Large-scale sequencing and assembly of cereal genomes using Blacklight

Philip D. Blood*. Pittsburgh Supercomputing Center, Carnegie Mellon University. 300 S. Craig St. Pittsburgh, PA 15213. +1-412-268-9329. blood@psc.edu

Shoshana Marcus. Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, One Bungtown Road. Cold Spring Harbor NY 11743. +1-516-367-8393. smarcus@cshl.edu

Michael C. Schatz*. Simons Center for Quantitative Biology, Cold Spring Harbor Laboratory, One Bungtown Road. Cold Spring Harbor NY 11743. +1-516-367-5218. mschatz@cshl.edu

* Co-corresponding authors

Abstract

Wheat, corn, and rice provide 60 percent of the world's food intake every day, and just 15 plant species make up 90 percent of the world's food intake. As such there is tremendous agricultural and scientific interest to sequence and study plant genomes, especially to develop a reference sequence to direct plant breeding or to identify functional elements. DNA sequencing technologies can now generate sequence data for large genomes at low cost, however, it remains a substantial computational challenge to assemble the short sequencing reads into their complete genome sequences. Even one of the simpler ancestral species of wheat, *Aegilops tauschii*, has a genome size of 4.36 gigabasepairs (Gbp), nearly fifty percent larger than the human genome. Assembling a genome this size requires computational resources, especially RAM to store the large assembly graph, out of reach for most institutions. In this paper, we describe a collaborative effort between Cold Spring Harbor Laboratory and the Pittsburgh Supercomputing Center to assemble large, complex cereal genomes starting with *Ae. tauschii*, using the XSEDE shared memory supercomputer Blacklight. We expect these experiences using Blacklight to provide a case study and computational protocol for other genomics communities to leverage this or similar resources for assembly of other significant genomes of interest.

Categories and Subject Descriptors
J.3 [Life and Medical Sciences]: Biology and Genetics

General Terms
Algorithms, Performance, and Experimentation.

Keywords

Genome Assembly, DNA Sequencing, Plant Genomics, NGS, shared memory, high-performance computing, data-intensive computing

1. Introduction

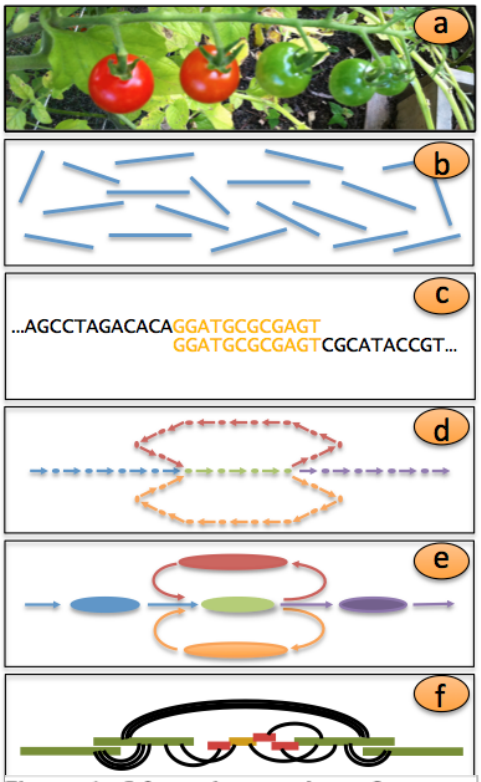

Figure 1. Schematic overview of genome assembly. (a). DNA is collected from the biological sample and sequenced. (b). The output from the sequencer consists of many billions of short, unordered DNA fragments from random positions in the genome. (c). The short fragments are compared to each other to discover how they overlap. (d) The overlap relationships are captured in a large assembly graph shown as nodes representing kmers or reads, with edges drawn between overlapping kmers or reads. (e). The assembly graph is refined to correct errors and simplify into the initial set of contigs, shown as large ovals connected by edges. (f) Finally, mates, markers and other long-range information are used to order and orient the initial contigs into large scaffolds, as shown as thin black lines connecting the initial contigs.

Recent dramatic improvements to DNA sequencing technology have greatly improved the costs and efficiencies to sequence complex DNA samples. For example, the human genome project was estimated to cost ~$3B and take years of effort to determine the 3 billion nucleotide genome in 2001 [1], whereas today, a single run of the Illumina HiSeq 2000 instrument can generate nearly 100 gigabasepairs (Gbp) of sequence for a few thousand dollars in a few days [2]. This makes it possible to rapidly sequence much larger genomes and many more genomes every year. For the first time it is now affordable and efficient to sequence the genomes of the cereal plants – wheat, corn, rice, barley, and others – whose genomes range from a few hundred megabases to many tens of gigabases. These projects will help unlock their genetic composition to improve our understanding of their growth and development from a molecular basis.

The sequences generated from DNA sequencing instruments are not end-to-end chromosomes or complete genomes. Instead, current instruments can only sequence a tiny fraction of the genome in a contiguous stretch, at most a few hundred or few thousand nucleotides at once. For example, the Illumina HiSeq 2000 mentioned above produces reads of at most 100 bp to 150 bp in length, although it can do so in great numbers and from random positions in the genome. Consequently it is possible to computationally reconstruct the complete genome sequence by oversampling the genome many fold with the short sequences, and then comparing how the billions of sequences relate to each other much like assembling a jigsaw puzzle (Figure 1, from [3]). This computation, called *de novo* genome assembly, is one of the most fundamental and important computations in biology as it lets us compute the genomes for species far beyond what we can directly sequence. In the case of the cereal genomes, a high quality de novo assembly would directly enable functional annotation of the genes and networks responsible for their growth and development, and enable directed plant

breeding and other important assays. These resources and these activities are extremely important as the cereals provide most of the calories consumed every day [4].

Unfortunately, cereal genomes, such as wheat, are extremely complex and difficult to assemble. The biggest difficulty comes from the high repeat content of these genomes making it difficult or impossible to correctly determine the genome sequence. Cereal genomes also have high ploidy (many copies of a chromosome), which further complicates efforts to identify the correct sequence. Finally, the sequencing instruments produce errors and have biases that must be corrected or the results will be highly fragmented or incorrect. Fortunately, advances in sequencing technology and assembly algorithms are beginning to address some of these challenges.

2. Algorithms and data for de novo assembly

De novo genome assembly has been researched for more than 35 years, starting with the first viral genomes assembled in the 70s and 80s [5], the first genome of a free living organism in 1995 [6], the human genome in 2001 [1], and dozens of microbes, plants, animals, and fungi since. The algorithmic process of *de novo* assembly begins by comparing the unassembled sequences to each other to form a large assembly graph of overlapping sequences [7]. Once the assembly graph is constructed, various network motif finding and path finding algorithms scan the graph to correct sequencing errors, identify repetitive sequences, and otherwise determine how the genome is organized.

The computational demands for large genomes are substantial. In the case of the cereal genomes, the assembly graph will consist of many billions of nodes and many tens of billions of edges, with no way to effectively partition the graph into separate connected components. Furthermore, the assembly problem has been demonstrated to be NP-complete under various theoretical formulations [8]. Given the enormous complexity involved, the leading assembly algorithms use an extensive set of local optimizations and heuristics to try to assemble the genome in a hierarchical fashion from local regions of high certainty and low complexity to distant regions with more complexity but less certainty.

There have been several major genome assembly competitions recently that evaluated more than a dozen different genome assemblers. These competitions have included the Assemblathon 1 [9] and Assemblathon 2 [10], and the Genome Assembly Gold Standard Evaluation (GAGE) [11]. In these studies, the genome assembler ALLPATHS-LG [12], developed at the Broad Institute, was consistently the best or among the best in all datasets and metrics used. The most widely used metric used for evaluating an assembly is the N50 size. This metric is a weighted average of the assembled sequence length, and is computed as the minimum length such that half of the assembly has been assembled into sequences this size or larger. All things being equal, a larger N50 size is better as it implies the genome has been assembled into larger, more contiguous sequences. Intuitively, studying fewer large sequences is easier and more complete than many small sequences. Other metrics measure the accuracy of the sequence by evaluating how well

the assembly represents the sequence data, how well the genes are represented, and others [10].

ALLPATHS-LG makes use of a combination of specialized experimental protocols and algorithms designed to address the challenges assembling complex genomes using short read data. A complete description of the algorithm has previously been described [12], but briefly, it begins by examining the unassembled reads to analyze and error correct the individual DNA sequences by counting occurrences of short words, called k-mers, in the data. It then constructs a large graph of how the k-mers in the unassembled sequences overlap each other. It then scans the assembly graph to further error correct and determine the genome sequence. This phase of the algorithm uses certain relationships between the sequencing reads called mate-pairs to construct larger assembled sequences called scaffolds. Unlike contigs, which are contiguous sequences of fully determined nucleotides, scaffolds may include unresolved sections ("gaps") in the sequence typically created by complex repeats flanked by well resolved sequences on either side.

Notably, unlike ABySS [13] or SOAPdenovo [14], ALLPATHS-LG requires minimal manual tuning or parameterization without requiring extensive pre-filtering of the data even with large gigabase sized genomes. The self-contained and self-tuning operations makes it much more robust than other programs, and often outperforms other algorithms substantially in terms of the quality of the assembly results. It is also a parallel algorithm using OpenMP, although it requires a large, shared memory server to store the large assembly graphs and sequence information in RAM. In general, the detailed error correction, graph construction, and scaffolding phases lead to biologically superior results. For these reasons, we chose ALLPATHS-LG for our initial assembly of the cereal genomes.

3. De novo assembly of *Ae. tauschii*

Our long term goal is to assemble all of the major cereal genomes. The first major goal was to de novo assemble the 4.36Gbp *Ae. tauschii* diploid wheat genome (wheat genome "DD") using ALLPATHS-LG and compare it to the recently published genome from BGI assembled using a much more complex protocol [15]. More than 100x coverage of the genome was sequenced at Cold Spring Harbor Laboratory using 50 to 100bp reads from three different libraries: a 180bp fragment library, a 2kbp jumping library, and a 5kbp jumping library. The sequencing data are publically available on the Cold Spring Harbor Website (http://www.cshl.edu/genome/wheat) and in the NCBI SRA (see CSHL website for accession numbers). In contrast, BGI used nearly 400x coverage of the genome in 45 libraries with reads ranging in size from 44 to 600bp and libraries sizes up to 20kbp.

Initially, we attempted to assemble the genome using *Jupiter*, one of the most powerful servers at Cold Spring Harbor Laboratory with 32 cores and 512GB RAM, but ALLPATHS-LG exhausted all available memory on the machine. We were successfully able to complete an assembly of the genome on *Jupiter* using a data reduction heuristic to

downsample highly repetitive reads using the algorithm *ReadFilterByKmerFreq* (an optional module of ALLPATHS-LG). However, this heuristic is unproven on genomes this large and risks disrupting the quality and contiguity of the assembly.

Fortunately, the XSEDE supercomputer Blacklight hosted at the Pittsburgh Supercomputing Center (PSC) had ample resources for the analysis, and together we set out to assemble these genomes on that machine.

3.1. Collaborative approach using XSEDE resources and services

Both hardware resources (Blacklight SGI Altix UV1000, 2x16 TB RAM, 4096 cores) and Extended Collaborative Support Services (ECSS) made available through XSEDE were essential to the initiation of this collaboration and its successful outcome. The Novel and Innovative Projects (NIP) branch of ECSS, which seeks out communities and applications that are not currently well-represented or supported on XSEDE, provided the flexibility to engage genomics researchers, understand their computational needs, and make them aware of resources such as Blacklight and ECSS to help them overcome their computational challenges. The initial community engagement included discussions with developers of key *de novo* assembly codes, including ALLPATHS-LG and Trinity [16] developed at the Broad Institute. As a result of these discussions, the availability of XSEDE resources, including Blacklight, was advertised on the web pages for these applications. This exposure, together with other personal outreach efforts, greatly increased awareness of these resources among genomics researchers and generated a large influx of demand to use Blacklight for de novo assembly [17].

To handle the high demand for de novo assembly, an XSEDE ECSS for Community Codes project was initiated for ALLPATHS-LG and Trinity. This provided the opportunity to engage further with the developers of these codes in order to port them to XSEDE systems, make any necessary modifications to enable them to run efficiently, and to assist scientists in using them effectively. The support provided by the Community Codes project was critical in overcoming various challenges to enable ALLPATHS-LG to assemble plant genomes of unprecedented size on Blacklight.

3.2. Computational challenges

The PSC's Blacklight supercomputer is a SGI Altix UV 1000 with two 2048 core partitions, each containing 16 TB of cache-coherent shared memory. The challenges involved in getting ALLPATHS-LG running well on this supercomputer fell into three general categories:

1. **Port ALLPATHS-LG to Blacklight:** In general, codes developed on a regular

workstation will run without modification on Blacklight. To the application, each Blacklight partition looks like a huge workstation running a single instance of the Linux operating system. The most common problem is that shared memory threaded codes designed for a 24-48 core workstation will by default try launch as many threads as available cores on the system. A single Blacklight partition has 2048 physical cores, and 4096 logical cores, so many codes using shared memory parallelism, including ALLPATHS-LG, will routinely attempt to launch 4096 threads when they have only been allocated a small subset of the entire partition (e.g. 128 cores). We tracked down and addressed this issue by adding appropriate defaults in various places in the ALLPATHS-LG code. Even after fixing the issue and submitting changes to be incorporated back into the main code branch, additional development sometimes re-introduced the same problem. Gradually this was addressed through continued close work with the development team.

2. **Get ALLPATHS-LG to perform well on Blacklight:** We obtained several standard benchmarks used by the ALLPATHS-LG developers and used these to evaluate ALLPATHS-LG performance on Blacklight. ALLPATHS-LG consists of several dozen different modules that execute separately, one after the other. The modules communicate by writing out (sometimes very large) files for the subsequent modules to operate on. We found that many of these modules perform many small reads and writes during their execution. On one benchmark the average read size was 8 KB and the average write size was 4 KB. While many of the major computational regions are parallelized using threads, some sections of the code execute in serial. These serial regions, together with the I/O limited regions, limit scaling of ALLPATHS-LG to ~16-32 threads on benchmarks. For bigger genomes, we found that ALLPATHS-LG suffered significant slow downs when running on Blacklight's parallel lustre file system. We began to explore improving I/O efficiency with the developers, but since other genomics codes had similar problems, we ultimately decided to attach a non-lustre local filesystem to Blacklight to handle the small reads and writes. This eliminated the slow downs and brought performance back to the level of standard large memory nodes.

3. **Enable ALLPATHS-LG to assemble genomes of unprecedented size:** To our knowledge, no one has successfully assembled with ALLPATHS-LG a data set as large as the one we have generated for the *Ae. tauschii* assembly. To prepare, we successfully reassembled the 450Mbp *Oryza sativa* ssp. japonica cv. Nipponbare rice genome [18] on Blacklight and tuned runtime parameters to obtain comparable performance to the CSHL server *Jupiter* with 512 GB of RAM. Based on this assembly, we estimated that we'd need ~1.5 TB of RAM for the assembly, so we started the assembly with 2 TB of RAM over 256 cores (utilizing 32 threads due to scaling limitations mentioned previously). We quickly ran out of RAM, and so increased the job to 3 TB of RAM over 384 cores. This provided sufficient memory, but during the second instance of the *RemoveDodgyReads* module, which operates on the jumping library reads, we noticed that although ALLPATHS-LG

was keeping the processors busy, it was not making any progress. We attached to the active process using gdb and found that ALLPATHS-LG was stuck in an infinite loop due to an integer overflow in a 32-bit loop counter which was trying to iterate over the 11 billion reads in our jumping library. This was fixed and contributed back to the main code repository, and the assembly was able to run to completion. We expect to expose additional limitations as we attempt to assemble even larger genomes.

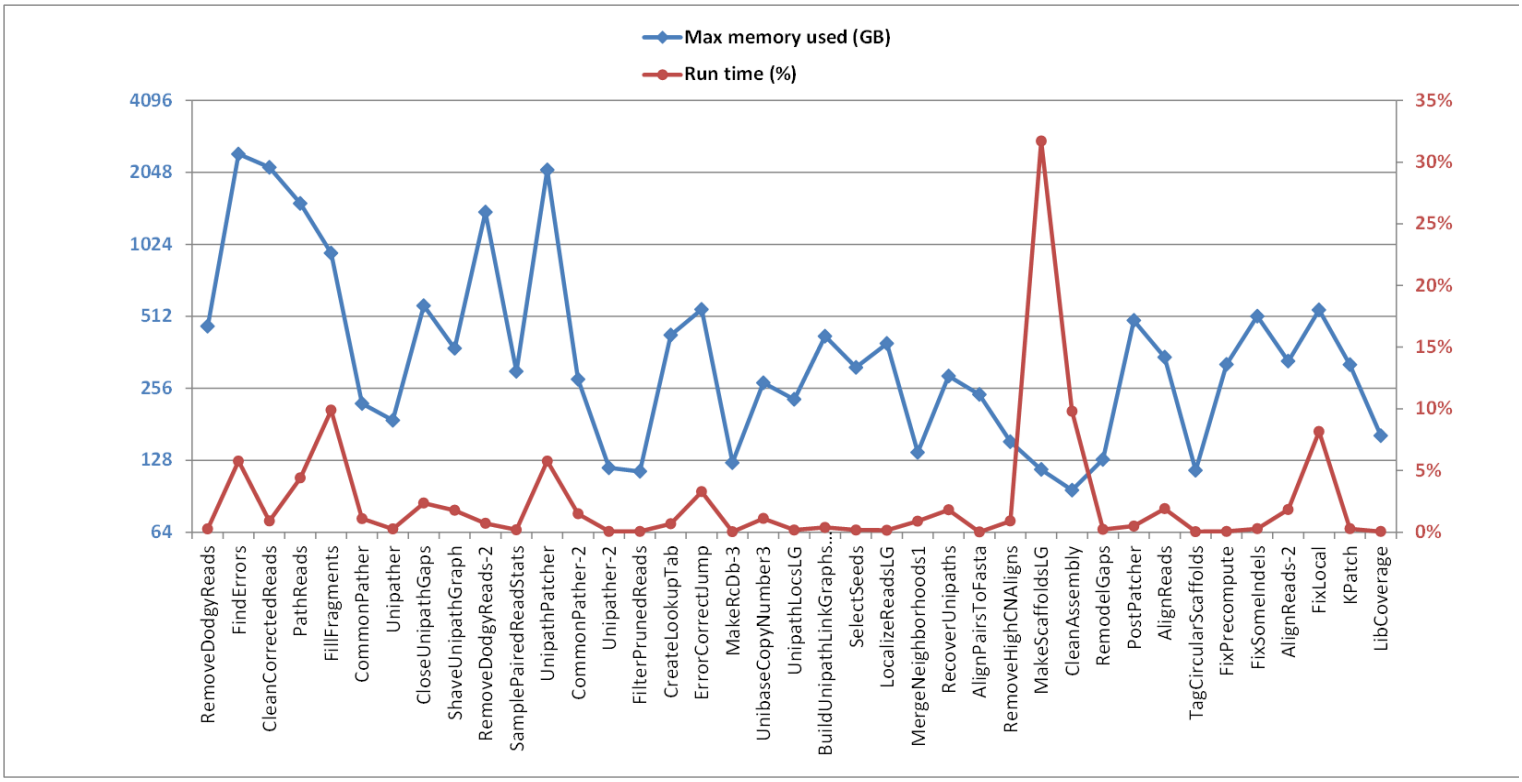

Figure 2. ALLPATHS-LG module memory usage and execution time as a percentage of the total execution time during the assembly of *A. tauschii*. Modules are listed in the order of execution, from left to right. Only modules that required greater than 100 GB of memory and/or accounted for at least 1% of total execution time are shown.

Figure 2 shows the major ALLPATHS-LG modules in their order of execution, the maximum memory they used, and the percentage of total run time. The *A. tauschii* assembly ran for a total wall time of 1146 hours on Blacklight, using a maximum of 2.44 TB of RAM during *FindErrors*. The initial, very high memory regions were run on 384 cores with 3 TB RAM, and this was then dropped to 256 cores and 2 TB RAM for the later stages. The data in Figure 2 can provide a guide to other groups planning massive genome assemblies to determine what stages will require the most RAM and overall

resources. They also provide a target for future optimizations of ALLPATHS-LG to better handle these massive genomes.

3.3 Assembly results

The overall assembly statistics are presented in Table 1, with both the results from the downsampled, repeat filtered assembly computed at CSHL along with the assembly computed on Blacklight with all the reads.

The most striking result is while the contig or scaffold N50 values are roughly equivalent between the downsampled and full dataset assemblies, the assembly computed on Blacklight with all of the reads has an additional 128Mbp of sequence in contigs and an additional 645 Mbp of sequence in scaffolds. In this regard, the assembly computed on Blacklight is far superior to the one computed using the downsampling algorithm with a 16% to 48% gain to contig and scaffold sequence lengths, respectively. For all future analysis of gene content, regulatory features, or any other genomic analysis the Blacklight assembly will be used.

Interestingly, the connectivity of the Blacklight assembly approaches that achieved by BGI (4.5kbp contig N50, 58kbp scaffold N50) at fraction of the cost and complexity. However, the BGI approach did capture a much greater fraction of the genome (~80%), presumably because they could resolve more repetitive elements with the larger jumping libraries that remain missing in our assembly. This suggests that overall, great biological insight can be gained from the relatively simple 3 library approach when used with ALLPATHS-LG, although adding additional larger libraries can be useful for resolving the larger repeats. A more complete evaluation of the two assemblies is underway.

Table 1. Server environments and assembly statistics. Coverage refers to the fraction of the estimated genome size (4.36 Gbp) represented by the sequence length. Note N50 has been computed with respect to each assemblies' total span.

	CSHL/Jupiter ***(Downsampled Reads)***	**PSC/Blacklight** ***(All Available Reads)***
Total Cores	32 cores	2048 cores
Physical RAM	512 GB	16TB
Peak RAM Usage	395 GB	2.44TB
Total Elapsed Time	337 hours	1146 hours
Input reads	2.84 B	11.09 B

Total Contig Length	760.9 Mbp	888.0 Mbp
Contig Coverage	17%	20%
Contig N50	6.2 kbp	5.4 kbp
Total Scaffold Length	1.34 Gbp	1.98 Gbp
Scaffold Genome Coverage	30%	45%
Scaffold N50	23 kbp	21 kbp

4. Conclusion and future outlook

The genomes of the cereal plants hold the promise to dramatically advance both our understanding of basic plant science, and of catalyzing practical advances in plant breeding and food production. The recent advances to DNA sequencing make it relatively affordable to sequence these genomes to deep coverage, if only we can apply the proper algorithms and computational systems to assemble the raw sequences into complete genomes. In particular, the widely used Illumina sequencing platform can only produce reads at most a few hundred base pairs, thus requiring a massive computation to compare the reads to each other to compute the overall genome sequences.

In this project we have demonstrated it is possible to execute one of the leading genome assembly algorithms, ALLPATHS-LG, to assemble a 4.36 Gbp plant genome. To the best of our knowledge, this is the largest successful assembly with ALLPATHS-LG ever completed, and we have demonstrated the assembly results approach the quality of the published genome computed with a much more complex experimental protocol. To reach these goals, we have leveraged the tremendous capabilities of Blacklight to be the enabling technology for the analysis. Without Blacklight, we were technically unable to assemble the complete dataset and would have been limited to analyzing the results from the repeat filtering, downsampling heuristic that discards hundreds of megabases of useful sequence. We expect now that this result has been established, we and many other groups will look to Blacklight and other large memory XSEDE resources for their large genome assembly needs. All of the modifications we made to the ALLPATHS-LG source code for this assembly have been incorporated back into the main code base [19]. In addition, a tutorial on performing extreme scale assembly with ALLPATHS-LG on Blacklight will be maintained here [20], which will provide an opportunity for any research group at a U.S. academic institution, or their international collaborators, to perform similar assemblies by obtaining an XSEDE allocation [www.xsede.org].

Despite these impressive advances, there is only so much that can be done with short read data to assemble large, complex, repetitive genomes. Both our assembly and the published BGI assembly of *Ae. tauschii* have relatively short contigs and scaffolds and a

significant amount of missing sequence. Recently, however, improvements in single molecule real time (SMRT) sequencing technology from PacBio have produced much longer reads, with average read lengths approaching 10kbp and maximum read lengths exceeded 50 kbp. [21]. We are optimistic that we will be able to effectively use these data to create even better assemblies of cereals and other large plant genomes, and again will be looking to resources like Blacklight, along with the services available through XSEDE to meet the computational demands.

Acknowledgements

The project was supported in part by National Institutes of Health award R01-HG006677 and National Science Foundation award DBI-126383 to MCS. The Blacklight system time was supported by XSEDE under project TG-MCB120113 to MCS. The sequencing data were supported by NSF award IOS-1032105 to WRM. We would like to thank David Jaffe, Iain MacCallum, Ted Sharpe, Filipe Joao Ribeiro, and all the ALLPATHS-LG developers and support staff for the assistance debugging and troubleshooting the assembly.

References

1. The International Human Genome Sequencing Consortium, *Initial sequencing and analysis of the human genome.* Nature, 2001. **409**(6822): p. 860-921.
2. Illumina. *HiSeq 2500 Sequencing System Specifications*. Available from: http://www.illumina.com/Documents/%5Cproducts%5Cappnotes%5Cappnote_hiseq2500.pdf.
3. Schatz, M., J. Witkowski, and W.R. McCombie, *Current challenges in de novo plant genome sequencing and assembly.* Genome Biology, 2012. **13**(4): p. 243.
4. *Dimensions of Need - Staple foods: What do people eat*. United Nations Food and Agriculture Organization: Agriculture and Consumer Protection; Available from: http://www.fao.org/docrep/u8480e/u8480e07.htm.
5. Sanger, F., A.R. Coulson, G.F. Hong, D.F. Hill, and G.B. Petersen, *Nucleotide sequence of bacteriophage lambda DNA.* J Mol Biol, 1982. **162**(4): p. 729-73.
6. Fleischmann, R.D., M.D. Adams, O. White, R.A. Clayton, E.F. Kirkness, et al., *Whole-genome random sequencing and assembly of Haemophilus influenzae Rd.* Science, 1995. **269**(5223): p. 496-512.
7. Schatz, M.C., A.L. Delcher, and S.L. Salzberg, *Assembly of large genomes using second-generation sequencing.* Genome Res, 2010. **20**(9): p. 1165-73.
8. Nagarajan, N. and M. Pop, *Parametric complexity of sequence assembly: theory and applications to next generation sequencing.* Journal of computational biology : a journal of computational molecular cell biology, 2009. **16**(7): p. 897-908.
9. Earl, D., K. Bradnam, J. St John, A. Darling, D. Lin, et al., *Assemblathon 1: a competitive assessment of de novo short read assembly methods.* Genome research, 2011. **21**(12): p. 2224-41.

10. Bradnam, K.R., et al., *Assemblathon 2: evaluating de novo methods of genome assembly in three vertebrate species.* Gigascience, 2013. **2**(1): p. 10.
11. Salzberg, S.L., A.M. Phillippy, A. Zimin, D. Puiu, T. Magoc, et al., *GAGE: A critical evaluation of genome assemblies and assembly algorithms.* Genome research, 2012. **22**(3): p. 557-67.
12. Gnerre, S., I. Maccallum, D. Przybylski, F.J. Ribeiro, J.N. Burton, et al., *High-quality draft assemblies of mammalian genomes from massively parallel sequence data.* Proceedings of the National Academy of Sciences of the United States of America, 2011. **108**(4): p. 1513-8.
13. Simpson, J.T., K. Wong, S.D. Jackman, J.E. Schein, S.J. Jones, et al., *ABySS: A parallel assembler for short read sequence data.* Genome Res, 2009.
14. Li, R., H. Zhu, J. Ruan, W. Qian, X. Fang, et al., *De novo assembly of human genomes with massively parallel short read sequencing.* Genome Res, 2009.
15. Jia, J., et al., *Aegilops tauschii draft genome sequence reveals a gene repertoire for wheat adaptation.* Nature, 2013. **496**(7443): p. 91-5.
16. Grabherr, M.G., et al., *Full-length transcriptome assembly from RNA-Seq data without a reference genome.* Nat Biotechnol, 2011. **29**(7): p. 644-52.
17. Brian Couger, et al., *Enabling large-scale next-generation sequence assembly with Blacklight.* Concurrency and Computation: Practice and Experience, 2014. doi: 10.1002/cpe.3231.
18. Goff, S.A., D. Ricke, T.H. Lan, G. Presting, R. Wang, et al., *A draft sequence of the rice genome (Oryza sativa L. ssp. japonica).* Science, 2002. **296**(5565): p. 92-100.
19. ftp://ftp.broadinstitute.org/pub/crd/ALLPATHS/Release-LG/
20. https://www.psc.edu/index.php/allpaths-lg
21. http://goo.gl/w7qNJQ