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

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

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

[2]Pompeu Fabra University (UPF), E08003 Barcelona, Spain.

[3]Universidade de Lisboa, Lisboa, Portugal

[4]Centre for Regulatory Genomics (CRG), E08001 Barcelona, Spain

[5]Catalan Institution for Research and Advanced Studies (ICREA), E08010 Barcelona, Spain

[6]Hospital del Mar Medical Research Institute (IMIM), E08001 Barcelona, Spain

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

* co-corresponding authors: malba@imim.es, eduardo.eyras@anu.edu.au

Single-molecule long-read sequencing with Oxford Nanopore Technologies (ONT) provides an unprecedented opportunity to describe transcriptomes from any sample and species[1,2]. However, current analysis methods rely on the comparison with a reference genome or transcriptome[2–5], or on the use of multiple sequencing technologies[6,7]. This precludes the use of ONT technologies for cost-effective studies in species with no available genome assembly and in individuals underrepresented in the existing reference, and for the discovery of disease-specific transcripts not readily identifiable from a reference genome. Methods for *de novo* DNA assembly[8–10] cannot be directly transferred to transcriptomes, since they do not model genes with multiple transcript isoforms, whereas short-read methods rely on high sequence accuracy, which is currently lacking in individual long reads. To address these challenges, we have developed RATTLE, the first tool to perform reference-free reconstruction and quantification of transcripts from Nanopore long reads. Using simulated data, isoform spike-ins, and sequencing data from human tissues and cell lines, we demonstrate that RATTLE accurately determines transcript sequences and their abundances, performs comparably to reference-based methods, and shows a tendency towards saturation in the number of predicted transcripts with increasing reads.

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 greedy deterministic clustering using a two-step k-mer based similarity measure (Fig. 1a). The first step consists of a fast comparison of k-mers (k=6) shared between two reads (Supp. Fig. 1a), whereas the second step solves the Longest Increasing Subsequence (LIS) problem to find the largest list of collinear matching k-mers between a pair of reads, which defines the RATTLE similarity score (Supp. Fig. 1b). Clusters are greedily generated by comparing reads to a representative of each existing cluster at every step of the iteration. This process generates clusters, where each cluster represents a potential gene containing reads originating from all possible transcript isoforms.

Gene clusters are subsequently split into sub-clusters defining candidate transcripts. These transcript clusters are built by determining whether every pair of reads in a gene cluster is more likely to originate from different transcript isoforms rather than from the same isoform. This is estimated from the distribution of gap-lengths between the previously calculated co-linear matching k-mers (Supp. Fig. 1c). RATTLE performs error correction within each one of the transcript clusters by generating a block-guided multiple sequence alignment (MSA) (Fig. 1a). Each read is assessed for error correction taking into account the error probability for each base and the average error probability of the consensus at each MSA column. RATTLE then builds the final transcripts after a polishing step to refine the cluster definitions. The final transcript sequence is obtained as a consensus from the final transcript cluster and the abundance is defined as the total read count in that cluster (Fig. 1a). More details are provided in the Methods section.

To evaluate the strength of RATTLE similarity score to perform read clustering, we simulated reads from multiple annotated transcripts with DeepSimulator[11], taking into account the read length distribution observed in a Nanopore cDNA sequencing run (Supp. Fig. 1d). RATTLE similarity score separated reads originating from different transcripts better than a minimizer-based score (Fig. 1b). To test the ability of RATTLE to separate reads from two transcript isoforms from the same gene, we considered reads simulated from two transcripts that differ from each other only by an internal alternative exon of length 154nt (Supp. Fig. 1e). RATTLE similarity score separated reads originating from each of the two transcript isoforms better than using the number of common bases between reads as similarity measure (Fig. 1c).

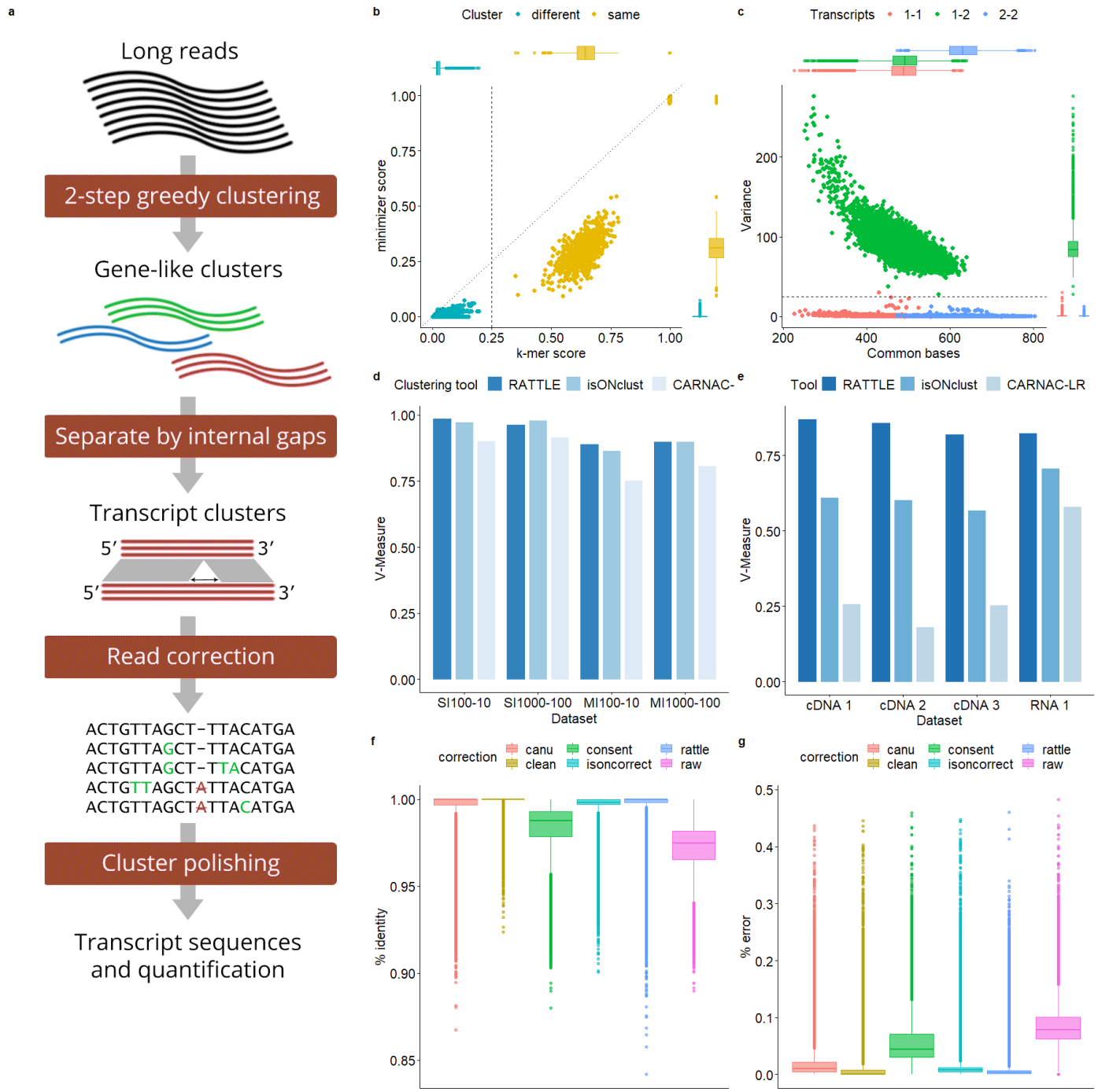

Figure 1. (a) Illustration of the RATTLE workflow. **(b)** Comparison of the RATTLE similarity score (x axis), based on the longest increasing subsequence, with a similarity score based on minimizers (y axis), using k=6. Each dot represents a comparison between a pair of reads simulated from the same (orange) or different (blue) transcripts from two different genes. The distribution of values for each comparison is given as box plots along each axis. **(c)** Number of common bases between two reads (x-axis) and variance in the distribution of gap-length differences between adjacent matching k-mers between the same two reads (y axis). Reads were simulated from two transcript isoforms from the same gene (Supp. Fig. 1e). Each dot is colored according to whether the reads originated from the same transcript, 1-1 (red) or 2-2 (blue) or not, 1-2 (green). **(d)** Clustering accuracy of RATTLE, CARNAC, and isONclust in terms of the V-measure (y axis), using simulated reads. Simulations (x-axis) were performed with a single (SI) or with multiple (MI) transcript isoforms per gene, using a different number of total transcripts (*t*) and a different number of reads per

transcript (*r*), indicated as SI*t*-*r* or MI*t*-*r*. Other accuracy metrics are provided in Supp. Table S1. **(e)** Clustering accuracy using spike-in isoform RNA variants (SIRVs) as reference. The plot shows the V-measure (y axis) for RATTLE, CARNAC, and isONclust using SIRV reads from four of the tested samples (x axis): three using the cDNA-seq protocol (cDNA1, 2, 3) and one using the dRNA protocol (RNA1) (Methods). **(f)** Box plots representing the percentage identity distributions of all SIRV reads before (raw) and after correcting with RATTLE, CONSENT, isONcorrect, Canu, and TranscriptClean (clean) for Nanopore cDNA-seq data (sample cDNA2). Percentage identity was calculated as the number of nucleotide matches divided by the total length of the aligned region. Other samples are shown in Supp. Fig. 2. **(g)** Error rate distribution of SIRV reads before (raw) and after correction with RATTLE, CONSENT, isONcorrect, Canu, and TranscriptClean (clean) for the same sample as in (f). The error rate was calculated as the sum of insertions, deletions, and substitutions divided by the length of the read. Other samples are shown in Supp. Fig. 3.

To test the accuracy in the identification of gene clusters, we compared RATTLE with two other methods that cluster 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 with a variable number of reads per transcript. RATTLE showed higher accuracy at recovering gene clusters in most of the comparisons and using different metrics (Fig. 1d) (Supp. Table S1). At the transcript level, RATTLE performed generally better than CARNAC and isONclust (Supp. Table S1). Moreover, RATTLE was faster than CARNAC and isONclust in all datasets (Supp. Table S1).

We further assessed the accuracy of RATTLE at recovering gene clusters using Spike-in isoform RNA Variants (SIRVs). The SIRV genome (SIRVome) is organized into 7 different genes containing a total set of 69 transcripts with known coordinates, sequences, and abundances. We performed 4 different experiments with added SIRVs: PCR-based cDNA sequencing (cDNA-seq) from human brain (two independent replicates, cDNA1, cDNA2) and human heart (cDNA3) samples, and direct RNA (dRNA-seq) from the same heart sample (RNA1). We first used the SIRV transcripts aggregated per gene to evaluate clustering at the gene level. RATTLE was faster and showed higher accuracy in the identification of SIRV genes compared with the other two methods (Fig. 1e) (Supp. Table S2). Similar results were found for dRNA and cDNA reads using SIRVs in mouse samples[14] (Supp. Table S2). Furthermore, RATTLE achieved high accuracy at clustering reads into SIRV transcripts using multiple metrics (Supp. Table S2).

To test RATTLE accuracy to correct errors in Nanopore reads without using a reference, we used the same SIRV reads and compared RATTLE results with methods developed for long-read self-correction. We used CONSENT[15] and isONcorrect[16], and the self-correction step from Canu[8]. Reads were mapped to the SIRVome with Minimap2[17] before and after correction by each method. As a comparison, we included

TranscriptClean[3], which uses mapping to a genome reference sequence to correct reads. After correction, all methods improved the percentage identity with the SIRV transcript isoforms (Fig. 1f) (Supp. Fig. 2) and showed a decrease in the error rates (Fig. 1g) (Supp. Fig. 3). RATTLE showed similar percentage identities to TranscriptClean and isONcorrect, and higher values than Canu and CONSENT. On the other hand, RATTLE showed lower error rates compared with the other self-correction methods, and very similar values to TranscriptClean and isONcorrect. Notably, RATTLE was faster than all other methods in all tested samples, with runtimes ranging from 1.64 minutes (mins) (for 14,958 reads) to 123.9 mins (for 214,107 reads) (Supp. Table S3). Furthermore, RATTLE corrected approximately as many reads as TranscriptClean and isONcorrect, and more than CONSENT (Supp. Table S3). We also performed error correction using reads from the human transcriptome. As before, RATTLE showed similar values to isONcorrect and TranscriptClean, and higher percentage identities and lower error rates compared with the other self-correction methods (Supp. Fig. 4) (Supp. Table S4).

We next evaluated the ability to accurately recover exon-intron structures after mapping the corrected reads to the SIRVome, focusing on introns and intron-chains. RATTLE recovered slightly fewer introns and intron-chains than other methods (Fig. 2a, left panel) but it displayed precision values similar to isONcorrect and higher than the other methods (Fig. 2a, middle panel). The precision was generally low, suggesting that tools may produce many false positives. To further investigate this, we calculated a read-precision metric, defined as the proportion of reads supporting correctly identified annotated features. All tools showed a very strong increase in read-precision with respect to precision (Fig. 2a, right panel), indicating that most of the wrongly predicted features are supported by very few reads. Similar trends were observed for internal and external exons (Supp. Fig. 5) (Supp. Table S5), with RATTLE showing slightly lower recall but higher or similar precision and read-precision compared with other methods.

We performed a similar analysis mapping the reads to exon-intron structures in the human gene annotation. As no ground truth was available, we used as reference the exon-intron structures of Gencode transcripts that had evidence of expression in each sample (>5 mapped reads). In the human transcriptome, RATTLE recovered a similar proportion of introns and intron-chains as the other tools (Fig. 2b, left panel). For precision (Fig. 2b, middle panel) and read-precision (Fig. 2b, right panel), RATTLE achieved similar or higher values than the other methods, except CONSENT, which showed higher precision and read-precision for cDNA sequencing. For internal and external exons, although RATTLE showed lower recall than other methods, it achieved higher precision and read precision values (Supp. Fig. 6) (Supp. Table S6).

We observed that false positives had generally lower read support than true positives (Supp. Fig. 7). We thus decided to exploit this observation to establish the accuracy in terms of read support. For each method, we calculated the recall, precision, and read-precision for SIRV introns at different thresholds of read support, and estimated the minimal read support needed to achieve a precision of 0.95 (i.e. 5% of the predictions would be false). All correction methods showed an improvement over the raw reads. Moreover, RATTLE and isONcorrect were the methods that required the least read support to achieve 0.95 precision (Fig. 2c) (Supp. Fig. 8). We did the same analysis using the expressed Gencode transcripts (>5 reads) as reference and found again that RATTLE and isONcorrect required the least read support for 0.95 precision (Fig. 2d) (Supp. Fig. 9)

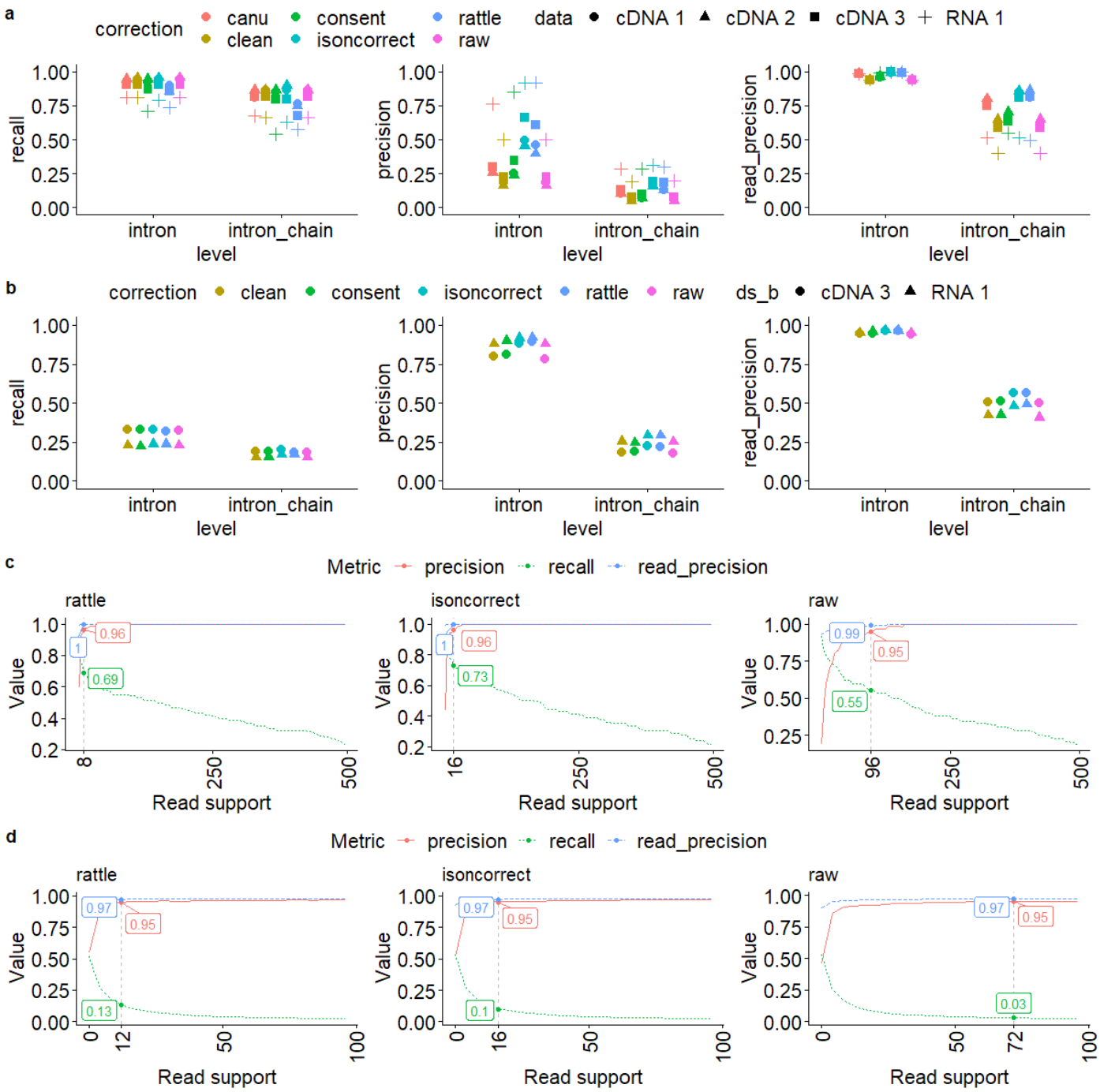

Figure 2. (a) Left panel: recall of unique SIRV introns and intron-chains obtained by mapping reads to the SIRV genome before (raw) and after correction with RATTLE, CONSENT, Canu, isONcorrect, and TranscriptClean (clean) for the dRNA-seq (RNA 1) and the cDNA (cDNA 1, 2, 3) samples. Recall was calculated as the fraction of unique annotated introns or intron-chains correctly (and exactly) found by each method with >5 supporting reads. Middle panel: Precision values for introns and intron-chains for the same methods and datasets. Precision was calculated as the fraction of unique introns or intron-chains predicted by reads that matched exactly the SIRV annotation with support of >5 reads. Right panel: Read-precision for introns and intron-chains for the same methods and datasets. Read-precision was calculated as the fraction of all introns or intron-chains predicted in reads that corresponded to SIRV introns or intron chains and had support of >5 reads. Similar plots for internal and external exons are given in Supp. Fig. 5. **(b)** Same accuracy plots as in (a) for the same methods and datasets but using expressed Gencode transcripts (>5

reads) as reference. **(c)** We plot the recall (green), precision (red) and read-precision (blue) of the SIRV introns (y axis), as a function of the number of minimum reads supporting the predictions (x axis). We indicate for each case the threshold at which a precision (red) of approximately 0.95 is achieved. For that threshold we indicate the corresponding recall (green) and read-precision (blue). The plot corresponds to the dataset cDNA2. Results for other samples are available in Supp. Fig. 7. **(d)** Same accuracy plots as in (c) but using expressed Gencode transcripts (>5 reads) as reference. Results for other samples are available in Supp. Fig. 8.

We next tested the accuracy of RATTLE to recover the abundance of transcript isoforms. We first used the known abundances of the SIRV isoforms and compared RATTLE with StringTie2[4], FLAIR[5], and TALON[18], which use the genome and annotation references to delineate transcript isoforms and their abundances. For RATTLE, we assigned each SIRV to the best matching predicted transcript. Despite not using any information from the SIRV genome or annotation, the correlation with SIRV isoform abundances achieved by RATTLE was comparable to that obtained with the other programs (Fig. 3a) (Supp. Table S7). We also observed that the correlation with cDNA reads was generally lower than with dRNA reads for all methods, also for reads directly mapped to the SIRVome with Minimap2 (Supp. Fig. 10) (Supp. Table S7), providing strong evidence that dRNA reads produce better transcript abundance estimates than cDNA reads. Furthermore, analysis of the relationship between recall and precision of the RATTLE transcripts mapped to the SIRVome showed that one could attain high recall (~0.75) while maintaining high precision (>0.75) (Supp. Fig. 11).

We further tested the accuracy of transcript predictions in human. First, we observed that the length distribution of the transcripts predicted by RATTLE was similar to those from reference-based methods, and all methods produced transcripts that were generally shorter than the annotation (Gencode v29) (Fig. 3b) (Supp. Fig. 12). Second, we mapped the transcripts predicted by each tool to the Gencode annotation and computed the fraction of each annotated transcript covered by the predicted transcript. We found that the distribution was similar across methods and that one could clearly differentiate between complete or near complete transcripts (above the dashed line in Fig. 3c) from partial reconstructions (see also Supp. Fig. 13). Moreover, matches with more than 75% coverage showed high percentage identity (median values 99% in cDNA, 98% in dRNA) (Supp. Fig. 14). We then determined the correlations between the abundances predicted by each method and those estimated by mapping reads to the annotated transcripts. RATTLE achieved similar correlations to those obtained by the reference-based methods and predicted fewer transcripts from cDNA reads but a similar number of transcripts from dRNA reads (Fig. 3d) (Supp. Fig. 15).

Next, we assessed the accuracy of the RATTLE transcriptome by comparing the exon-intron structures of the predicted transcripts mapped to the genome with the Gencode transcripts with evidence of expression (>5 reads) in the same sample. RATTLE recall values for introns and intron-chains were lower than FLAIR and StringTie values, but higher than TALON, whereas RATTLE precision values were lower but close to the reference-based methods (Fig. 3e). RATTLE performed similarly to reference-based methods for exons, and all methods performed better for internal exons than for external exons (Supp. Fig. 16).

To determine the reproducibility of the transcriptomes predicted by RATTLE, we analyzed replicates of cDNA and dRNA data from a lymphoblastoid cell line (LCL)[2]. We mapped the transcripts predicted by RATTLE, FLAIR, StringTie2, and TALON, to the human transcriptome annotation as previously, and compared the predicted abundances from each replicate. Although RATTLE generally recovered fewer transcripts, it showed similar correlation values to the reference-based methods (Fig. 3f, Supp. Table S8). Moreover, the correlation values for all methods were generally higher for dRNA than for cDNA reads (Fig. 3f).

To investigate the relationship between the number of predicted transcripts and the number of input reads, we used data from multiple dRNA-seq runs in LCLs[2] to generate datasets from 100k to 3M reads. None of the methods showed a clear tendency to saturate (Fig. 3g left panel). The total number of transcripts predicted by RATTLE and StringTie2 (without annotation) was lower than the other methods, whereas the other methods and especially TALON showed a steeper increase in the number of transcripts (Supp. Table S9). We further measured the number of matched annotated transcripts using the same approach as above. The shape of the curve relating the number of transcripts and the number of input reads was similar for the different methods, although TALON tended to recover a larger number of transcripts for > 2 million input reads (Fig. 3g right panel).

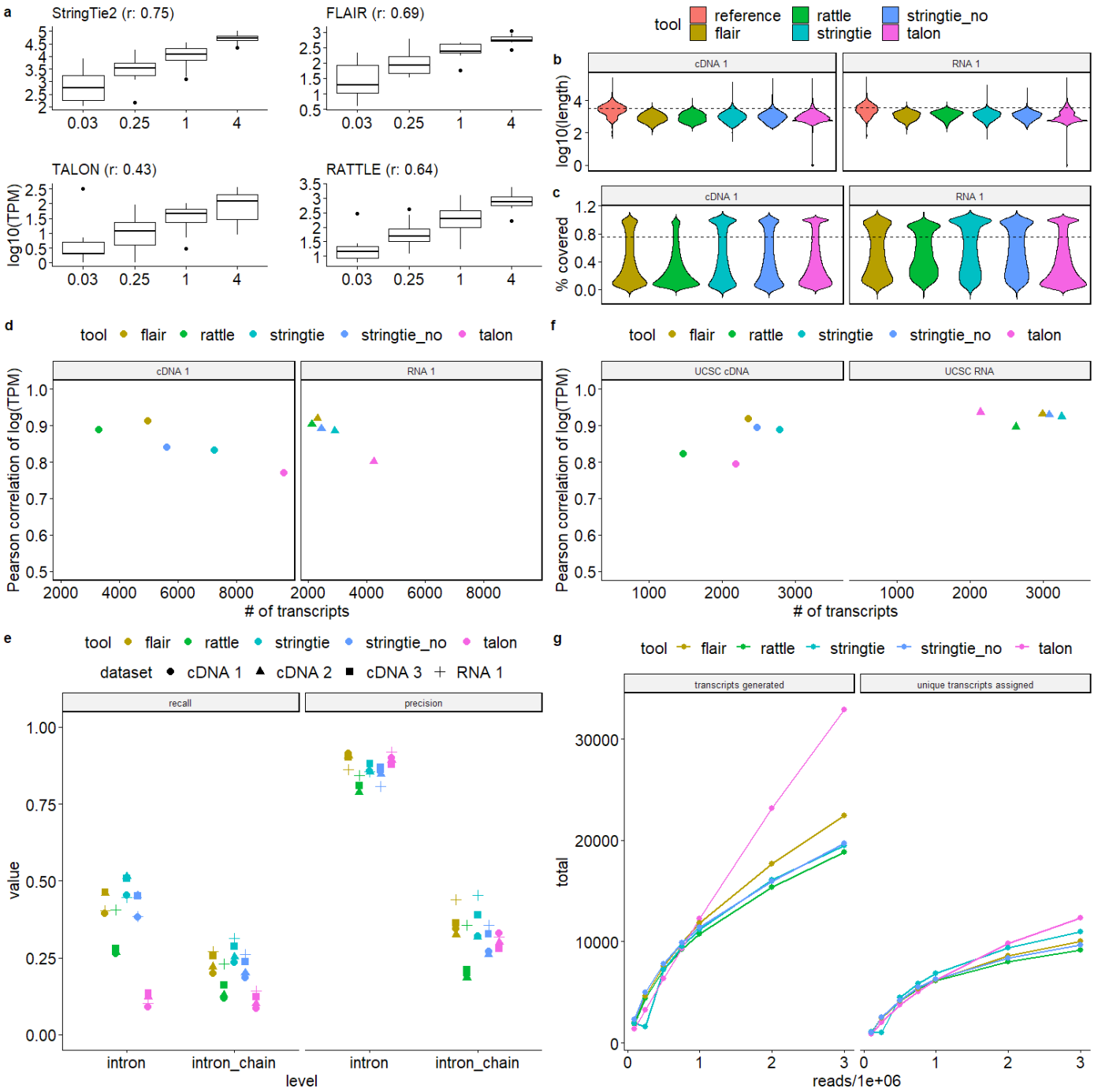

Figure 3. (a) Comparison of the transcript abundances (y axis) predicted by RATTLE, FLAIR, StringTie2, and TALON, with abundances of the SIRV spike-in transcript isoforms (x axis). Each panel shows the Pearson correlation *r* for the comparison of the abundance values. Units on the y-axis vary according to the method: RATTLE provides abundances as read counts per million, like TALON and FLAIR. StringTie2 produces a TPM value. SIRV data corresponds to the RNA1 sample. Correlations for other datasets are provided in Supp. Fig. 10. **(b)** Distribution of the lengths (y-axis, log10 scale) of the transcripts predicted by each method compared with the transcripts from the reference (Gencode v29) using cDNA and dRNA. StringTie2 was run with (stringtie) and without (stringtie_no) the annotation. The dashed line indicates the median value for the reference transcripts. **(c)** Distribution of the proportion of the annotated transcript covered (y-axis) by the best matching predicted transcript from each method. The dashed line indicates 0.75. **(d)** Correlation (y axis) of the predicted transcript abundances with the reference set of Gencode (v29) transcripts

with expression (>5 reads) in the same sample, for cDNA (cDNA1) and dRNA (RNA1) sequencing reads. We also give the number of predicted transcripts matched to the reference annotation (x-axis). Other samples are shown in Supp. Fig. 15. **(e)** Left panel: recall of unique introns and intron-chains from the Gencode (v29) annotation obtained by mapping the transcripts from each method to the genome. The recall was calculated as the fraction of unique annotated introns or intron-chains exactly matched by one or more predicted transcripts. Right panel: precision values for the introns and intron-chains after mapping the transcripts predicted by each method to the genome. The precision was calculated as the fraction of unique introns or intron-chains in the predicted transcripts that matched exactly the Gencode annotation. Similar plots for internal and external exons are given in Supp. Fig. 16. **(f)** Correlation (y-axis) of the predicted transcript abundances from two replicate cDNA and dRNA sequencing experiments from the Nanopore sequencing consortium. For each method, we give the number of predicted transcripts that were matched to the reference annotation in both replicates (x-axis). **(g)** Saturation plot of transcripts using an incremental number of dRNA reads. Left panel: for each input (x-axis), we give the total number of transcripts predicted by each method (y-axis) with an expression of >5 reads. Right panel: for each input (x-axis), we give the total number of annotated transcripts matched by the predicted transcripts (see Methods for details). The number of transcripts and genes obtained are given in Supp. Table S9.

Our analyses provide compelling evidence that RATTLE builds transcripts and estimates their abundances with accuracy comparable to reference-based methods, despite not using any information from the reference genome or annotation, and without the use of additional technologies. RATTLE generally achieved a high precision, which will be fundamental to reliably identify new genes and transcript isoforms in unannotated samples. Our analyses also showed that with an increasing number of input reads, RATTLE recovers a conservative number of transcripts compared to reference-based methods. RATTLE is also competitive in terms of running times, being faster in read clustering and error correction than other methods. Moreover, with an increasing number of input reads, RATTLE showed slightly above linear running time, and its memory footprint grew linearly in the clustering step and was sublinear in the correction step (Supp. Fig. 17) (Supp. Tables S10 & S11).

RATTLE modularity, with the ability to parameterize each step, means it can be easily adapted to any sample type. Additionally, RATTLE rich output, including information about the predicted transcripts and genes, as well as the reads used to build each transcript, will prove valuable in downstream applications, including the study of differential transcript usage[19], the identification of transcript sequence polymorphisms between individuals[20], and the analysis of single-cell long-read sequencing[21]. We anticipate that RATTLE will enable cost-effective transcriptomics in any sample and any species using Nanopore sequencing and, in particular, in species without an available reference genome and in populations underrepresented in the current human reference.

Data availability

Nanopore sequencing data generated in this study has been deposited in the European Nucleotide Archive (ENA) under the umbrella study PRJEB40410 (https://www.ebi.ac.uk/ena/browser/view/PRJEB40410) including the FASTQ files (https://www.ebi.ac.uk/ena/browser/view/PRJEB39835) and signal data (https://www.ebi.ac.uk/ena/browser/view/PRJEB40335). Further datasets used in this study is available in ENA under study accession PRJEB27590 (http://www.ebi.ac.uk/ena/data/view/PRJEB27590) (runs ERR2680375 and ERR2680377), and from the Nanopore sequencing consortium from https://github.com/nanopore-wgs-consortium/NA12878 (under nanopore-human-transcriptome).
Data files with the simulated sets of reads, as well as the output files generated by the different assembly programs have been deposited in figshare (https://figshare.com/projects/RATTLE_Paper_Data/113580).

Software availability

RATTLE is written in C++ and is available at https://github.com/comprna/RATTLE under the GPL-3.0 license.

Potential competing interests

Eduardo Eyras has received support from Oxford Nanopore Technologies (ONT) to present the results from this manuscript at scientific conferences. However, ONT played no role in the algorithm or software developments, study design, analysis, or preparation of the manuscript.

Methods

RATTLE clustering algorithm

Before running RATTLE, reads were pre-processed with porechop (https://github.com/rrwick/Porechop) and those of length 150nt or shorter were filtered out. RATTLE starts by performing a greedy clustering similar to the one used by CD-HIT[22]. RATTLE sorts the reads in descending order by their length and processes one at a time in that order. In the first iteration, RATTLE creates a new cluster with the first unclustered read. All subsequent unclustered reads are then compared against each existing cluster and assigned greedily if the scores resulting from the comparison are above certain thresholds. Otherwise, a new cluster is created. In each subsequent iteration, the threshold is decreased, and clusters are created or merged as initially. Each cluster

is selected, and all other clusters, including single-read clusters, i.e. singletons, are compared against the selected cluster. For all the clusters that 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 and circumvent the quadratic time complexity of an all-vs-all comparison, 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 0.15 x (number of reads in the cluster).

Reads are compared using a two-step similarity calculation. The first one provides a fast approximate comparison, whereas the second one provides a more sensitive comparison if a threshold is passed. 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 $s=b_1\ldots b_k$, $H(s) = 4^{k-1}H(b_1) + 4^{k-2}H(b_2) + \ldots + H(b_k)$. The first similarity score is calculated as the number of common 6-mers shared between two reads divided by the maximum number of 6-mers in either read. All 6-mers are extracted for each pair of reads and a 46-bit vector is created and the positions in the vector of the hashed 6-mers in each read are set to 1. An AND operation is then performed between the two vectors to obtain the number of common 6-mers. Extraction and hashing of 6-mers is performed only once per read in linear time, and the vector operations are performed in constant time. If this first score is above a previously set threshold, a second similarity calculation is performed. For the second metric, all k-mers from both reads are extracted. Now k can be chosen on the command line, and k-mers are hashed as before. The intersection of the k-mers from both reads and their positions in each read are used to generate a list of triplets (s, p_1, p_2), where s is the hashed k-mer, p_1 is the position of this k-mer in the first read, and p_2 is the position of the same k-mer in the second read. These triplets are then sorted by p_1 and the Longest Increasing Subsequence (LIS) problem is solved with dynamic programming for p_2. This produces the longest 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[23], RATTLE tests both relative orientations for each pair of reads. Consequently, all reads within a cluster have the same orientation.

The number of iterations to be performed for clustering is specified in the command line by setting the initial and final thresholds for the first bitvector-based score (default 0.4 to 0.2) and a decreasing step (default 0.05), i.e. as 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 threshold 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 initial and final thresholds in the bitvector-based comparison and the decreasing step, as well as the k-mer length and similarity thresholds in the LIS comparison (bases and variance) can be set up as input parameters.

RATTLE transcript-cluster identification and error correction

Initial read clusters produced by the algorithm described above are considered to approximately 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 considers 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 difference in exon content. The value 25 was used for the analyses, but it can be also modified as an 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. In our analyses, R was set to 200. If $R \leq N < 2R$, the cluster is split in half, and if $N < R$, we take 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 is made from reads in positions 1, K+1, 2K+1, …; block 2 from reads in positions 2, K+2, 2K+2, … ; and block K from reads in positions K, 2K, 3K, … . A multiple sequence alignment (MSA) is then built from each block using SIMD partial order alignment (SPOA) (https://github.com/rvaser/spoa)[24]. 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 base occurs with at least 60% frequency, but not if this base has an error probability (obtained from the FASTQ file) less than or equal to 1/3 times the average for the consensus base in that position of the alignment. Indels are treated similarly, but without the error constraint. This is only performed using aligned positions. That is, for each base in a given read, other reads of the MSA contribute to the correction if they have a base or an internal gap in that position. The consensus sequences from all blocks 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 considered for further analysis. Here we used transcript-clusters with more than 5 reads. The frequency of the consensus, error-probability cutoff, and the minimum number of reads for a transcript cluster can be set up as input parameters. We

observed that in MSAs many reads had a few bases wrongly aligned at the terminal regions. To fix these cases so that they do not affect the correction step, RATTLE identifies small blocks (<10nt) followed by large gaps (larger than or equal to 20 positions) at both ends of each aligned read and removes them. A block is defined as a subsequence that has gaps shorter than or equal to 3nt. RATTLE keeps removing blocks that satisfy these conditions at both ends of every aligned read until there no more such blocks are found.

RATTLE transcript polishing and quantification

To define the final list of transcripts, RATTLE performs a final polishing step of the transcript-cluster definitions. This is done to correct possible splitting of reads originating from the same transcript into different transcript-clusters (under-clustering). For this, RATTLE uses the same 2-step greedy clustering described above on the transcript-clusters using the representative read from each cluster in the comparison. In each of the resulting clusters, an MSA column consensus is calculated, with abundance given by all the reads contained in the final cluster. Additionally, the transcripts are given a gene ID that corresponds to the gene-clusters to which they belong. When two transcript clusters are merged, if they are part of the same original gene-cluster, the resulting transcript stays in the same gene. If they are 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. The same binary files are also used for the correction step, which outputs three files: one 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. The quantification of each transcript is included in the header line. The quality per base in each FASTQ entry is calculated as the average of the PHRED quality scores from the bases in each column of the MSA in the transcript-cluster.

Simulated reads and clusters

We developed a wrapper script (available at https://github.com/comprna/RATTLE) for DeepSimulator[11] to simulate a specific number of reads per transcript considering a 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), filtering out pseudogenes and genes from non-standard chromosomes, and removing transcripts that showed > 95% percentage identity with other transcripts using CD-HIT[22]. We then randomly selected different numbers 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 different numbers of reads per transcript and different numbers 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]. 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. The V-measure is the harmonic mean of completeness and homogeneity. We compared the clusters predicted by each method with the simulated clusters as the reference set. We run isONclust[13] (options: --ont --t 24), CARNAC-LR[12] with Minimap2 overlaps (options: -t 24 -x ava-ont), and RATTLE 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 with SIRVs

We performed Nanopore sequencing on two total RNA samples from human 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 the 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. The 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-LSK108 for an aliquot of the brain sample (cDNA 2), protocol SQK-LSK109 for 1 aliquot of the brain sample (cDNA 3) and 1 aliquot of the heart sample (cDNA 4). The direct RNA sequencing library was prepared using 500 ng of previously

prepared ribosomal depleted sample (RiboZero kit, catalog num. MRZH11124, Epicentre-Illumina) from heart (RNA 2) with standard direct RNA ONT protocol SQK-RNA002, following manufacturer's instructions, including the RT step. The final libraries were loaded on an R9.4.1 flowcell, and standard ONT scripts available in MinKNOW were used for 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 the corresponding protocol, as specified above (SQK-LSK109, SQK-LSK108 or SQK-RNA002). 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 24 -cx splice --splice-flank=no --secondary=no). To extract the subset of reads with a hit on the SIRVome and being at least 150nt in length we used seqtk (https://github.com/lh3/seqtk, option subseq). We also used data from cDNA (ERR2680377) (cDNA_br_mm) and direct RNA (ERR2680375) (RNA_br_mm) sequencing of mouse brain including the E2 SIRVs[14], as well as samples from the Nanopore sequencing consortium[2].

Clustering accuracy analysis

We first built SIRV isoform clusters by mapping reads to SIRV isoforms with Minimap2 and selecting for each read the SIRV isoform with the best mapping score. All reads that mapped to the same SIRV gene were then considered a gene-cluster. We then clustered reads with RATTLE, CARNAC, and isONclust and measured the accuracy of the predicted clusters by comparing them with the built SIRV gene clusters using the same metrics as with the simulated data. We used subsets of 25,000 reads (subsampled with seqtk) from each sample since we could not make CARNAC run for some of the complete datasets.

Assessment of error correction accuracy

Reads were mapped to the SIRV transcripts with Minimap2 before and after read correction. Here we used the complete dataset of SIRV reads. 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 nucleotide 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), isONcorrect[16] (options: --t 24), and TranscriptClean[3]. TranscriptClean was run with default parameters using

as input the reads mapped with Minimap2 (options: -t 12 -cx splice --splice-flank=no --secondary=no), but with no exon-intron information from the SIRV annotation.

To perform the accuracy analysis of the SIRV annotation features, PAF files from the mapping were compared with the annotation using ssCheck (available at https://github.com/comprna/RATTLE/). 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, intron-chains. internal exons and external exons. An intron-chain was defined as an ordered sequence of introns in an annotated transcript or mapped read. The recall was calculated as the fraction of unique annotated features correctly found; precision was calculated as the fraction of unique predicted features that were in the annotation and read-precision was calculated as the fraction of the total number of predicted features in reads that corresponded to annotated features. Read-precision is affected by abundance levels but better reflects the accuracy per read. We developed ssCheck to be able to calculate the read-precision, since gffcompare (https://ccb.jhu.edu/software/stringtie/gffcompare.shtml) merges identical intron chains with different identifiers, which precludes this calculation. Additionally, ssCheck calculates accuracy metrics for internal and external exons, which was not possible with gffcompare.

To perform the comparison with the human transcriptome we considered the Gencode transcripts (v29) after removing pseudogenes, genes from non-standard chromosomes, and transcripts with >95% percentage identity with other transcripts. We only used transcripts that had evidence of expression of >5 reads from mapping the reads to the transcriptome using minimap2 (options: -t 24 -x map-ont). We then mapped the reads corrected by each method to the human genome with minimap2 (options: -t 24 -cx splice --secondary=no). Considering the various genomic features in the expressed annotated transcriptome (introns, intron-chains, internal exons, external exons), we calculated the recall, precision, and read-precision with ssCheck as before.

Assessment of transcriptome sequence and abundance accuracy

We used FLAIR[5] (options: align, correct, collapse, quantify –tpm), StringTie2[4] (options: -p 24 –L) and TALON[18] (talon_initialize_database, talon, talon_summarize, talon_abundance, talon_create_GTF) with the cDNA and dRNA reads mapped to the SIRV genome with Minimap2 (options: -t 24 –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: -t 24). As an additional comparison, we mapped raw reads directly to SIRV isoforms with Minimap2 (options: -ax map-ont -t24) 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 isoforms with Minimap2 (options: -t 24 -cx map-ont --secondary=no). FLAIR, StringTie2, and TALON provide the SIRV isoform ID with the predicted abundance. If more than one predicted transcript mapped to the same annotation, 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 (options: -cx map-ont --secondary=no). If more than one transcript matched the same SIRV isoform, we selected the RATTLE transcript with the highest abundance.

For the abundance estimates using the reference human transcriptome (Gencode v29 filtered as described above), as the true transcript abundances were not known, we estimated them in the following way. We mapped the reads to the annotated transcripts annotations with minimap2 (options: -t 24 -x map-ont --secondary=no) and kept only transcripts with >5 mapped reads. The resulting transcriptome from each method, FLAIR, TALON, RATTLE, and StringTie2 (with and without using the annotation), was then mapped to the expressed annotated transcripts using minimap2 (options: -t 24 -x map-ont --secondary=no). To perform an accurate assignment of predicted to annotated transcripts, for each predicted transcript we only kept the best match if it covered at least 75% of the annotated transcript. Moreover, for each annotated transcript, we only assigned the best matching predicted transcript. If more than one passed the filter, we assigned the predicted transcript with the highest abundance. That is, the assignments of predicted and annotated transcripts were one-to-one. We then calculated the correlation of the transcript abundances by comparing the abundance estimated by the direct mapping of the reads to the Gencode annotation with the abundance of the assigned predicted transcripts.

To assess the accuracy of the transcriptome in terms of exon-intron structures, we compared the predicted transcripts mapped to the genome with the Gencode transcripts with evidence of expression (>5 reads) in the same sample. To make this analysis comparable across methods, all predicted transcriptomes were mapped against the genome with minimap2 with minimap2 (options: -t 24 -cx splice --secondary=no). We then calculated the exact matches of the predicted intron-exon structures (introns, intron chains, internal exons, external exons) with the annotation for that sample using ssCheck as before.

Correlation of human transcriptomes

We used data from cDNA and dRNA sequencing from the Nanopore consortium[2]: the samples from Johns Hopkins (cDNA 2 replicates, dRNA 2 replicates) and UCSC (cDNA 2 replicates, dRNA 2 replicates). Predicted transcripts were matched to the annotation, selecting the best match, and keeping only those that covered 75% or more of the annotated transcript. If two or more predicted transcripts passed those filters, the one with the highest abundance was selected. The correlation was calculated by comparing the predicted transcripts from the two replicates matched to the same annotated transcript.

Saturation analysis

We used 30 pooled direct RNA sequencing runs from the Nanopore sequencing consortium[2] (https://github.com/nanopore-wgs-consortium/NA12878/blob/master/RNA.md) and generated subsamples with 100k, 250k, 500k, 750k, 1M, 2M, and 3M reads. Subsampling was performed with seqtk (https://github.com/lh3/seqtk). We run FLAIR, TALON and StringTie (with and without annotation) on all these samples using the same options as before. RATTLE was run in all these samples (options: -B 0.5 -b 0.3 -f 0.2) using a machine with 24 cores. In each run, 64Gb RAM was requested although RATTLE only used 20-30Gb of RAM. We first considered the total number of transcripts predicted by each method for each input size with an expression >5 reads. For each input size we also performed a one-to-one assignment of the predicted transcripts to the transcriptome annotation as described above to calculate the total number of identified annotated transcripts for each input size.

Acknowledgments

We thank Roderic Guigó for providing access to the samples with SIRVs. We also thank the Nanopore sequencing consortium for providing early access to their Nanopore transcriptomic sequencing runs. We acknowledge funding support from the Spanish Government and FEDER with grants BFU2015-65235-P, BIO2017-85364-R, PGC2018-094091-B-I00, and MDM-2014-0370; from the Catalan Government (AGAUR) with grant SGR2017-1020; and Spanish National Bioinformatics Institute (INB), funded by ISCIII and FEDER (PRB2-ISCIII; Grant PT17/0009/0014). IdlR was supported by grant PRE2018-083413 from the Spanish Government.

Author contributions

EE and IdlR designed the algorithms in RATTLE with inputs from MMA. IdlR prototyped and implemented the algorithms. IdlR carried out the benchmarking analyses with some contributions from JI. SCS and JL generated the experimental data. EE, IdlR, and MMA 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. Wyman, D. & Mortazavi, A. TranscriptClean: variant-aware correction of indels, mismatches and splice junctions in long-read transcripts. *Bioinformatics* **35**, 340–342 (2019).
4. Kovaka, S. *et al.* Transcriptome assembly from long-read RNA-seq alignments with StringTie2. *Genome Biol.* **20**, 278 (2019).
5. Tang, A. D. *et al.* Full-length transcript characterization of SF3B1 mutation in chronic lymphocytic leukemia reveals downregulation of retained introns. *Nat. Commun.* **11**, 1438 (2020).
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. Sahlin, K. & Medvedev, P. Error correction enables use of Oxford Nanopore technology for reference-free transcriptome analysis. *Nat. Commun.* **12**, 2 (2021).
17. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
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. Trincado, J. L. *et al.* SUPPA2: Fast, accurate, and uncertainty-aware differential splicing analysis across multiple conditions. *Genome Biol.* **19**, (2018).
20. Rivas, M. A. *et al.* Human genomics. Effect of predicted protein-truncating genetic variants on the human transcriptome. *Science* **348**, 666–669 (2015).
21. Lebrigand, K., Magnone, V., Barbry, P. & Waldmann, R. High throughput error corrected Nanopore single cell transcriptome sequencing. *Nat. Commun.* **11**, 4025 (2020).
22. Fu, L., Niu, B., Zhu, Z., Wu, S. & Li, W. CD-HIT: accelerated for clustering the next-generation sequencing data. *Bioinformatics* **28**, 3150–3152 (2012).
23. 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).
24. Lee, C., Grasso, C. & Sharlow, M. F. Multiple sequence alignment using partial order graphs. *Bioinformatics* **18**, 452–464 (2002).
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).