CGAL: computing genome assembly likelihoods

*Genome Biology* 2013, 14:R8  doi:10.1186/gb-2013-14-1-r8

Atif Rahman (atif@eecs.berkeley.edu)
Lior Pachter (lpachter@math.berkeley.edu)

ISSN 1465-6906

Article type Method
Submission date 23 August 2012
Acceptance date 20 December 2012
Publication date 29 January 2013

Article URL [http://genomebiology.com/2013/14/1/R8](http://genomebiology.com/2013/14/1/R8)

This peer-reviewed article can be downloaded, printed and distributed freely for any purposes (see copyright notice below).

Articles in *Genome Biology* are listed in PubMed and archived at PubMed Central.

For information about publishing your research in *Genome Biology* go to [http://genomebiology.com/authors/instructions/](http://genomebiology.com/authors/instructions/)

© 2013 Rahman and Pachter
This is an open access article distributed under the terms of the Creative Commons Attribution License ([http://creativecommons.org/licenses/by/2.0](http://creativecommons.org/licenses/by/2.0)), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.
CGAL: computing genome assembly likelihoods

Atif Rahman\textsuperscript{1}, Lior Pachter\textsuperscript{1,2*}

\textsuperscript{1}Department of Electrical Engineering and Computer Sciences, 387 Soda Hall, UC Berkeley, Berkeley, CA 94720, USA
\textsuperscript{2}Departments of Mathematics and Molecular & Cell Biology, 970 Evans Hall, UC Berkeley, Berkeley, CA 94720, USA

Email: Atif Rahman - atif@eecs.berkeley.edu; Lior Pachter* - lpachter@math.berkeley.edu;

*Corresponding author
Abstract

Assembly algorithms have been extensively benchmarked using simulated data so that results can be compared to ground truth. However, in de novo assembly, only crude metrics such as contig number and size are typically used to evaluate assembly quality. We present CGAL, a novel likelihood-based approach to assembly assessment in the absence of a ground truth. We show that likelihood is more accurate than other metrics currently used for evaluating assemblies, and describe its application to the optimization and comparison of assembly algorithms. Our methods are implemented in software that is freely available at http://bio.math.berkeley.edu/cgal/.

Keywords

Genome assembly, evaluation, likelihood, sequencing.
**Background**

Genome assembly is the process of merging fragments of DNA sequence produced by shotgun sequencing in order to reconstruct the original genome. The assembly problem is known to be NP-hard for a number of formulations [1–3] and is also complicated by many types of sequencing errors, experimental biases and the volume of data that must be processed. For these reasons, in addition to differences in underlying theory and algorithms, popular assembly methods employ many different heuristics and assemblies produced by existing methods differ substantially from each other [4,5].

Paradoxically, the difficulties of sequence assembly have been compounded by sequencing advances in recent years collectively termed next-generation sequencing technologies. Next-generation sequencing technologies such as 454 pyrosequencing by Applied Sciences [6], Solexa/Illumina sequencing, the SOLiD technology from Applied Biosystems and the Helicos single-molecule sequencing [7] produce data of much greater volume at a much lower cost than traditional Sanger sequencing [8]. However, read lengths are considerably shorter and error rates are higher than those in Sanger sequencing. To allow de novo sequencing from short reads from next generation sequencing machines several assemblers have been developed such as Velvet [9], Euler-sr [10], ABySS [11], Edena [12], SSAKE [13], VCAKE [14], SHARCGS [15], Allpaths [16], SOAPdenovo [17], Celera WGA [18], the CLC bio assembler and others [4, 5]. A key problem that has arisen is to determine which assembler is “the best”. In the past this has been done with the help of a number of measures such as N50 scaffold or contig lengths - which is the maximum contig (scaffold) length such that at least half the total length is contained in contigs (scaffolds) of length greater or equal that length. Although simulation studies show that simple metrics correlate with assembly quality, currently used metrics are crude and provide only condensed summaries of the result. They can therefore be very misleading [5,19]. For example, the assembly consisting of simply gluing all reads end-to-end has a very large N50 length, but is obviously a poor assembly. Phillippy *et al.* presented a software called amosvalidate [20] that identifies mis-assembly features and suspicious regions but it does not have high specificity and has not been widely adopted. Narzisi *et al.* utilized feature-response curve [21] to rank assemblies based on features identified by amosvalidate. Studies such as [22–25] have discussed these issues and produce interesting insights into assembler performance but do not provide an intrinsic direct measure of assembly quality. The recent Assemblathon 1 competition used 10 different metrics [4] that attempt to reveal more information than just N50 values, but most of them can only be computed when the genome that is being assembled is known, and are therefore not useful in practice on real data.

In this paper we present a computationally efficient approach for computing the likelihood of an assembly
which provides a way to assess assemblies without a ground truth. Intuitively, the likelihood assessment evaluates the uniformity of coverage of the assembly, taking into account errors in the reads, the insert size distribution and the extent of unassembled data. Genome assembly by maximizing likelihood has been proposed previously by Myers [26] and Medvedev et al. [1] but their formulations are based on simplified models that omit evaluating and utilizing crucial parameters, especially sequencing error. To demonstrate the power of our approach for assembly quality evaluation we have implemented our methods in a program called CGAL that we evaluate by testing several assemblies from different programs with varying input parameters in setting where the desired target genome is known. For each assembly, we compute the likelihood using our tool and then compare our likelihood computation to standard measures such as N50 contig values, sequence similarity with the reference genome as well as values reported by amosvalidate. Although it is beyond the scope of this paper to compare all assemblers and explore all parameters, our results indicate that likelihood is meaningful and useful for evaluating assemblies.

Results

Our overall approach is simple: we describe a probabilistic generative model for sequencing that captures many aspects of sequencing experiments, and from which we can compute the likelihood of an assembly. This intuitive framework is, however, complicated by one major difficulty which is the problem we address in this paper: to compute the likelihood of an assembly it is necessary, in principle, to consider the possibility that a read was produced from every single location in the assembly. This results in an intractable computation, that we circumvent by approximating the likelihood via a reduction to a small set of “likely” sites from which each read originated (using a mapping of the reads to the assembly). This requires an examination of the quality of the approximation, and leads to yet another difficulty, which is how to compute the likelihood for reads that do not map to the assembly at all. These issues are addressed in this paper and their solution is what enables our program for likelihood computation to be efficient and practical.

We begin by describing the statistical model that forms the basis for our likelihood computation. We believe that our model incorporates many aspects of typical sequencing experiments, but it can be easily generalized to accommodate additional parameters if desired.

A generative model for sequencing

Let, $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$ be a set of $N$ paired end (or mate pair) reads generated from a genome, $\mathcal{G}$ (our model can, in principle be adapted to single end reads but we do not consider that here). We assume a
fragment represented by two paired-end reads \( r_i = (r_{i1}, r_{i2}) \) is generated according to the following model:

- A fragment length, \( l_i \) is selected according to a distribution, \( F \).
- A site for the 5' end of the fragment, \( s_i \) is selected according to a distribution, \( S \).
- The ends of the fragment are read as \( r_{i1} \) and \( r_{i2} \) according to an error model, \( E \) which comprises mismatches as well as indels.

The generative model is illustrated in Figure 1.

### Computing likelihood

Computing the likelihood of an assembly means that the probability of the (observed) set of reads is computed with respect to a proposed assembly using the model described in the previous section. The probability of a sequence of length \( L \) generating a paired-end or mate pair read (termed read from now on) \( r_i \) is

\[
p(r_i) = \sum_{l=1}^{L} p_F(l) \sum_{s=1}^{L-l+1} p_S(s) \sum_{e \in E} p_E(r_i | a_s \ldots a_{s+l-1}, e)
\]

where \( a_s \ldots a_{s+l-1} \) is the assembly subsequence starting at \( s \) of length \( l \), \( E \) denotes all possible ways of obtaining \( r_i \) from \( a_s \ldots a_{s+l-1} \) and

\[
p_F(l) = \text{Probability that the fragment length is } l
\]
\[
p_S(s) = \text{Probability that the 5' end of the fragment is at site } s
\]
\[
p_E(r_i | a_s \ldots a_{s+l-1}, e) = \text{Probability of obtaining } r_i \text{ from } a_s \ldots a_{s+l-1}
\]
\[
\text{with sequencing errors given by } e
\]

Although in theory a read could have been generated from any site (assuming that every base could have been an error), in practice the probability decreases considerably with increasing number of disagreements between the source sequence and the read sequence. We therefore approximate the probability \( p(r_i) \) by mapping the read to the assembly and ignoring mappings with large number of differences. If \( M_i \) is the number of such mappings of read \( r_i \), the probability is given by

\[
p(r_i) \approx \sum_{j=1}^{M_i} p_F(l_{i,j}) p_S(s_{i,j}) p_E(r_i | a_{i,j}, e_{i,j})
\]
where $l_{i,j}$, $s_{i,j}$, $a_{i,j}$ and $e_{i,j}$ are fragment length, start site, assembly subsequence and errors corresponding to $j$-th mapping of $i$-th read respectively. The above equation generalizes to assemblies with more than one contig. Given an assembly $A$ and a set of reads $\mathcal{R} = \{r_1, r_2, \ldots, r_N\}$, the log likelihood is given by

$$l(A; \mathcal{R}) = \log \prod_{i=1}^{N} p(r_i | A)$$

$$\approx \sum_{i=1}^{N} \log \sum_{j=1}^{M_i} p_F(l_{i,j}) p_S(s_{i,j}) p_E(r_i | a_{i,j}, e_{i,j}).$$

In the above equation $M_i \geq 1$ for all reads $r_i$, and in Methods we explain how we obtain alignments for all reads and how to learn the needed distributions.

**Validation with simulated data**

To test our implementation we developed a simulator that generates reads according to error parameters provided and fragment lengths distributed according to a Gaussian distribution.

We generated 3 million 35bp paired end reads from a strain of *Escherichia coli* ([NCBI: NC_000913.2]) and an assembly of *Grosmannia clavigera* ([DDBJ/EMBL/GenBank: ACXQ00000000]) reported in [27]. Table 1 shows percentage difference in likelihood values computed using true parameters provided to the simulator and using parameters inferred by CGAL.

**Performance of assemblers on E. coli reads**

We assessed performance of four assemblers: Velvet, Euler-sr, ABySS and SOAPdenovo on an *Escherichia coli* dataset ([SRA:SRR 001665] and [SRA:SRR 001666]). We chose *E. coli* because its assembly is a true “gold standard” without questions about reliability or accuracy. We assembled the reads using the assemblers mentioned for different hash lengths ($k$-mer used for constructing de Bruijn graph [10]). Likelihood values for assemblies along with the likelihood value for the reference ([NCBI: U00096.2]) are shown in Figure 2.

For this dataset ABySS outperforms others when likelihood is used as the metric. We also aligned the assemblies to the reference with NUCmer [28] and Figure 3 shows differences from reference against hash lengths. The relations among likelihood, N50 length and similarity are illustrated in Figure 4 and Additional file 1, Figure S1. They suggest that likelihood values are better at capturing sequence similarity than other metrics commonly used for evaluating assemblies such as N50 scaffold or contig lengths. We also ran the amosvalidate pipeline to obtain numbers of mis-assembly of features and suspicious regions (Figure 5) and plotted the feature response curves (FRC) [21] of the assemblies (Additional file 1, Figures S4-S5). FRC also ranks an ABySS assembly as the best one.
Similar analysis was performed on a different *Escherichia coli* dataset downloaded from CLC bio [29]. It consists of approximately 2.6 million 35bp paired end Illumina reads (approximately 40X coverage) along with a reference genome ([NCBI: NC_010473.1]). We noticed that many of the assemblies have better likelihood than the reference. However, we assembled reads that could not be mapped to the reference and after running BLAST [30] we found another substrain of *Escherichia coli* strain K-12, MG1655 ([NCBI: NC_000913.2]) that has a better likelihood than all assemblies. We conjecture that the reads were generated from NC_000913.2. Likelihood values are shown in Figure 6 and relationships among likelihood, similarity and N50 values are illustrated in Additional file 1, Figures S6-S10.

**Performance of assemblers on G. clavigera reads**

To assess assemblies of a larger genome, we used the dataset generated for sequencing an ascomycete fungus, *Grosmaninia clavigera* by DiGuistini *et al.* [27]. We ran Velvet, ABySS and SOAP on PE Illumina reads with fragment length mean of 200 bp [SRA:SRR 018008-11] and 700 bp [SRA:SRR 018012].

The likelihood values of the 200bp fragment reads for the assemblies are shown in Figure 7. It also shows likelihood values for assemblies [DDBJ/EMBL/GenBank: ACXQ00000000] and [DDBJ/EMBL/GenBank: ACYC00000000] reported in [27] which were generated using Sanger and 454 reads as well as Illumina reads. The numbers of mis-assembly features and suspicious regions identified by amosvalidate and the feature response curves (FRC) are shown in Additional file 1, Figures S14-S15.

Figure 8 shows that the assembly with most sequence coverage is produced by ABySS. However, in this case ABySS assemblies are much longer compared to other assemblies and references (Additional file 1, Tables S9-S11). This results in lower likelihoods compared to some assemblies by Velvet and SOAPdenovo. In FRC analysis, coverage is estimated using assembly length and so it does not take into account the unassembled sequences and ranks ABySS assemblies above others. It is interesting that assemblies with the best likelihood and sequence similarity are generated for higher values of hash length than are optimal for producing high N50 values.

**GAGE results**

We computed likelihoods for the assemblies generated in the GAGE project [5]. Additional file 1, Tables S12-S14 show likelihoods of Library 1 and number of reads mapped to assemblies by Bowtie 2 [31]. We found that likelihood values of Library 1 are dominated by coverage and contiguity does not affect these values greatly. However, contiguity have more affect on likelihoods of Library 2 with longer insert size (Additional
file 1, Tables S12-S14) as might be expected. Total likelihood along with coverage and N50 values are shown in Tables 2-5. For human we computed Library 2 likelihoods for assemblies with best three likelihoods of Library 1. Likelihood values of Library 2 for bumble bee assemblies were not computed as only a small fraction of the reads could be mapped to the assemblies.

**Assemblathon 1 results**

We also analyzed the assemblies submitted for Assemblathon 1 [4]. Likelihoods of library of insert size of mean 200bp for all assemblies are given in Additional file 1, Table S15 and Figure 10 shows the relationship between likelihood and coverage. Among these, we took the entries with highest likelihood for top ten participants and computed likelihoods of libraries of insert sizes of means 3000bp and 10000bp. Table 6 shows total likelihoods of top ten participants along with their Assemblathon 1 rankings.

**Discussion**

**E. coli**

We find that for both *E. coli* datasets assemblies with best likelihoods are constructed by ABySS. They also have most similarity with references (assuming [NCBI: NC_000913.2] is the reference for the CLC bio dataset). The $R^2$ values (Figures 4-5 and Additional file 1, Figure S1) reveal that likelihoods reflect sequence similarity better than contiguity statistics such as N50 values as well as numbers of mis-assembly features and suspicious regions identified by amosvalidate. Analysis of two different *E. coli* datasets also reveal that for assemblers like Velvet, SOAPdenovo higher likelihood values are achieved for different values of k-mer length used to construct the de Bruijn graph during assembly.

**G. clavigera**

For the *G. clavigera* dataset one of the Velvet assemblies has the highest likelihood. Although ABySS assemblies have more coverage, they have lower likelihood because of much longer total length. Despite this we see from $R^2$ values that likelihood values reflect sequence similarity better than N50 values (Figure 9 and Additional file 1, Figures S11,S14) and numbers of mis-assembly features and suspicious regions reported by amosvalidate. This suggests that likelihood values are useful in simultaneously evaluating coverage and total assembly length.
GAGE

For the GAGE S. aureus dataset, we find that the assembly generated using Velvet has the best likelihood but likelihoods of a few other assemblies are close. For R. sphaeroides, the ALLPATHS-LG assembly has the best likelihood which is also the assembly with highest coverage and N50 scaffold length. The CABOG assembly of human chromosome 14 is the one with best likelihood. The CABOG assembly also has the highest coverage and N50 contig length among the assemblies. In all three cases, we find that the reference sequences have the highest likelihoods and the highest number of reads mapped to them by Bowtie 2. For the bumblebee data, the assembly using CABOG has best likelihood among the three (the likelihood of SOAPdenovo assembly could not be computed as reads could not be mapped to it using Bowtie 2).

Assemblathon 1

Figure 10 reveals that for the Assemblathon 1 dataset, likelihoods of small fragment library correlates well with coverages. Overall, we find that participants with the ten highest likelihoods were ranked within the top eleven by Assemblathon 1 organizers but there are differences between the two rankings. The entry of highest likelihood is by Beijing Genomics Institute (BGI) which was ranked two in the original paper. The differences in rankings are primarily due to the emphasis on contiguity by Assemblathon 1 organizers while our likelihood model implicitly places high importance on coverage. This brings up the issue that better contiguity statistics can be achieved by not reporting hard to assemble regions and these values may be misleading if they are not used in conjunction with an indicator of coverage.

Applications

Currently, assembly evaluation projects rely mostly on simulated data or data from genomes that have been sequenced previously [4,5]. Having a tool that can assess quality of assembly without the need for a reference will allow researchers who work with real data from genomes that have not been sequenced before to assess the performance of different assemblers on their data, and to optimize parameters in the programs they are using.

Analysis of two different datasets from E. coli reveal that performance of some assemblers vary significantly depending on the k-mer chosen for constructing the de Bruijn graph. Moreover, the ‘optimal’ value depends on read length and sequence coverage. Likelihood values can therefore guide selection of parameter values.

Maximum likelihood genome assembly was introduced by Medvedev and Brudno [1] but they do not
consider sequencing errors or paired end reads. A likelihood model taking into account these may be the
next step towards genome assemblers for real data that try to maximize likelihood.

Conclusions
In this paper we presented a tool for computing the likelihood of an assembly. The result can be used as a
metric for evaluating and comparing assemblies. In the past this has been done using many different criteria
including N50 lengths, total sequence length, number of contigs. The likelihood model incorporates these
directly or indirectly in addition to other important factors such as genome coverage and assembly accuracy
and combines them into a single metric for evaluation.

We have also used our tool to assess performance of some assemblers on a few different datasets. Our
results indicate that likelihood reflects sequence similarity which is missed by other metrics commonly used
and is going to be a valuable tool for evaluating assemblies generated by different assemblers and for different
values for input parameters.

Materials and methods
Mapping reads
The first step in computing the likelihood is mapping reads to the assembly. A number of tools are available
for this such as Bowtie [31, 32], MAQ [33], BWA [34] and BFAST [35]. Our present implementation can
use either BFAST or Bowtie 2 for mapping reads as they support mapping with indels and report multiple
alignments in a way that gives all the required information without accessing the assembly sequence. But
any tool that reports multiple alignments of reads and allows for insertions/deletions can be used with some
minor modifications.

However, existing tools do not usually map all reads, and for the likelihood computation it is necessary
to assign probabilities to reads that are unmapped. We found that mapping tools were unable to map a
large fraction of reads in our experiments. One option is to assign probabilities to these reads assuming that
they could have been generated from any site with number and types of errors not handled by the mapping
tool. But it is then often the case that unmapped reads are deemed more probable than mapped ones,
which we believe is anomalous. Furthermore, in our analyses we determined that the resulting probabilities
were inaccurate (results not shown). Therefore, we chose to directly align the reads not mapped by BFAST
or Bowtie 2 using an adaptation of the Smith-Waterman algorithm. For this we have adapted the striped
implementation of Smith-Waterman algorithm by Farrar [36]. This step is time consuming, so we align only a random subset of reads with the number specified by the user and approximate probabilities using these.

Learning Distributions

To compute the likelihood from mapped reads, we need to learn the distribution of fragment lengths, their distribution across genome and error characteristics. Since they differ with library preparation methods and sequencing instruments, we have chosen to learn these from sequencing data generated in the experiment. We do this by mapping reads to the assembly and using reads that map uniquely. However, this can be easily extended to take into account all reads by using the EM algorithm at the expense of more iterations. We explain each distribution in more detail below.

- **Fragment length distribution**

  The distribution of fragment lengths depends on the method used for size selection and may not be approximated well by common distributions [37]. So, we use the empirical distribution.

- **Distribution of fragments along genome**

  In our implementation, we assume that fragments are distributed uniformly across the genome. We leave incorporating sequencing bias as future work.

- **Error model**

  In the error model used at present, we have made the assumption that sequencing errors are independent of one another. We learn an error rate for each position in the read since error rates are known to be different across positions in reads. [38] We also learn separate error rates for each type of base and substitution types. Although errors are known to depend on sequence context [38] we have ignored them for the sake of simplicity.

  To account for varying indel rates across positions in reads, we learn an insertion rate and a deletion rate for each position in the read. Since short indels are more likely than longer ones we also count number of insertions and deletions by length.

Implementation

As mentioned earlier, we use BFAST or Bowtie 2 to map reads to assemblies. The parameters are set so that they report all alignments of a read found.
The remaining code for computing likelihood is written in C++ and it consists of three parts.

- **convert**: It converts the output generated by BFAST or Bowtie 2 to an internal format. It also separates reads with no end or one end mapped and reads with ends mapped to different scaffolds if needed. Separating this module also allow us to support other mapping tools by writing a conversion routine.

- **align**: To align the reads not mapped by the mapping tool, we have adapted the striped implementation of Smith-Waterman algorithm by Farrar [36]. As this step is time consuming, we align a random subset of reads with the number determined by the user. This step is multithreaded to speed up the process.

- **cgal**: This part learns the fragment length distribution and parameters for the error model using uniquely mapped reads and then uses these to compute the likelihood value.

**Assembling genomes**

To assemble reads, we varied \( k \)-mer length used to construct the de Bruijn graph to obtain different assemblies for each assembly tool. For other parameters default values or values suggested in manuals were used.

**Data analysis**

Likelihoods were computed by running CGAL with default parameters and aligning between 300 and 1000 randomly chosen reads not mapped the mapping tool used. The running time of CGAL was approximately 1/3 the time taken to map reads using Bowtie 2.

To compute the difference between an assembly and the reference we aligned the assembly to the reference using NUCmer [28] and the difference refers to the number of bases in reference that are either not covered by the assembly or different in reference and assembly. Contigs have been generated by splitting scaffolds at sites with 25 or more N’s (character representing any base).

**Competing interests**

The authors have no competing interests.

**Authors’ contributions**

AR and LP conceived the project and developed methodology. AR implemented the method in the CGAL software and obtained the results of the paper. AR and LP wrote the manuscript. All authors read and
approved the final manuscript.

Acknowledgements

We thank Michael Eisen, Aaron Kleinman, Harold Pimentel and Adam Roberts for helpful conversations in the development of the likelihood based approach to assembly evaluation. Lior Pachter was funded in part by NIH R21 HG006583. Atif Rahman was funded in part by Fulbright Science & Technology Fellowship 15093630.
References

1. Medvedev P, Brudno M: Maximum likelihood genome assembly. Journal of computational biology : a journal of computational molecular cell biology 2009, 16:1101–1116, [http://dx.doi.org/10.1089/cmb.2009.0047].

2. Medvedev P, Georgiou K, Myers G, Brudno M: Computability of Models for Sequence Assembly. In Algorithms in Bioinformatics, Volume 4645 of Lecture Notes in Computer Science. Edited by Giancarlo R, Hannenhalli S, Springer Berlin / Heidelberg 2007:289–301, [http://dx.doi.org/10.1007/978-3-540-74126-8_27].

3. Nagarajan N, Pop M: Parametric complexity of sequence assembly: theory and applications to next generation sequencing. Journal of computational biology : a journal of computational molecular cell biology 2009, 16:897–908, [http://dx.doi.org/10.1089/cmb.2009.0005].

4. Earl DA, Bradnam K, St John J, Darling A, Lin D, Faas J, Yu HOK, Vince B, Zerbino DR, Diekhans M, Nguyen N, Nuwantha P, Sung AWK, Ning Z, Haimel M, Simpson JT, Fronseca NA, Birol n, Docking TR, Ho IY, Rokhsar DS, Chikhi R, Lavenier D, Chapuis G, Naquin D, Schatz MC, Kelly DR, Phillipsy AM, Koren S, Yang SP, Wu W, Chou WC, Srivastava A, Shaw TI, Ruby JG, Skewes-Cox P, Betegon M, Dimon MT, Solovyev V, Kosarev P, Vorobyev D, Ramirez-Gonzalez R, Leggett R, MacLean D, Xia F, Luo R, L Z, Xie Y, Liu B, Guerre S, MacCallum I, Przybylski D, Ribeiro FJ, Yin S, Sharpe T, Hall G, Kersey PJ, Durbin R, Jackman SD, Chapman JA, Huang X, DeRisi JL, Caccamo M, Li Y, Jaffe DB, Green R, Haussler D, Korf I, Paten B: Assemblathon 1: A competitive assessment of de novo short read assembly methods. Genome Research 2011, [http://genome.cshlp.org/content/early/2011/09/16/gr.126599.111.abstract].

5. Salzberg SL, Phillippy AM, Zimin A, Puini D, Magoe T, Koren S, Treangen TJ, Schatz MC, Delcher AL, Roberts M, Marçais G, Pop M, Yorke JA: GAGE: A critical evaluation of genome assemblies and assembly algorithms. Genome Research 2011, [http://genome.cshlp.org/content/early/2012/01/12/gr.131383.111.abstract].

6. Margulies M, Egholm M, Altman WE, Attiya S, Bader JS, Bemben LA, Bemben LA, Berg YJ, Chen Z, Dewell SB, Du L, Fierro JM, Gomes VM, Goodwin BC, He W, Helgesen S, Ho CH, Ho CH, Izyryk GP, Jando SC, Alenquer ML, Allen ML, Jarvis KB, Jirage KB, Kim JB, Knight DJ, Lanza JR, Leamon JH, Lefkowitz SM, Lei M, Li J, Lohman KL, Lu H, Makhijani VB, McAdams KE, McKenna MP, Myers EW, Nickerson E, Nobile JR, Plant R, Puc BP, Ronan MT, Roth GT, Sarkis GJ, Simons FY, Simpson JW, Srinivasan M, Tartaro KR, Tomasz A, Vogt KA, Volkmer GA, Wang SH, Wang Y, Weiner MP, Yu P, Begley RF, Rothenberg JM: Genome Sequencing in Microfabricated High-Density Picolitre Reactions. Nature 2005, 437:376–380, [http://dx.doi.org/10.1038/nature03959].

7. Harris TD, Buzby PR, Babcock H, Beer E, Bowers J, Braverman MS, Chen YJ, Chen Z, Dewell SB, Du L, Fierro JM, Gomes VM, Goodwin BC, He W, Helgesen S, Ho CH, Ho CH, Izyryk GP, Jando SC, Alenquer ML, Allen ML, Jarvis KB, Jirage KB, Kim JB, Knight DJ, Lanza JR, Leamon JH, Lefkowitz SM, Lei M, Li J, Lohman KL, Lu H, Makhijani VB, McAdams KE, McKenna MP, Myers EW, Nickerson E, Nobile JR, Plant R, Puc BP, Ronan MT, Roth GT, Sarkis GJ, Simons FY, Simpson JW, Srinivasan M, Tartaro KR, Tomasz A, Vogt KA, Volkmer GA, Wang SH, Wang Y, Weiner MP, Yu P, Begley RF, Rothenberg JM: Genome Sequencing in Microfabricated High-Density Picolitre Reactions. Nature 2005, 437:376–380, [http://dx.doi.org/10.1038/nature03959].

8. Sanger F, Nicklen S, Coulsen AR: DNA sequencing with chain-terminating inhibitors. Proceedings of the National Academy of Sciences of the United States of America 1977, 74:5463–5467, [http://www.pnas.org/content/74/12/5463.abstract].
15. Dohm JC, Lottaz C, Borodina T, Himmelbauer H: SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. Genome Research 2007, 17:1697–1706, [http://genome.cshlp.org/content/17/11/1697.abstract].

16. Butler J, MacCallum I, Kleber M, Shlyakhter IA, Belmonte MK, Lander ES, Nusbaum C, Jaffe DB: ALLPATHS: De novo assembly of whole-genome shotgun microreads. Genome Research 2008, 18:810–820, [http://genome.cshlp.org/content/18/5/810.abstract].

17. Li R, Zhu H, Ruan J, Qian W, Fang X, Shi Z, Li Y, Li S, Shan G, Kristiansen K, Li S, Yang H, Wang J, Wang J: De novo assembly of human genomes with massively parallel short read sequencing. Genome Research 2010, 20:265–272, [http://www.genome.org/cgi/content/207/5461/2196.abstract].

18. Myers EW, Sutton GG, Delcher AL, Dew IM, Fasulo DP, Flanigan MJ, Kravitz SA, Mobarry CM, Reinert KH, Remington KA, Anson EL, Bolanos RA, Chou HH, Jordan CM, Halpern AL, Lonardi S, Beasley EM, Brandon RC, Chen L, Dunn PJ, Lai Z, Liang Y, Nusskern DR, Zhan M, Zhang Q, Zheng X, Rubin GM, Adams MD, Venter JC: A Whole-Genome Assembly of Drosophila. Science 2000, 287:2196–2204, [http://www.sciencemag.org/content/287/5461/2196.abstract].

19. Vezzi F, Narzisi G, Mishra B: Feature-by-Feature – Evaluating De Novo Sequence Assembly. PLoS ONE 2012, 7:e31002, [http://dx.doi.org/10.1371%2Fjournal.pone.0031002].

20. Phillippy A, Schatz M, Pop M: Genome assembly forensics: finding the elusive mis-assembly. Genome Biology 2008, 9:R55, [http://genomebiology.com/2008/9/3/R55].

21. Narzisi G, Mishra B: Comparing De Novo Genome Assembly: The Long and Short of It. PLoS ONE 2011, 6:e19175, [http://dx.doi.org/10.1371%2Fjournal.pone.0019175].

22. Zhang W, Chen J, Yang Y, Tang Y, Shang J, Shen B: A Practical Comparison of De Novo Genome Assembly Software Tools for Next-Generation Sequencing Technologies. PLoS ONE 2011, 6:e17915, [http://dx.doi.org/10.1371%2Fjournal.pone.0017915].

23. Lin Y, Li J, Shen H, Zhang L, Papasian CJ, Deng HW: Comparative Studies of de novo Assembly Tools for Next-generation Sequencing Technologies. Bioinformatics 2011, [http://bioinformatics.oxfordjournals.org/content/early/2011/06/02/bioinformatics.btr319.abstract].

24. Darling AE, Tritt A, Eisen JA, Facciotti MT: Mauve Assembly Metrics. Bioinformatics 2011, 27:2756–2757, [http://bioinformatics.oxfordjournals.org/content/27/19/2756.abstract].

25. Alkan C, Sajjadian S, Eichler EE: Limitations of next-generation genome sequence assembly. Nature Methods 2011, 8:61–65, [http://dx.doi.org/10.1038/nmeth.1527].

26. Myers EW: The fragment assembly string graph. Bioinformatics 2005, 21:79–85.

27. DiGuistini S, Liao N, Platt D, Robertson G, Seidel M, Chan S, Docking TR, Biro I, Holt R, Hirst M, Mardis E, Marra M, Hamelin R, BohLMann J, Breuil C, Jones S: De novo genome sequence assembly of a filamentous fungus using Sanger, 454 and Illumina sequence data. Genome Biology 2009, 10:R94, [http://genomebiology.com/2009/10/9/R94].

28. Delcher AL, Phillippy A, Carlton J, Salzberg SL: Fast algorithms for large-scale genome alignment and comparison. Nucleic acids research 2002, 30:2478–2483, [http://dx.doi.org/10.1093/nar/30.11.2478].

29. CLC bio: NGS example data [http://www.clcbio.com/index.php?id=1290].

30. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ: Basic local alignment search tool. Journal of molecular biology 1990, 215:403–410, [http://dx.doi.org/10.1016/0022-2836(90)90570-K].

31. Langmead B, Salzberg SL: Fast gapped-read alignment with Bowtie 2. Nature Methods 2012, 9:357–359, [http://dx.doi.org/10.1038/nmeth.1923].

32. Langmead B, Trapnell C, Pop M, Salzberg S: Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. Genome Biology 2009, 10:R25+, [http://dx.doi.org/10.1186/gb-2009-10-3-r25].

33. Li H, Durbin R: Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Research 2008, 18:1851–1858, [http://genome.cshlp.org/content/18/11/1851.abstract].

34. Li H, Durbin R: Fast and accurate short read alignment with Burrows-Wheeler transform. Bioinformatics (Oxford, England) 2009, 25:1754–1760, [http://dx.doi.org/10.1093/bioinformatics/btp324].
35. Homer N, Merriman B, Nelson SF: **BFAST: An Alignment Tool for Large Scale Genome Resequerencing.** *PLoS ONE* 2009, 4:e7767+, [http://dx.doi.org/10.1371/journal.pone.0007767].

36. Farrar M: **Striped Smith-Waterman speeds database searches six times over other SIMD implementations.** *Bioinformatics* 2007, 23:156–161, [http://bioinformatics.oxfordjournals.org/content/23/2/156.abstract].

37. Roberts A, Trapnell C, Donaghey J, Rinn J, Pachter L: **Improving RNA-Seq expression estimates by correcting for fragment bias.** *Genome Biology* 2011, 12:R22, [http://genomebiology.com/2011/12/3/R22].

38. Dohm JC, Lottaz C, Borodina T, Himmelbauer H: **Substantial biases in ultra-short read data sets from high-throughput DNA sequencing.** *Nucleic Acids Research* 2008, 36:e105, [http://dx.doi.org/10.1093/nar/gkn425].

**Figures**

**Figure 1 - A generative graphical model for sequencing**

*N* paired end reads are generated independently from a genome. Here, *F* denotes the distribution of fragment lengths, *S* is the distribution of start sites of reads and *E* stands for error parameters.

**Figure 2 - Hash length vs log likelihood for E. coli**

Log likelihoods of assemblies of *E. coli* reads are shown on the *y*-axis. Assemblies are generated using different assemblers for varying *k*-mer length shown on *x*-axis. The dotted line corresponds to the log likelihood of the reference.

**Figure 3 - Hash length vs difference from reference for E. coli**

Differences between assemblies and the reference are shown on the *y*-axis where difference refers to numbers of bases in the reference not covered by the assembly or are different in the reference and the assembly.

**Figure 4 - Log likelihood vs N50 scaffold length for E. coli**

Log likelihoods are shown on the *x*-axis and N50 scaffold lengths are shown on the *y*-axis. Each circle corresponds to an assembly generated using an assembler for some hash length and sizes of circles correspond to similarity with reference. The *R*^2^ values are (i) log likelihood vs similarity: 0.9372048, (ii) log likelihood vs N50 scaffold length: 0.44011, (iii) N50 scaffold length vs similarity: 0.3216882

**Figure 5 - Log likelihood vs Numbers of mis-assembly features and suspicious regions for E. coli**

Log likelihoods are shown on the *x*-axis and Numbers of mis-assembly features and suspicious regions reported by amosvalidate are shown on the *y*-axis. Each symbol corresponds to an assembly generated using an assembler for some hash length and sizes of symbols correspond to similarity with reference. The *R*^2^ values
are (i) log likelihood vs # mis-assembly features: 0.8922, (ii) log likelihood vs # suspicious regions: 0.9039, (iii) similarity vs # mis-assembly features: 0.8211, (iv) similarity vs # suspicious regions: 0.7723.

**Figure 6 - Hash length vs log likelihood for E. coli data from CLC Bio**

Log likelihoods of assemblies of *E. coli* reads from CLC bio are shown on the y-axis. Assemblies are generated using different assemblers for varying k-mer length shown on the x-axis. The yellow dotted line corresponds to the log likelihood of the reference provided and the gray dotted line corresponds to the log likelihood of the strain we believe the reads were generated from.

**Figure 7 - Hash length vs log likelihood for G. clavigera**

Log likelihoods of assemblies of *G. clavigera* reads are shown on the y-axis. Assemblies are generated using different assemblers for varying k-mer length shown on the x-axis. The dotted lines correspond to log likelihood of the assemblies generated using Sanger, 454 as well as Illumina data.

**Figure 8 - Hash length vs difference from reference for G. clavigera**

Differences between assemblies and the reference are shown on the y-axis where difference refers to numbers of bases in the reference not covered by the assembly or different in the reference and the assembly.

**Figure 9 - Log likelihood vs N50 scaffold length for G. clavigera**

Log likelihoods are shown on the x-axis and N50 scaffold lengths are shown on the y-axis. Each circle corresponds to an assembly generated using an assembler for some hash length and sizes of circles correspond to similarity with reference. The $R^2$ values are (i) log likelihood vs similarity: 0.4545793, (ii) log likelihood vs N50 scaffold length: 0.002397233, (iii) N50 scaffold length vs similarity: 0.006084032

**Figure 10 - Coverage vs Log likelihood for Assemblathon 1 entries**

Coverage is shown on the x-axis and log likelihood is shown on the y-axis. Each circle corresponds to an assembly. The $R^2$ value is 0.989972.

**Tables**

Table 1 - Percentage difference between the simulator and CGAL
| Genome             | Length(bp) | % difference |
|--------------------|------------|--------------|
| E. coli            | 4.6M       | 0.074        |
| G. clavigera       | 29.1M      | 0.0755       |

Table 2 - Likelihoods of GAGE assemblies of S. aureus

| Assembler        | Likelihood       | # reads mapped | Coverage(%) | Scaffold N50 (kb) | Contig N50 (kb) |
|------------------|------------------|----------------|-------------|------------------|-----------------|
| ABYSS            | $-23.34 \times 10^4$ | 1236230        | 99.74†      | 34               | 29.2            |
| ALLPATHS-LG      | $-24.53 \times 10^7$ | 1220328        | 99.38       | 1092             | 96.7            |
| Bambus2          | $-23.76 \times 10^7$ | 1200527        | 98.68       | 1084             | 50.2            |
| MSR-CA           | $-25.85 \times 10^7$ | 1192001        | 98.70       | 2412             | 59.2            |
| SGA              | $-26.61 \times 10^7$ | 1018936        | 98.09       | 208              | 4.0             |
| SOAPdenovo       | $-23.55 \times 10^7$ | 1212384        | 99.62       | 332              | 288.2           |
| Velvet           | $-23.28 \times 10^7$ | 1203907        | 99.21       | 762              | 48.4            |
| Reference        | $-22.38 \times 10^7$ | 1268718        | -           | -                | -               |

† Value reported in original paper is 98.63

Table 3 - Likelihoods of GAGE assemblies of R. sphaeroides

| Assembler        | Likelihood       | # reads mapped | Coverage(%) | Scaffold N50 (kb) | Contig N50 (kb) |
|------------------|------------------|----------------|-------------|------------------|-----------------|
| ABYSS            | $-27.55 \times 10^7$ | 1199197        | 99.11†      | 9                | 5.9             |
| ALLPATHS-LG      | $-26.61 \times 10^7$ | 1237938        | 99.53       | 3192             | 42.5            |
| Bambus2          | $-32.56 \times 10^7$ | 1111596        | 95.07       | 2439             | 93.2            |
| CABOG            | $-39.23 \times 10^7$ | 1022732        | 92.49       | 66               | 20.2            |
| MSR-CA           | $-31.61 \times 10^7$ | 1155078        | 96.48       | 2976             | 22.1            |
| SGA              | $-31.58 \times 10^7$ | 1031547        | 97.69       | 51               | 4.5             |
| SOAPdenovo       | $-27.67 \times 10^7$ | 1212959        | 99.12       | 660              | 131.7           |
| Velvet           | $-28.77 \times 10^7$ | 1176125        | 98.40       | 353              | 15.7            |
| Reference        | $-25.99 \times 10^7$ | 1255750        | -           | -                | -               |

† Value reported in original paper is 96.99

Table 4 - Likelihoods of GAGE assemblies of human chromosome 14

| Assembler        | Likelihood       | # reads mapped | Coverage(%) | Scaffold N50 (kb) | Contig N50 (kb) |
|------------------|------------------|----------------|-------------|------------------|-----------------|
| ABYSS            | $-23.44 \times 10^8$ | 22096466       | 82.22       | 2.1              | 2               |
| ALLPATHS-LG      | $-22.77 \times 10^8$ | 23122569       | 97.24       | 81647            | 36.5            |
| CABOG            | $-21.26 \times 10^8$ | 23433424       | 98.32       | 393              | 45.3            |
| SOAPdenovo       | *                | *              | 98.17       | 455              | 14.7            |
| Reference        | $-19.04 \times 10^8$ | 23978017       | -           | -                | -               |

* Likelihood not computed as reads could not be mapped with Bowtie 2

Table 5 - Likelihoods of GAGE assemblies of bumble bee, B. impateins
| Assembler   | Likelihood  | # reads mapped | Scaffold N50 (kb) | Contig N50 (kb) |
|------------|-------------|----------------|-------------------|-----------------|
| ABysS      | $-30.83 \times 10^8$ | 72629126        | -                 | -               |
| CABOG      | $-19.99 \times 10^8$ | 92844610        | 1125              | 23.5            |
| MSR-CA     | $-22.84 \times 10^8$ | 78755756        | 1246              | 32.4            |
| SOAPdenovo | *           | *              | 1374              | 57.1            |

* Likelihood not computed as reads could not be mapped with Bowtie 2

Table 6 - Likelihoods of Assemblathon 1 assemblies

| Assembler   | Likelihood  | #reads mapped | Assemblathon 1 rank |
|------------|-------------|----------------|---------------------|
| BGI 1      | $-20.17 \times 10^8$ | 42005212        | 2                   |
| CSHL 2     | $-20.19 \times 10^8$ | 41973576        | 5                   |
| BCCGSC 5   | $-20.23 \times 10^8$ | 41891758        | 7                   |
| IoBUGA 2   | $-20.49 \times 10^8$ | 41931526        | 9                   |
| RHUL 3     | $-20.69 \times 10^8$ | 41753084        | 10                  |
| DOEJGI 1   | $-20.73 \times 10^8$ | 41836210        | 4                   |
| WTSI-P 2   | $-20.81 \times 10^8$ | 41748504        | 11                  |
| Broad 1    | $-21.75 \times 10^8$ | 41778343        | 1                   |
| EBI 1      | $-21.83 \times 10^8$ | 41377165        | 8                   |
| WTSI-S 4   | $-30.81 \times 10^8$ | 37442672        | 3                   |

Additional Files
Additional file 1 — Supplementary information for computing genome assembly likelihoods

Additional figures, tables and information to supplement the text.
Figure 2

- Euler
- Abyss
- Velvet
- SOAP
- U00096.2
Figure 3
Figure 4
Figure 5
Figure 6
Figure 7

Difference with reference

- Abyss
- Velvet
- SOAP
Figure 8
Additional files provided with this submission:

Additional file 1: Additional file 1.pdf, 166K
http://genomebiology.com/imedia/1735080950884838/supp1.pdf