De novo bacterial genome sequencing: Millions of very short reads assembled on a desktop computer

David Hernandez,[1,3] Patrice François,[1] Laurent Farinelli,[2] Magne Østerås,[2] and Jacques Schrenzel[1]

[1]*Genomic Research Laboratory, Infectious Diseases Service, Geneva University Hospitals and the University of Geneva, CH-1211 Geneva 4, Switzerland;* [2]*Fasteris SA, CH-1228 Plan-les-Ouates, Switzerland*

Novel high-throughput DNA sequencing technologies allow researchers to characterize a bacterial genome during a single experiment and at a moderate cost. However, the increase in sequencing throughput that is allowed by using such platforms is obtained at the expense of individual sequence read length, which must be assembled into longer contigs to be exploitable. This study focuses on the Illumina sequencing platform that produces millions of very short sequences that are 35 bases in length. We propose a de novo assembler software that is dedicated to process such data. Based on a classical overlap graph representation and on the detection of potentially spurious reads, our software generates a set of accurate contigs of several kilobases that cover most of the bacterial genome. The assembly results were validated by comparing data sets that were obtained experimentally for *Staphylococcus aureus* strain MW2 and *Helicobacter acinonychis* strain Sheeba with that of their published genomes acquired by conventional sequencing of 1.5- to 3.0-kb fragments. We also provide indications that the broad coverage achieved by high-throughput sequencing might allow for the detection of clonal polymorphisms in the set of DNA molecules being sequenced.

[Supplemental material is available online at www.genome.org. Edena is freely available for academic users at http://www.genomic.ch/edena.]

High-throughput sequencing technologies have the potential to decipher a bacterial genome during a single experiment and at a moderate cost (Mitra and Church 1999; Brenner et al. 2000; Margulies et al. 2005; Bentley 2006). However, such broad coverage is typically obtained at the expense of the read length. This article addresses the assembly of the sequences produced by the Illumina Genome Analyzer that generates millions of very short reads of the same length. The technology currently provides reads of 35 bases, although future technological improvements promise to increase the sequence length to 50 bases. The ability to efficiently sequence and assemble whole bacterial genomes has significant implications for evolutionary (Smith et al. 2006; Mwangi et al. 2007), metagenomic (Handelsman et al. 1998; Eisen 2007), and even diagnostic purposes (Fournier et al. 2006; Audic et al. 2007). However, the fact that only short overlaps can be considered represents a challenge for the assembly process. It was previously shown that de novo sequencing of a bacterial genome is possible using error-free reads that range from 20–50 bases in length (Whiteford et al. 2005). Recently, software applications that are dedicated to the assembly of very short reads were published. SSAKE was originally described by Warren et al. (2007), but its current version (3.0) has been recently described in a poster in the Pacific Symposium on Biocomputing (Hawaii 2008). Rather than claiming the production of accurate contigs, the investigators demonstrated that their approach identified bacteria from a complex soil metagenomic sample. Velvet was presented at the Cold Spring Harbor Biology of Genomes meeting in 2007. This application is based on a *k*-mer graph representation (Idury and Waterman 1995; Pevzner et al. 2001) that structures all *k*-mers observed in the reads. Finally, SHARCGS (Dohm et al. 2007) is an assembler that prefilters the reads according to their quality values and to their redundancy in the data set. In this study, we present Edena (Exact DE Novo Assembler), a novel software for de novo assembly of accurate contigs from data sets containing very short reads of the same length. The application is based on the classical assembly approach where all overlaps are computed and structured in a graph. Accurate contigs of several kilobases are produced that cover most of the genome being sequenced. A comparison with previously described assembly programs was performed by analyzing two different bacterial genomes sequenced on an Illumina Genome Analyzer. Finally, the broad coverage depth achieved by the new generation sequencing device suggests the presence of clonal polymorphisms in the set of DNA molecules being sequenced, but this statement needs to be formally proven.

[3]Corresponding author.
E-mail david.hernandez@genomic.ch; fax 41-22-372-9830.
Article published online before print. Article and publication date are at http://www.genome.org/cgi/doi/10.1101/gr.072033.107.

Results

Assembly of the *Staphylococcus aureus* strain MW2 genome

The *Staphylococcus aureus* strain MW2 data set is made up of 3.86 million of 35-bp reads among which 3.83 million are unambiguous (i.e., they do not contain any nondetermined nucleotide). The redundancy filter applied by Edena keeps 2.66 million unique reads. The raw coverage depth is therefore 48×. All programs were tested with several parameterizations, and only the

18:802–809 ©2008 by Cold Spring Harbor Laboratory Press; ISSN 1088-9051/08; www.genome.org

best one was considered. Due to the higher computational resources required by SSAKE and particularly by SHARCGS, these two programs were not extensively tested. However, efforts were made to optimize their results. Edena was parameterized to consider overlaps displaying a minimum length of 21 bases for the strict and nonstrict mode. Velvet was used with a *k*-mer value of 23. The best result for SHARCGS was achieved by setting the "max gap span" parameter to 14. Also, this program was provided with sequencing quality values. The best result for SSAKE was achieved with its default parameters. Since *S. aureus* strain MW2 was already sequenced and assembled by using conventional methods (Baba et al. 2002), we used the published sequence as the reference to evaluate the accuracy of the assemblies. The reference genomic and plasmid sequences measure 2.82 Mbp and 20.7 kbp, respectively (accession nos. NC_003923 and NC_005011, respectively). Contigs were aligned to their reference genomic and plasmid sequences using the Exonerate sequence alignment package (Slater and Birney 2005). To be considered valid, a contig must be aligned along its whole length with a base similarity of at least 98%. Only contigs larger than, or equal to, 100 bases were considered in this study. A graphical view of the contig map is shown in Figure 1, and the assembly results are summarized in Table 1. The contigs produced by Edena running in strict mode covered 98% of the reference sequences, and the N50 value is 6.0 kb. The nonstrict mode of Edena produced a few misassembled contigs. These misassemblies included 14 contigs totalizing 45.1 kb. In terms of contig length, the performance of Velvet was similar to that of the strict mode of Edena, but two contigs did not properly map the reference sequence. In addition, the correct contigs presented a total of 260 mismatches. All mismatches were located almost exclusively at the ends of the contigs. SHARCGS was not able to assemble significant contigs. This is probably due to the fact that this program relies mainly on its prefiltering step, which requires a very broad coverage depth to retain a sufficient number of correct reads. SSAKE generated numerous errors that were mainly located at the ends of the contigs. As Figure 1 revealed significant overlaps between contigs produced by Edena and Velvet, we constituted two additional data sets by merging the contigs generated by the two programs. The first data set was constituted from the results of Velvet and Edena when operated in the strict mode, while the second one combined the results of Edena in the nonstrict mode with those of Velvet. These two new data sets were then assembled with the Minimus assembler (Sommer et al. 2007), as shown in Table 2. By assembling the contigs produced by Velvet and Edena in the strict mode, N50 value and mean contigs size increased to 8.1 kbp and 3.6 kbp, respectively. By assembling the contigs produced by Velvet and Edena in the nonstrict mode, longer contigs were obtained but at the expense of 16 misassembled contigs representing a total of 78.7 kbp.

Assembly of the *Helicobacter acinonychis* strain Sheeba genome

This assembly comparison was performed on the set of reads obtained with *Helicobacter acinonychis* strain Sheeba and originally presented in the publication describing SHARCGS. This information is freely available at http://sharcgs.molgen.mpg.de/download.shtml. This data set is made up of 12.3 million of 36-bp reads, among which 11.6 million are unambiguous. The raw coverage depth is therefore 284×. The redundancy filter applied by Edena keeps 7.3 million unique reads. Edena was parameterized to consider overlaps displaying a minimum length of 27 bp for the strict mode and 26 bp for the nonstrict mode. Velvet has been used with a *k*-mer value of 27. The best result for SHARCGS was achieved by setting the "max gap span" parameter to 10 and by removing the last four bases of each reads. Also, this program was provided with sequencing quality values, as mentioned above. For SSAKE, the best result was achieved with its default parameters. Following the same procedure as described for the *S. aureus* assembly, contigs were evaluated against the published whole-genome sequence (Eppinger et al. 2006), which measures 1.55 Mbp and 3.66 kbp for the genome and plasmid, respectively (accession nos. NC_008229 and NC_008230, respectively). Assembly results are shown in Table 3. Once again, Edena and Velvet showed the best performance by reaching a N50 value of 10.4 kbp and 9.8 kbp, respectively. The nonstrict mode of Edena reached a N50 value of 14.2 kbp for the correctly assembled contigs. However, one contig of 24.1 kbp was misassembled.

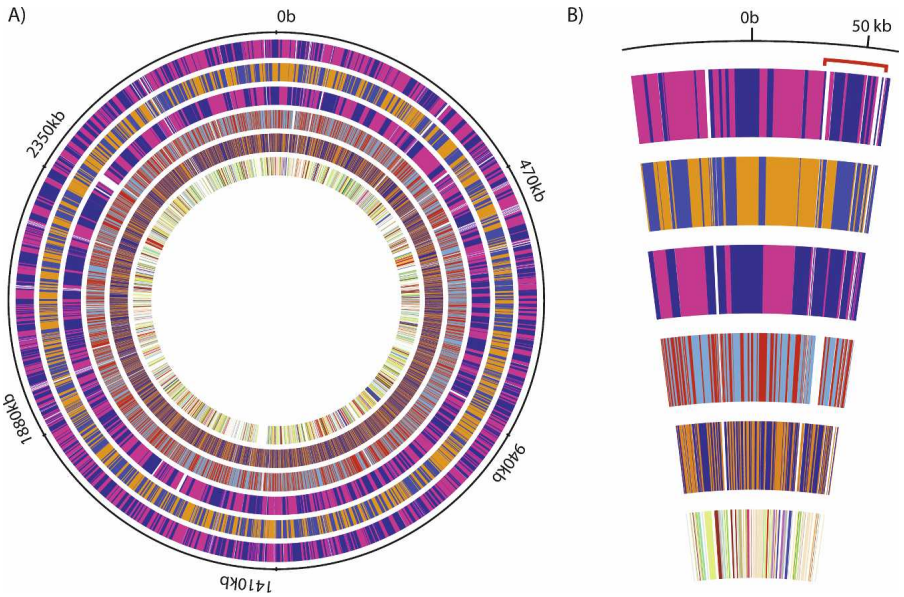

Figure 1. Mapping of the contigs on the reference *Staphylococcus aureus* MW2 genome. (*A*) From external to internal, the circles correspond to the contigs produced by (1) Edena strict, (2) Velvet, (3) Edena nonstrict, (4) SSAKE, and (5) SHARCGS. The contigs are colored by alternating two different colors, which allows distinguishing contig boundaries. The last inner circle shows the coding sequences. The gaps in the Edena nonstrict assembly correspond to large misassembled contigs that did not properly map the reference genome. (*B*) The magnification of the region around the origin of replication provides a better view to compare the contigs length and layout between the different assembly methods. It can be seen that the contigs assembled by Edena and Velvet are long enough to reveal entire genes. More importantly, significant overlaps exist between the contigs assembled by the two programs, which also means that even larger contigs could be assembled by merging both approaches. The position of the SSC*mec* cassette of type IV.1 (Chongtrakool et al. 2006) is indicated by the red line.

Table 1. Comparison of assembly results of *Staphylococcus aureus* strain MW2 as obtained by Edena, Velvet, SHARCGS, and SSAKE

Assembly software	No. of correct contigs (total size)	No. of misassembled contigs (total size)	Correct contigs				
			N50	Average length	Max length	Total no. of mismatches	Genome coverage
Edena strict	1122 (2762 kbp)	0 (0 bp)	6.0 kbp	2.5 kbp	25.7 kbp	1	98%
Edena nonstrict	733 (2737 kbp)	14 (45.1 kbp)	9.4 kbp	3.7 kbp	51.8 kbp	90	97%
Velvet	1093 (2768 kbp)	2 (362 bp)	5.4 kbp	2.5 kbp	22.9 kbp	260	98%
SSAKE	2334 (2782 kbp)	99 (85.1 kbp)	2.0 kbp	1.2 kbp	12.6 kbp	2427	97%
SHARCGS	3632 (2760 kbp)	3 (1.5 kbp)	1.2 kbp	760 bp	8.6 kbp	44	97%

Erroneous reads and putative clonal polymorphisms in the DNA samples being sequenced

Edena relies on two cleaning operations to remove ambiguous paths from the overlap graph: the short dead-end (DE) path removal and the p-bubble fixing (see Methods). By using the overlap graph obtained from the *S. aureus* data set with a minimum overlap length of 21, we investigated the efficiency of these cleaning operations. By considering the nonredundant data set, it appears that 39.3% of the reads do not have an exact occurrence on either the genomic or the plasmid reference sequences (which correspond to 28% of the complete redundant data set). We will refer to these reads as the negative read, while the positive read will refer to the remaining reads that have at least one exact occurrence in the reference sequences. We generated a simulated ideal data set by sampling a 35-bp read at every position in the genomic and plasmid reference sequence for *S. aureus* strain MW2. Thus, this ideal data set does not contain any error or polymorphism. It allowed us to compute the number of branching nodes and p-bubbles that are exclusively caused by exact and nonexact repetitions in the genomic sequence and to compare these values with those obtained from the real data set. Results are presented in Table 4. The difference between the two overlap graphs is significant. First, the proportions of branching nodes in the real and ideal data set are 32% and 0.2%, respectively; the later being exclusively due to genomic repetitions that cannot be resolved by the overlapping procedure. This indicates that almost all branching nodes in the real data set result from negative reads. The number of p-bubbles is 12 in the ideal overlap graph, which indicates the presence of nonexact repetitions in the genomic and/or plasmid sequences. This number however rises to 526 in the real overlap graph. For each bubble, we extracted the corresponding sequences. It appears that for 521 of the 526 bubbles, one of the two sequences does not show any exact occurrence in the reference sequences, this sequence being assembled from the negative reads. Moreover, the coverage depth ratio of the two possible paths showed an average of 0.1, the lower path corresponding to the one being made of negative reads. Such bubbles involving negative reads cannot be explained by base calling errors. A possible explanation is that these negative reads are issued from underrepresented subsets of DNA molecules in the sample; each of these subsets containing one or more mutations as compared to the majority sequence. The broad coverage achieved by the Illumina Genome Analyzer might thus allow for the detection of clonal polymorphisms.

The percentage of nodes that were removed by the DE path cleaning is 33% for the real data set versus 0% for the ideal set. Table 5 gives the number of positive and negative nodes that are removed by the DE procedure according to the *md* value (see Methods). As expected, almost all removed nodes belong to the set of negative reads. The value of *md* = 10 is a relevant choice to remove most of the negative nodes, yet this value limited the loss of the positive nodes. Most of the DE paths have a depth of 1, which corresponds to what is expected from random base calling errors. However, a significant number of longer DE paths that cannot be explained by random errors also exist in the overlap graph. The overwhelming majority of these unexpected long DE paths only involve negative nodes. This observation suggests that they might be caused by clonal polymorphisms on DNA molecule for which the abundance is insufficient to constitute a complete p-bubble.

Lander-Waterman statistics and genome coverage depth

As discussed above, the raw coverage depth of the *S. aureus* Illumina sequencing is 48×. However, since the required overlapping length represents a significant part of the read length, the effective coverage depth (Lander and Waterman 1988; Wendl and Waterston 2002) provides a more informative value. Effective coverage can be estimated by $E = N(L - T)/G$, and the expected number of gaps (i.e., region not properly represented by the reads) can be estimated by Ne^{-E} where N is the number of usable reads, L is the length of the reads, T is the required overlap length, and G is the target size. For this purpose, usable reads are defined as those having at least one overlap on each end. The number of usable reads is 2,900,674. By considering a required overlap length of 21 bases, the effective coverage depth is 14× and the expected number of gaps is two. Similarly, the number of usable reads is 6,258,923 for the *H. acinonychis* Illumina sequencing. By considering a required overlap length of 27 bases, the effective coverage depth is 36× and no gap is expected. Thus, by assuming that the genomes are uniformly sampled by

Table 2. Merging the results of Velvet and Edena using the Minimus assembler

Assembly software	No. of correct contigs (total size)	No. of misassembled contigs (total size)	Correct contigs				
			N50	Average length	Max length	Total no. of mismatches	Genome coverage
Velvet + Edena strict	779 (2771 kbp)	1 (154 bp)	8.1 kbp	3.6 kbp	40.4 kbp	176	98%
Velvet + Edena nonstrict	469 (2701 kbp)	16 (78.7 kbp)	12.6 kbp	5.8 kbp	69.2 kbp	278	96%

Table 3. Comparison of assembly results of *Helicobacter acininychis* strain Sheeba as obtained by Edena, Velvet, SHARCGS, and SSAKE

Assembly software	No. of correct contigs (total size)	No. of misassembled contigs (total size)	Correct contigs				
			N50	Average length	Max length	Total no. of mismatches	Genome coverage
Edena strict	336 (1525 kbp)	0 (0 bp)	10.4 kbp	4.5 kbp	37.0 kbp	0	99%
Edena nonstrict	302 (1504 kbp)	1 (24.1 kbp)	14.2 kbp	5.0 kbp	35.0 kbp	6	98%
Velvet	340 (1525 kbp)	0 (0 bp)	9.8 kbp	4.5 kbp	36.4 kbp	90	99%
SSAKE	1368 (1551 kbp)	78 (44.9 kbp)	1.9 kbp	1.1 kbp	8.6 kbp	1626	97%
SHARCGS	628 (1523 kbp)	0 (0 bp)	4.6 kbp	2.4 kbp	19.2 kbp	4	99%

the Illumina reads, this statistic predicts that no contig should end in a gap for the *H. acinonychis* assembly and only a few ones for the *S. aureus* assembly. However, we observed that among the 1122 contigs produced by Edena for the *S. aureus* genome assembly, 879 contig ends ended in a gap (i.e., could not be elongated due to the lack of an overlapping read). Also, among the 336 contigs produced for the *H. acinonychis assembly*, 192 contig ends ended in a gap. These values are significantly higher than what is expected given the coverage depth of both projects. Although the overall assembly results indicated that the genomes are roughly uniformly represented in the Illumina reads, some particular regions of the genomes are clearly weakly represented. Therefore, we mapped the reads of both projects against their respective reference sequence. Only exact matches were considered. Then, we extracted the parts of genome that were not sufficiently covered to be assembled. A simple visual inspection reveals that these extracted sequences contain large complexity moieties as well as long stretches of single base repeats that are likely to form secondary structures or that are not efficiently replicated during enzymatic steps.

Discussion

High-throughput sequencing technologies characterize bacterial genome sequences containing millions of nucleotides during a single experiment and within a few hours of "machine working time." If prices of genome sequencing are still higher than the target price of $1000 (Service 2006), the time required to sequence a bacterial genome has dramatically reduced in a few years. Previous technologies required several years of sequencing and assembly effort (Fleischmann et al. 1995). However, recent improvements in sequencing technology have decreased this requirement to only days or weeks of work. This revolution in sequencing contributes to the current situation: Today, more than 400 bacterial genomes are publicly available in databases, and ~600 projects are ongoing (http://cmr.tigr.org/tigr-scripts/CMR/CmrHomePage.cgi and http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi).

Two major high-throughput sequencing strategies are currently available: the 454 from Roche Diagnostics and Illumina's Solexa Sequencing Technology, which was used in this work. The Illumina platform relies on millions of small reads, ensuring that each nucleotide of a genome is sequenced by dozens of small reads, whereas the 454 generates larger fragments that average 230 nucleotides. Based on these characteristics, the 454 is probably more adapted to genomes containing abundant repeated regions. High-quality sequencing data are, of course, mandatory for such projects, and the frequency of errors arising from high-throughput sequencers appears reasonable (Sundquist et al. 2007). However, the limited length of sequenced elements requires elaborate assembly strategies that often require sophisticated hardware resources. Recently, Keane and Ning (2007) presented some assembly experiments using the Phusion assembler (Mullikin and Ning 2003). They reported a valuable result for the assembly of the genome of *Streptococcus suis* strain SC84 from 2 million of 41-bp reads. Eighty-one percent of the *S. suis* genome, which contains 2,007,491 bp, was assembled in 515 contigs. This result is remarkable considering that the Phusion assembler is not specifically dedicated to such short reads sequencing technology. However, a specific approach is required for assembling millions of very short sequences. Both Edena and Velvet showed that accurate contigs that nearly cover the entire bacterial genome being sequenced can be produced on simple desktop computers. These two programs outperform the others, both in terms of assembly quality and required computer resources. The length of the assembled contigs averages several kilobases, which makes them usable for numerous analyses, such as database search, gene detection, or the study of promoter sequences. The Velvet assembler that is based on a *k*-mer graph representation shows similar performances as Edena, which implements the classical overlap layout approach. However, it is interesting to notice that the two programs do not always have similar problems within regions of sequences that are difficult to assemble. Some regions that are not assembled by one of the approaches are successfully assembled by the other. Thus, the two programs are partially complementary, and their combined usage can even lead to longer contigs, as illustrated here (Table 2). The announced availability of the paired reads data in the near future will certainly allow for the production of even larger contigs. This precious information can be used to safely clean up some ambiguities in the overlap graph, thereby increasing the overall assembly performance.

An important alternative to the de novo assembly is the so-called comparative assembly (Pop et al. 2004). The latter relies

Table 4. Overlap graph properties of the *S. aureus* real data set and simulated ideal data set

	No. of nodes	Percentage of positive node	No. of branching nodes	No. of p-bubbles	No. of nodes removed by DE
Real data set	2,662,170	61%	842,756 (32%)	526	872,298 (33%)
Ideal data set	2,800,594	100%	4623 (0.2%)	12	0

Table 5. Number of positive and negative nodes that are removed by the DE procedure according to the *md* value

	md value											
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
Positive nodes	57	240	460	746	1044	1452	1825	2307	2761	3462	4129	4891
Negative nodes	674,106	816,374	852,786	863,225	866,523	867,831	868,318	868,557	868,709	868,765	868,795	868,815

on a reference sequence that must be closely related to the target being assembled. But even though the number of complete genome sequences is growing rapidly, the paucity of sequence information for some bacterial species will likely remain a problem for years to come. The development of efficient de novo assemblers thus remains an important endeavor for the efficient assembly and analysis of newly sequenced genomes. Furthermore, de novo sequence assembly permits the study of regions where comparison with the reference is not possible due to rearrangements or, in the case of transcriptomes, due to splicing.

Methods

Edena is based on the classical overlap layout assembly framework (Pop et al. 2002). In addition, it includes two features to improve the assembly of very short sequences: exact matching and detection of spurious reads. The exact matching choice was included for two reasons. First, the inherent sequencing errors result in a significant number of spurious overlaps, impairing the correct sequence determination. Allowing approximate matching would significantly increase the number of such nonspecific spurious overlaps. Second, exact matching is drastically faster than approximate matching. By using an appropriate index, overlaps between millions of short reads can be computed in a few minutes.

The key steps of Edena can be summarized as follows. First, the short reads data set is processed to remove redundant information. Second, all overlaps of a minimum size are computed, and an overlap graph is constructed. Third, the graph is cleaned by removing transitive and spurious edges and by resolving bubbles. Finally, all contigs of a minimum size that are unambiguously represented in the graph are provided as an output. The program assumes that all reads have the same length.

Reducing reads redundancy

Due to the high level of oversampling achieved by the Illumina Genome Analyzer, a significant number of reads are represented several times in the data set. We first process the data set in order to keep a single copy of each read. This step reduces the size of the data set without losing information. It is achieved by indexing all reads in a prefix tree. A given read and its reverse complement are considered to be the same read and are merged in the same tree key. Reads that contain ambiguous base symbols are discarded since they cannot be handled in the exact matching procedure. Since identical reads are merged in the same tree key, a nonredundant set of reads can be produced from the tree structure. The occurrence frequency of each read as observed in the initial data set is kept in order to compute the coverage depth in the contigs for quality control purposes.

Overlapping phase

The overlapping phase is performed by indexing the nonredundant read data set by a suffix array (Manber and Myers 1993). This structure reveals exact matches, i.e., exact overlaps, at a low memory cost. The set of revealed overlaps is loaded in a bidirected graph structure (Kececioglu and Myers 1995; Myers 2005) where each read r_i corresponds to a vertex v_i. Two vertices v_i and v_j are connected by a bidirected edge if r_i and r_j overlap. Bidirected edges have an arrowhead at each end and can independently be directed in or out of the vertex at each end of the edge. Subsequently, there are four different ways to connect two nodes, depending on the relative orientation of v_i and v_j, and the sides of the reads that are involved in the overlap. The arrowhead is directed in v_i or v_j if the overlap implies the left end of r_i or r_j, and out of v_i or v_j if it implies the right end of r_i or r_j. Edges are labeled with the overlap size. In order to build a valid read assembly, vertices must be traversed using two opposed arrowhead orientations. Entering v_i from an in-arrowhead and leaving it to an out-arrowhead indicates that the corresponding read is spelled in its direct strand, while traversing v_i from an out-arrowhead to an in-arrowhead indicates that it is spelled in the reverse direction.

The minimum overlap size is a determinant parameter for the assembly success. A small value will increase the frequency of overlaps that exist by chance, which creates significant branching in the graph. On the other hand, a large value will increase the number of reads that do not overlap on one of their sides, which leads to DE paths in the graph.

Removing transitive edges

Due to the high oversampling achieved by the Illumina sequencing technology, the great majority of edges in the overlaps graph correspond to transitive edges. These edges are not essential to represent every possible sequence in the graph. For example, consider two paths $v_1 \rightarrow v_2 \rightarrow v_3$ and $v_1 \rightarrow v_3$. The path $v_1 \rightarrow v_3$ is transitive because it bypasses v_2 and represents the same sequence as the first path. This is illustrated from the point of view of a multiple alignment in Figure 2. Transitive edge removal is an

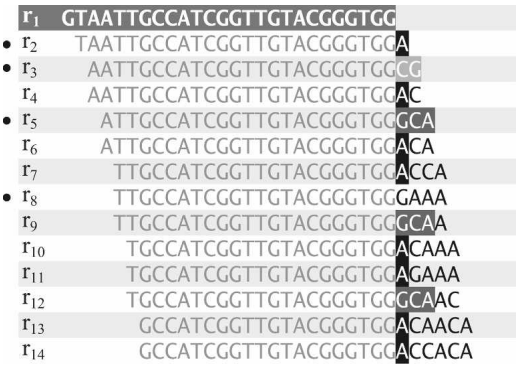

Figure 2. Removing transitive edges. A read r_1 with 13 other reads ($r_2 \ldots r_{14}$) that overlap on its right end side are shown in the form of a multiple alignment. The overlaps that do not correspond to transitive edges are indicated with a black dot. The transitive edges removal procedure consists in discarding the overlaps that are already overlapped by another read involved in a larger overlap with r_1. For example, the reads r_4, r_6, r_7, r_{10}, r_{11}, r_{13}, and r_{14} are overlapped by r_2; they are therefore removed from the set of overlapping reads of r_1. Same principle is applied to the reads r_3, r_5, and r_8. This example is issued from a real data set of reads of 26 bases.

essential procedure that reduces the graph complexity by a factor of the oversampling rate c calculated as NL/G, where N is the number of reads, G is the size of the genome being sequenced, and L the length of the reads (Myers 2005).

Cleaning up the graph

The transitively reduced overlap graph contains a significant amount of branching paths that compromise the production of long contigs. These branching paths are caused by genomic repetitions, sequencing errors, and clonal polymorphisms (see Results). Without additional information, branching paths caused by genomic repetitions cannot be fixed. However, we propose a simple and efficient method to fix the latter two problems. This step is the key to the success of the Edena approach. Base calling errors in reads cause short DE paths, while clonal polymorphisms create small bubbles in the overlap graph. The cleaning operations identify such features by a local graph exploration starting at each branching node. The first cleaning operation removes the nodes that are involved in short DE paths (Fig. 3). The underlying idea is that edge leading to a read that contains a sequencing error should rapidly reach a DE. Each branching node is thus explored for all possible path elongations up to a depth of *md* nodes. If no path of depth of *md* exists, the nodes are marked for removal. Once all DE paths have been detected, marked nodes are removed. The *md* value is the cutoff above which a branching path is considered to be valid. We determined that a value of $md = 10$ was a good compromise (see Results). The second cleaning operation identifies short bubbles in the graph (Fig. 4). Such bubbles can be caused by nonexact repetitions in the genomic sequence. However, most of these bubbles could be caused by single base substitutions carried by a subset of DNA molecules contained in the analyzed sample (see Results). We use the term of p-bubble to refer to bubbles that are caused by a single base substitution. In other words, a p-bubble represents a base alternative in the assembly. The length of a p-bubble is at most $ms = 4 \times L - 2 \times T - 1$, with L and T being the read length and the minimum required overlap size, respectively. Each branching path is explored up to a length *ms*. Detected p-bubbles are resolved by removing the nodes in its less covered side. The p-bubble is therefore resolved in a simple non-intersecting path corresponding to the most covered path. Despite the fact that the polymorphisms are not represented in the final assembly, this information can be kept in a separated file.

Strict and nonstrict assembly modes

An additional cleaning operation significantly increased the size of the contigs but also generated a few misassemblies. This cleaning operation corresponds to the "nonstrict" mode that is implemented in Edena. It is based on the fact that longer overlaps are more reliable than shorter ones. Each branching node is examined, and only the edge (or edges) maximizing the overlap value is (are) kept. This operation allows cleaning ambiguities when the edge corresponding to the maximum length overlap is unique. Once all graph cleaning operations are finished, the set of contigs is produced by spelling the sequences modeled by the non-intersecting simple paths, for which only nodes having in- and out-degree of exactly one are traversed.

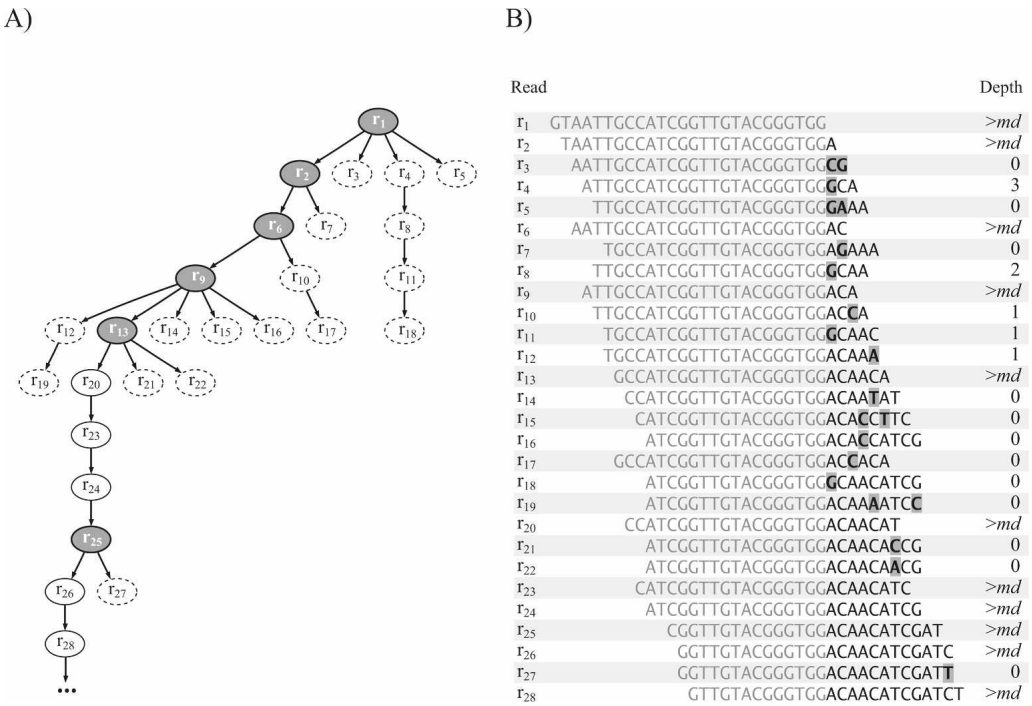

Figure 3. Removing short dead-end paths. (*A*) Possible path elongations from the right end of the read r_1 are represented by a tree. Nodes that are removed are dashed. Each path leaving a branching node (shown in gray) is tested for the minimum depth it can initiate. If the required depth of *md* cannot be reached, then the nodes forming the dead-end path are removed. (*B*) Multiple sequence alignment of the reads belonging to the possible right end elongation of the read r_1 is shown. The residues that do not agree with the consensus sequence are shaded. On the *right* side is indicated the depth value that can be reached by continuing the elongation from the corresponding read. The reads containing one or more mismatched residues have a low or a null depth value, indicating that no exact overlap exists for their right end in the entire reads data set. These reads are likely to contain sequencing errors.

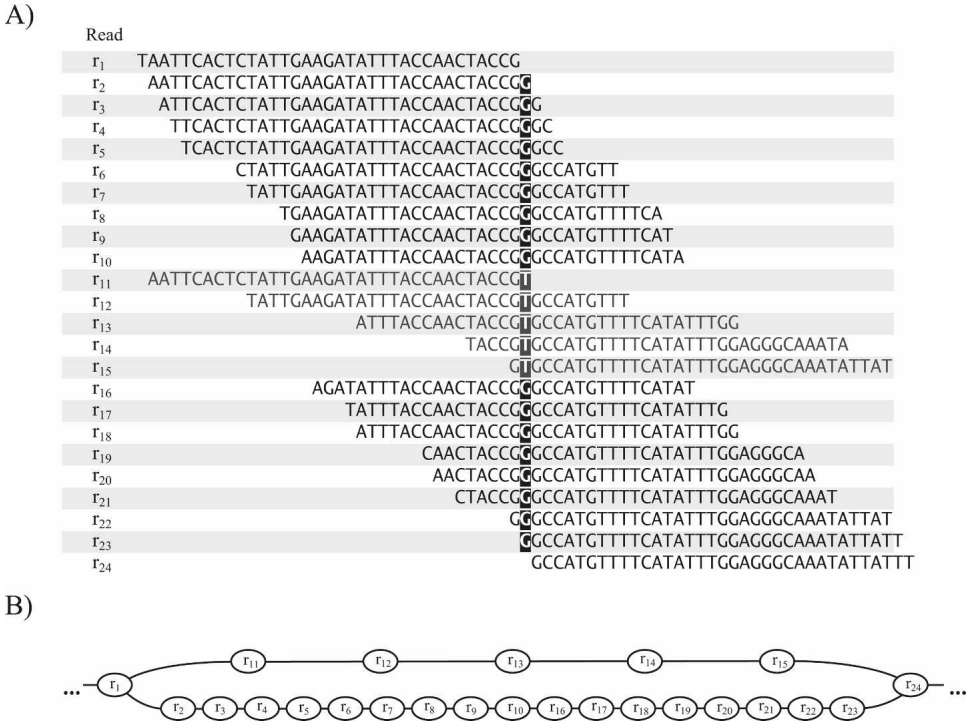

Figure 4. Fixing bubbles. This illustration shows a bubble caused by a polymorphism. This example is one of the many that can be found in the overlap graph constructed from the *Staphylococcus aureus* strain MW2 reads data set. (*A*) The 24 reads implicated in the bubble are shown. r_1 and r_{24} are the ends of the bubble, which is $35 \times 2 + 1$ bp in length. Reads showing the polymorphism are r_{11} to r_{15}. None of these reads have exact occurrence in the published genome of *S. aureus* strain MW2 sequence. (*B*) The corresponding transitively reduced overlap graph is shown. By considering the read redundancy, the total number of reads in the low and highly covered side is five and 27, respectively. Fixing of bubbles consists in removing nodes forming the less covered side of the bubble.

Strain and culture conditions

S. aureus strain MW2 was obtained from NARSA (http://www.narsa.net/) and grown in Mueller Hinton Broth (10 mL) for 5 h. Bacterial cells were rinsed twice in 10 mL TE (Tris-EDTA, 10 mM and 1 mM, respectively), suspended in 2 mL of TE containing 100 µg/mL lysostaphin (Ambicin, Applied Microbiology Inc.), and incubated for 10 min at 37°C. DNA was then extracted and purified according to the DNeasy kit (Qiagen). DNA purity and quantity were assessed using NanoDrop-1000.

Whole-genome sequencing with the Illumina Genome Analyzer technology

The genomic DNA of *S. aureus* strain MW2 was sequenced using the Solexa technology (P. Mayer, L. Farinelli, and E. Kawashima, 1997. Patent application WO98/44151) according to the manufacturer's protocol (Illumina). Briefly, 5 mg of genomic DNA was physically fragmented by nebulization into 50- to 100-bp fragments. After end-repair and ligation of the adaptors, the products were purified on agarose gel to recover 150- to 250-bp products. Quality control was performed by cloning the library into a TOPO plasmid and capillary sequencing of a few clones. The samples were then used to generate DNA colonies (or DNA clusters) using two channels of a flow-cell at dilutions of 4 or 6 pM, respectively. The flow-cell was then submitted to 36 cycles of sequencing reaction on the Illumina Genome Analyzer (Illumina). Data were analyzed using the Solexa Data Analysis Pipeline v0.2.2.5 software, and after quality filtration using standard parameters, we obtained a total of 3.86 million reads that were 35 bases in length.

Computer resources and software versions

The programs used in the assembly comparisons are Velvet 0.4 (http://www.ebi.ac.uk/~zerbino/velvet/), SSAKE 3.0 (http://www.bcgsc.ca/bioinfo/software/ssake), SHARCGS 1.2.11 (http://sharcgs.molgen.mpg.de/download.shtml), and Edena 2.0 (www.genomic.ch/edena). Edena, Velvet, and SSAKE were run on an Intel Pentium D CPU 2.8-GHz computer supplied with 4.0 Gb of RAM. SHARCGS was run on an AMD Opteron CPU 2.4 GHz supplied with 64 Gb of RAM. Edena performed the *H. acinonychis* assembly in less than 20 min and required 1.5 Gb of RAM. It performed the assembly of *S. aureus* in 10 min and required 850 Mb of RAM. Velvet performed the *H. acinonychis* assembly in 11 min and required 1.2 Gb of RAM. It performed the *S. aureus* assembly in 5 min and required 400 Mb of RAM. SSAKE required 2.5 Go of memory and 16 h for the *H. acinonychis* assembly and 1.1 Go of memory during 90 min for the *S. aureus* genome assembly. SHARCGS required 50 Gb of memory and 8 h to compute the *H. acinonychis* assembly and 20 Go of memory during 17 h to complete the *S. aureus* genome assembly.

Acknowledgments

We thank Chris Kolbert and Daniel DiCenso for the careful reading of this manuscript. We also thank Roberto Fabbretti and Ioannis Xenarios for providing fast access and efficient support to the Vital-IT computer resources (http://www.vital-it.ch). This work was supported by grants from the Swiss National Science Foundation nos. PP00B–103002/1 and 3100A0-112370/1 (J.S.) and no. 3100A0-116075/1 (P.F.). D.H. was supported by the COST B28 program.

References

Audic, S., Robert, C., Campagna, B., Parinello, H., Claverie, J.M., Raoult, D., and Drancourt, M. 2007. Genome analysis of *Minibacterium massiliensis* highlights the convergent evolution of water-living bacteria. *PLoS Genet.* **3:** e138. doi: 10.1371/journal.pgen.0030138.

Baba, T., Takeuchi, F., Kuroda, M., Yuzawa, H., Aoki, K., Oguchi, A., Nagai, Y., Iwama, N., Asano, K., Naimi, T., et al. 2002. Genome and virulence determinants of high virulence community-acquired MRSA. *Lancet* **359:** 1819–1827.

Bentley, D.R. 2006. Whole-genome re-sequencing. *Curr. Opin. Genet. Dev.* **16:** 545–552.

Brenner, S., Johnson, M., Bridgham, J., Golda, G., Lloyd, D.H., Johnson, D., Luo, S.J., McCurdy, S., Foy, M., Ewan, M., et al. 2000. Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nat. Biotechnol.* **18:** 630–634.

Chongtrakool, P., Ito, T., Ma, X.X., Kondo, Y., Trakulsomboon, S., Tiensasitorn, C., Chavalit, T., Song, J.H., and Hiramatsu, K. 2006. Staphylococcal cassette chromosome *mec* (SCC*mec*) typing of methicillin-resistant *Staphylococcus aureus* strains isolated in 11 Asian countries: A proposal for a new nomenclature for SCCmec elements. *Antimicrob. Agents Chemother.* **50:** 1001–1012.

Dohm, J.C., Lottaz, C., Borodina, T., and Himmelbauer, H. 2007. SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Res.* **17:** 1697–1706.

Eisen, J.A. 2007. Environmental shotgun sequencing: Its potential and challenges for studying the hidden world of microbes. *PLoS Biol.* **5:** e82. doi: 10.1371/journal.pbio.0050082.

Eppinger, M., Baar, C., Linz, B., Raddatz, G., Lanz, C., Keller, H., Morelli, G., Gressmann, H., Achtman, M., and Schuster, S.C. 2006. Who ate whom? Adaptive Helicobacter genomic changes that accompanied a host jump from early humans to large felines. *PLoS Genet.* **2:** e120. doi: 10.1371/journal.pgen.0020120.

Fleischmann, R.D., Adams, M.D., White, O., Clayton, R.A., Kirkness, E.F., Kerlavage, A.R., Bult, C.J., Tomb, J.F., Dougherty, B.A., Merrick, J.M., et al. 1995. Whole-genome random sequencing and assembly of *Haemophilus influenzae* Rd. *Science* **269:** 496–512.

Fournier, P.E., Vallenet, D., Barbe, V., Audic, S., Ogata, H., Poirel, L., Richet, H., Robert, C., Mangenot, S., Abergel, C., et al. 2006. Comparative genomics of multidrug resistance in *Acinetobacter baumannii*. *PLoS Genet.* **2:** e7. doi: 10.1371/journal.pgen.0020007.

Handelsman, J., Rondon, M.R., Brady, S.F., Clardy, J., and Goodman, R.M. 1998. Molecular biological access to the chemistry of unknown soil microbes: A new frontier for natural products. *Chem. Biol.* **5:** R245–R249.

Idury, R.M. and Waterman, M.S. 1995. A new algorithm for DNA sequence assembly. *J. Comput. Biol.* **2:** 291–306.

Keane, T. and Ning, Z. 2007. Assessing assemblability of reads from new sequencing platforms. 15th Annual International Conference on Intelligent Systems for Molecular Biology and 6th European Conference on Computational Biology, Vienna, Austria, July 21–25, 2007.

Kececioglu, J.D. and Myers, E.W. 1995. Combinatorial algorithms for DNA-sequence assembly. *Algorithmica* **13:** 7–51.

Lander, E.S. and Waterman, M.S. 1988. Genomic mapping by fingerprinting random clones: A mathematical analysis. *Genomics* **2:** 231–239.

Manber, U. and Myers, G. 1993. Suffix arrays—A new method for online string searches. *Siam J. Comput.* **22:** 935–948.

Margulies, M., Egholm, M., Altman, W.E., Attiya, S., Bader, J.S., Bemben, L.A., Berka, J., Braverman, M.S., Chen, Y.J., Chen, Z.T., et al. 2005. Genome sequencing in microfabricated high-density picolitre reactors. *Nature* **437:** 376–380.

Mitra, R.D. and Church, G.M. 1999. In situ localized amplification and contact replication of many individual DNA molecules. *Nucleic Acids Res.* **27:** e34. doi: 10.1093/nar/27.24.e34.

Mullikin, J.C. and Ning, Z. 2003. The phusion assembler. *Genome Res.* **13:** 81–90.

Mwangi, M.M., Wu, S.W., Zhou, Y., Sieradzki, K., de Lencastre, H., Richardson, P., Bruce, D., Rubin, E., Myers, E., Siggia, E.D., et al. 2007. Tracking the in vivo evolution of multidrug resistance in *Staphylococcus aureus* by whole-genome sequencing. *Proc. Natl. Acad. Sci.* **104:** 9451–9456.

Myers, E.W. 2005. The fragment assembly string graph. *Bioinformatics* **21:** 79–85.

Pevzner, P.A., Tang, H., and Waterman, M.S. 2001. An Eulerian path approach to DNA fragment assembly. *Proc. Natl. Acad. Sci.* **98:** 9748–9753.

Pop, M., Salzberg, S.L., and Shumway, M. 2002. Genome sequence assembly: Algorithms and issues. *Comput.* **35:** 47–54.

Pop, M., Phillippy, A., Delcher, A.L., and Salzberg, S.L. 2004. Comparative genome assembly. *Brief. Bioinform.* **5:** 237–248.

Service, R.F. 2006. Gene sequencing—The race for the $1000 genome. *Science* **311:** 1544–1546.

Slater, G.S. and Birney, E. 2005. Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics* **6:** 31. doi: 10.1186/1471-2105-6-31.

Smith, E.E., Buckley, D.G., Wu, Z., Saenphimmachak, C., Hoffman, L.R., D'Argenio, D.A., Miller, S.I., Ramsey, B.W., Speert, D.P., Moskowitz, S.M., et al. 2006. Genetic adaptation by *Pseudomonas aeruginosa* to the airways of cystic fibrosis patients. *Proc. Natl. Acad. Sci.* **103:** 8487–8492.

Sommer, D.D., Delcher, A.L., Salzberg, S.L., and Pop, M. 2007. Minimus: A fast, lightweight genome assembler. *BMC Bioinformatics* **8:** 64. doi: 10.1186/1471-2105-8-64.

Sundquist, A., Ronaghi, M., Tang, H., Pevzner, P., and Batzoglou, S. 2007. Whole-genome sequencing and assembly with high-throughput, short-read technologies. *PLoS ONE.* **2:** e484. doi: 10.1371/journal.pone.0000484.

Warren, R.L., Sutton, G.G., Jones, S.J., and Holt, R.A. 2007. Assembling millions of short DNA sequences using SSAKE. *Bioinformatics* **23:** 500–501.

Wendl, M.C. and Waterston, R.H. 2002. Generalized gap model for bacterial artificial chromosome clone fingerprint mapping and shotgun sequencing. *Genome Res.* **12:** 1943–1949.

Whiteford, N., Haslam, N., Weber, G., Prugel-Bennett, A., Essex, J.W., Roach, P.L., Bradley, M., and Neylon, C. 2005. An analysis of the feasibility of short read sequencing. *Nucleic Acids Res.* **33:** e171. doi: 10.1093/nar/gni170.

Received September 28, 2007; accepted in revised form March 5, 2008.