Published online 11 May 2012

Nucleic Acids Research, 2012, Vol. 40, No. 16 ***e125***
doi:10.1093/nar/gks393

Using *a priori* knowledge to align sequencing reads to their exact genomic position

René Böttcher[1,2], Ronny Amberg[2], F. P. Ruzius[3], V. Guryev[3], Wim F. J. Verhaegh[1], Peter Beyerlein[1,2] and P. J. van der Zaag[1,*]

[1]Philips Research Laboratories, High Tech Campus 11, 5656 AE Eindhoven, The Netherlands, and [2]University of Applied Sciences Wildau, Bahnhofstraße, 15475 Wildau, Germany and [3]Hubrecht Institute and University Medical Center Utrecht, KNAW, Uppsalalaan 8, 3584 CT Utrecht, The Netherlands

Received January 10, 2012; Revised March 16, 2012; Accepted April 15, 2012

ABSTRACT

The use of *a priori* knowledge in the alignment of targeted sequencing data is investigated using computational experiments. Adapting a Needleman–Wunsch algorithm to incorporate the genomic position information from the targeted capture, we demonstrate that alignment can be done to just the target region of interest. When in addition use is made of direct string comparison, an improvement of up to a factor of 8 in alignment speed compared to the fastest conventional aligner (Bowtie) is obtained. This results in a total alignment time in targeted sequencing of around 7 min for aligning approximately 56 million captured reads. For conventional aligners such as Bowtie, BWA or MAQ, alignment to just the target region is not feasible as experiments show that this leads to an additional 88% SNP calls, the vast majority of which are false positives (~92%).

INTRODUCTION

Since the introduction of so-called next-generation sequencing in 2005, developments in the field of DNA sequencing proceed at a very rapid pace (1). Initially, in the newer sequencing technologies based on massively parallel sequencing (2), the time required to complete a sequencing study was around three weeks, equally divided among sample preparation, the actual sequencing and the bioinformatics analysis. New sequencing technologies are emerging, which promise to reduce the actual sequencing time from the present one week to much shorter. Ultimately, nanopore-based sequencing methods may reduce sequencing run time to matters of seconds (3). Hence, it would be desirable to speed up also the time required in the sample preparation as well as the bioinformatics analysis.

Sequence alignment is a challenge in biology since the first DNA sequences have been determined in the 1970s, with the earliest approaches utilizing dot plots to compute the optimal alignment of the sequences (4). Because of their complexity, dot plots were replaced by the dynamic programming (DP) approach developed by Bellman and Viterbi, first implemented for biological use by Needleman and Wunsch (5,6). Since then, the Needleman–Wunsch algorithm has been modified several times to adapt it to other problems and to improve its performance (7,8). Nevertheless, DP requires too much computation time and space to handle the increasing amount of sequencing data. Therefore, heuristic approaches for searching sequence databases such as BLAST and FASTA were developed to overcome this problem (9,10). Though these programs and their successors are still commonly used, the upcoming of next-generation sequencing requires new software (11) to process the immense amount of short reads created, which lead to the development of hash table based aligners, as for example ELAND and MAQ (12,13). Since then, considerable further effort has been made to reduce the alignment time. One of the most succesful ones is the implementation of a Burrows–Wheeler transform to index the genome and speed up the alignment (14). Common examples of aligners utilizing the Burrows–Wheeler transform are Bowtie and BWA (15,16).

In many branches of electronic data processing the use of *a priori* information is a proven method to improve data analysis. Thus far such an approach has not been adopted in the field of DNA sequencing, although it is conceivable that information arising from so-called targeted sequencing (17–19) could be used to this effect. Typically in targeted sequencing using on-array hybridization (17,18), the fragments of the DNA sample are

*To whom correspondence should be addressed. Tel: +31 40 2749481; Fax: +31 40 2742944; Email: p.j.van.der.zaag@philips.com

hybridized to a microarray with probes designed to capture the fragments of interest. After washing away any non-bound fragments, the DNA fragments of interest for the biological or clinical question at hand are eluted from the array and are further processed to be sequenced. In current practice the resulting eluate is a random mixture of the captured DNA fragments. Moreover, the subsequent alignment of the sequencing reads is done to the whole genome as, at the current specificity of the enrichment methods, aligning to just the target region introduces an unacceptably high error rate, as we will show. In targeted sequencing, one in principle can retain the capture probe information of the micro-genomic selection array, for instance by conducting the sequencing step directly on the capture spot (20) or by using labeled capture beads. Specifically, the very recently proposed oligonucleotide-selective sequencing (OS-Seq) by Myllykangas *et al.* (20) enables this approach. In this method of targeted resequencing target-specific oligonucleotides are used to create 'primer-probes'. These primer-probes are immobilized on the surface of a flow cell and serve both as capture probes and sequencing primers i.e. after capturing the complementary targets from the library, these primer probes are extended. Subsequently, bridge PCR cluster formation is performed. These clusters can be sequenced twice to determine the captured target and subsequently the OS-Seq primer probe sequence (20). This enables the identification of the exact OS-Seq primer that mediated the targeting. Myllykangas *et al.* (20) have used this approach to facilitate the assessment of the performance of individual primer probes.

Here, we would like to investigate the potential benefit of this approach to improve the speed of sequence alignment. To do so we have performed computational experiments to investigate what benefit such an approach of using *a priori* information might bring to sequence alignment and to see whether this can reduce the still sizeable part of the time needed to perform DNA analysis. This investigation has been done by computer-generating a set of sequencing reads that contain the *a priori* known genomic position of their capture probes. These reads are then aligned with an implementation of the Needleman–Wunsch algorithm that uses the *a priori* information to map only to the corresponding sequence fragment. The required alignment time is compared to the time needed by a number of state-of-the-art aligners, which do not use this prior knowledge and which align to the whole genome. Although one could argue that conventional aligners would also be speeded up by aligning only to the target region, we will first show that this is not a viable option by analysis of real enrichment sequencing data, as this yields many false positive SNP calls.

METHODS

Evaluation of the error introduced by alignment to just the target region by conventional aligners

In targeted sequencing, capture arrays are used to reduce the total amount of bases to be sequenced. This reduction is achieved by capturing only the sequences of interest, known as target region. Since enrichment methods do not have a specificity of 100% but typically of around 70% (17,18,21), a considerable amount of off-target reads are generated. Consequently, data from targeted sequencing are aligned to the whole genome, using aligners such as Bowtie, BWA or MAQ, and not just to the target region. To evaluate the error introduced by aligning only to the target region, data (50 bp reads) from a previously published study (21) were used.

The sequencing reads were aligned against the whole genome as well as to the target regions (including 100bp flanks) to evaluate the errors introduced. Subsequently, SNP calling was performed using filtering with the following criteria:

(1) Positions with lower than 20× and higher than 2000× coverage were excluded.
(2) Bases with quality below 10 were excluded from SNP calling.
(3) No more than five reads that have identical mapping position and strand were included.
(4) Each of the non-reference alleles has to be supported by reads mapping to the forward as well as by reads mapping to the reverse strand of the reference genome.
(5) The non-reference allele should be observed in 20% or more reads covering the polymorphic position.
(6) Sites with more than four alleles were excluded as representing positions with increased error rate.

Positions that passed this filtering were called as candidate SNPs (or small indels).

Including *a priori* knowledge in sequence alignment

As the capture probes of hybridization arrays are designed to catch specific sequences, their position on the genome must be known in advance. Therefore, if the location of a capture probe on the array as well as its position on the genome are known, the corresponding sequencing read of the captured fragment can be associated with the sequence of its expected mapping position within the target region, provided that this information is retained during the sequencing process. Hence, the read can be aligned against this associated 'reference sequence' instead of the whole genome.

To computer-generate reads containing information about the genomic position of their capture sequence and their associated reference sequence, first several different target sequences on the genome were selected to construct a target region of interest (Figure 1). For each of these target sequences, a number of capture probes is assumed that would be present on a hybridization array and act as primers for sequencing. Therefore, the genomic position of a sequencing read as well as its associated reference sequence is located behind the capture probe. To cover the complete target sequence with sequencing reads, the capture probes need to be shifted along the genome, which results in the reference sequences being shifted as well to form a tiling of the target sequence with a constant offset (Figure 1A). Taking the reference

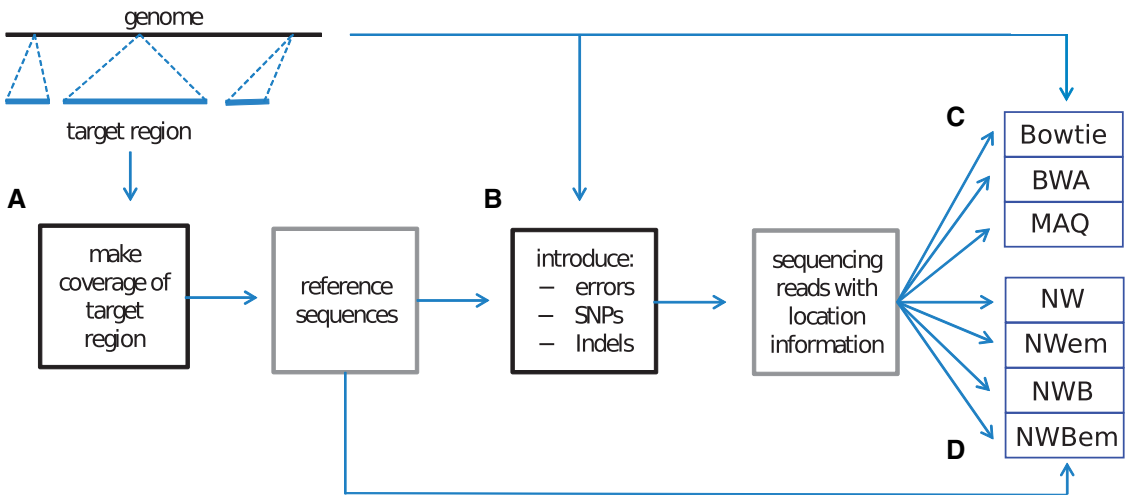

Figure 1. Overview of the workflow. (**A**) A target region was chosen from which the reference sequences were created. (**B**) Each reference sequence was then used to create the associated reads. To simulate realistic data, errors, SNPs and Indels were introduced. The resulting reads were then aligned to the whole genome (**C**) or to their associated reference sequence (**D**).

sequences as templates, we next introduced errors, SNPs and Indels to simulate the sequencing reads (Figure 1B). The resulting reads were used as input for the computations to determine the speed performance of our approach compared with a regular alignment. The regular alignment against the whole genome was performed with Bowtie, BWA and MAQ (Figure 1C). For the alignment using the position information, different implementations of the Needleman–Wunsch algorithm were used (Figure 1D). These consist of a regular Needleman–Wunsch (NW) and a pruned version of the Needleman–Wunsch algorithm following the beam search paradigm (22). We refer to the latter implementation as 'banded' Needleman–Wunsch algorithm (NWB). Additionally, both algorithms were implemented using exact matching prior to the alignment to increase the computation speed (NWem and NWBem), as we describe further in the following section.

Different alignment approaches

The first implementation of alignment using position information was realized through a regular Needleman–Wunsch algorithm (NW), which aligns each read to its associated reference sequence. Since the reads are expected to be very similar to the reference sequence, we realized that a *direct string comparison* might be applicable to skip the alignment for exactly matching sequences. This insight led to a second implementation (NWem), which performs the alignment in two steps. First, the information included in the header of each read is used to look up and identify the reference sequence associated to the read being processed, and subsequently the aligner checks whether the compared sequences match exactly. If so, the maximum alignment score is assigned; otherwise, a regular alignment is performed for the two sequences (as has been described in (10,22); allowing up to two Indels for the beam search approach). Since the Needleman–Wunsch algorithm can be optimized for similar sequences, a banded version was also implemented (NWB, as described in the previous section) and exact matching was added (NWBem), which works similarly to NWem.

To compare the new approaches with established alignment methods, the reads were also aligned against the whole human reference genome using Bowtie (0.12.7), BWA (0.5.9-r16) and MAQ (0.7.1). Default settings were used for MAQ (map) and BWA (aln & samse). Bowtie was run using '-a -n 2 -q –solexa1.3-quals – quiet' settings. The calculations were executed on a grid of 1648 cores divided over 206 Dell PowerEdge M600 blade servers, each utilizing two Intel Xeon L5420 Quadcore CPUs @ 2.5Ghz with 16, 32 or 64 GB of random access memory (BiG Grid, see www.biggrid.nl).

Generation of sequencing data

The data necessary to determine the gain of the new alignment approach by comparison to the regular aligments was obtained from reference human genome GRCh37 and a recent gene annnotation (Ensembl database, release 62; http://www.ensembl.org) (23). In total, 7368 exons were chosen as the target region, representing approximately 3 million bases (Mb) based on previous microarray genomic enrichment experiments (21). Exons originating from the X and Y chromosomes as well as extrachromosomal DNA were excluded. A subset of the chosen exons was taken to create also a 300 kb target region (784 exons), while a 30 Mb target region was also assembled to compare the performance for larger data sets (72 943 exons).

Figure 2 shows the principle of the data generation based on the captured sequences (dark green) which are complementary to the capture probes present for instance on a hybridization array. The capture probes would be designed in such a way that the reference sequences (light green) following the captured sequences form a tiling of the target sequence (continuous black). This target sequence is a part of the target region, and might

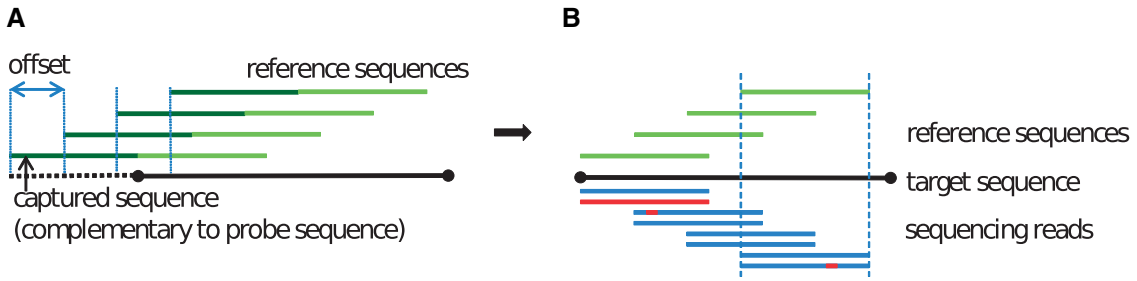

Figure 2. Principle of data generation. (**A**) Captured sequences (dark green) are complementary to the designed capture probes present on the array. These probes are designed in such a way that the following reference sequences (light green) form a tiling of the target sequence (continuous black) with a constant offset. Each reference sequence is therefore directly created from the target sequence. (**B**) For each reference sequence a number of associated reads (blue) is created, introducing different errors (red) in the process. The number of created reads per reference sequence is referred to as read redundancy (two in this example).

be an exon of interest. To generate the sequencing data, each associated reference sequence was created by selecting a substring from the target sequence, while the starting base of the next reference sequence was shifted by an offset of 10 bases, covering the target sequence in the process (Figure 2A). This procedure was repeated until the remaining target sequence was too small to create a new reference sequence with the required length.

The associated reads were then created from their associated reference sequences, with a number of copies referred to as the read redundancy (Figure 2B). As indicated in red in Figure 2, sequencing errors and incorrectly captured reads were introduced into the data set. SNPs and Indels were additionally introduced to the sequencing data with probabilities corresponding to typical occurences mentioned in literature (24). After the read sequence was prepared, the assembly of the read was finished by including the genomic position information.

In the above approach, the length of the reference sequences influences the number of total reference sequences and associated reads, as with increasing length of the reference sequences, fewer complete sequences can be fitted into the target sequences, e.g. the exons chosen. As shown in Table 1, the number of reference sequences decreases for each step of 25 bases. To determine the number of sequencing reads for each combination of target region and read length, the number of reference sequences has to be multiplied by the read redundancy.

Table 1. Number of reference sequences for the different target regions, depending on the length of each reference sequence (as described in the section Generating of sequencing data)

Target region	25 base sequences	50 base sequences	75 base sequences	100 base sequences
0.3 Mb	28 163	26 218	24 243	22 298
3 Mb	283 042	264 616	246 202	227 776
30 Mb	2 857 844	2 676 092	2 493 129	2 311 377

The decrease in number is due to the fact that fewer complete sequences cover the same target if the length of each generated sequence is increased.

Parameter space

To evaluate the influence of various parameters on the alignment time, we varied the values of five parameters:

- the size of the target region (0.3, 3 and 30 Mb),
- the length of the reads (25, 50, 75, 100 bases),
- the percentage of sequencing error per base (0.5%, 1%, 2%),
- the read redundancy (1, 2, 5, 10, 20) and
- the percentage of reads off-target but still captured and sequenced (0, 5, 10, 20, 40%).

RESULTS AND DISCUSSION

Introduction of errors by aligning solely to the target region

As mentioned, the alignment speed of conventional aligners in targeted sequencing could perhaps be improved by aligning just to the target region instead of to the whole genome, which is the current practice (21), because this could seriously reduce the computational effort. To test whether this is a viable option, we first examined the effectiveness of sequence alignment to just the target region, using conventional aligners. Sequencing data from a previous experiment (21) was used for this study.

When using common enrichment methods, two classes of reads are generated, the first one consisting of all reads that originate inside the target region (referred to as ITR) and the second one comprising all reads that originate outside of the target region (referred to as OTR). When all these reads are aligned solely to the target region, two possible errors may occur that influence subsequent analysis (e.g. SNP calling). Firstly, OTRs that now align uniquely inside the target region are falsely classified as uniquely matching reads (UMRs) *to the target*, as they align at a position from which they do not originate (Type 1 error). Secondly, all reads (ITR and OTR) that align uniquely inside the target region, but would also align *one or more times outside* the target region [known as multiple matching reads (MMR)] and that would normally be excluded from analysis, are falsely classified as UMRs as well (Type 2 error).

We compared mapping strategies where reads were aligned to the full genome reference or only to the target. The previously published set (21) features 13.24 million mapped reads of which 8.36 million were uniquely mapped to the target region of genome reference NCBI36. Using the same analysis methods as described in (21), but mapping only against the target region, 8.48 million UMRs were obtained. From these, 0.78% were uniquely mapped to a different location (Type 1 error) and 0.83% were originally MMRs (Type 2 error) when the whole genome was used as a reference.

Subsequently, we evaluated the number of mismatches that were observed in reads that map consistently and in those that correspond to erroneous mappings. The result of this analysis is given in Figure 3. The data show that reads that erroneously map to the target region typically have several mismatches, while the vast majority of consistently mapped reads contains one or no mismatches with the target sequences. However, the distributions overlap and cannot be distinguished easily. For instance, accepting only reads with at most two mismatches to capture most of the consistently mapped reads, would still result in the inclusion of about half the erroneously mapped reads. Setting the threshold to 1 or 0 would on the other hand greatly reduce the information needed for SNP calling. Moreover, the use of a lower threshold to reduce type 1 and 2 errors is not feasible, since an analysis of the distance between SNPs (i.e. SNPs called when mapped against the full reference genome) showed that a third of all SNPs have neighboring SNPs not further than 50 bases apart (see Figure 3). Hence we conclude that allowing fewer than two mismatches per read would reduce the reliability of SNP calling for a substantial part of the exome.

To test the effect of the additional 1.61% UMRs generated, supposedly uniquely mapping to the target region, on genomic analysis, SNP calling was performed [in the same way as done in (21)]. A direct comparison was made for sets mapped against the full genome reference and only to the target region. A total of 1886 SNPs were found in both sets, while an additional 1651 SNPs were specific to the set where mapping was done solely against the target region. Thus aligning to just the target region produces an additional 88% SNPs. The same analysis using 35 bp reads (20) yields similar results and a slightly higher overall false-positive rate (52 versus 47%), indicating that read length has an influence, but will unlikely solve the problem of mismapping. These two different SNP sets exhibit different overlap with a known SNP database: 78.8 and 8.4%, respectively (exact numbers: 1486 and 138, source Ensembl database v.54). The latter percentage implies that nearly 92% of these additionally found SNPs are false positives. In addition, both SNP sets have dissimilar distributions of percentage of non-reference calls, which are given in Figure 4. Figure 4A shows the histogram of the non-reference frequency for the overlapping SNPs in both data sets, while in Figure 4B this histogram is given for the SNPs that are unique to the mapping to the target only. The histogram in Figure 4A exhibits the expected profile with a peak at 100 (homozygous SNPs) and a secondary maximum a bit $<50\%$ expected for heterozygous SNPs. Interestingly the frequency spectrum in Figure 4B exhibits a $1/f$ trend with the frequency, f, which is indicative of noise (25) and suggests—in line with the low overlap with the SNPs known in Ensembl database—that nearly all of these SNPs are false positives. Therefore we conclude that, despite the small proportion of reads with 'paralogous origin' (1.61%) by mapping just to the target region, they are more divergent from the target sequences and therefore can have a significant contribution to false positive SNP calls when detecting sequence variants, in an enrichment experiment when aligning just to the target region.

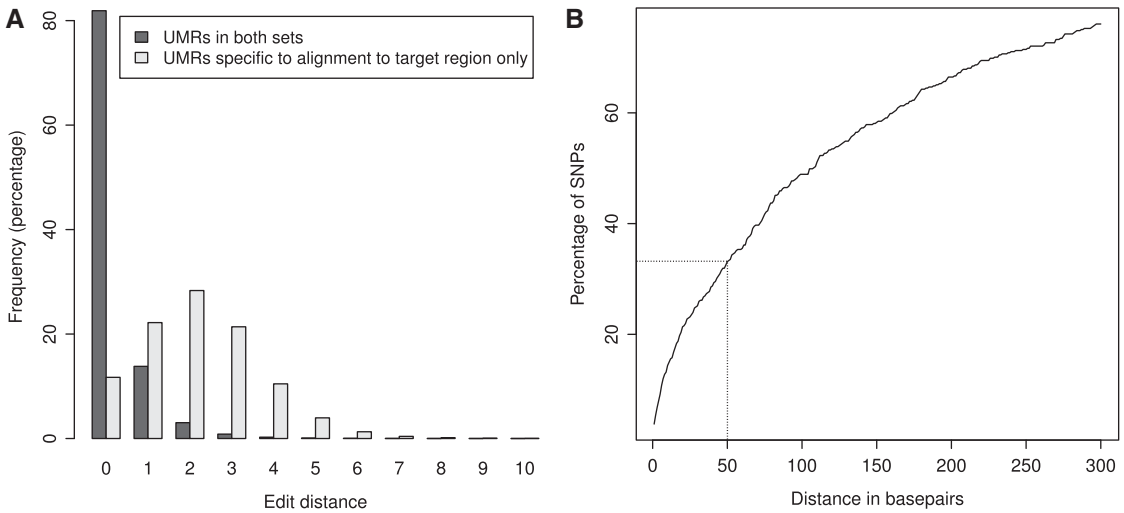

Figure 3. (**A**) Number of mismatches that were observed in reads that map consistently and in those that correspond to erroneous mappings. Reads which erroneously map to the target region typically have several mismatches, while the vast majority of consistently mapped reads have one or no mismatches with target sequences. (**B**) Distribution of distances between neighboring SNPs that map to the same target region of exome. Percentage of between-SNP ranges (Y-axis) that are below a certain distance (base pairs, X-axis) shows that one third of the between-SNP distances are 50 bp or less.

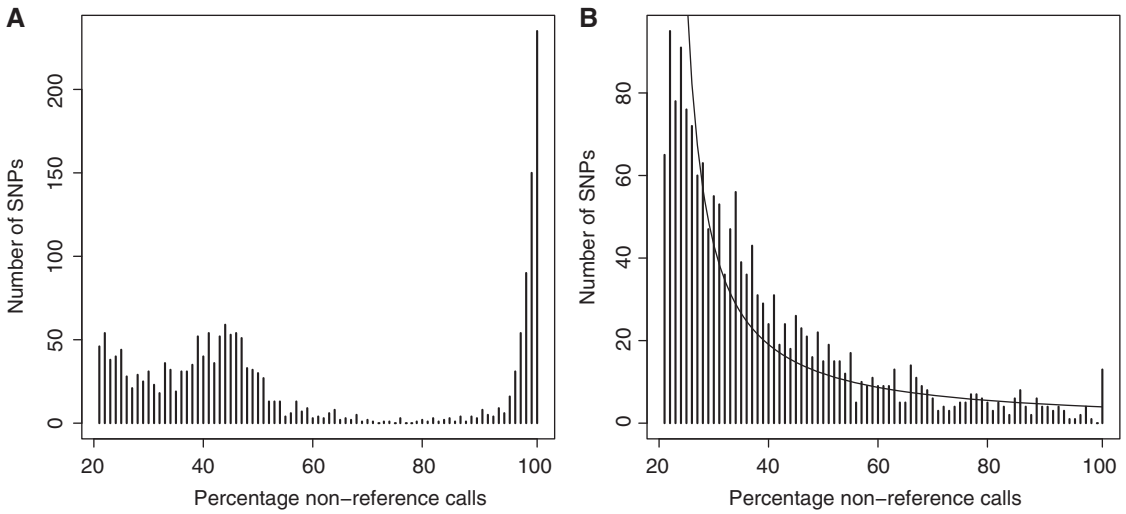

Figure 4. Distributions of percentage of non-reference calls for both SNP sets: (**A**) histogram of the non-reference call frequency for the overlapping SNPs in both data sets, (**B**) histogram for the SNPs specific to the set where read mapping was done to the target only.

Consequently, this validates the practice in targeted sequencing to perform whole genome alignment to avoid introducing additional errors during alignment. Thus, comparisons to determine the gain in alignment speed using *a priori* knowledge will be made by comparing the alignment speed of implementations of the Needleman–Wunsch algorithm, which align to just the target region, to the speed of conventional aligners (Bowtie, BWA, MAQ), which align to the whole genome.

Comparison of alignment speed

To evaluate the alignment speed of the new approach, the computation times required for aligning targeted sequencing experiments were compared to the performance of regular aligners (Bowtie, BWA and MAQ). These latter aligners do not use any *a priori* genome position information and align to the whole genome. Figure 5 shows the results of such a comparison for a 3 Mb target region, a read length of 75 bases, a sequencing error of 1% and with 10% reads off-target. These settings correspond to a total of 246202 reference sequences. Four different implementations of the Needleman–Wunsch algorithm (NW, NWem, NWB and NWBem, see Section Different alignment approaches) were used.

As can be seen, MAQ (red) is the slowest of the aligners used in this comparison, with its computation time ranging from 8713 s up to 69768 s depending on the read redundancy. The two Burrows–Wheeler transform-based aligners perform the same calculations much faster, requiring 661–9419 s (BWA, violet; ~6.86× faster than MAQ) and 159–2791 s (Bowtie, black; ~22.9× faster than MAQ) respectively. These results confirm previous observations concerning the alignment speed of Burrow–Wheeler transform-based aligners (15,16). Nevertheless, the Needleman–Wunsch algorithms using position information lead to considerably shorter alignment times. Compared to Bowtie, the computation time is decreased by a factor of ~1.4 for NW (blue; 106–1949 s), while

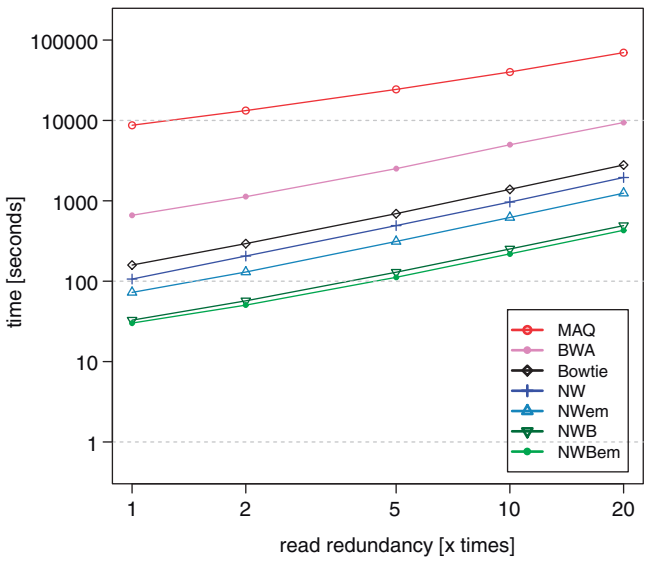

Figure 5. Comparison of the alignment speed of different aligners versus read redundancy. Bowtie, BWA and MAQ aligned against the whole genome; the Needleman–Wunsch implementations used the position information to align to the associated reference sequences. Settings: target size 30 Mb, read length 75 bases, 1% sequencing error, 10% reads off-target. Note that both axes are in logarithmic scale.

NWem (light blue; 73–1244 s) even gains a factor of ~2.2. This gain increases further for NWB (dark green; 32–491 s or ~5.7× faster than Bowtie) and NWBem (green; 30–430 s or ~6.6× faster than Bowtie). Concluding, the total computation time for approximately 49.2 million reads of 75 bases length can be reduced from 46.5 to ~7 min when adapting a pruned Needleman–Wunsch algorithm to use the *a priori* information and comparing to the fastest regular aligner Bowtie.

Figures 6–8 show a more extensive comparison of computational experiments, regarding only two of

the Needleman–Wunsch implementations (NW and NWBem) with a sequencing error of 1% per base in Figures 6 and 7, as well as 2% in Figure 8, respectively. Figure 5 is a subplot of Figure 6 and can be found in the second row and the third column. When investigating over a broader range of conditions, Bowtie (black) shows to be the fastest of the tested common aligners, outperforming MAQ (red) and BWA (violet) in every tested parameter combination. Though the use of the position information still leads to a considerable reduction in alignment time, NW shows limitations for longer reads lengths (due to the time complexity of the regular Needleman–Wunsch algorithm being $O(max(n,m)^2)$), which are overcome by NWBem by pruning the alignment matrix.

For example, in Figure 6, at a length of 100 bases and 40% reads off-target, Bowtie (164–2765 s) and NW (158–2750) compute at comparable speeds, while NWBem outperforms both (32–447 s). When considering shorter reads of 25 bases, both NW (42–583 s) and NWBem (29–396 s) are able to outperform Bowtie (106–1856 s). Concerning the amount of reads off-target, the exact matching shortcut of NWBem is skipped less often at 0% reads off-target and therefore fewer reads have to be aligned regularly (since NW performs no preselection, it is not influenced by this). Still the overall influence on computation time is only marginal, reducing alignment time to 32–445s.

We also investigated the performance of the aligners for the 3 Mb target region (Figure 7) as well as the 300 kb target region (data not shown), which resulted in similar outcomes. In case of the 3 Mb target region, the performance gain varies between a factor of ~1.0 to ~4.3 for NW (average: 2.2 ± 1.2) and a factor of ~5.0 to ~7.7 for NWBem (average: 6.8 ± 0.8) when comparing to Bowtie. Similar results were observed for the 300 kb target region (NW: 2 ± 0.9; NWBem: 6.5 ± 1.1).

When investigating the influence of 2% sequencing error per base for the 30 Mb target region at a length of 100 bases and 40% reads off-target, the results are consistent to previous observations (Figure 8). Compared to 1% sequencing error (see Figure 6 and above), NW (158–2758 s) and NWBem (33–460s) alignment times seem largely unchanged, while Bowtie (196–3311s) requires ~20% more computation time. Hence, for 2% sequencing error and the 30 Mb target region, the average gain for NWBem increases to 7.8 ± 0.8 compared to Bowtie, whereas for the 3 Mb target region it even reaches a factor of 8 ± 0.8. Also compared to

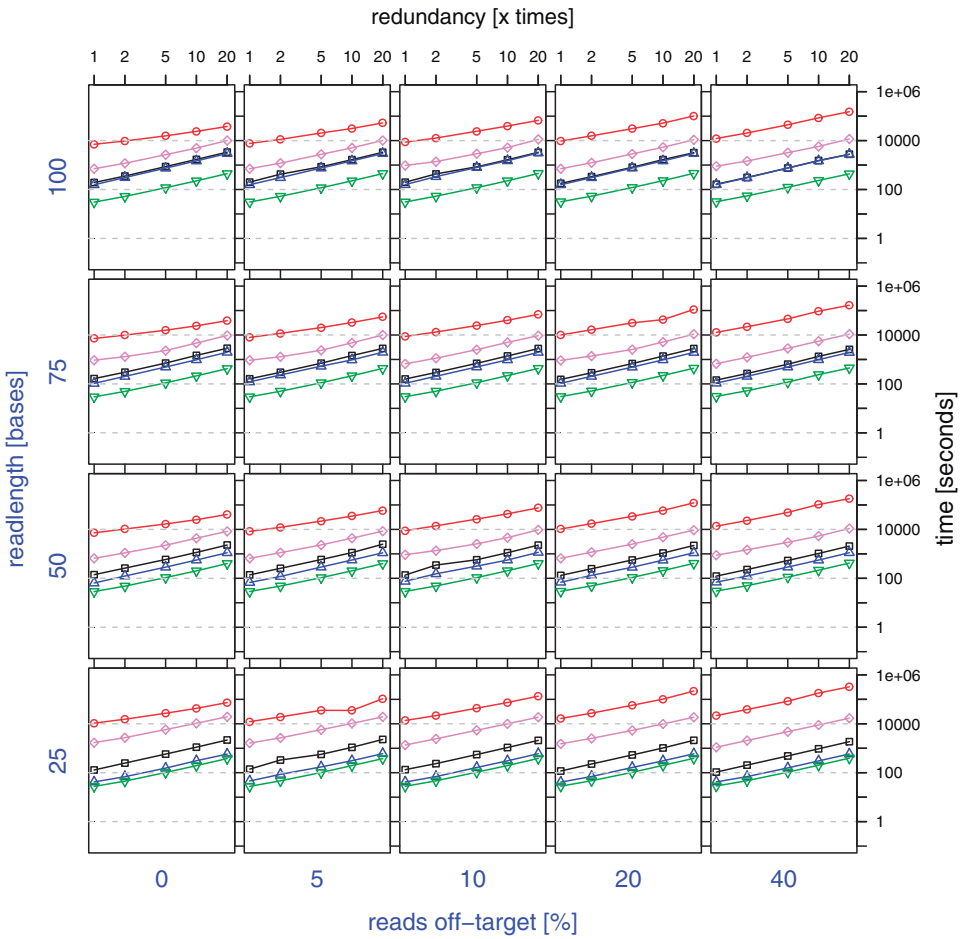

Figure 6. Comparison of different aligners for different read lengths, percentages of reads off-target and read redundancies. MAQ (red), BWA (violet) and Bowtie (black) aligned against the whole genome, NW (blue) and NWBem (green) used the position information to align to the associated reference sequence. Settings: target size 30 Mb, 1% sequencing error.

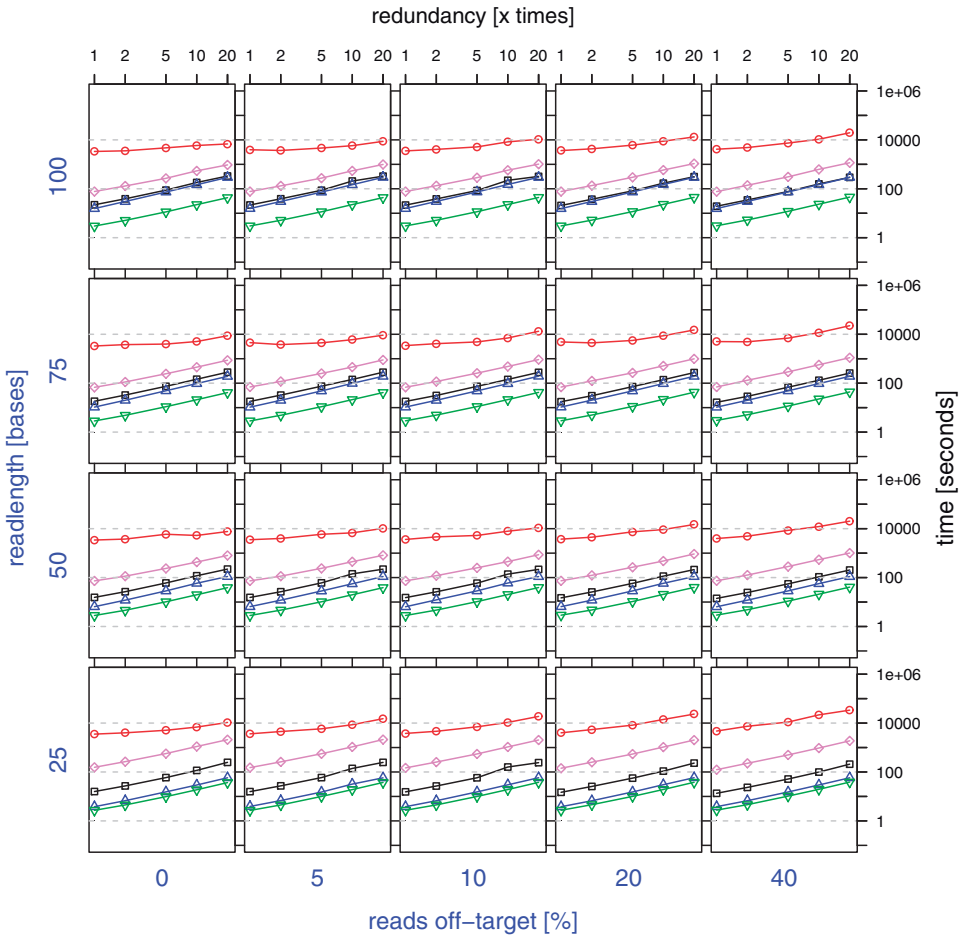

Figure 7. Comparison of different aligners for different read lengths, percentages of reads off-target and read redundancies. MAQ (red), BWA (violet) and Bowtie (black) aligned against the whole genome, NW (blue) and NWBem (green) used the position information to align to the associated reference sequence. Settings: target size 3 Mb, 1% sequencing error.

Bowtie, BWA exhibited a similar behaviour, while MAQ's performance remained stable.

As expected, the amount of reads processed has the biggest impact on the computation time for all of the aligners, with our new approach showing a behavior similar to Bowtie and BWA. The percentage of sequencing error (in our tests up to 2%) influences the computation time of the common aligners (except for MAQ), while it has only a minor effect on the computation time of both NW and NWBem. Nevertheless, this gain in speed is sensitive to the similarity of the aligned sequences to the expected sequences, as it influences the number of exactly matching sequences. Therefore, both implementations using preselection by exact matching (NWem and NWBem) will benefit from a high specificity in enrichment and a low sequencing error.

Concerning the amount of reads off-target, Figure 6 shows that variations in the percentage influence the computation time of both implementations (NW and NWBem) only marginally, with NWBem having the performance of NWB as an upper limit for the computation time when all of the reads need to be aligned in case no exact matches are found (compare Figure 5). This can be understood as for NW, no preselection is performed and therefore all reads are aligned regardless of their origin, while for NWBem the biggest gain in computation time is achieved due to the use of the pruned Needleman–Wunsch algorithm.

Implementation aspects

To investigate whether there is room to improve NW even further, the time consumption of different parts of the Needleman–Wunsch implementations were analyzed. As shown in Table 2, I/O makes up a major part of the total computation time, up to a fraction of 83.3%. Improvements should be possible by using a binary data format instead of the text format used in this study. In summary it can be said that our approach generally benefits from short reads with high quality, as the alignment time for dynamic programming implementations increases with the length of the reads. Furthermore, high-quality reads that match perfectly do not need to be aligned at all.

We next note that BWA and Bowtie benefit from using multiple computer cores, as they can perform their computations multithreaded. MAQ as well as the presented Needleman–Wunsch aligners are not implemented in a multithreaded form (yet) and therefore did not gain from multiple cores.

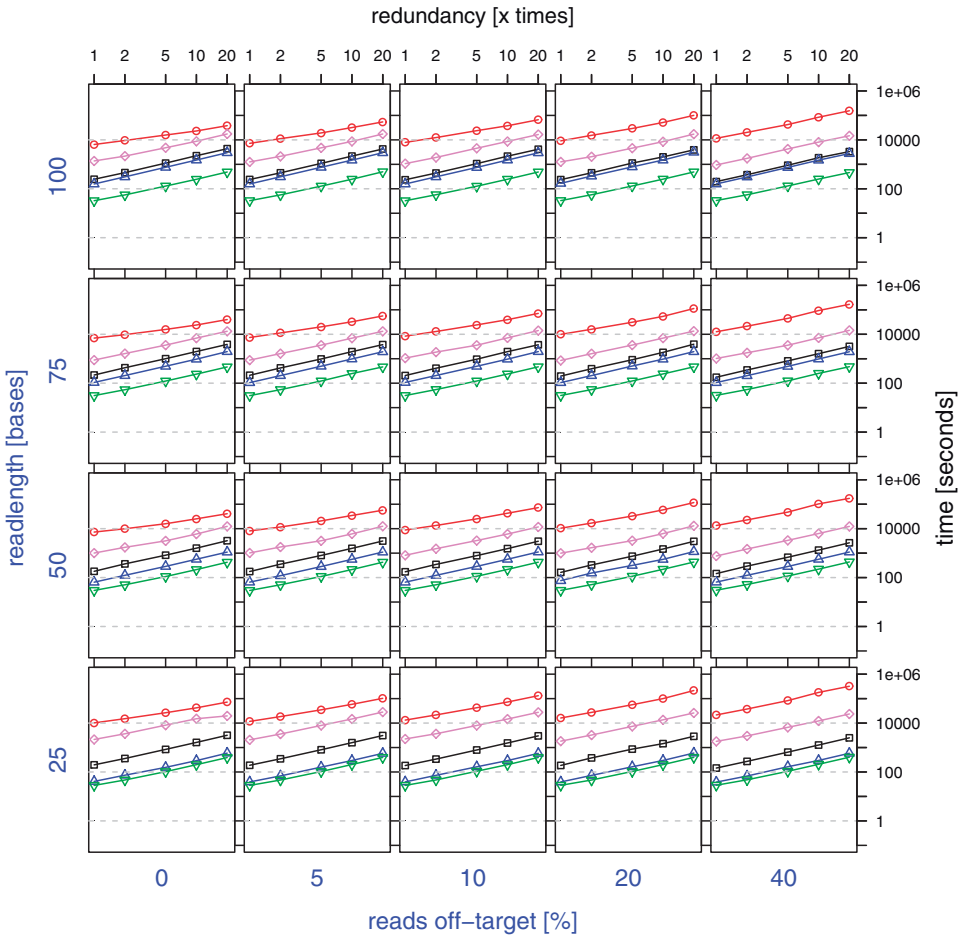

Figure 8. Comparison of different aligners for different read lengths, percentages of reads off-target and read redundancies. MAQ (red), BWA (violet) and Bowtie (black) aligned against the whole genome, NW (blue) and NWBem (green) used the position information to align to the associated reference sequence. Settings: target size 30 Mb, 2% sequencing error.

Table 2. Time consumption of alignment and input/output of the NW and NWBem aligners, for different read redundancies

Program part	1×, *n*(%)	2×, *n*(%)	5×, *n*(%)	10×, *n*(%)	20×, *n*(%)
NW—alignment	3.8 (60.7)	7.92 (66.4)	19.07 (68.9)	39.93 (70.84)	75.75 (69.82)
NW—I/O	2.47 (39.3)	3.99 (33.6)	8.61 (31.1)	16.43 (29.16)	32.75 (30.18)
NWBem—alignment	0.47 (16.71)	0.92 (19.32)	2.28 (21.83)	4.7 (23.15)	9.03 (22.7)
NWBem—I/O	2.33 (83.29)	3.85 (80.68)	8.15 (78.17)	15.59 (76.85)	30.77 (77.3)

The time was measured in seconds, percentages resemble the fraction of total computation time per program part. Settings: target size 3 Mb, read length 50 bases, 1% sequencing error, 10% reads off-target.

Furthermore, the memory requirements for the different aligners vary, making great amounts of RAM advantageous or in case of MAQ necessary for the regular aligners when aligning large numbers of reads. As shown in Table 3, NW and NWBem require only a fraction (7.5–16.6%) of the memory necessary for the other aligners to perform the calculations when aligning approximately 5 million reads from a 3 Mb target region. These low hardware requirements combined with the overall speed of the computations would allow one to

Table 3. RAM requirements (MB) of the different aligners when aligning approximately 5 million reads

Aligner/algorithm	NW	NWBem	Bowtie	BWA	MAQ
Virtual memory required	200	200	1202	2333	2666
Physical memory required	145	145	904	2322	2654

Physical memory required is part of the whole virtual memory required by the program. Settings: target size 3 Mb, 20× read redundancy, read length 50 bases, 1% sequencing error, 10% reads off-target.

include the alignment within the sequencing device, making this kind of post-processing of the sequencing data obsolete in clinical applications.

Outlook

Thus far our work has been focused on methods where the enrichment step and the sequencing are combined in what can be called *embedded enrichment*, such as in OS-Seq (20). However, our method for mapping targeted sequences could be exploited in studies that use other enrichment strategies such as long-range PCR or selector probes (26). One could envision that the high specificity that these methods offer could warrant confining the alignment just to the target region. However, this is not done in practice to avoid generating false SNPs, as even with 98–99% specificity, 1–2% of the amplicons may be misaligned to the target region, if alignment is restricted to this (M. Nilsson, personal communication). Furthermore, as has been shown in the first results section, the vast majority of any additional SNPs generated will be false positives. PCR- and selector-based methods do not necessarily retain a direct link between a probe and the corresponding sequence read through a positional dependence. However, for the selector approach to targeted resequencing (26) a link to the capture probe can be made as the hybridization probes are somewhere in the captured fragment to be read. If these are read as well, the read alignment could proceed by combining this information (giving the expected genomic location) and the read. In the work done by Johansson *et al*. (26) this was not done and alignment was performed against the full genome reference (M. Nilsson personal communication). However, if in between the two selector hybridization probes a specific label is incorporated, which upon sequencing indicates that adjacent to this site both hybridization probes are to be found, then upon the random rolling circle amplification-based multiple displacement amplification the hybridization probes can be easily found in the sequence. Consequently, the genomic location of the fragments would be known and alignment can be done just to the target location in the manner described in this article. For PCR-based enrichment methods the oligonucleotide primers, designed to flank the amplicons, could in principle also be used in the read alignment as *a priori* information. However, in this case new methods would still have to be developed to ensure that the primer information is retained through the concatamerization and/or shearing process, typically applied in the resulting next-generation sequencing library preparation as the PCR-products are longer than the currently typical read length. Thus, as the hybridization probe information can more readily be retained in the selector approach (26), in the latter target enrichment technique our method for targeted alignment might be more readily adopted.

CONCLUSION

In this article we have investigated the use of *a priori* information in sequence alignment, based on a new implementation of current enrichment methods for targeted sequencing. For this purpose, sequencing reads were computer generated from the human genome while varying five parameters to evaluate their impact on alignment time. The presented alignment algorithms are based on straightforward dynamic programming and use *a priori* knowledge to map each read to the expected part of the genome. These implementations prove to be faster than Bowtie, BWA and MAQ. The latter three algorithms align against the whole human genome, since alignment solely to the target region using conventional aligners introduces falsely classified UMRs. We investigated this and found that 1.61% of a total of 8.48 million of the UMRs were incorrectly classified as UMR by aligning just to the target region. This seemingly small percentage of incorrectly classified UMR leads to a significant increase of around 88% more SNP calls, close to 92% of which are false positives.

The gain in computation speed was investigated for a total of 900 parameter variations and was observed to range from an average of 6.2 ± 0.8 for a 30 Mb target region to an average of 8 ± 0.8 for a 3 Mb target region when comparing the fastest Needleman–Wunsch implementation (NWBem) to Bowtie. As the alignment itself consumes only a fraction of the total computation time, using a binary format to process the reads should give additional benefits. For example, speeding up the I/O by a factor of 3 would decrease the alignment time from ~40 s to ~20 s for the ~5 million reads of a 3 Mb target at 20× read redundancy, which is ~16× faster than Bowtie. Furthermore, since the alignment algorithm can be exchanged easily and the computations do not require sophisticated hardware, using *a priori* information proves from a bioinformatics point of view to be a flexible and efficient approach to minimize alignment efforts in targeted sequencing and to enable a clinical use of sequencing information without the necessity of large computing facilities. Finally, the alignment time of around 7 min or less for a targeted resequencing run of approximately 49 million reads would be very attractive for clinical use.

ACKNOWLEDGEMENTS

We would like to thank Peter van Hooft and Jurgen Rusch for their support concerning grid computing facilities and massive parallelized computations. Part of this work was performed using the WIOS pipeline (11). Therefore, we also thank the other WIOS team members, Steffen Pallarz and Anika Tillich for their support. Moreover we would like to thank Harma Feitsma for stimulating discussions and constructive feedback on the manuscript.

FUNDING

Funding for open access charge: Philips Research.

Conflict of interest statement. None declared.

REFERENCES

1. Margulies,M., Egholm,M., Altman,W.E., Attiya,S., Bader,J.S., Bemben,L.A., Berka,J., Braverman,M.S., Chen,Y.-J., Chen,Z. *et al*. (2005) Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, **437**, 376–380.
2. Metzker,M.L. (2010) Sequencing technologies - the next generation. *Nature Rev. Genet*., **11**, 31–46.
3. Dekker,C. (2007) Solid state nanopores. *Nat. Nanotechnol*., **2**, 209–215.
4. Gibbs,A.J. and McIntyre,G.A. (1970) The diagram method for comparing sequences. Its use with amino acid and nucleotide sequences. *Eur. J. Biochem*., **16**, 1–11.
5. Bellman,R. (1957) *Dynamic Programming*. Princeton University Press, Princeton, New Jersey, USA.
6. Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol*., **48**, 443–453.
7. Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. *J. Mol. Biol*., **147**, 195–197.
8. Gotoh,O. (1982) An improved algorithm for matching biological sequences. *J. Mol. Biol*., **162**, 705–708.
9. Altschul,S.F., Gish,W., Miller,W., Myers,E.W. and Lipman,D.J. (1990) Basic local alignment search tool. *J. Mol. Biol*., **215**, 403–410.
10. Lipman,D.J. and Pearson,W.R. (1985) Rapid and sensitive protein similarity searches. *Science*, **227**, 1435–1441.
11. Hammer,P., Banck,M.S., Amberg,R., Wang,C., Petznick,G., Luo,S., Khrebtukova,I., Schroth,G.P., Beyerlein,P. and Beutler,A.S. (2010) mRNA-seq with agnostic splice site discovery for nervous system transcriptomics tested in chronic pain. *Genome Res*., **20**, 847–860.
12. Illumina., Complete Secondary Analysis Workflow for the Genome Analyzer. *Technical Report*, Pub. No. 770-2009-033, Illumina Inc (2009).
13. Li,H., Ruan,J. and Durbin,R. (2008) Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res*., **18**, 1851–1858.
14. Burrows,M. and Wheeler,D. (1994) A block sorting lossless data compression algorithm. *Technical Report*. 124, Digital Equipment Corporation.
15. Langmead,B., Cole,T., Mihai,P. and Salzberg,S.L. (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*., **10**, R25.
16. Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, **25**, 1754–1760.
17. Albert,T.J., Molla,M.N., Muzny,D.M., Nazareth,L., Wheeler,D., Song,X., Richmond,T.A., Middle,C.M., Rodesch,M.J., Packard,C.J. *et al*. (2007) Direct selection of human genomic loci by microarray hybridization. *Nat. Methods*, **4**, 903–905.
18. Hodges,E., Xuan,Z., Balija,V., Kramer,M., Molla,M.N., Smith,S.W., Middle,C.M., Rodesch,M.J., Albert,T.J. and Hannon,G.J. (2007) Genome-wide in situ exon capture for selective resequencing. *Nat. Genet*., **39**, 1522–1527.
19. Gnirke,A., Melnikov,A., Maguire,J., Rogov,P., LeProust,E.M., Brockman,W., Fennell,T., Giannoukos,G., Fisher,S., Russ,C. *et al*. (2009) Solution hybrid selection with ultra-long oligonucleotides for massively parallel targeted sequencing. *Nat. Biotechnol*., **27**, 182–189.
20. Myllykangas,S., Buenrostro,J.D., Natsoulis,G., Bell,J.M. and Ji,H.P. (2011) Efficient targeted resequencing of human germline and cancer genomes by oligonucleotide-selective sequencing. *Nat. Biotechnol*., **29**, 1024–1027.
21. Mokry,M., Feitsma,H., Nijman,I.J., de Bruijn,E., van der Zaag,P.J., Guryev,V. and Cuppen,E. (2010) Accurate SNP and mutation detection by targeted custom microarray-based genomic enrichment of short-fragment sequencing libraries. *Nucleic Acids Res*., **38**, e116.
22. Tillmann,C. and Ney,H. (2003) Word reordering and a dynamic programming beam search algorithm for statistical machine translation. *Comput. Linguistics*, **29**, 97–133.
23. Ensembl Human (Homo sapiens). http://www.ensembl.org/Homo_sapiens/Info/Index.html (20 May 2011, date last accessed).
24. Durbin,R.M., Abecasis,G.R., Altshuler,D.L., Auton,A., Brooks,L.D., Gibbs,R.A., Hurles,M.E. and McVean,G.A. (2010) A map of human genome variation from population-scale sequencing. *Nature*, **467**, 1061–1073.
25. Hooge,F.N., Klein Penning,T.G.M. and van Damme,L.K.J. (1981) Experimental studies on 1/f noise. *Rep. Progress. Phys*., **44**, 479–532.
26. Johansson,H., Isaksson,M., Falk Srqvist,E., Roos,F., Stenberg,J., Sjblom,T., Botling,J., Micke,P., Edlund,K., Fredriksson,S. *et al*. (2011) Targeted resequencing of candidate genes using selector probes. *Nucleic Acid Res*., **39**, e8.