approaches 1 million bases, which for this genome represents chromosome length contigs for all but one chromosome containing a very long tandem repeat. The overall sequence accuracy was very high (>99.9%) as was the overall assembly performance (assembly N50 / chromosome N50) at 811kbp/924kbp = 88%.




Near-perfect assembly:
All but 1 chromosome
assembled as a single contig
PacBio assembly using HGAP + CA
12.4Mbp; 21 non-redundant contigs; N50: 811kbp
S288C Reference sequence
12.1Mbp; 16 chromo + mitochondria; N50: 924kbp





Through the analysis of real genome assemblies (stars), and dozens of simulated genome assemblies (dots) with different average read lengths, we determine assembly quality is primarily a function of repeat composition, read length, and coverage. We integrated these results with a new data driven model using support vector regression that accurately predicts assembly performance in these and novel genomes (solid lines) for today's and future technologies.

From this, we conclude most genomes up to 100Mbp should assemble into virtually complete chromosome arms using reads averaging at least 8kbp, as is currently available with PacBio sequencing. For larger genomes, the currently available read lengths dramatically improve the assembly, but will still need further improvement to achieve end-to-end chromosomes. The model of assembly performance is available as an interactive webapp that researchers can use to explore the tradeoffs in read length and coverage for genomes of any size.

Current Recommendations

< 100 Mbp:	HGAP/PacBio2CA @ 100x PB C3-P5 expect near perfect chromosome arms
< 1GB:	HGAP/PacBio2CA @ 100x PB C3-P5 expect high quality assembly: contig N50 over 1Mbp
> 1GB:	hybrid/gap filling expect contig N50 to be 100kbp – 1Mbp
> 5GB:	Email mschatz@cshl.edu

Caveats

Model only as good as the available references (esp. haploid sequences)
Technologies are quickly improving, exciting new scaffolding technologies