Improving genome assemblies using multi-platform sequence data

Pınar Kavak [1,2,*], Bekir Ergüner [1], Duran Üstek [3], Bayram Yüksel [4],
Mahmut Şamil Sağıroğlu [1], Tunga Güngör [2], and Can Alkan [5,*]

(1) Advanced Genomics and Bioinformatics Research Group (İGBAM)
BİLGEM, The Scientific and Technological Research Council of Turkey (TÜBİTAK),
41470 Gebze, Kocaeli, Turkey, pinar.kavak@tubitak.gov.tr

(2) Department of Computer Engineering
Boğaziçi University, 34342 Bebek, İstanbul, Turkey

(3) Department of Medical Genetics
İstanbul Medipol University, 34810 Beykoz, İstanbul, Turkey

(4) TÜBİ TAK - MAM - GMBE (The Scientific and Technological Research Council of Turkey, Genetic Engineering and Biotechnology Institute), 41470 Gebze, Kocaeli, Turkey

(5) Department of Computer Engineering
Bilkent University, 06800 Bilkent, Ankara, Turkey, calkan@cs.bilkent.edu.tr

Keywords: *de novo* assembly, assembly improvement, next generation multi-platform sequencing.

Abstract. *De novo* assembly using short reads generated by next generation sequencing technologies is still an open problem. Although there are several assembly algorithms developed for data generated with different sequencing technologies, and some that can make use of hybrid data, the assemblies are still far from being perfect. There is still a need for computational approaches to improve draft assemblies. Here we propose a new method to correct assembly mistakes when there are multiple types of data obtained using different sequencing technologies that have different strengths and biases. We apply our method to Illumina, 454, and Ion Torrent data, and also compare our results with existing hybrid assemblers, Celera and Masurca.

1 Scientific Background

Since the introduction of high throughput next generation sequencing (NGS) technologies, traditional Sanger sequencing is being abandoned especially for large-scale sequencing projects. Although cost effective for data production, NGS also imposes increased cost for data processing and computational burden. In addition, the data quality is in fact lower, with greater error rates, and short read lengths for most platforms. One of the main algorithmic problems to analyze NGS data is the *de novo* assembly: i.e. "stitching" billions of short DNA strings into a collection of larger sequences, ideally the size of chromosomes. However, "perfect" assemblies with no gaps and no errors are still lacking due to many factors, including the short read and fragment (paired-end) lengths, sequencing errors in basepair level, and the complex and repetitive nature of most genomes. Some of these problems in *de novo* assembly can be ameliorated through using data generated by different sequencing platforms, where each technology has "strengths" that may be used to fix biases introduced by others.

Overlap-layout-consensus (OLC) graph based assemblers [1, 2] work well on the long read assembly. de Bruijn graph based assemblers [3, 4, 5] are designed for the

*to whom correspondence should be addressed

assembly of short reads. There are also hybrid assemblers [6, 7, 8] which use multiple read libraries. There are some existing methods such as [9] to improve the hybrid assembly quality. Pre-processing and post-processing operations before and after the assembly takes an important role on the assembly quality.

In this work, we propose a method to improve draft assemblies (i.e. produced using a single data source, and/or single algorithm) by incorporating data generated by different NGS technologies, and applying novel correction methods. To achieve better improvements, we exploit the advantages of both short but low-error and long but erroneous reads. We show that correcting the contigs built by assembling long reads through mapping short (and high quality) read contigs produce the best results, compared to the assemblies generated by algorithms that use hybrid data.

2 Materials and Methods

We cloned a bacterial artificial chromosome (BAC) of human chromosome 13 for a previous study. We chose human DNA sequence for this study, because human genome sequence is commonly used and there are still unsolved problems on human genome assembly. The reason why we chose chromosome 13, is just because it was available from a previous study. We then sequenced the BAC separately using Illumina, Roche/454, and Ion-Torrent platforms. Data properties are shown in Table 1. We also obtained a "gold standard" reference assembly using template-based assembly with Mira [8] with Roche/454, which is then corrected with the Illumina reads. Since Roche/454 and Ion Torrent platforms have similar sequencing biases (i.e. problematic homopolymers), we worked on two separate groups: Illumina & 454 and Illumina & Ion-Torrent, which gives us an opportunity to compare Roche/454 and Ion-Torrent.

Table 1: Properties of the data

Technology	Length range	Mean length	Mean seq. qual (phred s.)	Paired
Illumina	101bp (all reads have equal length)	101bp	38	paired
Roche/454	40bp-1027bp	650bp	28	single-end
Ion-Torrent	5bp-201bp	127bp	24	single-end

Technology: The name of the sequencing technology used to produce the reads. **Length range**: Minimum and maximum lengths of the generated reads. **Mean length**: The mean length among all reads. **Mean seq. quality**: The average phred score sequence quality among all reads. Calculated by summing up all phred scores of all bases in all reads and dividing the sum to the base number. **Paired:** Represents if the sequencing is done as paired-end or single-end.

Pre-processing: We first discarded the reads that has low average quality value (phred score 17, i.e. $\geq$2% error rate). Next, we removed the reads with high N-density (with $>$10% of the read consisting of Ns). We then trimmed groups of bases that seem to be non-uniform according to sequence base content. We also inevitably applied each assembler's pre-processing operations.

Assembly: We used several assembly tools: Velvet[3], a de Bruijn graph based assembler to assemble the short reads; and two different overlap-layout-consensus (OLC) assemblers: Celera [1], and SGA [2] to assemble the long read data sets (Roche/454 and Ion Torrent) separately. Finally, we also used a de Bruijn based assembler, SPAdes[4] on the long read data. We then mapped all draft assemblies to the E. coli reference sequence to identify and discard E. coli contamination due to the cloning process. At the

end, we obtained one short read, and three long read assemblies.

Correction: We mapped the contigs obtained with the short reads onto the contigs generated by assembling long reads using BLAST[10]. Since BLAST may report multiple mapping locations due to repeats, we accepted only the "best" map locations. Reasoning from the fact that the short reads show less sequencing errors, we opted for the sequence reported by the short read based contigs over the long read contigs assemblies when there are disagreements between the pair, and patched the "less fragmented" long read assemblies. We repeated this process for each of the three long read assembly data sets. Correction algorithm is shown in Algorithm 1.

Evaluation: We mapped each of the final corrected assemblies onto the reference genome we constructed, calculated various statistics based on the comparisons, and estimated assembly qualities (Table 2). We also used two hybrid assemblers, Celera-CABOG [6] and Masurca [7] on the same data to compare our correction methodology with those of hybrid assembly algorithms.

Algorithm 1 Assemble the query (short reads contig) and the subject (long reads contig) according to mapping information

Require: mapping query and subject
  if the map does not start at the beginning of the subject **then**
    add the unmapping beginning of the subject
  end if
  if the map does not start at the beginning of the query **then**
    add the first part of the query to the result with lowercase letters
  end if
  add the mapping part of the query
  if the map does not end at the end of the query **then**
    add the last part of the query to the result with lowercase letters
  end if
  if the map does not end at the end of the subject **then**
    add the unmapping end of the subject
  end if

3 Results

We present a summary of the results in Table 2. Briefly, the Velvet assembly using only the Illumina reads showed better coverage (99%) and high average identity (97.5%) rates compared to Celera assembly using long reads. Correcting the Celera assembly with our method improves both coverage and average identity rates, which are then further improved by reiterative application of our method.

The coverage of 454 assembly increases up to 99.7% and the average identity rate increases up to 94.4% on the first correction cycle. The repetitive correction cycles increase the coverage and average identity rates. We stop the repetitive cycles if there is no improvement (≥ 0.001) on the average identity or regression on the average identity because of the increasing coverage. We see that correcting the long read assembly with the short read contigs works well with all kind of assemblers. Corrected SGA assembly has the highest coverage rate among all.

Assembling short and long reads separately with de Bruijn and OLC assemblers and correcting them give better results than assembling short and long reads together with a hybrid assembler such as Masurca or Celera. Masurca seems to have the best average identity rate on Illumina-Ion Torrent data, but the coverage for this run is just 1%. Celera-CABOG performs very well on Illumina-454 data, but no better than corrected SGA or corrected Celera with Illumina and 454. Celera-CABOG seems not to have any

contigs on Illumina-Ion-Torrent data, actually the method resulted with 487 contigs but all were eliminated on the E.coli contamination filtering phase.

Table 2: Results of assembly correction method on BAC data.

Name	Length	# of Contigs	# of Mapped Contigs	# of Covered bases	Coverage	Avg. Identity	# of Gaps	Size of Gaps
Reference	*176.843*							
Velvet								
Ill. Velvet	197,040	455	437	175,172	0.99055	0.97523	39	1,671
Celera								
454 Celera	908,008	735	735	172,563	0.97580	0.92599	18	4,280
Ion Celera	39,347	27	27	47,638	0.26938	0.96932	47	129,205
Corrected Celera								
Ill-454 Celera	4,945,785	895	270	176,368	0.99731	0.94370	5	475
Ill-454 Celera^{2*}	5,078,059	890	265	176,640	0.998852	0.944527	4	203
Ill-Ion Celera	93,909	30	28	81,819	0.46267	0.96327	36	95,024
Ill-Ion Celera^{2}	145,262	30	28	91,962	0.52002	0.97412	33	84,881
Ill-Ion Celera^{3}	216,167	30	28	99,645	0.56347	0.98066	34	77,198
SGA								
454 SGA	62,909,254	108,095	101,514	176,546	0.99832	0.97439	1	297
Ion SGA	842,997	6,417	6,122	153,092	0.86569	0.99124	197	23.751
Corrected SGA								
Ill-454 SGA	295,009	335	335	176,757	0.99951	0.96823	5	86
Ill-454 SGA^{2}	279,034	305	305	176,757	0.99951	0.96769	5	86
Ill-Ion SGA	197,509	291	291	175,052	0.98987	0.97501	45	1,791
Ill-Ion SGA^{2}	203,064	291	291	175,676	0.99340	0.97413	34	1,167
SPADES								
454 SPADES	12,307,761	49,824	49,691	176,843	1.0	0.98053	0	0
Ion SPADES	176,561	110	107	167,890	0.94937	0.92909	9	8,953
Corrected SPADES								
Ill-454 SPADES	290,702	298	298	176,454	0.99780	0.96538	5	389
Ill-Ion SPADES	198,665	52	52	171,977	0.97248	0.94215	4	4,866
Ill-Ion SPADES^{2}	200,307	52	52	172,101	0.97319	0.94230	2	4,742
Masurca								
Ill-454 Masurca	380	1	0	0	0	0	0	0
Ill-Ion Masurca	2,640	8	8	1,952	0.01104	0.98223	9	174,891
Celera-CABOG								
Ill-454 Celera	1,101,716	891	891	174,330	0.98579	0.92452	12	2,513
Ill-Ion Celera	0	0	0	0	0.0	0.0	0	0.0

Name: the name of the data group that constitute the assembly; # of contigs: the number of contigs that belong to the resulting assembly; # of Mapped Contigs: the number of contigs that successfully mapped onto the reference sequence; # of Covered bases: the number of bases on the reference sequence that are covered by the assembly; Coverage: percentage of covered reference; Avg. identity: percentage of the correctly predicted reference bases; # of Gaps: The number of gaps that cannot be covered on the reference genome; Size of Gaps: total number of bases on the gaps.

* "2" represents the results of the second cycle of correction, "3" represents the third cycle.

4 Conclusion

Assembly correction by using advantages of different technologies improves the resulting assembly. In this paper, we presented a new method to improve draft assemblies by correcting high contiguity assemblies using high quality short read contigs.

Our results show that our method is useful and gives better results than using all data for once with a hybrid assembler. However, the need to develop new methods that exploit different data properties of different NGS technologies, such as short/long reads or high/low quality of reads, remains. For future work,

Acknowledgements

Funding The project is supported by the Republic of Turkey Ministry of Development Infrastructure Grant (no: 2011K120020), BİLGEM - TÜBİTAK (The Scientific and Technological Research Council of Turkey) grant (no: T439000), and a TÜBİTAK

grant to C.A.(112E135).

References

[1] E.W.Myers *et al* (2000) A Whole-Genome Assembly of Drosophila, *Science*, 287(no:5461):2196-2204, doi:10.1126/science.287.5461.2196.

[2] J.Simpson *et al* (2012) Efficient *de novo* Assembly of Large Genomes Using Compressed Data Structures, *Genome Research*, 22:549-556, doi:10.1101/gr.126953.111.

[3] D.Zerbino, E.Birney (2000) Velvet: Algorithms for *de novo* Short Read Assembly Using de Bruijn Graphs, *Genome Research*, 18(5):821-829, doi: 10.1101/gr.074492.107.

[4] A.Bankevich *et al* (2012) SPAdes: A New Genome Assembly Algorithm and Its Applications to Single-Cell Sequencing, *Journal of Computational Biology*, 19(5):455-477, doi:10.1089/cmb.2012.0021.

[5] J.Butler *et al* (2008) ALLPATHS: *De novo* Assembly of Whole-Genome Shotgun Microreads, *Genome Research*, 18(5):810-820, doi:10.1101/gr.7337908.

[6] J.R.Miller *et al* (2008) Aggressive Assembly of Pyrosequencing Reads with Mates, *Bioinformatics*, 24(24):2818-2824, doi:10.1093/bioinformatics/btn548.

[7] A.Zimin *et al* (2013) The MaSuRCA Genome Assembler, *Bioinformatics*, 29(21):2669-2677, doi:10.1093/bioinformatics/btt476.

[8] B.Chevreux *et al* (1999) Genome Sequence Assembly Using Trace Signals and Additional Sequence Information, *Computer Science and Biology:Proceedings of the German Conference on Bioinformatics (GCB)*, 99:45-56.

[9] Y.Wang *et al* (2012) Optimizing Hybrid Assembly of Next-Generation Sequence Data from Enterococcus Faecium: a Microbe with Highly Divergent Genome, *BMC Systems Biology*, 6(Suppl 3):S21, doi:10.1186/1752-0509-6-S3-S21.

[10] S.Altschul *et al* (1990) Basic Local Alignment Search Tool, *Journal of Molecular Biology*, 215(3):403-410.