BMC Bioinformatics

RESEARCH ARTICLE Open Access

Effective machine-learning assembly for next-generation amplicon sequencing with very low coverage

Louis Ranjard*, Thomas K. F. Wong and Allen G. Rodrigo

Abstract

Background: In short-read DNA sequencing experiments, the read coverage is a key parameter to successfully assemble the reads and reconstruct the sequence of the input DNA. When coverage is very low, the original sequence reconstruction from the reads can be difficult because of the occurrence of uncovered gaps. Reference guided assembly can then improve these assemblies. However, when the available reference is phylogenetically distant from the sequencing reads, the mapping rate of the reads can be extremely low. Some recent improvements in read mapping approaches aim at modifying the reference according to the reads dynamically. Such approaches can significantly improve the alignment rate of the reads onto distant references but the processing of insertions and deletions remains challenging.

Results: Here, we introduce a new algorithm to update the reference sequence according to previously aligned reads. Substitutions, insertions and deletions are performed in the reference sequence dynamically. We evaluate this approach to assemble a western-grey kangaroo mitochondrial amplicon. Our results show that more reads can be aligned and that this method produces assemblies of length comparable to the truth while limiting error rate when classic approaches fail to recover the correct length. Finally, we discuss how the core algorithm of this method could be improved and combined with other approaches to analyse larger genomic sequences.

Conclusions: We introduced an algorithm to perform dynamic alignment of reads on a distant reference. We showed that such approach can improve the reconstruction of an amplicon compared to classically used bioinformatic pipelines. Although not portable to genomic scale in the current form, we suggested several improvements to be investigated to make this method more flexible and allow dynamic alignment to be used for large genome assemblies.

Keywords: Assembly, Amplicon, Machine learning, Western-grey kangaroo, Mitochondrion

Background

De novo assembly algorithms classically use graph, de Bruijn or overlap-layout-consensus, to join short sequencing reads into longer contigs. However, when the short-reads coverage is very low, only short contigs can be reconstructed because of the occurrence of uncovered gaps in the sequence [1]. In this case, the availability of a reference sequence can be beneficial to connect and order these contigs, an approach known as reference-guided assembly or homology-guided assembly [2, 3]. The reads are mapped onto this reference and a contig is constructed by taking the consensus of the short-reads at each position. However, some gaps in the mapping of the reads onto the reference may remain if the available reference is too distant phylogenetically from the sequence the short-reads originate from. This is because the short-reads that cannot, or can only partially, be mapped to the distant reference are discarded or trimmed. The information contained in the discarded or trimmed sequences of the reads is therefore lost. Hence, improvements in the alignments of the reads to the reference that are able to take advantage of this unexploited information should improve the assemblies.

Iterative referencing proposes to align all the reads to the reference and then update the reference sequence by

*Correspondence: louis.ranjard@anu.edu.au
The Research School of Biology, The Australian National University, Canberra, Australia

calling the consensus of the reads. Once the reference has been updated, several additional iterations of read mapping/reference update can be performed to progressively improve the results [4–8]. Significant improvements in the mapping accuracy of the reads is achieved thanks to this approach [9]. Subsequently, it has been shown that dynamic approaches can offer comparable improvements while performing less data processing, i.e. only requiring a single iteration of read mapping [9]. In dynamic mapping, the reference is updated continuously as the reads are aligned onto it in an online fashion. Hence, the information obtained from the alignments of previous reads is used to map future reads. Dynamic strategies can be especially useful when the read sequences are highly divergent from the reference [9]. However, the treatment of insertions and deletions (indels) remains a problem to dynamic mappers as the coordinates of the reads have to be continuously recalculated [9] with a new indexing of the reference.

Here, we introduce a new online read aligner, Nucleoveq [10], and assess how it can improve the alignment of the reads when the reference is distant phylogenetically from the reads. This is a difficult task because, in this case, a large portion of the reads cannot be mapped to the reference. Using a machine learning approach, we present an algorithm that is able to dynamically perform substitutions and indels in the reference. The probability of each base at each position is learned from the past read alignments. A dynamic time warping algorithm uses these probability vectors directly to measure the edit distance between a read and the reference at the best alignment position. This is contrasting from previously proposed dynamic mapping approaches that record a counter for the different possible variants between the sequential updates of the reference [9]. In the present method, the reference is updated after every read alignments. Note that our algorithm allows the reference to be updated with insertions and deletions at any position in the reference. We show that, because the reference sequence is continuously updated according to the alignment of the previous reads, the alignment of the read gradually improves. We demonstrate that this feature allows us to take advantage of distantly related reference sequence and improve the resulting short-reads assembly.

Results

In order to assess our method, we asked whether the improved read alignment provided by a dynamic approach results in better guided assemblies. We compared the assembly obtained from the dynamic aligner to classic assembly techniques. Briefly, we tested three assembly pipelines referred to as: **mapping**, mapping of all the reads to the reference followed by update of the reference; **learning**, dynamic time warping alignment of the reads with simultaneous machine learning approach to update the reference (Nucleoveq [10], see online Methods for details); **de novo**, reference-free assembly of the reads using a de Bruijn graph approach. Additionally, two hybrid approaches were evaluated, the **de novo + mapping** and the **de novo + learning** pipelines where the contigs obtained by the de novo assembly of the reads are respectively mapped and aligned before updating the reference. A set of computer simulations was performed to compare the reconstructed sequence obtained by these strategies when coverage is very low ($1 - 5\times$) and with varying phylogenetic distances between the original sequence and the sequence used as reference.

We used sequencing short-reads obtained from a study of mitochondrial amplicons of the western-grey kangaroo, *Macropus fuliginosus* [11, 12]. Focusing on a 5,000 bp amplicon allowed us to conduct extensive resampling of the reads. Published mitochondrial reference sequences from the following species were used as references: the eastern-grey kangaroo (*Macropus giganteus*, Genbank accession NC_027424), the swamp wallaby (*Wallabia bicolor*, Genbank accession KJ868164), the Tasmanian devil (*Sarcophilus harrisii*, Genbank accession JX475466) and the house mouse (*Mus musculus*, Genbank accession NC_005089). The computer simulations were performed using the most divergent amplicon (Amplicon 3) identified by [11] which is located from position 11,756 to 16,897 in the eastern-grey kangaroo mitochondrial genome, total length of 5,130bp. This region contains the mitochondrial D-loop and, at the time of this study, the nucleotide sequence is not covered in the western-grey kangaroo mitochondrial genome (Genbank accession KJ868120). These species were chosen at increasing phylogenetic distance from the western-grey kangaroo (Table 1) but with no changes in their gene order. The homologous regions were selected in each species by aligning the amplicon sequence to each mitochondrial genome in Geneious version 10.2.4 [13]. Then, a region spanning from position 11,000 bp to 1,200 bp was used for each circular reference genome except the eastern-grey kangaroo. For the eastern-grey sequence the homologous amplicon region was used [11]. This was done to reduced computational time while still keeping some part of the sequences located outside of the target region, i.e. from which the short-reads originate. The quality of the different assemblies was evaluated by using two statistics: first, the number of errors while aligning the reconstructed amplicon and the true western-grey kangaroo amplicon sequences; second, the length of the reconstructed sequence.

Reference positions covered

The total read coverage in the reference was recorded for both the **mapping** and **learning** approaches to assess

Table 1 The four different reference sequences used to guide the reconstruction of the western-grey kangaroo mitochondrial amplicon from short sequencing reads. For each circular mitochondrial genome, the genome coordinates of the extracted region are indicated as well as its length. The percentage identity to the western-grey amplicon is calculated on the homologous regions only, i.e. the non-aligned sections at the beginning and the end of the alignment are not taken into account

Species	Genbank accession	Start position	End position	Length of extracted region (bp)	Percentage identity to western-grey amplicon
Eastern-grey kangaroo	NC_027424	11,749	47	5,186	91.4%
Swamp wallaby	KJ868164	11,000	1,200	7,075	86.8%
Tasmanian devil	JX475466	11,000	1,200	7,336	65.7%
House mouse	NC_005089	11,000	1,200	6,500	59.0%

whether dynamic reference updates increases the reads alignment rate. As expected, the number of bases covered increases with the number of reads sampled (Fig. 1). However, with distant reference sequences, i.e. the Tasmanian devil and the house mouse, the mapping rate of the reads is very low while the alignment rate is less affected by the increasing phylogenetic distance of the reference. Moreover, with these two species used as reference, the mapping rate remains low even though the depth of coverage increases. Generally, it appears that the variance in the mapping rate is higher than for the alignment rate.

Assembly evaluation

A total of 2000 computer simulations were conducted. For coverage values ranging from 1× to 5×, the number of reads required to achieve such coverage was calculated and a corresponding subset of reads was randomly chosen among the full set. Then, for each of the four species reference sequence, the five pipelines were tested. A total of 100 replicates was performed for each setting. To compute the number of errors and length of the reconstructed sequence statistics, the pairwise alignment was computed using the Needleman-Wunsch algorithm with affine gap penalty scheme, the NUC44 scoring matrix and null gap penalties at the end of the sequences. The non-aligned sequences at the beginning and at the end of the alignment were discarded and the remaining sequence length was reported for comparisons between pipelines. The number of errors was computed as the Hamming distance between the remaining aligned sequences.

Overall, the **learning** approaches offered the best compromise between limiting the error rate and recovering the true length of the amplicon sequence (Fig. 2). In all simulation settings, the de Bruijn graph assemblies (**de novo assembly**) achieved a very low error rate. On the other hand, this approach was only able to generate relatively short assemblies compared to the other pipelines (Fig. 2). However, with increasing coverage the length of the de novo assembled contigs increased confirming the suitability of de Bruijn graph based methods for assembling short-reads when the depth of coverage is high. Specifically, our simulations showed that at least a 20× coverage is required to reconstruct the full length amplicon with this approach (Fig. 3).

When using distant references (Tasmanian devil and the house mouse), the hybrid approaches (**de novo** + **mapping** and **de novo** + **learning**) produced less errors than the same algorithms used on the raw reads (Fig. 2). However, when using more closely related sequences as references, the **de novo** + **mapping** method produced more errors than the **mapping** pipeline. This is putatively the consequence of the low coverage of the de novo assembly of the reads, i.e. the **de novo** only generated very short contigs. On the other hand, the **de novo** + **learning** and **learning** generated similar amount of errors with closely related reference sequences used as guides. With more distant reference sequences, the **de novo** + **learning** produced less errors than the **learning** pipeline. While both pipelines benefit from an increase in read coverage, the **de novo** + **learning** returned the lowest amount of errors with distant references.

When the reference sequence was chosen phylogenetically close to the reads sequence, i.e. eastern-grey kangaroo and swamp wallaby, and the coverage was set to 5×, all pipelines, except **de novo assembly**, generated assemblies of comparable length from the truth. With decreasing coverage, the reconstructed sequence length also decreased for all methods. This is particularly noticeable for approaches that use mapping of the reads as the mapping rate strongly decreases with increasing phylogenetic distance of the reference (Fig. 1). On the other hand, the two methods that use dynamic programming to align the reads were able to reconstruct sequences of length comparable to the western-grey amplicon using distant reference (Fig. 2). It is noticeable that in these cases the variance of both the length and the error rate for the mapping-based pipelines is comparatively very high. This is highly likely to be the consequence of the higher variance in the mapping rate for these pipelines and it may indicate that the mapping-based methods are more

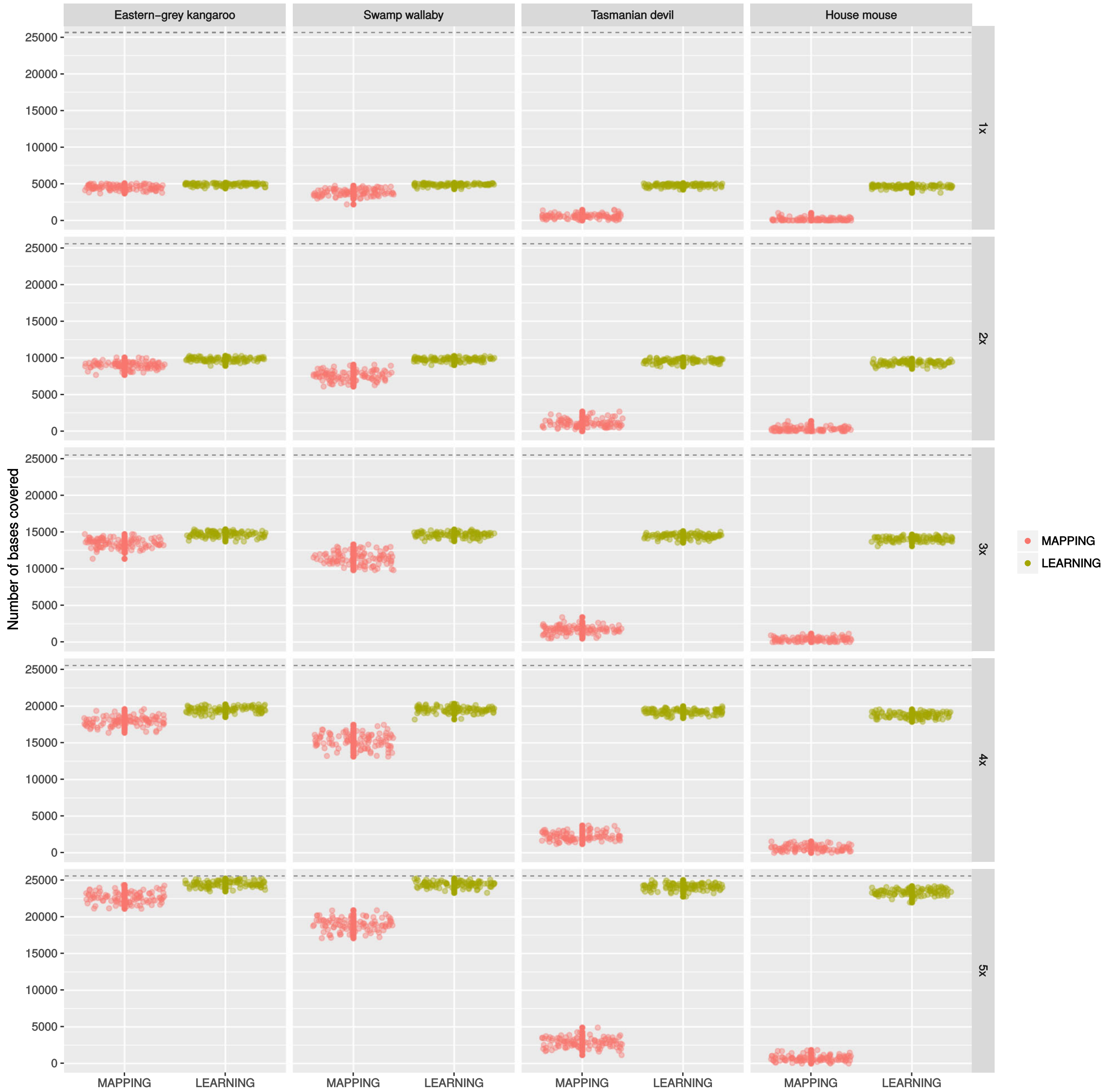

Fig. 1 Realised coverage obtained by mapping (MAPPING) or aligning (LEARNING) sequencing reads to increasingly distant homologous reference sequences. The short-reads originate from a western-grey kangaroo amplicon of length 5,130bp with 5× coverage, therefore the expected number of bases covered is ∼25,000 (dashed line)

sensitive to a non-uniform coverage of the re-sampled reads. Moreover, the variation between the different mitochondrial genomes is not uniformly distributed and the mapping of the reads would be more difficult when they originate from highly divergent regions.

Comparison to iterative referencing Additionally, an iterative mapping approach was implemented by repeating the **mapping** pipeline five times using the updated reference obtained at the previous iteration. This approach was tested with the Tasmanian devil reference sequence at coverage 5× as it is expected that the best improvements would be obtained with higher coverage. As expected iterative mapping improved the sequence reconstruction (Table 2). Each additional iteration of the mapping of the reads allowed the error rate to decrease as more reads could be mapped. However, the improvements were limited. After five iterations, the error rate and the length of the reconstructed sequence were still worse than the ones obtained with the **de novo + learning**

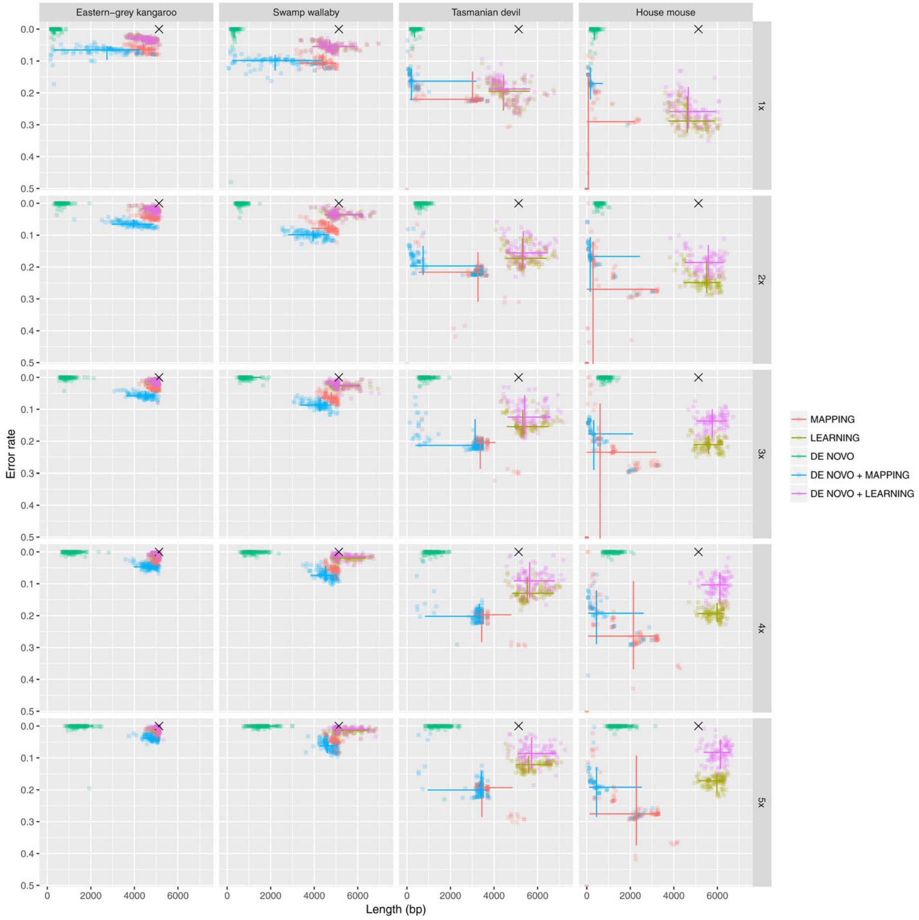

Fig. 2 Number of errors and length in nucleotide of the reconstructed amplicon for each bioinformatic pipeline and simulation settings. The 95% intervals are shown as solid lines for each method along both dimensions (reconstructed amplicon length and error rate)

pipeline (Fig. 2). Similar limited improvements were obtained using the other reference sequences and coverage values. No improvements in the number of bases covered was observed after three iterations for eastern-grey kangaroo and swamp wallaby references, and after eight iterations for the more distant relative references (Fig. 4).

Assembly of *Macropus fuliginosus* mitochondrial genome To demonstrate the applicability of the method, a full mitochondrial genome was assembled from short-reads using a sister species reference sequence. At the time of this study, the western-grey kangaroo mitochondrial genome is only partial and lacks the hyper variable region (Genbank accession KJ868120) [11]. We used our method to reconstruct the full mitochondrial genome of the individual identified as “KA” in [11]. First, the partial mitochondrial genome of the western-grey kangaroo was completed using the eastern-grey kangaroo reference (Genbank accession NC_027424) generating an hybrid full genome template. The sequencing reads generated from three western-grey kangaroo mitochondrial amplicons, of length 4641bp, 4152bp and 5140bp (83% of the genome, [11]), were then aligned to this reference template using Nucleoveq. One of the amplicon fully spans the missing region in the western-grey kangaroo mitochondrial genome reference. Reads were sub-sampled so that to obtain a coverage of 5×. Because the coverage was low, ten iterations were conducted to insure that the reference was fully covered by randomly sampled reads.

The ten replicates of the mitochondrial genome assembly were aligned with an average of 99% identity. Visual inspections of the alignment of the replicates showed that these differences occurred in regions with no coverage. The consensus sequence of the ten replicates was compared to the high coverage assembly of the mitochondrial assembly from [11]. As expected, some errors were observed at the beginning or end of the three mitochondrial amplicons. Because the short-read coverage was extremely low in these regions, it was very unlikely that the sub sampling of the reads retrieved these sequences. A new mitochondrial genome was generated by correcting

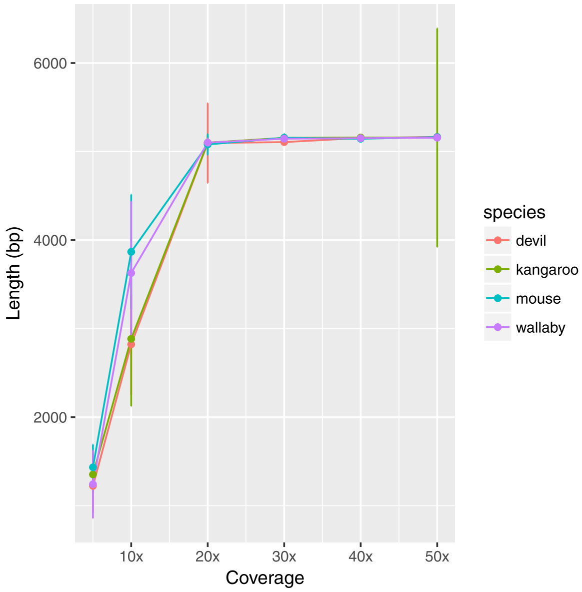

Fig. 3 With more than 20× coverage, the de Bruijn graph assembly is able to reconstruct the expected amplicon length (5,130bp)

the consensus sequence with the high coverage information. The newly assembled western-grey mitochondrial genome was annotated in Geneious version 10.2.4 [13] using the eastern-grey kangaroo mitochondrial genome as a reference. The western-grey complete mitochondrial genome is on Genbank under accession number MH717106.

Discussion

By iteratively aligning short sequencing reads and updating the reference sequence, we were able to improve the reconstruction of the read sequence, resulting in assemblies of comparable length to the truth while limiting the number of errors. The improvement of this dynamic alignment method over de Bruijn graph- or the mapping-based approaches tested here can be explained by two

Table 2 Iterative mapping lowers the error rate and the length of the reconstructed sequences

	Length (bp)	Error rate	Coverage (number of bases)
Mapping iteration 0	3376 [3244-3442]	0.206 [0.201-0.207]	1558 [1157-1750]
Mapping iteration 1	3376 [3269-3442]	0.198 [0.192-0.206]	3026 [2142-3422]
Mapping iteration 2	3386 [3269-3442]	0.183 [0.180-0.192]	3618 [2858-4296]
Mapping iteration 3	3423 [3279-3443]	0.173 [0.169-0.185]	4712 [3524-5127]
Mapping iteration 4	3423 [3279-3443]	0.168 [0.159-0.185]	4903 [3561-6593]
Mapping iteration 5	3442 [3279-3443]	0.162 [0.154-0.185]	5247 [3561-7129]

The median and the first and third quartiles are indicated for each statistic, the coverage was 5× with the Tasmanian devil used as reference sequence

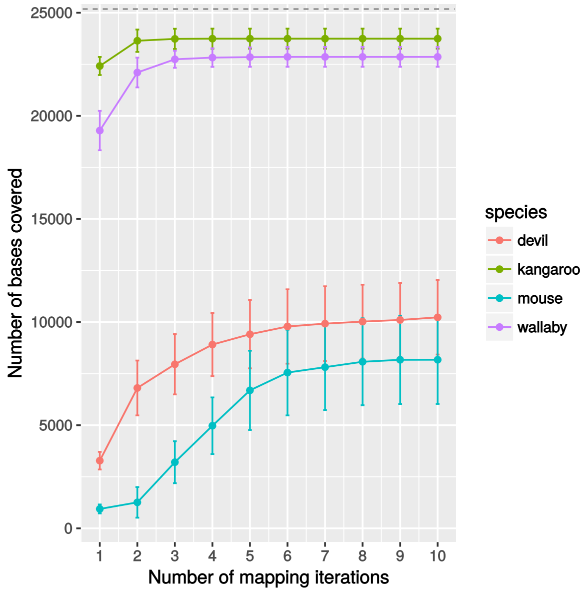

Fig. 4 Increasing the number of mapping iteration of the same reads does improve the number of aligned reads, measured as number of bases covered, but only to a limited extend. The short-reads originate from an amplicon of length 5,130bp with 5× coverage, therefore the expected number of bases covered is ∼25, 000 (dashed line)

factors. First, the alignment rate is higher when using dynamic programming over the Burrows-Wheeler transform approach used for mapping the reads. Second, the progressive modifications of the reference, as reads are aligned onto it, facilitate the alignment of the following reads because the reference is continuously pulled closer to the reads sequence [9]. This is particularly useful when only a phylogenetically distant reference sequence is available for a reference-guided assembly. Actually, our results showed that the static mapping of the reads is not possible when the reference is too distant from the reads, as demonstrated by a very low mapping rate.

The drawback of our dynamic programming method for read alignment is memory usage. The memory required to build the alignment matrix M (see Methods) precludes the direct usage of this method for large genome assemblies. While our approach is relevant to small genome assemblies, e.g. mitochondrial, supplementary work would be required to adapt this approach to large genome read alignments. For example, while it is not possible to directly align the reads to a large genome, a first search could help identify short windows, i.e. few thousands bases, in the reference sequence where the reads could then be aligned more accurately by our algorithm. In the current implementation of the method, it is optionally possible to take advantage of the known mapping positions of the reads by passing a mapping file as argument. This technique can massively reduce the memory requirements as only a window of specified size around these positions will be considered for performing the alignment. Our algorithm could also be combined with other methods to find the potential locations of each read in the genome prior to performing the alignments. The seed-based algorithm used by Blast [14] or some kmer-based seed searches [15, 16] are obvious candidates. However, when the reference sequence is distant from the reads, it is not possible to initially map all the reads onto it. It is therefore inevitable to re-align or re-map these reads once the reference has been partially updated.

Our method improves previous dynamic reference building approaches in that it allows the reference to be updated with insertions and deletions. Previously, Liao and co-authors [15] proposed a seed and vote approach to locate indels. [9] proposed a dynamic mapping approach where the reference is iteratively updated with the read sequences but indels were not fully supported [17]. Our method not only locates but also aligns and corrects the reference sequence with indels, facilitating further the subsequent read alignments. This approach comes at the computational cost of realigning each read onto the reconstructed reference. However, in our algorithm each read is treated independently and the updates of the reference are only performed according to the information from one read at a time. This is different from graph-based and iterative referencing methods that need all reads to be aligned before calling the variants. As a consequence, parallelization may be used to distribute batch of reads to be analysed independently prior to merging the several assemblies.

The threshold limit for performing insertions and deletions was set to be equal to the learning rate (see Methods). Therefore, indels will not be performed when the read alignment is poor. However, there is no particular reasons to use this value and other values could be used based on other statistics. Preliminary tests (data not shown) indicated that this value nevertheless returned best assemblies. Similarly, the indels costs was set to equal the maximum possible distance between a pair of nucleotide vectors. Preliminary tests using grid search showed that similar results were obtained while varying their values (data not shown). However, this hyperparameters could also be set to depend on some other parameters measured on the data and further investigations could be conducted to explore these possibilities.

Finally, the learning rate hyper-parameter was set to depend on the alignment distance. Classically in machine learning algorithms, the learning rate is set to decay through the learning process [18, 19]. Conversely, in our algorithm, it is expected that the rate will increase as the reference sequence gets closer to the reads. Alternative learning rate schedules could be tested, for example cyclic methods as proposed by [20] for training deep neural networks. Moreover, we only considered one epoch for learning, i.e. one iteration over the full set of reads. In other words, the total read set is only seen once to learn the amplicon sequence. Because the reads are chosen in a random order, the assembled sequence will potentially be different between distinct runs of the algorithm and there is no guarantee to converge on the best assembly. Performing the learning over multiple epochs could potentially improve the convergence among runs at the cost of processing time.

The presented method can therefore improve assemblies in experiments with low coverage of the input DNA material by the sequencing reads. While it is not common to design targeted sequencing strategies with low coverage, they can nevertheless be encountered in other situations. For example, when only a low amount of DNA is available, e.g. ancient DNA studies or challenging DNA extraction conditions. Moreover, assemblies are sometime conducted from experiments that were designed for different purposes. For instance, the reads obtained for a transcript sequencing experiment could be used to sequence the mitochondrial genome of a species lacking a reference [21]. Permitting assembly from lower amount of reads would therefore allow researchers to extract more information from sequencing experiments.

Conclusions

We introduced an algorithm to perform dynamic alignment of reads on a distant reference. We showed that such approach can improve the reconstruction of an amplicon compared to classically used bioinformatic pipelines. Although not portable to genomic scale in the current form, we suggested several improvements to be investigated to make this method more flexible and allow dynamic alignment to be used for large genome assemblies.

Methods

Learning from dynamic programming alignment of the reads to the reference

In essence, the algorithm consists in aligning the reads to the reference using dynamic time warping. Then, an "average" sequence of the aligned region is computed from the best path of the local free-ends alignment [22]. This approach was originally designed to perform unsupervised clustering of bioacoustic sequences [23]. In this work, a similar algorithm is implemented to analyse nucleotide sequences: each nucleotide position in a sequence is represented as a four elements vector, the Voss representation [24], encoding the probability of each base according to previously aligned reads. This numerical representation of DNA sequence is appropriate for the comparison of DNA sequences [25] and their classification[26]. In molecular biology, a similar algorithm has been applied to the clustering of amino acid sequences [27] where vector quantization is used to estimate the probability density of amino acids. In the area of genomic signal processing, dynamic time warping approaches have been successful at classifying various representations of genomic data [28–31].

We consider two sequences of nucleotide vectors, a reference $F = f_1...f_l$ and a read $R = r_1...r_n$, respectively representing the reference sequence of length l and a read of length n aligned onto it. The vectors f_x, where $1 \leq x \leq l$, and r_y, where $1 \leq y \leq n$, represent the probability vectors of each nucleotide at position x in the reference and position y in the read, respectively. Through a statistical learning process and vector quantization, the reference sequence vectors are updated according to the sequencing read nucleotides. Ultimately, the goal is to reconstruct, i.e. assemble, the original sequence S which the reads come from.

A probability vector r_y is calculated according to the quality scores of each base at position y in the read, with equal probability given to the alternative bases. More precisely, if the base b was called with calling error probability q at position y, $r_{yb} = 1 - q$ and $r_{yb'} = q/3$ for b' in $\{1..4\} \setminus \{b\}$. At initialisation, all f_x are only made of binary vectors defined by the reference sequence. Additionally, a "persistence" vector $P = p_1...p_l$, where p_i for $1 \leq i \leq l$ are initialised all to 1, is updated when indels occur for each nucleotide position in the reference. The distance between a pair of nucleotide vectors is defined as

$$\begin{aligned} d(f_x, r_y) &= d([f_{x1}, f_{x2}, f_{x3}, f_{x4}], [r_{y1}, r_{y2}, r_{y3}, r_{y4}]) \\ &= |f_{xi} - r_{yi}| \quad for \quad i = argmax_j([r_{yj}]), \quad j = 1...4. \end{aligned}$$

Therefore, only the nucleotide with the highest probability in the read is taken into account. A dynamic programming approach is used to align the reads to the reference sequence. Let $M(x, y)$ the minimum edit distance over all possible suffixes of the reference from position 1 to x and the read from position 1 to y.

$$M(x,0) = 0 \quad for \quad 0 \leq x \leq l$$

$$M(0,y) = c * y \quad for \quad 1 \leq y \leq n$$

$$M(x,y) = \min \begin{cases} M(x-1, y-1) + d(f_{x-1}, r_{y-1}) \\ M(x-1, y) + c & for \quad 1 \leq x \leq l \quad and \quad 1 \leq y \leq n, \\ M(x, y-1) + c \end{cases}$$

with the insertion/deletion cost is $c = 1$. The three elements correspond to three edit operations: insertion, deletion and substitution. The value in $e_{FR} = min_{1 \leq x \leq l} M(x, n)$ therefore consists in an edit distance between the read and the reference vector sequences of nucleotide vectors. It is then normalised by the length of the read to obtain a read "edit rate", $\hat{e}_{FR}$.

The optimal path is traced back and, at each position, the new reference vector is updated. In case of a substitution, $f_x = w * f_x + (1 - w) r_y$ with a learning rate w (see below). In cases of deletions or insertions, the f_x remains unchanged but the corresponding position in the persistence vector decreases or increases by an amount equal to $(1 - w)$, respectively. Then, the persistence value is assessed against a threshold: if $p_x > 1 + w$ or $p_x < 1 - w$, then an insertion or a deletion is performed at the position x in the reference sequence. For insertions, the inserted nucleotide vector is initialised to the same value r_y which is the nucleotide probability vector on the position y of the read r aligned to the inserted position in the reference. All the reads are chosen in random order and sequentially aligned to the reference sequence according to this procedure (Fig. 5).

Learning rate

The learning rate $(1 - w)$ is set to depend on the edit rate and governs how much the reference is updated. For low values of $(1-w)$ the reference mostly remains unmodified. When the distance between the read and the reference is low, there is high certainty in the positioning of the read onto the reference. Therefore, the learning rate can be increased to facilitate the update of the reference toward the sequence of the read. On the other hand, when the

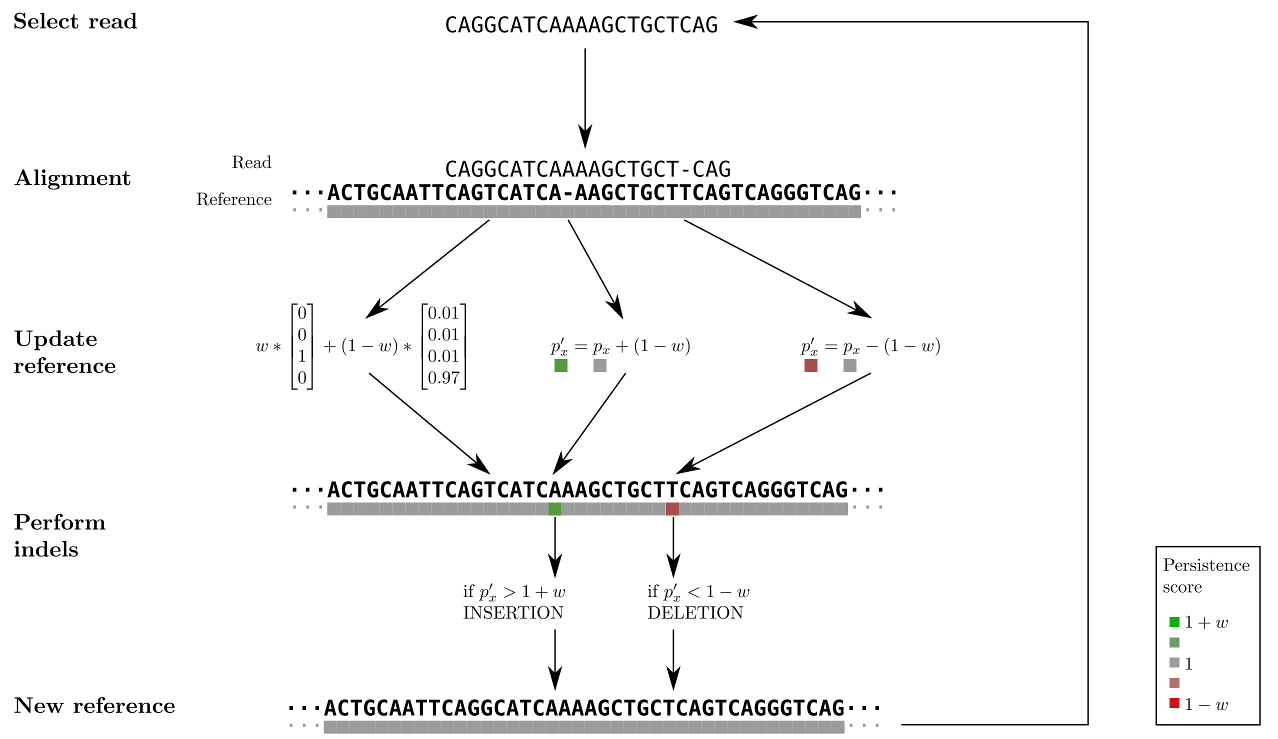
Select read
CAGGCATCAAAAGCTGCTCAG
Alignment
Read
Reference
CAGGCATCAAAAGCTGCT-CAG
···ACTGCAATTCAGTCATCA-AAGCTGCTTCAGTCAGGGTCAG···
Update reference
$w * \begin{bmatrix}0\\0\\1\\0\end{bmatrix} + (1-w) * \begin{bmatrix}0.01\\0.01\\0.01\\0.97\end{bmatrix}$
$p'_x = p_x + (1-w)$
$p'_x = p_x - (1-w)$
Perform indels
···ACTGCAATTCAGTCATCAAAGCTGCTTCAGTCAGGGTCAG···
if $p'_x > 1 + w$
INSERTION
if $p'_x < 1 - w$
DELETION
New reference
···ACTGCAATTCAGGCATCAAAAGCTGCTCAGTCAGGGTCAG···
Persistence score
$1 + w$
1
$1 - w$

Fig. 5 Overview of the algorithm. Reads are taken in random order and iteratively aligned to the reference. After each alignment, the reference sequence is updated according to the learning rate *w*, which is proportional to the normalised edit distance between the read and the reference. In this case, there is one substitution between the reference of the read; the read has a *G* with Phred quality score of 15 while the reference is *T*. One deletion and one insertion are treated thanks to a persistence vector. The persistence value $p_\bullet$ indicates the tendency of a base to be inserted or deleted at each position in the reference. This value can trigger indels update in the reference when it goes beyond a threshold

alignment of the read is more difficult, i.e. high edit distance, the learning rate is set to a low value so that the reference is only slightly updated and misalignments or errors in the read sequence are not affecting the learning process.

Computer simulations were conducted in order to determine the distribution of the edit distances between reads and increasingly divergent reference sequences. First, a nucleotide sequence of length $\mathcal{U}(500, 5000)$ was generated by randomly choosing nucleotides with 50% GC content. A read sequence of length 150 was generated by randomly choosing a position in the original sequence and using an error rate of 1% with the errors uniformly distributed along the sequence. Then, mutations were introduced in the original sequence, at a rate of $\{1, 5, 10, 30, 50\}$%, and single nucleotide indels were introduced at a rate of 10%. Additionally, random reference sequences of similar length were generated to build a random distribution of the distance. The process was repeated 1,000 times (Fig. 6).

From the empirical distributions of the distance (Fig. 6), the learning rate was determined to be equal to 0.95 when the distance is below 0.05, which corresponds to the range of distances expected due to sequencing errors. It is set

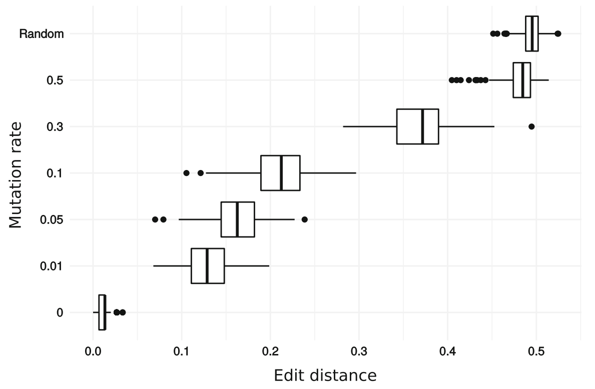

Fig. 6 Distribution of the normalised edit distance between reads and increasingly distant reference sequences. The mutation rate of the reference sequence is indicated on the y-axis. The top row (Random) shows the distribution of the edit distance when reads were aligned to randomly generated nucleotide sequences. For the lowest row, the reads were aligned to their original sequence and the departure from 0 of the edit distance only results from the simulated sequencing errors

to 0.05 when the distance is above 0.35, i.e. the distance expected when the read and the reference sequence have less than 70% sequence similarity. Between normalised edit distances of 0.05 and 0.95, the rate was set to linearly increase, i.e. $w = 3 \times \frac{\hat{e}_{FR}}{n} - 0.1$.

Five assembly pipelines

First, the whole set of reads, average coverage of $\sim 2000\times$, was mapped to the eastern-grey kangaroo to determine the western-grey kangaroo mitochondrial sequence for the amplicon (see [11] for details). Then, five different bioinformatic pipelines were tested at lower coverage. At first, the reads were preprocessed before running each pipeline: Illumina adapters and low quality bases were removed (Trimmomatic version 0.36, [32]) using a sliding window of 15 nucleotides, with steps of four bases and the resulting reads below length 36 were discarded. Additionally, kmer error correction was performed using Tadpole (BBMap version 37.95, Brian Bushnell). The five assembly pipelines (Fig. 7) are described below:

1. **Mapping** was performed using Bowtie2 version 2.2.6 [33]. Both "local" alignment with "soft trimmed" and "end-to-end" alignment of the reads were tested. In general, local alignment resulted in higher alignment rates and was therefore used in all simulations. Once the reads were aligned to the reference, Samtools version 1.5 [34] was used to order the reads. Freebayes version 1.1.0 [35] then allowed us to identify variants. Calls with high probability to be false positive, Phred score < 20, were removed with Vcffilter (Vcflib version 1.0.0) [36]. The consensus sequence was generated using Bcftools version 1.6 [34] by applying the alternative variants to the reference sequence. Finally, the uncovered parts at the beginning and at the end of the reference were removed.
2. **Learning** consisted in iteratively aligning the reads and dynamically updating the reference according to the machine learning approach previously described, the algorithm is implemented in Nucleoveq [10]. For these simulations, all the reads were aligned to the reference and no prior information about the mapping position was utilised to perform read alignments. At the end of the learning process, the uncovered regions located at the beginning and end of the reference were truncated to generate the final assembly.
3. **De novo assembly** was done with Trinity version 2.4.0 [37], using a kmer size of 17 and setting the minimum contig length to 100 so that assembly could be performed when coverage was very low. After assembly, the longest contig was selected for evaluation.

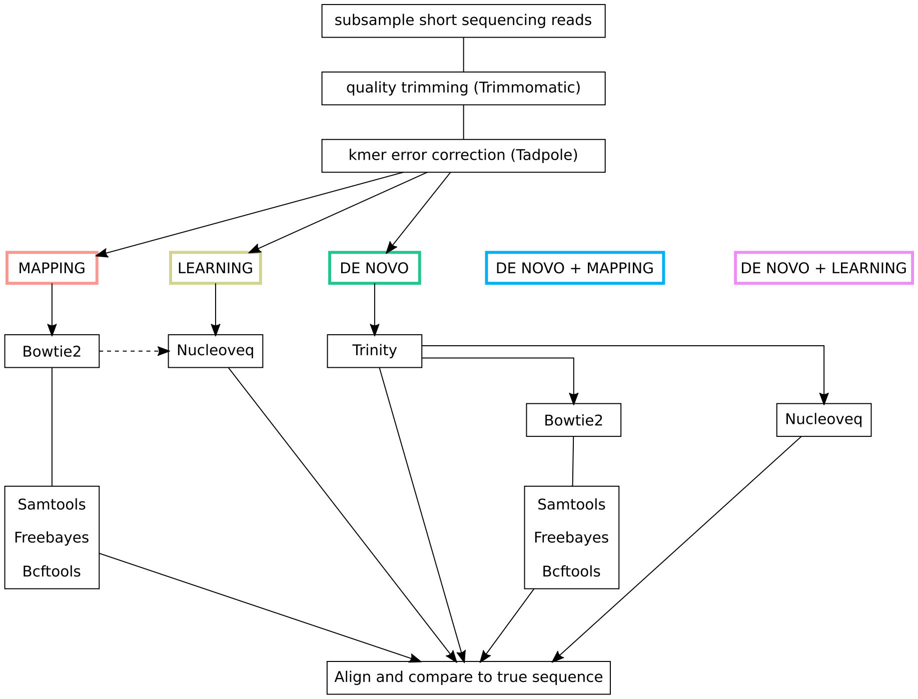

Fig. 7 Five bioinformatic pipelines for assembly. Dashed-line: it is possible to pass a priori mapping position of the reads to Nucleoveq to decrease memory requirements and speed up computation (option not used in the reported comparisons)

4 **De novo + Mapping** consisted in mapping all the **de novo assembly** contigs obtained from Trinity to the reference in an effort to connect them into a longer sequence. The same approach as for **mapping** pipeline was used to generate the consensus.
5 **De novo + Learning** consisted in feeding all the **de novo assembly** contigs obtained from Trinity to our machine learning algorithm. The same steps as for the above **learning** pipeline were performed while regarding the contigs instead of the reads as input.

Abbreviations

DNA: DeoxyriboNucleic acid; indels: insertions and deletions

Acknowledgements

We thank the anonymous reviewers for their constructive comments and ideas, which helped to improve the manuscript.

Authors' contributions

LR conceived the ideas; LR, TKFW and AGR designed methodology; LR wrote the code, analysed the data and led the writing of the manuscript. All authors contributed critically to the drafts and gave final approval for publication.

Funding

This research was funded by an Australian Research Council Discovery Project Grant #DP160103474. The funding body had no role in the design of the study and collection, analysis, and interpretation of data and in writing the manuscript.

Availability of data and materials

Software Nucleoveq is freely available at https://github.com/LouisRanjard/nucleoveq. Sequencing reads are available on Sequence Read Archive (SRA: SRP121381, BioProject: PRJNA415669).

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Received: 3 May 2019 Accepted: 20 November 2019
Published online: 11 December 2019

References

1. Miller JR, Koren S, Sutton G. Assembly algorithms for next-generation sequencing data. Genomics. 2010;95(6):315–27.
2. Rausch T, Koren S, Denisov G, Weese D, Emde A-K, Döring A, Reinert K. A consistency-based consensus algorithm for de novo and reference-guided sequence assembly of short reads,. Bioinforma (Oxford, England). 2009;25(9):1118–24.
3. Lischer HEL, Shimizu KK. Reference-guided de novo assembly approach improves genome reconstruction for related species. BMC Bioinformatics. 2017;18(1):474.
4. Otto TD, Sanders M, Berriman M, Newbold C. Iterative Correction of Reference Nucleotides (iCORN) using second generation sequencing technology,. Bioinforma (Oxford, England). 2010;26(14):1704–7.
5. Tsai IJ, Otto TD, Berriman M. Improving draft assemblies by iterative mapping and assembly of short reads to eliminate gaps,. Genome Biol. 2010;11(4):41.
6. Dutilh BE, Huynen MA, Gloerich J, Strous M. Iterative Read Mapping and Assembly Allows the Use of a More Distant Reference in Metagenome Assembly. In: Handbook of Molecular Microbial Ecology I. Hoboken: John Wiley & Sons, Inc.; 2011. p. 379–85.
7. Ghanayim A. Iterative referencing for improving the interpretation of dna sequence data. Technical Report CS-2013-05, Technion, Computer Science Department. 2013. http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2013/CS/CS-2013-05.pdf.
8. Hahn C, Bachmann L, Chevreux B. Reconstructing mitochondrial genomes directly from genomic next-generation sequencing reads–a baiting and iterative mapping approach. Nucleic Acids Res. 2013;41(13): 129.
9. Břinda K, Boeva V, Kucherov G. Dynamic read mapping and online consensus calling for better variant detection. arXiv. 20161–21.
10. Ranjard L. Nucleoveq. GitHub. 2018. https://github.com/LouisRanjard/nucleoveq.
11. Ranjard L, Wong TKF, Rodrigo AG. Reassembling haplotypes in a mixture of pooled amplicons when the relative concentrations are known: A proof-of-concept study on the efficient design of next generation sequencing strategies. PLoS ONE. 2018;13(4):0195090.
12. Wong TKF, Ranjard L, Lin Y, Rodrigo AG. HaploJuice : Accurate haplotype assembly from a pool of sequences with known relative concentrations. bioRxiv. 2018307025.
13. Kearse M, Moir R, Wilson A, Stones-Havas S, Cheung M, Sturrock S, Buxton S, Cooper A, Markowitz S, Duran C, Thierer T, Ashton B, Meintjes P, Drummond A. Geneious Basic: An integrated and extendable desktop software platform for the organization and analysis of sequence data. Bioinformatics. 2012;28(12):1647–9.
14. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ. Basic local alignment search tool. J Mol Biol. 1990;215(3):403–10.
15. Liao Y, Smyth GK, Shi W. The Subread aligner: fast, accurate and scalable read mapping by seed-and-vote. Nucleic Acids Res. 2013;41(10):108.
16. Břinda K, Sykulski M, Kucherov G. Spaced seeds improve <i>k</i> -mer-based metagenomic classification. Bioinformatics. 2015;31(22): 3584–92.
17. Břinda K, Boeva V, Kucherov G. Ococo: an online consensus caller. arXiv preprint. 2017;1712.01146. 2017.
18. Ranjard L, Withers SJ, Brunton DH, Ross HA, Parsons S. Integration over song classification replicates: Song variant analysis in the hihi. J Acoust Soc Am. 2015;137(5):2542–51.
19. Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint. 2016;1609.04747. 2016.
20. Smith LN. Cyclical Learning Rates for Training Neural Networks. arXiv preprint. 2015;1506.01186. 2015.
21. Ranjard L, Wong TKF, Kulheim C, Rodrigo AG, Ragg NLC, Patel S, Dunphy BJ. Complete mitochondrial genome of the green-lipped mussel, Perna canaliculus (Mollusca: Mytiloidea), from long nanopore sequencing reads. Mitochondrial DNA Part B. 2018;3(1):175–6.
22. Ranjard L, Ross HA. Unsupervised bird song syllable classification using evolving neural networks. J Acoust Soc Am. 2008;123(6):4358–68.
23. Ranjard L, Withers SJ, Brunton DH, Parsons S, Ross HA. Geographic patterns of song variation reveal timing of song acquisition in a wild avian population. Behav Ecol. 2017;28(4):1085–92.
24. Voss RF. Evolution of long-range fractal correlations and 1/ <i>f</i> noise in DNA base sequences. Phys Rev Lett. 1992;68(25):3805–8.
25. Mendizabal-Ruiz G, Román-Godínez I, Torres-Ramos S, Salido-Ruiz RA, Morales JA. On DNA numerical representations for genomic similarity computation. PLoS ONE. 2017;12(3):0173288.
26. Mendizabal-Ruiz G, Román-Godínez I, Torres-Ramos S, Salido-Ruiz RA, Vélez-Pérez H, Morales JA. Genomic signal processing for DNA sequence clustering. PeerJ. 2018;6:4264.
27. Olshen AB, Cosman PC, Rodrigo AG, Bickel PJ, Olshen RA. Vector quantization of amino acids: Analysis of the HIV V3 loop region. J Stat Plan Infer. 2005;130(1-2):277–98.
28. Legrand B, Chang CS, Ong SH, Neo S-Y, Palanisamy N. Chromosome classification using dynamic time warping. Pattern Recogn Lett. 2008;29(3):215–22.
29. Skutkova H, Vitek M, Babula P, Kizek R, Provaznik I. Classification of genomic signals using dynamic time warping. BMC Bioinformatics. 2013;14(Suppl 10):1.
30. Skutkova H, Vitek M, Sedlar K, Provaznik I. Progressive alignment of genomic signals by multiple dynamic time warping. J Theor Biol. 2015;385:20–30.
31. Loose M, Malla S, Stout M. Real-time selective sequencing using nanopore technology. Nat Methods. 2016;13(9):751–4.
32. Bolger AM, Lohse M, Usadel B. Trimmomatic: a flexible trimmer for Illumina sequence data. Bioinformatics. 2014;30(15):2114–20.

33. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. Nat Methods. 2012;9(4):357–9.
34. Li H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. Bioinformatics. 2011;27(21):2987–93.
35. Garrison E, Marth G. Haplotype-based variant detection from short-read sequencing. 2012.
36. Garrison E. a simple C++ library for parsing and manipulating VCF files. Github. 2016. https://github.com/vcflib/vcflib.
37. Grabherr MG, Haas BJ, Yassour M, Levin JZ, Thompson DA, Amit I, Adiconis X, Fan L, Raychowdhury R, Zeng Q, Chen Z, Mauceli E, Hacohen N, Gnirke A, Rhind N, di Palma F, Birren BW, Nusbaum C, Lindblad-Toh K, Friedman N, Regev A. Full-length transcriptome assembly from RNA-Seq data without a reference genome. Nat Biotechnol. 2011;29(7):644–52.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.