State of the art *de novo* assembly of human genomes from massively parallel sequencing data

*Yingrui Li,[1] Yujie Hu,[1,2] Lars Bolund[1,3] and Jun Wang[1,2]**

[1]BGI-Shenzhen, Shenzhen, Guangdong 518083, China
[2]The Graduate University of the Chinese Academy of Sciences, Beijing 100062, China
[3]Department of Biology, University of Copenhagen, Copenhagen DK-2200, Denmark; Danish Center for Translational Breast Cancer Research, Copenhagen, Denmark; Institute of Human Genetics, University of Aarhus, Denmark
**Correspondence to:* E-mail: wangj@genomics.org.cn

Date received (in revised form): 17th March 2010

Abstract

Recent studies in human genomes have demonstrated the use of *de novo* assemblies to identify genetic variations that are difficult for mapping-based approaches. Construction of multiple human genome assemblies is enabled by massively parallel sequencing, but a conventional bioinformatics solution is costly and slow, creating bottlenecks in the process. This review describes two public short-read *de novo* assembly applications that can handle human genomes, ABySS and SOAPdenovo. It also discusses the technical aspects and future challenges of human genome *de novo* assembly by short reads.

Keywords: *de novo assembly, de Bruijn graph, massively parallel sequencing*

Introduction

One of the important goals of bioinformatics is to decipher the genome DNA sequence of a species. The genome serves as the digital basis of any life science. Access to a reference genome sequence for a species significantly facilitates biological studies, as proven by all the genomics-guided research in the wake of the Human Genome Project.[1] It is conventionally believed that when a reference genome is available, any following studies will take a *mapping*-based 're-sequencing' approach aiming for variation detection, as seen in many projects of human genomics.[2,3] Recent studies, however, suggest that *assembly*-based approaches have greater potential to detect a more complete set of genetic variations, especially novel sequences[4] and structural variations,[5] even in relatively well-studied human genomes. Thus, assembly of individual genomes has again been brought to the frontier of bioinformatics. With multiple assembled individual genomes available, it would be very interesting to see how rearrangements of different length scales and individual-specific sequences are distributed in the populations.

The size of the human genome constrained individual human assembly by conventional Sanger sequencing because of costs. Second-generation sequencing technology produces large amounts of data more affordably, but the intrinsic high-throughput and short-read-length present considerable challenges to bioinformatics because of the difficulties in handling the data structure and in applying an appropriate assembly algorithm. Although many short-read *de novo* assemblers have been developed,[6] only two of them, ABySS[7] and SOAPdenovo,[8] are said to be capable of assembling human genomes *de novo*. This paper presents a review of the two software packages and discusses the technical aspects of human genome short-read *de novo* assembly.

Data structures

To be able to use short-read-length data to meet the general minimum requirement of overlap between reads in order accurately to assemble a human genome (26 base pairs [bp] in ABySS, 24bp in SOAPdenovo), the genome must be sequenced in depth owing to the significantly higher overlap-to-read-length ratio, as derived from the Lander–Waterman rule.[9] Conventional Sanger sequencing (the read length of which is generally hundreds of bp) allows assembly of a human genome from whole-genome shot-gun data at a depth of approximately sevenfold.[5] By contrast, both ABySS and SOAPdenovo require a total genome sequencing coverage of more than 40-fold to assemble a human genome from Illumina GA reads. Considering the short-read length, the number of reads that have to be processed is increased by at least one order of magnitude. Hence, with the highly redundant short-read sequencing data, it is difficult to compare potential overlaps between reads (even if heuristic methods were to be applied to reduce the computing complexity). As a succession to pioneering works on graph-based assembly algorithms,[10] de Bruijn graphs were introduced to compact potential overlaps by shared *k*-mers (vertex in the graph) among reads, to reduce the intense read comparison computation. Most of the short-read assemblers explicitly or implicitly take advantage of this idea.[11–13]

Assembly of a 3-gigabase (Gb) human genome, however, would still require large memory usage to store such a de Bruijn graph and companion information that facilitates operations (searching, etc) upon the graph or assembly condition constraints (paired-end information, read location, sequencing quality, etc). Both ABySS and SOAPdenovo keep only the key graph and parsimonious companion data in the memory. Other information that has been used in some other assemblers for the purpose of accuracy has been either discarded because of the relatively small computation cost–effectiveness ratio or moved to later steps. This is one of the major trade-offs adopted by the two assemblers that makes human genome assembly feasible with short reads, in terms of space. SOAPdenovo was designed to run on a single large-memory supercomputer, in which about 150 Gb memory is needed to assemble a human genome. This is still a large memory requirement, but it is fairly feasible considering the computing resources that current large-scale human genomics research can access. ABySS, however, developed a novel distributed de Bruijn graph that could be run on Linux clusters with message passing interface (MPI) library. This is a core strength and essential innovation of ABySS, making it more accessible to a potentially wider range of users. In addition, ABySS is formally able to handle colour–space sequencing data in its updated version, whereas SOAPdenovo does not.

Contiging

Building contigs from a de Bruijn graph is to work out the path through vertex *k*-mers that represent the correct biological sequence. When creating a de Bruijn graph, sub-sequences of *k*-bp in length (*k*-mers) are sequentially picked up from reads base by base to build vertices, and *k*-mers overlapped by $k-1$ bp were considered to have an edge in between. SOAPdenovo requires the edge to be supported by an original read, but ABySS does not. Each read serves as a continuous link thread through all vertex *k*-mers that are derived from it and a coverage support of respective edges.

The most challenging part in contiging is the algorithm to clean up the whole graph. Existing sequencing errors create false vertices and edges, which lead to an extremely complicated and much larger graph. Single nucleotide polymorphisms (SNPs) also diverge a continuous edge into two paths. Elimination of errors would in general reduce ambiguity when figuring out the correct path, which has a positive effect on the contig length and accuracy. Over-stringent error removal would also break authentic edges and corrupt the graph into pieces, however, which has a negative effect on contiging. An elegant design would make a great difference in contig length and accuracy. In ABySS, read errors are processed after construction

of the de Bruijn graph, which includes removing blunt ends and merging 'bubbles'. SOAPdenovo makes similar efforts with a different implementation. First, SOAPdenovo has a pre-assembly error correction step, which revises *k*-mers at low depth, significantly reducing erroneous reads and *k*-mers and memory usage. This step, in effect, helps to clean up the de Bruijn graph before its construction, and at a cost of only ~0.3 per cent reads with incorrect revision. The wrong revisions are claimed not to cause misassembly due to paired-end checking. Secondly, ABySS iteratively chops the 'dead ends' in the graph up to a certain threshold, while SOAPdenovo also removes low-coverage links and resolves tiny repeats spanned by reads in addition to 'dead ends'. An updated version of ABySS also expands tandem repeats, if the length can be defined. Thirdly, in the merging step, ABySS merges bubbles that are within 2-kilobase pairs (kbp), while SOAPdenovo compares the two potential paths and merges them based on a similarity threshold.

The importance of data quality in human genome assembly contiging by de Bruijn graphs should be emphasised. Unsuccessful runs of sequencing would generate data of relatively low quality. Although mapping-based human genome analyses are relatively insensitive to data quality, as error-rich reads are eliminated in alignment, graph-based assembly relies much more on the quality. Accurate sequencing significantly increases the number of error-free reads, greatly facilitating graph construction. Approaches that tackle errors, as described above, are able to work on data with small numbers of errors. Otherwise, an excess of spurious dead ends and links would result in a complex network-like graph, in which distinguishable patterns of errors, such as unique dead end and low-coverage edges, are difficult to identify. In some cases, a high-quality filtered subset of raw data could give better assembly results (longer and more accurate contigs) than the total raw read set.

After cleaning up the original graph, vertices that are unambiguously linked are merged into contigs. In the meantime, ABySS removes ambiguous edges to a log file for potential recovery, while SOAPdenovo breaks all boundaries of divergent points to keep multi-copy repeats as single contigs for further analysis.

Regarding the performance in the assembly of the NA18507 genome with the same read set, the initial contig size (excluding those smaller than 100 bp) of ABySS (860 bp) and SOAPdenovo (886 bp) is pretty similar. The sum length achieved by the two assemblers is also nearly identical (2.10 Gb).

Scaffolding

To reduce the memory use in the de Bruijn graph step, neither ABySS nor SOAPdenovo take advantage of paired-end information in the initial contig construction. This makes the scaffolding process an important step, with the following functions: 1) Revise initial contigs, if the contig sequence conflicts with paired-end mapping results; 2) Link contigs by paired-end information to form scaffolds; 3) Merge contigs by paired-end information and overlap to form longer contigs or 'scaftigs'. A scaftig refers to a continuous sequence formed by multiple initial contigs lined up in a scaffold with putative sequence overlaps. Scaftig-forming initial contigs that were not successfully linked as one in the original de Bruijn graph could be explained by inadequate overlap compared with the minimum overlap requirement ($[k-1]$ bp) and repetitive sequences that are either removed or cut out. With paired-end link support and the estimated overlap or distance between two initial contigs, it is possible confidently to join them together with a smaller ($<(k-1)$ bp) overlap, a recovered repeat sequence or a local assembly.

Although ABySS does not explicitly claim a scaffolding step, it actually has a second phase that utilises paired-end information to form scaftigs. The reads are first aligned to the initial contigs to create a linked contig set with mispaired/misaligned links filtered. With a minimal link number cut-off (five by default in ABySS) from each contig, the program searches through its linked contigs for single unique paths and consistent paths are used for stitching potential scaftigs. A branch-bound method is applied to constrain exhaustive searches in complex repeat

structures, to reduce unnecessary computational intensity. In the first release of ABySS, the resulting contigs had an N50 increase from 860 bp in initial contigs to 1,499 bp, which summed up to 2.18 Gb and covered 68 per cent of human genomes. In the updated version of ABySS, there is an added option '-scaffold' (by default disabled) that could fill Ns (stretches of unknown nucleotides with estimated lengths) in two linked, but not overlapping, contigs. However, the performance of this scaffold assembly has not been reported.

SOAPdenovo explicitly has a scaffolding step, in which it also aligns reads back to initial contigs, creates a contig linkage graph and builds scaffolds. The process is similar to that of ABySS, with some different assembly parameter settings, except for the processing of repeats. SOAPdenovo defines contigs with multiple entries and multiple exits in the linkage graph as repeats and masks them in scaffolding. This means that SOAPdenovo only uses unique sequences on the genome for scaffolding. One important step of SOAPdenovo is the internal gap closure that closes the internal gaps of scaffolds to build scaftigs. SOAPdenovo takes advantage of paired-end information to retrieve reads near a gap and performs overlap-based local assembly, trying to extend the contigs into the gaps and hopefully to solve them. This could, in principle, solve gaps caused by inadequate overlap and repeats, as long as the initial scaffolding is accurate and the paired-end insert size can cover them. This significantly improves the continuity of assembly, with N50 contig size increased from 886 bp to 4,611 bp (without paired-end reads from long-insert libraries), which covers 85 per cent of human genomes. Note that in ABySS, the genome coverage could be 90 per cent without the 100 bp minimum length threshold to claim a valid contig. This suggests that the differences are mainly attributable to contigs smaller than 100 bp.

It should be emphasised that in short-read assembly, connection of initial contigs to form scaftigs is more important than it is in conventional Sanger assembly. Theoretically, a repeat could be solved by either long reads or paired-end reads that span over the repeat. This is the underlying principle that allows short reads to be assembled into large genomes as long as insert sizes of paired-end libraries fit the genome repeat characteristics. A precise and sufficient use of paired-end reads then becomes the critical technical challenge in short-read assembly, however. How well the small gaps are filled could make a large difference to the results.

At this stage, SOAPdenovo outperforms ABySS in both length and genome coverage. As the two programs have pretty similar results in contiging, a major reason for the difference could be the more intensive data utilisation of SOAPdenovo in scaffolding and gap closure. In addition, considering that ABySS could achieve 90 per cent genome coverage by adding small contigs (length <100 bp), pre-assembly read error correction in SOAPdenovo may also play an important role in the quality of contigs and paired-end mapping reliability to make the final outcome less fragile.

The two genome assemblies presented in the SOAPdenovo paper[8] also show striking differences in scaffolding results, which are mainly explained by the multiple paired-end libraries with step-wise increasing insert sizes. Larger-insert paired-end reads would organise scaffolds of considerable sizes as long as shorter-insert ones solved the interleaving problem. This suggests that an appropriate strategy should be chosen to sequence and assemble a human genome.

Performances

The computing and overall performances of the two programs are summarised in Table 1. Both tools have an acceptable speed if installed on an appropriate architecture. With respect to final outcomes, SOAPdenovo seems to have a significant advantage, as described above. Since its first release, however, ABySS has had tens of updates, of which many have been essential and supposed to improve the performance significantly. Several genomes assembled by SOAPdenovo and a transcriptome assembled by ABySS have been published.

A critical problem for the human genome short-read assemblers is accuracy, yet neither ABySS nor SOAPdenovo has undergone a direct accuracy

Table 1. Comparison of ABySS and SOAPdenovo programs

Program	Computing features			
	Language	Peak memory on single node	Time used	Availability
ABySS	C++	<16 Gb	87 h in 8 × 21 CPU cores (much reduced in current version)	Open source
SOAPdenovo	C	140 Gb	40 h in 32 core	Free binary

	Algorithm and data structure					
	Pre-assembly error correction	de Bruijn graph	de Bruijn graph clean-up	Contiging	Scaffolding	Post-scaffolding process
ABySS	N/A	Distributed; can handle colour-space reads	Blunt-end, bubbles and tiny repeats	Removes and reports ambiguous edges	Implicitly, branch-bound search to tackle repeats	N/A
SOAPdenovo	Integrated	Single; nucleotide reads only	Blunt-end, bubbles, low coverage links and tiny repeats	Cuts ambiguous edges at boundaries	Explicitly, mask repeats	Gap closure by local assembly

	Genome assembly paramters*						
	Contig N50 (bp)	Sum length of contigs	Scaftig N50 (bp)	Sum Length of scaftigs	Scaffold N50 (bp)	Sum length of scaffolds	Genome coverage
ABySS	860	2.1 Gb	1,499	2.18 Gb	Not reported	Not reported	68%
SOAPdenovo	886	2.1 Gb	>4,000	2.37 Gb	>4,000	2.38 Gb	85%

*: Measurement is based on 40 × 210 bp paired-end Illumina GA reads of HapMap NA18507 individual published in 2008 by Dr Bentley *et al.*

evaluation. For ABySS, 99.4 per cent contigs could be aligned to other human DNA sequence resources, with fewer than five consecutive base mismatches at the termini and at least 95 per cent identity. This has not been translated to a per-base error rate. SOAPdenovo was used to try to identify SNPs in the Asian genome (and therefore could not be directly compared with ABySS on accuracy). 1.8 million SNPs were identified in the contigs aligned to the reference genome at a false

positive rate of 0.004 per cent, indicating a very high per-base accuracy in the aligned part. Neither tool evaluated the accuracy of scaftigs or scaffolds, however. The complex part of human genomes, where structural variations are enriched, is also difficult for *de novo* assemblers to scaffold correctly. Even with perfect simulation data, the error rate of scaftig by ABySS is still ~0.1 per cent, which is considered to be high when claiming structural variations. SOAPdenovo does not mention the error rate with simulated data, but examines aligned contigs with large insertions and deletions by an *in silico* method that counts the supporting paired-end reads and concludes that the majority are correct. There is still a need for experimental validation over a wider variation spectrum for all assemblers, however.

Another aspect of human genome short-read assemblers that should be discussed is the genome coverage. As shown with SOAPdenovo, gene coverage is significantly higher than genome coverage, which indicates that the missing part is mainly due to the highly repetitive sequences. A possible explanation is that the repeat sequences continue for so long in the DNA that they cannot be spanned by the paired-end insert size. The majority of scaffold internal gaps simply arise because the original repeat elements are not assembled, however. Repeat units that are similar but not identical to each other, with extremely high copy numbers, would complicate the original de Bruijn graph making it a highly complex net-like sub-graph (because of mismatches between paralogue copies). This is likely to be removed or cut into pieces in further steps. Alu sequences, which are abundant in human genomes, constitute a particular problem associated with this issue. This could be one reason for only observing a significant peak at 200–500 bp in the length distribution of deletions but not insertions detected in the African genome in AbySS, as mentioned above.[7]

Discussion

The algorithmic features, processes and performances of two public human genome-capable assembly softwares have been reviewed here. From the computational aspect and the outcome, it is concluded that human genome assembly from short sequences is feasible using novel bioinformatics.

Since a high-quality human reference genome has already been obtained, new human genome assemblies are expected to aim for comprehensive detection of genetic variation. This is not feasible using mapping-based approaches since they miss highly polymorphic regions, indels larger than a limited number of bp, structural variations, novel sequences etc. All of these variation categories will be of great interest in further human and medical genomics studies. Thus, *de novo* assembly will outperform resequencing in human genomics studies. It is not yet known whether *de novo* assemblers are able to distinguish genetic variations from misassemblies, as the methodologies have not been fully established. Assembly-based applications and the refined human *de novo* assembly approach itself are still very challenging topics in the human bioinformatics field.

As technology improves, methods should also be adapted. Current Illumina GA sequencing can deliver read lengths of more than 100 bp and paired-end libraries with insert sizes larger than 10 kb in some laboratories. It remains to be discussed how this will affect the suitability of different data structures and algorithms. It is expected that the longer reads will at least benefit the recovery of fragmented repeat sequences, which should improve the continuity of human genome assemblies. In addition, long-insert paired-end libraries would significantly help to build much larger scaffolds. We are also expecting a diploid genome assembler, which should build the two haploid sequences that are closer to the biological truth of any human genome.

All in all, whole-genome diploid assembly of human DNA should be the ideal solution to future human and medical genomics research, as it provides the full and unbiased solution to all the genetic variations of interest.

References

1. International Human Genome Sequencing Consortium (2004), 'Finishing the euchromatic sequence of the human genome', *Nature* Vol. 431, pp. 931–945.
2. Wang, J, Wang, W., Li, R. *et al.* (2008), 'The diploid genome sequence of an Asian individual', *Nature* Vol. 456, pp. 60–65.

3. Bentley, D.R., Balasubramanian, S., Swerdlow, H.P., Smith, G.P. *et al.* (2008), 'Accurate whole human genome sequencing using reversible terminator chemistry', *Nature* Vol. 456, pp. 53–59.
4. Li, R., Li, Y., Zheng, H., Luo, R. *et al.* (2010), 'Building the sequence map of the human pan-genome', *Nat. Biotechnol.* Vol. 28, pp. 57–63.
5. Levy, S., Sutton, G., Ng, P.C., Feuk, L. *et al.* (2007), 'The diploid genome sequence of an individual human', *PLoS Biol.* Vol. 5, p. e254.
6. Flicek, P. and Birney, E. (2009), 'Sense from sequence reads: Methods for alignment and assembly', *Nat. Methods* Vol. 6, pp. S6–S12.
7. Simpson, J.T., Wong, K., Jackman, S.D., Schein, J.E. *et al.* (2009), 'ABySS: A parallel assembler for short read sequence data', *Genome Res.* Vol. 19, pp. 1117–1123.
8. Li, R., Zhu, H., Ruan, J., Qian, W. *et al.* (2010), 'De novo assembly of human genomes with massively parallel short read sequencing', *Genome Res.* Vol. 20, pp. 265–272.
9. Lander, E.S. and Waterman, M.S. (1988), 'Genomic mapping by fingerprinting random clones: A mathematical analysis', *Genomics* Vol. 2, pp. 231–239.
10. Pevzner, P.A., Tang, H. and Waterman, M.S. (2001), 'An Eulerian path approach to DNA fragment assembly', *Proc. Natl. Acad. Sci. USA* Vol. 98, pp. 9748–9753.
11. Zerbino, D.R. and Birney, E. (2008), 'Velvet: Algorithms for de novo short read assembly using de Bruijn graphs', *Genome Res.* Vol. 18, pp. 821–829.
12. Maccallum, I., Przybylski, D., Gnerre, S., Burton, J. *et al.* (2009), 'ALLPATHS 2: Small genomes assembled accurately and with high continuity from short paired reads', *Genome Biol.* Vol. 10, p. R103.
13. Chaisson, M.J. and Pevzner, P.A. (2008), 'Short read fragment assembly of bacterial genomes', *Genome Res.* Vol. 18, pp. 324–330.