IsoQuant: a tool for accurate novel isoform discovery with long reads

Andrey Prjibelski
Saint Petersburg University
Alla Mikheenko
Center for Algorithmic Biotechnology, Institute of Translational Biomedicine, Saint Petersburg State University
Anoushka Joglekar
Weill Cornell Medicine https://orcid.org/0000-0002-7818-6867
Alexander Smetanin
Bioinformatics Institute
Julien Jarroux
Weill Cornell Medicine
Alla Lapidus
St.Petersburg State University https://orcid.org/0000-0003-0427-8731
Hagen Tilgner (hagen.u.tilgner@gmail.com)
Weill Cornell Medicine https://orcid.org/0000-0002-7058-3606

Brief Communication

Keywords:

Posted Date: April 26th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1571850/v1

IsoQuant: a tool for accurate novel isoform discovery with long reads

Andrey D. Prjibelski[1,+], Alla Mikheenko[1], Anoushka Joglekar[2,3,4], Alexander Smetanin[5], Julien Jarroux[3,4], Alla L. Lapidus[1], Hagen U. Tilgner[3,4+]

[1] Center for Algorithmic Biotechnology, Institute of Translational Biomedicine, St. Petersburg State University

[2] Tri-Institutional Computational Biology & Medicine, Weill Cornell Medicine, New York, NY, USA

[3] Brain and Mind Research Institute, Weill Cornell Medicine, New York, NY, USA

[4] Center for Neurogenetics, Weill Cornell Medicine, New York, NY, USA

[5] Bioinformatics Institute, St. Petersburg, Russia

+ Corresponding authors: hut2006@med.cornell.edu, andrewprzh@gmail.com

Long reads are reshaping RNA biology. However, determining alternative isoforms from long-read RNA data is a complex and incompletely solved problem even when the reference genome is known. Here we present IsoQuant — a reference-based tool that accurately discovers novel transcripts with at least 3-fold lower false positive rate and 1.8-fold increase in F1-score compared to other tools for Oxford Nanopore data. IsoQuant also increases performance for Pacific Biosciences data.

Long-read RNA sequencing is now widely used in bulk, sorted cells, single cells and spatial approaches. This wide field of applications has led to the development of multiple spliced alignment programs[1–4], transcript discovery methods[5–11], tools for transcript classification[12], annotation[13] and visualization[14,15]. Additionally, several reference-free tools for RNA long-read correction and assembly have been developed[16,17]. Current community efforts address the problem of understanding performance, weaknesses and advantages of each approach for various applications[18].

Here we present IsoQuant — a tool for transcript discovery and quantification with long RNA reads. IsoQuant takes as input a reference genome, a gene annotation, and a dataset containing PacBio or ONT RNA reads. By default, IsoQuant maps input reads to the genome via minimap2 in splice mode[2]. Alternatively, a user may provide BAM files generated with a spliced aligner of their choice, e.g. STARlong[1] for PacBio and uLTRA[4] or deSALT[3] for ONT reads.

IsoQuant first assigns reads to known isoforms via an inexact intron chain matching algorithm that takes into account splice site shifts, which are typical for alignment of error-prone reads[19]. Uniquely assigned reads with consistent intron chains are then used for isoform and gene quantification (Methods). For inconsistent reads that are likely to originate from unannotated isoforms, IsoQuant also reports newly detected alternative splicing events. IsoQuant exploits read-

to-isoform assignments for transcripts quantification and correction of inaccurately detected splice junctions and misalignments, such as skipped microexons. Corrected alignments are used to construct an intron graph, in which vertices are introns, and two vertices are connected with a directed edge if the corresponding introns are consecutive in at least one read (Methods). Finally, this graph is exploited for constructing paths that correspond to full-length transcripts (Figure 1a).

To compare IsoQuant performance against existing transcript discovery tools, we first simulated mouse PacBio and ONT data using realistic gene expression profiles with IsoSeqSim (https://github.com/yunhaowang/IsoSeqSim) and Trans-NanoSim[20] respectively. To mimic real-life datasets containing unannotated transcripts, we arbitrarily removed 5,311 (15%) of 35,684 expressed isoforms (the ones contributing to at least one read during the simulation) from the GENCODE gene annotation. These 5,311 hidden transcripts were further used as a ground truth for novel transcript discovery. The reduced GENCODE annotation was used as an input for all tools. Each output annotation was then separated into a set of known and a set of novel transcripts, which were compared against the respective baselines using gffcompare[21] (Methods).

For known transcripts, IsoQuant has the highest F1-score (the harmonic mean of precision and recall) compared to TALON[7], FLAIR[8], Bambu[11] and StringTie[5], but these advances are not dramatic (Supplemental Tables 1-3). However, IsoQuant produces novel transcripts with a 1.8-fold higher F1-score on ONT data compared to the second best tool, StringTie. In comparison to TALON, FLAIR and Bambu, the improvement in F1-score is even more noticeable (Figure 1b left). On PacBio data, IsoQuant again shows the best F1-score, but the difference from other tools is smaller than for ONT data (Figure 1b, right). While IsoQuant's F1-score is clearly higher than that of other tools for highly-expressed transcripts, IsoQuant is able to maintain these advances for novel transcript discovery regardless of the expression levels (Figure 1c, Supplementary Figure 1). Thus, IsoQuant is likely to be highly useful across many genes, including but not limited to lowly expressed long-non-coding RNAs and marker genes of cell types.

Compared to most tools, IsoQuant's improvements in F1-score is primarily caused by its very high precision of novel transcripts. As compared to TALON, FLAIR and StringTie, IsoQuant shows a minimum of 6-fold drop in false-positive rate on ONT data, while still maintaining slight gains in recall (Figure 1d). Importantly, the situation is of a different nature for Bambu. IsoQuant has higher precision (88.2% vs. 64.2%), but substantially higher recall: while Bambu only reconstructs 70 out of 5,311 novel isoforms (1.3% recall), IsoQuant reconstructs 3,031 (57.2%).

For novel transcripts generated with PacBio data, similar trends can be observed, although with a less drastic difference in specificity. Bambu shows slightly higher precision (95.8%) compared to IsoQuant (94.0%), but again has the lowest recall (18.7% for Bambu vs. 74.2% for IsoQuant). StringTie, TALON and FLAIR again predict transcripts with comparable recall, but have at least 4.5-fold higher false-positive rate compared to IsoQuant (Figure 1e).

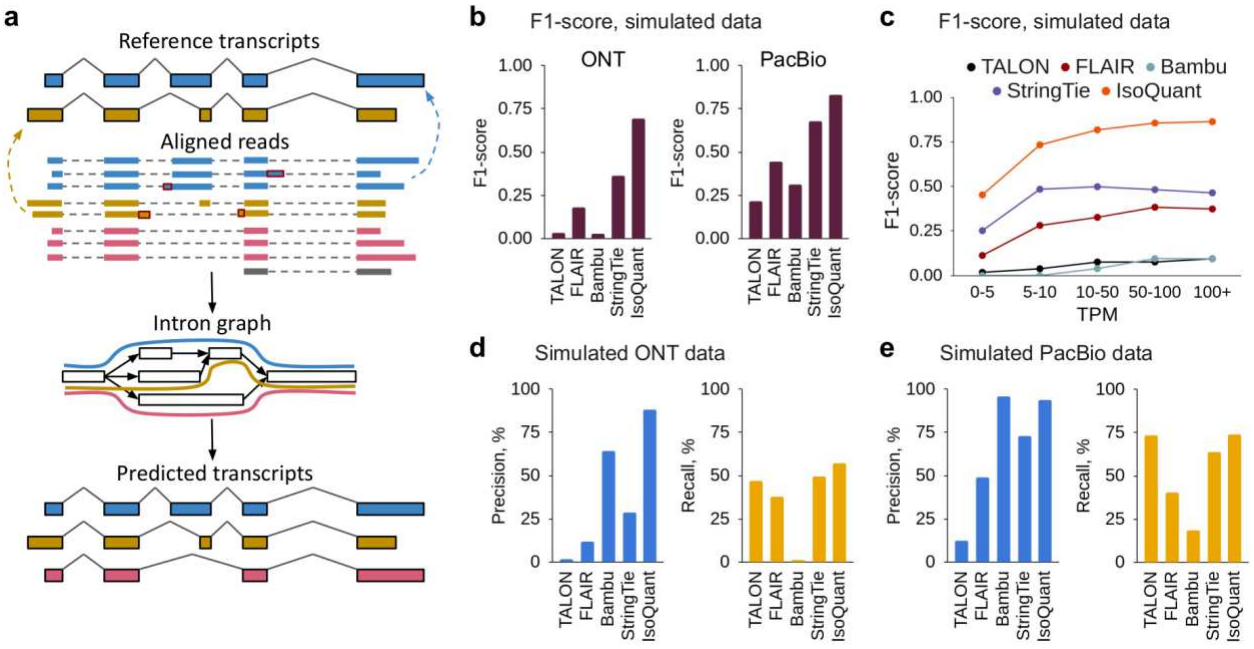

Figure 1. IsoQuant pipeline outline and characteristics of novel transcripts generated from mouse simulated data. a. Outline of the IsoQuant pipeline. Reads are assigned to annotated isoforms and alignment artifacts are corrected (top). The intron graph is constructed from corrected read alignments (middle) and transcripts are discovered via path construction (bottom). **b.** F1-score for novel transcripts reported by different tools on simulated ONT (left) and PacBio data (right). **c**. F1-score for novel transcripts reported by different tools on simulated ONT data broken up by expression levels in transcript-per-million (TPM). **d.** Precision (left) and recall (right) for novel transcripts reported by different tools on simulated ONT data. **e.** Same as d, but for simulated PacBio data.

To complement our benchmarks on simulated data, we also sequenced Lexogen SIRVs synthetic molecules on the Oxford Nanopore MinION (Methods). Along with the complete SIRV annotation, Lexogen provides an incomplete annotation, missing 26 out of the total 69 SIRV isoforms, which allows the evaluation of novel transcript discovery, similar to the one we performed for simulated data with the reduced GENCODE annotation.

Results on SIRV sequencing data resemble the ones obtained on simulated reads. When predicting novel isoforms, IsoQuant shows at least 5 times higher F1-score and 3-fold lower false positive rate than any other tool. In comparison to most tools, with the exception of TALON, IsoQuant shows high gains in both precision and recall. TALON and IsoQuant have identical recall of 42.3%, but IsoQuant has 16-fold higher precision (Figure 2a). Similar to simulated data, all tools are able to accurately predict SIRV transcripts kept in the annotation, with Bambu, StringTie and IsoQuant having perfect precision for known isoforms alone (Supplementary Table 4).

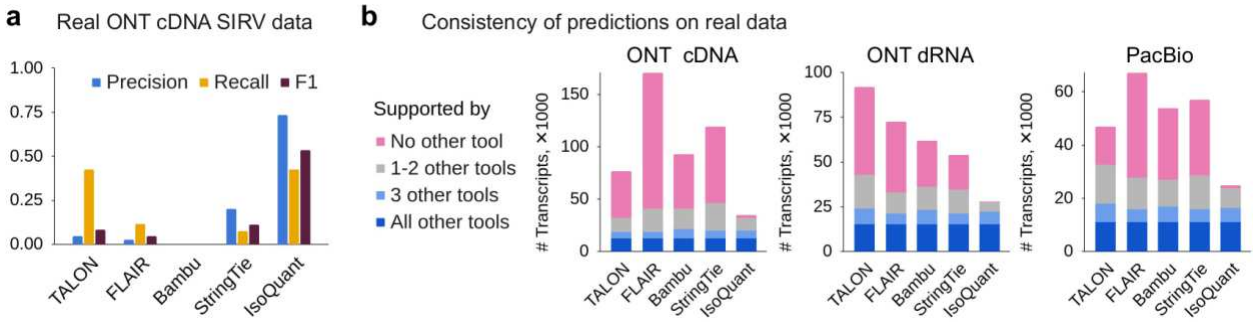

Figure 2. Characteristics of transcripts obtained from real sequencing data. a. Precision, recall, and F1-score for novel transcripts generated on real SIRV ONT cDNA sequencing data. **b.** Consistency of predictions made by different methods on real human ONT cDNA, ONT dRNA and PacBio data.

To support our observations, we also applied all tools to the real human ONT cDNA, ONT dRNA[22], and PacBio public datasets, for which the ground truth is indeed unknown. We used gffcompare to estimate the consistency of predictions by computing the number of identical transcript models reported by the different tools. On the human ONT cDNA dataset, IsoQuant shows the highest percentage of transcripts confirmed by at least 3 other methods (58.1%), while no other tool surpasses the 25% threshold. This suggests that IsoQuant transcript models are significantly more consistent with other methods (Figure 2b left). In comparison to the other approaches, IsoQuant also reports the lowest number of transcripts that are not predicted by any other method. If one interprets such transcript models as false positives, IsoQuant again stands out in the lowest false-discovery rate (7.7%, 2,655 transcripts). In contrast, other tools output annotations containing more than 50% of unconfirmed transcript models (varying from 44,000 to 130,000). Additionally, for each tool we computed the number of potentially missed transcripts that were reported by all other methods. While Bambu has the lowest number of such transcripts (574), IsoQuant shows the second-best results of 1507 possible false negatives (Supplementary Table 5).

Similar trends can be observed in ONT dRNA and PacBio datasets, although the overall percentage of common transcripts appears to be higher compared to ONT cDNA data (Figure 2b, middle and right). IsoQuant again shows the highest fraction of transcripts predicted by at least 3 other tools (79.8% for ONT dRNA, 65.9% for PacBio), while other programs have 40% at best. Of the others, all four tools produce annotations containing more than 30% of transcripts that are not confirmed by any other method, while IsoQuant's potential false predictions are below 5% on both datasets.

Although these values cannot be explicitly treated as false positives and false negatives, they advocate that unlike other tools, IsoQuant produces highly specific annotations that are strongly consistent with transcripts reported by several alternative approaches. Moreover, because IsoQuant typically misses very few isoforms predicted by all other tools simultaneously, it is likely to also be highly sensitive.

Additionally, we used long-read RNA sequencing data from a mouse brain sample, wherein a previous study reported 76 novel isoforms of high biological importance[23], which were confirmed by manual annotation by the GENCODE team. Here, we compared IsoQuant only with StringTie, which has the second-best F1-score across all simulated datasets. On PacBio data, IsoQuant correctly reconstructs 71% of the confirmed novel isoforms, while StringTie restores half as many novel transcripts — only 37% (Supplementary Table 6). Similarly, on two ONT datasets (single-cell and spatial) from the same brain sample IsoQuant restores up to 50% of these 76 novel isoforms, whereas StringTie reports 30% at best. Although it is not possible to evaluate specificity in this kind of experiment, it confirms that IsoQuant is capable of maintaining high recall values on real sequencing data.

Beside transcript discovery, IsoQuant implements additional functionality, such as read-to-isoform assignment and transcript quantification. Benchmarks of these supplementary features, information on computational performance, as well as IsoQuant results obtained with different spliced aligners can be found in the Supplementary Notes 2-5.

In summary, IsoQuant accurately predicts transcript models from PacBio or ONT RNA sequencing data. For known isoforms, IsoQuant has higher F1-score compared to other tested tools, but these differences are not dramatic. For unannotated isoforms, however, IsoQuant provides very strong increases in F1-score over other existing approaches. In comparison to most tools, it achieves this F1-score increase by maintaining higher recall, while substantially increasing precision. Thus, IsoQuant is a valuable tool for predicting novel alternatively spliced isoforms in the age of long-read sequencing.

References

1. Dobin, A. et al. *Bioinformatics* 29, 15–21 (2013).
2. Li, H. *Bioinformatics* 34, 3094–3100 (2018).
3. Liu, B. et al. *Genome Biol.* 20, 274 (2019).
4. Sahlin, K. & Mäkinen, V. *Bioinformatics* 37, 4643-4651 (2021).
5. Kovaka, S. et al. *Genome Biol.* 20, 278 (2019).
6. Tung, L.H., Shao, M. & Kingsford, C. *Genome Biol.* 20, 287 (2019).
7. Wyman, D. et al. *BioRxiv,* https://doi.org/10.1101/672931
8. Tang, A.D. et al. *Nat. Commun.* 11, 1438 (2020).
9. Kuo, R.I. et al. *BMC Genomics* 21, 751 (2020).
10. Byrne, A. et al. *Nat. Commun.* 8, 16027 (2017).
11. Chen, Y. et al. *Bioconductor*, https://doi.org/doi:10.18129/B9.bioc.bambu
12. Tardaguila, M. et al. *Genome Res.* 28, 396-411 (2018).
13. de la Fuente, L. et al. *Genome Biol.* 21, 119 (2020).
14. Reese, F. & Mortazavi, A.doi:10.1101/2020.06.09.143024
15. Stein, A.N. et al. *BioRxiv,* https://doi.org/10.1101/2022.04.14.488347
16. Sahlin, K. & Medvedev, P. *Nat. Commun.* 12, 2 (2021).
17. Nip, K.M. et al. *Genome Res.* 30, 1191–1200 (2020).

18. Pardo-Palacios, F. et al. *In Review,* https://doi.org/10.21203/rs.3.rs-777702/v1
19. Mikheenko, A. et al. *Genome Res.* 32, 726–737 (2022).
20. Hafezqorani, S. et al. *Gigascience* 9, (2020).
21. Pertea, G. & Pertea, M. *F1000Res.* 9, (2020).
22. Workman, R.E. et al. *Nat. Methods* 16, 1297–1305 (2019).
23. Joglekar, A. et al. *Nat. Commun.* 12, 463 (2021).

Methods

Data simulation. To simulate PacBio CCS reads we used IsoSeqSim (https://github.com/yunhaowang/IsoSeqSim), which generates a read by truncating a transcript sequence according to given probabilities and randomly inserts sequencing errors at a specified rate with uniform distribution. As reported in previous studies[24], a uniform error distribution is a realistic model for PacBio CCS reads. Here we used 5' and 3' truncation probabilities typical for PacBio Sequel II (provided within the package) and an overall error rate of 1.6%: 0.6% deletions, 0.6% insertions and 0.4% substitutions. While these discrepancies do not necessarily represent sequencing errors, they nevertheless have to be modeled, as they can confuse transcript reconstruction. The above values were obtained by mapping real PacBio CCS reads to the reference genome[18].

ONT reads were simulated with the NanoSim software in the transcriptome mode[20]. NanoSim is designed specifically for simulating ONT-specific sequencing errors and biases. It first constructs error-profile and length-distribution models, which are further used to mutate reference transcript sequences. Here we used a pre-trained model provided within the NanoSim package, which was obtained using publicly available ONT cDNA data[22] from NA12878 human cell line and has an average error rate of 15.9%: 6% deletions, 5.1% insertions, 4.8% substitutions. In addition, we turned off the simulation of intron retention events and random unaligned reads representing the background noise.

However, additional analysis of the simulated ONT data and NanoSim code revealed that NanoSim randomly selects a start position of a read in a transcript sequence with a uniform distribution, thus introducing no 5' or 3' bias. To simulate more realistic ONT reads, we aligned real ONT cDNA data obtained from the mouse brain sample to the reference transcriptome using minimap2 and derived empirical truncation probability distributions on both 5' and 3' ends. Further, we changed the NanoSim source code to enable sequence truncation with respect to obtained probabilities (Supplementary Figure 2). The modified version is available at https://github.com/andrewprzh/lrgasp-simulation.

For both ONT and PacBio simulation we used Mouse GENCODE v26 and Human GENCODE v36 basic annotations[25]. Prior to simulation, we also attached a 30 bp polyA tail to every transcript sequence. To simulate realistic mouse data, a transcript expression profile was obtained using PacBio data from a mouse brain sample[23]. For human data, a gene expression profile was computed with PacBio GM12878 data. Complete description of every dataset used in this study is provided in the Supplementary Table 7.

Sequencing Lexogen SIRV transcripts. First, total RNA from HeLa cells was extracted using the miRNeasy Tissue/Cells Advanced Mini Kit (Qiagen, 217604), and polyA transcripts were pulled-down using the NEBNext Poly(A) mRNA Magnetic Isolation Module (NEB, E7490S).

Next, the SIRV-Set 4 (Iso Mix E0 / ERCC / Long SIRVs) (Lexogen, 141.01) was spiked-in to the RNA and reverse transcribed using the Maxima H Minus RT (Thermo Scientific, EP0752). The RT reaction final concentrations are as follows: 1.25 ng/µl polyA HeLa RNA, 0.33 ng/µl SIRV-Set 4, 0.5 mM dNTP, 5 µM dT-VN oligo, 5 µM TSO, 1X RT Buffer, 2 U/µl RiboLock RNase Inhibitor (Thermo Scientific, EO0382) and 20 U/µl Maxima H Minus Reverse Transcriptase. The reaction was incubated for 30 minutes at 50°C and 5 minutes at 85°C. Then, 5 µl of RT reaction were amplified using the Platinum Superfi II Mastermix (ThermoFisher, 12368010) for 12 cycles, according to manufacturer instructions and using Forward- and Reverse-Amplification primers. Finally, the cDNA was cleaned up using SPRIselect beads at a 0.8x ratio (Beckman Coulter, B23318) and used as input for Amplicons by Ligation library preparation (Oxford Nanopore Technology, SQK-LSK110). Sequencing was run for 72 hours on a R9.4.1 MinION flowcell (ONT, FLO-MIN106D).

Quality evaluation of predicted novel transcripts. To mimic real-life situations and assess the ability of an algorithm to predict novel transcripts, we created reduced gene annotations by removing a fraction of expressed isoforms. First, we define a subset of true expressed transcripts that contributed to at least one read during the simulation. Among this set, we select a fraction of transcripts to be excluded from the annotation. These transcripts are denoted as the true novel isoforms. The remaining transcripts (among the expressed) are defined as true known isoforms. To create a reduced gene annotation we remove all true novel isoforms from the comprehensive GENCODE annotation. Here we created a reduced mouse annotation with 15% of expressed transcripts removed, and four human reduced annotations with 10%, 15%, 20% and 25% of excluded expressed isoforms (Supplementary Note 1).

To evaluate a transcript prediction tool, we provided the entire set of simulated reads and the reduced annotation as an input. Thus, true novel isoforms are hidden from the annotation, but present in the reads. We then compute precision and recall by running gffcompare[21] for (i) the entire output annotation vs. the complete set of expressed transcripts, (ii) reported known isoforms vs. the set of true known isoforms, and (iii) predicted novel transcript models vs. the true novel set. The information on whether a transcript is known or novel is obtained from the output GTF file. The script for computing these metrics can be found in the IsoQuant repository in misc/reduced_db_gffcompare.py.

To estimate how recall and precision of novel transcripts depend on the expression levels, predicted transcripts are grouped into bins by their TPM values. For computing recall, TPM values used during the simulation are used as the number of false negative calls is required (undetected transcripts). However, computing precision requires the number of false positive predictions within each bin and thus only reported TPM values can be used (the true TPM for a false prediction is 0). Thus, it may happen that the same transcript may fall into different bins when benchmarking different tools. Although it is not possible to compute precision, recall and F1-score exactly for an

arbitrary TPM range, the bias has a minor effect as only a small number of bins was used in this experiment (5). Therefore, despite being imperfect, these estimations can provide additional insights on whether a transcript discovery method has any bias towards highly- or lowly-expressed isoforms.

To evaluate SIRV transcripts we used an incomplete SIRV annotation containing only 43 out of 69 SIRV transcripts. The output annotations were again split into known and novel transcripts and compared against the respective reference set using gffcompare. The SIRV Set 4 annotations are available at https://www.lexogen.com/sirvs/download/.

Estimating consistency between annotations. Consistency between transcripts generated on real data was estimated using gffcompare (without providing a reference annotation). Based on gffcompare output, for each tool we computed how many of its transcripts are supported by (i) all 4 other tools, (ii) exactly 3 other tools, (iii) 1-2 other tools and (iv) no other tool (possible false predictions). We also counted the number of potentially missed transcripts that were reported by all methods except the one being evaluated (possible false negative). This approach is implemented in misc/denovo_model_stats.py.

Command line options. For PacBio data minimap2 was launched with "splice:hq" preset; for ONT data we used k-mer size 14 with the usual "splice" preset. We also provided annotated splice junctions in BED format as an input. In each experiment, all tools were provided with the same BAM file and the same reference annotation. IsoQuant was launched with the default parameters setting the appropriate data type via "--data_type" option. StringTie2 was launched with the "-L" option. All other tools were run with the default parameters in 20 threads. In contrast to all other tools, Bambu outputs all reference transcripts, including unexpressed ones. Thus, we filtered out all transcripts with read count values < 1 from the Bambu output. As recommended in the user manual, we also ran TALON using preliminary alignment correction with TranscriptClean[30] (https://github.com/mortazavilab/TALON). However, as the results with and without correction were almost identical, we decided to use the annotations obtained from raw data for a fair comparison. Complete information on all options and software versions are provided in the Supplementary Table 8.

IsoQuant algorithm. To process long RNA reads, IsoQuant requires a reference genome and a corresponding gene annotation. If the reads are provided in the FASTQ format, IsoQuant maps them to the reference with minimap2 in splice mode[2]. Alternatively, a user may provide a sorted and indexed BAM file generated with a spliced aligner of their choice. The IsoQuant algorithm consists of 4 main steps: (i) assigning mapped reads to known isoforms, (ii) transcript quantification, (iii) alignment correction and (iv) transcript model construction. Below we describe the key aspects of all 4 procedures.

IsoQuant algorithm: assigning long reads to known isoforms. The algorithm for assigning long reads to annotated isoforms is based on intron chain matching and detecting exonic overlaps. To assign reads, IsoQuant processes each gene individually by extracting reads that map to the respective region from the sorted BAM file.

IsoQuant first processes the annotation to construct intron and exon profiles of all known isoforms. A set of annotated introns in the gene is sorted according to their coordinates in the genome and enumerated from 1 to N. Thus, an annotated isoform can be represented as a vector of length N, in which the element at position *i* is set to 1 if this isoform includes the i-th intron and -1 otherwise (Supplementary Figure 3a). This vector is henceforth referred to as an isoform intron profile. The exon profile is constructed in a similar manner: all annotated exons are first split into a minimal set of M non-overlapping fragments, such that every exon can be represented as their combination, and these exonic fragments are sorted and enumerated. The exon profile for an annotated isoform is similarly denoted as a vector of length M, where the i-th element is set to 1 if this isoform contains the i-th exon fragment and -1 otherwise (Supplementary Figure 3b).

To assign a read to an annotated isoform, each intron from the alignment is matched against annotated introns from the current gene and a read intron profile is constructed (also a vector of length N). In this vector the i-th element is set to 1 if the annotated intron with index i matches to an intron from the read, -1 if it is overlapped or spanned by the read, but no match is detected, and 0 otherwise. A zero value indicates that the intron is located outside of the alignment region and therefore no information can be derived, e.g. due to read truncation. Similarly, the exon profile of the read is constructed based on M exonic fragments described above: 1 indicates that the respective exonic fragment is overlapped, -1 means it is spanned, and 0 is set for exonic fragments outside of the alignment region (Supplementary Figure 4).

Due to sequencing errors, an aligner may detect splice site positions inaccurately[19]. To avoid considering them as alternative or novel, the algorithm allows a small difference Δ between annotated and alignment splice site coordinates when matching introns. Formally speaking, an annotated intron (x_1, x_2) matches a read intron (y_1, y_2) if $|x_1-y_1| \leq \Delta$ and $|x_2-y_2| \leq \Delta$. The default Δ value varies for different types of input data: 4 bp used for PacBio CCS reads and 6 bp for ONT reads (can be set manually). Although an aligned read can be assigned to an isoform by simply comparing its intron chain and exonic coordinates to the annotation, vectorizing the alignment as described above allows one to easily implement inexact splice site comparison with a delta, and quickly detect candidate isoforms for read assignment.

Further, to assign a read to an isoform, its exon and intron profiles are matched against the respective profiles of the annotated isoforms. The distance between two profiles is computed simply as the number of distinct elements in which the read profile has non-zero values. A read is said to be consistent with an isoform if the distances between their exon and intron profiles are 0,

and the read has no unannotated introns/exons (Supplementary Figure 4). When a read is consistent with a single isoform, it is reported as a unique match. When a read is consistent with multiple isoforms simultaneously, it is classified as ambiguous, which may happen, for example, due to read truncation. If a read contains unannotated introns/exons, or its profiles are not consistent with any isoform, it is marked as inconsistent. For such alignments IsoQuant reports the most similar reference transcript and detected alternative splicing events.

Some inconsistencies can be, however, caused by misalignments, rather than by real alternative splicing events[17]: (i) skipped short exons, (ii) intron shifts exceeding Δ bp and (iii) short unannotated exons at transcript ends (Supplementary Figure 5). If an inconsistent alignment contains only these types of discrepancies, the read is reclassified as conditionally consistent.

IsoQuant algorithm: transcript quantification. Once long reads are assigned to annotated isoforms, quantification becomes rather trivial. Uniquely assigned reads are counted as a single detected transcript, while ambiguous reads are treated as multi-mappers and contribute to multiple assigned isoforms with lower weight. A transcript is reported as expressed only if it has at least one uniquely assigned read. Inconsistent reads are considered as potential novel isoforms and ignored during the quantification step. Beside genes and transcripts, IsoQuant can also count inclusion and exclusion abundances for separate exons and introns, which can be useful for computing percent-spliced-in (PSI) values.

IsoQuant implements additional functionality for barcoded long RNA reads, e.g. barcoded by single-cell or spatial location[23,26]. A user can provide information on how the reads are grouped, for example, as a TSV file that indicates a barcode or a cell type of origin for every read. Isoform and gene abundances are then calculated for every read group separately, which can facilitate an expression comparison between different groups or cell types.

IsoQuant algorithm: spliced alignment correction. IsoQuant corrects each uniquely assigned read individually. If a read contains misalignments described above (Supplementary Figure 5) or its intron chain is not identical to the intron chain of the assigned isoform, the alignment is corrected as follows. Short skipped exons are restored according to the annotation and minor intron shifts are replaced with the respective introns from the assigned transcript. Unannotated terminal micro-exons are simply removed from the alignment. Finally, any unannotated splice site is substituted with the nearest site from the assigned transcript if (i) these splice sites are located within Δ bp and (ii) read alignment contains sequencing errors near this splice site. Coordinates of corrected alignments are then saved in BED12 format.

IsoQuant algorithm: transcript model construction. The transcript reconstruction procedure implemented in IsoQuant includes 4 steps: (i) intron graph construction from read alignments, (ii) intron graph simplification, (iii) attaching terminal vertices, (iv) construction of paths representing

full-length transcripts. Below we provide a detailed description of all algorithms and intuition behind them.

Intron graph construction. To construct transcript models IsoQuant implements a concept of intron graph, which was influenced by the previously designed splice graph approach[27], used, for example, in StringTie[5]. For a given set of transcripts, an intron graph is constructed as follows. First, we define internal vertices as a set of all introns from all transcripts. Formally, each internal vertex is denoted by an ordered pair of coordinates in the genome. Two vertices are connected with a directed edge if the respective introns are consecutive in any transcript. Finally, for every first/last intron in a transcript, the corresponding vertex is connected with a terminating vertex that represents the transcript start/end position (formally, a single integer). Intron graph is a directed acyclic graph since every edge connects only consecutive elements. Each transcript can now be represented as a path in the graph that traverses from the initial to terminal vertex (Supplementary Figure 6a).

The described approach can be used to construct an intron graph from read alignments. Similarly to the read-to-isoform assignment procedure, the genes are processed by IsoQuant individually. First, the algorithm constructs a set of internal vertices corresponding to introns from the selected alignments. Two vertices are likewise connected when the respective introns are consecutive in any read alignment. Due to the presence of inexactly detected splice sites, which may remain even after the alignment correction, such a graph may contain false vertices and connections. These false nodes typically form topological patterns, such as tips and bulges. A tip is defined as a dead end (dead start) edge that has a starting (ending) vertex with outdegree (indegree) at least 2. A bulge consists of two alternative paths having the same start and end vertices (Supplementary Figure 6b). Similar patterns are also typical for de Bruijn graphs, which are used for short read assembly, where bulges and tips are caused by sequencing errors. To remove tips and bulges assemblers exploit various techniques broadly called graph simplification[28,29].

Intron graph simplification. Here we implement a graph simplification procedure based on the following observations: (i) a false intron is typically unannotated, (ii) splice site shifts that cause a false intron are short, and (iii) the number of reads supporting the correct intron often exceeds read support of a false intron. Formally, a bulge/tip is removed from the graph if it represents an unannotated intron that has at least twice lower read support compared to the alternative path and there is a known intron with splice sites within 20 bp (10 bp for PacBio). In other cases, when an unannotated intron has a high read support or no similar known intron exists, a bulge or a tip is likely to represent a part of a novel isoform and thus should be preserved (Supplementary Figure 6b). Although intron graph simplification strongly resembles naive intron clustering, it has an important difference: an intron is removed not only based on its properties, such as splice site positions and read support, but based on the graph topology as well, thus taking into account adjacent introns. Such a method allows one to, for example, preserve similar introns from distinct

isoforms. It is worth noting, that the simplification procedure keeps track of all collapsed tips and bulges, thus preserving the possibility to later traverse alignment containing removed introns though the graph.

Collecting terminal positions. After the graph is simplified, the algorithm proceeds to attach starting and terminal vertices. In contrast to annotated transcripts, read alignments do not provide the exact terminal positions, as their sequences can be truncated. Thus, to avoid having an extreme number of terminal vertices, terminal positions are detected using the heuristics presented below. Without loss of generality here we assume that the gene of interest is on the forward strand and polyA tails are on the right.

For every intron V in the graph the algorithm selects only read alignments that contain V as a terminal intron and processes them as follows. First, the polyA sites are collected and clustered. Clustered polyA positions $\{p_1, \ldots, p_k\}$ are added to the graph as terminal vertices and connected to vertex V (Supplementary Figure 7a). Further, the algorithm adds the rightmost non-polyA terminal position P as a terminal vertex if one of the conditions is satisfied: (i) V has no outgoing edges, (ii) V has an outgoing edge to an intron (u_1, u_2) and $P > u_1 + \Delta$, or (iii) V has adjacent polyA vertices $\{p_1, \ldots, p_k\}$ and $P > \max(p_1, \ldots, p_k) + \Delta$ (where Δ is the parameter defined above). Thus, a non-polyA terminal position can only be attached if it is located to the right of adjacent exons or polyA vertices. Starting positions are collected in a similar manner, but, indeed, without looking for polyA sites (Supplementary Figure 7b). The described approach, however, may lose information when several isoforms share the same starting intron but have distinct TSS/TES. Thus, we also apply an additional transcripts correction, which is described below.

Transcript discovery via path construction. Once the intron graph is constructed and simplified, IsoQuant detects full-length paths that connect starting and terminal vertices. Paths entirely supported by at least a single read alignment (i.e., full splice match) are marked as transcript prediction candidates (Supplementary Figure 7c). To filter out unreliable novel transcripts IsoQuant applies read support cutoffs: at least 5 full-splice match reads (3 for PacBio) and at least 2% from the maximum graph coverage. Since some isoforms may not have a full-splice matching alignment, IsoQuant also reports known transcripts that (i) have at least one uniquely assigned read and (ii) can be traversed through the intron graph. It also reports known mono-exonic transcripts that have (i) a uniquely assigned read and (ii) a confirmed polyA site.

To correct terminal positions of a novel transcript, the algorithm selects all alignments consistent with this transcript and uses them to extract terminal positions using the approach described above (Supplementary Figure 7d). In contrast to detecting terminal vertices for the entire graph, where all alignments are used, the subset of consistent reads likely belongs specifically to this isoform and thus provides correct start and end positions. The resulting transcripts are saved in GTF format, providing additional information about transcript types and their reference genes.

Data availability

Nanopore sequencing data obtained from the human NA12878 cell line is available at https://github.com/nanopore-wgs-consortium/NA12878/blob/master/RNA.md. PacBio human GM12878 data is available at ENCODE (https://www.encodeproject.org/search) under the accession numbers ENCFF450VAU and ENCFF694DIE. Sequencing data obtained from mouse brain samples is available at NCBI Gene Expression Omnibus (GEO; https://www.ncbi.nlm.nih.gov/geo/) under accession numbers GSE158450 and GSE178175. ONT SIRV data, simulated data and reduced gene annotations are available at https://data.cab.spbu.ru/index.php/s/dgcaSaGME2xF7ed?path=%2FIsoQuant. Complete information on all datasets used in this work is listed in the Supplementary Table 7.

Code availability

IsoQuant and the supplementary scripts used for the evaluation are available at https://github.com/ablab/IsoQuant.

Acknowledgements

We thank Nanopore WGS consortium and Ali Mortazavi's lab at UCI for making the ONT and PacBio data publicly available. This work was supported by St. Petersburg State University, Russia (grant ID PURE 93023437 to A.M., A.S., A.L.L. and A.D.P.). Scientific research was performed at the Research park of St.Petersburg State University «Computing Center».

Author contributions

A.D.P., A.M. and A.S. designed and implemented the software. A.D.P., A.M. and A.J. performed the benchmarks. J.J. performed the sequencing experiments. A.D.P., A.L.L., and H.U.T. acquired funding and supervised the project. A.D.P., A.M., A.J. and H.U.T. wrote the manuscript.

Competing interests

The authors declare no competing interests.

References

24. Ono, Y. et al. *Bioinformatics.* 29, 119-221 (2013).
25. Frankish, A. et al. *Nucleic acids research* 49, D916-D923 (2021).
26. Gupta, I. et al., *Nature biotechnology* 36, 1197-1202 (2018).
27. Heber, S.et al. *Bioinformatics* 18, S181-S188 (2002).
28. Zerbino, D.R. & Birney, E. *Genome research* 18, 821-829 (2008).
29. Bankevich, A. et al. *Journal of computational biology* 19, 455-477 (2012).
30. Wyman, D. & Mortazavi, A. *Bioinformatics* 35, 340 (2019).

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- IsoQuantsupplementarymaterial.pdf