

Crystallizing short-read assemblies around lone Sanger reads

Sajjad Hossain¹, Navid Azimi¹ and Steven Skiena^{1*}

¹Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-4400 USA

ABSTRACT

New short-read sequencing technologies produce large volumes of 25-30 base paired-end reads. In this paper, we present a sequencing protocol and de novo assembler program (SHORTY) targeted towards such microread data. Our protocol augments short-paired reads using a trivially small number of Sanger reads (only one to three reads per bacterial genome). Still, these “seed reads” enable us to produce significant assemblies using about half the short-read coverage (50-60X) of comparable assemblers, despite our assumption of base error rates at least 10 times that of other groups. SHORTY exploits two new ideas which we believe to be of interest to the short-read assembly community: (1) using single seed reads to crystallize assemblies, and (2) estimating intercontig distances accurately from multiple spanning paired-end reads.

Contact: skiena@cs.sunysb.edu

1 INTRODUCTION

Several new short-read sequencing technologies are now actively competing in the race towards the \$1,000 genome. Each of these technologies produces raw sequence data with particular characteristics and distinct error models. However, it has become clear that three major contenders (Solexa/Illumina, Agencourt/Applied Biosystems, and Helicos BioSciences) aim to produce high volume, 25-30 base paired-end reads.

Short read sequencing has led to a new surge of interest in the old problem of sequence assembly. These new technologies have only recently started producing data suitable for de novo assembly. Several teams are now building short-read assemblers (see Section 3), but the protocols optimizing assembly projects (e.g. optimal mixes of short- and long-reads) are still being invented.

In this paper we present our assembler SHORTY, targeted towards paired-end microread sequencing data. Unlike others in the literature, SHORTY uses a trivially low volume of Sanger reads (Sanger *et al.*, 1977), only one to three reads per bacterial genome. Still, these “seed reads” enable us to produce significant assemblies using about half the short-read coverage (50-60X) of other recent results. Further, our final assemblies prove very accurate even though our reads contain 5% base error rates, which is ten times that assumed by single-read Solexa/Illumina data. Certain previous work on microread assembly underestimates the complexity of the problem by simulating assembly on error-free reads.

SHORTY exploits two new ideas which we believe will be of interest to the short-read assembly community:

- *Seed reads for crystallizing assemblies* – Several other assemblers intermix low (say 2x) coverage from Sanger or 454 reads with a higher coverage of short reads to fill up gaps. Instead, we use a single 500-base Sanger read to grow a neighboring contig of greater or equal to Sanger length. By repeating this process on the new contig, we can walk across the full genome assembling perhaps 90% of the genome into 4-5kb contigs. Assembling the results of such walks from a trivial number of seeds (say three) produces contigs with an N50 size of 30kb on bacterial genomes and 98% coverage in non-trivial contigs. These numbers should grow with additional tuning; indeed we expect a substantially larger N50 size by the time of the conference.

The Sanger coverage assumed by our protocol is so modest it eliminates the need for a lab to own more than one type of sequencing platform. These Sanger reads can be contracted out to a core facility for under \$50 per genome, or likely even replaced by highly-conserved ribosomal RNA sequences scavenged from databases.

- *Inter-contig distance estimation from spanning paired-end reads* – Sequencing protocols specify the mean separation distance and variance between the ends of the paired-end reads. Typically, these insert lengths are normally distributed, say with a mean distance of 3200 bases and a standard deviation of 150 bases. Our walking assembly strategy naturally produces two neighboring contigs separated by some insert distance. The substantial number of paired-end reads with one end anchored in each contig provides the possibility of accurately estimating the distance between the contigs. Such estimation enables us to order contigs and fill gaps using shorter overlaps that would be unconvincing in the absence of distance information.

In this paper, we develop our distance estimation efforts, which we believe will be helpful in any paired-end short read assembler. To the best of our knowledge we are the first short-read assembler to exploit this idea.

Section 3 surveys related work on short-read assembly. The primary research issue today is not the head-to-head comparison of which assembler is “best”, but to identify the most cost-effective short-read sequencing protocol which results in data that can be reconstructed when coupled with the right assembly strategy. Most relevant to us are two other assembly strategies focusing on double-ended short-reads:

- ALLPATHS (Butler *et al.*, 2007) is an assembler being developed at the *Broad Institute* reporting excellent assembly on

*Corresponding author, skiena@cs.sunysb.edu

paired-end Solexa-type data with 80X coverage using a protocol with three different insert sizes ($50\text{kb} \pm 10\%$, $6\text{kb} \pm 10\%$, and $0.5\text{kb} \pm 1\%$). Our results in this paper are not directly comparable. SHORTY produces somewhat lower quality assemblies, however our experiments assume (1) a substantially simpler, single library experimental protocol, (2) employing shorter reads (25 vs. 30bp), (3) assuming reads with substantially higher base error rates (5% vs. 0.3%), and (4) requiring substantially less coverage (50X vs. 80X).

- Medvedev and Brudno's RECOMB 2008 paper (Medvedev *et al.*, 2008) reports assembly results for bacterial scale genomes which are more directly comparable to ours. They assemble simulated 25-base paired (although error-free) reads into contigs with N50 contig sizes around 25kb. These contig sizes are essentially identical to ours, however the results they report use twice the short-read coverage we assume or more (100-200X).

Our paper is organized as follows. Section 2 reviews the three primary short-read sequencing technologies. Section 3 surveys short-read assemblers targeted for 454 and single-end Solexa data, respectively. The algorithmic work flow of SHORTY is described in Section 4. Our techniques for establishing positional information from mate pair data are discussed in Section 5. Experimental results are presented in Section 6, with conclusions in Section 7.

2 SHORT READ SEQUENCING TECHNOLOGIES

Although the Sanger sequencing method (Sanger *et al.*, 1977) has been the dominant sequencing technology for decades, novel technologies for short read sequencing are have been developed by several groups (Shendure *et al.*, 2004; Brenner *et al.*, 2003; Kling, 2003; Miller *et al.*, 2003; Ronaghi *et al.*, 1998). See Mitchelson, 2007 for a recent survey and analysis of these technologies. Most of them are based on *pyrosequencing* (Nyren, 2007; Nyren *et al.*, 1993; Ronaghi *et al.*, 1996, 1998).

Pyrosequencing is based on a "sequencing by synthesis" principle. More specifically, the method uses a chemical light-producing enzymatic reaction which is triggered when a molecular recognition event occurs. Essentially, the method allows sequencing of a single strand of DNA by synthesizing the complementary strand along it. Each time a nucleotide, A, C, G or T is incorporated into the growing chain, a cascade of enzymatic reactions is triggered which causes a light signal. 454 Life Sciences (www.454.com) has developed an array based pyrosequencing which has emerged as an excellent platform for large scale DNA sequencing. Their read length is typically around 100 bases, which provides much more significance to read overlaps for an assembler than is possible when reads are as small as 25-30 bases.

These new short-read sequencing technologies differ in details of localizing molecules, amplification and sequencing approach. Our assembler has been developed for microread technologies that generate mate paired short reads. Hence, it is suitable for data generated by companies like:

- *Applied Biosystems* (www.appliedbiosystems.com): They recently released their sequencing machine SOLiDTM, which uses a sequencing technology that is acquired

from Agencourt Bioscience Corporation (www.agencourt.com). Agencourt commercialized their technology based on *Polony Sequencing* developed by Church and Mitra (Mitra *et al.*, 1999, 2003). Indeed, the parameters underlying our simulations were selected with SOLiDTM in mind.

- *Solexa* (www.solexa.co.uk): They were recently acquired by *Illumina* (www.illumina.com). Their sequencing machine, *Illumina Genome Analyzer*, can load to eight samples onto their flow cell surface for simultaneous analysis. The platform offers high accuracy, high throughput and relatively low cost (\$3000 per run, \$400 per channel), and promises real support for double-ended reads forthcoming very soon.
- *Helicos BioSciences* (www.helicosbio.com): Based on technology from Braslavsky *et al.*, 2003, they are pioneering a single-molecule approach to sequencing. This offers advantages in capacity and eliminating amplification-specific bias. Their HeliScopeTM sequencing machine contains two flow cells where billions of single molecules of sample DNA are captured on an application-specific proprietary surface to serve as templates for the sequencing-by-synthesis process.

Table 1 compares the primary performance characteristics of various short read sequencing technologies.

3 BACKGROUND AND RELATED WORK

The success of shotgun sequencing (Sanger *et al.*, 1977) led to the development of many successful assemblers for Sanger reads. Most of them were based on the overlap-layout-consensus (Kececioglu *et al.*, 1995) paradigm, while others took a graph-theoretic approach. Some assemblers were suitable for hierarchical sequencing, while others targeted *whole genome shotgun* (WGS) sequencing. A partial list of major assemblers include Arachne (Batzoglou *et al.*, 2002; Jaffe *et al.*, 2003), Celera (Myers *et al.*, 2000), TIGR (Sutton *et al.*, 1995), PHRAP (Green, 1994), EULER (Pevzner *et al.*, 2001), Phusion (Mullikin *et al.*, 2003), JAZZ (Shapiro, 2005), and CAP3 (Huang *et al.*, 1999). Also available are tools like Consed (Gordon *et al.*, 1998) and BAMBUS (Pop *et al.*, 2004), which can be used as finisher for other assemblers which may produce unrelated or mis-assembled contigs.

As short read sequencing technologies mature, several bioinformatics groups have started working on short read assembly projects. Most algorithms are still tested on simulated data, as true assembly-quality data is not yet readily available for most platforms. Solexa double-ended reads and Applied Biosystems' SOLiDTM system have just entered the market, so real data should be available in short order.

We classify short read assemblers in three different groups, based on the type of reads they expect. The two assemblers most similar to our own work (using short paired-end reads) have been discussed in the introduction. The second class are assemblers targeting 454 data, which include:

- *Newbler* (Margulies *et al.*, 2007) is a proprietary de novo assembler from 454 Life Sciences Corporation which is designed to handle their data which is in the form of flowgrams. It is based on the overlap-layout-consensus paradigm and consists of three modules: Overlapper, Unitigger and Multialigner.

Table 1. Comparison of different commercial short read sequencing technologies.

Company	Machine	Throughput (per run)	Read length (base)
454	GS FLX	100M bases/7 hours	100 or more
Helicos	HeliScope	2G bases/day	around 25
Applied Biosystems	SOLiD	4G bases	25 – 30
Solexa	Illumina Genome Analyzer	2G bases/2 days	25 – 30

As 454 doesn't typically produce paired-end data, *Newbler* generates a set of unlinked contigs.

- *EULER* (Chaisson *et al.*, 2004) analyzed the feasibility of short read assembly of read length 70–200 using *EULER*. On simulated data from several bacterial genomes, they produced a mix of long and short contigs.
- *EULER-SR* (Chaisson *et al.*, 2008), the new version of *EULER* is particularly designed for reads generated by next generation sequencing technologies. The results are based on a hybrid approach where they used 454 and Sanger type data together to generate an assembly. They presented some results for simulated paired 454 reads as well.
- *SHRAP* (Sundquist *et al.*, 2007) is another assembler that assembles reads of length around 200 base pairs using a proposed sequencing protocol for mammalian-scale genomes.

A final class of short-read assemblers focuses on single-ended reads produced by the first generation of Solexa machines:

- *SSAKE* (Warren *et al.*, 2007) is a short read assembler that was tested with simulated error-free 25 mers. It performs well with viral genomes. In a recent release, *SSAKE* started supporting paired end reads.
- *SHARCGS* (Dhom *et al.*, 2007) is a de novo short read assembler that handles short reads as short as 25–40 bases. It generates a set of large contigs but without any ordering information. Their algorithm was tested against *Illumina's* 1G sequencing instrument. It uses a method that it calls *contig elongation*: a read is extended by looking for other reads in a prefix tree for potential extensions. It doesn't work with paired-end reads.
- *Phusion* (Mullikin *et al.*, 2003) was used by *Sanger Institute* to assemble many genomes from shotgun sequences. Recently they showed (Keane *et al.*, 2007) possibilities of assembling short reads but with mixing a low coverage (0.5-2X) of capillary reads with them. They used 454 and Solexa data for their prototype.
- *Velvet* (Zerbino *et al.*, 2007) augments 50X Solexa data with significant coverage (2X) of paired Sanger reads to produce high quality assemblies.

4 ALGORITHMIC WORK FLOW

SHORTY is designed to work on paired end short reads. These reads (optionally accompanied by quality scores) are generated by recent technologies developed by Applied Biosystems, Solexa and others to come. Laboratory protocols aim to select targets whose insert

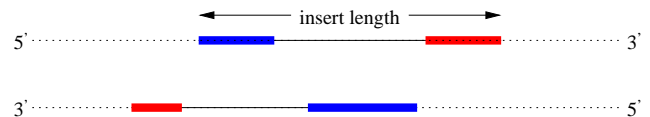


Fig. 1. Paired end reads are collected from both the strands where the left-mer (marked in blue) is always on the side of 5' (or 3') end.

separating the two reads is normally distributed around a given target length. Each paired read contains a distinct *left-mer* (*lmer*) and a *right-mer* (*rmer*) (Figure 1). These reads may contain insertion, deletion, substitution or homo-polymer errors. Our protocol will also need a few (1-3) Sanger sequencing reads (we call them *seeds*) to start with.

4.1 Read Hashing

In a typical data set, we will have millions of pairs of reads. The way we make use of these reads requires them to be accessed many times during the assembly process. Thus we hash all the pairs based on the *k-mers* present in the *left-mer* and *right-mer*. From each pair, we generate another pair of reads. The *left-mer* of the new pair will be the reverse complement of the *right-mer* of the original pair. Similarly the *right-mer* of the new pair will be the reverse complement of the *left-mer* of the original pair. We hash the new pair in a similar way. So, after the hashing process is finished, we know for a particular *k-mer* which pairs of reads contain that *k-mer* in their *left-mer* or *right-mer*. A typical value of *k* is 8 or 9 on a 2GB machine but can be larger based on available system memory and amount of errors in the reads.

4.2 Processing a Seed

For each *seed* we detect the group of read pairs whose *left-mers* (or *right-mers*) will map onto that seed. With high probability, this group of *right-mers* (or *left-mers*) belong to some neighboring reads in the reference sequence (Figure 2). The previously built hash table of reads is used to determine this group. We take into consideration various types of sequencing errors while trying to map a read on the *seed*. This group forms the basis of forming larger contigs and next generation *seeds*.

While processing a seed, we must be careful in handling repeat regions (Figure 3) as these might generate contigs with misleading positional information. Another situation where this problem can arise is when there are regions which are similar to their reverse complement. We call such regions *palindrome* regions. One way we detect seeds with repeats or palindrome regions is by checking the number of reads that a seed can map onto itself. If it attracts

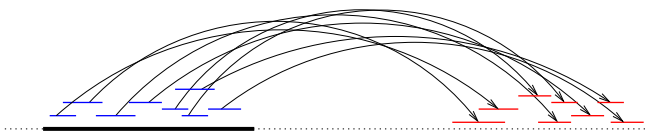


Fig. 2. Group of *right-mers* (red) whose corresponding *left-mers* (blue) can be mapped on the *seed* (black). It can also be in the other way, i.e. *right-mers* mapped on the *seed* to form a group of *left-mers*.

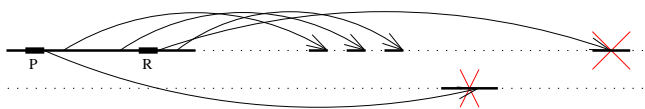


Fig. 3. Repeats (R) and palindromes (P) can generate contigs with misleading positional information.

too many reads, there is a strong chance that it has one of those problems.

4.3 Generating Contigs

The group of reads formed in the previous stage is expected to have overlapping reads when we are presented with enough input coverage. We greedily merge them based on overlaps among them. This generates a set of contigs who are expected to be in neighboring positions in the reference. While considering an overlap, we also take into account the quality of the bases. Considering the possibility that the reads may contain different kinds of errors, it is not always possible to have ‘clean’ overlaps. We use a *dynamic programming* based approach where we penalize for such error conditions. We also store voting information (for each position we count the number of A, T, G and C that comes from the constituting reads) for each position of a contig which helps us to choose the appropriate base for a position in case we have multiple candidates which might be the result of errors in the constituting reads. This can be used to determine the quality of that base.

4.4 Contig Geography

We also generate positional information for each contig (we call it *contig-geography*) using the *insert-length* of the *read-pairs* and its standard deviation. We exclusively use this information in future steps especially to identify potential overlaps and order the *contigs*. Contig geography is discussed in greater detail in Section 5.

4.5 Generating New Seeds

Some of the *contigs* generated in the previous stage might qualify to be used as *seeds* again. These *next generation seeds* are chosen based on the length and quality of the *contigs*. It is also possible to use a collection of neighboring *contigs* as a single *seed*.

4.6 Generating Larger Contigs

At this stage we are expected to have decent sized *contigs*. But we can even do better with *contig-geography* which we have been generating and updating so far. This information will help us filling up gaps and generating even larger *contigs* by allowing smaller overlaps.

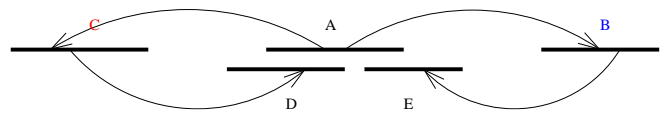


Fig. 4. Because of the standard deviation of *insert-length*, gap between successive *contigs* gets reduced.

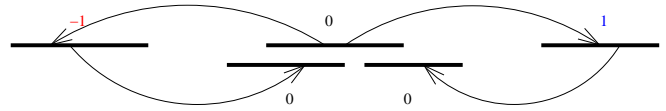


Fig. 5. Because of the standard deviation of *insert-length*, gap between successive *contigs* gets reduced.

Figure 4 explains how the gap between *contigs* get reduced. *Contigs B* and *C* are generated from seed *A*. If the standard deviation of *insert length* was zero, *D*, which is generated from *C*, must have overlapped completely with *A*. But we can safely assume that we will have 10-15% standard deviation in actual situation. However, we can still proceed in a situation where the standard deviation is zero using multiple *seeds*. As we can see in Figure 4, *A*, *D* and *E* now overlap to produce a larger *contig* while also reducing the gaps.

The *contig-geography* provides us with two very important pieces of information: *a*) an approximate distance between a *seed* and *contigs* generated from that *seed*, *b*) information about which *contig* is generated after which *contig*. The distance information helps us to calculate an approximate starting position for a *contig*. The later information allows us to generate an *id* for each *contig* which we call *generation-id*. The *generation-id* of a *contig* is 1 more than its *seed's generation-id* if it is composed of *right-mers*, otherwise it is 1 less than that of its *seed*. Figure 5, which corresponds to the *contigs* in Figure 4, shows the *generation-ids* of the *contigs*. *Contigs* which are neighbors should have similar *generation-ids* and also their approximate starting positions should be close. This gives us a clue for searching possible overlaps. Instead of looking for overlaps among all the *contigs* in our collection, we can only try among those who have near similar *generation-ids* and close enough starting positions. This accelerates the finishing job of SHORTY.

At this point we have *contigs* large enough to be considered as input for traditional shotgun sequence assemblers. In our experiments, we use the TIGR assembler (Sutton et al., 1995) for our final-stage processing (see Table 3).

4.7 Contig Scaffolding

We use mate pair data in such a way that it is possible to trace back which contig was generated from which seed (Figure 6). We construct a directed graph $G = (V, E)$ from this information. Vertices of G are the *contigs* and each seed is connected to all of its children. A chain of *contigs* in the same direction in the graph forms a scaffold. Contiguous *contigs* in the scaffold can be merged if they become large enough in the previous steps. We can also re-map our initial reads to fill the gaps in a scaffold (Figure 7). If the gaps still persist, we can provide an estimated gap length using *contig-geography*.

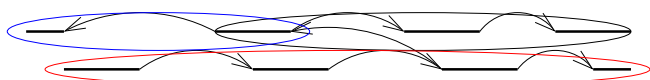


Fig. 6. A chain of contigs in the same direction is used as a scaffold. Three different scaffold are shown in this figure in red, blue and black.



Fig. 7. Gaps in scaffolds can be filled by re-mapping the reads onto the contigs.

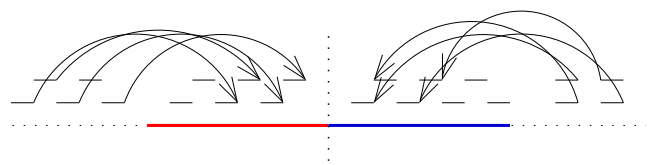


Fig. 8. Chimeras can be detected by mapping back the reads onto the contig. The contig can be partitioned at a point where no pair has its reads in both sides of that point.

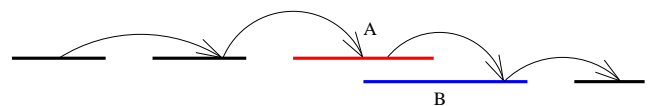


Fig. 9. Contigs shown here are from one scaffold. A and B are two consecutive contigs large enough to overlap. If they don't overlap, there is a chance that either A or B is a chimera produced due to mis-assembly.

4.8 Detecting Misassemblies

We can detect mis-assemblies by mapping the reads back to the contigs (Figure 8). If there is a point where a wrong merge occurred, there should not be any mate paired reads where both of them get mapped to the opposite sides of that point. We can split the contigs in such points. Our scaffolds can also be another source of identifying *chimeras* (Figure 9). Our scaffolds are ordered chains of contigs generated from mate paired reads. When two contigs become large enough to cover the gap between them, they should overlap. Otherwise, there is a chance that at least one of them is a *chimera*.

5 POSITIONAL INFORMATION FROM MATE PAIRED READS

The combination of relatively high-coverage as realized by paired-end microreads provides new opportunities to accurately estimate the distance between non-overlapping contigs. The simulations discussed in this paper assume 50x sequence coverage in 25-base paired reads. This yields an expectation of two reads starting from each position on the genome, half of which will represent the 5' read. This implies that the number of read-pairs spanning any interior position on the genome roughly equals the insert length (centered around 3200 bases in our simulations). Thus hundreds

or even thousands of read pairs connect each two non-overlapping contigs, all of whose insert sizes were drawn from a normal distribution of known mean and standard deviation. By analyzing where these read-pairs map on each contig we can accurately estimate the intercontig distance.

In this section, we explore techniques for analyzing *contig geography*. Accurate distance estimation is vital in later-stage contig merging in SHORTY. Many contigs generated from seeds overlap, but too weakly to be statistically significant over the scale of a genome assembly. Accurate information about position enables us to merge them confidently. Secondary benefits include reduced running times (by avoiding unpromising contig-overlap pairs) and dealing with repeats.

5.1 Gap Between a Contig and Its Seed

For these estimations, it is assumed that contigs and their respective seeds are from the same strand. Define distance d between two contigs to be the distance between their first nucleotides in the actual sequence. This definition is useful because it is independent of contig size. We reduce the problem of estimating this distance to a parameter estimation problem. Assume $C = \{c_1, c_2, \dots, c_n\}$ and $S = \{s_1, s_2, \dots, s_n\}$ where c_i is the position of the i^{th} read on the contig and s_i is the position of the other read (from the pair) on the seed. Assume that insert length for each short read pair is a random variable X which belongs to parametric family $f_{\theta_{x_1}, \theta_{x_2}, \dots, \theta_{x_m}}(x)$. $E[X] = \mu$ and $E[(X - \mu)^2] = \sigma^2$. Define $Y = X - d$. Note that Y also belongs to $f_{\theta_{y_1}, \theta_{y_2}, \dots, \theta_{y_m}}(y)$ with the same number of parameters. Here, for i^{th} read, $y_i = c_i + s_i$ is an observation of Y . Thus Y 's parameters can be estimated using this observation. Recall that $d = X - Y$, so we can compute the probability of the value of d by having the parameters of X and Y distribution. Since density of X is known (from parameters of the normal insert length distribution), our problem reduces to estimating the parameters of Y from the observations we get from each read.

We assume the *insert length* is governed by a normal distribution $N(\mu_x, \sigma_x^2)$ which means $\theta_1 = \mu_y$ and $\theta_2 = \sigma_y$. Using maximum likelihood estimation, we have:

$$\mu_y = \frac{1}{n} \sum_{i=1}^n c_i + s_i, \quad \sigma_y^2 = \frac{1}{n} \sum_{i=1}^n (c_i + s_i)^2$$

We define $d = X - Y$ as before. Since both X and Y are normally distributed, d must also have a normal distribution with $\mu_d = \mu_x - \mu_y$ and $\sigma_d^2 = \sigma_x^2 + \sigma_y^2$.

Figure 10 reports the accuracy of our distance estimation between contigs and their respective seeds. The sharp peak centered around zero indicates that we can position almost all contigs with an expected error which is small relative to the length of the contig itself. This means we can accurately order these contigs relative to their seed, which determines the neighbors to evaluate as possibly overlapping contigs. It is also accurate enough to determine whether we are likely to be able to bridge the gap using a smaller unpositioned contig (possibly a single read) under appropriately relaxed conditions.

5.2 Merging Contigs and Improving Distance Estimation

In the scaffold graph G we used to represent relationships between contigs (Section 4.7), each edge is assigned a weight estimating to

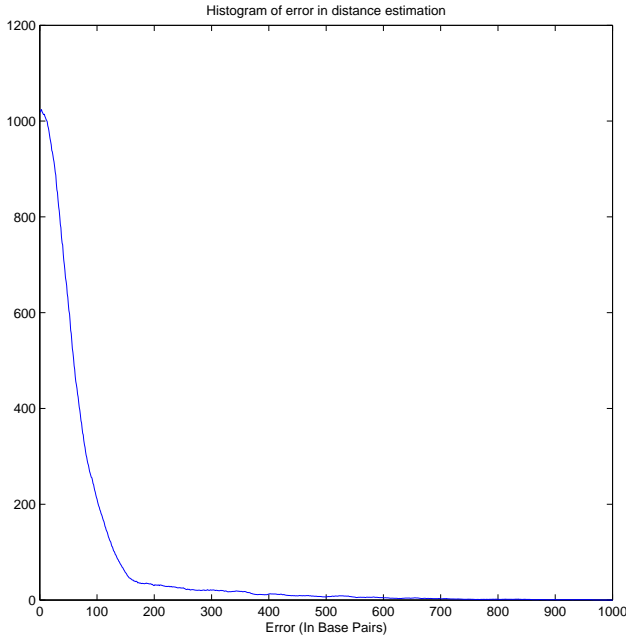


Fig. 10. Histogram showing the base distance estimation error in *local* contig/seed gaps, generated over 90,000 contig pairs.

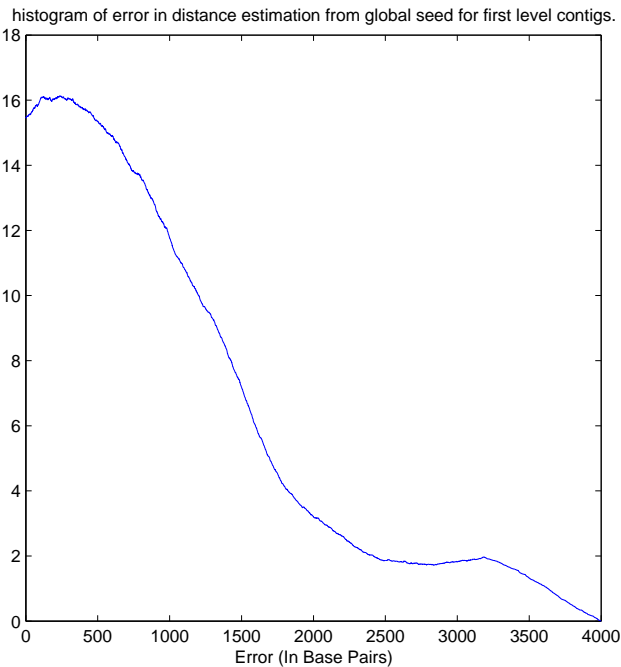


Fig. 11. Histogram showing the base distance estimation error in *global* contig/seed gaps, generated from 55,000 contig pairs.

the distance between the contig-pair. Initially this distance is estimated using the techniques of the previous subsection, but we can refine it through a computation on all scaffold paths between the contig pair.

Using the estimated seed-to-contig distances, we can sort the contigs and try to merge them based on predicted overlaps. When we merge two contigs we also merge the corresponding vertices on graph G . After each merge, we recalculate the edge weights keeping iterating for new merges. An optimal correction to the edge weights should make the length of all the paths from the root to the merged contig equal, while making the least possible changes to edge weights. This is a rather time-consuming computation, so instead we iterate the following set of steps to merge contigs:

1. Compute the distance of all the contigs from the initial seed. Find all the possible merge candidates.
2. Attempt to do all of the merges computed in step 1 and make new contigs.

We call contigs generated at i th iteration as i th level contigs and use d_{i_1, i_2, \dots, i_n} to show the distance estimation from the initial seed for the contig generated from merging the first level contigs i_1, i_2, \dots, i_n . We have already discussed the distance estimation for the first level contigs. For the other level contigs we can still use the read information but this is very time consuming task. Instead we use the following formula:

$$d_{i_1, i_2, \dots, i_n} = \frac{1}{\sum_{j=1}^n \frac{1}{depth_{i_j}}} \sum_{j=1}^n \frac{1}{depth_{i_j}} d_{i_j}$$

where $depth_{i_j}$ is the depth of contig i_j in G (the position i_j in its scaffold) and d_{i_j} is the distance for contig i_j from the initial seed.

Figure 11 shows the quality of our estimation of global distances of contigs. In the absence of repeats, we can quantify the separation between two contigs on a scaffold to within 4kb (with 3.2kb inserts) even though the contigs be hundreds of thousands of bases apart.

6 EXPERIMENTS

6.1 Data

All experiments in this paper are performed on simulated reads from *Streptococcus suis*, strain P1/7 (www.sanger.ac.uk/projects/s_suis). The bacteria has a genome containing a significant number of repeat regions, which complicates the process of assembly. Table 2 shows the repeat counts for this genome.

Our simulations were designed to conform as closely as possible to an assembly project on the Applied Biosystems' SOLiD platform. Indeed our coverage, insert distribution, and base error distribution are derived from an actual data set. Unfortunately, the reads were sampled in a highly non-uniform manner, presumably due to correctable problems in sample preparation. For this reason we believe that our simulated data gives a better perspective on how our assembler will perform in practice.

Specifics on our simulation parameters are given below. The mate pairs were collected uniformly from the 2,007,491 bases long reference sequence. All reads were 25 bases long, where 15% of the reads had 3 errors, 20% had 2 errors and 28% had 1 error. All errors

Table 2. Repeats in *S. suis* sequence. *Color Space* is the SOLiD representation of DNA sequences, whereas *Letter Space* is the traditional representation.

space	≥ 75	$\geq 1,000$	$\geq 3,000$	$\geq 5,000$	max
<i>letter</i>	382	25	6	6	6151
<i>color</i>	180	15	6	6	6137

Analysis was done using MUMmer (Kurtz *et al.*, 2004)

here are substitution errors. The insert lengths were normally distributed and centered around 3200bp with a standard deviation of 150bp. All of these parameters are in accord with real datasets we have analyzed. Sanger seeds used in the experiments were all 500 bases long and contain no error. They were selected uniformly at random from the reference sequence.

A characteristic of ABI SOLiD data is that it is generated in so-called *two base encoding* or *color space*. Properties of this encoding is that reverse complement of a sequence in letter space is equivalent to just the reverse in color space translation. More about two-base encoding can be found at www.appliedbiosystems.com. All the sequences in this paper were both assembled and compared to the reference genome in color space.

6.2 Results

Our results from several datasets with read coverage varying from 50x-75x are summarized in Table 3. Two assemblies are provided for each data set. Those labeled *SHORTY* represent the union of runs from distinct seeds with no attempt to further assemble them. Those labeled *SHORTY.T* report the results of applying a traditional assembler (TIGR assembler (Sutton *et al.*, 1995)) to merge overlaps in these sequences. We consider the *SHORTY.T* results more representative of the assemblies to be obtained in practice, although our contig accuracy rate suffers somewhat due to chimeras induced by this last stage assembly. Such misassemblies should be detectable by the unusually low frequency of read-pairs spanning the mistaken junction.

The results in Table 3 shows that for all five of these runs, the N50 size ranges from 17.1kb to 32.2kb, in terms of finished contigs with base accuracies of over $\geq 99.9\%$. The N50 size is a standard measure of assembly quality denoting the size of the smallest contig such that 50% of the reference sequence is contained in contigs of size N50 or greater. Table 3 also shows that between 96.7% and 99.0% of the reference genome occurs in substantial (≥ 100 base) finished contigs, each with base accuracies of over $\geq 99.9\%$. This result, coupled with the mate pairs linking these contigs implies we have tiled the genome with ordered, substantial contigs. We are confident that our N50 number will rise substantially as we explore methods to fill these (usually) trivial gaps using a second look at the original sequence reads.

Figure 12 illustrates the contig-size distribution of three of our assemblies (for 55x, 65x, and 75x coverage respectively) both pre- (left) and post- (right) final phase assembly. The insets highlight detail with respect to smaller but not trivial contig sizes; recall that we are starting with only 25-base reads!

7 DISCUSSION AND FUTURE DIRECTIONS

The results we have presented here provide evidence that high-quality short read assembly is indeed possible using simple and economical protocols on real short-read data. Unlike previous work, our protocol uses a single sample preparation as opposed to a mix of insert sizes or runs on a mix of different platforms (e.g. 454 and Solexa). Our assemblies thrive on significant variance in insert length, further simplifying preparation over others in the literature. We make do with about half the coverage reported elsewhere.

Our use of single Sanger reads is more of a nuisance than a problem, as this data can be obtained cheaply through outsourcing services. An interesting question is whether they are really necessary. Database sequence from closely-related species should suffice, but even more to the point is noting how little information they add to the process. Three 500 base Sanger reads represent only 3000 bits of information in an assembled genome of 4,000,000 bits, making it hard to believe they really are essential.

Our primary direction of further work is demonstrating significant de novo assemblies on each of the major short-read platforms, namely ABI SOLiDTM, Solexa paired-read data, and Helicos Biosciences data as they are available to us. We are also working to raise our N50 sizes through gap filling techniques based on accurate positional estimation.

FUNDING

This project is partially supported by NSF Grants EIA-0325123 and DBI-0444815.

ACKNOWLEDGEMENTS

We are grateful for earlier work on *SHORTY* by J. Chen and G. Sabbani. We also thank Francisco M. De La Vega and Michael D. Rhodes of Applied Biosystems for their interest in the project.

REFERENCES

- Batzoglou, S., Jaffe, D.B., Stanley, K., Butler, J., Gnerre, S., Mauceli, E., Berger, B., Mesirov, J.P., Lander, E.S. (2002) ARACHNE: a whole-genome shotgun assembler. *Genome Research*, **12**, 177-179.
- Braslavsky, I., Hebert, B., Kartalov, E., Quake, S. (2003) Sequence Information can be obtained from single DNA molecules. *PNAS*, **100**, 3960-3964.
- Brenner, S., Johnson, M., Bridgham, J., Golda, G., Lloyd, D., Johnson, D., Luo, S., McCurdy, S., Foy, M., Ewan, M., *et al* (2003) Gene expression analysis by massively parallel signature sequencing (MPSS) on microbead arrays. *Nat. Biotechnol.*, **18**, 630-634.
- Butler, J., MacCallum I., Kleber, M., Shlyakhter, I., Belmonte, M., Lander, E., Nusbaum, C., Jaffe, D. (2007) ALLPATHS: de novo assembly of whole-genome shotgun microreads, manuscript.
- Chaisson, M., Pevzner, P.A. (2008) Short Read Fragment Assembly of Bacterial Genomes. *Genome Research (in press)*.
- Chaisson, M., Pevzner, P.A., Tang, H. (2004) Fragment assembly with short reads. *Bioinformatics*, **20**, 2067-2074.
- Chen, J., Skiena, S. (2007) Assembly For Double-Ended Short-Read Sequencing Technologies. *Advances in Genome Sequencing Technology and Algorithms*, 123-141.
- Dohm, J., Lottaz, C., Borodina, T., Himmelbauer, H. (2007) SHARCGS, a fast and highly accurate short-read assembly algorithm for de novo genomic sequencing. *Genome Research*, **17**, 1697-1706.
- Garey, M.R., Johnson D.S. (1979) *Computers and Intractability*, 228.
- Gordon, D., Abajian, C., Green, P. (1998) Coned: A Graphical Tool for Sequence Finishing. *Genome Res.*, **8**, 195-202.
- Green, P. (1994) Documentation for PHRAP.

Table 3. Assembly results for different data sets. Read coverage is the ratio of total read length to the reference sequence length. Coverage is the fraction of the reference sequence covered by the contigs. Rows with *SHORTY_T* show results when contigs are fed through TIGR. Results are shown within different levels of contig accuracy (97%, 99%, 99.9%). Minimum length of a contig considered for this analysis was 100 bases. All contig lengths are expressed in kilo bases.

Data	Read Coverage	# of Seeds	Assembler	Coverage (%)			Maximum length (kb)			N50 length (kb)			% of total contigs		
				97%	99%	99.9%	97%	99%	99.9%	97%	99%	99.9%	97%	99%	99.9%
#1	50	3	<i>SHORTY</i>	97.3	97.3	96.9	45.0	45.0	45.0	9.5	9.1	8.9	99.2	99	98.6
			<i>SHORTY_T</i>	96.2	95.8	94.9	59.0	59.0	59.0	17.9	17.7	17.1	97.7	96.9	96
#2	55	3	<i>SHORTY</i>	98.8	98.8	98.7	50.7	50.7	50.7	9.0	9.0	8.8	99.2	99	98.7
			<i>SHORTY_T</i>	95.8	95.5	91.7	109.3	109.3	109.3	25.9	25.9	19.4	92.4	90.5	88
#3	60	3	<i>SHORTY</i>	99	99	98.9	28.1	28.1	28.1	2.6	2.6	2.6	99.2	98.9	98.6
			<i>SHORTY_T</i>	95.2	93.7	90.4	96.6	64.5	64.5	23.9	23.4	21.0	86.9	83.0	78.4
#4	65	3	<i>SHORTY</i>	99.2	99.1	99.0	33.0	33.0	33.0	2.8	2.8	2.8	99.3	99.0	98.8
			<i>SHORTY_T</i>	95	93.6	89.1	98.2	98.2	89.2	33.4	33.2	32.2	88.7	84.6	79.6
#5	75	2	<i>SHORTY</i>	97.8	97.8	97.7	29.7	29.7	29.7	2.7	2.7	2.7	99.2	99	98.7
			<i>SHORTY_T</i>	95.2	94.4	92.7	60.6	60.6	60.6	23.5	23.5	23.3	93.4	91.2	89.1

- Huang, X., Madan, A. (1999) CAP3: A DNA Sequence Assembly Program, *Genome Research*, **9**, 868-877.
- Jaffe, D., Butler, J., Gnerre, S., Lindblad-Toh, K., Mauceli, E., Berger, B., Zody, M., Mesirov, J., Lander, E. (2003) Whole-genome shotgun assembly for Mammalian Genomes: Arachne2, *Genome Research*, **13**, 91-96.
- Keane, T., Ning, Z. (2007) Assessing Assemblability of Reads from New Sequencing Platforms, *15th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) Poster*.
- Kececioğlu, J.D., Myers, E.W. (1995) Combinatorial Algorithms for DNA Sequence Assembly, *Algorithmica*, **13**, 7-51.
- Kling, J. (2003) Ultrafast DNA sequencing, *Nat. Biotechnol.*, **21**, 1425-1427.
- Kurtz, S., Phillippy, A., Delcher, A., Smoot, M., Shumway, M., Antonescu, C., Salzberg, S. (2004) Versatile and open software for comparing large genomes, *Genome Biology*, **5**, R12.
- Margulies, M., Jarvie, T. P., Knight, J. R., Simons, J. F. (2007) The 454 Life Sciences PicoLiter Sequencing System, *New High Throughput Technologies for DNA Sequencing and Genomics*, 151-186.
- Medvedev, P., Brudno, M. (2008) Ab Initio Whole Genome Shotgun Assembly With Mated Short Reads, *RECOMB 2008*.
- Miller, R.D., Duan, S., Lovins, E.G., Kloss, E.F., Kwok, P.-Y. (2003) Efficient high-throughput resequencing of genomic DNA, *Genome Res.*, **13**, 717-720.
- Mitchelson, K. (2007) New High Throughput Technologies For DNA Sequencing And Genomics, **2**
- Mitra, R., Church, G. (1999) In situ localized amplification and contact replication of many individual DNA molecules, *Nucleic Acids Research*, **27**, 1-6.
- Mitra, R., Shendure, J., Olejnik, J., Church, G. (2003) Fluorescent in situ Sequencing on Polymerase Colonies, *Analyt. Biochem.*, **320**, 55-65.
- Mullikin, J.C., Ning, Z. (2003) The Phusion Assembler, *Genome Res.*, **13**, 81-90.
- Myers, E.W., Sutton, G.G., Delcher, A.L., Dew, I.M., et al. (2000) A whole-genome assembly of *Drosophila*, *Science*, **287**, 2196-2204.
- Nyren, P. (2007) The history of pyrosequencing, *Methods Mol Biol.*, **373**, 1-14.
- Nyren, P., Pettersson, B., Uhlen, M. (1993) Solid phase DNA minisequencing by an enzymatic luminometric inorganic pyrophosphate detection assay, *Anal. Biochem.*, **208**, 171-175
- Pevzner, P., Tang, H., Waterman, M. (2001) An Eulerian path approach to DNA fragment assembly, *Proceedings of the National Academy of Sciences of the United States of America*, **98**, 9748-9753.
- Pop, M., Kosack, D., Salzberg, S. (2004) Hierarchical Scaffolding With Bambus, *Genome Res.*, **14**, 149-159.
- Ronaghi, M., Karamohamed, S., Pettersson, B., Uhlen, M., Nyren P. (1996) Real-time DNA sequencing using detection of pyrophosphate release, *Anal. Biochem.*, **242**, 84-89.
- Ronaghi, M., Uhlen, M., Nyren, P. (1998) DNA sequencing: a sequencing method based on real-time pyrophosphate, *Science*, **281**, 363-365.
- Sanger, F., Nicklen, S., Coulson, A.R. (1977) DNA sequencing with chain-terminating inhibitors, *Proc. Natl. Acad. Sci. USA.*, **74**, 5463-5467.
- Shapiro (2005) Outline of the Assembly process: JAZZ, the JGI In-House Assembler.
- Shendure, J., Mitra, R.D., Church, G.M. (2004) Advanced sequencing technologies: methods and goals, *Nature Rev. Gen.*, **5**, 335-344.
- Sundquist, A., Ronaghi, M., Tang, H., Pevzner, P., Batzoglou, S. (2007) Whole-genome sequencing and assembly with high-throughput, short-read technologies, *PLoS ONE*, **2**, e484.
- Sutton, G., White, O., Adams, M., Kerlavage, A. (1995) TIGR Assembler: A new tool for assembling large shotgun sequencing projects, *Genome Science & Technology*, **1**, 9-19.
- Warren, R.L., Sutton, G., Jones, S., Holt, R. (2007) Assembling millions of short DNA sequences using SSAKE, *Bioinformatics*, **23**, 500-501.
- Whiteford, N., Haslam, N., Weber, G., Prugel-Bennett, A., Essex, J.W., Roach, P.L., Bradley, M., Neylon, C. (2005), An analysis of the feasibility of short read sequencing, *Nucleic Acids Res.*, **33**, e171.
- Zerbino, D., Birney, E. (2007) Velvet: Algorithms for De Novo Short Read Assembly Using De Bruijn Graphs, manuscript, 2007.

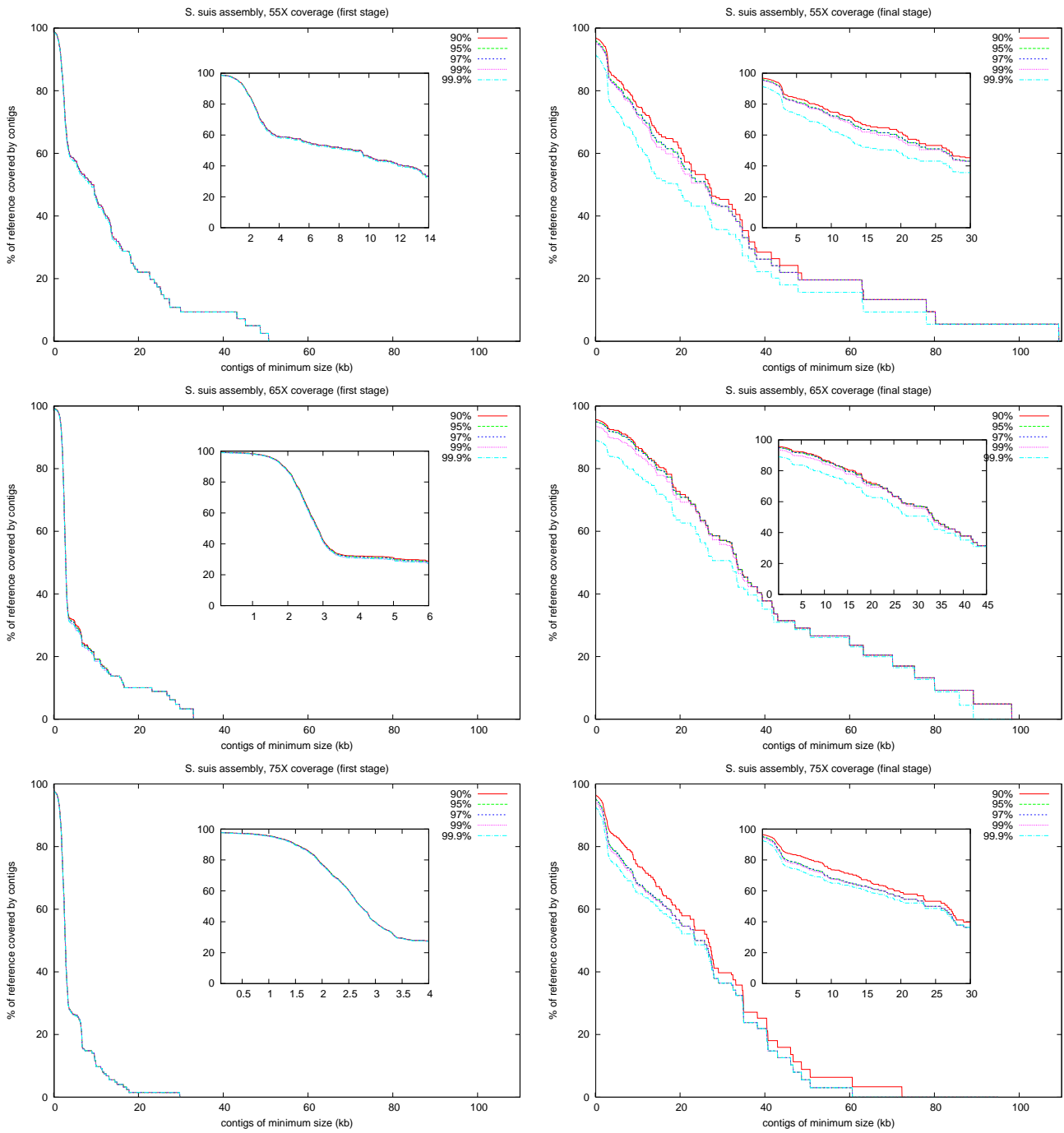


Fig. 12. Coverage of the reference sequence by various sizes of contigs with different level of contig accuracies (90%, 95%, 97%, 99%, 99.9%; shown in different color). Contig lengths are in kilo bases. First portion of a graph (especially the region containing N50) is zoomed inside the small rectangles.