Correcting Base-Assignment Errors in Repeat Regions of Shotgun Assembly

Degui Zhi, Uri Keich, Pavel Pevzner, Steffen Heber, and Haixu Tang

Abstract—Accurate base-assignment in repeat regions of a whole genome shotgun assembly is an unsolved problem. Since reads in repeat regions cannot be easily attributed to a unique location in the genome, current assemblers may place these reads arbitrarily. As a result, the base-assignment error rate in repeats is likely to be much higher than that in the rest of the genome. We developed an iterative algorithm, EULER-AIR, that is able to correct base-assignment errors in finished genome sequences in public databases. The *Wolbachia* genome is among the best finished genomes. Using this genome project as an example, we demonstrated that EULER-AIR can 1) discover and correct base-assignment errors, 2) provide accurate read assignments, 3) utilize finishing reads for accurate base-assignment, and 4) provide guidance for designing finishing experiments. In the genome of *Wolbachia*, EULER-AIR found 16 positions with ambiguous base-assignment and two positions with erroneous bases. Besides *Wolbachia*, many other genome sequencing projects have significantly fewer finishing reads and, hence, are likely to contain more base-assignment errors in repeats. We demonstrate that EULER-AIR is a software tool that can be used to find and correct base-assignment errors in a genome assembly project.

Index Terms—Fragment assembly, finishing, expectation maximization.

1 Introduction

In the past decade, genomes of all kinds of organisms, ranging from bacteria to mammals, have been sequenced by DNA sequencing projects. All DNA sequencing projects crucially depend on the accuracy of the base-assignment procedure. While base-assignment in nonrepetitive regions is relatively straightforward, base-assignment in repeat regions is an unsolved problem and the base-assignment error rate in repeats is likely to be much higher than in the rest of the genome. Suppose there is a 2,500 bp long, five copy repeat in a genome where each copy differs by less than 1 percent from the other copies. How would assembly algorithms process such a repeat? Unfortunately, any whole genome shotgun assembler is likely to make at least one of the following errors: 1) collapsing the repeat, thus reducing the number of repeat copies in the genome or 2) erasing differences between distinct copies of the repeat even while correctly identifying the number of copies.

These potential errors are worrisome since repeated regions are of great biological importance. Repeats are often hot spots for large-scale chromosomal rearrangements that are associated with evolution or diseases (e.g., segmental duplications in primates [10] and Alu-induced recombinations [11]). Some of the most recent evolutionary history is reflected in the subtle differences between different copies of repeats. A high-quality base-assignment procedure in a repeat region will enable researchers to address these important biological questions as well as others such as the analysis of haplotypes in repeat regions.

While the repeat collapsing problem has drawn considerable interest (e.g., [1], [2]), the *repeat base-assignment problem* has drawn significantly less attention. Some early references discussed the topics of base-assignment [3] and finishing experiment design [4], [5], but the complications from repeats were ignored. The problem of repeat base-assignment is recognized in the works by Kececioglu and Yu [6], Myers [7], and Tammi et al. [8]. However, as of this writing, none of the algorithms described in these papers generated a practical software tool that would be able to correct base-assignment errors in repeated regions. Indeed, we observe that many genome sequences in the public databases contain base-assignment errors! Even in the best finished genomes we tested, we found strong evidence for base-assignment errors in repeated regions: In the *Wolbachia sp.* [9] genome, about 16 positions have ambiguous base-assignments and two positions are erroneous.

We developed a new algorithm, EULER-AIR (Almost Identical Repeats) for the task of repeat base-assignment. Given a genome sequence and the set of shotgun reads that were used in its assembly, EULER-AIR suggests positions of potential base-assignment errors. EULER-AIR also displays an alignment of reads that are assigned around each such position. These alignments summarize essential information, based on which the finishers can either have a clear base-assignment or design additional finishing experiments to resolve any ambiguities.

- *D. Zhi is with the Bioinformatics Program, University of California, San Diego, La Jolla, CA 92093. E-mail: dzhi@ucsd.edu.*
- *U. Keich is with the Department of Computer Science, 4130 Upson Hall, Cornell University, Ithaca, NY 14853. E-mail: keich@cs.cornell.edu.*
- *P. Pevzner is with the Department of Computer Science and Engineering, University of California, San Diego, La Jolla, CA 92093. E-mail: ppevzner@cs.ucsd.edu.*
- *S. Heber is with the Department of Computer Science, 1519 Partner II (Centennial Campus), North Carolina State University, Raleigh, NC 27695-7566. E-mail: sheber@ncsu.edu.*
- *H. Tang is with the School of Informatics and Center for Genomics and Bioinformatics, Indiana University, Bloomington, IN 47408. E-mail: hatang@indiana.edu.*

Manuscript received 15 Sept. 2004; revised 20 Nov. 2005; accepted 16 Jan. 2006; published online 9 Jan. 2007.
For information on obtaining reprints of this article, please send e-mail to: tcbb@computer.org, and reference IEEECS Log Number TCBB-0147-0904.
Digital Object Indentifier No. 10.1109/TCBB.2007.1005.

2 METHODS

2.1 Preliminaries

Many genomes are sequenced by the whole genome shotgun (WGS) strategy, followed by finishing experiments for gap closure and repeat resolution. In the shotgun phase, chromosomes are sheered into fragments. Fragments with length falling into the desired range are isolated and cloned into vectors. Subsequently, the fragment ends are sequenced and the resulting reads are assembled into a scaffold.

In the finishing phase, a subset of clones is selected so that each selected clone is uniquely mapped to the genome and it covers a region which requires more resolving power. Typically, such a region would be either a low coverage one, a region containing ambiguities, or a repeat region. These clones are then used to generate additional sequencing reads by primer walking or transposon insertion.

In a DNA fragment assembly pipeline, base-assignment is done by first aligning all reads along a layout and then applying a likelihood test [3] or simply taking a majority vote. While most reads can be assigned to a unique genome location (*unique reads*), some can be assigned to multiple locations almost equally well (*repeat reads*). Existing assembly tools may assign a repeat read to a randomly chosen location.[1] Under this random assignment, many reads may not be assigned correctly, resulting in base-assignment errors and the elimination of differences between different instances of repeats.

2.2 Previous Works

Kececioglu and Yu [6] described a 4-step procedure including statistical tests to distinguish true differences between repeat copies from sequencing errors. They reduced the problem of separating repeat copies to an integer programming problem. Myers [7] focused on the combinatorial optimization algorithm to find a best bipartition of the reads that minimizes the sum of intrapartition parsimony scores among the reads, and devised branch and bound algorithms. These algorithms place more emphasis on solving theoretical problems.

Along another line, Tammi et al. [8] developed an approximate statistical test for the identification of *defined nucleotide positions* (DNPs). A pair of candidate positions are modeled together to achieve a probability distribution that better separates DNPs from sequencing errors. They also developed an algorithm for separating tandem repeats based on identified DNPs.

The above methods do not address the following points: First, real repeats often exhibit tangle structures (see Fig. 1) [16] and identifying repeat boundaries can be a challenging task. Second, these methods did not consider mate pairs from double barrelled sequencing, which is very valuable for repeat resolution.

1. New assembly tools, such as EULER [2] and ARACHNE [15], typically design sophisticated methods to resolve repeated regions, although with varied efficiency depending on the degree of similarity between repeat copies.

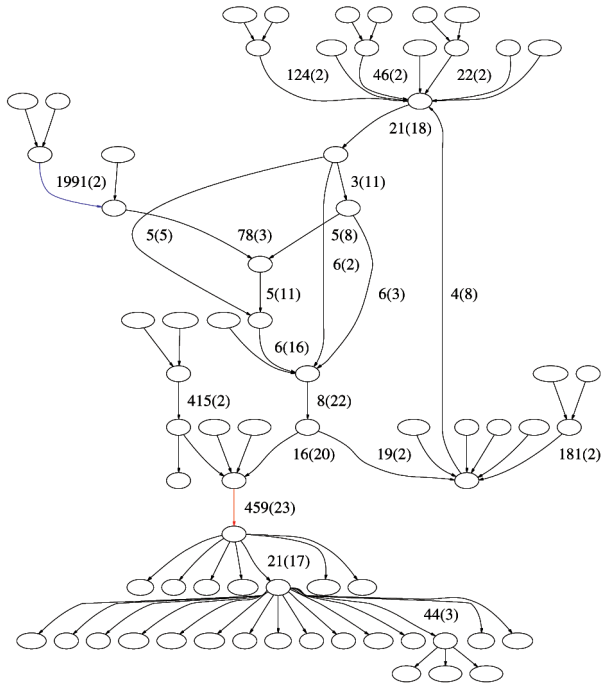

Fig. 1. Repeat graph [16] for repeat family 2 in the *Wolbachia* genome. An edge with label $x(y)$ corresponds to a subrepeat of length x and multiplicity y. Unlabeled edges have multiplicity 1 (unique regions flanking repeats). This repeat family essentially contains two subrepeats (marked by colored edges). The red edge subrepeat has 23 copies. The blue edge subrepeat has two copies and they are at the upstream regions of twp copies of the red edge subrepeat. The corrected base at 462,023 is located on the red edge.

2.3 The EM Algorithm

The problems of read assignment and base-assignment in repeat regions are a case of the "chicken and egg" problem: If the real sequences for different repeat copies are given, the assignment of reads is trivial; on the other hand, if the read assignment is given, it is easy to determine the real sequence of each repeat copy. Unfortunately, in reality, both the sequences of repeat copies and the read assignments are unknown. We therefore adopted an iterative approach. Namely, assuming we have identified some of the bases in the distinct copies of the repeat, we can assign reads based on those bases. Subsequently, once we have assigned the reads, we can reevaluate our base-assignments and repeat this procedure. Technically, this is done in an EM (expectation maximization) framework [13], where our read assignments are probabilistic, meaning a read can be assigned to multiple locations with different weights that sum up to one.

We begin with formulating the base-assignment procedure as a parameter estimation problem. Let M be the number of distinct regions (copies) of the repeat identified by the preprocessing described above. The parameters $\Theta = \{\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M\}$ are the actual bases in the regions we are trying to sequence. More precisely, $\boldsymbol{\theta}_i = \theta_{i1}, \theta_{i2}, \ldots, \theta_{in_i}$ are the n_i bases of the ith region, where each $\theta_{ij} \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$.

Imagine the following procedure to generate a random read S from the sequence Θ: Begin with randomly (uniformly) choosing the starting position of S by choosing a pair (i, j_0) (sequence and location within the sequence) and continue with a random choice of the length of S. Having determined $\{\boldsymbol{l}(j)\}$, the location vector of the read,

proceed to take a random "imprint" $\alpha \in \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$ of the subsequence $\theta_{l(j)}$ according to the rule

$$p(\alpha|\Theta, \boldsymbol{l}(j)) = P(S(j) = \alpha|\Theta, \boldsymbol{l}(j)) = \begin{cases} 1-\varepsilon & \alpha = \Theta_{\boldsymbol{l}(j)} \\ \varepsilon/3 & \alpha \neq \Theta_{\boldsymbol{l}(j)}, \end{cases} \quad (1)$$

where $\varepsilon \approx 0.02$ is the rate of sequencing error. We assume the imprint of each letter in a read is independent of the others and that reads are independent of one another.

With this model in mind, given the set of reads $\mathcal{S} = \{S_1, S_2, \ldots, S_K\}$, we can readily write down the likelihood of the sequence Θ as:

$$L(\Theta; \mathcal{S}) = P(\mathcal{S}|\Theta) = \prod_k \frac{1}{N} \sum_{\text{locations } \boldsymbol{l}} \prod_j p(S_k(j)|\Theta, \boldsymbol{l}(j)),$$

where N is the number of possible location vectors $\boldsymbol{l}$ or, more precisely, of their possible starting point (since the vector's length must equal that of the read).

In this context, base-assignment "simply" amounts to finding Θ that maximizes $L(\Theta; \mathcal{S})$. This is where expectation-maximization or EM comes in. It is an iterative procedure designed to monotonically converge to a local maximum of $L(\Theta; \mathcal{S})$. The idea is to rely on naturally associated hidden variables, in this case the location or mapping vectors, $\boldsymbol{l}_k$.

EM alternates between an E-step and an M-step. At the E-step, we try to improve our guess of the (starting position of the) location vectors based on $\tilde{\Theta}$, the previous step assignment of the parameters Θ. This is equivalent to updating the read assignment based on the current sequence. Technically, this is done using Bayes' rule:

$$P(\boldsymbol{l}_k = \boldsymbol{l}|\tilde{\Theta}, \mathcal{S}) = \frac{\prod_j p(S_k(j)|\tilde{\Theta}, \boldsymbol{l}(j))}{\sum_{\text{locations } \hat{\boldsymbol{l}}} \prod_j p(S_k(j)|\tilde{\Theta}, \hat{\boldsymbol{l}}(j))}. \quad (2)$$

While, in theory, we should compute this for all possible locations $\boldsymbol{l}_k$, in practice, the only terms which are essentially different from 0 are the ones which correspond to the initial mappings of the read (obtained by BLAST in the preprocessing stage). The same comment holds for evaluating the denominator in (2).

At the M-step, we update our estimate of the parameters Θ based on the previous assignment of the location vectors. This is equivalent to updating our base-assignments after updating the read assignment. More precisely, let $w(k, \boldsymbol{l})$ denote the right-hand side of (2) which we computed in the preceding E-step. Our updated estimate is:

$$\arg\max_\Theta \sum_k \sum_{\boldsymbol{l}} w(k, \boldsymbol{l}) \log p(S_k|\Theta, \boldsymbol{l}), \quad (3)$$

where again, the sum over $\boldsymbol{l}$ extends only over a limited predetermined options. Note that

$$\begin{aligned} &\sum_k \sum_{\boldsymbol{l}} w(k, \boldsymbol{l}) \log p(S_k|\Theta, \boldsymbol{l}) \\ &= \sum_k \sum_{\boldsymbol{l}} w(k, \boldsymbol{l}) \sum_j \log p(S_k(j)|\Theta, \boldsymbol{l}(j)) \\ &= \sum_{(i,m)} Q_{i,m}(\theta_{im}), \end{aligned}$$

where

$$Q_{i,m}(\theta_{im}) = \sum_k \sum_{\substack{\boldsymbol{l}: \boldsymbol{l}(j) = (i,m) \\ \text{for some } j}} w(k, \boldsymbol{l}) \log p(S_k(j)|\theta_{im}).$$

The task of maximizing (3) is now reduced to separately maximizing each $Q_{i,m}(\theta_{im})$, which can be achieved simply by enumerating over θ_{im}'s range of values: $\{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}\}$.

Remarks.

- The *probabilistic coverage* of the letter α in position (i, m) is

$$\sum_k \sum_{\substack{\boldsymbol{l}: \boldsymbol{l}(j) = (i,m) \\ \text{for some } j}} w(k, \boldsymbol{l}) \cdot \mathbf{1}\{\mathbf{S_k}(\mathbf{j}) = \alpha\}$$

  and it roughly measures how many reads support α at that position.
- The analysis above was made using reads; mate pairs can be treated as a single unit of assignment.
- In our experiment, we use the quality-trimmed reads so we can assume a uniformly low error rate across the entire reads. One can get a better error rate estimating by incorporating quality values in (1). We did not consider quality values in our implementation since some genome sequencing projects we tested do not provide quality values.
- In reality, the imprinting process can also insert and delete letters, especially in homopolymeric tracts. These insertions/deletions can significantly reduce the utility of our EM procedure. Our preprocessing step mentioned in Section 2.4.2 is designed to compensate for that. Effectively, it adds the gap letter, "-", to the alphabet and ensures that the reads are consistently aligned to the enriched sequence.

2.4 Implementation and Software

2.4.1 Identifying Clusters of Repeat Reads

Read assignment within a repeat is independent of read assignment in other repeats that do not overlap with it. Therefore, it suffices to consider only a cluster of repeat reads that forms an *overlap closure*, i.e., a set of reads such that any read in the set does not overlap with any repeat read outside the set. In our base-assignment procedure for repeat regions in a genome, we first identify repeat read clusters, each corresponding to a (minimal) overlap closure. We then apply our EM algorithm individually to each such cluster.

All reads are aligned onto the genome sequence using the NUCMER program in MUMMER 3.0 package [25] with default options. The beginning and ending positions of all significant alignments of a read are recorded. Only alignments with length > 200 bp are selected. A repeat overlap closure is identified by the following procedure:

1. Pick a read from the set of repeat reads (i.e., those reads which have multiple significant alignments) and paint all the genomic regions to which this read is mapped as blue.

2. Find all repeat reads that are mapped to genomic regions overlapping with the blue regions.
3. Paint all genome regions to which these repeat reads are mapped as blue.
4. Repeat Steps 2 and 3 until the blue regions do not expand.

Finally, the set of repeat reads form an overlap closure and their corresponding genomic regions form distinct copies of a repeat. To identify other repeats, remove the blue reads from the set of repeat reads and perform the above procedure with different colors.

2.4.2 Generating Multiple Alignment of Reads

Prior to applying the EM algorithm, we apply the following procedure to obtain a consistent mapping of all repeat reads to the genome. Starting with a set of repeat reads that form a minimal overlap closure, we extract its corresponding (painted) genomic regions. Each distinct copy (region) of the repeat is extended so as to include a unique flanking region of 4,000 bp (a typical insert length for a mate pair). We then apply the BLASTALL program [26] with the "-m 3" option to generate the alignments between each distinct region of the repeat and all the reads which are mapped to it. Note that many reads will be aligned to more than one region. Finally, we need to combine these pairwise alignments into a consistent alignment between each distinct genomic region of the repeat and all the reads that are mapped to it. We first parse the BLAST result and translate it into vertical format, then apply ReAligner [27] for the fine-tuning. We used a modified version of ReAligner as the original one does not handle multiple alignment of hundreds of reads.

2.4.3 Implementation of the EM algorithm

The EM algorithm is implemented in a perl program, euler_air. To handle real trace data, euler_air has the options to load the following data: 1) clone map information, 2) list of finishing reads, and 3) mate pair information. Each of these types of information can help in resolving ambiguous read mappings. Mate pair information is particularly useful for the mapping of repeat reads. If one read from a mate pair is a unique read, we can simply locate its mate pair reads with the constraint of clone insert length. If both reads in a mate pair are repeat reads, we consider the associations of their mappings that agree with the clone insert length constraint and treat them as a single unit of assignment in the EM iterations.

Since we consider each repeat overlap closure individually, the memory usage is only dependent on the largest repeat overlap closure in the genomic sequence. Each repeat overlap closure corresponds to a set of highly similar, nonoverlapping repeats in the genome. In prokaryotic and lower eukaryotic genomes, low copy number repeats are dominant and, thus, the sizes of repeat overlap closures are moderate. The current implementation may encounter memory problems when applied to large eukaryotic genomes where a large number of highly similar nonoverlapping repeats are present. In that case, it is possible to reimplement the algorithm in a more memory efficient way.

TABLE 1
Summary of Repeat Regions and Repeat Reads in the *Wolbachia* Sequencing Project

Genome region	# bases	Read type	Shotgun	Finishing
Repeat	108756	Repeat reads	1271	1153
Unique	1159026	Unique reads	12447	1542
Total	1267782	Total reads mapped	13718	2695

The result is based on our read alignment procedure (Section 2). Repeat reads: reads with ambiguous mappings; unique reads: reads with unique mapping. Repeat region: genome regions covered by repeat reads; unique region: genome regions not covered by repeat reads.

The euler_air program and the trace data handling scripts were tested under linux/unix platform and can be downloaded from our companion Web site.

3 Results

To assess the ability of EULER-AIR in separating almost identical repeats, we generate simulated data and compare the EM read assignment method of EULER-AIR with other read assignment methods. Our method gives coverage estimates that are in perfect agreement with the simulation and recovers all differences between the repeat copies (see the companion Web site for details). Below, we present the result of EULER-AIR over data from real sequencing projects.

3.1 Bacterial Genome Example: *Wolbachia*

The *Wolbachia sp.* genome is sequenced and finished at TIGR [9] by a WGS approach. Below, we use this genome to demonstrate how EULER-AIR can correct base-assignment errors. We will show that EULER-AIR can

1. discover and correct base-assignment errors,
2. provide accurate read assignments,
3. utilize finishing reads for accurate base-assignment, and
4. provide guidance for designing finishing experiments.

EULER-AIR's results for some other assembly projects are available at our companion Web site [14].

The *Wolbachia* genomic sequence and read data were taken from the genome assembly benchmark data at TIGR [9], introduced in [17]. The genome contains a single circular chromosome of length 1,267,782 bp and 8.3 percent of the genome are repeat regions. Accordingly, 9.3 percent of the shotgun reads lay entirely (or almost entirely) within repeat regions and, thus, are classified as repeat reads. The statistics of the repeat regions and repeat reads are presented in Table 1. Out of the 1,271 shotgun repeat reads, 248 form mate pairs with other repeat reads and 52 do not have verified mate pairs. These 300 reads cannot be uniquely aligned with the genome. Finishing reads are enriched with repeat reads (42.8 percent of all finishing reads are repeat reads), indicating significant efforts made by the finishers for repeat resolution. We will show that the design of the finishing experiments in this sequencing project could have been improved by EULER-AIR such that many of these finishing experiments could have been avoided, whereas additional experiments should be carried

TABLE 2
Characterization of Individual Repeat Families of the *Wolbachia* Genome Using the RepeatGluer Algorithm [16]

	repeats			# repeat reads	
index	# bases	mul.*	length**	shotgun	finishing
1	61519	17	3619	658	766
2	17059	23	742	243	114
3	13757	5	2751	194	127
4	3454	3	1151	40	24
5	2362	2	1181	24	57
6	2303	4	576	26	2
7	2058	5	412	17	4
8	1349	3	459	21	1
9	1132	2	566	11	1
sum	104993			1234	1096

*Repeat families are ordered by the total number of bases in all copies of repeats. * Multiplicity (**mul.**) of a repeat is the maximum number of mapping positions its repeat reads can have. Multiplicity characterizes the subrepeat presented in all repeat copies with length comparable to a read's length. ** **Length** of a repeat equals its number of repeat bases divided by its multiplicity. Length characterizes the average length of the copies of this repeat. This is nonetheless a crude characterization of repeats as large repeats often have copies of different length and sometimes a complex structure (e.g., the one in Fig. 1)*

out in order to resolve all of the repeat regions in the genome.

The 1,271 shotgun repeat reads are organized into clusters (see Section 2.4.1), each corresponding to a repeat family (or, in an unambiguous context, simply, a repeat). The nine largest repeats each containing more than 10 repeat reads are selected for analysis (Table 2).

The large repeats typically have a complex repeat structure consisting of shorter subrepeats of varying length and multiplicity. Fig. 1 depicts a repeat graph (defined in [16]) illustrating the complex structure of the second largest repeat in *Wolbachia*. This repeat structure contains a central 23×459 (multiplicity $\times$ length) subrepeat flanked by other subrepeats, including a long $2 \times 1,991$ one. The largest repeat contains an even more complex repeat structure: Its instances cover 40 intervals in the genomic sequence with lengths varying from 413 to 3,965 bp.

3.1.1 EULER-AIR Analysis of Shotgun Reads

Suppose we are at the end of the shotgun phase of the *Wolbachia* sequencing project when only a set of shotgun reads and an assembled scaffold (a set of contigs) from those reads are available. Here, we assume that the assemblers did not collapse any repeats so all copies of a repeat are represented in some contigs.[2] We invoke EULER-AIR with the finished genome sequence and the shotgun reads. The results are presented in the left half of Table 3.

EULER-AIR reports three erroneous positions and eight ambiguous positions in repeat regions and flanking unique regions. The correction of erroneous position 670,009 is well supported by eight shotgun reads and three finishing reads (see Section 3.1.3 for details), indicating an apparent base-assignment error in the final genome sequence. We further analyze the multiple alignment provided by people at TIGR. Apparently, finishers misplaced many repeat reads around this position because of the lack of appropriate software tools for repeat resolution. As a result, the distinguishable bases at this position were obscured by repeat reads from other copies.

The nucleotides at erroneous positions 462,023 and 547,482 are also supported by multiple reads. Moreover, the nucleotides at these positions are associated with the nucleotides at nearby ambiguous positions (462,023 with 462,031, 547,482 with 547,531 and 548,248). The reads around these positions appear to come from two distinct copies of a repeat, suggesting a possible collapsing of repeats.

The ambiguous positions are mostly in regions where read alignment is unclear. For example, out of the eight reads covering position 795,149, five have a run of three "C"s and three have a run of two "C"s. Additional finishing experiments at those locations are desired.

3.1.2 Assessing Accuracy of Read Assignment

In order to assess the accuracy of EULER-AIR's read assignment in repeat regions, we conducted the following experiment: We supplied EULER-AIR with finishing reads together with shotgun reads. The finishing reads were treated as shotgun reads even though the locations of many of those can be inferred from the locations of some unique reads from the same clone where these finishing reads were sampled. The accuracy of EULER-AIR's read assignment is thus assessed by checking if the finishing reads are placed within the range of the clone.

Table 4 shows that EULER-AIR makes few errors in assigning repeat reads. About 60 percent of the repeat reads are correctly mapped, whereas only 4 percent are wrongly mapped. The rest of the reads are ambiguously assigned because they matched completely to repeats with identical copies.

3.1.3 EULER-AIR-Finishing: Initializing EM Using Finishing

Suppose we are at the end of the finishing stage of our sequencing project, with both the sets of shotgun and finishing reads available. Although the location of the finishing reads can be inferred from the location of their finishing clones, most assemblers do not allow one to specify such clone constraints (M. Pop, *pers. comm.*) and finishing reads are just treated as additional shotgun reads.

It is, however, trivial to specify such constraints in the framework of EULER-AIR: We designed a variation of EULER-AIR, EULER-AIR-Finishing, that eliminates the mappings of finishing reads that violate the placement of their clones. Effectively, both finishing reads and unique shotgun reads are used to initialize the EM procedure.

The results of EULER-AIR-Finishing are displayed in the right half of Table 3, side-by-side with the result of EULER-AIR for comparison. For position 670,009, EULER-AIR-Finishing confirmed the result of EULER-AIR (see Section 3.1.4 for a detailed discussion). Interestingly, for positions 462,023 and 547,482, additional finishing reads support the second bases in the alignment. Even though the finishers seemed to trust the finishing reads and chose to

2. In reality, the assemblers may collapse some repeats, but, in this paper, we focus on the repeat base-assignment problem.

TABLE 3
Problematic (Erroneous and Ambiguous) Positions Suggested by EULER-AIR and EULER-AIR-Finishing

EULER-AIR					EULER-AIR-Finishing				
proposal	**pos**	**ori**	**maj**	**sec**	**pos**	**ori**	**maj**	**sec**	**conclusion**
err.	462023	A	–(5)	A(2)	462023	A	–(5)	A(4)	amb.
	547482	T	–(15)	T(1)	547482	T	–(16)	T(5)	err.
	670009	G	A(7.973)	G(1.014)	670009	G	A(10.977)	G(0.013)	err.
amb.	354879	A	A(4)	–(3)	354879	A	A(4)	–(3)	amb.
	462031	C	C(4)	–(3)	462031	C	–(5)	C(4)	amb.
	547531	A	A(7)	–(6)	547531	A	A(10)	–(6)	amb.
	548248	A	A(6)	–(4)	548248	A	A(7)	–(4)	amb.
	795149	C	–(5)	C(3)	795149	C	–(5)	C(3)	amb.
	950070	C	C(7)	G(4)	950070	C	C(10)	G(4)	good
	950071	A	A(7)	C(4)	950071	A	A(10)	C(4)	good
	964261	T	T(4)	–(2.504)	964261	T	T(7)	–(3)	good
					45129	T	T(44)	G(41)	amb.
					168579	G	G(5)	–(3)	amb.
					293316	T	G(3)	T(2)	amb.
					293317	C	G(3)	C(2)	amb.
					548714	–	–(5.993)	T(3)	amb.
					835791	T	T(3)	–(3)	amb.
					840931	C	C(5)	–(3)	amb.
					952321	A	A(35.912)	–(23)	amb.
					952353	A	A(31.912)	–(26)	amb.
					952403	A	A(25.906)	–(19)	amb.

*The problematic positions identified by EULER-AIR with shotgun reads are shown in the first column (**proposal**) and the verification of EULER-AIR-Finishing with shotgun plus finishing reads are in the last column (**conclusion**). **ori** base is the base in published sequence; **maj** and **sec** bases are the top two most frequent bases at this position. The number next to a base is its probabilistic coverage (defined in Section 2.3). Classification of a base: 1) an* ambiguous base *(**amb.**) if the coverage of the second nucleotide is larger than half of the coverage of the major nucleotide and the coverage of the major nucleotide is larger than 3; 2)* erroneous base *(**err.**) if it is not ambiguous and the major base is different from the old base and is supported by at least three reads.*

use their bases at the finished sequence, the existence of a large number of inconsistent shotgun reads at this position (16 at 547,482 and five at 462,023) should be explained.

In summary, we found that *Wolbachia* has been carefully finished, many finishing reads cover erroneous, ambiguous, and low coverage positions, and are very helpful in improving the quality of the sequences in the repeat regions. In Table 5, we show that, with finishing reads, the number of low coverage positions are reduced drastically. However, there are still 16 ambiguous bases and 47 bases of low coverage left. This suggests that there is still room for improvement for finishing experiments, especially after using specifically designed repeat resolution program, such as EULER-AIR.

TABLE 4
Accuracy of Assignment of Finishing Repeat Reads

	# finishing repeat reads				
Repeat	**total**	**in good clone**	**correct**	**wrong**	**ambiguous**
1	766	300	234	13	53
2	114	37	29	2	6
3	127	109	21	3	85
4	24	20	17	0	3
5	57	11	10	0	1
6–9	8	0	0	0	0
Total	1096	477	311	18	148

*A clone is **good** if all unique reads it contains map to genome locations close to each other. A read assignment is 1) **correct** if its true mapping location receives a probabilistic assignment (see Section 2.3) ≥ 0.8, 2) **wrong** if its true mapping location does not receive the maximum assignment among its possible mapping positions, or 3) **ambiguous** if its true mapping location receives a maximum assignment among its possible mapping positions but the assignment probability is < 0.8.*

3.1.4 A Close-Up Study

We next focus on repeat family 3 to illustrate the nature of one particular base-assignment error corrected by EULER-AIR. The RepeatGluer [16] algorithm characterizes the structure of this repeat as a $5 \times 2,378$ core subrepeat flanked by subrepeats of varying lengths. In the downloaded sequence, the difference between the five copies is extremely small. There is even a region of length 1,910 bp where all five copes are 100 percent identical!

Both results of EULER-AIR and EULER-AIR-Finishing are shown in Fig. 2. With only shotgun reads, EULER-AIR makes one correction: base 670,009 "G" $\rightarrow$ "A". Base "A" at 670,009 has a high probabilistic coverage 7.973 (summing over probabilistic assignments of all reads that have "A" at that position) out of eight reads. The corresponding positions at the other four copies all have base "G" with high coverage. This is a typical situation when random read assignment can compromise quality of base-assignment in the minority copies. Indeed, we observe a random read assignment at the region around 670,009 in the multiple alignment of reads that was used to construct the finished sequence. Our EM procedure accurately assigns repeat reads, thus reducing the influence of wrongly assigned reads in this difficult base-assignment decision.

TABLE 5
Result of Correcting Base-Assignment Errors in Repeat Region of *Wolbachia* Genome Assembly

Repeat	# repeat reads		# err. bases		# amb. bases		# low cov. bases	
	S	F	S	S+F	S	S+F	S	S+F
1	658	766	1	1	4	8	2360	46
2	243	114	1	0	2	3	164	0
3	194	127	1	1	2	3	0	0
4	40	24	0	0	0	0	356	0
5	24	57	0	0	0	0	0	0
6	26	2	0	0	0	0	0	0
7	17	4	0	0	0	0	2	1
8	21	1	0	0	0	2	0	0
9	11	1	0	0	0	0	0	0
sum	1234	1096	3	2	8	16	2882	47

Both results for EULER-AIR with shotgun reads only $(\mathbf{S})$ *and EULER-AIR-Finishing with shotgun reads and additional finishing reads* $(\mathbf{S}+\mathbf{F})$ *are reported. The definitions of err. bases and amb. bases follow Table 3.* low coverage base *(**low cov. base**) is defined if the coverage of the major base is at most 2. Note that a low coverage base is commonly defined as a base covered by less than two reads, regardless of whether or not the reads have different bases at that position. Our definition is more rigorous and captures more potential problematic bases.*

When finishing reads are added, most repeat regions receive a substantial increase of coverage. EULER-AIR-Finishing confirms the correction at base 670,009, with a higher support (10.977) out of 11 reads. The alignment of reads around base 670,009 is shown in Fig. 3.

3.1.5 Using EULER-AIR for Designing Finishing Experiments

It is hard to estimate the coverage in repeat regions for designing finishing experiments without an accurate repeat read assignment. Indeed, in the set of finishing reads of the *Wolbachia*, we observe both regions with excessively high coverage (wasteful finishing efforts) as well as regions with insufficient coverage of finishing reads (requiring additional finishing efforts).

Copy 4 of repeat family 3 in Fig. 2 contains a region of excessive coverage. This is made by 78 finishing reads from a single clone (DMGSB01), representing an extremely redundant finishing. Since some of these finishing reads have base disagreements with the consensus sequence, they create three ambiguous bases, visible in Fig. 2 as downward spikes in copy 4. On the other hand, the region around bases 548,325-548,475, which is in an instance of a 3-copy repeat, has an average coverage of only five (one from finishing reads) and there are disagreements between these reads. As a result, although EULER-AIR found some potential base-assignment errors, it remains unclear with the current set of reads whether or not and how these errors should be corrected.

Incorporating EULER-AIR in the finishing stage would help the finishers to design a more balanced allocation of resources for producing accurate base-assignments in repeat regions.

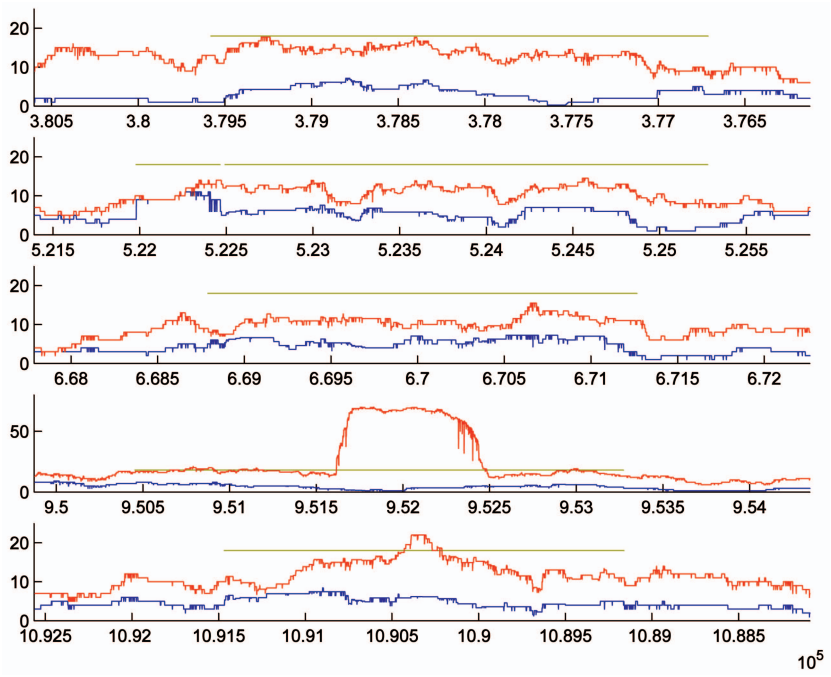

Fig. 2. Coverage plot of repeat family 3 in *Wolbachia* genome. The coverage (support for the majority base at each position) in five genomic windows, each containing one copy of the repeat, is shown. Copy 1 and copy 5 are on the reverse strand (as indicated by decreasing coordinates on the x-axis), while the other copies are on the forward strand. The repeat regions are marked by gray horizontal bars. Varying bar lengths indicate these copies have different repeat boundaries. Blue lines display the coverage by shotgun reads, red lines the coverage by shotgun reads plus finishing reads. The difference between the two lines corresponds to the coverage by finishing reads. Copy 4 (y-axis shown at larger scale) contains a plateau revealing an excessive coverage by finishing reads. Downward spikes on a red line around the plateau correspond to disagreements between the reads at some positions, suggesting possible cloning errors.

```
Genome position: 670,009
                                                @
Old Consensus                ACCAAAAAGA-TCCTGCGCAGACGCTACTGTGGCTTGTCAC
New Consensus                ACCAAAAAGA-TCCTGCGCAAACGCTACTGTGGCTTGTCAC
--------------------------------------------------------------------------------
DMGHR27TF                : : ACCAAAAAGA-TCCTGCGCAAACGCTACTGTGGCTTGTCAC (1.000)
DMGNJ71TF                : : ACCAAAAAGA-TCCTGCGCAAACGCTACTGTGGCTTGTCAC (0.987)
DMGMG19TF                : : ACCAAAAAGA-TCCTGCGCAAACGCTACTGTGGCTTGTCAC (1.000)
DMGNR95TF                : : ACCAAAAAGA-TCCTGCGCAAACGCTACTGTGGCTTGTCAC (1.000)
DMGHT66TF                : : ACCAAAAAGA--CCTGCGCAAACGCTACTGTGGCTTGTCAC (1.000)
DMGJH77TACR9E10776*      : : ACCAAAAAGA-TCCTGCGCAAACGCTACTGTGGCTTGTCAC (1.000)
DMGJH77TBER9F10888*      : : ACCAAAAAGA-TCCTGCGCAAACGCTACTGTGGCTTGTCAC (1.000)
DMGOD93TF                : : ACCAAAAAGA-TCCTGCGCAAACGCTACTGTGGCTTGTCAC (1.000)
DMGJU72TF                : : ACCAAAAAGA-TCCTGCGCAAACGCTA-TGTGGCTTGTCA  (0.990)
DMGJH77TADR9E10776*      : :              CCTGCGCAAACGCTACTGTGGCTTGTCAC (1.000)
DMGJH77TBFR9F10888*      : : ACCAAAAAGA-TCCTGCGCAAACGCTACTGTGGCTTGTCAC (1.000)
```

Fig. 3. EULER-AIR-Finishing's read multiple alignment centered at base 670,009 of the *Wolbachia* genome. A window of 41 columns is shown. Each row corresponds to a mate pair, but is named after one read in the pair. The probability of assigning the mate pair to this location is indicated in parentheses at the right-hand side. The first seven characters in a read name are the clone name. Finishing reads are marked with "*" and are located here by their clone location. Omitted from the alignment are an additional 25 reads, all of which have "G" at position 670,009 and are assigned to this location with essentially vanishing probability.

TABLE 6
EULER-AIR Results on Some WGS Sequencing Projects [9], [19] (See Table 5 for Definitions)

	# bases				
Genome	**genome**	**repeat**	**err.**	**amb.**	**low cov.**
Staphylococcus epidermidis RP62A	2,616,530	121,850	> 10	> 24	> 313
Lactococcus lactis	2,365,589	123,005	> 28	> 17	> 6677

Obtaining the accurate number of errors in these projects demands a careful examination of each individual case. We provide here only lower bound estimates of the number of errors.

TABLE 7
Four Erroneous Bases Found by EULER-AIR in *Lactococcus lactis* [19]

Pooled Alignment		**Separation by EULER-AIR**							
maj	**sec**	**copy 1**				**copy 2**			
		position	**old**	**maj**	**sec**	**position**	**old**	**maj**	**sec**
C(7)	T(5)	1041537	C	T(5)	C(2)	1451719	C	C(5)	none
C(7)	T(5)	1041691	C	T(5)	C(1)	1451565	C	C(6)	none
G(5)	A(3)	1041723	G	A(3)	N(1)	1451533	G	G(5)	none
A(5)	G(4)	1041750	A	G(4)	none	1451506	A	A(5)	none

These bases are located in one copy of a $2 \times 1,200$ *repeat (see Table 3 for definitions).*

3.2 Other Genomes

We also applied EULER-AIR to data from some other genome sequencing projects and found a significant number of base-assignment errors (Table 6). The sequencing data of *Staphylococcus epidermidis RP62A* is also obtained from the TIGR benchmark depository [9] and that of *Lactococcus lactis* [19] is obtained from INRA [20]. The repetitive nature of *Lactococcus* demands an alternative strategy other than WGS [19]. EULER-AIR found a number of erroneous bases, ambiguous bases, and low coverage bases in both genomes. The detailed results are available at our companion Web site.

For example, EULER-AIR found four potential errors even in a relatively short low-copy repeat ($2 \times 1,200$) in *Lactococcus*. As shown in Table 7, in the alignment pooling reads from both copies together, copy 2 (1,451,245-1,452,273) overshadows copy 1 (1,040,984-1,042,012), resulting in the elimination of four distinct bases of copy 1 in the finished sequence. EULER-AIR separates reads and corrects these errors. The four positions in copy 1 are covered by three reads and the corresponding positions in copy 2 are covered by five reads, so we are confident in these corrections.

Admittedly, the *Wolbachia* genome is very accurately finished: EULER-AIR only found a few errors. Most bacterial genomes have significantly fewer finishing reads than *Wolbachia* and are likely to have more base-assignment errors in repeats. The base-assignment in early-sequenced genomes is particularly prone to errors, due to insufficient experimental data (e.g., mate pair information and finishing data). Therefore, the analysis based on the sequences in repeat regions in these genomes should be cautious.

For example, the *Campylobacter jejuni* genome [21] has very few repeats, with only 50,000 bp in repeat regions. Interestingly, a significant portion of its repeat content is concentrated in the three copies of a 6,000 bp long ribosomal RNA operon. In the published sequence, all three copies are perfectly identical. However, EULER-AIR reveals that there are regions of very low coverage around the boundaries of this repeat (Fig. 4). Clearly, accurate base-assignment in this repeat is impossible without additional finishing efforts.

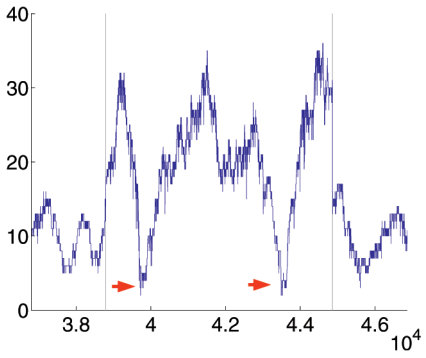

Fig. 4. Coverage plot of a repeat in *Campylobacter jejuni* genome. The genome region contains one copy (between the gray bars) of a $3 \times 6,000$ repeat. Two thousand bp flanking unique region at both sides is also shown. Coverage in the repeat region is calculated by bringing in reads from all three copies. Two regions (arrows) of very low coverage are clearly visible.

4 Discussions

4.1 Application of EULER-AIR in Finishing

The goal of finishing is two-fold: 1) closure of gaps and low-coverage region and 2) repeat resolution. The first goal includes bridging physical gaps and filling the low coverage unique regions. Most finishing-aid software tools are mainly developed for this task (e.g., Autofinish [4]). In practice, repeat resolution often requires more efforts than gap closure, especially when the genomes are very repetitive [18], [22]. However, as of this writing, there are no automatic software tools aimed at facilitating this task.

A traditional finishing model is shown in Fig. 5a. In order to increase the coverage, shotgun reads with high similarity to the finishing reads are also considered for base-assignment. In repeats with copies that differ from one another by more than 2 percent, a strict criterion for screening shotgun reads should suffice to ensure the correct assignment of most of the reads. However, for almost identical repeats (with copies that differ by less than 2 percent where the difference is comparable to the sequencing error rate), this scheme will inevitably erroneously assign many of the reads such that accurate base-assignment is impossible.

We therefore propose the following *repeat-aware* finishing model, i.e., with the emphasis on the task of repeat resolution (Fig. 5b): The input to the finishing process includes shotgun reads and a layout. We classify shotgun reads into unique reads and repeat reads. Accordingly, the layout is a *multilayout* that records multiple mapping locations of repeat reads, a difference from conventional layout where repeat reads are assigned to an arbitrary mapping location.

Using only uniquely mapped reads, we look for areas of insufficient coverage that would suggest additional reads. Once sequenced, these new finishing reads can be pooled together with the existing reads to get a better assembly. At this point, we can invoke EULER-AIR-Finishing initialized with the uniquely mapped reads (most finishing reads and nonrepeat shotgun reads) to obtain an updated version of read assignment. And, this process repeats.

We emphasize that the set of repeat reads and unique reads may change in between finishing runs because EULER-AIR-Finishing may be able to assign reads that are previously classified as repeat reads to unique locations. As a result, base-assignment at some positions in repeats may change during finishing cycles. In the end, all bases of the genome should have decent coverage from finishing reads or unique shotgun reads, thus a high quality finished sequence with strictly controlled base-assignment errors in both unique and repeat regions is obtained.

4.2 Application to Large WGS Projects

Most large eukaryotic genomes are sequenced using the WGS strategy. The finishing of these WGS projects are extremely difficult because of the existence of transposon insertions with high copy number and large-scale segmental duplications. Transposons that are inserted in the euchromatin regions often have their paralogous copies in the heterochromatin regions as well. Since a draft assembly mainly represents the euchromatin sequences, copies of transposons in the heterochromatin regions are usually missing in the draft assembly. As a result, it is impossible to

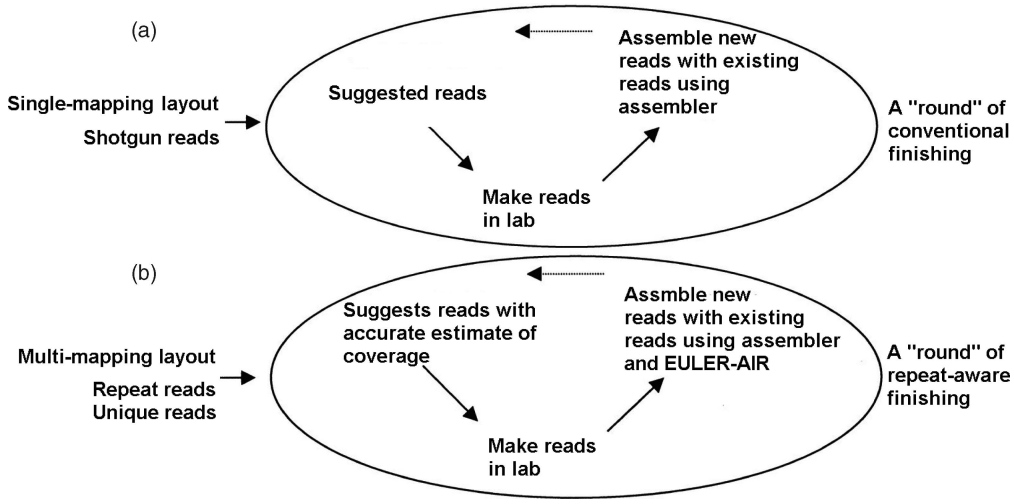

Fig. 5. Conventional finishing model (adopted from Fig. 1 of [4]) and our proposed repeat-aware finishing model.

assign a shotgun read if it is from such a transposon completely (i.e., it has no overlap to any unique regions).

Repeat regions in eukaryotic WGS projects are often finished by additional BAC sequencing. In case of tandem segmental duplications occurring within a single BAC, the BAC can be treated as a bacterial genome and EULER-AIR can be applied directly. If the duplicated segments are dispersed so that they are mapped to distinct BACs, the BAC reads generated in the finishing phase can be uniquely mapped to ease the task of recovering differences between duplicated segments. Therefore, EULER-AIR-Finishing can be initialized by the uniquely mapped BAC reads before shotgun repeat reads are located via EM procedure.

4.3 Future Works

The current method can be improved in several ways. First, we do not use quality values and chromatograms in our experiments, partially because the quality values are not always publicly available. It was not until recently the trace files were made publicly available. And, we plan to incorporate quality values as an additional input to our program in the future.

Second, polymorphism in repeats is not considered in current method. In the presence of single nucleotide polymorphisms (SNPs), some WGS projects may get reads from multiple individuals in the population, thus there may not be a unique consensus sequence to be reconstructed for a genomic region. We are working on an extension of the current EM algorithm that can resolve repeats and SNPs in the same time.

5 Conclusions

Finishing is still a bottleneck for whole genome shotgun sequencing projects, especially in the presence of repeats. In each of the genome sequences we examined, we observed erroneous base assignments in repeat regions. This fact can be partly attributed to the lack of algorithms and software tools for checking base-assignment errors in repeat regions. We developed the EULER-AIR program based on an EM algorithm and showed that it can be used to find base-assignment errors in genome assembly projects. EULER-AIR can also be integrated into the finishing efforts to help the design of efficient finishing experiments that takes into account the ambiguous assignment of repeat reads. Thus, EULER-AIR can effectively reduce the labor of finishing repeat regions and improve the base-assignment quality in repeat regions.

Acknowledgments

The authors thank Mihai Pop for providing additional information and data regarding the TIGR genome assembly benchmark Web site. They thank Jonathan Eisen, Scott O'Neill, and Steven Gill for permission to use sequencing data prior to publication of the genomes. They also thank Julian Parkhill and Alexei Sorokin for sharing sequence data. This work was supported by US National Institutes of Health grant 1 R01 HG02366-01.

References

[1] J.A. Bailey, A.M. Yavor, H.F. Massa, B.J. Trask, and E.E. Eichler, "Segmental Duplications: Organization and Impact within the Current Human Genome Project Assembly," *Genome Research,* vol. 11, no. 6, pp. 1005-1017, 2001.

[2] P.A. Pevzner and H. Tang, "Fragment Assembly with Double-Barreled Data," *Bioinformatics,* vol. 17 (suppl 1 (special ISMB 2001 issue)), pp. 225-233, 2001.

[3] G.A. Churchill and M.S. Waterman, "The Accuracy of DNA Sequences: Estimating Sequence Quality," *Genomics,* vol. 14, no. 1, pp. 89-98, 1992.

[4] D. Gordon, C. Desmarais, and P. Green, "Automated Finishing with Autofinish," *Genome Research,* vol. 11, no. 4, pp. 614-625, 2001.

[5] E. Czabarka, G. Konjevod, M. Marathe, A. Percus, and D. Torney, "Algorithms for Optimizing Production DNA Sequencing," *Proc. 11th Ann. ACM-SIAM Symp. Discrete Algorithms (SODA),* pp. 399-408, 2000.

[6] J. Kececioglu and J. Yu, "Separating Repeats in DNA Sequence Assembly," *Proc. Fifth ACM Conf. Computational Molecular Biology (RECOMB),* pp. 176-183, 2001.

[7] G. Myers, "Optimally Separating Sequences," *Genome Informatics,* vol. 12, pp. 165-174, 2001.

[8] M.T. Tammi, E. Arner, T. Britton, and B. Andersson, "Separation of Nearly Identical Repeats in Shotgun Assemblies Using Defined Nucloetide Positions," *DNPs. Bioinformatics,* vol. 18, no. 3, pp. 379-388, 2002.

[9] *TIGR Benchmark Data for Genome Assembly,* http://www.tigr.org/tdb/benchmark, 1995-2005.

[10] R.V. Samonte and E.E. Eichler, "Segmental Duplications and the Evolution of the Primate Genome," *Nature Rev. Genetics,* vol. 3, no. 1, pp. 65-72, 2002.

[11] M.A. Batzer and P.L. Deininger, "Alu Repeats and Human Genomic Diversity," *Nature Rev. Genetics,* vol. 3, no. 5, pp. 370-379, 2002.

[12] Int'l Human Genome Sequencing Consortium, "Initial Sequencing and Analysis of the Human Genome," *Nature* vol. 409, pp. 860-921, 2001.

[13] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc. B,* vol. 39, no. 1, 1977.

[14] EULER-AIR, Web site http://www-cse.ucsd.edu/groups/bioinformatics/euler_air, 2004.

[15] S. Batzoglou, D.B. Jaffe, K. Stanley, J. Butler, S. Gnerre, E. Mauceli, B. Berger, J.P. Mesirov, and E.S. Lander, "ARACHNE: A Whole-Genome Shotgun Assembler," *Genome Research,* vol. 12, no. 1, pp. 177-189, 2002.

[16] P. Pevzner, H. Tang, and G. Tesler, "De Novo Repeat Classification and Fragment Assembly," *Proc. Eighth ACM Conf. Computational Molecular Biology (RECOMB),* 2004.

[17] M. Pop, D. Kosack, and S. Salzberg, "Hierarchical Scaffolding with Bambus," *Genome Research,* vol. 14, no. 1, pp. 149-159, 2004.

[18] P. Chain et al., "Complete Genome Sequence of the Ammonia-Oxidizing Bacterium and Obligate Chemolithoautotroph Nitrosomonas europaea," *J. Bacteriology,* vol. 185, no. 9, pp. 2759-2773, 2003.

[19] A. Bolotin, P. Wincker, S. Mauger, O. Jaillon, K. Malarme, J. Weissenbach, S.D. Ehrlich, and A. Sorokin, "The Complete Genome Sequence of the Lactic Acid Bacterium Lactococcus lactis ssp. lactis IL1403," *Genome Research,* vol. 11, no. 5, pp. 731-753, 2001.

[20] Nat'l Inst. Agricultural Research, France, http://www.inra.fr, 2001.

[21] J. Parkhill et al., "The Genome Sequence of the Food-Borne Pathogen Campylobacter Jejuni Reveals Hypervariable Sequences," *Nature,* vol. 403, pp. 665-668, 2000.

[22] H. Tettelin et al., "Complete Genome Sequence of Neisseria meningitidis Serogroup B Strain MC58," *Science,* vol. 287, no. 5459, pp. 1809-1815, 2000.

[23] J. Kaminker et al., "The Transposable Elements of the Drosophila Melanogaster Euchromatin: A Genomics Perspective," *Genome Biology,* vol. 3, no. 12, research0084.1-0084.20, 2002.

[24] J.W. Kent, "BLAT—The BLAST-Like Alignment Tool," *Genome Research,* vol. 12, no. 4, pp. 656-664, 2002.

[25] A.L. Delcher, A. Phillippy, J. Carlton, and S. L. Salzberg, "Fast Algorithms for Large-Scale Genome Alignment and Comparison," *Nucleic Acids Research,* vol. 30, no. 11, pp. 2478-2483, 2002.

[26] S.F. Altschul, T.L. Madden, A.A. Schffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman, "Gapped BLAST and PSI-BLAST: A New Generation of Protein Database Search Programs," *Nuclear Acid Research,* vol. 25, no. 17, pp. 3389-3402, 1997.
[27] E.L. Anson and E.W. Myers, "ReAligner: A Program for Refining DNA Sequence Multi-Alignments," *Proc. First ACM Conf. Computational Molecular Biology (RECOMB),* pp. 9-16, 1997.

Degui Zhi received the BS degree in computer science from Beijing University and the PhD degree in bioinformatics from the University of California, San Diego in 2006. His current research interests include repeat analysis and protein structure analysis.

Uri Keich received the PhD degree from the Courant Institute of Mathematical Sciences, New York University. He is an assistant professor in the Computer Science Department at Cornell University, Ithaca, New York. His current research interests include computational statistics and algorithmic problems in biological sequence analysis.

Pavel Pevzner received the PhD degree in 1988 from the Moscow Institute of Physics and Technology. He holds the Ronald R. Taylor Chair in Computer Science at the University of California, San Diego. His current research interests include genome rearrangements, repeat analysis, and computational proteomics.

Steffen Heber received the PhD degree in mathematics from the University of Heidelberg, Heidelberg, Germany, in 2001 and did postdoctoral work at the University of California, San Diego. He is currently an assistant professor in computer science at North Carolina State University, Raleigh, with a joint appointment in the Department of Computer Science and the Bioinformatics Research Center. His research interests include bioinformatics and computational biology.

Haixu Tang received the PhD degree in molecular biology from the Shanghai Institute of Biochemistry in 1998. He worked as an assistant project scientist in the Department of Computer Science and Engineering at the University of California, San Diego from 2001 to 2004. Between 1999 and 2001, he was a postdoctoral researcher in the Department of Mathematics at the University of Southern California. He is currently an assistant professor in the School of Informatics, also affiliated with the Center for Genomics and Bioinformatics, at Indiana University, Bloomington. His research interests include computational proteomics and evolutionary genomics.