Accurate spliced alignment of long RNA sequencing reads

Kristoffer Sahlin[1] and Veli Mäkinen[2]

[1] Department of Mathematics, Science for Life Laboratory, Stockholm University, 106 91 Stockholm, Sweden
[2]Department of Computer Science, P. O. Box 68, Pietari Kalmin katu 5, 00014 University of Helsinki, Finland

Abstract

Long-read RNA sequencing techniques are quickly establishing themselves as the primary sequencing technique to study the transcriptome landscape. Many such analyses are dependent upon splice alignment of reads to the genome. However, the error rate and sequencing length of long-read technologies create new challenges for accurately aligning these reads. We present an alignment method uLTRA that, on simulated and synthetic data, shows higher accuracy over state-of-the-art with substantially higher accuracy for small exons. We show several examples on biological data where uLTRA aligns to known and novel isoforms with exon structures that are not detected with other aligners. uLTRA is available at https://github.com/ksahlin/ultra.

Introduction

The transcriptome has been identified as an important link between DNA and phenotype and is therefore studied in many biological and biomedical questions. For these analyses, RNA sequencing (RNA-seq) has established itself as the primary experimental method. Some of the most common transcriptome analyses using RNA-seq include predicting and detecting isoforms and quantifying their abundance in the data. These analyses are fundamentally underpinned by splice alignment of RNA-seq reads to genomes.

Because of this, a plethora of different alignment algorithms have been proposed for splice alignment of short-read RNA-seq, with some of the key algorithmic advances given in TopHat (Trapnell, Pachter, and Salzberg 2009), STAR (Dobin et al. 2013), HISAT (Kim, Langmead, and

Salzberg 2015), GMAP (Wu et al. 2016), and HISAT2 (Kim et al. 2019). While short-read RNA-seq has shown unprecedented insights into transcriptional complexities of various organisms, the read-length makes it difficult to detect some isoforms, which, consecutively, limits quantification at isoform level.

A particularly challenging task of splice alignment is alignment to short exons. Firstly, short exons are difficult for the aligner to index because their length makes them highly repetitive in the genome. Therefore, the aligner may not index them for practical reasons. Secondly, a small exon is less likely to have a match of sufficient length to the read because of its length and errors in the reads. The inability to align a read to short exons may cause downstream analysis tools to predict and quantify erroneous isoforms. In addition, we show in this study that splice aligners that use junction-specific alignment penalties can create spurious junctions by overfitting alignments to canonical splice sites such as GT-AG junctions. To this end, we have designed and implemented a splice alignment algorithm uLTRA to alleviate these limitations significantly. uLTRA aligns long-reads to a genome using an exon annotation.

We demonstrate using controlled datasets that uLTRA is more accurate than other aligners, particularly for short exons. We also use a dataset with ONT sequencing of synthetic SIRV transcripts (known isoforms) to demonstrate that uLTRA aligns more reads to transcripts that are known to be in the sample. Furthermore, we show on biological data from both PacBio and ONT that uLTRA aligns more reads to annotated isoforms and has alignments to more distinct isoform structures. Finally, we demonstrate that uLTRA produces alignments to known and novel isoform structures in the PacBio Alzheimer dataset that are not found by other aligners. These isoform structures come from genes that have been studied or linked to Alzheimer's disease and motivate the utility of our method for a range of downstream analysis tasks such as isoform prediction and detection, splice-site analysis, isoform quantification and more. uLTRA is available at https://github.com/ksahlin/ultra.

Results

Evaluation overview

We used three *in silico*, one synthetic, and two biological datasets (Table 1) to evaluate the alignment algorithms. Of the biological datasets, two were from ONT and one from the PacBio Iso-Seq protocol. We used simulated datasets with known annotations to investigate the accuracy of spliced alignments as a whole, and the performance of alignment accuracy of individual exons as a function of exon size. We used the synthetic SIRV data to investigate how aligners perform when aligning real sequencing reads to isoforms structures known to be in the sample. Finally, we measured the concordance in alignments between alignment methods on

the biological data where we do not have the ground truth annotations. We also demonstrate that relying on alignment concordance as a proxy for alignment accuracy can be misleading due to similar alignment biases between aligners. We also report runtime and memory usage.

Alignment accuracy

We used three *in silico* datasets to test the alignment accuracy in a controlled setting (Table 1). First, we used 234,207 distinct cDNA sequences downloaded from ENSEMBL (denoted ENS) without introducing any simulated errors. We then simulated a dataset of 1,000,000 reads uniformly at random from the 234,207 ENSEMBL sequences with a mean error rate of 8.6% (denoted SIM_ANN for simulated annotated transcripts). Finally, to test the ability to align to transcripts containing novel combinations of exons, we simulated a dataset with the same error mean error rate as SIM_ANN, that we call SIM_NIC for simulated Novel-In-Catalog transcripts. This dataset consists of reads from transcripts with novel exon combinations that we generated from GTF from gencode (release 34, including haplotype scaffold annotations). See supplementary note B for details on the simulations. Since we have the true exon annotation of each read, we classify the read alignments as correct, inexact, exon difference, incorrect location, and unaligned. For details of these classifications, see Supplementary note B.

For SIM_ANN, which contains simulated reads from annotated transcripts, uLTRA has the highest fraction of correct alignments with a 2.6% percentage point increase compared to the second-best performing tool deSALT_GTF (Fig. 1A). uLTRA also significantly reduces errors classified as exon differences compared to the other aligners. Furthermore, we observed that uLTRA achieves considerably higher accuracy than other aligners for small exons (Fig. 1B). We observed a similar trend for the ENS dataset (Fig. S1).

As for the SIM_NIC, which contains only reads with novel combinations of exons, uLTRA's accuracy is comparable to the ENS and SIM_ANN datasets (Fig. S2A). However, on this dataset uLTRA has a 9.6% percentage point more correctly aligned reads compared to the second-best performing aligner deSALT_GTF (Fig. S2A). Our results show that the accuracy is significantly lower for the other aligners across exons sizes on this dataset (Fig. S2B), which disputes the explanation that the decrease in accuracy is caused solely by a larger fraction of smaller exons in the SIM_NIC dataset. The decreased accuracy may be explained by the fact that transcripts are simulated from all the reference sequences present in the Gencode release 34 (including haplotype scaffold annotations). This simulation will, therefore, produce more reads from transcripts with similar gene copies. As uLTRA's accuracy remains similar to the other two datasets, it highlights the accuracy of aligning transcripts also to a reference genome with alternative haplotypes.

Splice site annotation performance on SIRV

We used the subset of 59 isoforms with distinct splice site positions from the ONT SIRV dataset (Sahlin et al. 2020) to investigate alignment performance around splice sites (for details see Suppl. Note C). In this dataset, as the sequenced isoforms are known, we have a complete isoform annotation. We computed all alignments that had perfect matching splice sites to the annotations and denoted these reads as Full Splice Matches (FSM) following the notation in (Tardaguila et al. 2018). With the SIRV dataset, we have the properties of real ONT sequencing errors and genes, each expressing several known isoforms. The downside with SIRV data is that it does not represent the sequence complexity of a genome. For minimap2 and deSALT, we used non-default alignment parameters not to penalize non-canonical splice sites as much as in biological data. After this modification, we observed significantly improved alignment performance over default parameters (for details, see Suppl. Note A). On this dataset, deSALT_GTF, GraphMap2, and GraphMap2_GTF exited with errors before producing output and could not be evaluated.

uLTRA aligned 1,092,311 reads as FSM (72%), while deSALT and minimap2 aligned 810,636 (54%) and 698,838 (46%) respectively. If a read aligns as FSM to the same isoform for two aligners, we say that the read is FSM-concordant between the two alignment methods. The majority of FSM reads (51.3%) are FSM-concordant between all the three methods (Figure 2). Furthermore, uLTRA shares 22.2% of the total number of FSM aligned reads with either one of the other two aligners. We also investigated the 1.5% of FSM reads where deSALT and minimap2 agreed on alignment to an FSM, but uLTRA did not. Out of these 16,214 reads, 79% of them were explained by a 1nt offset in the downstream splicing junction between two isoforms on SIRV6. For some of these reads, uLTRA had a higher alignment identity over the junction. In these cases, minimap2_GTF and deSALT fit the alignment to a GT-AG splicing pattern by placing an insertion near the junction.

Finally, the second largest category of FSM reads in Figure 2 are the reads that uLTRA aligned uniquely as FSM (21.7%). As uLTRA had a significantly higher number of reads aligning to FSMs, we wanted to investigate to which isoforms they were aligned. We looked at the number of reads aligning to FSMs broken down per gene region and isoform (Fig. S3). While sequencing bias may distort the read coverage per isoform and produce a dataset with different coverage distribution to what is present in the sample, the E0 mix contains transcripts at roughly equal abundances. We see that the coverage distribution of the number of aligned FSM reads of uLTRA closely follows the coverage of deSALT and minimap2_GTF but, in general, aligns more reads as FSM per isoform. A notable difference is that minimap2 only aligns FSM reads to 54 unique isoforms, even after employing specific alignment parameters for SIRV data (Suppl. Note A). In comparison, deSALT and uLTRA were able to align FSM reads to all 59 unique isoforms. Comparing the number of alignments to each isoform (Fig. S3), we observed no alignment bias in uLTRA towards a specific isoform.

Evaluating alignments on biological data

We neither have the correct read annotations, nor are we guaranteed to have a complete gene annotation for the biological datasets, which presents a challenge when evaluating accuracy. We took the following approaches. We first compared the aligned read categories between alignment algorithms following the definitions in (Tardaguila et al. 2018) (presented in the next section). Secondly, we looked at the alignment concordance between methods. Here we investigated concordance with respect to both alignment location on the genome and concordance in the categories of reads. Thirdly, we provide several examples of uniquely detected isoforms by uLTRA, which, by example, demonstrate that alignment concordance analysis to assess alignment accuracy has caveats.

Alignment categories on biological data

We evaluated alignment categories to the previously annotated database using the categories defined in (Tardaguila et al. 2018). As in (Tardaguila et al. 2018), we classify an alignment of a read to the genome as a full splice match (FSM), incomplete splice match (ISM), novel in catalog (NIC), novel not in catalog (NNC), or NO_SPLICE. An FSM alignment means that the combination of splice junctions in the read alignment has been observed and annotated as an isoform. An ISM alignment means that the combination of splice junctions is in the annotation, but it is missing junctions compared to the annotated models in either the 3’ or 5’ end. A NIC alignment means that all the individual splice sites of a read are in the annotation, but at least one of the splice junctions are not. An NNC alignment means that the read aligns with at least one junction that is not in the annotation, while NO_SPLICE are all alignments without splice sites. See (Tardaguila et al. 2018) for details regarding these definitions.

In (Tardaguila et al. 2018), the authors noted that a higher fraction of isoforms represented by FSM and NIC reads could be validated using orthogonal techniques compared to the NNC reads, where the large majority could be validated and may stem from sequencing artifacts or misalignments. As we neither know the true isoforms present in the samples nor have a complete annotation of all isoforms on the genome, comparing the alignment categories between alignment methods does not evaluate alignment performance. Nevertheless, the categories can be compared between aligners for general insight on alignment concordance. Furthermore, these alignment categories are important for various downstream isoform detection and classification methods such as SQANTI (Tardaguila et al. 2018), TAMA (Kuo et al. 2019), or TALON (Wyman et al., 2020). Therefore, we present the results here.

We observed that uLTRA aligns more reads as FSM and fewer reads as NNC than other aligners for both the DROS and ALZ dataset (Fig. 3). We also observe that deSALT_GTF, similarly to uLTRA, has a notable increase in FSMs, and a decrease in NNC compared to the

rest of the alignment methods. Notably, uLTRA and deSALT_GTF are also the two best performing aligners on the simulated datasets (Fig. 1, Fig S1, S2). Furthermore, uLTRA had FSM alignments to more unique isoforms (39,173) compared to deSALT_GTF (36,013) and minimap2_GTF (34,294) in the ALZ dataset. We observed this trend also in the DROS dataset, where uLTRA had FSM alignments to 13,939 unique isoforms compared to deSALT_GTF (13,361) and minimap2_GTF (13,030). However, uLTRA also aligns a higher fraction of NIC reads and has a high fraction of unaligned reads on the ALZ dataset (Fig. 3B). These results, therefore, prompted further analysis of alignment concordance between alignment methods.

Alignment concordance on biological data

We furthermore investigated concordance in alignments between the three best performing tools uLTRA, deSALT_GTF, and minimap2_GTF (based on the evaluations on simulated data). First, we investigate the large fraction of unaligned reads that uLTRA has compared to the two other methods in the ALZ dataset. It is known that a substantial fraction of reads in long-read transcriptome sequencing data is coming from so-called intra-priming reads (Tardaguila et al. 2018). These reads are characterized by aligning without splice junctions (i.e., category NO_SPLICE) to an unannotated genome location that contains a poly-A stretch downstream from their 3′ end. While not fully characterized, these reads are likely to be artifacts in the sequencing protocol and often filtered out in downstream analysis (Tardaguila et al. 2018). As uLTRA will only align reads to and around annotated regions (see methods), these reads should, in the best-case scenario, be either unaligned or aligning to genomic regions flanking exon regions as category NO_SPLICE. In the worst case, they will be spuriously aligned to an annotated region. We wanted to investigate alignment concordance while keeping this limitation in mind. Therefore, we compared alignment results on the biological data by first categorizing reads as either *exonic* or *genomic,* based on the following criteria. We classify as a *genomic read*, a read that has a primary alignment, produced by both deSALT_GTF and minimap2_GTF, that does not overlap with any annotated exonic region. A read is classified as exonic if it is not genomic. As uLTRA should not align genomic reads from genomic regions, we then analyzed genomic and exonic reads separately.

For the ALZ dataset, 753,072 reads (17.6%) were classified as genomic. Out of those reads, uLTRA assigned 92.2% as unaligned, 7.2% as NO_SPLICE, and 0.6% (0.1% of the total number of reads) as any of the remaining categories (FSM, NIC, or ISM). For the DROS dataset, only 28,645 reads (0.8%) were classified as genomic. Out of those reads, uLTRA assigned 66.7% as unaligned, 32.2% as NO_SPLICE, and 1.1% (0.0% of total number of reads) as any of the remaining categories (FSM, NIC, or ISM). This indicated that uLTRA did not align a large fraction of these reads spuriously to exonic regions. This analysis suffers from the caveat of not knowing the correct locations of reads, while also relying on deSALT_GTF and minimap2_GTF to classify reads as genomic.

For the reads classified as exonic reads, we first looked at a relaxed measure of alignment concordance. We define a read to have *concordant alignments* between two methods if the two alignments have a non-zero overlap on the genome (based on the start and stop coordinates). Note that this definition only captures discordance if the read aligns to different genes and not smaller differences around exons. A caveat with this definition is when measuring alignment concordance between more than two aligners. An alignment spanning positions A to C in one alignment may overlap with two disjoint alignments A to B and B+1 to C. In this case, we treat all the alignments as discordant. Finally, there are genes with multiple identical copies on the genome. In these cases, the alignment methods may choose different alignment locations simply by randomly picking a location. With these limitations in mind, we observed 88.8%, and 92.3% of all aligned reads had concordant genomic positions in DROS and ALZ, respectively (Fig. S4). The second largest category was the alignment concordance between deSALT_GTF and uLTRA for the DROS dataset (5.7%), and the alignment concordance between deSALT_GTF and minimap2_GTF for the ALZ dataset (5.4%).

We also looked at exact concordance for reads classified as FSMs, ISMs, NICs and NNC, and NO_SPLICE individually, because of the somewhat approximate metric of alignment concordance. For FSMs, ISMs, NICs, the splice sites are known, and we compare exact matches in splice sites. For NNC and NO_SPLICE, we use genomic overlap as described above. In general, We observed a large concordance of alignments in the categories FSM, ISM, and NO_SPLICE in both DROS and ALZ (Fig. S5-6), but low concordance for NIC and NNC. The NIC and NNC categories contain fewer reads (Fig. 3).

Finally, we also looked in more depth at the concordance of unique isoforms that had FSM predictions (Fig. S7). We observed a large concordance in the predicted FSMs between all methods on both datasets (90.5% and 80.3% for DROS and ALZ respectively), with the second and third largest categories being the isoforms predicted by either both uLTRA and deSALT (3.8% and 5.7%) or uniquely by uLTRA (3.3% and 8.7%) for DROS and ALZ respectively.

Caveats of assessing alignments based on concordance

We further investigated some of uLTRA's FSM and NIC alignments uniquely detected by uLTRA, and where deSALT_GTF and minimap2_GTF agreed on another structure. We observed that in several cases, uLTRA's alignments were correct. This analysis highlights that alignment concordance between aligners may not indicate correct alignment as concordance can come from the same algorithmic decisions between aligners such as customized alignment penalties for canonical regions or inability to align to very short exons.

FSM isoforms uniquely found by uLTRA

On the DROS dataset, uLTRA aligned a total of 2,316 FSM reads to 470 distinct isoforms (3.4% of total distinct isoforms) that minimap2_GTF and deSALT_GTF did not align to. Of these isoforms, 24 had more than ten reads aligned, while most other isoforms had a coverage of 1-10 reads (Fig. S8A). On the ALZ dataset, uLTRA aligned a total of 15,720 FSM reads to 3,492 distinct isoforms (8.7% of total distinct isoforms) that minimap2_GTF and deSALT_GTF did not align to. A total of 154 of these isoforms had more than 10 reads aligned, while most other isoforms had a coverage of 1-10 reads (Fig. S8).
We manually inspected a subset of the more abundant uniquely predicted isoforms for the ALZ dataset using IGV (Robinson et al. 2011). We observed that some of these isoforms contained small exons (<10nt) that appeared correctly aligned to by uLTRA (Fig. S9). However, deSALT_GTF and minimap2_GTF agreed on a different splicing structure, missing to align the small exon. These alignments would show up as concordant between deSALT_GTF and minimap2_GTF in our previous analysis, although they are not likely to be correct. The isoforms in Figure S9 come from the genes AP2, APBB, HNRNPM, and DCTN2, which come from gene families that have appeared in studies related Alzhemers's disease (Tian et al. 2013) (Tanahashi and Tabira 1999) (Geuens, Bouhy, and Timmerman 2016) or other neurodegenerative disorders (Boland et al. 2018). All of these genes are supported by more than 100 reads and have perfect alignment across the junctions.

In addition, we highlight another case of a potential subtle misalignment (Fig. S10) that makes the best fit FSM isoform go undetected. This potential misalignment is caused by using GT-AG specific alignment penalties and causes 530 reads to support a GT-AG splice junction in deSALT_GTF and minimap2_GTF. In this example, uLTRA's alignments support a GC-AG junction. While we have no ground truth, and an insertion of one nucleotide near the splice site is plausible, uLTRA's alignments best fit the data (omitting prior belief of GT-AG junction) and also supports a previously annotated isoform. The PRNP gene has also been studied in Alzheimer's disease (Bagyinszky et al. 2019).

Finally, we illustrate an example (Fig. S11) of an instance of 176 reads where all three aligners have discordant alignments, caused by a segment of 9nt from a transcript from the SPOCK gene, which has also appeared in studies on neurodegenerative disorders (Charbonnier et al. 1997). Here, uLTRA and deSALT_GTF align the 9nt portion of the read corresponding to two different exons while minimap2_GTF does not align this region. Both uLTRA and deSALT_GTF alignments are FSM but to different isoforms, and both upstream and downstream junctions are GT-AG. With this information, it is ambiguous as to which alignment is the correct one.

NIC isoforms uniquely found by uLTRA

We also observed that uLTRA aligns more reads as NIC alignments compared to deSALT_GTF and minimap2_GTF (Fig. 3, Fig. S5C, and S6C). Many of these NIC reads could be spurious due to inaccuracies in uLTRA alignments when both upstream and downstream flanks of the junction contain the same nucleotide. We looked at the most abundant NIC uniquely aligned to by uLTRA, a transcript from the MBP gene (Fig. S12; predicted by 1208 reads). uLTRA aligned reads to this NIC because of the homopolymer length difference of C's in the reads, together with that both the upstream and downstream junction contained C's. However, deSALT_GTF and minimap2_GTF always aligned to the CT-AC junction by creating insertions of C at downstream junctions if needed (matching an FSM), while uLTRA chooses a CT-TA junction for the reads where the homopolymer length was four cytosine nucleotides (creating a NIC). It is ambiguous as to what is the correct isoform in this example.

As for the uniquely predicted FSMs, the NICs also contained predictions with small exons (in total 50 unique NIC isoforms have exons smaller than 20nt). We manually inspected some of the more abundant predictions of which, similarly to the FSMs with small exons, the data supports their correctness (Fig. S13 A-C). The three isoforms presented in Fig. S13 are highly supported isoforms from the MICU1, SEPTIN7, and APBB1 genes and are, furthermore, novel with respect to the Gencode v34 annotation and have appeared in studies on Alzheimer's disease (Wang et al. 2018) (Calvo-Rodriguez et al. 2020) (Tanahashi and Tabira 1999).

Runtime and memory usage

We used a 512Gb memory node with 64 cores to evaluate the performance of tools. We ran uLTRA, Graphmap2, and minimap2, with 62 cores and deSALT with 48 cores (the largest permitted number of cores for deSALT).

Across our datasets, minimap2 was significantly faster on both indexing and alignment than all the other tools (Table 2). When comparing the two best performing tools in terms of accuracy (uLTRA and deSALT_GTF) uLTRA is slower than deSALT_GTF on all the datasets in our evaluation except the ALZ dataset, the largest one. On this dataset, uLTRA is almost twice as fast.

uLTRA's index takes about 6Gb for human (with parameter settings used in this paper), and each instance (if parallelized) needs a separate copy of the index. So parallelization for computers or clusters with 8Gb per core is particularly straightforward. Our predictions of uLTRA's memory consumption is 372Gb (6Gb multiplied by the number of cores), which is in rough agreement with experimental data (Table 2). The slight increase in peak memory usage (particularly for the ALZ dataset) occurs because of (i) memory consumption of the main process keeping the reads in memory and (ii) memory consumption of the dynamic programming alignment matrix in parasail (Daily 2016), particularly for the longest reads.

On HG38, we observed that roughly 2Gb of uLTRA's data structures stored in memory were occupied by data structures for the segments, flanks and, and parts. The remaining 4Gb was occupied by the tiled data structures used only for a smaller subset of poorly fit reads (details in Methods). Therefore, an alternative that would decrease uLTRA's memory usage with almost 70% would be to compute tilings on the fly for each read that requires it. Such implementation may slow down runtime as the tiling would be recomputed for each read that needs it.

For indexing, uLTRA and minimap2 are relatively fast, while deSALT is slower (Table 3). uLTRA used the smallest amount of memory (Table 4), which is not surprising as it is processing a smaller region of the genome.

Methods

Overview

uLTRA solves the algorithmic problem of chaining with overlaps to find alignments. The method consists of three steps. An overview of uLTRA is shown in Figure 4. We first construct subsequences of the genome referred to as *parts*, *flanks*, and *segments* (Fig 4; details in Step 1 below). This step is similar to the indexing step in other alignment algorithms, where the data structures do not need to be reconstructed for new sequencing datasets.

To align reads, uLTRA first finds maximal exact matches (MEMs) between the reads and the parts and flanks using slaMEM (Fernandes and Freitas 2014) (Fig. 4). Each read will have a set of MEMs to the genome reference sequences (e.g., a set of chromosomes). Furthermore, we partition the instances within chromosomes if two consecutive MEMs on the chromosome are separated by more than a parameter threshold provided to uLTRA. For each instance, uLTRA finds a collinear chain of MEMs covering as much of the read as possible (allowing overlaps of MEMs in the read). We use Algorithm 1 in (Mäkinen and Sahlin 2020) to find such optimal chaining (see Step 2 below). The optimal solutions to the instances produce candidate alignment sites.

In the third step, each solution to the MEM chaining is processed as follows. The MEMs in the chaining solution overlap distinct segments on the genome (defined later; see Fig. 4 for illustration). Each segment belongs to a set of at least one gene. uLTRA aligns these segments together with all small exons (from the same genes) using edlib (Šošić and Šikić 2017). Each such alignment produces a maximal approximate match (MAMs; defined below), and uLTRA uses all MAMs with alignment accuracy greater than a threshold T as input for the next chaining problem. There can be several MAMs of the same segment or small exon within a read. In the chaining of MAMs, we, roughly, optimize the total weight of MAMs with discrepancy and overlap

penalties. Here, weight is defined by the alignment accuracy and the length of the match (see Step 3 below). The final set of MAMs produced from the optimal solution(s) constitutes a final set of segments on the genome. Finally, we align the final set of segments to the read using parasail (Daily 2016) (semi-global mode), which produces the final alignment(s) and cigar strings to the genome.

Step 1: Indexing and processing the genome annotation

A *part* is defined as the smallest genomic region fully covering a set of overlapping exons (Fig. 4). By construction, parts are disjoint regions of the genome. *Flanks* are constructed by taking regions of size *F* nucleotides downstream and upstream of parts. If two parts are separated with a distance of less than *F* nucleotides, then the non-overlapping region between the two parts is chosen as a flank region (Fig. 4). By construction, flanks are disjoint regions, both to each other and to parts. Finally, segments are constructed from start and end coordinates of exons. Segments are constructed for each part individually as follows. For a sorted array of exon start and stop coordinates within a part, a segment is constructed for each pair (x_i, x_{i+1}) of adjacent coordinates in the array if $x_{i+1} - x_i \geq X$ where X is a parameter to uLTRA (set to 25). If $x_{i+1} - x_i < X$, uLTRA iteratively attempts to add segments in each direction until success. That is, uLTRA attempts to add (x_{i-k}, x_{i+1}) and (x_i, x_{i+1+k}) for $k = 1,\ 2, ...,$ until first success in each direction. Finally, there may be parts where $y - x < X$ (see exon e7 in Fig. 4). Small segments, exons, or parts have a lower probability of containing a MEM (and therefore be omitted from alignment). We address this complication as follows. uLTRA stores all exons and segments smaller than a threshold in a container that links gene ID to the small segments. This data structure will be queried, and all small segments will be included, whenever there are MEMs to segments linked to the same gene ID.

Step 2: Collinear chaining with MEMs

A *Maximal Exact Match (MEM)* ([a..b],[c..d]) means that genome segment [a..b] matches read segment [c..d], and that such match cannot be extended to either direction. We use notation A[i].x to denote the endpoints of MEMs for x ∈ {a,b,c,d}. Let array A[1..n] contain the MEMs. A *chain* S is a collinear subset of A, meaning that S[i].a<S[i+1].a and S[i].c<S[i+1].c for 0<i<n (i.e. satisfying the weak precedence (Mäkinen and Sahlin 2020)). Coverage(S) is defined as the number of identities in an alignment induced by S, i.e., the length of the anchor-restricted LCS (longest common subsequence) of reference and the read, where anchor now means a MEM (Mäkinen and Sahlin 2020): If there are no overlaps between MEMs in chain S, Coverage(S) is the overall length of MEMs in S, but if there are, the score is adjusted by adding only the minimum length of the non-overlapping parts of the consecutive MEM intervals [MS20]. Here we look for chains that have no overlaps in the genome, so for finding S that maximizes Coverage(S), we use Algorithm 1 in (Mäkinen and Sahlin, 2020) that runs in O(n log n) time.

Step 3: Collinear chaining of MAMs

We refer to an *approximate match*, as an alignment of a genome segment [a..b] to a read segment [c..d] with an accuracy higher than a threshold (parameter to uLTRA) . Here, accuracy is defined as the number of matches divided by the length of the alignment. We find approximate matches of the genome segment by aligning it in semi-global mode to the read using edlib. The length of the alignment is defined by the genome segment's first and last nucleotide coordinates. A *Maximal Approximate Match (MAM)* ([a..b],[c..d]) means that genome segment [a..b] matches approximately read segment [c..d] and that no other approximate match has higher accuracy on the read. Furthermore, we let $\lambda \in [0,1]$ be the penalty for each nucleotide that overlap (on the read) between two MAMs and $\delta \in [0,1]$ the penalty for the distance between two MAMs (on the read). Let array $A[1,...,N]$ contain the MAMs where we use the following notation: $A[i].a$, $A[i].b$, $A[i].c$, $A[i].d$ $A[i].acc$ to denote the genome start, genome stop, read start, read stop, and accuracy of MAM i. Let $S[1,...,m]$ be a chain of the MAMs in A under the weak precedence constraint (Mäkinen and Sahlin 2020). For two MAMs x,y in A, we introduce the following functions. Let $v(x) = (x.d - x.c)$, $o(x,y) = max\{0,\ x.d - y.c\}$ (the overlap), and $d(x,y) = \max\{0, y.c - x.d\}$ (the distance between MAMs) on the read, then the $score(S)$ of a MAM-chain is defined as

$$\sum_{i=1}^{m} (v(S[i]) - o(S[i-1],\ S[i]))S[i].acc - \lambda o(S[i-1],\ S[i]) - \delta\, d\,(S[i-1],\ S[i]),$$

where $o(S[0],\ S[1]) = d(S[0],\ S[1]) = 0$

We find the chain $S^{max} = \max_{chains\ S} score(S)$. This formulation intuitively selects the solution with the best coverage and accuracy, while penalizing overlapping MAMs or MAMs that occur far apart. This formulation is solved with a dynamic programming algorithm: Sort array $A[1,...,N]$ by values $A[i].a$. Let $W[0,...,N]$ be the target array, where we wish to store for each $W[i]$ the maximum score over chains ending at MAM $A[i]$. To compute $W[i]$, one can consider adding $A[i]$ to chains ending at $A[i']$, $\forall i' < i$ with $A[i'].c < A[i].c$. This increases the score by $w(i',i) = (v(A[i]) - o(A[i'],\ A[i]))A[i].acc - \lambda\, o(A[i'],\ A[i]) - \delta\, d\,(A[i'],\ A[i])$. After initializing $W[0] = 0$, we can set $W[i]$ to the maximum over $W[i'] + w(i',i)$ for $0 \le i' < i$ with $A[i'].c < A[i].c$ from left to right, and the maximum scoring chain can be traced back starting from the maximum value in $W[1,..,N]$. Although this computation takes quadratic time, in practice the instances of segments are small enough to be solved quickly.

Tiling segment solution for poor fit alignments

The set of segments in the MAM solution is concatenated, and parasail (Daily 2016) in semi-global mode is used to obtain the final alignment. Some solutions in the MAM chaining

may produce a set of segments that do not fully cover the read at junction sites. While smaller offsets are expected due to, e.g., read errors, a larger indel in the final alignment indicates that the read contains a junction that is missing from the annotation or that the read has a larger structural error (e.g., occurring in the experimental protocol). To detect such anomalies, uLTRA looks for indels larger than 15nt in junctions. If there is an indel of at least this size in at least one junction, uLTRA performs chaining of subsegments of size Y (parameter; default 25) of all the segments in the MAM step (denoted a *tiling*). In this case, the weight of each tile i is defined as $w_i = (A[i].d - A[i].c) \cdot A[i].acc$. Then we use an algorithm analogous to Algorithm 1 in (Mäkinen and Sahlin, 2020) to obtain, by construction, the solution with maximal weight (instead of Coverage as for the MEM step). If the alignment score obtained from the tiling solution is better than the original solution, uLTRA reports the tiling solution.

Implementation

Chaining of MEMs

In the implementation, the optimal solution instance is found through backtracking. If several possible tracebacks paths lead to the same optimal value for a given optimal value in the traceback vector, uLTRA will always choose the closest MEM, i.e., the one with the highest index j. This means that the mem with the closest genomic coordinate is chosen if several exist.

If several optimal chaining solutions are found, i.e., several positions in the vector traceback vector have the optimal value, uLTRA will report all of the solutions by backtracking each instance (as described above). This is not a rare case since there can be identical or highly similar gene copies annotated on the genome that give the same optimal value.

Since each read can have several chaining instances to solve, uLTRA pre-calculates the theoretical maximum MEM coverage that an instance can have, which is upper bounded by the sum of all the regions covered by mems in the reads. uLTRA then solves the chaining instances by highest upper bound on coverage. If at any point the upper bound drops below a drop-off threshold (parameter to uLTRA) the current best solution uLTRA skips to calculate the rest of the instances. There is also a parameter to limit the number of reported alignments.

Chaining of MAMs

MAMs are formed by aligning segments and exons with at least an alignment identity of X% (default 60), and in case of exons between 5-8 nucleotides in length, an exact match is required. Exons of 4bp or less are ignored because of the potential blowup in the number of matches across the read. Similarly to the MEM chaining, the traceback will choose the MAM with the highest index j.

Alignment reporting

The exons that are included in an optimal solution of the MAM chaining are concatenated into an augmented reference, and the read is aligned to this reference using parasail in semi-global mode. The alignment score and cigar string are computed from the alignment. Among all MAM instances for a read, the highest scoring one is selected as the primary alignment. If a read has multiple best scoring alignments, the one with the shortest genomic span of the alignment is reported, and if still a tie, an FSM is preferred over the other read labels.

A read is assigned as unaligned if the alignment score is lower than X*m*r, where r is the read length, m is the match score (set to 2 in parasail; see Suppl. Note A for details), and X is a parameter to uLTRA (set to 0.5). The default setting roughly corresponds to classifying a read as unaligned if it has more than 25% errors, or if a larger segment of the read is from, e.g., from a region that is not included in the indexing.

Output

uLTRA outputs alignments in SAM-file format with genomic coordinates as annotated by the transcript database. In addition, uLTRA outputs an annotation of the alignment following the definitions in (Tardaguila et al. 2018) in the SAM-file in the optional field “CN”.

Discussion

We have presented a novel splice alignment algorithm that aligns long transcriptomic reads to a genome using an annotation of coding regions. We evaluated its implementation, uLTRA, on simulated, synthetic, and biological data. In addition to the alignments reported in SAM-format, uLTRA classifies the splice alignments (with the classification given in (Tardaguila et al. 2018)). It outputs the annotation information in the SAM-file under an optional tag. Using simulated data, we demonstrated uLTRA’s increased accuracy over other aligners. Notably, we demonstrated the difficulty for state-of-the-art splice aligners to align reads to small exons. Our algorithm has addressed this issue and significantly increases the alignment accuracy to small exons.

We also observed a drastic decrease in accuracy for other alignment algorithms when aligning to novel isoform structures (Fig. S2) compared to simulated data of already annotated isoforms. The decrease in accuracy occurs across exon sizes (Fig. S2B). However, uLTRA does not see a dropoff in accuracy on this dataset. As the simulated data of novel isoforms does not honor

the distribution of canonical junctions, splice-junction specific alignment penalties tuned for, e.g., the human genomes, may cause a more significant fraction of misalignments.

We used synthetic data to investigate the performance of alignment algorithms when aligning to known splice sites. Our experiments demonstrate that uLTRA aligns a much higher percentage of reads to known isoforms in the data. A large fraction of these FSMs are shared either with deSALT or minimap2 (Fig 2.). Furthermore, uLTRA's increase in the number of aligned FSM reads is distributed across the 59 isoforms with distinct splice sites without any indication of alignment bias towards specific isoforms (Fig. S3).

On biological data, we demonstrated several examples where uLTRA aligns reads to the correct isoform structure while the other aligners do not. We showed several examples where isoforms containing small exons were misaligned (Fig. S9, S11). We also illustrated that employing junction specific alignment penalties may lead to concordant but erroneous alignment around junctions (Fig. S10). Finally, we observed cases where homopolymer differences in reads may lead to subtle alignment differences causing alignment novel junctions (Fig. S12). For this example, we do not know whether the homopolymer length is present in the transcripts or caused by sequencing errors. If this discrepancy is because of homopolymer errors, employing junction-specific alignment penalties is of advantage in such scenarios. Another solution could be to look for junctions already occurring in the annotations and weigh them higher. In summary, the examples we provide on biological data demonstrate that using simple concordance analysis between aligners to measure accuracy can be misleading.

Splice alignment is an algorithmic problem central for the detection and prediction and quantification of isoforms. Our analysis in this study highlights some of the challenges with splice alignment and the current state-of-the-art approaches. To demonstrate the importance, We chose examples in the ALZ dataset from genes that have been studied or linked to Alzheimer's disease (Fig. S9-S13), with many of them highly abundant in the dataset. As these isoforms may not be detected with other alignment software, we demonstrated the utility of uLTRA and highlighted the significance of further development of splice alignment techniques.

uLTRA does not align to unannotated regions, which is a limitation for biological datasets that can contain genomic reads from both artifacts in the experimental protocol and novel and unannotated exons (Zhang et al. 2020). This limitation could be overcome by running, e.g., minimap2 within uLTRA and flag reads that align outside the indexed regions. uLTRA's results will also further improve with new exons added to the gene annotation.

As alignment to pan-genome graphs has demonstrated its advantage over linear genomes (Garrison et al. 2018), it is of interest to explore such approaches in a transcriptomic setting. Our alignment strategy facilitates the addition of variant sequences in a relatively straightforward manner by adding alternative segments to uLTRA's index (containing variations obtained from variant annotation file). We, therefore, aim to continue working on uLTRA in this direction.

Adding segments to encompass variation will increase the memory of the data structures. However, in uLTRA the majority of memory consumption comes from keeping pre-computed tiling data structures in memory. These data structures could be removed and, instead, be computed for each instance and region that requires the tiling structure.

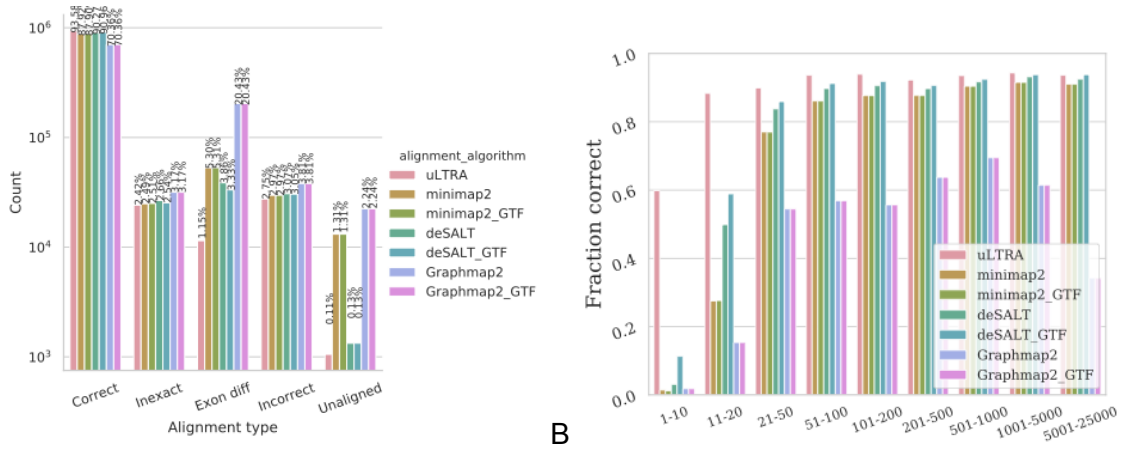
Data availability

The biological and synthetic datasets are publicly available datasets. The pacbio Alzheimer dataset can be downloaded at https://downloads.pacbcloud.com/public/dataset/Alzheimer2019_IsoSeq/. The drosophila ONT and SIRV datasets can be downloaded from ENA under project accession number PRJEB34849. The Ensembl cDNA was downloaded from https://www.ensembl.org/biomart/martview/. All scripts used for simulating datasets and to run the evaluation are found at https://github.com/ksahlin/ultra/tree/master/evaluation.

Acknowledgements

The computations were performed on resources provided by the Swedish National Infrastructure for Computing (SNIC) at Uppsala Multidisciplinary Center for Advanced Computational Science (UPPMAX). The work was partially supported by the Academy of Finland (grant 309048).

Figures and Tables



A B

Figure 1. Alignment results on simulated data for the SIM_ANN dataset. (**A**) Percentage of reads in each respective category. (**B**) The fraction of correctly aligned exons (y-axis) as a function of exon size (x-axis). GraphMap2 encounters an error on the ENS data and could therefore not be evaluated.

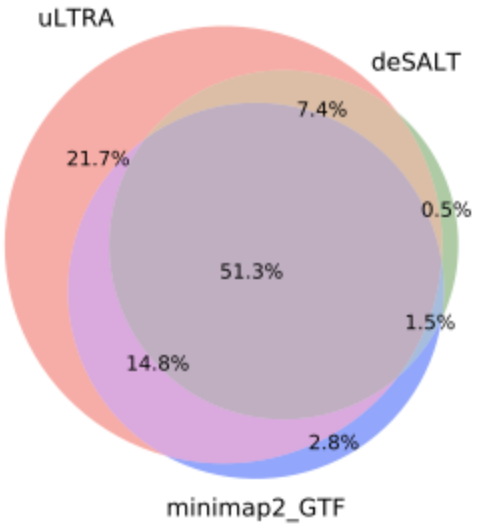


Figure 2. Venn diagram of the set of annotated FSM reads to the 59 SIRV isoforms that had at least one splice site. The intersections of the methods display the percentage of reads that were FSM reads to the same SIRV isoform.

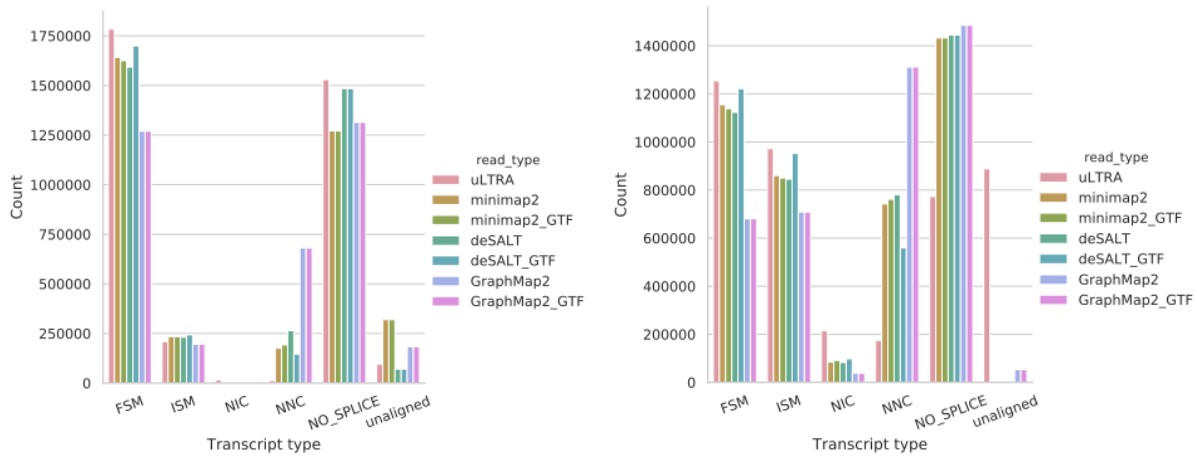


Figure 3. Number of reads annotated in different splicing categories for DROS (A) and ALZ (B).

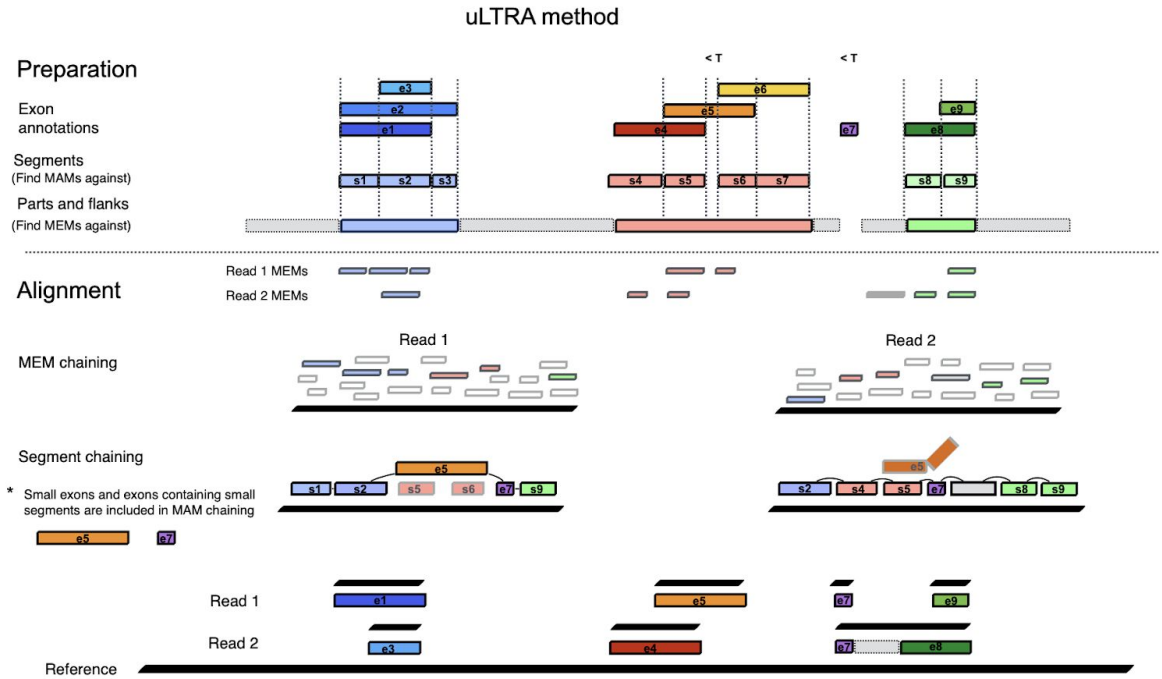


Figure 4. Overview of uLTRA alignment algorithm. Segments, Parts and Flanks are stored and indexed for alignment. In the alignment step, MEMs in the reads to the parts and flanks are computed. Collinear chain(s) of MEMs covering as much of the read as possible are found for each read. The MEMs in the solution overlap segments and/or flanks that are linked to gene IDs. The overlapping segments, flanks and all small exons to the same gene IDs are aligned to the read to form a set of MAMs. Collinear chains of MAMs are found by optimizing for coverage and alignment identity. The collinear chaining solution of MAMs is used to produce the final alignment of the read to the genome.

	Dataset	Nr reads	Median read length	Median error rate	Genome	Annotation
Simulated	ENS	234,207	890	0.0%	GRCh38.p12	Gencode v34 (hapl scaff)
	SIM_ANN	1,000,000	864	8.6%	GRCh38.p12	Gencode v34 (hapl scaff)
	SIM_NIC	1,000,000	1,272	8.6%	GRCh38.p12	Gencode v34 (hapl scaff)
Synthetic	SIRV	1,514,274	538	6.9%*	SIRV genome	SIRV annotation C_170612a
Biological	ALZ	4,277,293	2,699	1.2%*	GRCh38.p12	Gencode v34 (hapl scaff)
	DROS	3,646,342	559	7.0%*	BDGP6.28	Ensembl v100

Table 1 Datasets included in evaluation. *Measured from minimap2's alignments. Due to biological sequence variations, the error rate may be lower than the number presented here.

Dataset	uLTRA	minimap 2	minimap2_ GTF	deSALT	deSALT_GTF	GraphMap 2	GraphMap2_G TF
ENS	1h 05m	12m	1h 2m	14m	14m	-	-
SIM_AN N	2h 01m	30m	3h 37m	47m	48m	4h 02m	4h 03m
SIM_NIC	2h 13m	46m	6h 02m	1h 32m	1h 35m	4h 03m	4h 01m
SIRV	23m	3m	18m	3m	-	-	-
ALZ	5h 35m	1h 25m	12h 9m	9h 03m	9h 30m	22h 33m	22h 21m
DROS	47m	4m	36m	8m	9m	1h 08m	1h 11m

Table 2. Runtime of alignment.

Dataset	uLTRA	minimap2	minimap2 _GTF	deSAL T	deSAL T_GTF	GraphMap2	GraphMap2_GT F
ENS	375Gb	52Gb	52Gb	40Gb	40Gb	-	-
SIM_ANN	378Gb	57Gb	58Gb	43Gb	42Gb	249Gb	256Gb
SIM_NIC	378Gb	64Gb	65Gb	45Gb	50Gb	248Gb	248Gb
SIRV	9Gb	10Gb	10Gb		-	-	-
ALZ	405Gb	53Gb	57Gb	166Gb	167Gb	472Gb	472Gb
DROS	72Gb	25Gb	23Gb	49Gb	49Gb	200Gb	200Gb

Table 3. Peak memory usage of alignment.

Dataset	uLTRA	minimap2	deSALT	GraphMap2
HG38	27m	5m	2h 49m	14m
SIRV	0m	0m	1m	0m
DROS	3m	0m	9m	1m

Table 4. Runtime of indexing.

Dataset	uLTRA	minimap2	deSALT	GraphMap2
HG38	12Gb	19Gb	74Gb	57Gb
SIRV	0Gb	0Gb	2Gb	0Gb
Drosophila	4Gb	2Gb	6Gb	3Gb

Table 5. Peak memory usage of indexing.

References

Bagyinszky, Eva, Min Ju Kang, Jungmin Pyun, Vo Van Giau, Seong Soo A. An, and Sangyun Kim. 2019. “Early-Onset Alzheimer’s Disease Patient with Prion (PRNP) p.Val180Ile Mutation.” *Neuropsychiatric Disease and Treatment*. https://doi.org/10.2147/ndt.s215277.

Boland, Barry, Wai Haung Yu, Olga Corti, Bertrand Mollereau, Alexandre Henriques, Erwan Bezard, Greg M. Pastores, et al. 2018. “Promoting the Clearance of Neurotoxic Proteins in Neurodegenerative Disorders of Ageing.” *Nature Reviews. Drug Discovery* 17 (9): 660–88.

Calvo-Rodriguez, Maria, Steven S. Hou, Austin C. Snyder, Elizabeth K. Kharitonova, Alyssa N. Russ, Sudeshna Das, Zhanyun Fan, et al. 2020. “Increased Mitochondrial Calcium Levels Associated with Neuronal Death in a Mouse Model of Alzheimer’s Disease.” *Nature Communications*. https://doi.org/10.1038/s41467-020-16074-2.

Charbonnier, F., J. P. Périn, G. Roussel, J. L. Nussbaum, and P. M. Alliel. 1997. “[Cloning of testican/SPOCK in man and mouse. Neuromuscular expression perspectives in pathology].” *Comptes rendus des seances de la Societe de biologie et de ses filiales* 191 (1): 127–33.

Daily, Jeff. 2016. “Parasail: SIMD C Library for Global, Semi-Global, and Local Pairwise Sequence Alignments.” *BMC Bioinformatics* 17 (February): 81.

Dobin, Alexander, Carrie A. Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R. Gingeras. 2013. “STAR: Ultrafast Universal RNA-Seq Aligner.” *Bioinformatics* 29 (1): 15–21.

Fernandes, Francisco, and Ana T. Freitas. 2014. “slaMEM: Efficient Retrieval of Maximal Exact Matches Using a Sampled LCP Array.” *Bioinformatics* 30 (4): 464–71.

Garrison, Erik, Jouni Sirén, Adam M. Novak, Glenn Hickey, Jordan M. Eizenga, Eric T. Dawson, William Jones, et al. 2018. “Variation Graph Toolkit Improves Read Mapping by Representing Genetic Variation in the Reference.” *Nature Biotechnology* 36 (9): 875–79.

Geuens, Thomas, Delphine Bouhy, and Vincent Timmerman. 2016. “The hnRNP Family: Insights into Their Role in Health and Disease.” *Human Genetics* 135 (8): 851–67.

Kim, Daehwan, Ben Langmead, and Steven L. Salzberg. 2015. “HISAT: A Fast Spliced Aligner with Low Memory Requirements.” *Nature Methods* 12 (4): 357–60.

Kim, Daehwan, Joseph M. Paggi, Chanhee Park, Christopher Bennett, and Steven L. Salzberg.

2019. “Graph-Based Genome Alignment and Genotyping with HISAT2 and HISAT-Genotype.” *Nature Biotechnology* 37 (8): 907–15.

Kuo, Richard I., Yuanyuan Cheng, Jacqueline Smith, Alan L. Archibald, and David W. Burt. 2019. “Illuminating the Dark Side of the Human Transcriptome with TAMA Iso-Seq Analysis.” https://doi.org/10.1101/780015.

Mäkinen, Veli, and Kristoffer Sahlin. 2020. “Chaining with Overlaps Revisited.” In Proc. CPM 2020, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, LIPIcs 161:25:1-25:12. doi: 10.4230/LIPIcs.CPM.2020.25.

Robinson, James T., Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S. Lander, Gad Getz, and Jill P. Mesirov. 2011. “Integrative Genomics Viewer.” *Nature Biotechnology* 29 (1): 24–26.

Sahlin, Kristoffer, Botond Sipos, Phillip L. James, Daniel J. Turner, and Paul Medvedev. 2020. “Error Correction Enables Use of Oxford Nanopore Technology for Reference-Free Transcriptome Analysis.” https://doi.org/10.1101/2020.01.07.897512.

Šošić, Martin, and Mile Šikić. 2017. “Edlib: A C/C Library for Fast, Exact Sequence Alignment Using Edit Distance.” *Bioinformatics*. https://doi.org/10.1093/bioinformatics/btw753.

Tanahashi, Hiroshi, and Takeshi Tabira. 1999. “Molecular Cloning of Human Fe65L2 and Its Interaction with the Alzheimer’s β-Amyloid Precursor Protein.” *Neuroscience Letters*. https://doi.org/10.1016/s0304-3940(98)00995-1.

Tardaguila, Manuel, Lorena de la Fuente, Cristina Marti, Cécile Pereira, Francisco Jose Pardo-Palacios, Hector Del Risco, Marc Ferrell, et al. 2018. “SQANTI: Extensive Characterization of Long-Read Transcript Sequences for Quality Control in Full-Length Transcriptome Identification and Quantification.” *Genome Research*, February. https://doi.org/10.1101/gr.222976.117.

Tian, Yuan, Jerry C. Chang, Emily Y. Fan, Marc Flajolet, and Paul Greengard. 2013. “Adaptor Complex AP2/PICALM, through Interaction with LC3, Targets Alzheimer’s APP-CTF for Terminal Degradation via Autophagy.” *Proceedings of the National Academy of Sciences of the United States of America* 110 (42): 17071–76.

Trapnell, Cole, Lior Pachter, and Steven L. Salzberg. 2009. “TopHat: Discovering Splice Junctions with RNA-Seq.” *Bioinformatics* 25 (9): 1105–11.

Wang, Xinlu, Fei Fei, Jie Qu, Chunyuan Li, Yuwei Li, and Shiwu Zhang. 2018. “The Role of Septin 7 in Physiology and Pathological Disease: A Systematic Review of Current Status.” *Journal of Cellular and Molecular Medicine* 22 (7): 3298–3307.

Wu, Thomas D., Jens Reeder, Michael Lawrence, Gabe Becker, and Matthew J. Brauer. 2016. “GMAP and GSNAP for Genomic Sequence Alignment: Enhancements to Speed, Accuracy, and Functionality.” *Methods in Molecular Biology* 1418: 283–334.

Wyman, Dana, Gabriela Balderrama-Gutierrez, Fairlie Reese, Shan Jiang, Sorena Rahmanian, Stefania Forner, Dina Matheos, et al. 2020. “A Technology-Agnostic Long-Read Analysis Pipeline for Transcriptome Discovery and Quantification.” https://doi.org/10.1101/672931.

Zhang, David, Sebastian Guelfi, Sonia Garcia-Ruiz, Beatrice Costa, Regina H. Reynolds, Karishma D’Sa, Wenfei Liu, et al. 2020. “Incomplete Annotation Has a Disproportionate Impact on Our Understanding of Mendelian and Complex Neurogenetic Disorders.” *Science Advances* 6 (24): eaay8299.