BIOINFORMATICS ORIGINAL PAPER

Vol. 27 no. 17 2011, pages 2330–2337
doi:10.1093/bioinformatics/btr393

Genome analysis

Advance Access publication June 30, 2011

TotalReCaller: improved accuracy and performance via integrated alignment and base-calling

Fabian Menges[1,*], Giuseppe Narzisi[1] and Bud Mishra[1,2]

[1]Computer Science Department, Courant Institute, New York University, NY 10012 and [2]Quantitative Biology Center, Cold Spring Harbor Laboratory, Cold Spring Harbor, NY 11791 USA

Associate Editor: Martin Bishop

ABSTRACT

Motivation: Currently, re-sequencing approaches use multiple modules serially to interpret raw sequencing data from next-generation sequencing platforms, while remaining oblivious to the genomic information until the final alignment step. Such approaches fail to exploit the full information from both raw sequencing data and the reference genome that can yield better quality sequence reads, SNP-calls, variant detection, as well as an alignment at the best possible location in the reference genome. Thus, there is a need for novel reference-guided bioinformatics algorithms for interpreting analog signals representing sequences of the bases ($\{A, C, G, T\}$), while simultaneously aligning possible sequence reads to a source reference genome whenever available.

Results: Here, we propose a new base-calling algorithm, TotalReCaller, to achieve improved performance. A linear error model for the raw intensity data and Burrows–Wheeler transform (BWT) based alignment are combined utilizing a Bayesian score function, which is then globally optimized over all possible genomic locations using an efficient branch-and-bound approach. The algorithm has been implemented in soft- and hardware [field-programmable gate array (FPGA)] to achieve real-time performance. Empirical results on real high-throughput Illumina data were used to evaluate TotalReCaller's performance relative to its peers—Bustard, BayesCall, Ibis and Rolexa—based on several criteria, particularly those important in clinical and scientific applications. Namely, it was evaluated for (i) its base-calling speed and throughput, (ii) its read accuracy and (iii) its specificity and sensitivity in variant calling.

Availability: A software implementation of TotalReCaller as well as additional information, is available at: http://bioinformatics.nyu.edu/wordpress/projects/totalrecaller/

Contact: fabian.menges@nyu.edu

Received on March 30, 2011; revised on May 24, 2011; accepted on June 23, 2011

1 INTRODUCTION

Recent advances in sequencing technology continue to introduce novel and diverse high-throughput DNA sequencing platforms, such as Illumina . These innovations, promise to revolutionize biological, biomedical and translational research by rapidly transferring genomics breakthroughs from the byte banks to bedside. For any large-scale genomic sequencing effort aiming to fulfill these goals, an important and rate-limiting initial step involves *Base-Calling*, which is now assuming an even more critical role, especially for these new (next- and next-next-generation) sequencing platforms. When the source genome references are available, the sequence-reads obtained by the base-caller is next subjected to alignment (followed by variant-caller) procedures successively—although, in principle, all three steps could be carried out in one integrated step while promising improvements in accuracy, computational complexity, data storage and transmission.

Base-calling takes the vector analog time series of signals generated by the sequencing machines as input, and produces a base-by-base digitized estimate of the underlying DNA sequence that is most likely to have given rise to those signals. Although the next-generation sequencing technologies listed above have reduced the cost and increased the throughput, these platforms pose new challenges for base-calling, as their technology is based on either relatively small number of unsynchronized molecules (e.g. Illumina, 454 Life Science, Ion Torrent, SOLiD) or single molecules with weak signal (e.g. Pacific Biosciences, Oxford Nanopore, Helicos, Life Technologies). As a consequence, these platforms are not only error prone, they also corrupt signals in the data by non-stationary errors and generate much shorter reads than what is needed for both proper alignment and sequence assembly, as well as what used to be routinely possible with the traditional Sanger sequencers.

Motivated by such challenges, novel base-calling frameworks have been proposed to deal with many unknown sources of noise in these data. These methods have already demonstrated that considerable improvement both in quality and read length is possible through sophisticated signal processing methods such as *statistical learning* (Kao *et al.*, 2009), *supervised learning* and *support vector machine* (SVM) (Erlich *et al.*, 2008; Kircher *et al.*, 2009), and *model-based clustering* and *information theory* (Rougemont *et al.*, 2008).

However, in doing so, these methods also expose *Base-Calling* to several other limitations. (i) Over-fitting: parametric models (e.g. Alta-Cyclic, Ibis, BayesCall, Rolexa) are likely to suffer from over-fitting to the in-sample data and thus are unlikely to be very robust in dealing with varying out-of-sample datasets, even from the same sequencing platforms. (ii) Computational cost: in a preprocessing step, all base-callers must learn the error model in the training data in order to build a classifier that then corrects the errors in the signal. This preprocessing step can be very time consuming (as in the case of BayesCall, Alta-Cyclic and Ibis) and may require a cluster computer facility (as in the case of Alta-Cyclic), thus preventing them from real-time base-calling, as would be needed

*To whom correspondence should be addressed.

in many clinical applications. (iii) Training phase: since a subclass of these base-callers (e.g. Alta-Cyclic, Ibis) needs a training phase using a library of 'correct' reads, they require both a secondary base-caller and a sequence aligner to get started with the training library. On the contrary, this concern has been somewhat alleviated in the newer base-callers (e.g. Bustard, BayesCall), which estimate their parameters solely from intensity files. (iv) Technology dependent: many base-callers (e.g. Alta-Cyclic, Ibis and BayesCall) use a detailed parametric model to describe the signal distortion as a function of successive cycles. Such models require and hard-wire specific knowledge of the underlying sequencing technology into the algorithm, thus making it harder to customize the base-caller to support other platforms.

Currently, a major application of the next-generation sequencing technologies is in *re-sequencing* (e.g. DNASeq: DNA sequencing with known reference source genome; RNASeq: mRNA sequencing with known gene isoforms; CHIPSeq: chromatin immunoprecipitation sequencing with known binding sites). Despite the obvious centrality of alignment in these applications, traditional base-callers have avoided performing alignment until the end of the base-calling process; in fact, the typical pipeline for a re-sequencing process traditionally consists of two sequential steps:

(1) *Base-calling*: each single base of the read is called according to the intensity signal and error profiles.

(2) *Alignment*: sequence reads are aligned to a reference genome.

Because the base-calling process is error prone, and because correct alignment to the reference genome is non-trivial, high coverage is required in order to reduce the errors in re-sequencing and recover the true full DNA sequence.

To perform the alignment to the reference genome, the standard Smith–Waterman (Smith and Waterman, 1981) approach (or some variations of it) may seem appropriate. Note that the Smith–Waterman was developed in the context of homology analysis, is based on dynamic programming and thus requires its score function for evaluating the alignment quality to satisfy 'the principle of optimality (Bellman *et al.*, 1959; Dreyfus, 2002)'. This algorithm [as well as others in this framework starting with Needleman–Wunsch, (Needleman and Wunsch, 1970)] is based on certain Markovian/stationarity assumptions that give rise to additive and affine score functions, which, while fine in the context of evolutionary analysis, are not necessarily valid in the alignment applications for current and future sequencing technologies. Even worse, Smith–Waterman or Needleman–Wunsch algorithms are considered too impractical computationally, and are only used for local alignment after few plausible initial 'seed' locations are determined using BLAST- or BWT-like heuristic aligners (thus adding to the false negatives).

Motivated by the limitations of current base-callers and the challenges of re-sequencing, we have designed a new base-caller, TotalReCaller, which attempts to ameliorate the problems discussed above and significantly improves the quality of reads by injecting knowledge of the reference genome into the base-calling step. In this article, we present a framework (with its theoretical underpinnings) that replaces the typical sequential re-sequencing protocol into a combined pipeline that has the ability to concurrently perform base-calling, alignment and SNP detection. It is important to

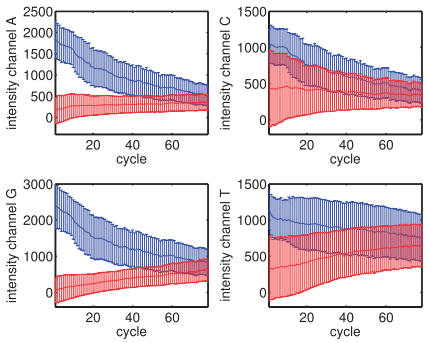

Fig. 1. The statistics for high and low intensity levels depicted with their means and SDs for four channels—one for each base $B \in \{A, C, G, T\}$, shown clockwise. A high intensity level [blue] (with a value above a threshold) in one of the channels indicates that this base should be called at a given cycle. A low intensity level [red] (below threshold) in a channel means that this base should not be called at a given cycle. In a 'good' set of intensities, it is expected that one channel is higher than a threshold, while all other three are lower than it. The panels depict that in later cycles the low and high intensities become increasingly indistinguishable, which causes erroneous base-calling for distal positions.

emphasize that, since base-calling and alignment in TotalReCaller are performed concurrently with the help of a reference genome, the generated reads could be biased toward the available genome. For this purpose, the trade-off analysis between SNP's sensitivity and specificity is also addressed in the article. While, for illustrative purposes, this article specifically addresses base-calling for the Illumina platform, our method is, in principle, applicable to any other sequencing technology. Adaptation to a different sequencing platform only requires redesigned score functions encapsulating error correction and alignment—and, thus, accommodating the different features and error profiles of the variant system in a technology-agnostic manner.

1.1 Illumina sequencing pipeline

As the Illumina sequencing pipeline has been described elsewhere in great details [see for instance Metzker (2005), Bentley (2006)], we report here, for convenience, only a minimal description—essentially what is needed in this article to understand the source of errors in the data and how it is handled by TotalReCaller. The Illumina pipeline, generating raw signals prior to base-calling and alignment, consists out of five steps in which: (i) DNA sample is prepared; (ii) pieces of DNA in the sample are randomly fragmented and placed on a flow cell; (iii) fragments are amplified into clusters. Each cluster consists out of $\sim$1000 identical strands; (iv) fluorescent markers, lasers and CCD sensors are used to read the clusters base by base (so called cycles), resulting in four images for each cycle; and finally (v) the images are analyzed and a single analog intensity value is determined for each image. Base-calling is the process of determining the correct sequence of bases $\sigma \in (A+C+G+T)^*$ from four sequences of intensities, one for each base $B \in \Sigma = \{A, C, G, T\}$. While there are many challenges faced by the base-callers attempting to correctly interpret Illumina intensities data, the major one is due to relatively low signal-to-noise ratio (SNR) of the output, which worsens precipitously with the cycle numbers—ascribed primarily to *polymerase desynchronization* and its low *processivity*. Figure 1

highlights the magnitude of this problem by showing the intensity for each channel averaged over a large number ($\sim 5 \times 10^5$) of 78 bp sequence reads for the bacteriophage *phiX* genome. As shown in the figure, calling the bases *A* and *G* from their intensity channels does not pose a severe problem, at least for the first 40 cycles. In contrast, channels *C* and *T* appear hopelessly disrupted within the first few cycles.

Source of errors in Illumina raw sequencing data: as reported previously in Erlich *et al.* (2008), there are four dominant sources of noise affecting the intensities generated by Illumina:

(1) *Crosstalk:* the intensity channels are not independent. This inter-dependence is due to the fact that the fluorescent markers for A, C and G, T emit photons with similar wavelengths and get excited by the same lasers, and because fluorescent markers from one cycle can only be chemically partially removed ('washed') before the cycles for the next nucleotides (all performed in the same flow cell).

(2) *Fading:* with successive cycles, the absolute intensity of light emitted from the cluster of DNA strands diminishes because fluorescent markers are only able to bind to fewer and fewer strands within the clusters.

(3) *Lagging (Phasing):* some strands in the clusters start to lag behind the population, as in each cycle some of the polymerases fail to operate synchronously, but then rejoin the other strands in subsequent cycles, whence producing ambiguous intensity values. Eventually, the correct channel gets obscured by the other wrong channel intensities, leading to wrong base-calls.

(4) *Leading (Pre-phasing):* some strands in the clusters start to lead ahead of the population, which also causes ambiguous and incorrect channel intensity values in a fashion analogous to lagging.

These noise factors dominate and affect the signal differently in different cycles: in the first few cycles, cross-talk is the major source of base-call errors; however; in later cycles, fading, lagging and leading prevail. We have observed that lagging often causes many false-positive insertions in the distal extending-end of sequence reads: in later cycles, intensities measured in cycle k more and more reflect what would have been the value in cycle $(k-1)$. This process leads directly to 'base-stuttering', which occurs much more frequently after some threshold value for k, the cycle number. This dynamic can be modeled by a step function appearing randomly but more frequently in later cycles, thus making it extremely difficult to analyze. This effect has important implications for the succeeding alignment step, since many popular short-read sequence aligners cannot align gapped sequence reads [Langmead *et al.* (2009); Li and Durbin (2009)]. We have observed that in our datasets the effects of leading on the SNR are negligible in comparison to the other three causes of error (cross-talk, fading and lagging).

2 RESULTS

Including the base-caller introduced in this article, currently there are six base-callers for Illumina sequencing machines: TOTALRECALLER , Bustard (Illumina), Alta-Cyclic [CSHL, Erlich *et al.* (2008)], BayesCall [UC Berkley, Kao *et al.* (2009)], Ibis [Max Planck, Kircher *et al.* (2009)] and Rolexa [Universit de Lausanne, Rougemont *et al.* (2008)]. In the following, we compare the performance of these base-callers[1] according to the same metrics that have been used in the literature: namely, base-calling error rate, alignment rate and base-calling speed. Since TOTALRECALLER utilizes a reference genome for base-calling, base-calling statistics from datasets of three different organisms (with reference genomes of varying qualities) were analyzed (Table 1). Furthermore, base-calling results of TOTALRECALLER are presented with different weights on reference alignment with respect to intensity information. Finally, we conclude this section with detailed statistics on TOTALRECALLER's SNP sensitivity and specificity, as they would be the most important in many clinical applications.

Table 1. Datasets

Genome	Genome size	No. of clusters	No. of cycles	Description
phiX	~5.4 kB	11803171	78	One lane of a *Genome Analyzer I* run.
E.coli	~4.5 MB	35027442	125	The second pair of a paired-end *Genome Analyzer II* run (one lane).
poplar	~420 MB	31445866	109	The first pair of a paired-end *Genome Analyzer II* run (one lane).

Since TOTALRECALLER injects knowledge from a reference genome into the base-calling process, base-calling becomes dependent on the genome structure. Therefore, we evaluated TOTALRECALLER performance on three different datasets: phiX, *E.coli* and poplar.

The base-calling error rate: This measures the quality per cycle (base pair position) of the sequence reads produced by a given base-caller (Fig. 2). In order to generate error rates based on the same set of reads for each of the base-callers, we aligned all Bustard reads to the respective reference genome (Table 1) in order to create a set of 'correct reads'. We then perform a base-by-base comparison between this set of 'correct reads' and the sequence reads created by each of the base-callers, resulting in an error rate for each position (cycle) in a sequence read. Since we used the output of Bustard to create the set of correct reads, we introduced a bias, favoring Bustard. For the small genomes, *phiX* and *E.coli*, we used the aligner BLAT (Kent, 2002), which allows accurate, gapped alignment to create the set of 'correct reads'. For the poplar dataset, we used Bowtie (Langmead *et al.*, 2009) to create the set of 'correct reads'. We had to use Bowtie instead of BLAT in order to properly handle the current draft of the poplar (Tuskan *et al.*, 2006) genome, which is of relatively lower quality in comparison to *E.coli* and *phiX*, e.g. poplar consists of many contigs (out of 2518) that have not been yet phased to a scaffold (http://www.phytozome.net/poplar). The low quality, in conjunction with the length and complex structure of the poplar genome, results in an unusually large number of false positive alignments, which, when analyzed by a sensitive aligner, makes it

[1]As we lacked the required hard- and software, we were unable to compare against Alta-Cyclic.

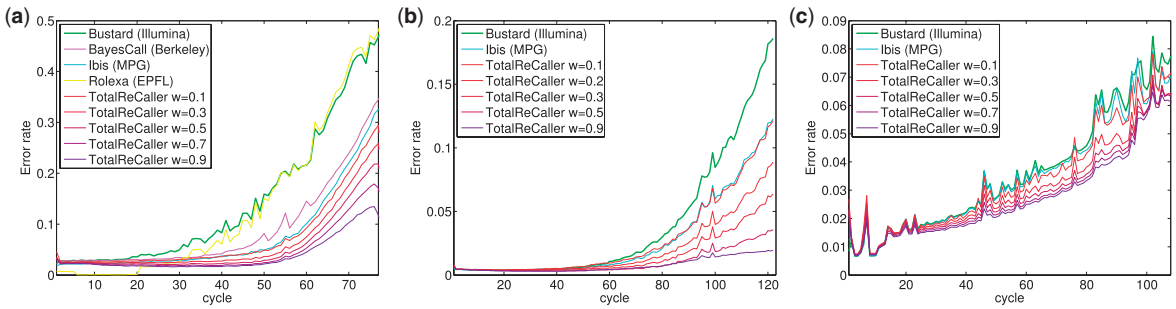

Fig. 2. Error rates: the sequence read error rates per cycle are presented for each of the three datasets (Table 1). In order to compute the error rates, the sequence reads, generated by each of the base-callers, were compared base-by-base (cycle-by-cycle) to a corresponding 'correct read'. As shown, sequence reads generated by TOTALRECALLER have a significantly lower base-calling error rate in comparison to all other base-callers. Furthermore, the error rate can be controlled by choosing an alignment weight w_{align}. Note also that reads produced by the GAIIx Illumina machines are in general of a higher quality than those generated by the older GAI machines. The fluctuations in the poplar error rates are primarily due to the poor quality of the poplar reference genome, which led to a limited set of 'correct reads'. It was not possible to present statistics for BayesCall and Rolex for *E.coli* and *poplar*, since these base-callers do not support the newer Illumina file formats (RTA pipeline). (**a**) *PhiX*; (**b**) *E.coli*; (**c**) *Poplar*.

impossible to create a valid set of 'correct reads'. Since Bowtie and other suffix tree-based algorithms are generally less sensitive than BLAST-like (Altschul *et al.*, 1997) alignment algorithms (e.g. they do not allow gapped alignment), they produce fewer but significantly more accurate sets of 'correct reads', especially in the case of a 'bad' reference genome. The base-calling error rates based on the reads produced by the base-callers and the set of 'correct reads' can be found in Figure 2.

It is shown that sequence reads generated by TOTALRECALLER have a significantly lower base-calling error rate in comparison to all other base-callers. Furthermore, the error rate can be controlled by choosing an alignment weight w_{align}. However, it is necessary to understand that increasing the alignment weight w_{align} does have negative influences which are discussed together with SNP sensitivity and specificity. Based on the previous discussion, it is safe to conclude that the error rates for the poplar dataset may be used only for a qualitative (and not a quantitative) comparison.

The alignment rate (or mapping rate): this describes how many reads, produced by a specific base-caller, can be aligned back to the source reference genome. This rate provides an important metric, because it quantifies how many of the estimated reads possess good enough quality to permit high-level of genome analysis (such as SNP detection). Of course, similar to the base-calling error rate, the alignment rate depends to a large extent on the specific sequence alignment tool that is used. In Table 2, we present the alignment rates for the sequence reads produced by the different base-callers for each of the three datasets. The reads were aligned using Bowtie with conservative parameters (low sensitivity).

It is shown that for all three datasets, a bigger percentage of the sequence reads can be aligned back to the reference if TOTALRECALLER's strategy is used for base-calling. Similarly to the base-calling error rate, a higher alignment weight w_{align} for TOTALRECALLER directly relates to a higher alignment rate.

Base-calling speed: this describes the time needed to perform base-calling for a given dataset. For most base-callers, the total time to perform base-calling can be divided into a training phase and the actual base-calling. During the training phase, a given base-caller computes the parameters for its underlying error models. Similar to Ibis and Rolexa, TOTALRECALLER depends on a set of correct reads to compute its error model parameters. These reads are generated by aligning Bustard reads with Bowtie on a reference genome. In addition, TOTALRECALLER requires the construction of the Burrows–Wheeler transform of the reference genome, which is then used by the base-by-base aligner during base-calling. Table 2 gives a comparison of the base-calling speed of the different base-callers.

It is shown that in comparison to Ibis, Rolexa and BayesCall, TOTALRECALLER uses a shorter training phase. The relatively long base-calling time for TOTALRECALLER can be accounted by the time TOTALRECALLER implicitly spends on genome alignment while base-calling.

SNP specificity and sensitivity: aside from TOTALRECALLER's relative performance advantage in terms of error and alignment rates, it may be questioned whether TOTALRECALLER's bias due to reference-based Bayesian prior is the source of this advantage, and could affect (perhaps adversely) its single nucleotide polymorphism (SNP) sensitivity and specificity. Specifically, since TOTALRECALLER injects knowledge from a reference genome into the base-calling process, it is possible that sequence reads at true SNP positions (containing information from positions where the reference genome differs from the genome that is sequenced) are called incorrectly. Thus, it is vital to examine what happens when a sequence read is called if that sequence contains one or more SNPs with respect to the reference genome. In order to assess this bias toward the reference genome, particularly with respect to reference independent base-callers, we collected the statistics described below and presented the results graphically (Fig. 3). We defined two such important statistics: *Specificity*, SPC_k (also known as *true negative rate*, TNR) and *Sensitivity*, SNS_k (also known as *true positive rate*, TPR) for each cycle (BP position) k.

$$\text{SPC}_k = \frac{\text{True negatives}_k}{\text{True negatives}_k + \text{False positives}_k}, \text{ and}$$

$$\text{SNS}_k = \frac{\text{True positives}_k}{\text{True positives}_k + \text{False negatives}_k}$$

These statistics are based on artificially SNP-inserted reference genomes. On average, we inserted one SNP every n bases randomly

Table 2. Speed and alignment comparison

Genome	Base-caller	t_{training} (h)	t_{calling}	Alignment rate (%)
phiX	Bustard	–	–	15.29
	Ibis	∼8	∼2	31.80
	BayesCall	∼50	∼32	32.66
	Rolexa	∼2	∼90	11.40
	TotalReCaller($w=0.1$)	∼1	∼17	40.50
	TotalReCaller($w=0.3$)			46.85
	TotalReCaller($w=0.5$)			51.56
	TotalReCaller($w=0.7$)			56.59
	TotalReCaller($w=0.9$)			62.89
E.coli	Bustard	–	–	28.70
	Ibis	∼20	∼10	36.31
	TotalReCaller($w=0.1$)	∼1	∼28	46.45
	TotalReCaller($w=0.2$)			55.77
	TotalReCaller($w=0.3$)			64.37
	TotalReCaller($w=0.5$)			77.19
	TotalReCaller($w=0.9$)			87.47
poplar	Bustard	–	–	25.55
	Ibis	∼16	∼9	25.97
	TotalReCaller($w=0.1$)	∼1.5	∼23	25.60
	TotalReCaller($w=0.3$)			27.74
	TotalReCaller($w=0.5$)			29.83
	TotalReCaller($w=0.7$)			31.84
	TotalReCaller($w=0.9$)			33.62

In the third column, 't_{training}' tabulates the approximate duration of the training phase, the parameter estimation, for each of the base-callers. In the fourth column, 't_{calling}' tabulates the duration of the actual base-calling. In the last column, 'alignment rate' shows the percentage of how many of the reads, called by a specific base-caller, could be aligned back to the reference genome using Bowtie (Langmead *et al.*, 2009). Consider the *E.coli* dataset as an example: TotalReCaller, with an alignment weight of $w_{\text{align}} = 0.5$ requires a training phase of 1.5 h, calls 35,027,442 reads, each 125 bp long, in 28 h. This corresponds to 43bp/ms. Out of these $\sim 3.5 \times 10^7$ reads, 77.19% could be aligned back to the *E.coli* reference genome. Base-calling was performed on the datasets presented in Table 1 utilizing a single CPU thread. No precise times are given since they vary depending on runtime parameters.

into each of the reference genomes (Table 1), where n was chosen to be equal to the number of cycles available for the given intensity data. Then the SNP-inserted genome was used as a reference for TotalReCaller. Based on this data, we were able to define a true positive SNP count as the base pair positions in sequence reads, which were called correctly even though the SNP-inserted reference stated otherwise. False positives are positions in the sequence read that match with neither the correct nor the SNP-inserted reference. True negatives are positions in the sequence read that match both references. Finally, false negatives are positions in the sequence reads that match the SNP-inserted reference only. Although all other base-callers ignore all side-information, e.g. information in a reference genome, these same statistics can be computed for all of them for comparison purposes. For those base-callers, sensitivity and specificity depend only on their raw error rates. In Figure 3, we present TotalReCaller's SNP-specificity and sensitivity for different alignment weights w_{align}.

It is shown that TotalReCaller's specificity SPC is higher in comparison to all other base-callers for each of the presented alignment weights. TotalReCaller's sensitivity with low alignment weight w_{align} is as high or higher compared to other base-callers in all cases, with one exception—TotalReCaller's sensitivity for a low alignment weight is *surpassed by Ibis and Bustard for the* E.coli *dataset*. Increasing TotalReCaller's alignment weight increases the specificity and reduces the sensitivity. Considering the significantly higher alignment rate of TotalReCaller (Table 2), the loss of sensitivity from increasing w_{align} is at least partially compensated for by the significantly higher alignment rate of TotalReCaller.

The performance of TotalReCaller leads to the conclusion that, by using a Bayesian approach that relies on dynamically creating a reference-based prior, it is possible to significantly lower the error rates of Illumina short reads for both small and large genomes, independent of the sequencing technology used (GAI or GAIIx). This performance improvement leads directly to higher alignment rates, which in turn implies that more reads from a sequencing run can be used for high-level analysis. The SNP sensitivity and specificity evaluation show that lower error rates and higher alignment rates come at a cost: SNPs can get lost within a single read. However, relating SNP sensitivity and alignment rates it is clear that TotalReCaller represents a significant improvement in the field of base-calling.

METHODS

TotalReCaller combines the knowledge from sequencers' raw intensity data with information from a reference genome. In other words, it generates the most plausible n-base string (out of 4^n possibilities) that is most likely to have generated the channel intensity data, and also most likely to have originated at some location on the reference genome. The main innovation of TotalReCaller is to tame the worst-case exponential complexity of the implementation by using a branch-and-bound strategy. Specifically, this strategy is used to concurrently extend multiple high-quality reads that are immediately validated not only by the intensity signals, but also by the likely alignments to a reference genome (thus the genome providing a weak prior to a Bayesian inference). This scheme builds on a rigorously defined Bayesian score function that accounts for both, thus resulting in a single score quantifying the quality of a given sequence read. In order to execute this task, TotalReCaller implements four different components that are described in detail in the following sections: (i) linear error model; (ii) base-by-base sequence alignment; (iii) branch-and-bound read extension; and finally (iv) score function.

2.1 Linear error model and filter

We devise a simple linear model to correct errors resulting from cross-talk, fading and cycle-dependent synchronous-lagging. The model is based on a cycle-dependent transition matrix (thus dynamic) in order to filter the raw intensity channels.

We first derive a linear-algebraic model for cross-talk and fading, and then extend it to include lagging. Let $\boldsymbol{I}_k = (I_A^k I_C^k I_G^k I_T^k)^\top$ be the vector of the four raw intensity channels. In order to model cross-talk in cycle $k \in \mathbb{N}$, we introduce the cross-talk matrix $\boldsymbol{A}_k \in \mathbb{R}^{4\times 4}$ and the cross-talk-free channels $\boldsymbol{X}_k = (X_A^k X_C^k X_G^k X_T^k)^\top \in \mathbb{R}^4$. We model their relationship with the following formula:

$$\boldsymbol{I}_k = \boldsymbol{A}_k \cdot \boldsymbol{X}_k. \tag{1}$$

Since a separate cross-talk matrix is computed for every cycle k, we implicitly normalize the intensities, thus additionally accounting for fading.

Lagging is then modeled by introducing a dependency between the current cycle and the previous cycle, resulting in:

$$\begin{pmatrix} \boldsymbol{I}_{k-1} \\ \boldsymbol{I}_k \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_{k-1} & 0 \\ \boldsymbol{\Upsilon}_k & \boldsymbol{A}_k \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{X}_{k-1} \\ \boldsymbol{X}_k \end{pmatrix} \tag{2}$$

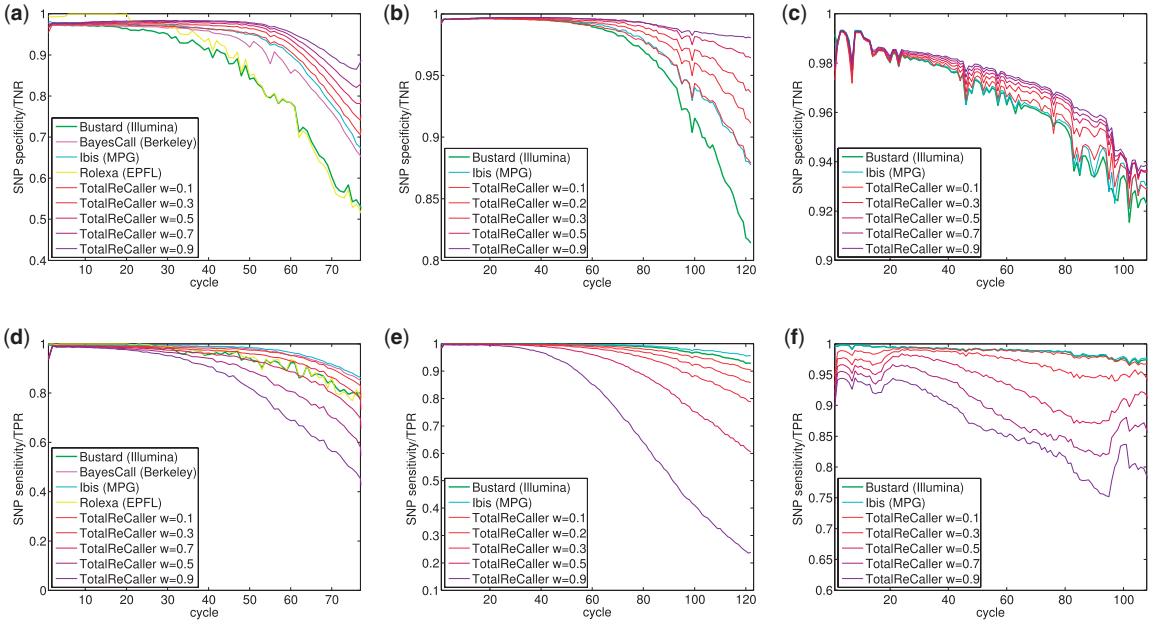

Fig. 3. SNP specificity (SPC) and sensitivity (SNS): the figures above show the effect of the alignment on base-calling, as the weights w_{align} are varied. SNP specificity (SPC) measures the rate at which a difference between a read and its reference represents a SNP and not a base-calling error. SNP sensitivity (SNS) measures the rate of called SNPs with respect to all SNPs that should be called. (**a**) *phiX*-SPC; (**b**) *E.coli*-SPC; (**c**) *poplar*-SPC; (**d**) *phiX*-SNS; (**e**) *E.coli*-SNS; (**f**) *poplar*-SNS.

$$\Rightarrow \begin{pmatrix} \boldsymbol{X}_{k-1} \\ \boldsymbol{X}_k \end{pmatrix} = \underbrace{\begin{pmatrix} \boldsymbol{A}_{k-1} & 0 \\ \boldsymbol{\Upsilon}_k & \boldsymbol{A}_k \end{pmatrix}^{-1}}_{\boldsymbol{G}_k} \cdot \begin{pmatrix} \boldsymbol{I}_{k-1} \\ \boldsymbol{I}_k \end{pmatrix}, \tag{3}$$

where $\boldsymbol{\Upsilon}_k \in \mathbb{R}^{4\times 4}$ describes the coupling between $\boldsymbol{I}_k$ and $\boldsymbol{I}_{k-1}$. A matrix inversion results in a simple transition matrix $\boldsymbol{G}_k$, which is then used to filter the raw intensity channels. The elements of the matrices $\boldsymbol{\Upsilon}_k$ and $\boldsymbol{A}_k$ are obtained by statistical analysis of the intensities, using a library of correct reads similar to the training sets used for the parameter estimation of the support vector machines in Alta-Cyclic and Ibis. However, notice that for TotalReCaller this is not a computationally expensive task since it only solves a simple linear system. After applying the filter to the set of raw intensities (Fig. 1), we were able to significantly improve the quality of the intensity channels, as shown in Figure 4.

The error model and filter can easily be extended to include leading (pre-phasing). Our preliminary results showed that the improvement in quality is only minor compared to the increase in computational cost, so we decided to keep the model simpler and excluded modeling leading.

With the intensity channels suitably filtered, we needed a metric to compare the intensity channels with one another. For this purpose, we focused on the conditional probabilities $P_k(X_B \,|\, B)$ and $P_k(X_B \,|\, \neg B)$ with $B \in \{A, C, G, T\}$. $P_k(X_B \,|\, B)$ denotes the conditional probability of the filtered intensity X_B of channel B, given that the correct base to call is base B, whereas $P_k(X_B \,|\, \neg B)$ denotes the conditional probability of the filtered intensity X_B, given that the correct base to call is not base B. Since the filtered channels X_B are independent of each other, we can approximate these probabilities assuming that they are normally distributed (with subscript k suppressed to avoid clutter),

$$X_B \,|\, B \sim \mathcal{N}(\mu_B, \sigma_B) \quad \text{and} \quad X_B \,|\, \neg B \sim \mathcal{N}(\mu_{\neg B}, \sigma_{\neg B}).$$

That is:

$$P_k(X_B \,|\, B) = \frac{1}{\sqrt{2\pi}\sigma_B} \exp\left(-\frac{(X_B - \mu_B)^2}{2\sigma_B^2}\right), \tag{4}$$

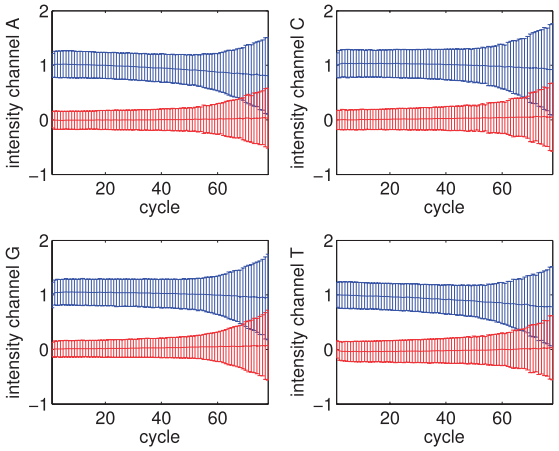

Fig. 4. Filtered intensity channels and separation. Cross-talk and lagging are corrected using a linear filter, developed here. The high and low intensity levels are now cleanly separated for the first 60 cycles.

$$P_k(X_B \,|\, \neg B) = \frac{1}{\sqrt{2\pi}\sigma_{\neg B}} \exp\left(-\frac{(X_B - \mu_{\neg B})^2}{2\sigma_{\neg B}^2}\right). \tag{5}$$

The means together with their SDs have already been presented in Figure 4, in order to show the workings of the linear error model and filter.

2.2 Base-by-base sequence alignment

The key idea of TotalReCaller is to perform alignment while the sequence is being base-called. The partially generated sequences, which are grown one base at a time, must be aligned back to the reference genome. To account for

Table 3. Example: probabilities from FM search for each base preceded by 'ACGAC'

Sequence	Frequency f_B	$P(B)$	$P(\neg B)$
ACGAC**A**	100	0.10	0.90
ACGAC**C**	20	0.02	0.98
ACGAC**G**	500	0.50	0.50
ACGAC**T**	380	0.38	0.62

this computationally intensive task, we designed an efficient base-by-base aligner that is based on a suffix tree search algorithm. Inspired by the many Burrows–Wheeler based short read sequence aligners [Bowtie (Langmead *et al.*, 2009); SOAP2 (Li *et al.*, 2009); BWA (Li and Durbin, 2009)], we constructed our base-by-base aligner essentially on the same principles, specifically the Ferrangina–Manzini (FM) search algorithm (Ferragina and Manzini, 2000) and the Burrows–Wheeler transformation (BWT) (Burrows and Wheeler, 1994).

Ferrangina and Manzini showed how the suffix tree of a reference genome can be accessed through its BWT, which does not require more memory than the reference genome itself. Thus, searching for a (partial) sequence read in a BWT reference can be performed highly efficiently. In addition, not only information about the existence but also the number of occurrences in the given reference of the (partial) sequence read can also be computed concurrently. For more details about the FM search algorithm, see the related publication (Ferragina and Manzini, 2000).

Note that, in general, sequence aligners are usually only interested in whether and where a sequence read is located in a given reference. In contrast, building solely on the alignment information for base-calling, it suffices for us to retain only the frequency of a partial sequence read. Consider the example that the (partial) sequence 'ACGAC' is contained in a reference 1000 times. We can use the FM search to count how often the sequences 'ACGAC**B**' with $B \in \{A, C, G, T\}$ are contained in the reference, from which we can then compute the probability $P_k(B)$, that the next base (at cycle k) in the sequence is B:

$$P_k(B) = \frac{f_B}{f_A + f_C + f_G + f_T} \quad \& \quad P_k(\neg B) = 1 - P(B) \tag{6}$$

Table 3 shows a complete example of how the base probabilities are computed using the FM search. Now that we have introduced the intensity filter and base-by-base alignment components, what remains to be shown (in the next section) is how to combine them in order to score and prune the candidate solutions.

2.3 Branch-and-bound read extension

Recall that, for efficiency, TotalReCaller uses a banch-and-bound strategy to combine intensity and alignment information by sequentially constructing a tree of hypothetical sequences, N_k. In order to reduce the computational complexity, the tree is only partially constructed and repeatedly evaluated. For the sake of clarity of exposition, we first describe the algorithm as if the full tree had been constructed, but then present a concrete and complete implementation of the algorithm, in which unpromising solutions are carefully, but rapidly, pruned.

In order to be able to recover the best sequence out of the N_k possible sequences, every node is scored according to a Bayesian score function immediately upon creation. This score function combines terms for intensity and alignment information and is described in the next section. The most likely estimate for the correct sequence read(s) is therefore obtained by simply choosing the node with the (globally) highest score.

Of course, this tree (without any pruning) grows exponentially with increasing number of cycles: at cycle k, $|N_k| = 4^k$ sequence reads must be evaluated. Moreover, since asynchronous lagging causes wrong insertions into the sequence read, we need to consider deletions as a fifth child, which means that a tree $\overline{N}_k$, with $|\overline{N}_k| = 5^k$ sequence reads must be created and evaluated. Since TotalReCaller dynamically prunes unpromising sequences based on the evaluation of the score function in a branch-and-bound scheme (Land and Doig, 1960; Lawler and Wood, 1966), the worst-case complexity is rarely encountered in practice (and can be further controlled by beam-search). Note that the special situations in which the exponential worst-case behavior would be exhibited occur when the sequencer is extremely noisy and/or when the reference is incorrect (or highly error-ridden), thus producing exponentially more plausible hypothetical sequence reads; a judicious solution in these cases would then involve terminating the sequence read at a smaller read length or rejecting it outright. The algorithm can now be described as a sequence of three consecutive steps that are repeated once for each cycle:

(1) Branching: for each sequence in the solution space N_{k-1} , all four possible successor sequences are generated, resulting in the solution space N_k. Note that at this point $N_{k-1} \subset N_k$.

(2) *Bounding*: each sequence in N_k is evaluated according to a score function f_{score}. The score s_k of a specific sequence at cycle k is updated as follows:

$$s_k = s_{k-1} + f_{\text{score}} \tag{7}$$

(3) *Pruning*: all but the best (highest score) $l \in \mathbb{N}$ sequences are pruned, thus reducing the size of N_k to $|N_k| = l$.

Note that, by reducing the computational complexity through bounding the solution space, we are not guaranteed anymore to generate the optimal solution. However, in practice the accuracy of the algorithm's outputs is only slightly affected. Wherever necessary, the computational cost can be traded off for higher accuracy by setting a parameter that controls the beam-width of a beam-search.

2.4 Score function

To complete the description of our base-caller, we need to define the score function, f_{score}, used to evaluate the quality of the candidate sequences in the tree. Using Bayes' theorem, we estimate the probability P_k that a specific base $B \in \{A, C, G, T\}$ is indeed the correct base to call at cycle k, given the filtered intensity vector $\boldsymbol{X}_k$, is:

$$P_k(B \mid \boldsymbol{X}_k) = \frac{P_k(\boldsymbol{X}_k \mid B) P_k(B)}{P_k(\boldsymbol{X}_k)} \tag{8}$$

$$= \frac{1}{1 + \frac{P_k(\boldsymbol{X}_k \mid \neg B)}{P_k(\boldsymbol{X}_k \mid B)} \cdot \frac{P_k(\neg B)}{P_k(B)}} \tag{9}$$

Since for our purposes it is sufficient to have a quantitative measurement (not a probability) to compare all different solution to one another, we introduce a simplified score function f_{score} which is based on $P_k(B \mid \boldsymbol{X}_k)$:

$$f_{\text{score}} = \underbrace{\log\left(\frac{P_k(\boldsymbol{X}_k \mid B)}{P_k(\boldsymbol{X}_k \mid \neg B)}\right)}_{\text{intensities [Equation (4)]}} + w_{\text{align}} \cdot \underbrace{\log\left(\frac{P_k(B)}{P_k(\neg B)}\right)}_{\text{alignment [Equation (6)]}} \tag{10}$$

Therefore, the score function consists of two parts, both of which can be computed independently according to the sections discussing the intensity filter and the base-by-base alignment algorithm. The weight $w_{\text{align}} \in [0, 1]$ permits a user-defined control over the impact of the alignment on the overall score, thus enabling the user to adjust the Bayesian bias, appropriate for a particular application.

3 DISCUSSION

This article introduces a new base-caller, TotalReCaller, which opens new avenues for reducing errors significantly in

short sequencing reads simply by injecting knowledge from a reference source genome through a base-by-base alignment algorithm. Stepping back and re-examining the overall strategies for base-calling, alignment, resequencing, polymorphism-detection, sequence assembly and assembly validation, it appears that there is much to be gained from integrating various steps in genomics analysis that have been traditionally performed sequentially. A Bayesian framework can be developed rigorously for the purpose of such integration, and classical branch-and-bound algorithms can be used to tame the efficiency, which otherwise could make the integration intractable.

In addition to a software implementation of TotalReCaller, we have also implemented and validated a proof-of-concept Field-Programmable-Gate-Array (FPGA) design. Our hardware design is capable of performing all parts of the algorithm, presented here, including: filtering the raw intensity channels, FM search, score function calculation and the branch-and-bound algorithm. Building upon this FPGA design, we wish to next show that re-sequencing can be performed in an embedded environment. Coupled with recent advances in sequencing chemistry (e.g. rapid field-effect-transistor-based assay of pH changes after a base incorporation by a polymerase), this process could pave the way for a real-time SNP-calling machine.

To move further in this direction and to fully exhaust TotalReCaller's potential, much remains to be done. For instance, we will next need to compare TotalReCaller to other sequence aligners. This article primarily focused on TotalReCaller's base-calling qualities, describing the algorithm and comparing it to its peers. As our analysis begins to focus on sequence alignment (e.g. including gapped alignment), we expect to better understand how TotalReCaller handles various structurally complex portions of the genome that often frustrate most suffix tree-based aligners.

As discussed earlier, another direction of research currently focuses on developing a non-parametric approach to handle intensity errors—thus, without having to learn optimal strategies from a calibrating genome (or training data). Currently, TotalReCaller (like all other parametric base-callers) depends on a training set of correct reads and runs the risk of being misled, should the training data be corrupted by some fraction of incorrect reads.

Last but not least it will be important to study the effects bounding onto the the overall performance. The results presented in this article were generated with a static bound of $l=32$. A static bound was chosen in order to allow an efficient low-level hardware implementation. The value $l=32$ was empirically chosen. Future improvements of TotalReCaller's software implementation will include a dynamic bound, which has the potential to improve both accuracy and/or speed.

ACKNOWLEDGEMENT

We would like to thank Drs Christian Haudenschild and Nan Leng of Illumina, Yaniv Erlich of Whitehead Institute and Prof. Alberto Policriti of University of Udine and 'Istituto di Genomica Applicat' (IGA) as well as IGA's staff, for providing raw signals from Illumina sequencing machines that were then analyzed to compile the statistics presented in the article.

Funding: Grant from NSF CDI program and Abraxis BioScience, LLC.

Conflict of Interest: none declared.

REFERENCES

Altschul,S.F. *et al.* (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.*, **25**, 3389–3402.

Bellman,R. and Dreyfus,S. (1959) Functional approximations and dynamic programming. *Mathematical Tables and Other Aids to Computation*, **13**, 247–251.

Bentley,D.R. (2006) Whole-genome re-sequencing. *Curr. Opin. Genet. Dev.*, **16**, 545–552.

Burrows,M. and Wheeler,D. (1994) A block-sorting lossless data compression algorithm. *Technical Report 124*, SRC (digital, Palo Alto).

Dreyfus,S. (2002) Richard Bellman on the birth of dynamic programming. *Operat. Res.*, **50**, 48–51.

Erlich,Y. *et al.* (2008) Alta-Cyclic: a self-optimizing base caller for next-generation sequencing. *Nat. Methods*, **5**, 679–682.

Ferragina,P. and Manzini,G. (2000) Opportunistic data structures with applications. *Annu. Sympos. Found. Comput. Sci.*, **41**, 390–398.

Kao,W. *et al.* (2009) BayesCall: a model-based base-calling algorithm for high-throughput short-read sequencing. *Genome Res.*, **19**, 1884.

Kent,W.J. (2002) BLAT—the BLAST-Like Alignment Tool. *Genome Res.*, **12**, 656–664.

Kircher,M. *et al.* (2009) Improved base calling for the Illumina Genome Analyzer using machine learning strategies. *Genome Biol.*, **10**, R83.

Land,A. and Doig,A. (1960) An automatic method of solving discrete programming problems. *Econometrica* , **28**, 497–520.

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

Lawler,E. and Wood,D. (1966) Branch-and-bound methods: a survey. *Operat. Res.*, **14**, 699–719.

Li,H. and Durbin,R. (2009) Fast and accurate short read alignment with burrows-wheeler transform. *Bioinformatics*, **25**, 1754.

Li,R. *et al.* (2009) SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, **25**, 1966–1967.

Metzker,M.L. (2005) Emerging technologies in DNA sequencing. *Genome Res.*, **15**, 1767–1776.

Needleman,S.B. and Wunsch,C. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, **48**, 443–453.

Rougemont,J. *et al.* (2008) Probabilistic base calling of Solexa sequencing data. *BMC Bioinformatics*, **9**, 431.

Smith,T. and Waterman,M. (1981) Identification of common molecular subsequences. *J. Mol. Biol.*, **147**, 195–197.

Tuskan,G.A. *et al.* (2006) The genome of black cottonwood, Populus trichocarpa (Torr. & Gray). *Science* , **313**, 1596–604.