Bioinformatics, 32, 2016, i529–i537
doi: 10.1093/bioinformatics/btw458
ECCB 2016

OXFORD

Improve homology search sensitivity of PacBio data by correcting frameshifts

Nan Du and Yanni Sun*

Department of Computer Science and Engineering, Michigan State University, East Lansing, MI 48824, USA

*To whom correspondence should be addressed.

Abstract

Motivation: Single-molecule, real-time sequencing (SMRT) developed by Pacific BioSciences produces longer reads than secondary generation sequencing technologies such as Illumina. The long read length enables PacBio sequencing to close gaps in genome assembly, reveal structural variations, and identify gene isoforms with higher accuracy in transcriptomic sequencing. However, PacBio data has high sequencing error rate and most of the errors are insertion or deletion errors. During alignment-based homology search, insertion or deletion errors in genes will cause frameshifts and may only lead to marginal alignment scores and short alignments. As a result, it is hard to distinguish true alignments from random alignments and the ambiguity will incur errors in structural and functional annotation. Existing frameshift correction tools are designed for data with much lower error rate and are not optimized for PacBio data. As an increasing number of groups are using SMRT, there is an urgent need for dedicated homology search tools for PacBio data.
Results: In this work, we introduce Frame-Pro, a profile homology search tool for PacBio reads. Our tool corrects sequencing errors and also outputs the profile alignments of the corrected sequences against characterized protein families. We applied our tool to both simulated and real PacBio data. The results showed that our method enables more sensitive homology search, especially for PacBio data sets of low sequencing coverage. In addition, we can correct more errors when comparing with a popular error correction tool that does not rely on hybrid sequencing.
Availability and Implementation: The source code is freely available at https://sourceforge.net/projects/frame-pro/.
Contact: yannisun@msu.edu

1 Introduction

Single-molecule, real-time sequencing (SMRT) developed by Pacific BioSciences (referred as PacBio sequencing hereafter) produces longer reads than second generation sequencing technologies such as Illumina. The long read length enables PacBio sequencing to close gaps in genome assembly (Conlan *et al.*, 2014; Koren *et al.*, 2013), reveal structural variations (Chaisson *et al.*, 2015), and quantify gene isoforms with higher accuracy (Tilgner *et al.*, 2014) in transcriptomic sequencing. More recently, it has been applied to sequence microbial communities (Tsai *et al.*, 2016) and achieved better assembly quality than using only Illumina reads. Because of the promising results, there is an increasing number of research groups adopting SMRT for their sequencing needs.

When compared with Illumina, the representative second generation sequencing platform, the major disadvantages of PacBio include high sequencing error rate (11–15%), lower throughput, and higher cost per base (Quail *et al.*, 2012; Rasko *et al.*, 2011). Similar to pyrosequencing data, most of the errors are insertion or deletion errors. The high error rate poses challenges for all downstream sequence analysis. In particular, during homology search for genome annotation, sequences are aligned to characterized protein sequences or families. Insertion or deletion errors in genes will cause frameshifts and may only lead to marginal alignment scores and short alignments (Zhang and Sun, 2011). As a result, it is hard to distinguish true alignments from random alignments. The inaccurate homology search results can incur errors in structural and functional annotation.

Different strategies have been proposed or implemented to avoid or correct sequencing errors in PacBio data. There are various PacBio sequencing projects that mainly use circular consensus sequencing (CCS) reads with sufficient sequencing passes. A coverage of 15 passes yields >99% accuracy (Rhoads and Au, 2015). However, CCS reads are much shorter than the continuous long reads of PacBio data. In addition, the amount of CCS reads is much less than all the output of PacBio data. Thus, using only CCS reads

does not take full advantage of the sequencing power and strength of PacBio data.

One popular strategy to handle sequencing errors of PacBio data is based on hybrid sequencing (Koren *et al.*, 2012). As Illumina produces many more accurate but shorter reads, methods are developed to correct errors by aligning short reads to long PacBio reads. Yet, this method needs preparation of at least two sequencing libraries and several types of sequencing runs, which is not cost-effective for many applications.

Unlike hybrid sequencing, there are methods that do not require highly accurate short reads for error correction. One representative method is described in Chin *et al.* (2013) hierarchical genome-assembly process (HGAP), which aligns short sequences to the longest reads of the same sequencing library of PacBio. As the sequencing errors in PacBio reads occur randomly, the inferred consensus sequence from the alignment between the short reads and long reads represent the high-quality sequence. Despite its success, there is still room to improve the error correction performance for the consensus sequence extraction stage in HGAP. In particular, its performance is heavily affected by the coverage of the aligned short sequences against the long seed sequences. The regions with more short sequences aligned have better error correction performance than other regions.

After error correction, corrected PacBio reads can usually achieve more sensitive homology search results compared with the raw data. In particular, when the coverage is high, the corrected reads from HGAP can achieve alignment scores similar to the ground truth. However, in practice, not all PacBio sequencing projects can have sufficient coverage for all regions, transcripts, or genomes. For example, HGAP failed to assemble the data from the arm sample in human skin microbial community (Tsai *et al.*, 2016) because of low coverage of the data set. Figure 4 in our experimental results show that the difference of the alignments' scores, lengths and *E*-values between HGAP's corrected reads and the ground truth is still significant. Thus, there is still a need for homology search tools designed for PacBio data.

In this work, we designed and implemented Frame-Pro, a homology search tool for PacBio reads. The experimental results showed that our tool can significantly improve the homology search sensitivity while also correcting sequencing errors. Our method incorporated two key observations. First, as shown by HGAP, sequencing errors in PacBio are distributed randomly and thus the consensus sequences tend to be closer to ground truth. Our work incorporated this method. Second, we identify frameshifts caused by sequencing errors using characterized protein families as the guidance. Essentially our method corrects errors by maximizing both alignment score against protein families and local coverage score in a constructed alignment graph. Both observations are used together to boost the performance of both homology search and error correction.

The remainder of this article is organized as follows. Section 2 briefly reviews other frameshift error detection tools and their limitations in protein domain classification in PacBio data sets. Section 3 describes the dynamic programming algorithm that incorporates consensus sequence finding and Viterbi algorithm for error correction and sequence alignment. In Section 4, we demonstrate the results of error correction and homology search by applying our tool to simulated and real PacBio data. We also benchmark our tool with HGAP, a successful error correction method without relying on hybrid sequencing. Finally, Section 5 concludes and suggests directions for future work.

2 Related work

2.1 Profile homology search

Homology search is still an important step in sequence-based functional analysis for genomic data. By comparing query sequences against reference sequences or profiles, i.e. a family of homologous reference sequences, functions and structures can be inferred. The representative tools for sequence homology search and profile homology search are BLAST (Altschul *et al.*, 1990) and HMMER (Eddy *et al.*, 2015), respectively. Profile homology search has several advantages over pairwise alignment tools such as BLAST. First, the number of gene families is significantly smaller than the number of sequences, rendering much faster search time. For example, there are only about 13 000 manually curated protein families in Pfam, but these cover nearly 80% of the UniProt Knowledgebase and the coverage is increasing every year as enough information becomes available to form new families (Finn *et al.*, 2016). The newest version of HMMER (Eddy *et al.*, 2015) is more sensitive than BLAST and is about 10% faster. Second, previous work (Durbin *et al.*, 1998) has demonstrated that using family information can improve the sensitivity of a remote protein homology search, which is very important for various sequencing data such as metagenomic data analysis. These data sets may contain species remotely related to ones in the reference database and require sensitive homology search. Thus, in this work, we focus on implementing profile homology search for PacBio data. The method can be extended to pairwise sequence alignment. As HMMER is the most widely used profile alignment tool, we focus on evaluating the alignment performance using HMMER.

The protein domains families used in our experiments are downloaded from Pfam. Other databases such as TIGRFAM (Haft *et al.*, 2003), FIGfams (Meyer *et al.*, 2009), InterProScan (Zdobnov and Apweiler, 2001) and FOAM (Prestat *et al.*, 2014) can be used too as long as profile hidden Markov models (HMMs) can be trained.

2.2 Related work on frameshift correction

Usually, when comparing a DNA sequence with a protein sequence or family, six-frame translations are conducted and one of the reading frame should lead to statistically significant alignment if the query and the reference are homologous. However, frameshifts caused by insertion or deletion errors make the correct translation consist of alternating reading frames. Without knowing the error positions, choosing the correct frames for each fragment between errors is challenging.

A number of programs exist to handle frameshifts through DNA vs. protein sequence alignment. Simple methods such as BLASTX discard sequences that might contain frameshifts rather than trying to fix them. Other tools (Brown *et al.*, 1998; Chang and Lawler., 1994; Gîrdea *et al.*, 2009, 2010; Guan and Uberbacher, 1996; Halperin *et al.*, 1999; Peltola *et al.*, 1986; Pellegrini and Yeates, 1999; Zhang *et al.*, 1997) are available to detect and fix frameshift errors automatically. Besides detecting frameshift in sequence alignment, some programs (Antonov and Borodovsky, 2010; Borodovsky and McIninch, 1993; Kislyuk *et al.*, 2009; Schiex *et al.*, 2003) focus on frameshift detection during gene finding and use *ab initio* methods. These programs are not designed for PacBio data. In addition, they cannot be applied to protein profile homology search.

Alternatively, Genewise (Birney *et al.*, 2004), a widely used DNA vs. protein alignment tool allows comparison of a DNA sequence with a protein family. But it does not consider the sequencing error properties of NGS data. The most relevant works to ours include HMM-frame (Zhang and Sun, 2011), which modified Viterbi algorithm to improve homology search for pyrosequencing data. But it does not have satisfactory performance for PacBio data because the sequencing error properties of pyrosequencing and PacBio are different. Pyrosequencing reads have lower error rate and

most of the errors are located inside homopolymer regions. These properties make error correction easier than PacBio, which have higher error rates and the errors can occur more randomly. FrameBot (Wang *et al.*, 2013) is another relevant work for correcting frameshifts caused by sequencing errors. But it is not designed for profile homology search. And, like other tools, it is only optimized and tested on sequencing data with lower error rate than PacBio.

HGAP (Chin *et al.*, 2013) is another highly relevant work because it contains error correction stage for PacBio data. As discussed in Section 1, its performance is heavily affected by sequencing coverage.

3 Methods

Frame-Pro is designed to improve profile homology search performance for PacBio data. It incorporates consensus-based error correction and a modified Viterbi algorithm for finding optimal alignment. While a standard Viterbi algorithm aligns a single sequence to a HMM, our algorithm aligns an alignment graph to an HMM. In addition, we defined a new scoring system that combines the path score from the alignment graph and the alignment score in the HMM. We first introduce the construction of the alignment graph.

3.1 Generate sequence alignment graph

Following HGAP, we first construct a sequence alignment graph (SAG) from PacBio data. The details of the graph construction can be found in the supplementary material of Chin *et al.* (2013) work. Here we simply summarize the major steps. Reads with a length longer than a chosen cut-off are selected as seed sequences. Other reads are then aligned to the seed sequences by BLASR (Chaisson and Tesler, 2012) for construction of an alignment graph (see top panel of Fig. 1). In each aligned column, different bases are modeled as nodes for the corresponding position in an SAG. Consecutive bases are connected by an edge in the graph. The edge weight represents the number of aligned sequences that go through this edge.

The consensus sequence from the alignment graph represents the most reliable sequence. Figure 1 shows an example SAG and the consensus sequence. Usually, when there are sufficient reads aligning to the seed sequences, the consensus sequence can be reliably used for downstream analysis including conducting homology search. However, when the coverage is not sufficient, the chosen consensus sequence does not show significantly higher score than alternative sequences. Thus, it is difficult to extract a path that is closest to the reference sequence.

Our algorithm is much less sensitive to sequencing coverage as it uses characterized protein families as reference to choose optimal path in the alignment graph. Essentially, it aligns an alignment graph with a profile HMM, which represents a protein family. During the alignment, the algorithm chooses a sequence path in SAG and a state path in the HMM in order to maximize the combined coverage score and alignment score. Below we describe the modified Viterbi algorithm.

3.2 Viterbi algorithm for aligning an alignment graph with a profile HMM

Let π be a state path in a profile HMM Model M. Let π' be a sequence path in an SAG G. Our goal is to search for the optimal path pair (π^*, π'^*) such that $(\pi^*, \pi'^*) = \text{argmax}(\alpha S_M(\pi, \pi') + \beta S_G(\pi'))$, where $S_M(\pi, \pi')$ is the alignment score between a sequence in G and the profile HMM. α and β are the weights of the HMM score and the coverage score in G, respectively. Intuitively this algorithm searches for an optimal alignment between a DNA sequence in the alignment graph G and a profile HMM M by simultaneously considering (i) the probability of a profile HMM alignment, represented by log-odds score $S_M(\pi, \pi')$, and (ii) path weight in G, represented by $S_G(\pi')$. To solve the above equation, we developed the following dynamic programming algorithm, which is an augmented Viterbi algorithm (Durbin *et al.*, 1998). The recursive equations can be extended to forward algorithm and also posterior probability calculation for an HMM.

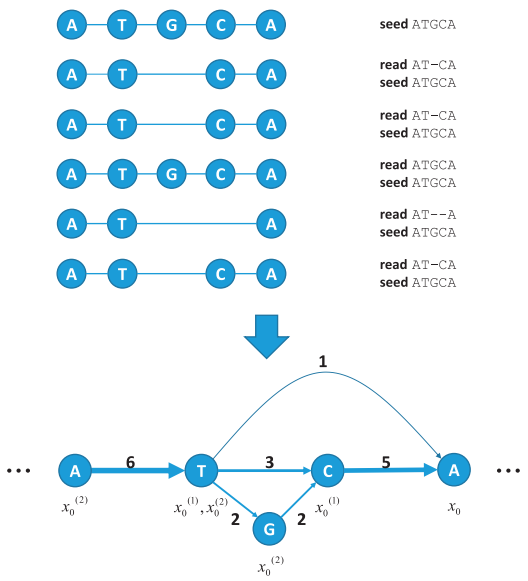

Fig. 1. Build an example SAG from an alignment. Top panel: multiple sequence alignment of six sequences. The top sequence is the seed sequence. Bottom panel: SAG. For node x_0, if we trace back for two more edges, we can identify three codons ending with x_0: TCA, ATA and GCA. Each path has its specific nodes $x_0^{(1)}$ and $x_0^{(2)}$ marked. The consensus path of this part of graph is ATCA

Input: a SAG G generated by alignments between a long DNA seed sequence x_{seed} and multiple shorter sequences, a profile HMM M. Notations of M will be described below.

Output: the error-corrected DNA sequence x_{seed}' and its optimal alignment with M.

Algorithm: we first define notations that will be used in the recursive equations.

- **Notations of the profile HMM model *M*:** The detailed descriptions of a profile HMM model can be found in the literature (Durbin *et al.*, 1998; Eddy, 1998). A profile HMM model M consists of match states M_j, deletion states D_j, and insertion states I_j for each position j, which is the index of a conserved column in a multiple sequence alignment. $a_{s_i s_j}$ is the transition probability from state s_i to s_j. As the HMM is constructed by aligned protein sequences while the graph model is constructed by aligned DNA sequences, we need to translate the sequences in G into amino acids. Here, let $T(x_{i-2}x_{i-1}x_i)$ be the amino acid translated from a codon $x_{i-2}x_{i-1}x_i$. $e_s(T)$ is the emission probability for state s to emit T. When compared with the topology of Plan 7 model used by HMMER (Eddy *et al.*, 2015), one of the major changes we made is that N and C are responsible for emitting all DNA bases that are outside of the protein domain.
- **Notations of the SAG:** Let x_i represent the ith node in the topologically sorted list of the graph. As the alignment graph is constructed in DNA space, x_i also represents a base from the aligned sequences. $x_i^{(k)}$ represents a node, from which to node x_i there exists a path consisting of k edges. According to this definition, when $k = 1$, there is an edge from $x_i^{(1)}$ to x_i. When $k = 2$, a codon

is formed by three bases: $x_i^{(2)}$, $x_i^{(1)}$, and x_i. For example, in Figure 1, both nodes labeled with T and C are $x_0^{(1)}$. $S_G(\pi')$ represents the path score for π', which can be a sequence of any length in G. We will define the path score following the equations.

- **Subproblems and the recursive equations:** For a sequence path π' ending at node x_i in G, and a state path π ending with index j in M, the dynamic programming algorithm intends to maximize the combined path score and alignment score: $\alpha S_G(\pi'_{1..i}) + \beta S_M(\pi'_{1..i}, \pi_{1..j})$. Depending on the ending states, we need to consider multiple cases.

  $V_j^M(x_i)$ is maximum score of aligning a sequence path ending with x_i in G to the HMM up to state M_j, under the constraint that the animo acid translated by the last three bases x_i, $x_i^{(1)}$, and $x_i^{(2)}$ in G is emitted by match state M_j. Note that there can be multiple sequence paths in G ending with x_i. So the last three bases of any such sequence path can be generally represented by $x_i^{(2)}x_i^{(1)}x_i$. And the translated amino acid is thus $T(x_i^{(2)}x_i^{(1)}x_i)$.

  $V_j^I(x_i)$ is the maximum score of aligning a sequence path ending with x_i in G to the HMM up to state V_j, under the constraint that the animo acid translated by the last three bases x_i, $x_i^{(1)}$, and $x_i^{(2)}$ in G is emitted by insertion state I_j.

  $V_j^D(x_i)$ is the maximum score of aligning a sequence path ending with x_i in G to the HMM up to state D_j. $V^N(x_i)$ is the maximum score of aligning a sequence path ending with x_i in G to the HMM up to state N, given x_i being emitted by the special state N. Similarly, $V^C(x_i)$ is the maximum score of aligning a sequence path ending with x_i in G to the HMM up to state C, given x_i being emitted by the special state C.

  For brevity of presentation, S in the following equations represents $S(x_i^{(3)}x_i^{(2)}x_i^{(1)}x_i)$, which is the path score from the possible third base of the preceding codon to the current codon ending with x_i. Note that the path score is not a simple summation of edge weights as in a standard graph. We will define the path score after the recursive equations. T is the abbreviation of $T(x_i^{(2)}x_i^{(1)}x_i)$ in following equations.

$$V_j^M(x_i) = \max\begin{cases} V_{j-1}^M(x_{i^{(3)}}) + \alpha e_{M_j}(T) + \alpha \log a_{M_{j-1},M_j} + \beta S \\ V_{j-1}^I(x_{i^{(3)}}) + \alpha e_{M_j}(T) + \alpha \log a_{I_{j-1},M_j} + \beta S \\ V_{j-1}^D(x_{i^{(3)}}) + \alpha e_{M_j}(T) + \alpha \log a_{D_{j-1},M_j} + \beta S \\ V^N(x_{i^{(3)}}) + \alpha e_{M_j}(T) + \beta S \end{cases}$$

$$V_j^I(x_i) = \max\begin{cases} V_j^M(x_{i^{(3)}}) + \alpha e_{I_j}(T) + \alpha \log a_{M_j,I_j} + \beta S \\ V_j^I(x_{i^{(3)}}) + \alpha e_{I_j}(T) + \alpha \log a_{I_j,I_j} + \beta S \end{cases}$$

$$V_j^D(x_i) = \max\begin{cases} V_{j-1}^M(x_i) + \alpha \log a_{M_{j-1},D_j} \\ V_{j-1}^D(x_i) + \alpha \log a_{D_{j-1},D_j} \end{cases}$$

$$V^N(x_i) = \max\left\{ V^N(x_i^{(1)}) + \beta S_{x_{i^{(1)}}x_i} \right.$$

$$V^C(x_i) = \max\begin{cases} V^C(x_i^{(1)}) + \beta S_{x_{i^{(1)}}x_i} \\ V^M(x_i^{(1)}) + \beta S_{x_{i^{(1)}}x_i} \end{cases}$$

- **Calculate path scores *S*:** we followed the same equation in HGAP paper (Chin *et al.*, 2013) to calculate a path score, which is the sum of the scores of all nodes in the path. In a SAG G, the local coverage for a position in the aligned graph is defined by the number of reads aligned to that position. For example, the local coverage is 5 for all positions in Figure 1 as there are 5 reads (excluding seed sequence) aligned to the seed sequence. For a node, if one of the incoming edge's weight is more than half of the local coverage at that position, the node gets a positive score. Otherwise, we will assign a negative score. The path score is the sum of the scores of nodes in the path. The detailed pseudocode can be found in HGAP.
- **Combine HMM score and path score:** users can choose the weights of the path score in graph G and the HMM score in M. Ideally, α and β should be adjusted based on the local coverage. For high local coverage, we can assign bigger α than β. By default, we use the same weight, which is the chosen parameters for all of the experiments in this work.
- **Running time analysis** The time complexity of our algorithm is $O(\delta|N||M|)$, where $|N|$ is the number of nodes in the G and $|M|$ is the number of states in M. δ is the average number of possible paths that contain a codon ending with a node. Based on our test, for 20× coverage, δ is about 4–6 per node. We also found at high local coverage area, the edge with weight 1 can be ignored as the probability to include that edge in the optimal path is extremely low. Thus, we removed those edges to further speed up the algorithm. After this pruning, δ can be as low as near 1 per node on average.

3.3 Filtration stage for removing unrelevant protein domain families

During homology search, we align the constructed alignment graphs from PacBio data with all characterized domains in Pfam. But for any given species or a community, not all domains are relevant. In order to reduce the search space, we apply a filtration stage to remove domains that are not relevant to the given data. In practice, we apply HMMER to all consensus sequences extracted from the alignment graph with a very loose *E*-value cutoff (1000 is used in all experiments). Only domains that yield at least partial alignments will be used as input to our dynamic programming. Other domains without any hit will be discarded because it is unlikely that they will be true domains in this data set. Each consensus sequence might be aligned to multiple domains, for regions that cannot be aligned to any domain, we also remove them from next stage. Thus, after filtration, the trimmed alignment graph and the corresponding domain will be used as input to our tool. Note that this domain may just incur a very small score and non-significant *E*-value. It will be realigned using our algorithm and the final alignment score will decide whether it is a true domain. In our experiments, we refer to the input as sequence-domain pair. In all of our experiments, true domains found by corrected sequences using suggested cutoffs is always less than the input sequence-domain pairs.

4 Experimental results

Our method is used to improve the sensitivity of profile homology search by correcting insertion or deletion errors in PacBio sequencing data. Thus, we will focus on evaluating the performance of our implementation in both error correction and homology search. We applied Frame-Pro to three datasets: a simulated *Escherichia coli* PacBio RS sequencing dataset, a real *Meiothermus ruber* PacBio RS sequencing dataset, and a Human arm PacBio RSII metagenomic sequencing dataset. The three datasets enable us to evaluate the

performance of error correction for data with different sequencing coverage. As both the genomes and their protein domain annotations of the first two data sets are available at NCBI and Pfam, we are able to quantify the accuracy of error correction and profile HMM search. Specifically, we used BLAST to evaluate its error correction performance by aligning corrected sequences to reference sequence. We also quantified the performance of protein domain annotation by applying HMMER3 to corrected sequences and the reference genomes.

We benchmarked our method with the error correction stage DAG-Con in HGAP (Chin *et al.*, 2013). The error correction in HGAP and our method do not rely on short sequences generated by another platform. And the error correction in HGAP has been extensively tested and has satisfactory performance for sequencing data with reasonable coverage. DAG-Con only outputs corrected sequences. In order to evaluate the performance of profile homology search, we apply HMMER to the corrected sequences of DAG-Con. Although Frame-Pro outputs both corrected sequences and the profile alignments, we rerun the corrected sequences against input domains using HMMER to ensure a fair comparison by the same alignment tool.

For our experiments, all detailed commands, parameters and output can be found along with the source code of Frame-Pro. To achieve a fair comparison on data of low coverage, we set the coverage threshold of DAG-Con as 1 to make sure the outputs are comparable.

4.1 Simulated *E.coli* sequences

In order to evaluate the accuracy of Frame-Pro in detecting and correcting insertion and deletion errors, we generated a simulated *E.coli* K-12 MG1655 (NCBI tax. ID 511145) sequence dataset by PBSIM (Ono *et al.*, 2013). We used the reference genome sequence [NC_000913.3, genome size 4 641 652 base pairs, (Hayashi *et al.*, 2006)] generated by Sanger sequencing as input for PBSIM. The quality information and the sequencing length distribution from real PacBio RS sequencing after secondary analysis (Pacific Biosciences, 2013) were used as the simulation parameters. PBSIM generated 34 898 sequences with 92 810 130 base pairs with average sequencing coverage of 20X. The average length of reads is 2660 bp and the reads' average error rate is 14.42%.

In the dataset, 3280 sequences fulfilled seed criterion. To control the scale of experiment, we randomly select 451 seed sequences for graph construction. After graph construction and protein domain filtration, 7093 sequence-domain pairs were kept as input to our program.

4.1.1 Performance of error correction

Both Frame-Pro and DAG-Con produced 7093 corrected sequences from the input PacBio simulation data. We first evaluated the performance of error correction of both tools by comparing their corrected sequences to the reference sequences. The comparison is conducted using BLAST (Altschul *et al.*, 1990). Figure 2 summarized the comparison of both tools on each read. For this data set, our tool can correct more errors in about half of the reads while DAG-Con corrects more errors in <5% of reads. In total, our method corrects 7884 more errors than DAG-Con. If we only count insertion and deletion errors, our method corrects 8374 more errors than DAG-Con. DAG-Con corrects 490 more mismatches than our method. It is expected because the profile HMM search is much more sensitive to frameshifts caused by gaps as they will significantly impact the alignment length and score. Figure 3 compares the number of remaining errors in corrected reads produced by both tools. It is clear that our method can produce more reads with no error or just 1 error.

4.1.2 Performance of profile HMM search

One of our major goals is to improve performance of profile homology search. This section will focus on evaluating the performance of protein domain homology search of corrected sequences. After correcting frameshifts, we expect that the homology search program can generate longer alignments with higher scores and smaller *E*-values for protein domains of interest. So the users can distinguish true domains from random alignments with higher confidence.

HMMER 3.1b2 was used to generate profile HMM alignments from corrected sequences. The domain composition of *E.coli* K12 (NCBI tax. ID 83333) proteome from Pfam (Release 29.0, Finn *et al.*, 2016) was used as the reference. 2,347 protein profile HMM domains were found in 7,093 output sequences. Some sequences have multiple hits. For all the data, including the PacBio simulation data, and the corrected sequences by both tools, we only keep the best alignment for each domain in our comparison. The changes of alignments' *E*-values, alignment lengths, and bit scores due to error correction are presented in Figure 4. On average, the length of the domain alignment increases from DAG-Con's 108.51 amino acids (a.a) to 150.36 a.a, which is closer to the average alignment length of 163.62 a.a in the reference proteome from Pfam. A two-sample Kolmogorov-Smirnov test (K-S test) on the alignments' lengths and *E*-values from our method and DAG-Con was applied to examine the statistical significance of difference from two methods. The p-values for the alignments' length, *E*-value, and bit score distributions were 5.4436×10^{-25}, 1.3237×10^{-25} and 1.8881×10^{-25}, respectively. Thus, the improvement by applying our method is statistically significant.

4.2 *Meiothermus ruber* DSM1279 sequences

The simulated data enables us to thoroughly test the parameters of our tool and also to evaluate various aspects of the performance. For the next two experiments, we apply our tool to real PacBio data with different coverage. As the real sequencing projects, in particular the transcriptomic sequencing projects and metagenomic sequencing projects, can contain transcripts or genomes with heterogeneous coverage, it is thus important to evaluate tools for data of different coverage.

In this experiment, we tested Frame-Pro on real *M.ruber* DSM1279 Pacbio Sequencing data. *Meiothermus ruber* (NCBI tax. ID 504728) is usually found in hot springs (Tindall *et al.*, 2010). It has genome size of 3 098 881 bps. The raw sequencing data from 1 SMRT cell in HGAP (Chin *et al.*, 2013) was used for this experiment. The raw data in total contains 177.4M bps with 59.6% GC content with approximate coverage of 60×.

Standard SMRT data processing pipeline (Pacific Biosciences, 2014) was used to filter the raw sequencing data to generate filtered subreads, which contain sequences passing the commonly used length and quality criteria. The filtered dataset has 90 114 302 bps and 36 180 sequences with 30× coverage. After protein domain families filtration, 33 911 sequence domain pairs passed the threshold and were input to downstream error correction pipelines.

4.2.1 Evaluate error correction performance by aligning outputs against the reference genome

To compare the error correction performance of Frame-Pro and DAG-Con on this real PacBio sequencing dataset, we used BLAST

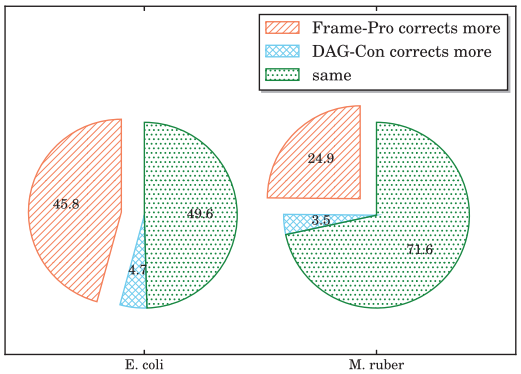

Fig. 2. Comparison of error correction performance for Frame-Pro and DAG-Con in simulated *E.coli* dataset and *M.ruber* dataset. Frame-Pro corrects more errors in larger fraction of PacBio reads

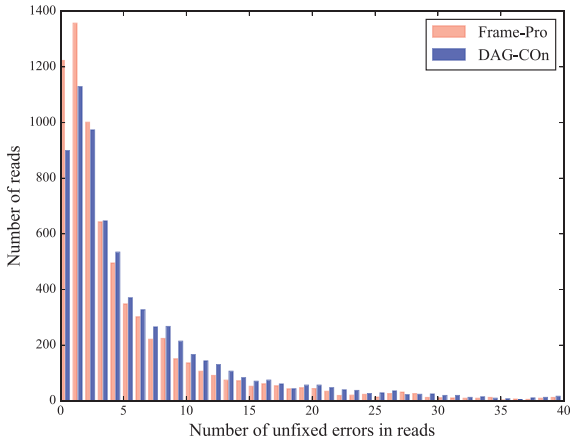

Fig. 3. Histogram of unfixed errors after error correction. X-axis represents the number of remaining errors in each read. Y-axis is the corresponding numbers of reads. Bin width is 1 and the figure only included the first 40 bins (i.e. up to 40 errors) due to space limitation

to search all outputs against the reference genome of *M.ruber* (NC_021081.1). The comparison of error correction for each read is summarized in Figure 2. In total, our method corrects 14 613 more errors than DAG-Con. If we only count insertion and deletion errors, our method corrected 15 924 more errors than DAG-Con. DAG-Con corrected 1311 more mismatches than our method.

Comparing to the previous simulation experiment, the difference of the error correction performance between Frame-Pro and DAG-Con decreased. The main reason is that the performance of DAG-Con usually improves with increased coverage (20–30×).

4.2.2 Evaluate the performance of profile homology search

The corrected sequences from Frame-Pro and the consensus sequences from DAG-Con were searched against protein domains in Pfam by HMMER3.1b2. The higher coverage of this dataset did improve the performance of DAG-Con.

For 2984 domains identified by both tools, we compared the best hit for each of them from two tools. The annotated domains from the reference proteome were downloaded from Pfam and were used as the reference. The changes of alignments' bit scores due to error correction are presented in Figure 5. No significant improvement can be observed for bit scores. Similar observations were made for alignments' lengths and *E*-values. Thus, the other two figures

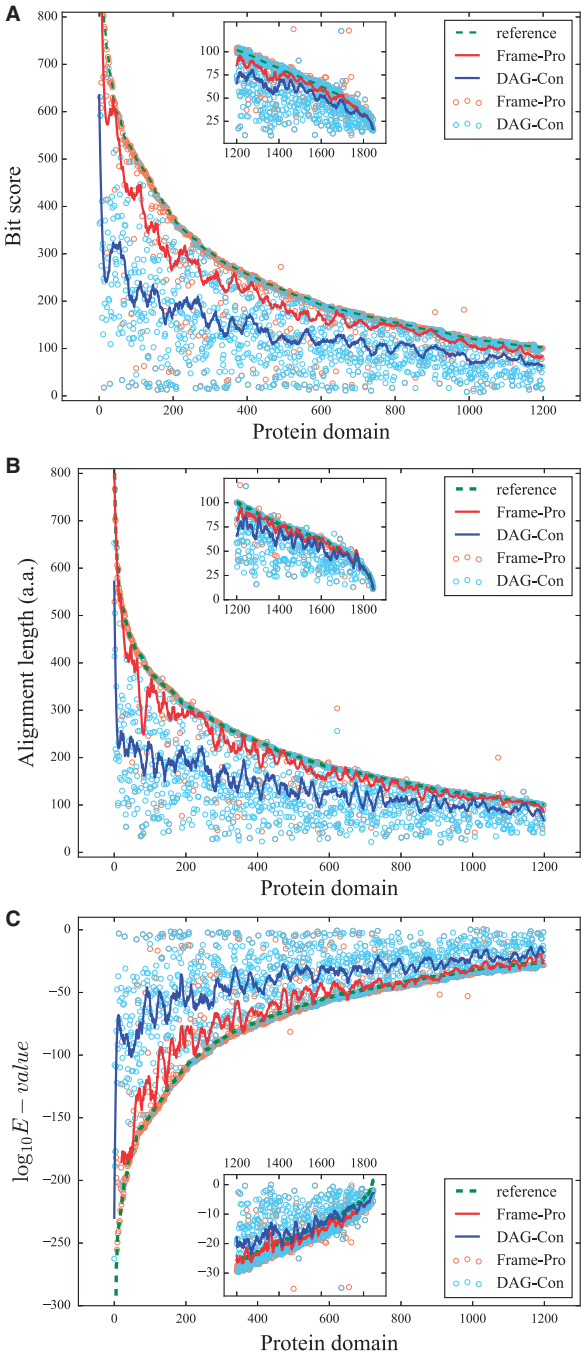

Fig. 4. The comparison of alignments' bit scores **(A)**, lengths in a.a. **(B)**, and *E*-values **(C)** for reference sequences and corrected sequences produced by Frame-Pro and DAG-Con in the *E.coli* simulated dataset. X-axis represents the domains. All domains are sorted by the reference values from Pfam. Red circles represent the values of HMM alignments for corrected sequences output by Frame-Pro. Blue circles represent the values of HMM alignments for corrected sequences output by DAG-Con. As there are many data points, all numbers produced by one tool are processed by SavityzkyGolay filter to generate a smoothed curve for clarity of presentation

were omitted. When compared with the average alignment length of 148.68 a.a from DAG-Con's consensus sequence, the average length of 157.92 a.a by Frame-Pro is much close to the reference's 158.69 a.a. We conducted a two-sample K-S test on the alignments' lengths, *E*-values, and bit scores from our method and DAG-Con. The *P*-values for alignments' lengths, *E*-values, and bit scoress distributions

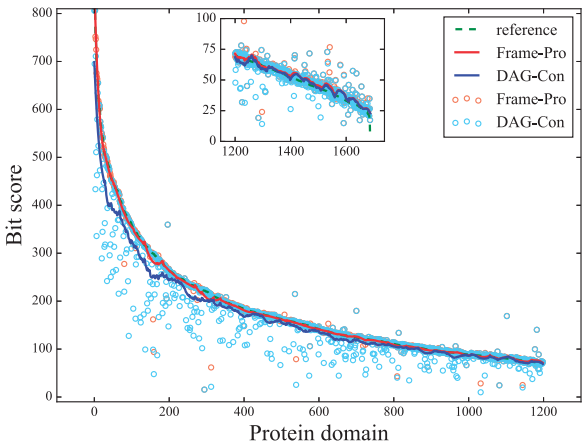

Fig. 5. The comparison of alignments' bit scores for reference sequences and corrected sequences produced by Frame-Pro and DAG-Con in the *M.ruber* dataset. X-axis represents the domains. All domains are sorted by the reference bit scores from Pfam. Red circles represent the values of HMM alignments for corrected sequences output by Frame-Pro. Blue circles represent the values of HMM alignments for corrected sequences output by DAG-Con. As there are many data points, all numbers produced by one tool are processed by SavityzkyGolay filter to generate a smoothed curve for clarity of presentation

were 0.2859, 0.2321 and 0.2007, respectively. Although Frame-Pro still generated longer alignments with better scores, DAG-Con achieved satisfactory performance of homology search for this data set because of the sufficient coverage.

4.3 Human arm metagenomic dataset

When PacBio is applied to metagenomic sequencing, one challenge is that some datasets do not have enough coverage for effective downstream analysis. Here, we applied Frame-Pro to analyze the protein domain composition in the human skin metagenomic data, which were sequenced from the human arm and foot (Tsai *et al.*, 2016).

The human arm sample was sequenced by linear PacBio RSII TdT (terminal deoxynucleotidyl transferase) sequencing platform. Sequences can be mapped with CHM1 human genome were removed as host human-derived DNA. This dataset cannot be further assembled by the HGAP pipeline due to the insufficient coverage, providing challenges to downstream analysis including protein domain classification. Thus in this experiment, we focus on the arm data set, which includes 16 388 sequences with 2 662 7191 bps.

4.3.1 Generate SAG and filtrate profile HMM domain

When compared with previous two datasets, the average read length of the human arm metagenomics dataset is only 1629 bps. To generate sufficient seed reads, the cut-off was changed to 3000 bps. Although Frame-Pro is not as sensitive as DAG-Con to the sequences' coverage, the filtration steps were affected as we use HMMER to search the consensus against Pfam with *E*-value 1000. After filtration, 602 sequence domain pairs were kept for further analysis. For each sequence domain pair, a corrected sequence using Frame-Pro and a consensus sequence using DAG-Con, both from the same graph, were generated.

4.3.2 Protein domain search using GA cutoff

Unlike previous data sets, we don't know all the composite species in the arm sample. Thus we cannot obtain all the ground truth protein domains in this data set, making the comparison with the reference domains difficult. In addition, without complete reference genomes of all species, we don't have the actual sequences corresponding to these reads and thus we cannot evaluate the number of unfixed errors. In order to examine the results, we thus use a stringent threshold to examine the domain sets identified for corrected sequences output by Frame-pro and DAG-Con. We use the gathering (GA) cutoffs from Pfam as the threshold for domain composition analysis because GA cutoffs are family specific bit score thresholds aiming to minimize false positive rate and to maximize the coverage (Punta, 2012). In Frame-Pro's results, 84 domains are above the GA cutoff, comparing to 49 domains in DAG-Con's results. Frame-Pro only missed one domain in DAG-Con's result with score slightly less than the corresponding GA cutoff. According to Pfam, all domains uniquely found by Frame-Pro were annotated in microbial species. The list of domains can be found in our website.

For the commonly identified 48 domains, we compared the alignments' bit scores, alignment length, and *E*-values on corrected sequences output by two tools (See Fig. 6). We conducted a two-sample K-S test on the three metrics for all alignments. The quality of alignment improve significantly. The *P*-values for the the alignments' length, *E*-value, and bit score distributions were 6.05×10^{-6}, 3.66×10^{-12} and 7.64×10^{-13}, respectively.

Without reference sequences, there could be one possibility that Frame-Pro over-corrects the errors in order to maximize the alignment score. In order to test this possibility, we examined the identified domains for one reference species: *Corynebacterium aurimucosum*. Although the complete composite species of the arm sample is unknown, the phylogenetic analysis in (Tsai *et al.*, 2016) and other articles (Gao *et al.*, 2007; Trost *et al.*, 2010) showed that *C.aurimucosum* is relatively abundant in this sample. In addition, this species has annotated domains (NCBI tax. ID 548476) in the Pfam database. Thus, we focus on evaluating how many of the annotated domains in this species can be identified by tested methods. By using GA cutoff, Frame-Pro can recover 41 domains while DAG-Con can recover 24 domains. The set identified by DAG-Con is a subset of ours. Thus, this experiment shows that extra domains identified by us are not likely false predictions. This experiment adds evidence that Frame-Pro can identify more domains for data with very low coverage and thus provides complementary analysis for metagenomic data.

5 Conclusion and discussion

In this work, we developed a profile homology search tool for PacBio sequencing data. By correcting insertion or deletion errors, our implementation can improve homology search performance, including alignment scores, lengths and *E*-values. In particular, for sequencing data with low sequencing coverage (around or lower than 20×), our tool can significantly correct more errors and improve the sensitivity of homology search by finding more correct domains. Being able to conduct sensitive homology search for sequencing data of low coverage is important for various sequencing projects including metagenomic and transcriptomic sequencing. Usually, as the transcripts or species have highly different abundance and thus heterogeneous coverage, conducting homology search needs to consider the input of a full spectrum of coverage. Many rare species in an environmental community or rare transcripts are particularly interesting for biological discovery.

Although our current implementation is based on profile homology search, which compares sequences with profile HMMs. Our

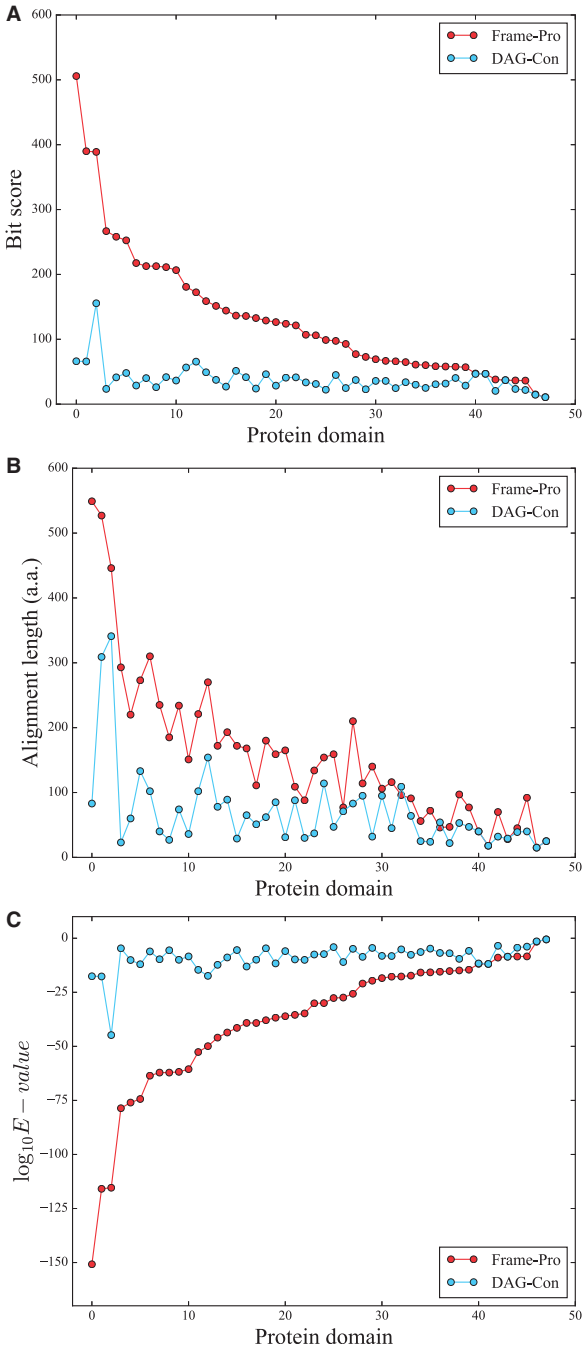

Fig. 6. The comparison of alignments' bit scores **(A)**, lengths in a.a. **(B)**, and *E*-values **(C)** for 48 domains commonly identified by Frame-Pro and DAG-Con in the arm data set. X-axis represents the domains

method can be easily extended to pairwise alignment as well. In that case, the profile HMM will be replaced a single sequence. Existing pairwise alignment algorithm can be extended to align an alignment graph with a single sequence.

Like the error correction stage in HGAP, our tool does not rely on hybrid sequencing either, making it convenient for various applications. However, one limitation is that our tool is not a general error correction tool because it can only correct errors in regions that are homologous to reference sequences. This is not a problem for species with highly packed coding regions. But for genomes with large fractions of non-coding regions, our tool is not designed for error correction in the whole genome.

Finally, we only tested our application in PacBio data. But our method can be extended to other sequencing data with similar types of errors. For example, we will test our tool on the data produced by Nanopore technology.

Acknowledgements

We would like to thank Dr Julia Oh for providing us the human skin metagenomics dataset.

Funding

This work is supported by NSF CAREER Grant DBI-0953738.

Conflict of Interest: none declared.

References

Altschul,S. *et al.* (1990) Basic local alignment search tool. *J. Mol. Biol.*, **215**, 403–410.

Antonov,I. and Borodovsky,M. (2010) Genetack: frameshift identification in protein-coding sequences by the viterbi algorithm. *J. Bioinformatics Comput. Biol.*, **08**, 535–551.

Birney,E.M., et al. (2004) Genewise and genomewise. *Genome Res.*, **14**, 988–995.

Borodovsky,M. and McIninch,J. (1993) Genmark: Parallel gene recognition for both dna strands. *Comput. Chem.*, **17**, 123–133.

Brown,N.P. *et al.* (1998) Frame: detection of genomic sequencing errors. *Bioinformatics*, **14**, 367–371.

Chaisson,M.J. and Tesler,G. (2012) Mapping single molecule sequencing reads using basic local alignment with successive refinement (blasr): application and theory. *BMC Bioinformatics*, **13**, 1–18.

Chaisson,M.J. *et al.* (2015) Resolving the complexity of the human genome using single-molecule sequencing. *Nature*, **517**, 608–611.

Chang,W.I. and Lawler,E. (1994) Sublinear expected time approximate string matching and biological applications. *Algorithmica*, **12**, 327–344.

Chin,C.S. *et al.* (2013) Nonhybrid, finished microbial genome assemblies from long-read SMRT sequencing data. *Nat. Methods*, **10**, 563–569.

Conlan,S. *et al.* NISC Comparative Sequencing Program. (2014) Single-molecule sequencing to track plasmid diversity of hospital-associated carbapenemase-producing enterobacteriaceae. *Sci. Trans. Med.*, **6**, 254ra126.

Durbin,R. *et al.* (1998). *Biological Sequence Analysis*. Cambridge University Press. Cambridge Books Online.

Eddy,S.R. (1998) Profile hidden markov models. *Bioinformatics*, **14**, 755–763.

Eddy,S.R. *et al.* (2015). Hmmer 3.1b2. http://hmmer.org/. March 2016, last date accessed.

Finn,R.D. *et al.* (2016) The pfam protein families database: towards a more sustainable future. *Nucleic Acids Res.*, **44**, D279–D285.

Gao,Z. *et al.* (2007) Molecular analysis of human forearm superficial skin bacterial biota. *Proc. Natl. Acad. Sci.*, **104**, 2927–2932.

Gîrdea,M. *et al.* (2009). *Algorithms in Bioinformatics: 9th International Workshop, WABI 2009, Philadelphia, PA, USA, September 12-13, 2009. Proceedings*, chapter Back-Translation for Discovering Distant Protein Homologies, pp. 108–120. Springer Berlin Heidelberg, Berlin, Heidelberg.

Gîrdea,M. *et al.* (2010) Back-translation for discovering distant protein homologies in the presence of frameshift mutations. *Algorithms Mol. Biol*, **5**.

Guan,X. and Uberbacher,E. (1996) Alignments of dna and protein sequences containing frameshift errors. *Comput. Appl. Biosci.*, **12**, 31–40.

Haft,D.H. *et al.* (2003) The TIGRFAMs database of protein families. *Nucleic Acids Res.*, **31**, 371–373.

Halperin,E. *et al.* (1999) FramePlus: aligning DNA to protein sequences. *Bioinformatics*, **15**, 867–873.

Hayashi,K. *et al.* (2006) Highly accurate genome sequences of Escherichia coli k-12 strains mg1655 and w3110. *Mol. Syst. Biol.*, **2**, 2006.0007.

Kislyuk,A. *et al.* (2009) Frameshift detection in prokaryotic genomic sequences. *Int. J. Bioinformatics Res. Appl.*, **5**, 458–477.

Koren,S. *et al*. (2012) Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nat. Biotechnol*., **30**, 693–700.

Koren,S. *et al*. (2013) Reducing assembly complexity of microbial genomes with single-molecule sequencing. *Genome Biol*., **14**, 1–16.

Meyer,F. *et al*. (2009) FIGfams: yet another set of protein families. *Nucleic Acids Res*., **37**, 6643–6654.

Ono,Y. *et al*. (2013) Pbsim: Pacbio reads simulatortoward. *Bioinformatics*, **29**, 119–121.

Pacific Biosciences (2013) E coli k12 mg1655 resequencing. https://github.com/PacificBiosciences/DevNet/wiki/E-coli-K12-MG1655-Resequencing (March 2016, date last accessed).

Pacific Biosciences (2014) Smrt analysis. http://www.pacb.com/products-and-services/analytical-software/smrt-analysis/ (March 2016, date last accessed).

Pellegrini,M. and Yeates,T.O. (1999) Searching for frameshift evolutionary relationships between protein sequence families. *Proteins*, **37**, 278–283.

Peltola,M. *et al*. (1986) Algorithms for the search of amino acid patterns in nucleic acid sequences. *Nucl. Acids Res*., **14**, 99–107.

Prestat,E. *et al*. (2014) FOAM (Functional Ontology Assignments for Metagenomes): a Hidden Markov Model (HMM) database with environmental focus. *Nucleic Acids Res*., **42**, e145.

Punta,M. (2012) Pfam: the protein families database. *Nucleic Acids Res*., **40**, D290D301.

Quail,M.A. *et al*. (2012) A tale of three next generation sequencing platforms: comparison of ion torrent, pacific biosciences and illumina miseq sequencers. *BMC Genomics*, **13**, 1–13.

Rasko,D.A. *et al*. (2011) Origins of the E. coli strain causing an outbreak of hemolyticuremic syndrome in Germany. *N. Engl. J. Med*., **365**, 709–717. PMID: 21793740.

Rhoads,A. and Au,K.F. (2015) Pacbio sequencing and its applications. *Genomics Proteomics Bioinformatics* **13**, 278–289.

Schiex,T. *et al*. (2003) Framed: a flexible program for quality check and gene prediction in prokaryotic genomes and noisy matured eukaryotic sequences. *Nucleic Acids Res*., **31**, 3738–3741.

Tilgner,H. *et al*. (2014) Defining a personal, allele-specific, and single-molecule long-read transcriptome. *Proc. Natl. Acad. Sci. USA*, **111**, 9869–9874.

Tindall,B.J. *et al*. (2010) Complete genome sequence of meiothermus ruber type strain (21). *Stand. Genomic Sci*., **3**, 26–36.

Trost,E. *et al*. (2010) Complete genome sequence and lifestyle of black-pigmented corynebacterium aurimucosum atcc 700975 (formerly c. nigricans cn-1) isolated from a vaginal swab of a woman with spontaneous abortion. *BMC Genomics*, **11**, 1–16.

Tsai,Y.C. *et al*. (2016) Resolving the complexity of human skin metagenomes using single-molecule sequencing. *mBio*, **7**, e01948–15.

Wang,Q. *et al*. (2013) Ecological patterns of nifh genes in four terrestrial climatic zones explored with targeted metagenomics using framebot, a new informatics tool. *mBio*, **4**, e00592–13.

Zdobnov,E.M. and Apweiler,R. (2001) InterProScan an integration platform for the signature-recognition methods in InterPro. *Bioinformatics*, **17**, 847–848.

Zhang,Y. and Sun,Y. (2011) Hmm-frame: accurate protein domain classification for metagenomic sequences containing frameshift errors. *BMC Bioinformatics*, **12**, 1–10.

Zhang,Z. *et al*. (1997). *Proc of RECOMB 97: the first international conference on computational molecular biology*, chapter Aligning a DNA sequence with a protein sequence. ACM. http://dl.acm.org/citation.cfm?id=267893.