Reference-free reconstruction and quantification of transcriptomes from long-read sequencing

Ivan de la Rubia[1,2], Joel A. Indi[1,3], Silvia Carbonell[2,4], Julien Lagarde[2,4], M Mar Albà[2,5,6], Eduardo Eyras[1,6,7]

[1]EMBL Australia Partner Laboratory Network at the Australian National University, Acton ACT 2601, Canberra, Australia

[2]Pompeu Fabra University, E08003 Barcelona, Spain.

[3]Universidade de Lisboa, Lisboa, Portugal

[4]CRG, E08001 Barcelona, Spain

[5]ICREA, E08010 Barcelona, Spain

[6]IMIM, E08001 Barcelona, Spain

[7]Australian National University, Acton ACT 2601, Canberra, Australia

Correspondence to: eduardo.eyras@anu.edu.au

Abstract

Single-molecule long-read sequencing provides an unprecedented opportunity to measure the transcriptome from any sample [1–3]. However, current methods for the analysis of transcriptomes from long reads rely on the comparison with a genome or transcriptome reference [2,4,5], or use multiple sequencing technologies [6,7]. These approaches preclude the cost-effective study of species with no reference available, and the discovery of new genes and transcripts in individuals underrepresented in the reference. Methods for the assembly of DNA long-reads [8–10] cannot be directly transferred to transcriptomes since their consensus sequences lack the interpretability as genes with multiple transcript isoforms. To address these challenges, we have developed RATTLE, the first method for the reference-free reconstruction and quantification of transcripts from long reads. Using simulated data, transcript isoform spike-ins, and sequencing data from human and mouse tissues, we demonstrate that RATTLE accurately performs read clustering and error-correction. Furthermore, RATTLE predicts transcript sequences and their abundances with accuracy comparable to reference-based methods. RATTLE enables rapid and cost-effective long-read transcriptomics in any sample and any species, without the need of a genome or annotation reference and without using additional technologies.

RATTLE starts by building read clusters that represent potential genes. To circumvent the quadratic complexity of an all-vs-all comparison of reads, RATTLE performs a deterministic greedy clustering using a two-step k-mer based similarity measure (Fig. 1a) (Methods). The first step consists of a fast comparison of the common k-mers between two reads (Supp. Fig. 1a), whereas the second step is based on the Longest Increasing Subsequence (LIS) of co-linear matching k-mers between a pair of reads to define the RATTLE similarity score (Supp. Fig. 1b) (Methods). Clusters are generated greedily by comparing reads to a representative of each existing cluster at every step of the iteration. This generates clusters that represent potential genes with reads from all transcript isoforms.

Gene-clusters are subsequently split into sub-clusters representing transcripts. Transcript-clusters are built by determining for each pair of reads in a cluster whether they are more likely to originate from different transcript isoforms rather than from the same isoform according to the relative size of the gaps found between co-linear matching k-mers (Supp. Fig. 1c). RATTLE then performs error correction within each of these transcript-clusters by generating a multiple sequence alignment (MSA) (Fig. 1a) (Methods). Each read is assessed for error correction taking into account the error probability for the base and the average error probability for the consensus at that column of the MSA. RATTLE then builds the final transcripts after a polishing step to refine the cluster definitions (Methods). The transcript sequence is the consensus of the transcript-cluster MSA and the abundance is calculated as the total read count of the transcript cluster (Fig. 1a).

To evaluate the strength of RATTLE similarity score to perform read clustering, we simulated reads from different transcripts with DeepSimulator [11], taking into account the read length distribution observed in a Nanopore cDNA sequencing run (Supp. Fig. 1d) (Methods). RATTLE similarity score separates better reads originating from the same transcript than using a minimizer-based score (Fig. 1b). To test the ability of RATTLE to separate reads from two transcript isoforms, we considered the reads simulated from two transcripts that differ from each other by an internal exon of 154nt. The number of common bases cannot differentiate reads coming from the same or from different transcripts, since one transcript is an exact substring of the other one, whereas the variance used by RATTLE can separate them (Fig. 1c) (Supp. Fig. 1e).

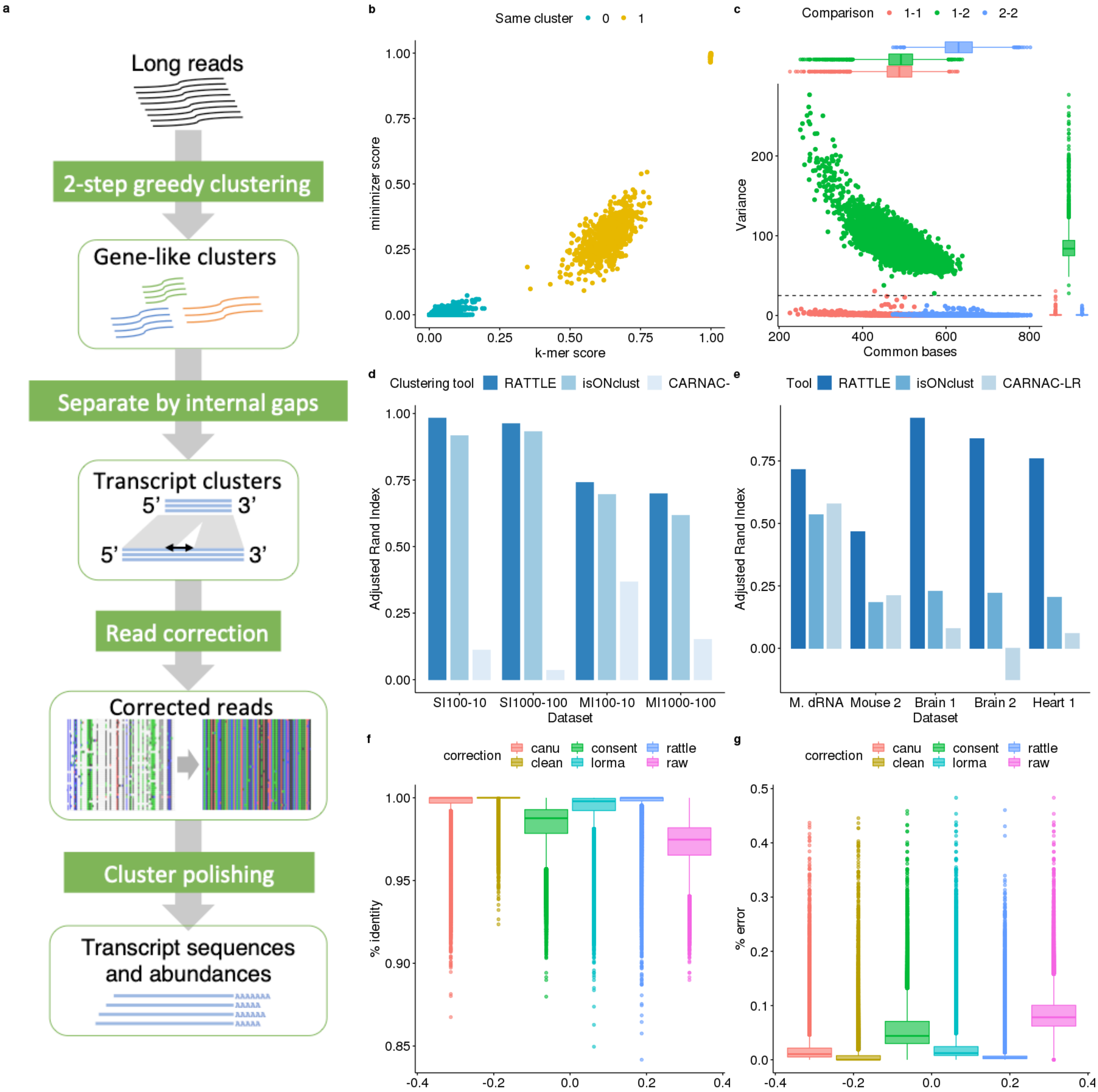

Figure 1. (a) Illustration of RATTLE workflow. **(b)** Comparison of the RATTLE similarity score (x axis), based on the longest increasing subsequence, and a similarity score based on minimizers (y axis), using k=6. Each dot represents a comparison between two simulated reads belonging to the same (orange) or different (blue) clusters. **(c)** The plot shows the number of common bases (x-axis) and variance in the distribution of gap-length differences in adjacent matching k-mers (y axis) from the comparison of reads simulated from two transcripts, indicating when the reads originate from the same transcript (1-1, 2-2) or from different transcript (1-2). **(d)** Clustering accuracy using simulated reads for RATTLE, CARNAC and isONclust in terms of the adjusted rand index (y axis). Simulations (x-axis) were performed with a single (SI) or multiple (MI) isoforms per gene, and using different number of reads (*y*) and different number of transcripts per gene (*x*), indicated as SI*x*-*y* or MI*x*-*y*. Other accuracy metrics are provided in Supp. Table S1. **(e)** Clustering accuracy using Spike-in RNA Variant (SIRV) genes as reference. The plot

shows the adjusted rand index (y axis) for RATTLE, CARNAC and isONclust for each sample (x axis): Nanopore direct RNA sequencing from mouse brain tissue (M. dRNA) and cDNA sequencing from human (Brain1, Brain2, Heart) and mouse (Mouse 2) tissues (Methods). **(f)** Percentage identity distributions of SIRV reads before (raw) and after correcting with RATTLE, CONSENT, Lorma, Canu and TranscriptClean (clean) for Nanopore cDNA-seq data from human Brain tissue (sample Brain 1). Percentage identity was calculated as the number of correct matches divided by the total length of the aligned region. **(g)** Error rate distribution of SIRV reads before and after correction with RATTLE, CONSENT, Lorma, Canu and TranscriptClean (clean) for the same sample as in (f). Error rate was calculated as the sum of insertions, deletions and substitutions divided by the length of the read. Other samples are shown in Supplementary Figure 2.

To test the accuracy in the identification of gene-clusters, we compared RATTLE with two other methods for the clustering of long reads, CARNAC [12] and isONclust [13]. We built several reference datasets of simulated reads from multiple genes with one or more transcripts per gene and for a different number of reads per transcript. RATTLE showed higher accuracy at recovering gene clusters in all comparisons and using different metrics (Fig. 1d) (Supp. Table S1). We further assessed the accuracy of RATTLE at recovering gene-clusters using Lexogen Spike-in RNA Variant Control Mixes (SIRVs) (Methods). The SIRV genome (SIRVome) is organized into 7 different gene loci, each containing several transcript isoforms with known SIRVome coordinates, sequence and abundance, with a total of 69 isoforms. We used the SIRVs in 5 different sequencing experiments with the Oxford Nanopore Technologies (ONT) MinION platform: cDNA sequencing (cDNA-seq) from human brain (two replicates) and heart tissues (Methods), and direct RNA (RNA-seq) and cDNA-seq from mouse brain [14]. We first used the SIRV transcripts aggregated per gene to evaluate the clustering at gene-level (Methods). RATTLE showed higher accuracy in the identification of SIRV genes than the other methods (Fig. 1e).

To test RATTLE accuracy to correct errors in Nanopore reads without using a reference, we next used the same SIRV reads and compared RATTLE results with CONSENT [15], Canu [8], and Lorma [9], which are self-correction methods designed for DNA sequencing reads. Additionally, we considered TranscriptClean [4], a reference-based method to correct transcriptomic long-reads that can operate without using any additional information like annotations or splice-site coordinates. Reads were mapped to the SIRVome before and after correction by each method with Minimap2 [16]. After correction, all methods achieved an increase in the percentage identity to the SIRV isoform sequences (Fig. 1f) (Supp. Fig. 2) and a decrease in the error rate (Fig. 1g) (Supp. Fig. 3). Compared with the other self-correction methods, RATTLE corrected more reads (Supp. Table S2) and showed on average higher percentage identity and lower error rate. These distributions were very similar between RATTLE and TranscriptClean, which used the SIRVome sequence. Notably, RATTLE had much shorter runtimes in all samples, with times ranging 26.95-123.9 minutes (mins) (Supp.

Table S2). The fastest from the other methods was TranscriptClean, which took 43.48-223.00 mins on the same datasets, not taking into account the mapping to the SIRVome, which only took 0.85-3.92 mins (Supp Table S2).

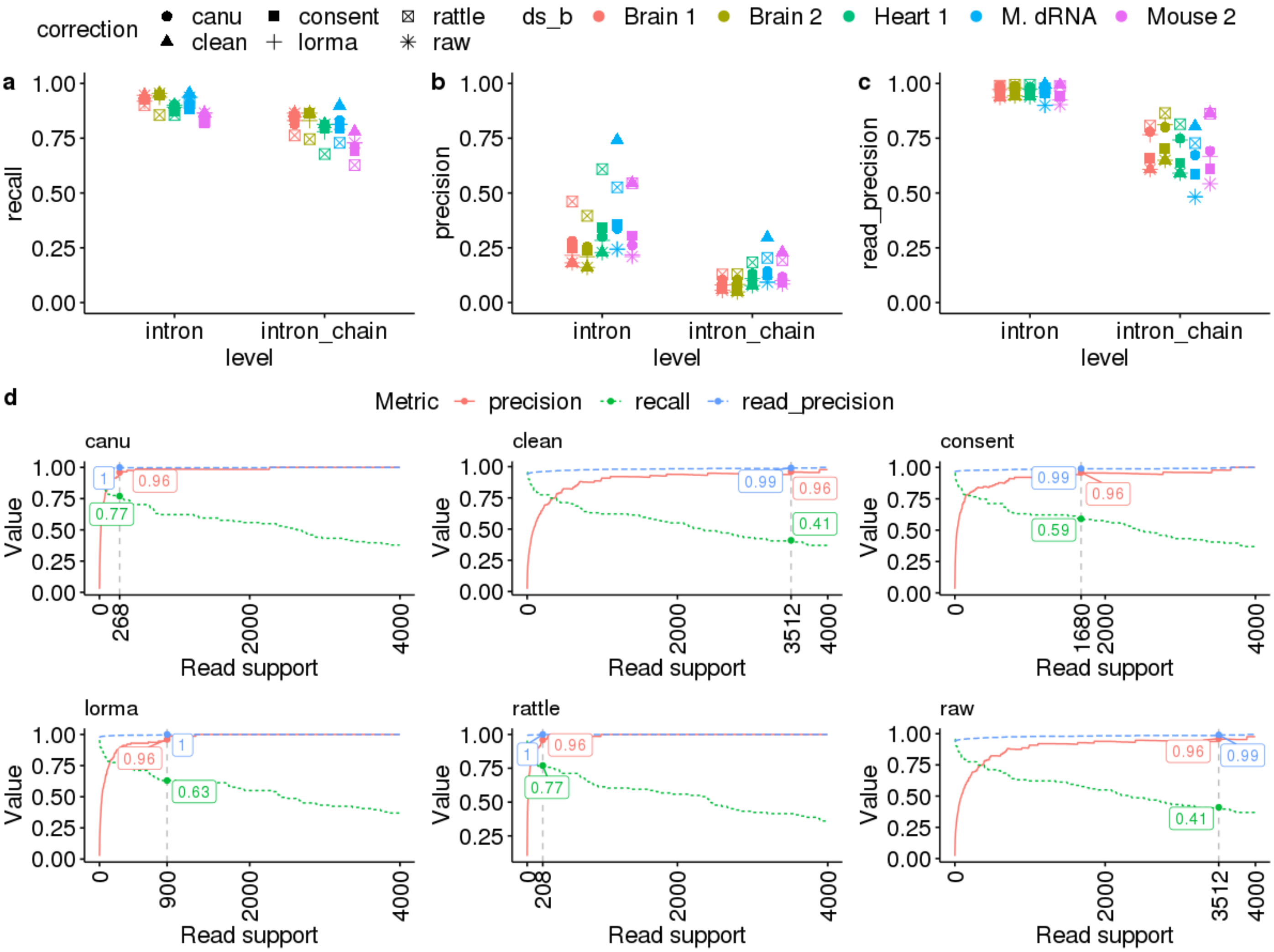

Figure 2. (a) Recall of unique SIRV introns and intron-chains obtained by mapping reads to the SIRVome before (raw) and after correction with RATTLE, CONSENT, Canu, LORMA and TranscriptClean (clean) for all samples tested: two cDNA-seq samples from brain (Brain 1, Brain 2), one cDNA-seq sample from heart (Heart), and one cDNA-seq and one direct RNA (M. dRNA) from mouse from [14]. Recall was calculated as the fraction of unique annotated introns or intron-chains correctly found by each method with 5 or more supporting reads. **(b)** Precision of unique SIRV introns and intron-chains for the same methods and datasets as (a). Precision was calculated as the fraction of unique introns or intron-chains predicted by reads that matched correctly the annotation and had support of 5 or more reads. **(c)** Read-precision for annotated SIRV introns and intron-chains for the same methods and datasets as in (a). Read-precision was calculated as the fraction from the total number of introns (or intron-chains) predicted in reads that corresponded to annotated introns (or intron chains) and had support of 5 of more reads. **(d)** We plot the recall (green), precision (red) and read-precision (blue) of the SIRV introns, as a function of an expression cut-off (x axis)

in terms of the number of reads supporting the introns. We indicate for each case the threshold at which a precision (red) of approximately 0.95 is achieved, and for that threshold we indicate the corresponding recall (green) and read-precision (blue) values. The plot corresponds to the Nanopore cDNA-seq from the human brain 1 sample. The results for other samples are available in Supp. Fig. 5.

We next evaluated the ability to recover exon-intron structures accurately after mapping the corrected reads to the SIRVome. We measured the exact match of the coordinates of specific genomic features: introns and intron-chains (Methods). Strikingly, although RATTLE found in general fewer introns and intron-chains (Fig. 2a), it showed a much higher precision (Fig. 2b), suggesting that other methods produce a large number of false positives. To further test this, we calculated a read-precision, defined as the proportion of correctly identified SIRV features (introns and intron-chains) over the total number of features predicted by all corrected reads, that is, counting all reads supporting each defined genomic feature. RATTLE showed the highest read-precision values, especially for intron-chains (Fig. 2c), indicating that a greater proportion of reads corrected by RATTLE were correct.

Overall all methods improved in read-precision values compared with precision, which suggests that most of the false positives may be lowly expressed. This is further supported by a comparison of the read-support between true positives and false positives (Supp. Fig. 4). Thus, there appears to be a relation between the read support and the accuracy that could help distinguishing correct from incorrect cases. To investigate further the capacity of each method to separate between true and false positives, we calculated the recall, precision, and read-precision for SIRV introns at different cut-offs of the read-support. All correction methods showed an improvement over the use of raw reads (Fig. 2d) (Supp. Fig. 5). In particular, RATTLE needed lower read-support to achieve a precision of 0.95 and had higher recall at that precision value, compared with other methods (Fig. 2d) (Supp. Fig. 5).

We next tested the capacity of RATTLE to estimate the abundance of predicted transcripts without using any information from the genome or annotation. Using the same SIRV datasets, we compared RATTLE with StringTie2 [5], FLAIR [17], and TALON [18], which use the genome and annotations to predict transcript isoforms and their abundances. Additionally, we considered the approach of mapping raw reads directly to SIRV isoforms with Minimap2 and either assigning reads to SIRVs according to the best match or using NanoCount (https://github.com/a-slide/NanoCount), which assigns reads to isoforms using an expectation-maximization (EM) algorithm. To compare the abundances predicted by RATTLE with the other methods, we assigned each SIRV to the best-matching predicted transcript (Methods). Despite not using any information from the SIRV genome or SIRV annotations, the correlation of RATTLE (Pearson R = 0.76) with SIRV isoform abundances is comparable to those obtained with StringTie2 (R=0.73) and FLAIR (R=0.79), and superior to TALON (R=0.62), for RNA-seq data (Fig. 3a). Minimap2 showed the highest correlation (R=0.89), whereas NanoCount (0.73) was similar to the other reference-based methods (Fig. 3a). The correlation using cDNA-seq reads was generally lower for all methods (Fig. 3b). RATTLE showed in this case a higher correlation than the reference-based methods. Minimap2 and NanoCount showed similar correlation values for cDNA-seq and comparable to other reference-based methods (Fig. 3b). We next assessed whether RATTLE final transcript predictions would accurately define intron coordinates if mapped to the reference. We thus mapped the predicted transcripts from the SIRV datasets to the SIRVome and calculated the precision and recall for the recovery of annotated introns, at different thresholds of transcript abundance (Fig. 3c). For all datasets tested, RATTLE maintained high precision (>0.75) with increasing threshold values (direction left to right in Fig. 3c), and achieved a maximum of approximately 0.75 recall at this precision, confirming that RATTLE predictions attain high precision at different expression levels.

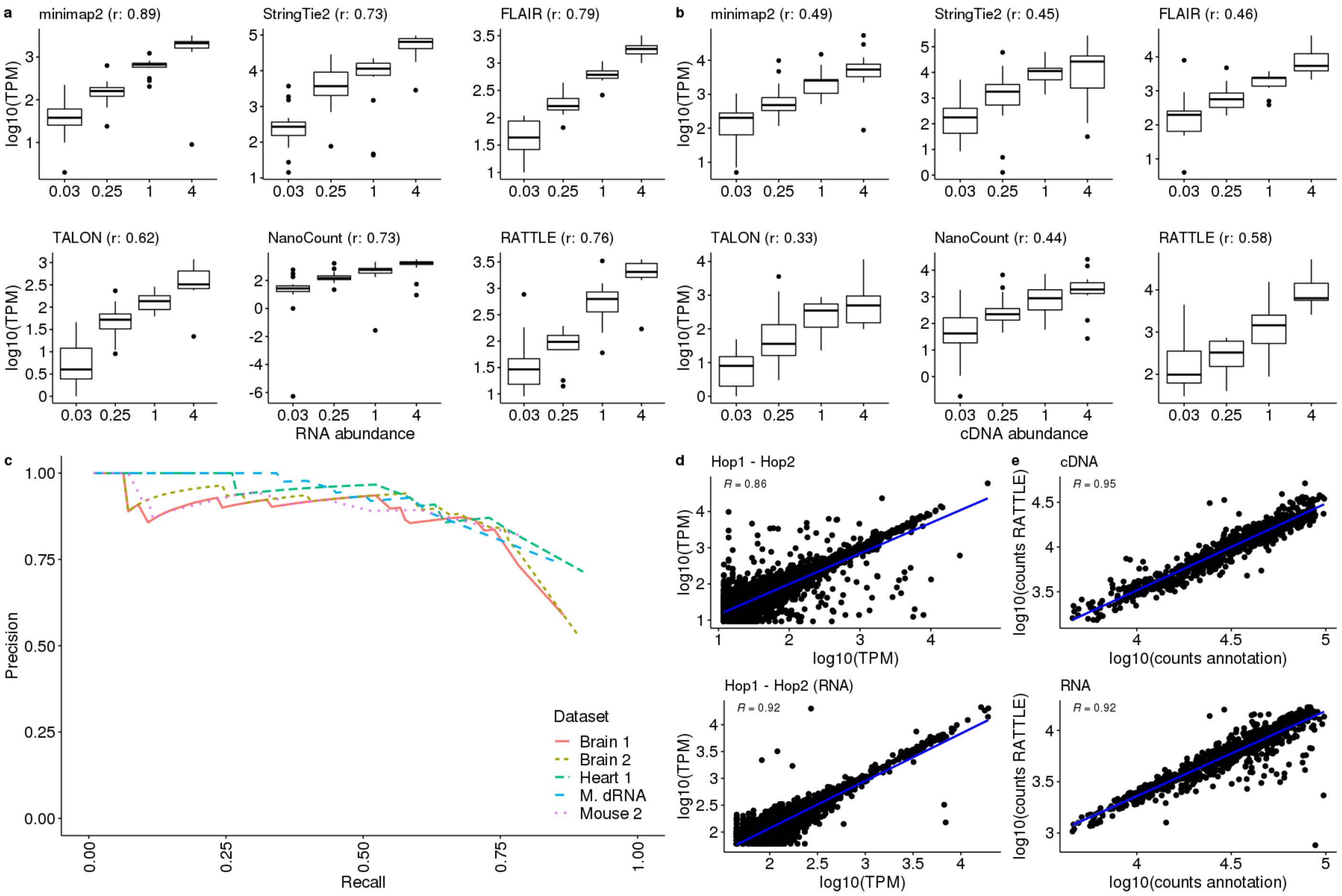

Figure 3. (a) Comparison of the predicted transcript abundances (y axis) by RATTLE, FLAIR, StringTie2, TALON, NanoCount, and selecting the best match from minimap2, with abundances of the SIRV transcript isoforms (x axis). In each panel we show

the Pearson correlation R from the comparison of the abundance values for that method. Units on the y-axis vary according method: RATTLE provides abundances in terms of read counts per million, similar to TALON and FLAIR. StringTie2 produces a TPM value using the same formula as for short reads. For NanoCount and best-match mapping with minimap2, we give read counts per million. Data corresponds to the Nanopore RNA-seq of mouse brain [14]. **(b)** Comparison of predicted transcript abundances with abundances of the SIRV transcript isoforms as in (a) using Nanopore cDNA-seq of mouse brain [14]. **(c)** Precision-recall curve for the prediction of annotated SIRV introns by RATTLE transcripts. **(d)** Correlation of the transcript abundances calculated with RATTLE between two cDNA-seq (upper panel) and between two RNA-seq (lower panel) replicates from a human cell line (Johns Hopkins samples from the Nanopore sequencing consortium [2]) for transcripts that are expressed in both replicates according to RATTLE. **(e)** Correlation between 5-mer frequencies in the annotation and in the transcript sequences predicted by RATTLE from cDNA-seq (upper panel) and RNA-seq (lower panel) for the same samples as in (d).

To establish the robustness of the transcript predictions performed by RATTLE, we analyzed two replicates of cDNA and RNA sequencing with MinION from a human cell line from the Nanopore sequencing consortium (Workman et al., 2018) (Methods). RATTLE identified 8951 and 11468, and 3370 and 2795 transcripts in the RNA replicates. To identify the annotations they correspond to, we mapped the predicted transcripts to the human transcriptome annotation with Minimap2 (Methods). We recovered 7309 transcripts in at least one of the cDNA replicates (4005 in common), and 3651 in at least one of the two RNA replicates (1878 in common). The transcripts predicted by RATTLE from cDNA-seq in common between the replicates had a high correlation of their abundance (4005 pairs, Pearson R=0.87, p-value<2e-16) (Fig. 3d, upper panel). Similarly, for RNA-seq we also found a high correlation of the RATTLE predicted abundances (1878 pairs, Pearson R=0.91, p-value<2e-16) (Fig. 3d, lower panel). As a comparison, we mapped reads to the same transcript annotation with Minimap2 and considered transcripts with more than 5 reads mapped, which was the threshold used to build RATTLE transcripts (Methods). This recovered 10086 transcripts with more than 5 reads in at least one of the cDNA replicates (7568 in common), and 3534 transcripts with more than 5 reads in both RNA replicates together (2197 in common). Finally, to determine whether the RATTLE transcripts resemble the annotation, we compared the 5-mer frequencies observed in all predicted transcripts with those observed in the human transcriptome annotation, using those transcripts with more than 5 reads mapped to them. RATTLE transcripts showed a significant correlation in their 5-mer content with the annotation with using cDNA (Pearson R=0.95, p-value<2e-16) (Fig. 3e, upper panel) of RNA (Pearson R=0.92, p-value<2e-16) sequencing data, providing further support for the accurate reconstruction of reference-free transcriptomes with RATTLE.

In summary, RATTLE is the first method to build transcripts and estimate their abundance from single-molecule long-reads without the use of a reference genome or annotation, and without the use of additional technologies. Our analyses indicate that error-correction impacts the ability of RATTLE and other methods to identify lowly expressed molecules. However, as we have shown, it is an essential step to achieve high precision. In particular, RATTLE achieves in general higher precision, whereas other methods tested produced a large number of false positives. High precision is crucial when searching for new genes and transcripts, or to annotate a new species. Importantly, RATTLE estimation of transcript abundances achieves accuracy comparable to methods that make full use of the genome sequence and annotation. The accuracy of all methods appears generally higher for direct RNA-seq reads than for cDNA-seq reads, which has been attributed before to read fragmentation due to internal priming [14]. In our analyses, we also observed that a highly abundant SIRV isoform fully included in another also impacted negatively the correlation.

The ability of RATTLE to build and quantify transcripts with high accuracy provides an unprecedented opportunity to perform cost-effective study of the transcriptomes from non-model organisms and samples without genome or annotation reference available, and without using additional technologies [19]. Furthermore, the flexibility of RATTLE to parameterize the modular steps, and the structured output with information about transcripts and gene clusters, as well as the reads per cluster, should prove valuable in downstream applications, including the study of differential transcript usage [20], the identification of transcript sequence polymorphisms between individuals [21], and the analysis of single-cell long-read sequencing [22]. RATTLE closes the existing technological gaps to enable the reconstruction and interpretation of transcriptomes with single-molecule long-read sequencing from any sample and any species.

Methods

Software availability

RATTLE is written in C++ and is available at https://github.com/comprna/RATTLE under the MIT license.

RATTLE clustering algorithm

Reads are pre-processed with porechop (https://github.com/rrwick/Porechop). They are also filtered to be longer than 150nt. RATTLE then sorts reads in descending order by their length and processes one at the time in that order. In the first iteration, RATTLE selects the first unclustered read and forms a new cluster with it. All the other unclustered reads are then compared against this read and assigned greedily to this new cluster if the scores resulting from the comparison are above certain thresholds. The process is repeated until there are no more reads left unclustered.

To circumvent the quadratic time complexity of an all-vs-all comparison, RATTLE performs a two-step similarity calculation to achieve both fast and sensitive comparisons. To reduce memory usage and for efficient calculation, sequence k-mers in reads are hashed to 32-bit integers with the hashing function H(A)=0, H(C)=1, H(G)=2, H(T)=3, such that for any k-mer $s=b_1 \ldots b_k$, $H(s) = 4^{k-1}H(b_1) + 4^{k-2}H(b_2) + \ldots + H(b_k)$. All k-mers are extracted for each pair of reads and a 4^k-bit vector is created and the positions in the vector of the hashed k-mers in each read are set to 1. An AND operation is then performed between the two vectors to obtain the number of common k-mers. Extraction and hashing of k-mers is performed only once per read in linear time, and the vector operations are performed in constant time. First, a similarity score is calculated as the number of common k-mers shared between two reads divided by the maximum number of k-mers in either read. If this count is above a set threshold, a second similarity calculation is performed. For the second metric, all k-mers from both reads are extracted along with their positions in each read, generating a list of triplets. These triplets are then sorted by the position on the first read and the Longest Increasing Subsequence (LIS) problem is solved with dynamic programming for the position of the k-mers on the second read. This produces the maximum set of common co-linear k-mers between a pair of reads. The similarity value is defined as the number of bases covered by these co-linear common k-mers over the length of the shortest read in the pair. If the orientation for cDNA reads is unknown[19], RATTLE tests both relative orientations for each pair of reads. As a consequence, all reads within a cluster are oriented the same way.

In subsequent iterations, thresholds are decreased and clusters are created or merged as initially. The first cluster is selected, and all other clusters are compared against that cluster, including single-read clusters, i.e. singletons. If they are similar above the set thresholds, a new cluster is formed with the reads from the selected cluster and all the similar clusters. To ensure fast computation, cluster comparisons are performed using a representative read from each cluster, which is defined by the position in the ranking of read lengths within the cluster and can be set as a parameter by the user. In our analyses, we used the read at the position defining the 15% of the ranking. The number of iterations is specified in the command line by setting the initial and final thresholds for the first hashing-based score (default 0.4 to 0.2) and a decreasing step (default 0.05), i.e. default iterations are performed for thresholds 0.4, 0.35, 0.3, 0.25, and 0.2. A final comparison is done using a threshold of 0.0, i.e. all remaining singletons and all cluster representatives are compared to each other using the LIS-based score. The LIS-based score remains fixed over the entire clustering process.

In our analyses, it was required to be 0.2 or larger. Analyses shown here were carried out for k=10. The k-mer length, the initial and final thresholds in the hash-based comparison as well as the decreasing step, and the similarity thresholds in the LIS comparison (bases and variance), can be modified as parameters in the command

RATTLE transcript-cluster identification and error correction

Read clusters produced by the algorithm described above are considered to correspond to genes, i.e. gene-clusters. Reads within each cluster are then separated into subclusters according to whether they are likely to originate from different transcript isoforms to form transcript-clusters. RATTLE takes into account the relative distances between co-linear k-mers calculated from the LIS-based score. Two reads in the same gene-cluster are separated into different transcript-clusters if the distribution of the relative distances between co-linear matching k-mers has a variance greater than a given threshold. That is, if co-linear matching k-mers calculated from the LIS algorithm show relative distances that would be compatible with a different in exon content. Different thresholds were tested and the value 25 was used for the analyses. This value can be modified as input parameter.

RATTLE performs read correction within each transcript-cluster in two steps. First, each cluster with N reads is separated into blocks; each with a number of reads R. Here we used R=200. If R≤N<2R, the cluster is split in half, and if N<R, we took a single block. To avoid length bias, blocks are built in parallel from the reads in the cluster sorted by length: to build K blocks, block 1 will be made from reads 1, K+1, 2K+1, etc., block 2 will be made from reads 2, K+2, 2K+2, etc., … and block K is made from reads K, 2K, 3K, etc. A multiple sequence alignment (MSA) is obtained from each block using SIMD partial order alignment (SPOA) (https://github.com/rvaser/spoa) [23]. A consensus from each column in the MSA is then extracted in the following way: for each read and each base of the read, the base is changed to the consensus if the consensus occurs with at least 60% frequency, but not if the base being assessed has an error probability less or equal to 1/3 times the average for the consensus base. Indels are treated similarly, but without the error constraint. This is only performed using aligned positions, i.e. not considering terminal gaps of each read. The consensi from each block are then realigned with SPOA to obtain a final MSA for the transcript-cluster and an associated consensus is obtained as before. Only transcript-clusters with a minimum number of reads are corrected and taken further for analysis. In our analyses here we used transcript-clusters with more than 5 reads. The frequency of the consensus, error-probability cutoff, minimum number of reads for a transcript cluster, and length of terminal regions can be set up as input parameters.

RATTLE transcript polishing and quantification

To define the final list of transcripts, RATTLE performs a final polishing step of the transcript-cluster definitions. RATTLE uses the same 2-step greedy clustering described above on the transcript-clusters. From each final cluster, an MSA column consensus is calculated, with abundance given by all the reads contained in the final consensus. Additionally, the transcripts are given a gene ID that corresponds to the gene-clusters they belong to. When two transcript clusters are merged, if they were part of the same gene-cluster, the resulting transcript stays in the same gene. If they were part of different genes, the gene with more transcripts absorbs the transcripts from the other gene to become one single gene. RATTLE outputs different files at different stages of its execution. In the clustering step, it can either output gene-clusters or transcript-clusters in binary files. These files can then be used to extract a CSV file containing each read ID and the cluster it belongs to. These same files are also used for the correction step. This step outputs three files, one file with the corrected reads, one with those that are left uncorrected, and one containing the consensus sequence for each cluster from the input (in FASTQ format). Finally, the transcript-cluster polishing step receives as input the consensus sequences from the correction step and outputs a new file in FASTQ format with the final transcriptome, containing in the header of each read the transcript and gene IDs, and its quantification.

Simulated reads and clusters

We developed a wrapper script (available at https://github.com/comprna/) for DeepSimulator [11] to simulate a specified number of reads per transcript considering an read length distribution. The length-distribution was calculated from a human cDNA sequencing sample from the Nanopore consortium [2]. To simulate the read sequences, we used the Gencode transcript annotation (v29), after removing pseudogenes and genes from non-standard chromosomes, and after removing transcripts that showed > 95% percentage identity with other transcripts using CDHIT [24]. We then randomly selected different number of genes and transcripts to simulate reads. We considered genes with one single transcript isoform (SI), or genes with multiple isoforms (MI). For each case, various datasets were simulated using a different number of reads per transcript and different number of transcripts per gene. To determine the accuracy of the clustering we used the adjusted rand index, which is a measure of the similarity between two cluster sets corrected for chance [25]. Additionally, we used homogeneity, completeness and the V-measure [26]. The V-measure is the harmonic mean of the completeness and homogeneity. Homogeneity is maximal when each cluster contains only elements of the same class. Completeness is maximal when all the elements of a single class are in the same cluster. We compared the clusters predicted by each method with the simulated clusters as reference set. We run isONclust [13] (options: --ont --t 12), CARNAL-LR[12] with Minimap2 overlaps (-t 24 -x ava-ont), and RATTLE clustering (clustering (options: -t 24, -k 10, -s 0.20 –v 1000000 –iso-score-threshold 0.30 –iso-kmer-size 11 –iso-max-variance 25 –p 0.15).

MinION sequencing and SIRV reads

Two commercial total RNA samples were used to prepare libraries: brain (Ambion - product num. AM7962; lot num. 1887911) and heart (Ambion - product num. AM7966; lot num. 1866106). Unless otherwise noted, kit-based protocols described below followed the manufacturer's instructions. Regular quality controls using qBIT, nanodrop and Bioanalyzer were performed according to manufacturer's protocols to assess the length and the concentration of the samples. rRNA depletion was performed using Ribo-Zero rRNA Removal Kit Human/Mouse/Rat (Epicentre - Illumina). 12 ug of total RNA from each sample were prepared and divided into 3 aliquots (4 ug of total RNA each). 8ul of a 1:100 dilution (1 ng total) of synthetic controls (E2 mix lot number 001418 from SIRV-set, Lexogen) were added to each total RNA aliquot. Resulting ribosomal depleted RNAs were purified using 1.8X Agencourt RNAClean XP beads (Beckman Coulter). Samples were finally resuspended with 11 ul of RNA-free water and stored at -80ºC. The cDNA was prepared using 50 ng of rRNA depleted RNA. The cDNA synthesis kit (Takara) based on SMART (Switching Mechanism at 5' End of RNA Template) technology coupled with PCR amplification was used to generate high yields of full-length double-stranded cDNA. The sequencing libraries were prepared using 1 ug of full-length double-stranded cDNA following the standard ONT protocol SQK-LSK109 for 1 aliquot of the heart sample (Heart) and 1 aliquot of the brain sample (Brain 1), and SQK-LSK108 for another aliquot of the brain sample (Brain 2). The final libraries were loaded on an R9.4.1 flowcell, and standard ONT scripts available in MinKNOW were used a total of 48 hours run for each flowcell. ONT sequencing data was basecalled using Guppy 2.3.1+9514fbc (options: --qscore_filtering --min_qscore 8 --flowcell FLO-MIN106 --kit <kit> --records_per_fastq 0 --recursive --cpu_threads_per_caller 14 --num_callers 1), where <kit> is SQK-LSK109 for the brain (Brain1) and heart samples, and SQK-LSK108 for the brain sample replicate (Brain2). Additionally, we used data from cDNA (ERR2680377) and direct RNA (ERR2680375) sequencing of mouse brain including the E2 SIRVs[14]. To select reads corresponding to SIRVs, we run porechop (https://github.com/rrwick/Porechop) and mapped the reads to the SIRV genome (SIRVome) with Minimap2 (options: -t 32 -cx splice --splice-flank=no --secondary=no). We used seqtk (https://github.com/lh3/seqtk) to extract reads with a hit on the SIRVome and being at least 150nt in length. These reads were then considered for further analyses.

Clustering accuracy with SIRV reads

We first built SIRV isoform clusters by mapping reads to SIRV isoforms with Minimap2 [16] and selecting for each read the SIRV isoform with the best mapping score. All reads mapped to the same SIRV gene were then considered a cluster. We then clustered reads with RATTLE, CARNAC and isONclust and measured the accuracy of the predicted clusters by comparing with the built SIRV gene clusters using the same metrics as with the simulated data.

Assessment of error correction accuracy

Reads were mapped to the SIRV transcripts with Minimap2 before and after read-correction. Each read was assigned to the best matching transcript according to the mapping score. From the CIGAR string of the SAM output the error rate was calculated as the sum of insertions, deletions and substitutions divided by the length of the read, and the percentage identity as the number of correct matches divided by the total length of the aligned region. We compared RATTLE (options: -t24 –g 0.3 –m 0.3 –s 200) with CONSENT [15] (options: consent-correct --type ONT), Canu [8] (options: minReadLength=200 stopOnLowCoverage=0.10 executiveMemory=16 executiveThreads=24), LORMA [9] (options: -s -n -start 19 -end 61 -step 21 -threads 24 -friends 7 -k 19), and TranscriptClean [4]. TranscriptClean was run using as input the reads mapped with Minimap2 (parameters: -t 12 -cx splice --splice-flank=no --secondary=no), but with no annotation information,

Accuracy analysis of the SIRV annotation features was performed in the following way. Reads before and after correction were mapped to the SIRV genome with Minimap2 with parameters as above. TranscriptClean was applied to the raw reads after mapping to the SIRV genome, without using the SIRV annotations. PAF files from the mapping were compared with the annotation using ssCheck (available at https://github.com/comprna/RATTLE/). We developed ssCheck, as other methods like gffcompare (https://ccb.jhu.edu/software/stringtie/gffcompare.shtml) did not count correctly the matches when multiple copies of the same annotation feature are present in the reads. ssCheck works by comparing annotation features in the mapped reads with the annotation, and calculates the number of unique features as well as the total number of features predicted in the mapped reads. As annotation features, we used introns and intron-chains. An intron-chain was defined as an ordered sequence of introns in an annotated transcript or mapped read. Recall was calculated as the fraction of unique annotated features correctly found, precision was calculated as the fraction of unique features predicted that were in the annotation, and read-precision was calculated as the fraction from the total number of features predicted in corrected reads that corresponded to annotated features. Read-precision is influenced by abundance levels but better reflects the accuracy per read.

Assessment of transcript quantification accuracy

We used FLAIR[17] (options: align, correct, collapse, quantify –tpm), StringTie2[5] (options: -p 12 –L) and TALON[18] (talon_initialize_database, talon, talon_summarize, talon_abundance, talon_create_GTF) with the cDNA and RNA reads mapped to the SIRV genome with Minimap2 (options: -t 12 –ax splice –splice-flank=no –secondary=no, with –MD tag for TALON). These methods perform read correction (FLAIR and TALON) and transcript quantification (FLAIR, StringTie2 and TALON) of annotated and novel transcripts using the mapped reads with the help of the annotation. For the same samples, RATTLE was run for clustering (options: -t 24, -k 10, -s 0.20 –v 1000000 –iso-score-threshold 0.30 –iso-kmer-size 11 –iso-max-variance 25 –p 0.15), read correction (options: -t24 –g 0.3 –m 0.3 –s 200) and transcript polishing (options: -t12). As additional comparison, we mapped raw reads directly to SIRV isoforms with Minimap2 (options: -ax map-ont -t6) and estimated abundances with NanoCount (https://github.com/a-slide/NanoCount), which assigns reads to isoforms with an expectation-maximization (EM) algorithm. We also assigned reads directly to SIRV isforms with Minimap2 (options: -t 12 -cx map-ont --secondary=no). FLAIR, StringTie2 and TALON provides the SIRV isoform ID with the predicted abundance. Sometimes, these methods give twice the same ID with two different abundances and exon-intron structures, likely due to both being equally good approximate matches to the annotation. In these cases, we only considered the prediction with the highest abundance. To assess the accuracy of RATTLE, we matched transcripts predicted by RATTLE to the SIRV isoforms using Minimap2 (-cx map-ont --secondary=no). If more than one transcript matched the same SIRV isoform, we selected the RATTLE transcript with the highest abundance.

Analysis of human transcriptomes

We downloaded data from cDNA and RNA sequencing from the Nanopore consortium [2], for the samples from Johns Hopkins (cDNA rep1 and rep2, RNA rep1 and rep2) and UCSC (cDNA rep1 and rep2, RNA rep1 and rep2). We predicted the transcripts and their abundances with RATTLE for each sample independently. To calculate the correlation of RATTLE abundances between replicates we mapped RATTLE transcript sequences to the transcripts from the annotation (Gencode v29, after removing pseudogenes and genes from non-standard chromosomes, and after removing transcripts with > 95% percentage identity with other transcripts). Abundances of predicted transcripts mapped to the same annotated transcripts were then compared. For comparison, raw reads were mapped to the same annotated transcripts with Minimap2 (-cx map-ont --secondary=no). As we built RATTLE transcripts from transcript-clusters with >5 reads, only transcripts in the annotation with >5 reads mapped were considered.

Acknowledgements

We thank Roderic Guigó for providing access to the SIRV datasets. We acknowledge funding support from the Spanish Government and FEDER with grants BFU2015-65235-P, BIO2017-85364-R and MDM-2014-0370, and by Catalan Government (AGAUR) with grant SGR2017-1020. IdlR was supported by grant PRE2018-083413 from the Spanish Government (MINECO)

Author contributions

EE and IdlR designed the algorithms in RATTLE with inputs from MMA. IdlR prototyped and implemented the algorithms. IdlR and JI carried out benchmarking analyses. SC and JL generated the experimental data. EE and IdlR wrote the paper with inputs from all authors.

References

1. Garalde, D. R. *et al.* Highly parallel direct RNA sequencing on an array of nanopores. *Nat. Methods* **15**, 201–206 (2018).
2. Workman, R. E. *et al.* Nanopore native RNA sequencing of a human poly(A) transcriptome. *Nat. Methods* **16**, 1297–1305 (2019).
3. Eid, J. *et al.* Real-time DNA sequencing from single polymerase molecules. *Science* **323**, 133–8 (2009).
4. Wyman, D. & Mortazavi, A. TranscriptClean: variant-aware correction of indels, mismatches and splice junctions in long-read transcripts. *Bioinformatics* **35**, 340–342 (2019).
5. Kovaka, S. *et al.* Transcriptome assembly from long-read RNA-seq alignments with StringTie2.

Genome Biol. **20**, 278 (2019).

6. Koren, S. *et al.* Hybrid error correction and de novo assembly of single-molecule sequencing reads. *Nat. Biotechnol.* **30**, 693–700 (2012).
7. Fu, S. *et al.* IDP-denovo: de novo transcriptome assembly and isoform annotation by hybrid sequencing. *Bioinformatics* **34**, 2168–2176 (2018).
8. Koren, S. *et al.* Canu: scalable and accurate long-read assembly via adaptive k-mer weighting and repeat separation. *Genome Res.* **27**, 722–736 (2017).
9. Salmela, L., Walve, R., Rivals, E. & Ukkonen, E. Accurate self-correction of errors in long reads using de Bruijn graphs. *Bioinformatics* **33**, 799–806 (2017).
10. Xiao, C.-L. *et al.* MECAT: fast mapping, error correction, and de novo assembly for single-molecule sequencing reads. *Nat. Methods* **14**, 1072–1074 (2017).
11. Li, Y. *et al.* DeepSimulator: a deep simulator for Nanopore sequencing. *Bioinformatics* **34**, 2899–2908 (2018).
12. Marchet, C. *et al.* De novo clustering of long reads by gene from transcriptomics data. *Nucleic Acids Res.* (2018). doi:10.1093/nar/gky834
13. Sahlin, K. & Medvedev, P. De novo clustering of long-read transcriptome data using a greedy, quality-value based algorithm. in *International Conference on Research in Computational Molecular Biology* 227–242 (Springer, 2019).
14. Sessegolo, C. *et al.* Transcriptome profiling of mouse samples using nanopore sequencing of cDNA and RNA molecules. *Sci. Rep.* **9**, 14908 (2019).
15. Morisse, P., Marchet, C., Limasset, A., Lecroq, T. & Lefebvre, A. CONSENT: Scalable self-correction of long reads with multiple sequence alignment. *bioRxiv* 546630 (2019). doi:10.1101/546630
16. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
17. Tang, A. D. *et al.* Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns. *bioRxiv* 410183 (2018). doi:10.1101/410183
18. Wyman, D. *et al.* A technology-agnostic long-read analysis pipeline for transcriptome discovery and quantification. *bioRxiv* 672931 (2019). doi:10.1101/672931
19. Ruiz-Reche, A., Srivastava, A., Indi, J. A., de la Rubia, I. & Eyras, E. ReorientExpress: reference-free orientation of nanopore cDNA reads with deep learning. *Genome Biol.* **20**, 260 (2019).
20. Trincado, J. L. *et al.* SUPPA2: Fast, accurate, and uncertainty-aware differential splicing analysis across multiple conditions. *Genome Biol.* **19**, (2018).

21. Rivas, M. A. *et al.* Human genomics. Effect of predicted protein-truncating genetic variants on the human transcriptome. *Science* **348**, 666–9 (2015).

22. Singh, M. *et al.* High-throughput targeted long-read single cell sequencing reveals the clonal and transcriptional landscape of lymphocytes. *Nat. Commun.* **10**, 3120 (2019).

23. Lee, C., Grasso, C. & Sharlow, M. F. Multiple sequence alignment using partial order graphs. *Bioinformatics* **18**, 452–64 (2002).

24. Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–2 (2012).

25. Vinh, N. X., Epps, J. & Bailey, J. Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *J. Mach. Learn. Res.* **11**, 2837–2854 (2010).

26. Rosenberg, A. & Hirschberg, J. V-measure: A conditional entropy-based external cluster evaluation measure. in *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)* 410–420 (2007).