Genomics Insights

Libertas Academica
FREEDOM TO RESEARCH

ORIGINAL RESEARCH

Single Nucleotide Polymorphisms Caused by Assembly Errors

Jürgen Kleffe[1], Robert Weißmann[2] and Florian F. Schmitzberger[3]

[1]Institute of Molecular Biology and Bioinformatics, Charité Berlin, Arnimallee 22, Berlin, Germany. [2]Department of Mathematics and Computer Science, Freie Universität Berlin, Takustr. 9, Berlin, Germany. [3]Stanford University Biomedical Informatics, Stanford, California, USA. Email: juergen.kleffe@charite.de

Abstract: We compare the results of three different assembler programs, Celera, Phrap and Mira2, for the same set of about a hundred thousand Sanger reads derived from an unknown bacterial genome. In difference to previous assembly comparisons we do not focus on speed of computation and numbers of assembled contigs but on how the different sequence assemblies agree by content. Threefold consistently assembled genome regions are identified in order to estimate a lower bound of erroneously identified single nucleotide polymorphisms (SNP) caused by nothing but the process of mathematical sequence assembly. We identified 509 sequence triplets common to all three de-novo assemblies spanning only 34% (3.3 Mb) of the bacterial genome with 175 of these regions (~1.5 Mb) including erroneous SNPs and insertion/deletions. Within these triplets this on average leads to one error per 7,155 base pairs. Replacing the assembler Mira2 by the most recent version Mira3, the letter number even drops to 5,923. Our results therefore suggest that a considerably high number of erroneous SNPs may be present in current sequence data and mathematicians should urgently take up research on numerical stability of sequence assembly algorithms. Furthermore, even the latest versions of currently used assemblers produce erroneous SNPs that depend on the order reads are used as input. Such errors will severely hamper molecular diagnostics as well as relating genome variation and disease. This issue needs to be addressed urgently as the field is moving fast into clinical applications.

Keywords: single nucleotide polymorphism, SNP, genome variation, genome comparison, sequence assembly

Genomics Insights 2010:3 1–8

This article is available from http://www.la-press.com.

1. Introduction

Different lines of next generation sequencers are going to make research institutions, hospitals and service companies independent of specialized genome centres for obtaining all the genetic information they want. Corresponding markets are boosting up and cause growing social interest in safe genetic diagnosing. Hence consumer protection will soon play an important role in medicine but also agriculture, fishery, forestry and environmental research. It will soon require strict measures of quality control, standardization and certification.

Compared with about 3 billion dollars spent for the human reference sequence, Applied Biosystems Sequencers produced Craig Venter's personal genome for only 10 million dollars.[1] In 2008, the 454 Roche Genome Sequencers of the Baylor Human Genome Sequencing Centre derived James Watson's genome for just 1 million dollars.[2] The 1000 genomes project is on its way and scientists expect from it the cost per genome to drop to a few thousand dollars. Last but not least, the X-Prize Foundation offers a 10 million dollars award for the first private group that can sequence 100 human genomes in 10 days. Genome sequencing appears to become a routine affordable task in near future.

However, the first shotgun assembly of the human genome carried out by Celera Genomics took 10 days of computation alone using 10–20 computers with 4 processors each.[3] Therefore, software developers entered a worldwide race of inventing improved sequence assembly programs that keep up with the current speed of data generation. The new tools are typically compared with the older products by counting the numbers of contigs, which should be as small as possible, the total size of all contigs, which should be as close as possible to genome size, time of computation and numbers of mis-assemblies. These are mainly compression and expansion errors due to improper handling of repeats which, together with expensive gap closing, are left to fix in the finishing phase.[4–6] Currently and in the light of high speed data generation, most exciting is the size of data a modern assembler can handle. But what about the qualities of assembled genome contigs which have passed the usual first tests of validation? These are usually not diagnosed for refinement by the finishing process, but still may include errors?

Surprisingly, we have not yet seen a single exhaustive comparison of seemingly correctly assembled genome regions derived by different assemblers and from exactly the same data. The errors we find there are merely due to mathematics and hence should be avoided. This paper presents the first results of moving towards this direction.

2. Consistently Assembled Genome Sections

Assume three different assemblers, applied to the same trace data, have produced three different but correct sets of contigs called A, B, and C, presented by the dark horizontal arrows in Figure 1. The red line represents the unknown genome sequence which all correctly assembled contigs have to match somewhere in some direction. Contig assembly alone cannot assign sequence strand. For this reason the compared sets of contigs are extended to also include all complementary sequences.

A contig is called to match if either it or its complement matches the forward strand of the unknown genome sequence. If the contig is correctly assembled it must also match completely, i.e. from start to end. Partial matches would indicate mis-assembly. It is the assembler's job to break contigs at unreliably assembled places and also to clip off such contig ends. But how to select for such contigs given the genome sequence is not known?

2.1. Pairwise complete matches

If all derived contigs would match the unknown genome sequence completely, pairs of contigs originating from different assemblies and matching overlapping genome sections also have to match completely as shown by blue boxes in Figure 1. Such matches

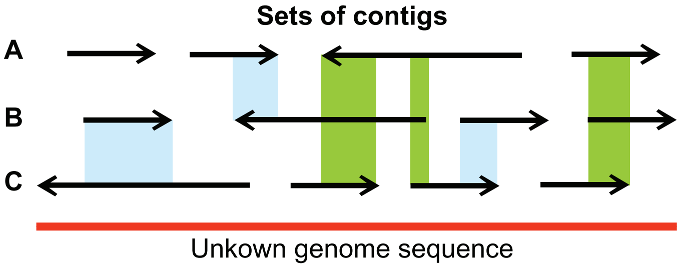


Figure 1. Three pairwise complete matches shown by blue boxes and three 3-fold complete matches shown by green boxes. The arrows indicate contigs of the three assemblies A to C. The boxes show regions of near-perfect alignment between assemblies.

extend to both sides until the end of one of both contigs. A completely matching pair of contigs is called consistently assembled even though the overlap may be small. To be realistic a small number of mismatches and gaps must be tolerated as part of the overlap alignment. Figure 1 shows only some complete matches between contigs of sets A and B or B and C, respectively. Matches between the sets A and C are not shown for clarity.

Our software ClustDB[7] quickly derives from the sets A, B and C all pairs of completely matching contigs where match quality satisfies given criteria. Subsequently a clustering algorithm derives disjoint subsets of unrelated contigs which can be processed independently. By default all overlap alignments have to include an exact match of at least 50 nucleotides implying that the overlap has to be at least that long and not more than 5 errors are allowed within each alignment subsection of length 50 along the entire match. Hence, difference blocks with relatively large numbers of local errors are not allowed even though they would not seriously increase relative error of long overlap alignments. Such events are considered inconsistent and unreliable work of both assemblers the two contigs come from. Possibly at least one assemblers failed to clip off low quality ends of contigs. Or we could have wrongly paired contigs originating from different assemblies caused by spurious matches, a risk also controlled by requiring a minimal length of the match. By these measures we want to select those parts of the contigs which are very unlikely changed in the finishing phase of genome sequencing.

However, the maximum number of errors tolerated in each window should not be too small and the length of the exact match should not be too large. Edit distance is not transitive and two considered contigs could match the true genome sequence at the same place with k errors each while both contigs match each other with 2k errors.

2.2. 3-fold complete matches

Going one step further, we derive from all pairs of completely matching contigs all triplets of contigs which share a common subsequence found by all three assemblers. These genome sections are shown as green boxes in Figure 1, called 3-fold complete matches, and are not very easy to derive. Note that a single contig can be involved in more than one 3-fold complete match.

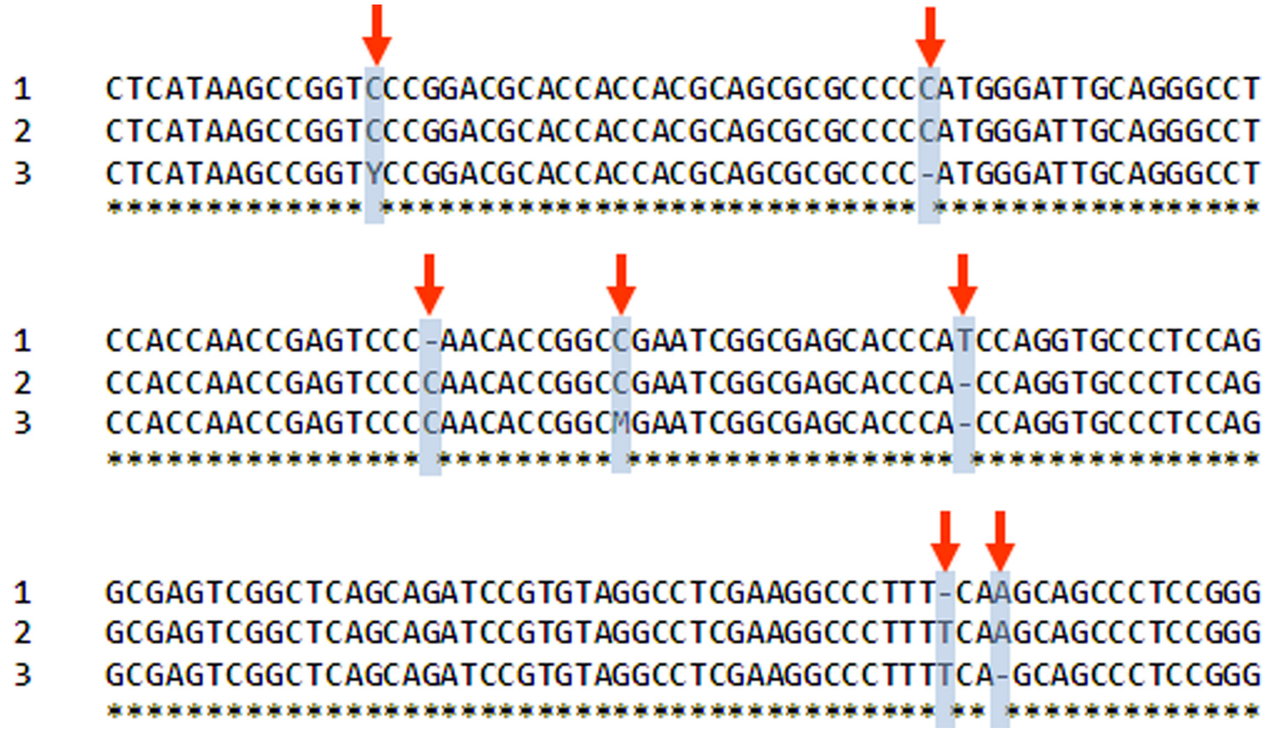
The 3-fold completeness property requires the match to extend to both sides until the end of one of the contributing contigs. All pairwise matches of these contigs have to be complete and all these matches must define a unique layout. This is quite a number of requirements. The resulting genome sections are considered most trustfully assembled and should include the smallest possible numbers of errors. Ideally they should be error free. But as we will report in the next section, they are not, and allow us to observe erroneous genome variation since all three assemblers use the same trace data.

Figure 2 shows parts of a multiple alignment of a 3-fold completely matching genome section. It contains mismatches and gaps similar to biological single nucleotide polymorphisms which are used to describe genetic variation. Two indels also appear like typical errors originating from 454-sequencing miscounting the number of cytosine. But in fact all these errors are only caused by ambiguous steps taken by different algorithms for sequence alignment and assembly.

Interestingly, when comparing the personal genomes of Venter and Watson[1] with the human reference sequence, the authors were surprised to find unexpectedly high numbers of single nucleotide polymorphisms. Experimental verification of 26 selected mutations was reported to have failed in 6 cases.[8] Also a recent comparison of eight human genomes[9] suggests that there is reason to believe that a considerable proportion of reported mutations could be due to a particular sequence assembly process but nothing else.

3. Results

The problem we discuss is not restricted to new generation short read assembly but also shows up for traditional Sanger sequencing often used in de-novo high quality sequencing projects. We assembled 109,287 Sanger reads of a newly sequenced bacterial genome with coverage of roughly 6.8 and utilized the latest versions of the Celera assembler (http://wgs-assembler.sourceforge.net/), the assembler Phrap (http://www.phrap.org/) and also Mira2 (http://www.chevreux.org) as well as its more recent version Mira3. All necessary assembly calculations were performed by Sven Klages, an experienced computer scientist who is in charge of genome sequencing for a number of years. Therefore misapplication of assembly software can be ruled out. Unfortunately, and since the finished



```
1     CTCATAAGCCGGTCCCGGACGCACCACCACGCAGCGCGCCCCCATGGGATTGCAGGGCCT
2     CTCATAAGCCGGTCCCGGACGCACCACCACGCAGCGCGCCCCCATGGGATTGCAGGGCCT
3     CTCATAAGCCGGTYCCGGACGCACCACCACGCAGCGCGCCCC-ATGGGATTGCAGGGCCT
      ************* **************************** *****************

1     CCACCAACCGAGTCCC-AACACCGGCCGAATCGGCGAGCACCCATCCAGGTGCCCTCCAG
2     CCACCAACCGAGTCCCCAACACCGGCCGAATCGGCGAGCACCCA-CCAGGTGCCCTCCAG
3     CCACCAACCGAGTCCCCAACACCGGCMGAATCGGCGAGCACCCA-CCAGGTGCCCTCCAG
      **************** ********* ***************** ***************

1     GCGAGTCGGCTCAGCAGATCCGTGTAGGCCTCGAAGGCCCTTT-CAAGCAGCCCTCCGGG
2     GCGAGTCGGCTCAGCAGATCCGTGTAGGCCTCGAAGGCCCTTTTCAAGCAGCCCTCCGGG
3     GCGAGTCGGCTCAGCAGATCCGTGTAGGCCTCGAAGGCCCTTTTCA-GCAGCCCTCCGGG
      ******************************************* ** *************
```

Figure 2. Some parts of multiple alignments of 3-fold complete matches.

genome sequence has not been published yet, we must regret that we cannot providing more detail on these data. Here we report about just two of a number of mathematical experiments carried out with these data. Our software developed for these studies is freely available for application to other sets of assemblies.

3.1. Different assemblers

Table 1 provides summary data describing the results obtained by the assemblers Celera, Phrap, Mira2 and Mira3. All assemblers produce different numbers of contigs but their total sizes, respectively, well approximate expected genome size. Mira3 predicts the largest genome size but the correct answer is not yet known. The average contig length ranges from 63 KB for Celera down to only 9 KB for Mira2 and 7 KB for Mira3. However all three assemblers also produced a number of interestingly long contigs. The column called W50 provides the smallest number of contigs that includes at least 50% of all contig sequence. For the Mira2 assembly just 6% of the contigs include half of the genome assembly. Hence also Mira2 generates sufficiently long contigs, but in addition it reports large numbers of short fragments. However, all these numbers do not tell us how well the three assemblies

Table 1. Comparing the results of the assemblers Celera, Phrap, Mira2 and Mira3 obtained for 109,289 Sanger reads.

	Contigs	Total size	Av. length	W50
Celera	162	10,232,260	63,162	22
Phrap	390	10,383,774	26,625	42
Mira2	1,289	11,583,563	8,986	81
Mira3	1,478	10,691,233	7,233	226
Mira2		**All sections**	**With errors**	
3-fold matches		509	175	31%
Total size		3,341,792	1,501,621	45%
Alignment errors		467	467	
Bases per error		7,155	3,940	
Mira3				
3-fold matches		514	216	42%
Total size		3,115,451	1,400,282	45%
Alignment errors		526	526	
Bases per error		5,923	2,662	

The upper block provides summary data for all considered assemblies; the middle block describes 3-fold complete matches of Celera, Phrap and Mira2, while the last block gives the same data for replacing Mira2 by Mira3.

really agree. Using ClustDB with default parameters described in Section 2.1 and comparing the contigs generated by the Celera, Phrap and Mira2 assemblers we identified 509 3-fold completely matching genome sections. As expected most of these sequence triplets are formed by identical sequence sections and may be considered the most reliably assembled part of the genome.

However 175 sequence triplets include errors as seen in Figure 2. With a total of 467 error columns found in their multiple alignments and considering their total size we may well expect on average at least one error for each 7 KB of all considered genome sections. These numbers are more likely lower than upper error bounds for the entire genome assemblies since we only studied the most consistently assembled sections of all assemblies. Extrapolating from our estimate we would expect up to 1500 programmatically introduced single-base errors in the full assembly of this genome. The data also shows that errors are not uniformly distributed over all contigs. Relating the total number of errors to the total size of the 175 multiple alignments with errors yields one error in 4 KB of sequence. So we must carefully distinguish error prone from less error prone genome sections. The third part of Table 1 provides very similar results obtained by replacing the Mira2 contigs by those generated by Mira3. Here we end with one error column for every 2 KB of the three-fold consistently assembled genome sections.

Table 2 shows some letter replacement statistics derived from the observed error columns for both experiments, including Mira2 contigs (upper triangle) and Mira3 contigs (lower triangle), respectively. Only for including Mira2 just four alignment mismatch columns contained three different letters. In all other cases two assemblers agreed as seen in Figure 2 and suggested deciding on majority reasons. The Phrap assembler, track 2 in Figure 2, most frequently matched the majority base, in 434 and 335 cases, respectively, closely followed by the Celera assembler in 397 and 339 cases. Mira2 matched the majority base 114 times while Mira3 did so 97 times. One could conclude that the assemblies by Celera and Phrap are more close to each other than to Mira2 and Mira3. The relative frequencies of letter pairs we observed in all error columns show an under representation of AT for both experiments. However, we should be careful in drawing conclusions since both experiments are highly correlated.

Table 2. Relative frequencies of letter replacements.

	A	C	G	T	gap
A		0.139	0.142	0.061	0.076
C	0.125		0.144	0.144	0.094
G	0.131	0.188		0.104	0.050
T	0.047	0.153	0.106		0.046
gap	0.108	0.070	0.057	0.015	

All 3-fold matching contig sections were aligned using ClustalW and pairs of letters reported that appeared in error columns as seen in Figure 2. The upper part shows relative frequencies observed for Celera, Phrap and Mira2 contigs, while the lower part results from replacing Mira2 by Mira3.

Also worrying is the observation that all 3-fold completely matching genome sections account for no more than 34% of the whole genome. But our contigs have not passed any process of finishing. There is also danger that some of our derived 3-fold completely matching genome sections represent overlapping parts of the true genome sequence and cause overestimation of its proportion. Hence the three alternative assemblies may seriously differ on more than 66% of the genome. There may be cases of relocations, rearrangements and copy number variations all not caused by nature but ambiguities of mathematics, only. Surprisingly or not, just one assembler is already enough to generate different assemblies as we can show in the next sections.

3.2. Shuffled data

In Table 3 we compare three runs of the Celera assembler for alternative data obtained by reversing the order of our reads in the input file in one case, and randomly shuffling of reads in the other.

When shuffling reads the quality values are moved along with these so that we would expect the same results from the three data sets called original, reversed and shuffled, respectively.

The results are not that different as seen in Table 1, but unequal numbers of contigs are generated, too. There are fewer but much longer 3-fold complete matches. The three sets of contigs include 72 exactly matching triplets formed by one contig of each set compared to none for the experiment described by Table 1. That means a large proportion of the genome is assembled independent of the order reads are given to the assembler. In another 69 cases the assemblies of the reversed or shuffled sets of reads include contigs

Table 3. Results of the Celera assembler for 109,289 Sanger reads in original, reversed and shuffled order.

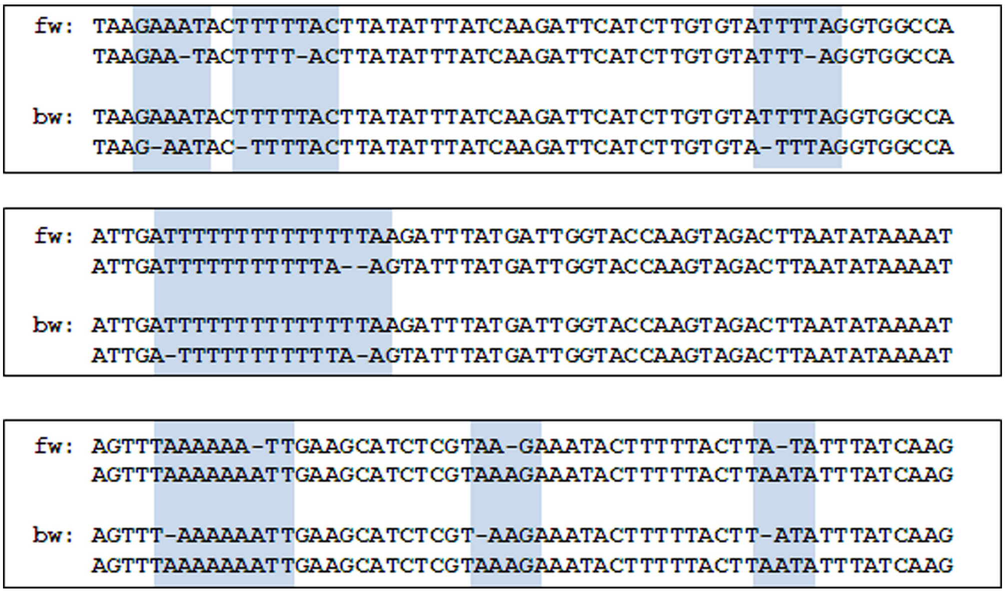
	Contigs	Total size	Av. length	W50
Original	162	10,232,260	63,162	22
Reversed	164	10,229,889	62,377	22
Shuffled	165	10,230,667	62,004	22
		All sections	**With errors**	
3-fold matches		215	36	17%
Total size		9,850,412	378,653	4%
Alignment errors		162	162	
Bases per error		60,805	2,337	

The upper block provides summary data for the three assemblies while the lower block describes 3-fold complete matches of contigs resulting from original, reversed and shuffled reads.

which are the exact complements of the contig derived from the original data. As contig assembly cannot identify sequence strand these triplets define another large part of the genome that must be considered assembled independent of input order. Seven more triplets were identified which agree with just one error (mismatch or gap) in six cases and 11 errors over a length of 13,664 base pairs in the seventh case. There remain 15 contigs from the original, 17 contigs from the reverse and 18 contigs from the shuffled data, which deviate from each other more seriously. In one case the Celera assembler derived the same long contig from the reversed and shuffled data, but from the original data, it derived two shorter contigs exactly matching both ends of the long contig, leaving an nearly 1 KB gap between the two. In another interesting case the assembler generated exactly the same contig for all three data sets but since the first 462 nucleotides are the same as the last 462 nucleotides, the contig is self-overlapping and causes a number of redundant 3-fold complete matches. This example proves the possibility of overestimating the total size of 3-fold completely matching genome sections. In case such sections match with errors we also overestimate the number of errors caused by sequence assembly. Therefore our method of counting assembly errors will still need some improvement. But even if our estimated number of errors due to shuffling input is too large such errors are proven to exist and we suggest working on this problem first. Similar results were found studying the assemblers Phrap, Mira2 and Mira3 on the shuffled data.[10]

3.3. Alignment ambiguities

One possible reason why assembly results depend on shuffling reads is the way alignments are calculated in the overlapping phase and how they are used to form layouts. Overlap alignments generally depend on the direction they are calculated. We show in Figure 3 a



```
fw: TAAGAAATACTTTTTACTTATATTTATCAAGATTCATCTTGTGTATTTTAGGTGGCCA
    TAAGAA-TACTTTT-ACTTATATTTATCAAGATTCATCTTGTGTATTT-AGGTGGCCA

bw: TAAGAAATACTTTTTACTTATATTTATCAAGATTCATCTTGTGTATTTTAGGTGGCCA
    TAAG-AATAC-TTTTACTTATATTTATCAAGATTCATCTTGTGTA-TTTAGGTGGCCA
```

```
fw: ATTGATTTTTTTTTTTTTAAGATTTATGATTGGTACCAAGTAGACTTAATATAAAAT
    ATTGATTTTTTTTTTA--AGTATTTATGATTGGTACCAAGTAGACTTAATATAAAAT

bw: ATTGATTTTTTTTTTTTTAAGATTTATGATTGGTACCAAGTAGACTTAATATAAAAT
    ATTGA-TTTTTTTTTTA-AGTATTTATGATTGGTACCAAGTAGACTTAATATAAAAT
```

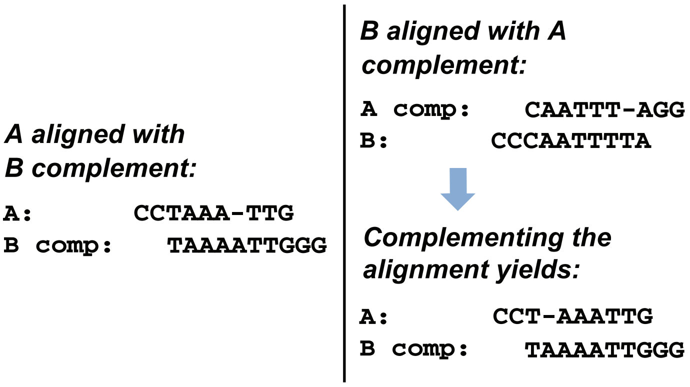
```
fw: AGTTTAAAAAA-TTGAAGCATCTCGTAA-GAAATACTTTTTACTTA-TATTTATCAAG
    AGTTTAAAAAAATTGAAGCATCTCGTAAAGAAATACTTTTTACTTAATATTTATCAAG

bw: AGTTT-AAAAAATTGAAGCATCTCGT-AAGAAATACTTTTTACTT-ATATTTATCAAG
    AGTTTAAAAAAATTGAAGCATCTCGTAAAGAAATACTTTTTACTTAATATTTATCAAG
```

Figure 3. Examples of direction dependent alignments.

few examples of ClustalW alignments calculated from left to right and from right to left with gaps placed in different sequence positions. There seems to be no way of alignment back tracking that can produce a direction independent solution in all cases. The set of all possible optimal alignments is clearly independent of the direction of computation but its consideration is computationally prohibitive.

Hence as each assembler calculates, and for saving time, also stores pairwise alignments for later use, a simple alignment ambiguity can make its way up to the final result of assembly.

Note that each assembler more or less arbitrarily forms as long as possible chains of overlapping reads using algorithms supposed to guard from errors caused by repeated genome sequence, chimerical sequences, base calling errors and other dangers. Assuming at some point the assembler finds read A to overlap with the complement of read B, the overlap alignment is calculated as seen on the left of Figure 4 and stored away to avoid re-computation of this alignment. But given the input was shuffled; the assembler may first find read B to overlap with the complement of read A. The overlap alignment is calculated as shown on the right of Figure 4. Later on during the layout phase, the assembler searches for ways to extend a layout that ends on read A, it remembers the overlap of read B with the complement of A and turns this match around in order to extend the layout beyond read A. The result, also shown at the right of Figure 4, now differs from that obtained by processing the original set of reads and possibly so also does the consensus sequence derived form the layout of reads. Alignment calculation and complementing a pair of sequences are two not interchangeable operations. A new line of assembly algorithms may have to pay attention to this fact. Figures 2, 3 and 4 suggest that ambiguities caused by single letter runs should first be considered, for simplicity and its importance for 454-sequencing which uses signal intensity to estimate single letter multiplicities.



A aligned with B complement:

```
A:       CCTAAA-TTG
B comp:    TAAAATTGGG
```

B aligned with A complement:

```
A comp:    CAATTT-AGG
B:        CCCAATTTTA
```

Complementing the alignment yields:

```
A:        CCT-AAATTG
B comp:    TAAAATTGGG
```

Figure 4. The effect of exchanging the calculation of an alignment with complementing sequences. Details are described in the text.

4. Conclusions

The comparison of 3-fold complete matches found in different assemblies of the same trace data reveals potential assembly errors that are prone to be missed by evaluating the quality of a single assembly alone. Nothing is easier than feeding an assembly pipeline with reads in shuffled order and to trace differences of the outputs. This causes the assembler to tell us about at least some of the genomic regions which are difficult to assemble, given the data, or in general.

Once knowing potential errors, there are a whole lot of programs available to carefully reconsider the genome regions[11] involved based on remapping all contributing reads. Polymorphisms which are due to alignment ambiguities could be found inconclusive for given data and correlated polymorphisms may be indicative for mixing up reads that come from different instances of repeat regions.

We should perhaps first concentrate on detection of error prone genome regions rather than producing ambiguous results by inventing majority based or other decision rules for combining the results of different assemblers. Although such ideas have a certain potential for improving assembly algorithms, a responsible molecular diagnostic will perhaps prefer to re-sequence genome sections in question and appreciate software that is able to quickly outline such genome regions.

The complex interaction between the true genome sequence, the sample of clones, and the heuristics of the assembly algorithm causes all these problems. While nearly random-like and low repetitive sequences may be assembled easily, large chunks of low complexity and repetitive sequences may render the assembly problem impossible.

In a cross-validation study we observed omission of a single out of five hundred 454-Roche reads to double the length of the two contigs derived from the data. However, computational feasibility of such a cross-validation analysis for large data requires novel and efficient methods to identify highly influential and perhaps instability causing single reads. In order to study the effect of sampling reads, it would be interesting to compare result obtained from a high coverage sequencing experiment with those obtained from multiple lower coverage samples.

Taking such and other measures to improve sequence quality it seems possible that more safe sequence assembly methodology becomes the major effort of genome sequencing. In a current large scale paired end sequencing project we observed the read mapper to run much longer than the sequencing machines. But the more elaborate methods are affordable and even necessary for sequencing short enough genome sections as they are becoming increasingly important for molecular diagnostics of disease and detecting local genome variation.

In summary, by the complex nature of the assembly problem, assemblers are prone to make errors. Different assemblers are expected to make different errors and comparing their results provides important clues for identifying errors.

Such careful considerations are necessary for safely distinguishing true genetic sequence variation from artefacts of sequencing which are due to sampling of reads, erroneous analytics, and imperfect algorithms and software. The far reaching goal is to design new strategies that produce the fewest experimental artefacts. It is also necessary to develop methods that help to distinguish data that can, from those which cannot correctly be assembled under strict error limitations and allow to assign each nucleotide a meaningful measure of reliability. Investigations of sensitivity and specificity of read assemblers to different types of poor data possibly based on multiple independent runs of sequencing are required. We must also work on more robust and faster methods for pair wise and multiple sequence alignment being more adequate for sequence assembly.

Acknowledgements

Thanks to Sven Klages from the MPIMG in Berlin who provided the data and the assembler outputs that were used in this contribution and to two referees for valuable comments.

Disclosures

This manuscript has been read and approved by all authors. This paper is unique and is not under consideration by any other publication and has not been published elsewhere. The authors report no conflicts of interest.

References

1. Levy S, Sutton G, Ng PC, et al. The Diploid Genome Sequence of an Individual Human. *PLoS Biology*. 2007;5(10):e254.
2. Wadman M. James Watson's genome sequenced at high speed. *Nature*. 2008;452(7189):788.
3. Venter JC, Adams MD, Myers, et al. The sequence of the human genome. *Science*. 2001;291:1304–51.
4. Semple CAM, Morris SW, Porteous DJ, Evans KL. Computational Comparison of Human Genomic Sequence Assemblies for a Region of Chromosome 4. *Genome Research*. 2002;12:424–9.
5. Salzberg SL, Yorke JA. Beware of mis-assembled genomes. *Bioinformatics*. 2005;21(24):4320–21, DOI:10.1093/bioinformatics/bt1769.
6. Zimin AV, Smith DR, Sutton G, Yorke JA. Assembly reconciliation. *Bioinformatics*. 2008;24(1):42–5.
7. Kleffe J, Möller F, Wittig B. Simultaneous identification of long similar substrings in large sets of sequences. *BMC Bioinformatics*. 2007;8(Suppl 5):S7 DOI:10.1186/1471-2105-8-S5-S7.
8. Ng PC, Levy S, Huang J, et al. Genetic Variation in an Individual Human Exome. *PLoS Genetics*. 2008;4(8):e1000160.
9. Kidd JM, Cooper GM, Donahue WF, et al. Mapping and sequencing of structural variation from eight human genomes. *Nature*. 2008 May;453(1), DOI:10.1038/nature06862.
10. Hirsekorn A. Untersuchung der Stabilität verschiedener Algorithmen zur Sequenzassemblierung. *Bachelor-Arbeit. Technical University of Applied Sciences Wildau*. 2009.
11. Phillippy AM, Schatz MC, Pop M. Genome assembly forensics: finding the elusive mis-assembly. *Genome Biology*. 2008;9:R55, (DOI:10.1186/gb-2008-9-3-r55).