BIOINFORMATICS ORIGINAL PAPER

Vol. 28 no. 17 2012, pages 2215–2222
doi:10.1093/bioinformatics/bts441

Genome analysis

Improved gap size estimation for scaffolding algorithms

Kristoffer Sahlin[1,*], Nathaniel Street[2], Joakim Lundeberg[3] and Lars Arvestad[1,4]

[1]Department of Computational Biology, mKTH Royal Institute of Technology, Science for Life Laboratory, School of Computer Science and Communication, Solna, [2]Department of Plant Physiology, Umeå Plant Science Centre, Umeå University, Umeå, Sweden, [3]KTH Royal Institute of Technology, Science for Life Laboratory, School of Biotechnology, Division of Gene Technology, Solna, Sweden and [4]Department of Numerical Analysis and Computer Science, Swedish e-Science Research Centre (SeRC), Stockholm University, Stockholm, Sweden

Associate Editor: Alex Bateman

ABSTRACT

Motivation: One of the important steps of genome assembly is scaffolding, in which contigs are linked using information from read-pairs. Scaffolding provides estimates about the order, relative orientation and distance between contigs. We have found that contig distance estimates are generally strongly biased and based on false assumptions. Since erroneous distance estimates can mislead in subsequent analysis, it is important to provide unbiased estimation of contig distance.

Results: In this article, we show that state-of-the-art programs for scaffolding are using an incorrect model of gap size estimation. We discuss why current maximum likelihood estimators are biased and describe what different cases of bias we are facing. Furthermore, we provide a model for the distribution of reads that span a gap and derive the maximum likelihood equation for the gap length. We motivate why this estimate is sound and show empirically that it outperforms gap estimators in popular scaffolding programs. Our results have consequences both for scaffolding software, structural variation detection and for library insert-size estimation as is commonly performed by read aligners.

Availability: A reference implementation is provided at https://github.com/SciLifeLab/gapest

Supplementary information: Supplementary data are availible at *Bioinformatics* online.

Contact: ksahlin@csc.kth.se

Received on March 13, 2012; revised on July 5, 2012; accepted on July 9, 2012

1 INTRODUCTION

The decreasing cost of sequencing has made whole-genome sequencing commonplace (Mardis, 2008). When sequencing a genome, one obtains a large set of short fragments of DNA (usually referred to as reads). Assembling these reads (and optimally reconstruct the genome or chromosomes that were sequenced into one piece) remains a complex task (Pop and Salzberg, 2008; Pop, 2009). Due to the presence of repeats, allelic differences and sequencing errors, the result of an assembly is rarely a complete reconstruction of the genome. Instead, the result is often subsets of reads assembled into longer fragments of genomic sequences referred to as contigs. Since the goal is to recreate the genome, ordering and placing these contigs as they appear on the genome (given that the contigs are correctly assembled) is an important assembly step after contig construction in a process called scaffolding (Huson *et al.*, 2002).

In scaffolding, the problem is to link together contigs in their correct order and orientation using paired reads. These paired reads have some known distance (up to a distribution) between them on the genome. If the two separate reads from a pair map to two different contigs, a relation between the two contigs can be inferred. Contigs that can be linked together represent a longer fragment of the genome where sub-parts of the fragment remain unknown (i.e. the gaps between contigs) and such a fragment consisting of more than one contig is called a scaffold. Modern scaffolder programs involve two steps (either separate or intermixed):

- Finding optimal order and orientation of the contigs with respect to some objective function.
- Checking for paired-read inconsistencies. This involves removing reads that have been mapped with too small/large an insert size (with respect to some generously set thresholds) and reads that have been mapped in the wrong relative orientation (often removed in the optimization step).

After this is done, the distance between contigs is often estimated separately using a maximum likelihood (ML) estimation from the reads that are linking the contigs (Dayarian *et al.*, 2010; Gao *et al.*, 2011; Salmela *et al.*, 2011).

The issue of gap length estimation should not be undervalued: high-quality scaffolding provides an understanding of what is necessary for finishing a genome (Nagarajan *et al.*, 2010), with the contig distances effectively quantifying the unassembled parts of the genome. Bad estimates of contig distances, in particular under-estimates, can mislead finishing. Bad distance estimates can also interfere in genome annotation, e.g. with an underestimated gap suggesting that there is not room for an expected feature, or e.g., suggesting that a potential intron is too large in the case of over-estimation. Furthermore, with contig distances being an essential indicator in the scaffolding process, a systematic bias can impair correct scaffolding.

From currently implemented techniques of gap size estimation, we noticed surprisingly poor estimates even in ideal cases without mapping errors or duplicated reads. Both over- and underestimation of hundreds of base pairs could occur. This prompted our group to model gap size. Our aim is to provide

*To whom correspondence should be addressed.

an accurate gap size estimation to be applied after ordering and orientation of contigs and filtering of paired-read inconsistencies.

It is commonly assumed that the distribution of insert sizes for reads that span over a certain gap (or even over a single position in the genome) is the same as the distribution of the paired end library (often approximated as a normal distribution). This is an erroneous assumption. There is an observation bias from the assumption that reads that span the gap are coming from the whole insert size distribution of the library. Fig. 1 gives examples of two types of biases that can occur and which reads would actually be observed in these cases. A 'negative bias' occurs when the gap size is underestimated due to only observing reads from the upper part of the distribution. Similarly, a 'positive bias' occurs when overestimating the gap by only observing reads from the lower parts of the library insert size distribution.

Another example of positive bias is when aligning reads to a short contig, a case occurring frequently when working with fragmented assemblies. Then, only pairs with shorter insert size will have both reads mapped. This is highly relevant for read aligners such as Bowtie (Langmead *et al.*, 2009) and Burrows Wheeler Aligner (BWA) (Li, H. and Durbin) that are routinely estimating the insert sizes for paired-read libraries. Since they are designed mainly for high-quality reference genomes, they are implicitly using an 'infinite contig length' assumption, and hence underestimate the insert size on fragmented assemblies. A heuristic way to deal with this bias is by choosing only large contigs (compared to the insert size of the library) and consider only paired reads that place sufficiently far from contig ends (Phillipphy *et al.*, 2008).

In Sections 2.1–2.3, we explain a model for estimating gap lengths. Section 2.4 derives an estimate from the model in a special case and gives some intuition behind the estimate. In Section 2.5, the model is used to derive a closed form ML expression and we discuss this expression and its practical implications.

Section 3 shows results of gap estimations from our formula compared to estimations from popular scaffolding programs. The results are discussed in Section 4.

2 GAP SIZE MODELLING

Consider two contigs c_1 and c_2. Our objective in this section is to find a model for the distribution of the insert size lengths x of reads spanning a given gap of length d between c_1 and c_2 (see Fig. 2). Let X be the stochastic variable denoting the insert size of a read pair and d, $|c_1|$, $|c_2|$ be parameters denoting the gap length and the two contig lengths. As input to a scaffolder, one provides the positions of where reads align to a contig, usually obtained using an external read mapper. For a paired read i, we call these observations o_i^1 and o_i^2 and let $o_i^1 + o_i^2 = o_i$. This gives the equality $x_i = o_i + d$ (see Fig. 2). This equality is important to keep in mind, since we will alternate between expressing functions in either $o + d$ or x depending on the context. From this relation, we introduce the stochastic variable O, which is a variable describing the distribution of the observations c_i that comes from pairs linking c_1 and c_2. We want to describe the distribution of O given some gap size d and contig lengths $|c_1|$, $|c_2|$, i.e. O|d, $|c_1|$, $|c_2|$.

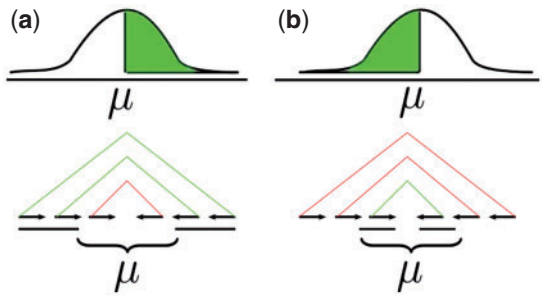

Fig. 1. Illustrating bias in conditioned read-pair insert-length distributions. (**a**) Example of negative bias. Only the longer reads from the library are observed spanning the gap. Thus, deriving the ML estimate for the gap between the contigs assuming the reads are coming from the whole support of the distribution will lead to underestimation of the gapsize. (**b**) Positive bias, only the shorter reads from the library are observed. Thus, deriving the ML estimate for the gap assuming the reads are coming from the whole support of the distribution will lead to overestimation of the gap size.

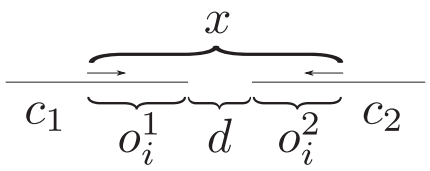

Fig. 2. A paired read with insert size x linking contigs c_1 and c_2. The (unknown) distance between c_1 and c_2 is given by d, which results in the observed distance $o_i = o_i^1 + o_i^2$. Since we have information about the insert size x and $x_i = o_i + d$, this observation carries information about the size of d.

2.1 The model

We let the probability that a paired-end read of length x_i is observed spanning a gap of size d between two contigs c_1 and c_2 be denoted by $p(o|\ |c_1|,|c_2|)$,. Furthermore, let $f(x)$ be the distribution of the insert sizes of the paired-end library.

Now, with the library distribution $f(o+d)$ and the probability of a paired-end read spanning a gap $p(o|\ |c_1|,|c_2|)$, we obtain the distribution of insert sizes from observed reads that span a gap of size d as

$$h(o \mid d, |c_1|, |c_2|) = \frac{p(o \mid |c_1|, |c_2|) f(o+d)}{\int_{-\infty}^{\infty} p(y \mid |c_1|, |c_2|) f(y+d) dy}. \quad (1)$$

Here, the denominator is only for normalizing the function to a probability distribution. The probability function $h(o|\ d,|c_1|,|c_2|)$ indicates that the distribution of reads spanning a gap of size d is explained by the probability that a paired read of given insert size will span the gap times the probability that this pair is being generated by the sequencing protocol (with the functions expressed in o instead of x). Now we have seen the general outline for the model, but what assumptions can we make for f and p?

2.2 Distribution for f

The distribution of insert-size lengths of a paired-end library can be very complex. Usually, the insert-size plots from such libraries look like a normal distribution with thicker tails that have been cut. Since different protocols can vary slightly in the distributions

they generate, it can be nearly impossible to try to model the exact distribution for each sequencing protocol generated. We will here assume a general 'not too incorrect' consensus model for these libraries and therefore work with the normal distribution, in agreement with other authors (e.g. Gao *et al.*, 2011; Lysholm *et al.*, 2011; Richter *et al.*, 2008). One motivation for this, that will become clear after the derivation of the ML estimate, is that with thicker tails, the negative bias will be even larger than in the case of a regular normal distribution. Thus, compared to current methods that estimate gap sizes (Boetzer *et al.*, 2010; Dayarian *et al.*, 2010; Gao *et al.*, 2011; Pop *et al.*, 2004; Salmela *et al.*, 2011), assuming a normal distribution will take us in the right direction of estimating the gap size but we may still leave some of the negative bias due to the assumption of a normal distribution instead of the 'thicker tail' distribution.

To summarize, we let the read insert size be modeled by $X \sim N(\mu, \sigma^2)$, thus f will be the normal density function.

2.3 Deriving p

What is the probability that a read pair of length x will span a gap of size d located somewhere on a genome of length of size $|G|$ given certain lengths of the contigs c_1 and c_2? Under the assumption that the paired reads are generated uniformly throughout the genome (not entirely correct but a reasonable simplification), we can think of this problem as the number of ways that we can cover a segment of d positions with a segment of x positions such that the segment x is within the total length of $c_1| + d + |c_2|$ (recall notations in from Fig. 2). We must, however, take into account that both reads of a pair must map to the contigs. To express the model conveniently, we say that the read lengths of both reads within a paired-end read are equal to r and that the reads must lie completely within a contig in order to span the gap. These restrictions are easy to relax when implementing the model. To be able to untangle the problem of possible placings, we look separately at each of the possible cases that can occur. It can be divided into three different cases (Fig. 3 illustrates these cases).

- The first case (Fig. 3a) is where the contigs are long enough such that the paired read of length x can 'be placed' around the gap of length d freely. In this case, we can cover the gap in precisely $x - d - 2r + 1$ ways.
- In the second case, one of the contigs is so small (say c_1) that it limits the number of placements of a read of length x to $|c_1| - r + 1$.
- The third case is when x is so large compared to the distance $|c_1| + |c_2| + d$ that the number of placements of x over d is limited to $|c_1| + |c_2| + d - x + 1$.

This suggests that $p(x - d \mid |c_1|, |c_2|)$ is a min function consisting of these three functions. So, if $c_{min} = \min\{|c_1|, |c_2|\}$, the probability that a paired-end read of length x will span a gap of size d located somewhere on a genome of length $|G|$ becomes

$$\frac{1}{|G|}\min\Big\{\max\{x - d - 2r + 1, 0\}, c_{min} - r + 1, \max\{|c_1| + |c_2| + d - x + 1, 0\}\Big\}.$$

This function takes the number of possible covering options divided by the number of possible placements of the read to the genome. The max functions within the min function describes that we cannot observe a paired read that is shorter than $d + 2r$ (it will have at least one read placed within the gap) or longer than $c_1 + c_2 + d$ (since they will have at least one read placed outside the two contigs). In practice, reads can be placed partly in contigs if sufficiently many positions overlap, one could then change the integration bounds to the exact ones specified by the read aligners criteria. This will 'however' have a small effect and the general boundaries given here are sufficient to work with.

Note that the divisor $|G|$ is not exact since the true expression of 'the number of possible coverings divided by the number of possible placements of the read to the genome' would give a division with $|G| - x$ instead, but $|G|/(|G| - x) \approx 1$. The motivation for replacing this probability is that the maximum likelihood formulas with respect to d will become a lot easier to differentiate and analyze (x depends on the observation and d) while the approximation says that the change in results will be marginal, if any at all.

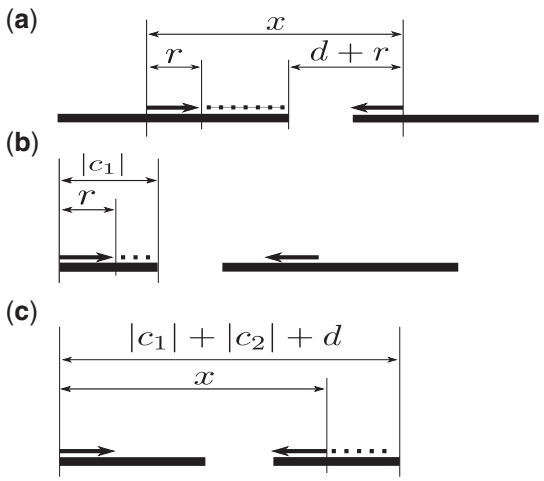

Fig. 3. Range of possible 'placings' of a paired read over a gap (shown as dotted lines). The paired reads (arrows facing each other) are positioned furthest to the left. We can 'slide' the paired read across the contigs until we encounter a restriction. This will give us the number of the possible placings for the paired read. (**a**) Contigs are large compared to x, which makes the restriction of possible placings of a paired read be the insert size of the read (i.e. the number of possible placings is $x - d - 2r + 1$). (**b**) The shortest of the two contigs is so short that it limits the number of possible placements to be the length of the shortest contig minus the read length plus one (i.e. $\min\{|c_1|, |c_2|\} - r + 1$). (**c**) x is so long compared to $|c_1| + |c_2| + d$ that the placement of the read is restricted to $|c_1| + |c_2| + d - x + 1$.

2.4 A special case

Now we have distributions for f and p and therefore implicitly have the distribution of h. As an example, say that we have two different contigs with lengths larger than the longest insert size of a paired-read library and that they have a gap length of 0 bp between them (they are completely adjacent, i.e. $d = 0$). What is then $E_h[O|d]$ (the observed mean link length) of the paired-end reads that connect these contigs? With these restrictions, we have $p(x - d \mid |c_1|, |c_2|) = x - d - 2r + 1/|G|$ (contigs are always sufficiently large). Note that this problem is equivalent to the

problem of finding the distribution of insert sizes of paired reads that are spanning over a position. As mentioned above, it is commonly assumed that this expected value is μ [if $X \sim N(\mu, \sigma^2)$]. Let us see what we get with our model. For the sake of simplicity, we ignore the read lengths r since they are not important for the example (they only subtract or add $2r$ bases if they are included respectively not). We then get

$$\begin{aligned}
E_h[O \mid d=0, |c_1|, |c_2|] = \\
&= E_h[X - d \mid d=0, |c_1|, |c_2|] \\
&= E_h[X \mid d=0, |c_1|, |c_2|] \\
&= \int_{-\infty}^{\infty} x \cdot h(x|d, c_1, c_2)dx = \\
&= \int_{-\infty}^{\infty} y \cdot \frac{\frac{(x+1)}{|G|}f(x)}{\int_{-\infty}^{\infty}\frac{y+1}{|G|}f(y)dy}dx \\
&= \int_{-\infty}^{\infty} \frac{(x^2+x)\cdot f(x)}{\int_{-\infty}^{\infty}(y+1)f(y)dy}dx \\
&= \int_{-\infty}^{\infty} \frac{x^2 \cdot f(x)}{\mu+1}dx + \int_{-\infty}^{\infty}\frac{x\cdot f(x)}{\mu+1}dx \\
&= \frac{1}{\mu+1}E[X^2] + \frac{\mu}{\mu+1} \\
&= \frac{1}{\mu+1}(Var(X) + E[X]^2) + \frac{\mu}{\mu+1} \\
&= \frac{1}{\mu+1}(\sigma^2+\mu^2) + \frac{\mu}{\mu+1} = \mu + \frac{\sigma^2}{\mu+1}.
\end{aligned} \tag{2}$$

This signifies that the expected insert length of the paired reads that link two contigs of distance 0 is some number that is higher than the mean insert size of the library. Similarly, this in fact shows that the distribution of insert sizes over any position on the genome is not the same as the distribution of the library—the mean is larger.

This is intuitive in some sense, since if you would take the 'upper half' of the library and spread across the genome, they would of course cover a larger fraction of the genome than if you take the lower half of the library. Thus, reads from the 'upper half' of the library are much more frequently seen spanning a position than are reads from the 'lower half'.

Furthermore, the additional quantity $\sigma^2/\mu + 1$ depends on the relation between the variance and the mean of the library. This, as well, is not surprising. If you have a high variance, you will have even larger differences (in size) between many of the largest insert sizes compared to the smallest ones. Basically, the smallest insert-size reads do not span many positions together so they are almost never seen spanning a position, while the contrary holds for large insert-size reads. In addition, the divisor basically says that given any variance, if the mean is large enough, the ratio of a small to large reads given the library distribution will not be too large.

2.5 Inferring gap size with maximum likelihood estimation

We will now derive an ML estimator (Le Cam, 1990) from our model in (1). Let $\mathbf{o}$ be a vector of observations. It is then natural to ask: 'What is the most probable value of the gap size provided observations $\mathbf{o}$?' Let $c_{\max} = \max\{|c_1|, |c_2|\}$ and $\mathrm{erf}(x)$ denote the error function of x. The following theorem gives the ML equation.

THEOREM 1. The ML-equation of d for $h(o \mid d, |c_1|, |c_2|)$ as defined above is given by

$$d + \frac{g'(d)}{g(d)}\sigma^2 = \frac{n\mu - \sum_{i=1}^{n} o_i}{n},$$

where

$$\begin{aligned}
g(d) = &\frac{c_{min} - r + 1}{2}\left[erf\left(\frac{c_{max}+d+r-\mu}{\sqrt{2}\sigma}\right) - erf\left(\frac{c_{min}+d+r-\mu}{\sqrt{2}\sigma}\right)\right] \\
&+ \frac{|c_1|+|c_2|+d+1-\mu}{2} \\
&\quad\left[erf\left(\frac{|c_1|+|c_2|+d+1-\mu}{\sqrt{2}\sigma}\right) - erf\left(\frac{c_{max}+d+r-\mu}{\sqrt{2}\sigma}\right)\right] \\
&+ \frac{d+2r-1-\mu}{2} \\
&\quad\left[erf\left(\frac{d+2r-1-\mu}{\sqrt{2}\sigma}\right) - erf\left(\frac{c_{min}+d+r-\mu}{\sqrt{2}\sigma}\right)\right] \\
&+ \frac{\sigma}{\sqrt{2\pi}}\left[e^{\frac{-(|c_1|+|c_2|+d+1-\mu)^2}{2\sigma^2}} + e^{\frac{-(d+2r-1-\mu)^2}{2\sigma^2}}\right] \\
&- \frac{\sigma}{\sqrt{2\pi}}\left[e^{\frac{-(c_{max}+d+r-\mu)^2}{2\sigma^2}} + e^{\frac{-(c_{min}+d+r-\mu)^2}{2\sigma^2}}\right]
\end{aligned}$$

and

$$\begin{aligned}
g'(d) = &\frac{1}{2}\left[erf\left(\frac{|c_1|+|c_2|+d+1-\mu}{\sqrt{2}\sigma}\right) + erf\left(\frac{d+2r-1-\mu}{\sqrt{2}\sigma}\right)\right] \\
&- \frac{1}{2}\left[erf\left(\frac{c_{max}+d+r-\mu}{\sqrt{2}\sigma}\right) + erf\left(\frac{c_{min}+d+r-\mu}{\sqrt{2}\sigma}\right)\right].
\end{aligned}$$

PROOF. See Appendix A.

The nature of this function makes it hard to derive its behavior analytically and attempts of proving monotonicity of this function have not succeeded. However, extensive simulation supports the conjecture that this function is monotonically increasing with d for reasonably large contig sizes $|c_1|, |c_2| \geq \sigma + r$. This result is important since monotonicity leads us to perform binary search of this function to find the ML estimate of d. In practice, it suffices to perform binary search in the interval

$$[\max\{-l, \mu - |c_1| - |c_2| - m\sigma + 2r\}, \mu + m\sigma + 2r]$$

with $m \approx 3$ since it is highly unlikely that we can span other ranges with the given library. That is, we cannot expect to span gaps that lie more than around $\mu + 3\sigma$ bp away. In the same way, we cannot expect to span contig pairs with $|c_1| + |c_2| + d < \mu - 3\sigma$. We have here restricted the lower boundary to $-l$, which is the smallest correct gap we can expect to encounter. How to choose l depends on assembly parameters. A frequently occurring case in a de Brujin-based assembly is that the neighboring contigs overlap with one k-mer size (the algorithm splits at a given node in the graph leaving k base pairs commonly shared). However, since binary search in an interval of length n has complexity $O(\log n)$, it is not too important to chose a tight l.

3 RESULTS

To investigate the bias in gap estimation of current dedicated scaffolders, we simulated a 300 000 bp genome with no biological structure (e.g. free from repeats). This is done to allow us to focus on the concept of pure gap estimation rather than e.g. removal of repeat contigs and mapping errors. Under these simple scaffolding circumstances a reasonable gap estimation model should perform well. Three different libraries were simulated from this genome where paired read positions were uniformly distributed throughout the genome (see Table 1).

In the evaluation, we tested our gap estimation model, implemented in the program GapEst (code provided at https://github.com/SciLifeLab/gapest), against three state-of-the-art scaffolders: SOPRA (v1.4.6), SSPACE (v2.0) and OPERA (v1.02). GapEst takes a SAM-file as input and infers the gaps between contigs by parsing the positions of the mapped reads in the SAM-file. The gap estimation is based on the formula in Theorem 1. Reads were mapped with BWA (Li, H. and Durbin) using default parameters for all programs except OPERA which is coupled with Bowtie (Langmead *et al.*, 2009). We chose BWA since it is able to align soft-clipped reads (read aligned partially to contigs) which gives more observations to our tests.

3.1 Gap underestimation

We tested gap estimation where negative bias can occur (see Case a in Fig. 1). For this, we simulated four different sets of contigs from the genome with fixed gap size set to 30, 300, 650 and 950 bp. The contigs were set to a fixed size of 3000 bp. This gives 12 possible combinations of data (three read libraries and four sets of contigs). The results of the mean gap estimates are shown in Fig. 4.

We see that SOPRA and OPERA are systematically underestimating gap size in most cases and the underestimation increases as d increases. This is intuitive since as the gap gets larger, more reads from the lower parts of the distribution fail to cover the gap and only the longest are left, as explained (Fig. 1). The bias also increases as the variation of the library increases which was predicted in Section 2. SSPACE does not show the same systematic underestimation bias as OPERA and SOPRA, but the predictions are still far from the true ones. GapEst, in contrast, produces unbiased estimates across all combinations of gap size and standard deviations. There are two estimation points that deviate slightly from the true value, gap size 650 for the SD65 library and gap size 950 for the SD300 library. This can occur since there are actually only between 5 and 10 edges that span these gaps in both of the cases, thus estimations become more uncertain.

3.2 Gap overestimation

We tested gap estimation where positive bias can occur (see Case b) in Fig. 1). For this, we simulated four different sets of contigs from the genome with fixed gap size set to −30, 30, 150 and 300 bp with fixed contig length of 300 bp. The negative gap case frequently occurs since a de Brujin-based assembler splits its contigs at a given node in the de Bruijn graph that leaves an overlap (negative gap) of one k-mer length. With these four sets of contigs, we used the paired-end libraries with 65 and 150 bp standard deviation (we advocate the restriction $|c_1|,|c_2| \geq \sigma+r$ for reasonable precision in GapEst). This gives eight possible combinations of data (two read libraries and four sets of contigs). The results of the mean gap estimates are shown in Fig. 5.

Table 1. Synthetic paired reads

	Mean	SD	No. of reads	coverage
lib1	650	65	149 689	∼50×
lib2	650	150	149 654	∼50×
lib3	650	300	147 463	∼49×

Read library statistics from the synthetic dataset.

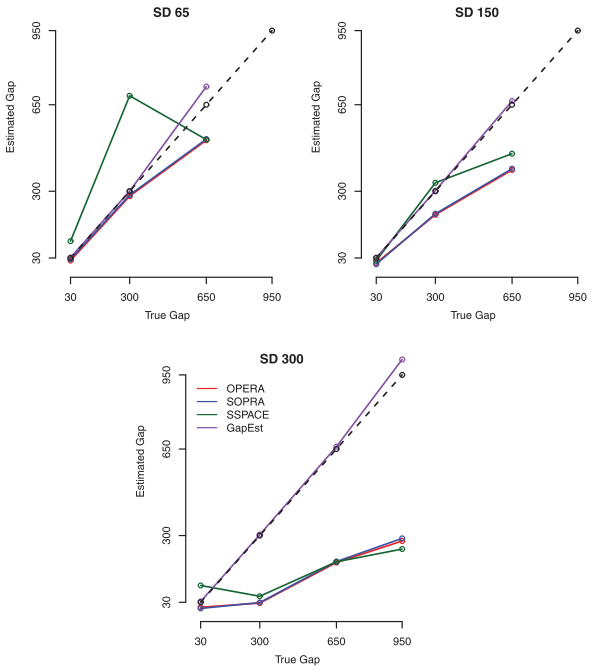

Fig. 4. Gap size estimates for OPERA, SOPRA, SSPACE and GapEst. The three plots visualize the mean estimates of the gaps of sizes of 30, 300, 650 and 950 bp. Each plot corresponds to one paired-read library where the standard deviations of the libraries are of 65, 150, respectively, 300 bp. Data points are missing where the gap is 950 bp since reads fail to span the gaps for SD 65 and SD150.

3.3 A biological insert size library

To show that our model is adequate and applicable for real library insert-size distributions, we downloaded the genome and Solexa/Illumina mate-pair reads for *Staphylococcus aureus* from Genome Assembly Gold-standard Evaluations (GAGE) homepage (Salzberg *et al.*, 2012). The library had 45× coverage, mean insert size of 3.6 kbp and a SD of 275 bp. Contigs were obtained by splitting the reference genome into segments of 5 kbp with gaps of fixed size 500, 1500, 2500 and 3500 bp. Reads were mapped with BWA (Li and Durbin, 2009) using default parameters.

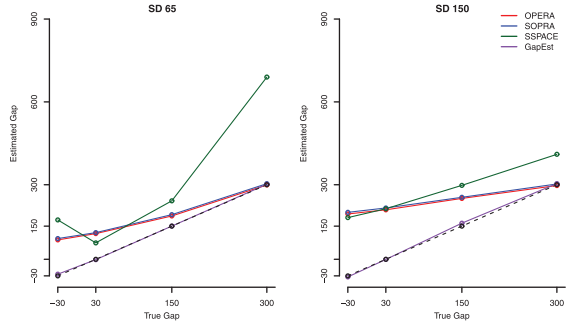

Fig. 5. Gap size estimates for OPERA, SOPRA, SSPACE and GapEst. The three plots visualize the mean estimates of the gaps of sizes of −30, 30, 150 and 300 bp. Each plot corresponds to one paired-read library where the standard deviations of the libraries are of 65 and 150 bp.

Due to the repeated nature of a real genome, a fraction of the mate-pair reads are likely to be mapped incorrectly, causing spurious links between contigs. Since we are only focusing on gap estimation given that contig links are correct, we removed the gap estimates that gave obvious indicators of alignment errors (about 2% of the estimations), as would be done in a scaffolder. Obvious error estimates were conservatively classified as gap estimations over 1500 bp away from true gap sizes together with a significant deviation in number of spanning links. Inspection indicates that there are still spurious links left in the data and a more strict filtering may reduce the standard deviations.

We compared GapEst to SOPRA on this dataset. The results are illustrated in Table 2. For GapEst, the average gap estimates are very close to the true ones. The increasing standard deviation in the estimations of GapEst is due to the decreasing number of links that are able to span the gap. Compared to GapEst, SOPRA has a high standard deviation. One possible explanation for this is that SOPRA does not filter out all spurious links. We found that running GapEst without the filtering step gave only minor differences in mean gap length but a significantly higher standard deviation.

3.4 Gap estimation performance on real assemblies

GAGE provides (unscaffolded) assemblies from seven different assemblers on *S. aureus*. We tested how GapEst performed on these assemblies using the same mate-pair reads as in previous section. MUMmer (Kurtz *et al.*, 2004) was used to align the contigs from an assembly onto the reference genome. For all pairwise combinations of contigs, a gap was estimated if two contigs were on a distance of less than $\mu + 4\sigma$ and the two contigs had over 90% alignment identity and over 95% aligned length. The gaps calculated by MUMmer in this way are referred to as 'true gaps'. We also ran GapEst and obtained gaps between contigs referred to as 'estimated gaps'. A gap was calculated by GapEst for contig pairs passing the following heuristic filters:

- Links placing at distance $\geq \mu + 6\sigma$ were ignored (presumed false mappings).
- Contig pair had at least 10 supporting links.

Table 2. Estimated mean gap lengths

Gap	SOPRA	GapEst	No. of gaps
500	256 ± 234	496 ± 72	465
1500	1176 ± 263	1491 ± 123	402
2500	2119 ± 415	2500 ± 137	334
3500	2805 ± 535	3490 ± 156	283

Gap length estimations for *Staphylococcus aureus* dataset. The true gap sizes are shown in the first column. Average SOPRA and GapEst estimations together with standard deviations are shown in second and third column. Number of gaps are listed in column 4.

- Let the mean of the 10 smallest observations and the 10 largest observations over a gap be $\mu_{\text{low}}^{\text{obs}}$, $\mu_{\text{high}}^{\text{obs}}$, respectively. Then gaps with observations $\mu_{\text{high}}^{\text{obs}} - \mu_{\text{low}}^{\text{obs}} > 6\sigma$ were ignored (presuming that at least part of the observations are false mappings since the spread of the library is so wide).

We plotted the intersection of the gaps inferred by MUMmer and GapEst for the ABySS assembly, Fig. 6 shows the result. Plots of the six other assemblies of *S. aureus*. are given in Supplementary material.

4 DISCUSSION

The scaffolders we tested perform poorly and suffer from severe bias even in ideal conditions. OPERA and SOPRA show very similar gap estimations. Although OPERA estimates several gap sizes simultaneously, this has no advantage in our tests since paired reads do not span over more than one gap. Under these circumstances, OPERA's gap estimator works with one gap at a time, and effectively becomes the one used by SOPRA.

Using simulated datasets, GapEst outperforms the scaffolders tested when estimating gap lengths (Figs 4 and 5). Given the assumptions (in agreement with scaffolding literature) of reads being sequenced uniformly throughout the genome and that the library insert size is somewhat normal, GapEst appears to be unbiased in all cases. GapEst also produces accurate results with the biological data, as shown in Sections 3.3 and 3.4.

This type of gap estimation could open up new ways to tackle the scaffolding problem. For example, using the gap information in the ordering step of the scaffolding algorithm could improve relative placings of the contigs.

As indicated in Section 1, our model for gap size estimation can be used in more general contexts. For example, read aligners commonly estimate insert size by observing reads pairs that map to the same contig. This is a slightly different problem. On one hand, paired reads with large insert size are more likely to span any position on the genome (as seen in Section 2.4), giving rise to negative bias. On the other hand, one will observe less paired reads with large insert size on a given contig, since long reads are more likely to have one of the reads placed outside the contig. This gives rise to positive bias and the problem is exacerbated if contigs are short relative to insert size distribution.

Another problem that requires sensitive estimates of insert-size distributions on a given contig is structural variation detection with paired read information. Here we are facing the same

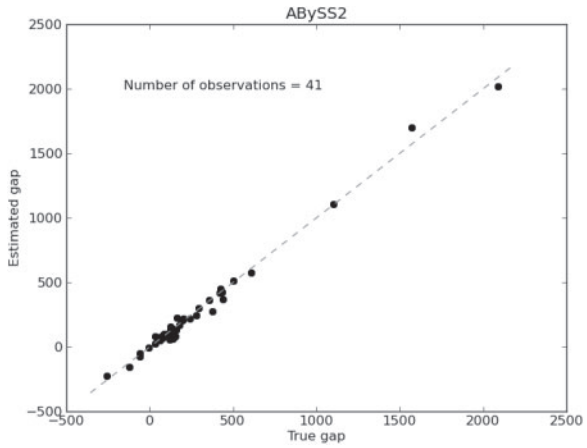

Fig. 6. Gap size estimates for GapEst compared to gap sizes inferred from alignment of contigs to reference genome using MUMmer.

observation bias as in the read alignment problem. Our model provides the framework for deriving the exact distribution in each individual case and using this, the sensitivity in detection of structural variants can be improved.

5 CONCLUSION

We have, in this article, discussed the mathematical theory behind gap estimation. This theory has resulted in a model that explains the most likely length of the gap between two contigs given the observed read pairs that span the gap. Moreover, we have implemented the formula of the ML equation and compared the estimates to state-of-the-art scaffolding software. Empirical results suggest that our model gives a significant improvement in estimating gap length. We encourage developers of existing scaffolder programs to use this formula for gap size estimation where possible.

ACKNOWLEDGEMENT

The authors thank Mattias Frånberg for inspiring discussions about the modeling and Francesco Vezzi for advice on data and testing.

Funding: This work was in part funded by the Swedish Research Council (grant 2010-4634).

Conflict of Interest: none declared.

REFERENCES

Boetzer,M. *et al.* (2010) Scaffolding pre-assembled contigs using SSPACE. *Bioinformatics*, **4**, 578–579.

Dayarian,A. *et al.* (2010) SOPRA: Scaffolding algorithm for paired reads via statistical optimization. *BMC Bioinformatics*, **11**, 345.

Gao,S. *et al.* (2011) Opera: reconstructing optimal genomic scaffolds with high-throughput paired-end sequences. *J. Comput. Biol.*, **11**, 1681–1691.

Huson,D.H. *et al.* (2002) The greedy path-merging algorithm for contig scaffolding. *J. ACM*, **49**, 603–615.

Kurtz,S. *et al.* (2004) Versatile and open software for comparing large genomes. *Genome Biol.*, **5**, R12.

Langmead,B. *et al.* (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.*, **10**, R25.

Le Cam,L. (1990) Maximum likelihood: an introduction. *Int. Stat. Rev.*, **58**, 153–171.

Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*, **25**, 1754–1760.

Lysholm,F. *et al.* (2011) An efficient simulator of 454 data using configurable statistical models. *BMC Res. Notes 2011*, **4**, 449.

Mardis,E.R. (2008) The impact of next-generation sequencing technology on genetics. *Trends Gene.*, **24**, 133–141.

Nagarajan,N *et al.* (2010) Finishing genomes with limited resources: lessons from an ensemble of microbial genomes. *BMC Genom.*, **11**, 242.

Phillippy,A.M. *et al.* (2008) Genome assembly forensics: finding the elusive mis-assembly. *Genome Biol.*, **9**, R55.

Pop,M. (2011) Hierarchical scaffolding with Bambus. *Genome Res.*, **14**, 149–159.

Pop,M. and Salzberg,S.L. (2008) Bioinformatics challenges of new sequencing technology. *Trends Genet.*, **24**, 133–141.

Pop,M. (2009) Genome assembly reborn: recent computational challenges. *Brief. Bioinform.*, **10**, 354–366.

Richter,D.C. *et al.* (2008) MetaSim—A sequencing simulator for genomics and metagenomics. *PLoS One*, **3**, e3373.

Salmela,L. *et al.* (2011) Fast scaffolding with small independent mixed integer programs. *Bioinformatics*, **23**, 3259–3265.

Salzberg,S.L. *et al.* (2012) GAGE: a critical evaluation of genome assemblies and assembly algorithms. *Genome Res.*, **22**, 557–567.

A: PROOF OF THEOREM 1

Given our function $h(o \mid d, |c_1|, |c_2|)$, we have the likelihood function of d (denoted $L(d)$) as

$$\begin{aligned} L(d) &= \prod_{i=1}^{n} h(o_i \mid d, |c_1|, |c_2|) \\ &= \prod_{i=1}^{n} \frac{p(o_i \mid |c_1|, |c_2|) f(o_i + d)}{\int_{2r-1}^{|c_1|+|c_2|+1} p(y \mid |c_1|, |c_2|) f(y + d) dy}, \end{aligned} \tag{3}$$

noting that the integration bounds in the normalizing constant follow from the definition of $p(o|d, |c_1|, |c_2|)$ (it is zero outside this interval). From this, the log likelihood can be written as

$$\begin{aligned} l(d) = &\sum_{i=1}^{n} \ln p(o_i \mid |c_1|, |c_2|) + \sum_{i=1}^{n} \ln f(o_i + d) \\ &- \sum_{i=1}^{n} \ln \int_{2r-1}^{|c_1|+|c_2|+1} p(y \mid |c_1|, |c_2|) f(y + d) dy. \end{aligned} \tag{4}$$

We want to differentiate this expression with respect to d to obtain the maximum likelihood equation for d. By linearity of the differential operator, differentiation of (4) can be performed separately for each term. The first term is constant in d and cancels out. The second term becomes

$$\frac{1}{\sigma^2} \sum_{i=1}^{n} (o_i + d - \mu).$$

Now, if we let $g(d) = \int_{2r-1}^{|c_1|+|c_2|+1} p(y \mid |c_1|, |c_2|) f(y + d) dy$, we can write the third term as

$$\sum_{i=1}^{n} \frac{\frac{\partial}{\partial d}(g(d))}{g(d)} = n \frac{\frac{\partial}{\partial d}(g(d))}{g(d)}.$$

We therefore have

$$l'(d) = \frac{1}{\sigma^2} \sum_{i=1}^{n} (o_i + d - \mu) + n \frac{g'(d)}{g(d)}.$$

Letting this equation be equal to zero to find the ML estimate, we get (after some algebraic manipulations)

$$d + \sigma^2 \frac{g\prime(d)}{g(d)} = \frac{(n\mu - \sum_{i=1}^{n} o_i)}{n}.$$

Now it remains to evaluate $g(d)$ and its derivative. We now use the relation $o = x - d$ and express this integral in terms of $x - d$. With this variable change, we have

$$g(d) = \int_{d+2r-1}^{d+|c_1|+|c_2|+1} p(y-d|d, |c_1|, |c_2|) f(y) dy$$

We then get (by the method 'differentiation under the integral sign')

$$\begin{aligned}\frac{\partial}{\partial d} g(d) = & \overbrace{p(|c_1|+|c_2|+d+1 \mid d)}^{=0} f(|c_1|+|c_2|+d+1) \\ & \cdot \frac{\partial(|c_1|+|c_2|+d+1)}{\partial d} \\ & - \overbrace{p(d+2r-1 \mid d)}^{=0} f(d+2r-1) \cdot \frac{\partial(d+2r-1)}{\partial d} \\ & + \int_{d+2r-1}^{|c_1|+|c_2|+d+1} \frac{\partial}{\partial d} p(y|d) f(y) dy \\ & = \frac{1}{|G|} \int_{d+2r-1}^{|c_1|+|c_2|+d+1} I(y) f(y) dy.\end{aligned} \quad (5)$$

Here, $I(y)$ is the stepwise function obtained from differentiating $p(y|d)$ and it is defined by

$$I(y) = \begin{cases} -1 & \text{if } d+2r-1 \le y \le c_{min}+d+r \\ 1 & \text{if } c_{\max}+d \le y \le |c_1|+|c_2|+d+1, \\ 0 & \text{else} \end{cases}$$

where $c_{min} = \min\{|c_1|, |c_2|\}$ and $c_{max} = \max\{|c_1|, |c_2|\}$. Furthermore, evaluating the integral in (5) gives

$$\begin{aligned}& \frac{1}{|G|}\left[\frac{1}{2}+\frac{1}{2}\mathrm{erf}\left(\frac{y-\mu}{\sqrt{2}\sigma}\right)\right]_{c_{max}+d+r}^{|c_1|+|c_2|+d+1} - \frac{1}{|G|}\left[\frac{1}{2}+\frac{1}{2}\mathrm{erf}\left(\frac{y-\mu}{\sqrt{2}\sigma}\right)\right]_{d+2r-1}^{c_{min}+d+r} \\ & = \frac{1}{2|G|}\left[\mathrm{erf}\left(\frac{|c_1|+|c_2|+d+1-\mu}{\sqrt{2}\sigma}\right)+\mathrm{erf}\left(\frac{d+2r-1-\mu}{\sqrt{2}\sigma}\right)\right] \\ & - \frac{1}{2|G|}\left[\mathrm{erf}\left(\frac{c_{max}+d+r-\mu}{\sqrt{2}\sigma}\right)+\mathrm{erf}\left(\frac{c_{min}+d+r-\mu}{\sqrt{2}\sigma}\right)\right].\end{aligned}$$

Evaluating $g(d)$ gives

$$\begin{aligned}& \left[\frac{1}{2}+\frac{(c_{min}-r+1)}{2|G|}\mathrm{erf}\left(\frac{y-\mu}{\sqrt{2}\sigma}\right)\right]_{c_{min}+d+r}^{c_{max}+d+r} \\ & + \left[\frac{1}{2}+\frac{(|c_1|+|c_2|+d+1-\mu)}{2|G|}\mathrm{erf}\left(\frac{y-\mu}{\sqrt{2}\sigma}\right)+\frac{\sigma}{\sqrt{2\pi}}e^{\frac{-(y-\mu)^2}{2\sigma^2}}\right]_{c_{max}+d+r}^{|c_1|+|c_2|+d+1} \\ & - \left[\frac{1}{2}+\frac{(d+2r-1-\mu)}{2|G|}\mathrm{erf}\left(\frac{y-\mu}{\sqrt{2}\sigma}\right)+\frac{\sigma}{\sqrt{2\pi}}e^{\frac{-(y-\mu)^2}{2\sigma^2}}\right]_{d+2r-1}^{c_{min}+d+r}\end{aligned}$$

Which can be written as

$$\begin{aligned}& \frac{c_{min}-r+1}{2|G|}\left[\mathrm{erf}\left(\frac{c_{max}+d+r-\mu}{\sqrt{2}\sigma}\right)-\mathrm{erf}\left(\frac{c_{min}+d+r-\mu}{\sqrt{2}\sigma}\right)\right] \\ & + \frac{|c_1|+|c_2|+d+1-\mu}{2|G|}\left[\mathrm{erf}\left(\frac{|c_1|+|c_2|+d+1-\mu}{\sqrt{2}\sigma}\right)\right. \\ & \left. - \mathrm{erf}\left(\frac{c_{max}+d+r-\mu}{\sqrt{2}\sigma}\right)\right] \\ & + \frac{d+2r-1-\mu}{2|G|}\left[\mathrm{erf}\left(\frac{d+2r-1-\mu}{\sqrt{2}\sigma}\right)\right. \\ & \left. - \mathrm{erf}\left(\frac{c_{min}+d+r-\mu}{\sqrt{2}\sigma}\right)\right] \\ & + \frac{\sigma}{\sqrt{2\pi}|G|}\left[e^{\frac{-(|c_1|+|c_2|+d+1-\mu)^2}{2\sigma^2}}+e^{\frac{-(d+2r-1-\mu)^2}{2\sigma^2}}\right] \\ & - \frac{\sigma}{\sqrt{2\pi}|G|}\left[e^{\frac{-(c_{max}+d+r-\mu)^2}{2\sigma^2}}+e^{\frac{-(c_{min}+d+r-\mu)^2}{2\sigma^2}}\right].\end{aligned}$$

Finally, we note that $|G|$ will cancel when dividing $g\prime(d)$ with $g(d)$. This is the result we wanted to prove.