Detecting and correcting mis-assembled reads in contigs

Hicham Benzekri[1], Darío Guerrero-Fernández[1], Rocío Bautista[1], and M. Gonzalo Claros[1,2,*]

[1] Plataforma Andaluza de Bioinformática-SCBI, University of Málaga,
C/ Severo Ochoa 34, 29590 Málaga, Spain
{bhicham,dariogf,rociobm,claros}@uma.es
http://www.scbi.uma.es

[2] Molecular Biology and Biochemistry Department, University of Málaga,
Campus de Teatinos s/n, 29071 Málaga, Spain
http://www.bmbq.uma.es/fmp

Abstract. *De novo* assemblies do not have the possibility of quality control with an external sequence. In fact, accuracy and reliability of these assemblies is highly affected by sequencing errors and mis-assemblies. Here, a frequency-based algorithm is developed in Ruby and intended to discern assembly errors from polymorphisms/read errors and then edit or remove the misassembled read(s) to provide more but highly reliable contigs. The software reads and writes the ACE assembly format. Transcriptome and genome assemblies were tested.

Keywords: contigs, OLC, de novo, assembly.

1 Introduction

Sequence assembly errors exist in any *de novo* assembly. Identification of mis-assemblies is a difficult issue due to the high amount of data and its error-prone quality because of biochemical and mechanical complications in sequencers. This usually requires additional efforts for manual validation of the most accurate reconstruction of the analyzed genome or transcriptome [1]. Too often, assembly quality is judged only by contig size or N50, with larger contigs being preferred [2], even though large contigs can be chimeric as a result of mis-assembling.

A widely-used contig-testing tool is *Hawkeye* [3]. It can be used with assemblies of all sizes to facilitate the visual inspection of large-scale assembly data while minimizing the time needed to detect mis-assemblies and make accurate judgments for assembly quality. In fact, it guides users to the most likely areas of mis-assembly, allowing its manual edition and correction. In contrast to other contig editors such as GAP5 [4], *Hawkeye* combines computational predictors with interactive visualizations to decrease verification costs. However, visual inspection and manual edition are cumbersome tasks, particularly for assemblies from next-generation sequencing (NGS) data. This is the reason why *amosvalidate* [2], an automated

* Corresponding author

validation pipeline for contigs based on several independent criteria, was developed. But this tool only tagged regions that appear mis-assembled, and the correction requires visualization again and manual edition with *Hawkeye*.

The aim of our work is to develop a fully automated algorithm called CoMiner with the aim of editing and correcting prominent mismatches in contigs, reducing the manual intervention dedicated to increase the quality of assemblies, based only on the contig assembly *per se*.

2 Implementation

CoMiner was programmed in Ruby and tested in a dual core iMac at 3.06 GHz with 4 GB of RAM. Contig data can be read and written in ACE format [5], which is generated by various assembly programs, such as Phrap, CAP3, GAP4-5, Newbler, Arachne, Minimus and TIGR Assembler, all of them based on overlay-layout-consensus algorithms (CoMiner is not ready for analyzing De Bruijn contigs, but could be adapted for mapping alignments in a near future).

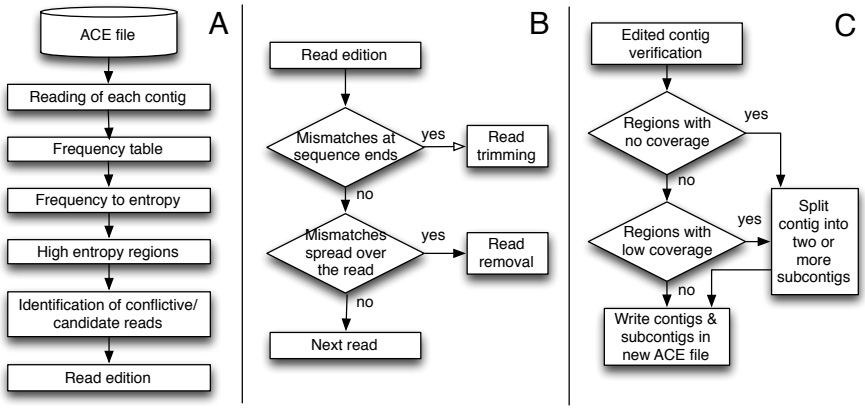

Fig. 1. Flowgram of CoMiner algorithm. A: Discovery of high entropy regions and identification of conflictive reads. B: Conflictive-read edition/removal within a contig depending on mismatch distribution. C: Once all conflictive reads in a contig have been edited, the contig coverage is verified, split in two or more contigs if necessary, and then saved into a new ACE file.

Since the final goal of CoMiner is to increase the assembly accuracy without human intervention, the algorithm (Fig. 1) can be divided in the following main steps: (i) discovery of high-entropy regions (HERs); (ii) identification of conflictive read(s); (iii) read edition (trimming or removal); (iv) contig verification and saving in a new ACE file.

(i) *HER discovery*: The aim of this step is to retrieve assembly fragments where reads do not align perfectly. We have elected the entropy of consensus nucleotide at position i [$-H(i)$] as a measure of the alignment goodness as described in [6]. Therefore, the frequency of each of the four nucleotides at each position of the

Fig. 2. Different instances of mismatch distribution in a conflictive read. A: All mismatches are located at one end; therefore, nucleotides within the distance *d* were trimmed from the read, provided that $d < 40\%$ of the read length, and the remaining read is longer than 40 nt. B: There are mismatches at both ends; again, nucleotides within distances *d1* and *d2* are trimmed provided that $d1 + d2 < 40\%$ of the read length, and the remaining read is longer than 40 nt. C: Mismatches are spread over the whole read; when the number of mismatches is over 2% of the read length, the complete read is removed.

consensus sequence is assessed and then used to calculate entropy at each consensus position. SNPs and point sequence are considered equivalent events in this rationale, and do not significantly affect assemblies unless they are closely located. This is the reason why entropy data were sieved by a fast Fourier transform as described [6] for converting contiguous sharp peaks into high entropy regions. Sequence ranges whose Fourier-transformed entropy is over a cutoff value that corresponds to the median entropy of the contig will be considered HERs and will focus subsequent analyses.

(ii) *Identification of conflictive-read(s)*: Several calculations are performed to determine whether a HER was caused by one or more mis-assembled reads or whether it was caused by mismatches scattered over all involved reads. These possibilities are discerned calculating the mismatch frequency of one read *r* [$F_{read}(r)$] as follows: for a contig containing *m* number of reads for which a *k* number of HERs have been defined, *n(j)* being the length of one HER, $F_{read}(r)$ is calculated dividing the total number of mismatches of the read *r* within all HERs against the consensus by the total number of nucleotides involved in all HERs:

$$F_{read}(r) = \sum_{j=1}^{k}\sum_{i=1}^{n} err(i,j) \Big/ \sum_{j=1}^{k} n(j)$$

The total mismatch frequency of the contig (F_{contig}) is calculated dividing the total number of mismatches of every contig read within all HERs by the total length of all HER regions in each read, as follows:

$$F_{contig} = \sum_{r=1}^{m}\sum_{j=1}^{k}\sum_{i=1}^{n} err(i,j,r) \Big/ \sum_{r=1}^{m}\sum_{j=1}^{k} n(j,r)$$

Both $F_{read}(r)$ and F_{contig} will define a robust F_{cutoff} value as:

$$F_{cutoff} = F_{contig} + K \times \left(\sum_{r=1}^{m} \left| F_{read}(r) - F_{contig} \right| \Big/ m \right)$$

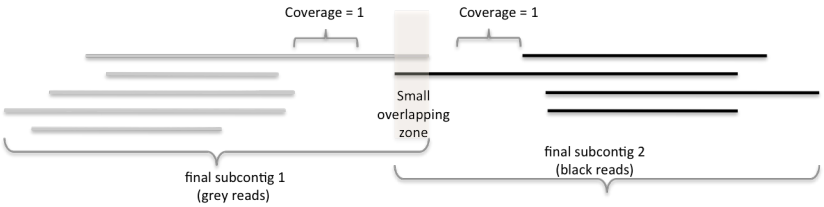

Fig. 3. Example of contig after read edition, where only two reads slightly connect two putative subcontigs. CoMiner will divide it in two new subcontigs.

where K = 1.4826 to consider outliers only those values beyond the third quartile. Therefore, reads with $F_{read}(r) > F_{cutoff}$ are considered conflictive and candidate for edition.

(iii) *Edition of candidate-read(s)*: The distribution of mismatches in candidate reads is analyzed as detailed in Fig. 2, driving to the trimming or removal of conflictive read(s) depending on mismatch distribution.

(iv) *Contig verification and saving*: Read edition may modify the contig coverage and some region(s) may be now devoid of any read. The algorithm looks for this type of situations and splits the contig in two new, independent subcontigs, each one with a new, independent consensus sequence. There is a special case where a contig can contain an overlapping region of two reads flanked by a coverage of only one read (Fig. 3). This contig will be split into two independent subcontigs when the overlapping fragment is below 40 nt or the identity is below 90%. Unedited contigs, edited contigs and new subcontigs are then written into a new ACE file.

Table 1: Results of two assemblies before (–) and after (+) CoMiner treatment

	Transcriptome		**Genome**	
CoMiner	–	+	–	+
Contig #	76 824	76 894	35 777	35 823
Mean contig size (nt)	429	428	471	470
N50 (nt)	484	482	506	505
N90 (nt)	251	252	307	307
Edited contigs		21 002		1465
Split contigs		146		74
Mapped contigs			35 412	35 458
Mapped nt			3 362 691	3 362 923

3 Results and Discussion

CoMiner performance was tested for transcriptome and genome assemblies (Table 1). A total of 1 110 923 454/FLX reads from *Solea senegalensis* transcriptome were trimmed using SeqTrimNext (http://www.scbi.uma.es/seqtrimnext) [7] and then

assembled using MIRA3 (http://www.scbi.uma.es/mira) with the standard parameters for 454/FLX data. In the resulting assembly (Table 1, Transcriptome columns), CoMiner detected HERs in 33 912 contigs (44.1%), but only edited 21 002 (27.3%), corresponding to contigs where at least one HER was caused by mismatches concentrated in at least one read. An example of this is shown in Fig. 4A, in which the algorithm identified a read containing several mismatches that were responsible for the wide, central HER. The right (3') end of this read contained all mismatches and was consequently trimmed. A new entropy analysis after CoMiner automatic edition showed that the HER had disappeared (Fig. 4B), suggesting that the edited contig was more reliable than the initial one.

Integrity of most contigs was unaffected by CoMiner edition, but 146 (0.7%) were split into two subcontigs and other 2 contigs were split into 3 different subcontigs each. It should be noted that when a subcontig consisted of only one read, it is not considered a contig and the read is removed from the final contig count. An example of contig splitting is shown in Fig. 5, where a 1570 bp contig was divided into two smaller subcontigs. Another example of this situation can be the chimeric contig group1_solea_c8241 of 1500 nt, since it was divided into a 5' subcontig of 887 nt

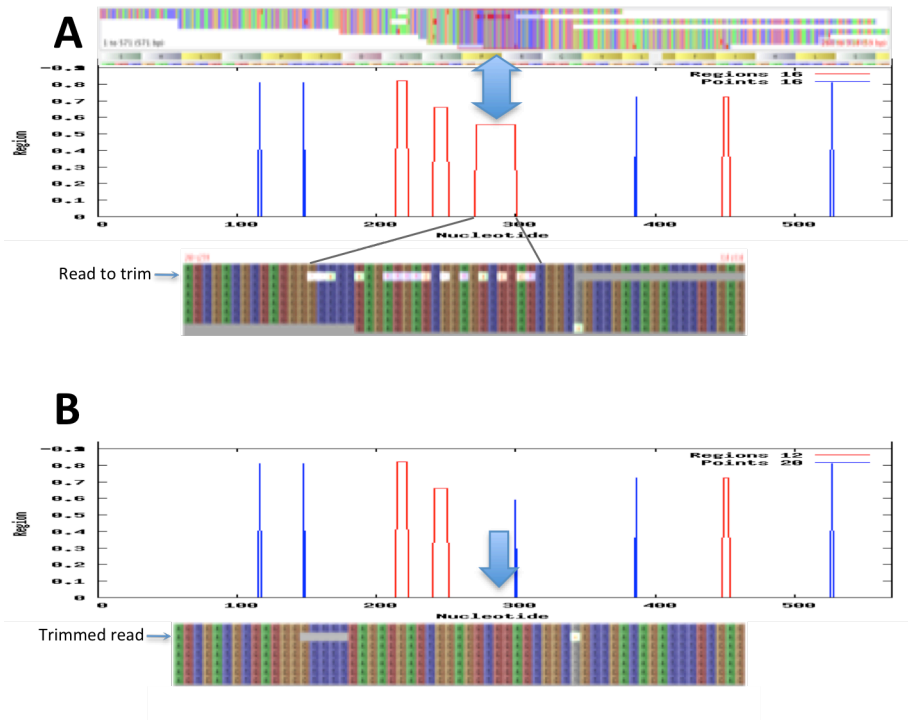

Fig. 4. Example of a 571 bp contig with several HERs before (A) and after (B) self-edition using CoMiner. After sieved entropy analysis, HERs spanning only one nucleotide are considered point errors or SNPs (in blue), while true HERs (in red) span two or more nucleotides. The wider HER (nt 273-296) is marked by a double arrow and magnified below, showing that all mismatches come from one single read. After edition (B), this wide HER disappeared. The other three smaller HERs were still present, suggesting that their mismatches were not concentrated in a single read and, therefore, will not be edited.

with similarity to an ORM1 like protein (B0V340; $E = 10^{-107}$) of 153 amino acids, and a 3' subcontig of 601 nt without similarity in databases. As a result, automatic edition of 27.3% of contigs did not significantly increase the number of contigs and did not significantly change the general parameters of the assembly (Table 1, Transcriptome columns), while contig reliability was presumably improved.

Genome DNA assembly was tested using 317 692 genomic reads of *Arabitopsis thaliana* (SRX105465). They were pre-processed with SeqTrimNext and assembled with CAP3 (http://www.scbi.uma.es/cap3) to obtain 35 777 contigs (Table 1, Genome columns). CoMiner detected 3259 contigs (9.1%) with one or more HERs, but only edited 1465 (4,1%), 74 of them being split into two or more contigs. Contigs before and after CoMiner treatment were mapped to *A. thaliana* genome using an in-house algorithm (H. Benzekri, unpublished results) to test the putative increase of contig reliability. A total of 35 412 (98.97%) and 35 458 (98.98%) contigs were mapped, respectively, providing a total of 3 362 691 and 3 362 923 mapped nucleotides, respectively (Table 1, Genome columns). When mapping was performed with a more restrictive mapper, such as Bowtie2 [8], 8453 original contigs (23.62%) and 8494 CoMiner-edited contigs (23.71%) were mapped. Edition slightly increased (1.001-1.004 fold) the amount of mapped contigs and nucleotides in any case. Unfortunately, this increase is in the same range as the total contig number, indicating that more analyses are required to provide statistical significance for this weak increase.

Even though CoMiner is currently only able to manage mismatches in overlay-layout-consensus assemblies, it seems to be a promising tool for automatic editing of mis-assembled reads. CoMiner performance was tested with transcriptome and genome data, and quality of edited contigs presumably seems improved. However, more real-world assemblies should be performed to give statistical significance to the qualitative results presented here. Finally, CoMiner edited contigs can always be

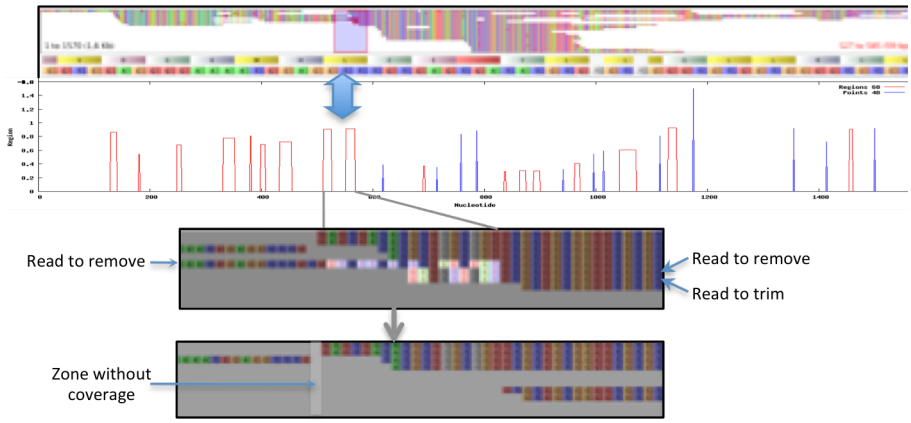

Fig. 5. Example of a 1570 bp contig with several real HERs in red. The arrow is signaling a couple of HERs that was resolved by left-trimming one read and removing two other reads. Edition caused a gap within the contig that CoMiner resolved splitting it into two subcontigs, the left subcontig of 551 nt, and the right subcontig of 1018 nt.

analyzed and visualized to search for more, different tentative errors by means of *amosvalidate* and *Hawkeye*, with the aim of spending less manual edition efforts.

Acknowledgement

The authors gratefully acknowledge Josep Planas (University of Barcelona, Spain) for kindly providing 454/FLX data from the AQUAGENET project (funded by Interreg-Sudoe). We would also like to acknowledge the computer resources and technical support provided by the Plataforma Andaluza de Bioinformática of the University of Málaga, Spain. This study was supported by grants from the Spanish MICINN (BIO2009-07490) and Junta de Andalucía (P10-CVI-6075), as well as institutional funding to the research group BIO-114 and an agreement with AQUAGENET project.

References

1. Istrail, S.; Sutton, G.G.; Florea, L.; Halpern, A.L.; Mobarry, C.M.; Lippert, R.; Walenz, B.; Shatkay, H.; Dew, I.; Miller, J.R.; Flanigan, M.J.; Edwards, N.J.; Bolanos, R.; Fasulo, D.; Halldorsson, B.V.; Hannenhalli, S.; Turner, R.; Yooseph, S.; Lu, F.; Nusskern, D.R.; Shue, B.C.; Zheng, X.H.; Zhong, F.; Delcher, A.L.; Huson, D.H.; Kravitz, S.A.; Mouchard, L.; Reinert, K.; Remington, K.A.; Clark, A.G.; Waterman, M.S.; Eichler, E.E.; Adams, M.D.; Hunkapiller, M.W.; Myers, E.W.; Venter, J.C. Whole-genome shotgun assembly and comparison of human genome assemblies. *Proc Natl Acad Sci U S A* **2004**, *101*, 1916-1921.
2. Phillippy, A.M.; Schatz, M.C.; Pop, M. Genome assembly forensics: finding the elusive mis-assembly. *Genome Biol* **2008**, *9*, R55.
3. Schatz, M.C.; Phillippy, A.M.; Shneiderman, B.; Salzberg, S.L. Hawkeye: an interactive visual analytics tool for genome assemblies. *Genome Biol* **2007**, *8*, R34.
4. Bonfield, J.K.; Whitwham, A. Gap5--editing the billion fragment sequence assembly. *Bioinformatics* **2010**, *26*, 1699-1703.
5. Gordon, D.; Abajian, C.; Green, P. Consed: a graphical tool for sequence finishing. *Genome Res* **1998**, *8*, 195-202.
6. Guerrero, D.; Bautista, R.; Villalobos, D.P.; Canton, F.R.; Claros, M.G. AlignMiner: a Web-based tool for detection of divergent regions in multiple sequence alignments of conserved sequences. *Algorithms Mol Biol* **2010**, *5*, 24.
7. Falgueras, J.; Lara, A.J.; Fernandez-Pozo, N.; Canton, F.R.; Perez-Trabado, G.; Claros, M.G. SeqTrim: a high-throughput pipeline for pre-processing any type of sequence read. *BMC Bioinformatics* **2010**, *11*, 38.
8. Langmead, B.; Salzberg, S.L. Fast gapped-read alignment with Bowtie 2. *Nat Methods* **2012**, *9*, 357-359.