BIOINFORMATICS

Vol. 20 no. 13 2004, pages 2067–2074
doi:10.1093/bioinformatics/bth205

Fragment assembly with short reads

Mark Chaisson[1,*], Pavel Pevzner[2] and Haixu Tang[2]

[1]Bioinformatics Program and [2]Department of Computer Science and Engineering, University of California San Diego, La Jolla, CA 92093, USA

Received on October 8, 2003; revised on February 3, 2004; accepted on February 9, 2004
Advance Access publication April 1, 2004

ABSTRACT

Motivation: Current DNA sequencing technology produces reads of about 500–750 bp, with typical coverage under 10×. New sequencing technologies are emerging that produce shorter reads (length 80–200 bp) but allow one to generate significantly higher coverage (30× and higher) at low cost. Modern assembly programs and error correction routines have been tuned to work well with current read technology but were not designed for assembly of short reads.

Results: We analyze the limitations of assembling reads generated by these new technologies and present a routine for base-calling in reads prior to their assembly. We demonstrate that while it is feasible to assemble such short reads, the resulting contigs will require significant (if not prohibitive) finishing efforts.

Availability: Available from the web at http://www.cse.ucsd.edu/groups/bioinformatics/software.html

Contact: mchaisso@bioinf.ucsd.edu; ppevzner@cs.ucsd.edu; htangg@cs.ucsd.edu

1 INTRODUCTION

The Sanger sequencing method has enjoyed great success since its debut in 1977 as it is used in virtually all sequencing projects. The amount of sequence information has exponentiated due to advances in automated sequencing technology and investment from the Human Genome Project. State-of-the-art automated sequencers can sequence a staggering 3 mb/day. However, despite its strong points, there are some drawbacks to Sanger sequencing. First, it is dependent on clone libraries for sample preparation, and some genomes have regions not amenable to cloning. Also, future genomics projects may require sequence-throughput orders of magnitude faster than what is currently possible. Because of this, researchers are searching for alternative sequencing techniques that would bypass the shortcomings of Sanger sequencing (Ronaghi *et al*., 1998; Böcker, 2003a; Drmanac *et al*., 1989; Preparata and Upfal, 2000; Mitra *et al*., 2003; Braslavsky *et al*., 2003). These techniques aim at clone-free sequencing to enable sequencing of regions that are not amenable to cloning and to decrease dramatically the time and cost of sequencing. Almost all the new techniques produce short reads, i.e. 80–200 bases, as opposed to the 500–750 length reads produced by Sanger sequencing. It is hardly believed that such short reads can be assembled efficiently; and efforts are being made to increase the read length. In this paper, we examine the limits of assembling short reads and estimate the amount of additional finishing effort that will be be required for assembly.

The assembly of short reads is applicable directly to several new non-electrophoretic sequencing technologies being developed, such as Pyrosequencing™ (Ronaghi *et al*., 1998; Ronaghi, 2001), mass spectrometry-based sequencing (Böcker, 2003a), sequencing by Polonies (Mitra *et al*., 2003) and single-molecule sequencing (Braslavsky *et al*., 2003). These have the goal of sequencing genomes several orders of magnitude faster than what is possible with electrophoretic methods. The feasibility of using non-electrophoretic methods to sequence a long DNA molecule was demonstrated recently by the 454 Life Sciences Corporation by the sequencing of Adenovirus. This resulted in the first DNA sequence entry in GenBank that was sequenced by a technique alternative to Sanger sequencing (454 Life Sciences Corporation press release). The reads produced by the new sequencing techniques differ in both read length and quality profiles characteristic of electrophoretic sequencing. In particular the read length is sacrificed for much higher throughput, resulting in high coverage with short reads.

The success of the new high-throughput whole-genome shotgun sequencing technologies will depend not only on the speed of the analytical process of producing reads but also the solution of the computational problem of assembling sequence fragments into complete genomes. The common approach in many fragment assembly programs, such as Phrap (Ewing and Green, 1998), CAP (Huang and Madan, 1999), PCAP (Huang *et al*., 2003) and TIGR assembler (Sutton *et al*., 1995), is the overlap–layout–consensus approach. The new genome assembler from Celera (Myers *et al*., 2000) and the ARACHNE assembler (Batzoglou *et al*., 2002; Jaffe *et al*., 2003) mask the repeat regions in the reads in order to reduce the complexity of the overlap graph and alleviate the difficulty of the layout stage. This idea has been applied successfully to assembling large eukaryotic genomes with whole-genome shotgun reads, including fruitfly

*To whom correspondence should be addressed.

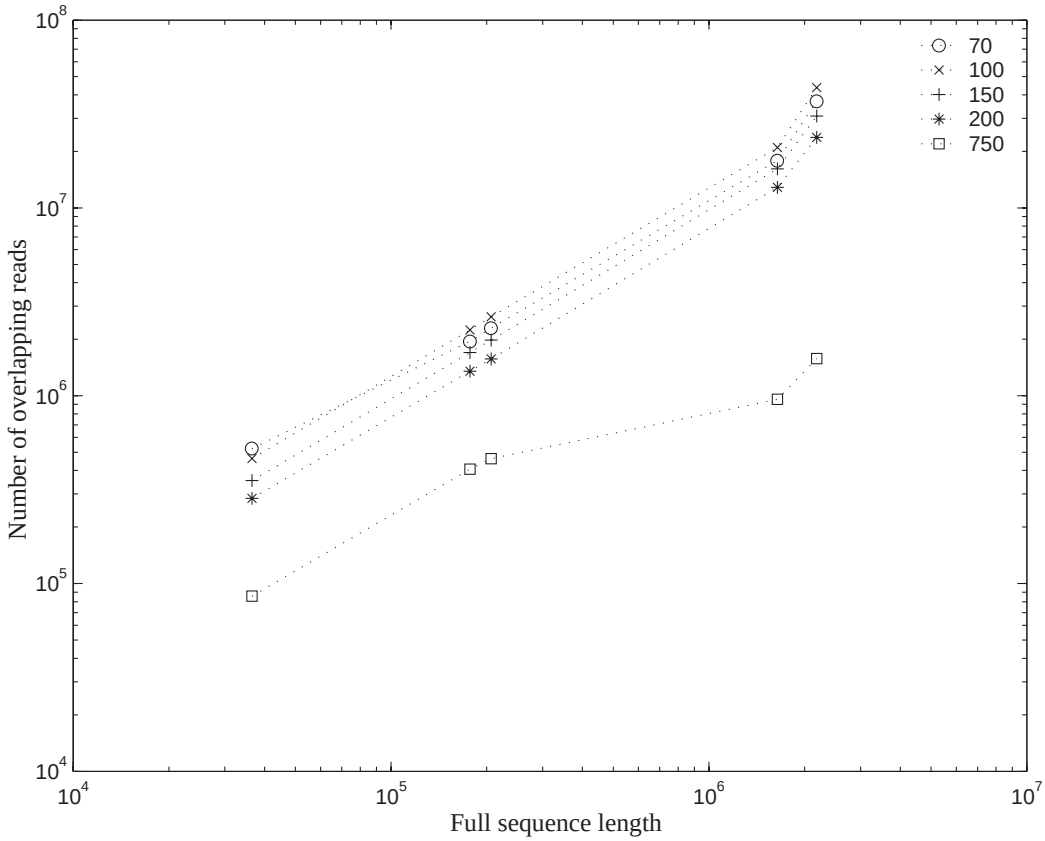

Fig. 1. Number of overlaps versus complete sequence (genome) length. Data were sampled at 30× coverage from Adenovirus, *Campylobacter jejuni* (CJ), NM and two BACs. Overlaps were found using megablast (Zhang *et al*., 2000) and filtered for exact overlap alignments at least 30 nt long. Read length 750 was chosen as a reference of the assembly complexity when assembling reads with the current read length.

(Adams, 2000), human (Venter *et al*., 2001) and mouse (Mouse Genome Sequencing Consortium, 2002). The major difficulty in assembly is the presence of repeats that are longer than the typical read length. Masking out repeats in such an approach leads to a fragmentation of the overlap graph. Therefore, the shorter the read length the larger the number of repeats that present difficulties to assembly and a greater enhancement of loss of information of the repeat masking procedure in the ARACHNE and Celera assemblers. To demonstrate the increase in the assembly complexity with the decrease in the read length, we computed the number of perfect read-end overlaps found in hypothetical sequencing projects with varying read lengths, shown in Figure 1. While for a given genome the number of overlapping reads scaled roughly linearly with read length, the number of repeat pairs increased dramatically in the larger and more complex *Neisseria meningitidis* (NM) genome. With short reads, assembling a BAC became as complicated as assembling a bacterial genome with normal reads, even when reads had no sequencing errors. The assembly of the NM genome, known to have a large number of repeats, became a formidable problem. In practice, it was difficult to assemble even very short genomes with short reads using Phrap[1], an excellent program for assembling BACs with normal read lengths. Therefore scaling short read assembly to longer and longer genomes presents a difficult computational problem. The recent announcement of the Adenovirus sequence by 454 Life Sciences Corporation did not reveal the computational approach behind this advance. However, we remark that the Human Adenovirus E has only one repeated region 51 bases long, and can otherwise be modeled as a 'random' sequence. This 51-bp-long repeat does not present a problem to assembly because it can be bridged by even extremely short reads; Phrap was able to assemble this with no modifications to the code. Moreover, one would be able to assemble the equivalent of a human genome with short reads if the human genome sequence followed a random distribution. Therefore, even after the 454 Life Sciences Corporation announcement of the Adenovirus genome, sequence assembly of short reads

[1]Phrap produced only a few very short (100 bp) contigs as well as a core-dump when assembling a BAC with short reads. The Adenovirus genome was assembled using less than 20× coverage.

for more complex genomes (even BACs) remains an open problem.

In an extreme case of high enough coverage and short enough reads, the fragment assembly problem becomes effectively the same as sequencing by hybridization. In this extreme case the read length is fixed (e.g. $l = 20$), and there is one read starting at every position in the genome. The Eulerian path approach to solving sequencing by hybridization was presented by Pevzner (1989), and proposed for the assembly of regular reads by Idury and Waterman (1995). This has been implemented in the EULER fragment assembly program (Pevzner *et al*., 2001) and further developed by Pevzner and Tang (2001). EULER has an advantage over assemblers based on an overlap–layout approach since it works meticulously with repeats and instead of masking repeats generates a repeat graph that reveals the underlying structure of the repeats within a genome. The repeat graph can direct biologists toward experiments that allow the resolution of repeats (Pop *et al*., 2002).

2 ALTERNATIVE SEQUENCING TECHNOLOGIES

2.1 Pyrosequencing

Pyrosequencing™ (Ronaghi *et al*., 1998; Ronaghi, 2001) is a non-electrophoretic DNA sequencing technique that uses sequencing-by-synthesis. In this method, a solution of an amplified sequence and the corresponding primer is mixed with four enzymes: polymerase, ATP sulfurylase, luciferase and apyrase. Sequencing is carried out by adding iteratively solutions of alternating nucleotides (A,C,T,G,A,C,T,G...). When a nucleotide is incorporated into the template strand by the polymerase, PPi is released and converted subsequently into ATP by the ATP sulfurylase. This drives a light emitting luciferin oxidation that can be detected by a light sensor. Unincorporated nucleotides are degraded by apyrase. The light intensities measured upon the addition of nucleotides are combined into a pyrogram from which the sequence can be read. Pyrosequencing has been used effectively to sequence short primer extensions for SNP detection (Fakhrai-Rad *et al*., 2002). It remains to be seen what the error rates are of Pyrosequencing; however, it is feasible that they are higher than those of electrophoretic methods.

2.2 Sequencing by matrix-assisted laser desorption–ionization time of flight (MALDI–TOF)

A recent development in *de novo* DNA sequencing by MALDI–TOF mass spectrometry has been described by Böcker (2003a). In this technology, sequences of up to 200 nt are digested partially by base-specific endonucleases, and the resulting fragments are analyzed through mass spectrometry. Each peak in the mass spectra corresponds to a number of nucleotides called compomers. By solving the sequencing by compomers problem proposed by Böcker (2003a), the spectra are resolved into high-quality reads. The detection of compomers through MALDI–TOF has been used for SNP discovery (Böcker, 2003b) and is implemented in the SEQUENOM MassArray platform.

3 ASSEMBLY OF ERROR-FREE READS

Errors in read data greatly complicate the task of fragment assembly. It is possible to correct many of the errors in reads prior to assembly. However as a first step in proof of concept for short read assembly it is necessary to show the upper bound on the resolution that can be achieved in assembly of short reads. A modified version of EULER was used to assemble the reads.

As described by (Pevzner *et al*., 2001; Pevzner and Tang, 2001) the Eulerian approach to fragment assembly resolves a sequence by finding an Eulerian path through a de Bruijn graph representation of a genome. Let G_l (G_{l-1}) be the set of l ($l-1$)-tuples present in a genome G. To form the de Bruijn graph, a vertex, v, labeled $\langle t_1 \cdots t_{l-1} \rangle$ is created for each ($l-1$)-tuple in G_{l-1}, and for each l-tuple $t \in G_l$ a directed edge ($\langle t_1 \cdots t_{l-1} \rangle$, $\langle t_2 \cdots t_l \rangle$) is added. The sequence corresponds to an Eulerian path through this de Bruijn graph. In a sequencing project, G_l (and therefore G_{l-1}) is approximated by the set of l-tuples present in a set of reads. Repeats in a genome create tangles in the de Bruijn graph (Pevzner and Tang, 2001). The equivalent transformation approach proposed by Pevzner *et al*. (2001) uses reads and mate-pair information if available to define paths through the read-generated de Bruijn graph and thus resolves many tangles. Tangles that remain require additional finishing experiments to be resolved.

In this study we assembled simulated reads from several test viral genomes, BACs and bacterial genomes: Adenovirus (Adeno), BAC 85H09, BAC 47A01, CJ and NM.[2] The genomes were sampled for 30× coverage, and ignoring end effects, all test cases had full coverage. The contigs resulting from assembly for various read lengths are listed in Table 1. The number of contigs drops at a decreasing rate with increasing read length. BAC 47A01 contains many low-complexity regions that were difficult to assemble with the shortest read length. Also, the number of contigs may be artificially large since many of the contigs generated for short reads are short and most of the genome is contained in a few long contigs. For example, the largest nine contigs give 95% coverage using reads averaging 70 bases for BAC 85H09; the largest 10 contigs for BAC 47A01. In CJ, 21 contigs produced over 95% coverage when assembling reads averaging length of 70, while 11 contigs covered at least 95% of the genome when assembled with 750 base reads. The complex nature of the NM genome was reflected in the smaller size of

[2]NCBI accession: (Adeno) AF394196, (85H09) AC084390, (47A01) AC123854, (CJ) CJ11168X1, (NM) NMA1Z2491.

Table 1. Assemblies of sample genomes using read lengths averaging 70, 100, 150, 200 and 750, all ± 10 bases

Genome [length (kb)]	70 bp			100 bp			150 bp			200 bp			750 bp		
	rep	*asm*	≥1K(%)	*rep*	*asm*	≥1K(%)	*rep*	*asm*	≥1K(%)	*rep*	*asm*	≥1K(%)	*rep*	*asm*	≥1K(%)
Adeno (35)	1	1	1 (100)	1	1	1 (100)	1	1	1 (100)	1	1	1 (100)	1	1	1 (100)
85H09 (177)	16	25	10 (98.9)	10	20	7 (98.9)	10	12	5 (99.5)	7	12	5 (99.6)	1	1	1 (100)
47A01 (206)	69	59	13 (98.2)	25	28	6 (98.9)	29	13	5 (99.0)	13	13	4 (99.3)	1	1	1 (100)
CJ (1662)	73	72	37 (99.5)	63	58	34 (99.6)	54	48	29 (99.5)	45	48	22 (99.5)	22	27	22 (99.9)
NM (2184)	1803	1864	309 (93.4)	1299	1247	299 (94.6)	563	296	215 (96.2)	266	296	114 (97.8)	92	59	48 (99.7)

Paths consisting entirely of nodes of degree 2 are collapsed into one edge, representing the theoretical limit of resolution using the specified read length. For each read length, the following are reported: *rep*, the number of edges in the repeat graph (de Bruijn graph) constructed from the test genome using tuples of size equal to the average read length +10; *asm*, the total number of contigs generated from assembling simulated reads; and ≥1K(%), the number of contigs >1 kb and the percentage of the genome the contigs cover.

contigs; 95% coverage required 344 contigs assembled from 70 bp reads (20% of all contigs generated); however, only 28 contigs were required for this coverage when assembled from 750 base reads.

4 ASSEMBLY OF READS WITH ERRORS

As mentioned by Pevzner and Tang (2001) the EULER assembly method works well on error-free reads; however, sequencing errors in reads can complicate the de Bruijn graph, making it difficult to determine the true sequence. This can be alleviated by detecting and fixing errors prior to sequencing.

Rather than using alignment and base-calling during a post-processing (consensus) step of fragment assembly, error correction is performed prior to assembly by solving the error correction problem (Pevzner *et al*., 2001). In this problem, let G_l be the set of l-tuples in a genome G. A read that has errors will contain l-tuples not in G_l and is transformed so that it only contains l-tuples in G_l. The transformation used is the spectral alignment problem (SAP) (Pevzner *et al*., 2001). In the SAP, one is given a collection of l-tuples, T, and a string. A string is called a T-string if all its l-tuples belong to T. Let $\mathcal{T}$ be the collection of all T-strings. The SAP is to find $s^\star$ such that the edit distance $d(s, s^\star)$ is minimal for all $s^\star \in \mathcal{T}$. Error correction is performed by setting T to G_l and applying spectral alignment to each read. In *de novo* sequencing, G_l is not known, and an approximation is made by choosing a threshold M and then labeling any l-tuple present more than M times throughout all reads as 'solid', and 'weak' otherwise. A read is solid if all tuples in it are solid. An example of the distribution of l-tuple multiplicities for sample reads from the CJ genome is given in Figure 2. For this genome, we can use a threshold $M = 10$ that will eliminate, false positives, erroneous l-tuples that should not be in G_l, and minimize false negatives and l-tuples in G_l that do not have the multiplicity to be solid.

An approximation to the solution for the SAP is used currently in the EULER assembler. In the approximation, all unambiguous changes are made that can transform weak l-tuples into solid ones. This method performs well on the read lengths and error profiles of modern sequencing projects. High coverage, repeats and high error rates make it difficult to determine which l-tuples should belong in T. Fluctuations in coverage by shotgun sampling require a low threshold M so that valid regions are not considered weak. Unfortunately, with a large number of reads and high error rates, this will also include many false positives. Increasing l will decrease the false positive rate, but in reads with high error rates this makes it difficult to determine which changes are appropriate for transforming the read into a T-string. We present a dynamic programming solution that allows for choosing a small M while eliminating many false positives since it considers multiple possible modifications to a string before selecting the optimal ones.

5 DYNAMIC PROGRAMMING SOLUTION TO THE SAP

In the context of resequencing by hybridization, Pe'er *et al*. (2002) studied a problem similar to the SAP where a dynamic programming solution was presented, although it was not formulated as the SAP. This approach finds the highest probability path through a de Bruijn graph constructed from probe matches in an universal array. While the approach taken by Pe'er *et al*. (2002) is similar to ours, it cannot be applied directly here. First, the scoring scheme is probabilistic rather than discrete, although this may be updated easily to fit the edit distance score presented here. More importantly, the search for optimal path transitions from 4^l possibilities is considered at each iteration. This is computationally tractable for universal arrays, where l can range from 6 to 9; however, in the case of error correction, such small values of l can lead to saturation of the T-spectrum so that nearly all 4^l l-tuples are solid. In this approach the values of l range from 15 to 20, and so it is important to only search through a small subset of the 4^l possibilities. Here we describe a complete solution to SAP and give a heuristic that solves the SAP in most cases without searching this complete search space.

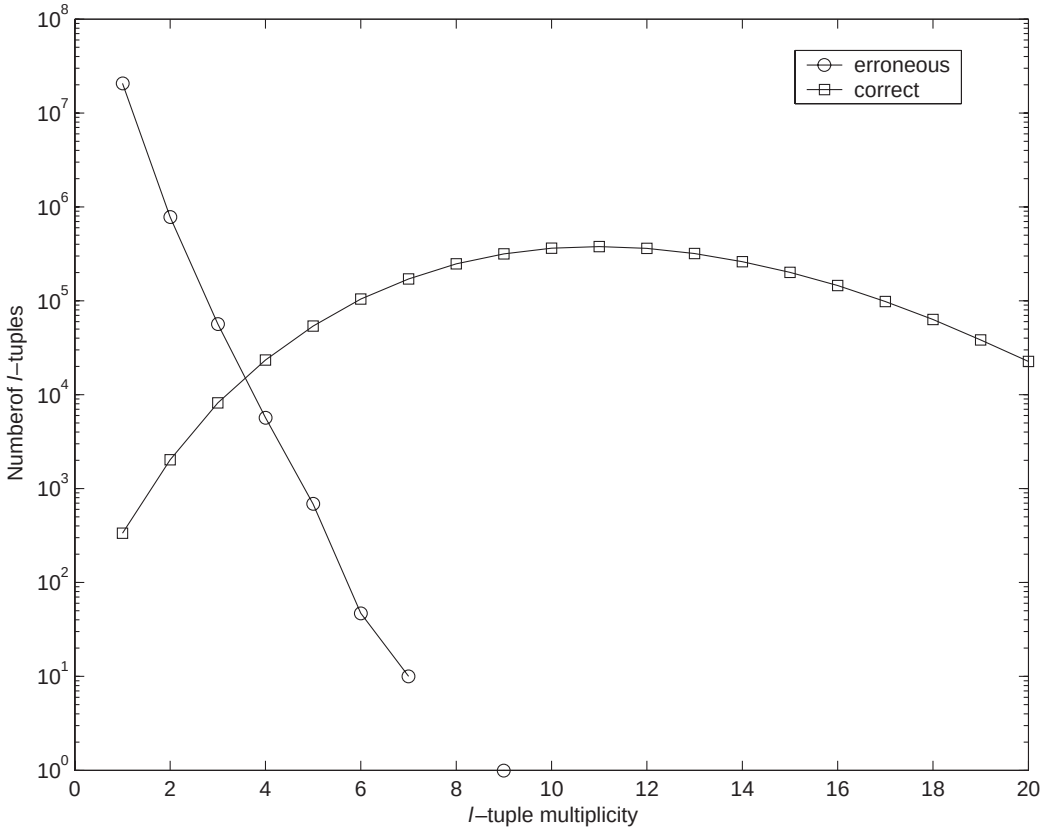

Fig. 2. Number of erroneous and correct l-tuples found in simulated reads from the CJ genome versus multiplicity. The genome was sampled at 30× coverage with average read length 100. Reads were corrupted with a 2% error rate uniformly distributed between mutations and insertions/deletions. Most erroneous l-tuples are present in low multiplicities.

In our solution we use the notation $\vec{a} = a_1, \ldots, a_{l-1}$ to denote the $l-1$ tuple from the first $l-1$ characters in l-tuple $a = a_1, \ldots, a_l$. The concatenation of the character X with $\vec{a}$ is denoted $X \circ \vec{a}$. For example, if $a = ACGT$ and $X = G$, then $X \circ \vec{a} = GACG$. For a string s and an l-tuple a from spectrum T, we define score(i, a) to be the minimum edit distance for the prefix of s of length i and all T-strings ending in a [if $a \notin T$, score(i, a) is infinity]. This may be computed using the following recurrence relation:

$$\text{score}(i,a) = \min_{X \in \{A,C,T,G\}} \begin{cases} \text{score}(i-1, X \circ \overrightarrow{a}) + \begin{cases} 1 & \text{if } a_l \neq s_i \\ 0 & \text{otherwise} \end{cases} \\ \text{score}(i-1, a) + 1 \\ \text{score}(i, X \circ \overrightarrow{a}) + 1 \end{cases}$$

The initial condition is score$(0, a) = 0$. Denote the length of string s as $|s|$. The minimum of score$(|s|, a)$ for all $a \in T$ corresponds to the minimum cost T-string for s. This recursion relation may be represented as a graph where each vertex corresponds to a prefix length i and an l-tuple a. Edges connect vertices referenced by a recurrence relation. The in-degree of a vertex is equal to the number of non-infinite vertices referenced to compute its score (for $i = 0$ the score is 0), and the out-degree is equal to the number of times the vertex is used in a recurrence relation (0 for vertices corresponding to $s_{|s|}$). The minimum score T-string corresponds to a shortest path from any vertex with in-degree of 0 to any vertex with out-degree of 0. All edges have non-negative weight, and so there are no negative cycles, and the shortest path may be computed using any shortest path algorithm such as Bellman–Ford (Corman *et al*., 1990). An example of the graph representation of the recurrence relation is shown in Figure 3.

We applied heuristic approximations to the SAP in order to improve performance in fixing errors in projects using short reads. When performing spectral alignment for reads with a known genome, the spectrum is set to G_l. Since G_l is not known in a *de novo* sequencing project, it is approximated by the set G_l^M, all l-tuples with multiplicity not less than M within all reads. Although the size of G_l^M is smaller than the total possible 4^l l-tuples, it is desirable to reduce the search space for T-strings even further. If a string s contains a substring u that is a T-string, we only consider T-strings containing u. The scoring function for this is score$^u(u, i, l)$, the minimum edit distance for string s to all T-strings containing u. Since this is easily computed if u is a prefix of s, we can solve the SAP for a substring of s such that u is a prefix of s. In our

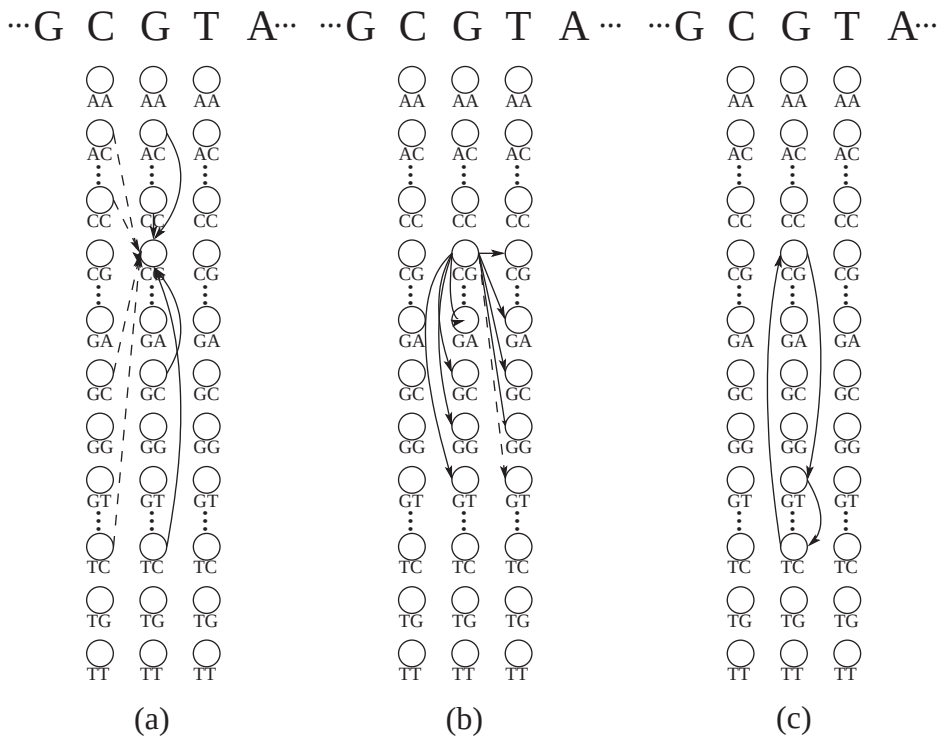

Fig. 3. Example of a graphical solution to the SAP. Dashed edges have a score of 0 and solid edges 1. Left, vertices used to compute the score of vertex CG. Center, vertices that reference CG. Right, a positive cycle. The complete cycle will not be used in a shortest path.

implementation we set u to be the first solid substring in s of length $l+\delta$ (in our application δ is about $\frac{1}{6}l$). Furthermore, the search space may be reduced when there is some intuition of the maximum number of errors expected in a read. If no more than Δ errors are expected in a read, paths of score greater than Δ may be ignored.

In a high-coverage project, a read covering a high-multiplicity repeat may contain errors that are not corrected using the spectrum G_l^M. Rather than changing G_l^M for repeat regions, we note that the nucleotide alphabet is small, and the multiplicity of consecutive l-tuples generally does not decrease rapidly. An example is the case of an independent identical distribution of nucleotides immediately after a repeat region. To account for this, we let $\text{score}^u(u,i,a)$ be the minimum edit distance from s to all T-strings ending in a that contain u and do not contain adjacent l-tuples with a ratio of multiplicities less than τ. Since this is a rare case, we generally set τ to 0.01.

A read with minimum score less than Δ is replaced by the corresponding minimum score T-string. If it is not possible to transform a read into a T-string in under Δ changes, it is considered unfixable, and no changes to the read are made. Likewise, it is ambiguous if there are multiple paths with the same cost, and such reads are also considered unfixable. Finally, it is difficult to determine if changes made near the ends of reads are correct, and so reads with erroneous edges are trimmed. After error correction is applied to the set of reads, G_l^M will change. The spectrum G_l^M may be updated and used iteratively to correct reads that were unfixable in a previous iteration. Reads that remain unfixable in the last iteration are discarded.

In order to test our error correction algorithm, simulated reads from sequenced genomes were corrupted with errors and subsequently treated with error correction. Two different error models were used, uniform error distribution, and homopolymeric errors. In the former, errors are distributed uniformly among positions in a read at a specified rate. Errors are distributed randomly between insertions, deletions and mutations, with rates ranging over 0.5, 1, 1.5, 2, 2.5, 3, 3.5, 4, 4.5 and 5%. In Pyrosequencing, there may be difficulty in sequencing homopolymeric regions (Ronaghi, 2001) of >5–6 nt, and the latter error model attempts to simulate this. Homopolymeric regions longer than 6 nt were either increased or decreased in length randomly, with a bias toward a decrease.

The results of running error correction on simulated reads at various error rates are presented in Table 2. An additional 'fictitious' genome was created to show the operation of error correction on random genomes. The fictitious genome is a 50 kb randomly generated sequence, with 28 exact repeats ranging from 80 to 1000 bases. Exact repeats were used rather than inexact ones, so that the l-tuple multiplicities of repeats were greatly enhanced over random regions. Errors in such

Table 2. The result of running error correction on erroneous 120 ± 10 bp reads

Simulated error rate	Fictitious				85H09				47A01			
	i	*r*	*c*	%	*i*	*r*	*c*	%	*i*	*r*	*c*	%
0.5	10.2	54	37	99.2	302	354	277	99.5	35.3	907	717	99.6
1.0	30.7	46	17	99.2	603	948	754	99.4	70.4	1658	1276	99.4
1.5	20.6	90	37	98.7	892	1364	1080	98.8	104	2831	2348	98.9
2.0	40.0	123	35	97.5	117	2255	1836	97.9	137	3735	3089	98.0
2.5	48.7	180	77	96.2	141	3129	2524	95.8	165	5278	4376	96.2
3.0	55.8	233	106	92.3	162	3884	3154	93.3	190	8138	6759	93.8
3.5	60.6	256	101	88.8	175	4910	3942	88.8	206	9593	8009	89.3
4.0	62.0	280	132	82.2	180	6468	5353	82.5	213	10 511	8791	83.2
4.5	60.4	332	137	73.7	174	5797	4775	73.2	209	12 149	10 225	74.8
5.0	57.2	353	156	64.7	157	5942	4832	61.2	187	12 474	10 512	62.6
Homopolymeric	17.3	274	223	99.3	55.9	5846	4144	99.6	58.9	4148	3039	93.9

For each genome the following are reported: *i*, the number of errors in kb found prior to error correction in reads retained after error correction; *r*, the number of remaining (unfixed) errors, in bases; *c*, the the number of data corruptions (nucleotides that were correct in the original read and incorrect in the fixed read), in bases %; and the percentage of reads retained after error correction.

regions could often only be detected using the multiplicity ratio threshold. Our error correction method is more effective on the random dataset. Upon inspection of the error-corrected reads for the BACs, many of the errors that were unfixed or corrupted were located in regions of low complexity. In high coverage projects, the correct modifications to make are further obscured in low-complexity regions because many erroneous *l*-tuples appear with high multiplicity. Also, many of the corruptions in low-complexity regions were in homopolymeric regions, unfortunately the regions that are most likely to be missequenced in sequencing-by-synthesis. We believe that this problem can be fixed by changing the scoring model to accommodate a bias toward indels within homopolymeric regions. However, such a change will be postponed until a true error distribution of this sequencing technology is known.

While this method works well on short reads, it does not scale well to longer reads. The expected number of errors per read increases with read length, and consequently the size of the search space increases. For high error rates on long reads the size of the search space becomes prohibitive.

After error correction was performed on the test data, the retained (corrected) reads were assembled using a new version of EULER (Pevzner *et al*., 2004). This version can detect some of the topological features that are added to the de Bruijn graph when there are sequencing inconsistencies (errors or a heterozygous sample) that otherwise fragment the repeat graph. The errors and corruptions after error correction complicate the repeat graph. Results of assembling the fictitious genome and BACs 47A01 and 85H09 with an average read length of 120 and varied error rates are summarized in Table 3. To obtain 95% coverage for BAC 85H09, 29 contigs (38% of the total contigs) were required at a 3.0% simulated error rate. BAC 47A01 contained more difficult-to-fix low-complexity regions and required 109 (48% of total) contigs for 95% coverage. With very low error rates, 0.5% BAC 85H09 required only eight contigs for 95% coverage.

Table 3. Number of contigs for assembled reads at increasing error rates

Simulated error rate	Fictitious	85H09	47A01
0.5	29	29	47
1.0	27	25	67
1.5	29	39	69
2.0	24	38	84
2.5	32	86	122
3.0	37	74	225
3.5	41	122	72
4.0	56	184	200
4.5	74	183	116
5.0	84	355	460
Random walk	28	41	189

Genomes were sampled with 30× coverage with length of 120 ± 40.

6 CONCLUSIONS

We have examined the feasibility of assembling reads produced by emerging sequencing technologies. While many large contigs may be assembled using even very short reads, substantial (if not prohibitive) finishing efforts are required for resolving entirely all but the simplest of genomes. This is especially true in the assembly of reads that have high error rates that may be a feature of some of these technologies.

Finally, we note that a powerful tool used in current sequencing projects is the use of mate-pairs for repeat resolution (Weber and Myers, 1997; Pevzner and Tang, 2001). The Eulerian approach to fragment assembly is well suited for

mate-pair analysis (Pevzner and Tang, 2001), and it is easy to incorporate mate-pairs in our algorithm for short read assembly. However, since some of the new technologies aim to produce genomes using clone-free data, we did not use mate-pairs in the existing version of the assembly algorithm. The resolution described by repeat graphs in Table 1 dictates that an extra experimental step is required to finish a sequencing project. It remains a challenge for molecular biologists to determine a high-throughput method for producing data similar to mate-pair information without the tedious task of producing a clone library. We emphasize, however, that due to the extreme complexity of the repeat graph for short reads, there may be a need for more accurate insert distance estimates, and it presents yet another challenge for developers of short read sequencing. Alternatively, the repeat structure that is assembled after equivalent transformation can provide information for a high-throughput mapping technology. We are eager to see the progress of this field.

ACKNOWLEDGEMENTS

We would like to thank Drs Mostafa Ronaghi, Lei Du and Dirk van den Boom for useful comments regarding new sequencing technologies.

REFERENCES

454 Life Sciences Corporation press release. (2003) 454 Life Sciences Completes First Whole Genome Sequence Using First Novel Technology Since 1977 Invention of DNA Sequencing.

Adams,M.D. (2000) The genome sequence of *Drosophila melanogaster*. *Science*, **287**, 2185–2195.

Batzoglou,S., Jaffe,D.B., Stanley,K., Butler,J., Gnerre,S., Mauceli,E., Berger,B., Mesirov,J.P. and Lander,E.S. (2002) ARACHNE: A whole-genome shotgun assembler. *Genome Res.*, **12**, 177–189.

Böcker,S. (2003a) Sequencing from compomers: using mass spectrometry for DNA *de-novo* sequencing of 200+ nt. *Lect. Notes Comput. Sci.*, **2812**, 476–497.

Böcker,S. (2003b) SNP and mutation discovery using base-specific cleavage and MALDI-TOF mass spectrometry. *Bioinformatics*, **19**, 44i–53i.

Braslavsky,I., Hebert,B., Kartalov,E., and Quake,S.R. (2003) Sequence information can be obtained from single DNA molecules. *PNAS*, **100**, 3960–3964.

Corman,T., Leiserson,C. and Rivest,R. (1990) *Introduction to Algorithms*. MIT Press, Cambridge, MA.

Drmanac,R., Labat,I., Brukner,I. and Crkvenjakov,R. (1989) Sequencing of megabase plus DNA by hybridization: theory of the method. *Genomics*, **4**, 114–28.

Ewing,B. and Green,P. (1998) Base-calling of automated sequencer traces using PhredII. Error probabilities. *Genome Res.*, **8**, 186–194.

Fakhrai-Rad,H., Pourmand,N. and Ronaghi,M. (2002) Pyrosequencing™: an accurate detection platform for single nucleotide polymorphisms. *Hum. Mutat.*, **19**, 479–485.

Huang,X. and Madan,A. (1999) CAP3: a DNA sequence assembly program. *Genome Res.*, **9**, 868–877.

Huang,X., Wang,J., Aluru,S., Yang,S.-P. and Hillier,L. (2003) PCAP: a whole-genome assembly program. *Genome Res.*, **13**, 2164–2170.

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

Jaffe,D.B., Butler,J., Gnerre,S., Mauceli,E., Lindbad-Toh,K., Mesiror,J.P., Zody,M.C. and Lander,E.S. (2003) Whole-genome sequence assembly for mammalian genomes: Arachne 2. *Genome Res.*, **13**, 91–96.

Mitra,R., Shendure,J., Olejnik,J., Edyta-Krzymansk-Olejnik, and Church,G. (2003) Fluorescent *in situ* sequencing on polymerase colonies. *Anal. Biochem.*, **320**, 55–65.

Mouse Genome Sequencing Consortium (2002) Initial sequencing and comparative analysis of the mouse genome. *Nature*, **420**, 520–562.

Myers,E.W., Sutton,G.G., Delcher,A.L., Dew,I.M., Fasulo,D.P., Flanigan,M.J., Kravitz,S.A., Mobanry,C.M., Reinert,K.H., Renuington,K.A. (2000) A whole-genome assembly of Drosophila. *Science*, **287**, 2196–2204.

Pe'er,I., Arbili,N. and Shamir,R. (2002) A computational method for resequencing long DNA targets by universal oligonucleotide arrays. *PNAS*, **99**, 15492–15496.

Pevzner,P. (1989) *l*-tuple DNA sequencing: computer analysis. *J. Biomol. Struct. Dyn.*, **7**, 63–73.

Pevzner,P. and Tang,H. (2001) Fragment assembly with double-barreled data. *Bioinformatics*, **17**, S225–S233.

Pevzner,P.A., Tang,H. and Waterman,M.S. (2001) An Eulerian path approach to DNA fragment assembly. *PNAS*, **98**, 9748–9753.

Pevzner,P., Tang,H. and Tesler,G. (2004) *De novo* repeat family classification and fragment assembly. In *RECOMB 2004 Proceedings of the Eighth Annual International Conference on Research in Computational Molecular Biology*, pp. 213–222.

Pop,M., Salzberg,S. and Shumway,M. (2002) Genome sequence assembly: algorithms and issues. *IEEE Comput.*, **35**, 47–54.

Preparata,F.P. and Upfal,E. (2000) Sequencing-by-hybridization at the information-theory bound: an optimal algorithm. *J. Comput. Biol.*, **7**, 621–630.

Ronaghi,M. (2001) Pyrosequencing sheds light on DNA sequencing. *Genome Res.*, **11**, 3–11.

Ronaghi,M., Mathias,U. and Nyrén,P. (1998) DNA sequencing: a sequencing method based on real-time pyrophosphate. *Science*, **281**, 363–365.

Sutton,G., White,O., Adams,M. and Kerlavage,A. (1995) TIGR assembler: a new tool for assembling large shotgun sequencing projects. *Genome Sci. Tech.*, **1**, 9–19.

Venter,J.C., Adams,M.D., Myers,E.W., Myers,E.W., Li,P.W., Mural,R.J., Sutton,G.G., Smith,H.O., Yandell,M., Evans,C.A., Holt,R.A. The sequence of the human genome. *Science*, **291**, 1304–1351.

Weber,J.L. and Myers,E.W. (1997) Human whole-genome shotgun sequencing. *Genome Res.*, **7**, 401–409.

Zhang,Z., Schwartz,S., Wagner,L. and Miller,W. (2000) A greedy algorithm for aligning DNA sequences. *J. Comput. Biol.*, **7**, 203–214.