BIOINFORMATICS ORIGINAL PAPER

Vol. 30 no. 1 2014, pages 9–16
doi:10.1093/bioinformatics/btt255

Sequence analysis

Advance Access publication May 17, 2013

Specificity control for read alignments using an artificial reference genome-guided false discovery rate

Sven H. Giese, Franziska Zickmann and Bernhard Y. Renard*

Research Group Bioinformatics (NG4), Robert Koch-Institut, Nordufer 20, 13353 Berlin, Germany

Associate Editor: Michael Brudno

ABSTRACT

Motivation: Accurate estimation, comparison and evaluation of read mapping error rates is a crucial step in the processing of next-generation sequencing data, as further analysis steps and interpretation assume the correctness of the mapping results. Current approaches are either focused on sensitivity estimation and thereby disregard specificity or are based on read simulations. Although continuously improving, read simulations are still prone to introduce a bias into the mapping error quantitation and cannot capture all characteristics of an individual dataset.

Results: We introduce ARDEN (artificial reference driven estimation of false positives in next-generation sequencing data), a novel benchmark method that estimates error rates of read mappers based on real experimental reads, using an additionally generated artificial reference genome. It allows a dataset-specific computation of error rates and the construction of a receiver operating characteristic curve. Thereby, it can be used for optimization of parameters for read mappers, selection of read mappers for a specific problem or for filtering alignments based on quality estimation. The use of ARDEN is demonstrated in a general read mapper comparison, a parameter optimization for one read mapper and an application example in single-nucleotide polymorphism discovery with a significant reduction in the number of false positive identifications.

Availability: The ARDEN source code is freely available at http://sourceforge.net/projects/arden/.

Contact: renardb@rki.de

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

Received on March 15, 2013; revised on April 25, 2013; accepted on April 30, 2013

1 INTRODUCTION

Throughout the past years, the rapid development of next-generation sequencing (NGS) technologies has shaped computational biology. The analysis of large amounts of data from sequencing runs, which regularly reaches millions of reads, is a key challenge in retrieving biological information. Hence, a common part of most NGS applications is to perform a *read mapping*, the search of given read sequences in a much larger reference sequence.

Various methods have been published to efficiently solve the read mapping problem. Popular mappers include Bowtie2 (Langmead and Salzberg, 2012), mrsFAST (Hach *et al*., 2010), BWA (Li and Durbin, 2009) and RazerS (Weese *et al*., 2009). For a comprehensive overview on current read mappers, we refer the reader to (Fonseca *et al*., 2012).

However, it is difficult to judge whether a mapping result is appropriate for a given dataset and how to efficiently compare read mappers and how to choose their increasingly large number of tuning parameters. These challenges in read mapping become particularly apparent in the search for genomic variations, such as single-nucleotide polymorphisms (SNPs). By definition, the sequence of reads indicating SNPs differs from the reference genome. Hence, the difficulty is to distinguish true SNPs from sequencing errors or computational mapping errors. The distinction between error and variant is not obvious in case a read does not match perfectly to the reference. Here, the parameterization of a read mapper plays a crucial role. Using only default settings may result in imperfect mappings, as they might be optimized for certain organisms or sequencing platforms. Allowing mismatches may result in a high number of mappings, but these may be error prone and have a low quality. In contrast, requiring a high similarity might hinder the detection of SNPs. Thus, a method for evaluation and quality control is required to find an optimal setting for a read mapper. To the best of our knowledge, quality control of read mappers is primarily based on sensitivity measurements (Holtgrewe *et al*., 2011) or relies on read simulation as in, e.g. Huang *et al*. (2012), Oliver (2012) and Ruffalo *et al*. (2012), as, in general, no ground truth is available for NGS experiments. However, we observed that adequate read simulations are difficult to achieve and prone to introduce a bias. It is infeasible to model all influence factors on data acquisition and the continuously improving sequencing chemistry poses a challenge to keep simulations up to date. Further, the correlation between simulation result and a specific real dataset is inexplicit, as it is challenging to set parameters such as the error rate of the instrument a priori. To avoid this bias, we developed ARDEN (artificial reference driven estimation of false positives in NGS data), which takes the opposite approach: rather than replacing reads by a simulation with a known ground truth, ARDEN uses real reads and a simulated decoy reference genome for generating confidence measurements. Thereby, ARDEN is able to estimate and to control the number of incorrect alignments.

Similarly to the widely used decoy strategy in proteomics (Elias and Gygi, 2007), our decoy approach is used to estimate the number of false-positive read mappings. This is motivated by the assumption that the number of hits on the decoy genome provides an estimate of the expected number of false positives in the original genome. The expectancy is that the occurrence of one hit on the decoy genome (considered as a random hit) has

*To whom correspondence should be addressed.

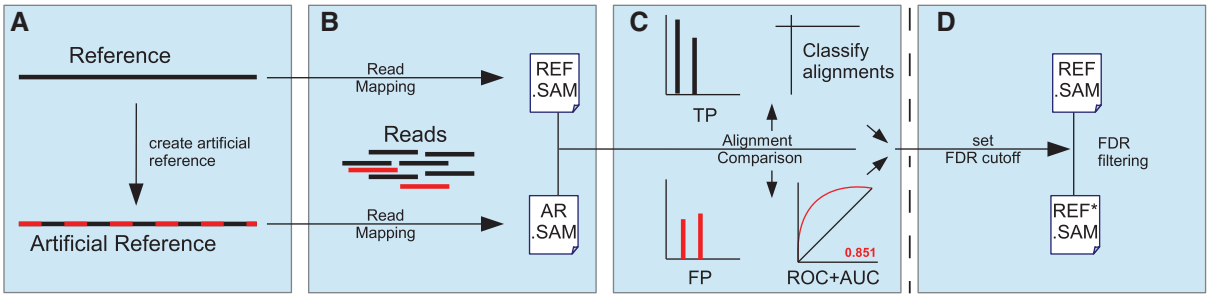

Fig. 1. ARDEN workflow. (**A**) An artificial reference genome is created based on the original genome. (**B**) In a second step, reads are aligned to both references using all read mappers of interest. It is important to note that the settings for one read mapper need to be equal for both, the artificial and the reference genome mappings. (**C**) Based on the mapping results, a comparison is performed by classifying hits as true positives (TP) and false positives (FP). This is used to calculate an ROC table and its corresponding AUC. (**D**) An optional step can be performed to improve the mapping results by excluding aligned reads with suboptimal properties

approximately the same probability as an incorrectly mapped read on the original reference genome. This leads to an approximation of a false discovery rate (Choi and Nesvizhskii, 2008). As no read simulation is necessary, ARDEN is applicable on every dataset and adjusts to the specific sequencing runs. Thus, it can be applied as a concurrent quality control and allows adjusting specificity settings separately for single experiments and the exclusion of potentially incorrect mappings from subsequent analyses. Further, it provides a novel approach to benchmark read mappers or different read mapping settings. We demonstrate the applicability of ARDEN in several evaluations: first, we use ARDEN for a basic read mapper comparison. Further, we determine the best parameter setting for a specific read mapper, and, finally, we show a specific application example for SNP discovery.

2 METHODS

The analysis of a given set of NGS reads includes three major steps and one optional post-processing step, as illustrated in Figure 1. The first step is the creation of an artificial reference genome. This genome serves as a decoy for the read mapper; this strategy is comparable with the false discovery rate approach in proteomic peptide identification using a reversed sequence decoy database (Elias and Gygi, 2007). In the second step, an alignment is performed. The reads are mapped to the reference, as well as to the artificial reference genome using the same settings. The third step is the analysis of the resulting mappings in terms of sensitivity and specificity based on the comparison of the number of mappings with the artificial reference genome (considered as indicators of false positives) and to the reference genome. In the following, each step is explained in detail.

2.1 Creating an artificial reference genome

First, ARDEN creates an artificial reference genome (A) only differing from the reference (R) in single-nucleotide substitutions. Neither structural changes nor insertions or deletions (indels) are introduced. The aim is to change the original reference in a way that none of the input reads has the same origin in R and A. If this is achieved, any hit on A can be classified as a random hit (see Section 2.3 for more details). At the same time, the artificial reference genome requires a maximal similarity to the reference genome to avoid the introduction of any biases. For instance, a change of the GC-content could change the performance of the mapper. Hence, the resulting artificial reference genome is still close to the original sequence, but contains substitutions in a pre-defined distance. These mismatches are randomly chosen, but fulfill the (optional) conditions that a substitution does not change the following properties between A and R:

(i) the nucleotide distribution and thus the GC-content,
(ii) the amino acid distribution,
(iii) the amino acid neighborhood,
(iv) any putative start/stop codons.

A rudimentary gene predictor is implemented to ensure that condition (iv) is not violated. Note that this algorithm is only designed to detect basic open reading frames and, therefore, does not respect splice sites.

It has been shown that these characteristics are specific for each organism (Botzman and Margalit, 2011; Foerstner *et al.*, 2005) and have a major impact on sequencing (Benjamini and Speed, 2012; Schwartz *et al.*, 2011). Especially the amino acid properties are essential, as random substitutions could lead to a bias in the structural folding of the corresponding proteins (Dill and MacCallum, 2012).

Instead of optimizing the conditions (i)–(iv) by an objective function, we choose an exact algorithm to guarantee that the distributions as well as the neighborhood stay the same. This is achieved by exploiting the degeneration of the genetic code. An example iteration for the algorithm to obtain the artificial reference genome is illustrated in Figure 2. The algorithm works as follows:

(1) Choose randomly a position n in the translated protein sequence (first frame translation of the complete genome) and its corresponding codon c_n.
(2) Store the amino acids at positions $n-1$ and $n+1$, as well as the corresponding codons c_{n-1} and c_{n+1}.
(3) Generate a list of possible amino acids whose codon c_i^* has Hamming distance $= 1$ to c_n.
(4) Search for every amino acid triplet corresponding to c_{n-1}, c_i^*, c_{n+1} from (3) in the protein sequence [respecting the constraints (i)–(iv)] and stop when one is found at a position pos_i.
(5) Switch the codon at position n and pos_i.
(6) Start again with (1) until no valid starting positions are left.

2.2 Performing the read mapping

Two mappings have to be computed in one ARDEN run: one on the reference and one on the artificial reference genome. ARDEN is applicable to any read mapper as long as the mapping is presented in the common SAM file format (Li *et al.*, 2009) sorted by read names and

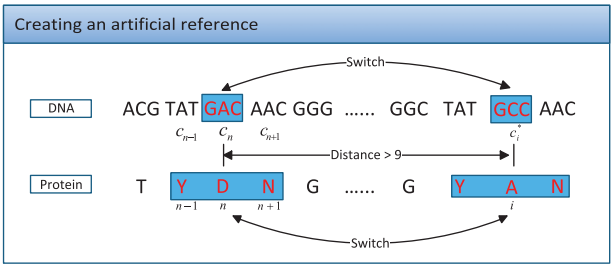

Fig. 2. Example iteration for creating an artificial reference genome. The algorithm starts by choosing amino acid *D* and its codon *GAC* as the triplet to mutate. A possible candidate *GCC* is generated (Hamming distance 1 to *GAC*), and the corresponding amino acid triplet *YAN* serves as a target sequence. Then all occurrences of *YAN* in the protein sequence are checked whether the center amino acid of YAN was generated by a *GCC* in the reference sequence. If that is the case and the distance constraint is fulfilled, the triplets GAC and GCC and the corresponding amino acids are switched. No change has been introduced in the nucleotide distribution and in the amino acid neighborhood. Annotations denote the algorithmic description in Section 2.1

including the MD tag. Refer to the Supplementary Material for a description of formatting details.

2.3 Analysis

To compare different read mappers, we classify the alignments (hits) of one read in potential true positives (PTP) and clear false positives (FP). As we have no ground truth available, we regard each hit on the reference sequence as PTP. For the mappings on the artificial reference genome, a distinction in two cases is possible. Intuitively, all alignments on the artificial reference genome that map to the same position as in the reference sequence are considered as PTPs. A priori, we assume that no mistake has been made when for a read the same origin is found in the reference and artificial genome. Hence, every alignment on the artificial reference genome that does not have the same origin in the reference is denoted as *distinct false hit* and thereby contributes to the FP. The sum of all PTP and FP is declared as *joint hits* (*JH*). Knowing PTP and FP, a global measure between read mappers can be defined. Addressing the PTP comparison, a measure of sensitivity (*Sn*) is calculated as

$$Sn = \frac{PTP}{JH} \cdot M, \tag{1}$$

where M denotes the fraction of mapped reads. This fraction serves as a normalization constant to compensate the fact that some alignment strategies map more reads than others, which may influence the calculation. Note that this way the read mapping probability normalizes the hit probability. Even though there are more sophisticated approaches to define the sensitivity of a read mapper (Holtgrewe *et al.*, 2011), normalizing by the percentage of mapped reads yields a suitable metric. For FPs, a measure of specificity (*Sp*) is defined as

$$Sp = 1 - \left(\frac{FP}{JH} \cdot M\right). \tag{2}$$

This follows the intuition that mappers cannot be specific if they tend to map more reads *distinctly* on the artificial reference.

Combining these measurements, results in a robust statistical analysis of the receiver operating characteristic (ROC) and its corresponding area under the curve (AUC). The ROC curve is created by plotting the sensitivity estimate *Sn* in relation to the specificity estimate *Sp*. Hence, the ROC analysis provides a possibility to identify an optimal trade-off between the sensitivity of a read mapper and its specificity.

To compute positions on the ROC curve for different trade-offs between sensitivity and specificity, we filter all reads according to various alignment features. Thus, features are selected that uniquely define points on the ROC curve. The set of features comprises the number of gaps in the alignment, the number of mismatches and a read quality score (*RQS*), which is calculated as

$$RQS = \frac{\bar{q}_r}{max(q_b) - min(q_b) + 1}, \tag{3}$$

where $\bar{q}_r$ denotes the average quality [a PHRED-based probability that a base call is wrong (Li *et al.*, 2009)] of read r, and q_b is the base quality for a base of the sequence of read r. All alignment features contribute equally to the classification in *PTP* or *FP* and have an individual score range. To reduce the computational effort, we divide each of these individual ranges into five equally sized intervals, yielding a total maximum of $3^5 = 243$ possible combinations (sub-classes). By defining the sensitivity as well as the specificity for each sub-class, a point can be drawn in the ROC curve (Löwer *et al.*, 2012; Sing *et al.*, 2005).

There is no linear relationship between the features, and their impact on read filtering differs. To ensure that the ROC curve is indeed monotonously increasing, we use the envelope of all these points (see Supplementary Material and Fig. 1 for an illustration) and thus implicitly select the most suitable filtering criterion based on the trade-off between sensitivity and specificity. Based on the envelope, an AUC can be calculated using an approximation of the integral of the ROC curve. This AUC provides a value for a combined measure of sensitivity and specificity. All in all, defining the overall sensitivity and specificity and the AUC for a given read mapper yields a robust method for read mapper comparison.

2.4 Optional filtering

As an optional step, it is possible to increase the accuracy of any read mapping result by applying a false discovery rate filter on the input alignments (Fig. 1D). This filter is based on the specified feature list and removes alignments with potentially suboptimal mappings. Hence, it generates a refined SAM-file only, including alignments that fulfill the user specified criteria.

3 EXPERIMENTAL SET-UP AND RESULTS

The use of ARDEN is demonstrated in three different applications. First, we perform a general comparison of read mapping tools on a specific dataset, where we indicate the best mapper in terms of sensitivity and specificity. Second, we perform a read mapper parameterization comparison to identify the best setting for a specific read mapper. Third, we perform a variant calling where we investigate the impact of ARDEN as a pre-processing step on the accuracy of variant identifications.

3.1 Comparison of different read mapping tools

As a test dataset, we chose *Caenorhabditis elegans* reads available on the Short Read Archive (SRA, accession number SRR065388). From these reads, we sampled 1 million single-end reads. As a reference, we used the whole genome assembly from wormbase (Yook *et al.*, 2012) of *C. elegans* (Release WS227). This reference was used along with the artificial reference genome, which was created using ARDEN (Supplementary Material for details). To represent different categories of read mapping approaches, we selected Bowtie2 and BWA as

Table 1. Comparison of the sensitivity (*Sn*) and specificity (*Sp*) of different read mappers using a *C.elegans* dataset with 1 million single-end reads

Mapper	PTP	FP	Sn	Sp	AUC	M
BWA	981 515	25 438	0.895	0.977	0.896	0.919
RazerS2	4 590 324	25 151	**0.921**	0.995	**0.92**	0.926
Bowtie2	945 238	77 084	0.874	0.929	0.859	**0.945**
mrsFAST	**6 528 165**	**328**	0.92	**1**	**0.92**	0.92

Note: M refers to the fraction of mapped input reads. The exact parameters for each mapper are available in the Supplementary Material. The analysis was performed using the analysis module of ARDEN. All values are rounded to three decimal digits. For each column best values are marked in bold.

any-best-mappers (reporting one single best hit), mrsFAST as an all-mapper (reporting all hits regarding a certain error level) regarding Hamming distance and RazerS2 configured as an all-mapper regarding edit distance (Holtgrewe *et al.*, 2011). Each mapper was applied to the test dataset (all parameters are available in the Supplementary Material) and analyzed with ARDEN.

As illustrated in Table 1, the analysis shows that for this dataset, mrsFAST and RazerS2 are the most sensitive read mappers and also have the largest AUC. It can also be observed that although the PTP differ considerably between the mappers, the achieved sensitivity measurements are still within a comparably close range. The same holds for the AUC. Here, the normalization effect of *M* becomes apparent: for instance, although mrsFAST yielded more PTPs than RazerS2, *M* is slightly smaller because of ambiguous mappings, which results in a similar AUC. For this reason, a mapper is regarded as good if it not only yields many PTPs but if these PTPs also have been derived by mapping many different reads.

In terms of FPs, Bowtie2 mapped approximately three times as many reads *distinctly* to the artificial reference genome compared with RazerS2 and BWA. Thus, the AUC for Bowtie2 is the lowest, although Bowtie2 achieves the highest fraction of mapped reads. Hence, many of these reads have been mapped incorrectly, which is reflected in all three accuracy measurements (lowest *Sn*, *Sp* and AUC). The most specific read mapper in this example is mrsFAST, but this advantage is affected by the second lowest *M* value as pointed out earlier in the text.

The corresponding ROC curve (Fig. 3) and its resulting AUC metric provides an easily interpretable measure for the performance of a read mapper. Table 1 can serve as a guideline for the decision, which read mapper to choose for this particular dataset. For instance, here RazerS2 is preferable to mrsFAST to gain a higher sensitivity, at the cost of a decrease in specificity.

Optionally, ARDEN can be used to find a feasible cut-off based on *RQS*, number of gaps and number of mismatches to improve the performance of a specific read mapper (for instance to minimize *Sn* loss while maximizing *Sp*). This can be achieved by filtering the initial SAM-file based on the results from the ROC table created by ARDEN. As an example, an excerpt of the ROC table for Bowtie2 is shown in Table 2 (the complete table is provided in the Supplementary Material). Table 2 can be used to define a cut-off for specificity control because here all alignments are classified according to the used feature set. For example, instead of using all alignments (Table 1), it is possible to

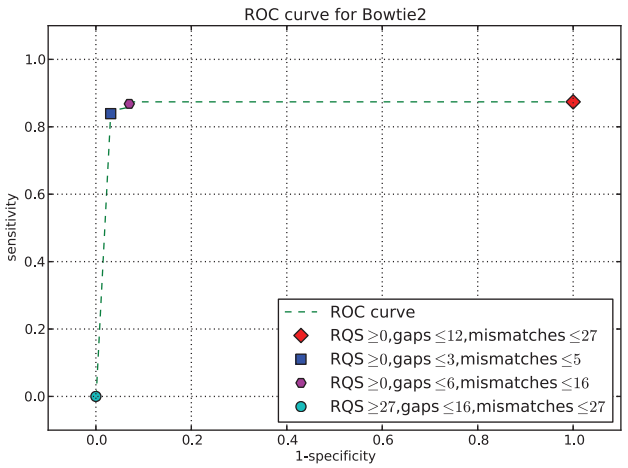

Fig. 3. Example of a ROC curve for Bowtie2 generated by ARDEN. Here, we used a *C.elegans* dataset with 1 million single-end reads. Selected cut-off parameters for *RQS*, number of gaps and number of mismatches are accentuated in the plot

use a specific group of alignments that are selected depending on a user defined cut-off on the alignment features (gaps, mismatches, *RQS*) from Table 2. To achieve a better specificity for Bowtie2 in default settings, it is advisable to choose the cut-off shown in the highlighted row four of Table 2. When only alignments with number of gaps ≤ 16, number of mismatches ≤ 5 and $RQS \geq 0$ are selected, the number of FPs decreases by ~56%, whereas only 1.55% of the PTPs are lost. Choosing this cut-off improves the specificity from 0.929 to 0.969.

3.2 Comparison of different parameterizations for one read mapper

ARDEN not only facilitates the evaluation of different read mappers for a given dataset but also the comparison of different parameterizations to derive the optimal setting for a read mapper on a specific dataset. Without any a priori knowledge, a set of mapping parameters can be tested on the artificial ground truth provided by ARDEN. We demonstrate the usefulness of this strategy on an example using Bowtie2 with different settings on the same dataset as in Section 3.1. As Bowtie2 provides pre-defined settings, we chose three of them and also designed one *custom* setting. The pre-defined settings used here are *very*

Table 2. Excerpt from a resulting ROC table using Bowtie2 and ARDEN

RQS	GAPS	MM	PTP	FP	Sn	Sp	M
0	16	27	945 238	77 084	0.874	0.929	0.945
0	16	10	941 245	73 445	0.866	0.932	0.941
0	3	10	936 150	72 564	0.857	0.934	0.936
0	16	5	930 631	34 217	0.847	0.969	0.931
0	3	5	926 103	33 771	0.839	0.969	0.926
27.215	3	5	775	33	0.0	0.99	0.001

Note: The columns RQS, GAPS and MM indicate the cut-off parameters to divide the alignments in sub-classes. For instance, the first row includes a sub-class that includes all alignments that have an RQS ≥ 0, gaps ≤ 16 and mismatches (MM) ≤ 27.

Table 3. Comparison of different parameterizations for Bowtie2

Setting	PTP	FP	Sn	Sp	AUC	M
Very fast	944 473	**63 344**	**0.885**	**0.941**	**0.875**	0.945
Default	945 238	77 084	0.874	0.929	0.859	0.945
Custom	944 768	78 304	0.873	0.928	0.856	0.945
Very sensitive	**945 487**	84 281	0.868	0.923	0.851	**0.946**

Note: The settings reflect pre-defined configurations of Bowtie2 (*very fast*, *default* and *very sensitive*), as well as a *custom* configuration that adds the $-N = 1$ option to the default setting. For each column, best results are highlighted in bold.

fast, *default* and *very sensitive*, whereas the *custom* setting is similar to the *default* setting along with the $-N = 1$ option (error in seed allowed). Table 3 shows the differences between the performances of the four settings. The number of mapped reads is ≈94.5% and is thereby similar for all four settings. Although the *custom* setting was the only setting allowing mismatches in the seeds, Bowtie2 mapped most false positives when configured as *very sensitive* (it contributes ≈28% to the sum of all FP) and least in the *very fast* configuration, which contributes ≈21% to the sum of all FP. Therefore, the *very fast* setting yields the highest AUC value (Table 3).

As most results are similar (in *PTP*, *FP* and *M*), the AUC values are narrowly distributed (0.851–0.875). For this dataset, the *very fast* configuration is preferable to the other settings, as it yields the highest sensitivity, as well as the highest specificity. Based on ARDEN, the user has the choice to derive more sensitive or more specific results (guided by the artificial ground truth). For example, also the *custom* setting might be a good choice, as it yields more PTPs than the *very fast* modus, whereas it still has a higher *Sn* and *Sp* than the *very sensitive* option. Note that it might still be possible to improve a chosen setting by decreasing the number of FPs. This can be achieved as outlined in Section 3.1 by analyzing the ROC tables provided by ARDEN.

3.3 Improving SNP calling

In general, ARDEN can be applied before any application requiring read mapping, as it provides information on the mapping accuracy and can help to select alignments with a desired trade-off between sensitivity and specificity. To investigate the impact of ARDEN on analysis results, we apply ARDEN to an SNP calling problem. We simulated a SNP ground truth on the complete genome of *Escherichia coli* str. K-12 substr. MG1655 (accession: NC_000913) and chromosome 21 of *Homo sapiens* (accession: NC_000021) and evaluated the impact of our method on the accuracy of variant calling. Similar to the experimental set-up described by Ruffalo *et al*. (2012), we created two new genomes based on the original ones by randomly introducing 150 single point mutations to the *E.coli* genome and 1000 to chromosome 21 of *H.sapiens*. To avoid any biases by mutations introduced during a read simulation process, we chose to use two real datasets, for *E.coli* from the Ion Torrent community and for *H.sapiens* from the SRA (accessions DOC-1443 and DRX000307, respectively; see Supplementary Material for details). The artificial reference genomes were created using the parameters described in the Supplementary Material. The mappings are computed using Bowtie2, RazerS2 [RazerS3 (Weese *et al*., 2012)] and BWA and samtools mpileup (Li *et al*., 2009) were applied for variant calling (refer to the Supplementary Material for details).

Table 4 shows the results for the *E.coli* genome. The category *filtered* refers to all reads remaining after the application of a cut-off determined by the ROC analysis with ARDEN, whereas in *all*, all the alignments reported by the mappers are retained.

When calling SNPs on the *filtered* results the number of TPs remains constant, whereas the number of FPs is reduced by a varying degree (strongly reduced with 72.04% for RazerS2, whereas slightly reduced with 5.05% for BWA). The filtering had almost no effect on Bowtie2. The cut-off for BWA was chosen at (*RQS* $\geq$0/gaps $\leq$6/ mismatches $\leq$ 2), for Bowtie2 at

Table 4. Comparison of SNP calling using *all* alignments and SNP calling with a set of *filtered (Filt.)* alignments defined by ARDEN on a modified *E.coli* genome

Mapper	True positives			False positives		
	All	Filt.	Δ in %	All	Filt.	Δ in %
BWA	127	127	0	198	188	−5.05
Bowtie2	126	126	0	225	224	−0.44
RazerS2	130	130	0	701	196	−72.04

Note: The ground truth contained 150 simulated SNPs. ARDEN decreases the number of FP while retaining all TPs. The effect of filtering depends on the particular mapper and the respective results of ARDEN. For Bowtie2, BWA and RazerS2, the percentage of all alignments that have been removed by the filter are ≈6.8%, ≈2.5% and ≈ 3.4%, respectively. The relative difference between the All and Filt. category is denoted as Δ.

Table 5. Comparison of SNP calling using *all* alignments and SNP calling with a set of *filtered (Filt.)* alignments defined by ARDEN on a modified chromosome 21 of *H.sapiens*

Mapper	True positives			False positives		
	All	Filt.	Δ in %	All	Filt.	Δ in %
Bowtie2	45 342	45 805	+1.02	10 191	10 144	−0.46
RazerS3	46 592	44 069	−5.42	56 954	26 010	−54.33
BWA	48 715	45 058	−7.51	15 681	9612	−38.7

Note: TPs were compared with a simulated ground truth containing 1000 simulated SNPs and to public available SNP data (a more detailed distinction is available in the Supplementary Material). For RazerS3 and BWA, the filtering with ARDEN considerably reduced the numbers of FPs along with a comparably small loss of TPs. For Bowtie2, the number of FPs is decreased along with a gain in TPs. The relative difference between the All and Filt. category is denoted as Δ.

(*RQS* ≥0/gaps ≤2/ mismatches ≤ 4) and for RazerS2 at (*RQS* ≥0/gaps ≤4/ mismatches ≤ 2). The complete ROC tables are provided in the Supplementary Material along with the sensitivity and specificity table.

Table 5 summarizes the results for the application on chromosome 21 of *H.sapiens*. In addition to the simulated ground truth, the results were compared with public available SNP data retrieved via the UCSC table browser [track: All SNPs (137)] (Karolchik *et al.*, 2012). The reduction of false positives is again more pronounced for RazerS3 and BWA than for Bowtie2 with a maximum of 54.33% reduced false positives for RazerS3. On the *H.sapiens* dataset, RazerS3 and BWA lost 5.42 and 7.51% TPs, respectively, whereas the TPs increased on the Bowtie2 mapping. When only considering the simulated ground truth, the TP loss was <1% for all mappers (Supplementary Material). The cut-off for BWA was chosen at (*RQS* ≥0/gaps ≤9/ mismatches ≤ 2), for Bowtie2 at (*RQS* ≥0/gaps ≤13/ mismatches ≤ 4) and for RazerS3 at (*RQS* ≥0/gaps ≤14/ mismatches ≤ 4).

4 DISCUSSION

ARDEN is a method for the identification and control of false positives in mappings of NGS data, for which we demonstrate a broad range of applications.

ARDEN allows the comparison of mapping algorithms on any dataset of interest rather than relying on a simulated dataset with potentially differing properties. The here presented comparison study also gives insight into characteristic algorithmic properties of different classes of read mappers. For example, fewer reads are expected to map *distinctively* to different positions on reference and artificial reference for Hamming-based methods, such as mrsFAST. For these approaches, a single point mutation does not change the start or end position of an alignment that falls into the same region on the reference and the artificial reference. Moreover, Hamming-based methods have harder constraints for finding an alignment, as they only consider substitutions. Because of the seed and extend step, index-based methods suffer from a higher probability for mapping a read to the same region but on a different shifted position. Thus, two classes of errors may contribute to false-positive alignments, shifted alignments and alignments that map to diverse regions. In general, this leads to a higher error rate for mappers, such as Bowtie2 or BWA, than for Hamming distance-based methods. Also, in general, edit distance mappers align a higher percentage of mapped reads (in comparison with Hamming distance mappers) at the cost of an increased probability of false mappings. The reason is that edit distance mapping has relaxed constraints for finding an alignment than Hamming distance mapping, as it allows substitutions and indels. Accordingly, RazerS2 (configured in edit distance mode) and Bowtie2 mapped a higher

fraction of reads than mrsFAST at the cost of specificity. However, BWA did not achieve a higher fraction of mapped reads because it only reports alignments with up to four mismatches, and mrsFAST was configured to map with at most five mismatches (refer to the Supplementary Material).

Note that the ability of ARDEN to improve mappings strongly corresponds to the reporting strategy of the mapper. This fact is demonstrated by Table 3: for instance, Bowtie2, which reports the single best alignment (*any-best*) has a decreased specificity because it makes a random decision for one alignment in cases where n equally good alignments exist. In those cases, the mapping tool makes an error in $\frac{n-1}{n}$ cases because it has a uniformly distributed probability to pick the wrong alignment. If all best mappings are reported, ARDEN can distinguish correct and incorrect alignments without random choices. Hence, the information lost during the *any-best* report hinders the capacity to improve the alignment. For this reason, methods reporting *all* or *all-best* alignments for one read (e.g. RazerS2) have a higher chance to be improved by ARDEN, as more suboptimal hits allow a stronger effect on distinguishing random hits and true positives.

ARDEN allows a dataset-specific optimization of mapping parameters based on the ground truth derived by the artificial reference genome. ARDEN gives an estimate for the trade-off between sensitive and specificity for any parameter setting. For instance, we observed that Bowtie2 achieved the best performance when configured as *very fast*, which opposed the initial expectation. This is valuable information, as it is the fastest of all settings and still the most sensitive. In addition, ARDEN allows the determination of a cut-off where higher error levels stop the improvement of the alignment accuracy but start increasing the number of random hits (FP). Instead of choosing *very sensitive* as the setting that yields the highest percentage of mapped reads (M), the focus shifts to finding a mapping with comparable M and few false positives (e.g. *very fast*). This is important to note, as we see that taking only the percentage of mapped reads M as a general measure of goodness of a read mapper is a common, but suboptimal strategy. Here, we emphasize that maximizing M could lead to error-dominated alignments. Thus, we generally advise to optimize settings for a specific dataset rather than relying on default settings. For example, the *custom* setting also yielded a better sensitivity and specificity than the *very sensitive* mode. This again differs from our expectation, as we assumed that allowing errors in the seed leads to a lower specificity.

Further, ARDEN can be applied to filter alignments to improve the overall specificity. Our experiments showed that *RQS* is the strictest of all three filter dimensions, as one single low base call probability will have a strong negative effect on the score. In addition, Hamming distance alignments do not incorporate gaps in the alignments and will thereby not profit from the gap cut-off dimension. In practice, such an alignment filtering based on the ROC table leads to an improvement of the overall quality of mappings. This has a positive effect on follow-up analyses, as we demonstrated in a SNP calling application. In our experiment, the filtering allowed a correction of up to 72% of falsely called SNPs with few to zero losses of true positives. This emphasizes the strength of the ROC-based filter because considering reads that have been mapped accurately improves on the accuracy of SNP calling. ARDEN helps to identify these reads, e.g. by excluding reads with alignments to *distinctively* different positions on the artificial and reference genome. Naturally, these alignments are prone to errors, and thus disregarding them avoids incorrect SNP calls. However, Bowtie2 did not benefit as strongly from the filtering, as it maps reads with more mismatches and gaps compared with the other methods. These alignments are more likely to be incorrect and are thus more likely not to pass the quality threshold in SNP calling with samtools. Therefore, alignments excluded by ARDEN would have been excluded anyway by the quality filtering for SNP calling by samtools. That is why we see a much smaller effect of ARDEN for Bowtie2 than for other mappers. Note that although samtools provides a robust statistical framework for SNP calling, it does not perform realignments or *de novo* assemblies to deal with complex regions of the genome. For tools supporting these features, such as GATK (DePristo *et al.*, 2011), the described filtering effect might be less effective.

One drawback of ARDEN is the runtime and the necessary post-processing of SAM files. Depending on the mapper and its characteristics, the run time may remain unchanged or increases at maximum by a factor of two, as all reads are mapped to both references. To overcome the problem of increased runtimes, it is possible to approximate the results for the whole dataset by sampling a representative number of reads from the complete set. This smaller set can be used to perform an initial mapping and the following analysis. Our experiments show that the results from the complete dataset and the sampled dataset are similar (see the Supplementary Material). Thus, for most applications, the computational effort can be safely reduced by using a representative sample.

Special care has to be taken when choosing the distance for substitutions for the creation of the artificial reference genome. As in general a substitution introduces a mismatch in the alignment, it has to be ensured that a read mapper can handle the increased error level. Thereby, the optimal distance for substitutions in the artificial reference does depend on the read length and the error threshold for the actual application. Applications that come with high error levels and thus need highly error tolerant read mappers can profit strongly from the approach, whereas ARDEN is of limited benefit for settings with short and error-free reads as well as small, non-redundant reference genomes where alignments are only rarely incorrect.

5 CONCLUSION

Previous approaches to benchmark read mappers either rely on the true sensitivity for a read mapper based on the mathematical formulation of the read mapping problem or are based on read simulations. However, we showed that only considering the sensitivity in read mapping can lead to error-dominated alignments, whereas read simulations are prone to introduce a large simulation bias to the analysis. Thus, it remains doubtful whether read simulation results can be transferred to a specific real dataset. ARDEN overcomes this problem by introducing a decoy reference as an artificial ground truth to obtain an error measurement based on the *distinct* alignments to this decoy genome. This way, ARDEN can concurrently be run to compute and control the rate of incorrect alignments for a specific dataset of interest.

The error measurement allows the dataset specific optimization of mapping parameters and makes read mappers comparable by an AUC metric. Moreover, ARDEN provides the possibility to determine a user-specified cut-off to improve the accuracy of alignments for a specific read mapper based on sensitivity, specificity and the corresponding AUC. More accurate mappings improve the quality of follow-up applications, as we demonstrated in an SNP discovery experiment where using ARDEN decreased the number of false positives by up to 72% while maintaining the majority of true positives.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge Kévin Merlo (NG4, RKI) for partly initial code contribution and Martin Lindner (NG4, RKI) for critical reading of the manuscript.

Funding: Institutional start-up funding by the Robert Koch-Institute (RKI) to B.Y.R.

REFERENCES

Benjamini,Y. and Speed,T.P. (2012) Summarizing and correcting the GC content bias in high-throughput sequencing. *Nucleic Acids Res.*, **40**, e72.

Botzman,M. and Margalit,H. (2011) Variation in global codon usage bias among prokaryotic organisms is associated with their lifestyles. *Genome Biol.*, **12**, R109.

Choi,H. and Nesvizhskii,A.I. (2008) False discovery rates and related statistical concepts in mass spectrometry-based proteomics. *J. Proteome Res.*, **7**, 47–50.

DePristo,M.A. *et al.* (2011) A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.*, **43**, 491–498.

Dill,K.A. and MacCallum,J.L. (2012) The protein-folding problem, 50 years on. *Science*, **338**, 1042–1046.

Elias,J.E. and Gygi,S.P. (2007) Target-decoy search strategy for increased confidence in large-scale protein identifications by mass spectrometry. *Nat. Methods*, **4**, 207–214.

Foerstner,K.U. *et al.* (2005) Environments shape the nucleotide composition of genomes. *EMBO Rep.*, **6**, 1208–1213.

Fonseca,N.A. *et al.* (2012) Tools for mapping high-throughput sequencing data. *Bioinformatics*, **28**, 3169–3177.

Hach,F. *et al.* (2010) mrsFAST: a cache-oblivious algorithm for short-read mapping. *Nat. Methods*, **7**, 576–577.

Holtgrewe,M. *et al.* (2011) A novel and well-defined benchmarking method for second generation read mapping. *BMC Bioinformatics*, **12**, 210.

Huang,W. *et al.* (2012) ART: a next-generation sequencing read simulator. *Bioinformatics*, **28**, 593–594.

Karolchik,D. *et al.* (2004) The UCSC Table Browser data retrieval tool. *Nucleic Acids Res.*, **32** (**Suppl. 1**), D493–D496.

Langmead,B. and Salzberg,S.L. (2012) Fast gapped-read alignment with Bowtie 2. *Nat. Methods*, **9**, 357–359.

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

Li,H. *et al.* (2009) The sequence alignment/map format and SAMtools. *Bioinformatics*, **25**, 2078–2079.

Löwer,M. *et al.* (2012) Confidence-based somatic mutation evaluation and prioritization. *PLoS Comput. Biol.*, **8**, e1002714.

Oliver,G.R. (2012) Considerations for clinical read alignment and mutational profiling using next-generation sequencing. *F1000 Res.*, **1**, doi: 10.12688/f1000research.1-2.v2.

Ruffalo,M. *et al.* (2012) Accurate estimation of short read mapping quality for next-generation genome sequencing. *Bioinformatics*, **28**, i349–i355.

Schwartz,S. *et al.* (2011) Detection and removal of biases in the analysis of next-generation sequencing reads. *PLoS One*, **6**, e16685.

Sing,T. *et al.* (2005) ROCR: visualizing classifier performance in R. *Bioinformatics*, **21**, 3940–3941.

Weese,D. *et al.* (2009) RazerS–fast read mapping with sensitivity control. *Genome Res.*, **19**, 1646–1654.

Weese,D. *et al.* (2012) RazerS 3: faster, fully sensitive read mapping. *Bioinformatics*, **28**, 2592–2599.

Yook,K. *et al.* (2012) WormBase 2012: more genomes, more data, new website. *Nucleic Acid Res.*, **40**, D735–D741.