





# Jumper enables discontinuous transcript assembly in coronaviruses

Palash Sashittal <sup>1</sup>, Chuanyi Zhang <sup>2</sup>, Jian Peng <sup>1,3</sup> & Mohammed El-Kebir <sup>1</sup>✉

Genes in SARS-CoV-2 and other viruses in the order of *Nidovirales* are expressed by a process of discontinuous transcription which is distinct from alternative splicing in eukaryotes and is mediated by the viral RNA-dependent RNA polymerase. Here, we introduce the DISCONTINUOUS TRANSCRIPT ASSEMBLY problem of finding transcripts and their abundances given an alignment of paired-end short reads under a maximum likelihood model that accounts for varying transcript lengths. We show, using simulations, that our method, JUMPER, outperforms existing methods for classical transcript assembly. On short-read data of SARS-CoV-1, SARS-CoV-2 and MERS-CoV samples, we find that JUMPER not only identifies canonical transcripts that are part of the reference transcriptome, but also predicts expression of non-canonical transcripts that are supported by subsequent orthogonal analyses. Moreover, application of JUMPER on samples with and without treatment reveals viral drug response at the transcript level. As such, JUMPER enables detailed analyses of *Nidovirales* transcriptomes under varying conditions.

<sup>1</sup> Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. <sup>2</sup> Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. <sup>3</sup> College of Medicine, University of Illinois at Urbana-Champaign, Urbana, IL 61801, USA. ✉email: [melkebir@illinois.edu](mailto:melkebir@illinois.edu)

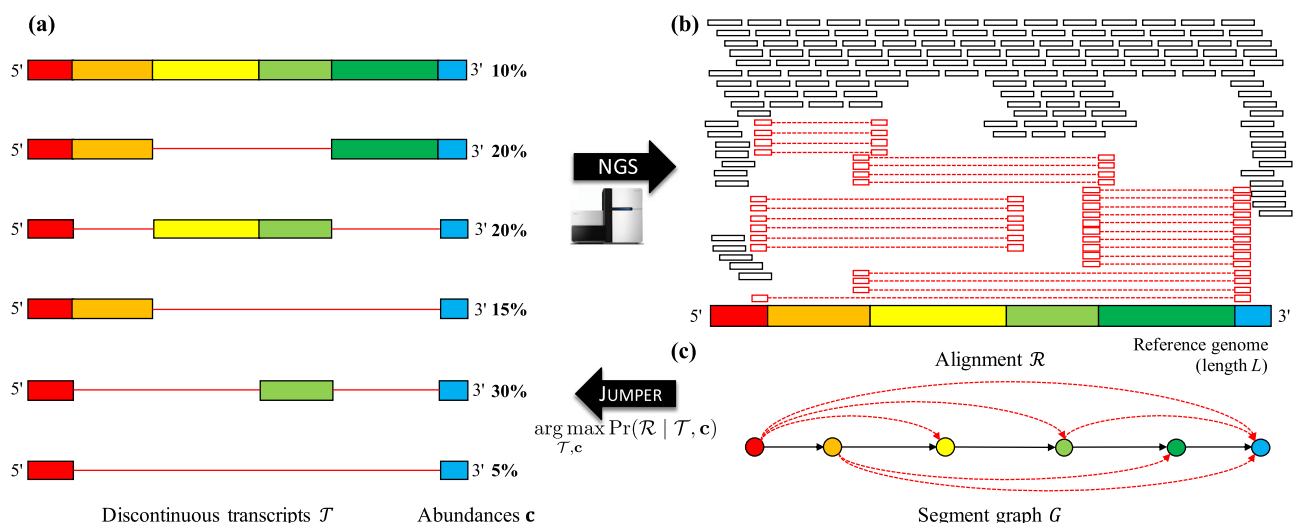
Coronaviruses, and more generally viruses in the taxonomic order of *Nidovirales*, are enveloped viruses containing a positive-sense, single-stranded RNA genome that encodes for non-structural proteins near the 5' end as well as structural and accessory proteins near the 3' end<sup>1</sup>. Since the host ribosome processes mRNA starting at the 5' end, translation of the viral genome only generates the non-structural proteins. Expression of the remaining genes is achieved by discontinuous transcription performed by the viral RNA-dependent RNA polymerase (RdRp)<sup>2</sup>, a protein that is encoded in the non-structural part of the viral genome. Specifically, RdRp can skip over contiguous genomic regions, or segments, in the viral RNA template, resulting in a repertoire of discontinuous transcripts that correspond to distinct subsequences of segments ordered as in the reference genome (Fig. 1a). Several recent studies have analyzed SARS-CoV-2 sequencing samples, identifying split reads—i.e. reads that span non-contiguous parts of the viral genome—that provide evidence for canonical discontinuous transcription events that produce an intact 3' open reading frame as well as non-canonical discontinuous transcription events whose role is unclear<sup>3–5</sup>. However, to the best of our knowledge, no study has attempted to assemble coronavirus transcriptomes, which could provide important clues about the viral life cycle under various conditions such as drug treatment.

Current methods for transcript assembly are mainly designed for eukaryotes and fall under two broad categories: (i) reference-based methods and (ii) de novo assembly methods. The main distinction is that the former require the reference genome as input while the latter have no such requirement. As such, de novo assembly methods<sup>6–10</sup> are useful when the reference genome is unavailable or when the diversity of different species in the sample is too large. On the other hand, reference-based methods<sup>11–21</sup> generally achieve higher accuracy as they use the reference genome as a scaffold on which to align sequencing reads. Typically, such methods construct a splice graph  $G$ —i.e. a directed graph whose nodes correspond to contiguous genomic regions and edges indicate splice junctions—and subsequently aim to decompose the graph into paths that correspond to transcripts. In addition to inferring transcripts  $\mathcal{T}$  given an alignment  $\mathcal{R}$ , a subset of reference-based methods simultaneously

estimates their abundances  $\mathbf{c}$ <sup>17,19</sup>. Alternatively, transcripts abundances  $\mathbf{c}$  may also be quantified using separate tools<sup>22,23</sup>. We refer to Supplementary Note A.1 for a more detailed overview of previous work.

There are important differences between transcription in eukaryotes and coronaviruses. In eukaryotes, a gene may express multiple transcripts that differ in their composition due to alternative splicing, which is predominantly mediated by the spliceosome and results in the generation of multiple mRNAs with differentially joined or skipped exons (segments) from the same gene. By contrast, transcripts in coronaviruses result from discontinuous transcription, which is mediated by viral RdRp and results in the removal of contiguous segments due to jumps of the RdRp. There are two key differences between these two biological processes. First, in discontinuous transcription, there is no shuffling of segments and the ordering of the segments is maintained in each transcript. Since exon shuffling in eukaryotes is rare during alternative splicing, this constraint is commonly used in existing transcript assembly methods as well<sup>11,12,19–21</sup>. Second, the complete viral genome, without any jumps, is always part of the transcriptome. Consequently, due to these two biological constraints, an alignment  $\mathcal{R}$  of coronavirus samples will yield a splice graph  $G$  with additional constraints that current methods do not leverage.

In this study, we introduce the DISCONTINUOUS TRANSCRIPT ASSEMBLY (DTA) problem of finding discontinuous transcripts  $\mathcal{T}$  and their abundances  $\mathbf{c}$  (Fig. 1a) given an alignment  $\mathcal{R}$  of paired-end reads (Fig. 1b). Underpinning our approach is the concept of a segment graph (Fig. 1c), which is an acyclic splice graph with a Hamiltonian path due to the aforementioned constraints. This enables us to characterize discontinuous transcripts  $\mathcal{T}$  as small subsets of non-overlapping edges in this graph. Our method, JUMPER, uses this compact representation to solve the DTA at scale via a progressive heuristic that incorporates a mixed integer linear program. Using simulations, we show that JUMPER outperforms SCALLOP<sup>11</sup> and STRINGTIE<sup>12</sup>, existing methods for reference-based transcript assembly in eukaryotes. In real data<sup>3</sup>, we run JUMPER on paired-end short-read data of virus-infected Vero cells and use long-read data of the same sample for validation. We find that JUMPER not only identifies canonical



**Fig. 1 Overview of Jumper.** **a** Coronaviruses generate a set  $\mathcal{T}$  of discontinuous transcripts with varying abundances ( $\mathbf{c}$ ) during infection. **b** Next-generation sequencing will produce an alignment  $\mathcal{R}$  with two types of aligned reads: reads that map to a contiguous genomic region (black) and split reads that map to distinct genomic regions (red). **c** From  $\mathcal{R}$  we obtain the segment graph  $G$ , a directed acyclic graph with a unique Hamiltonian path. JUMPER solves the DISCONTINUOUS TRANSCRIPT ASSEMBLY to infer  $\mathcal{T}$  and  $\mathbf{c}$  with maximum likelihood. While this figure shows single-end reads, our problem statement and method make use of the additional information provided by paired-end reads.

transcripts that are part of the reference transcriptome, but also predicts expression of non-canonical transcripts that are well supported by long-read data. Similarly, JUMPER identifies canonical and non-canonical transcripts in SARS-CoV-1 and MERS-CoV samples<sup>24</sup>. Finally, we demonstrate the use of JUMPER to study viral drug response at the transcript level by analyzing samples with and without treatment prior to infection<sup>25</sup>. In summary, JUMPER enables detailed analyses of coronavirus transcriptomes under varying conditions.

**Results**

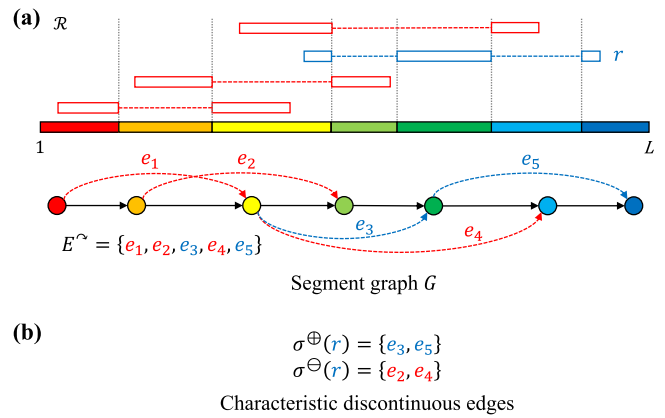
**Discontinuous Transcript Assembly problem.** To formulate the DISCONTINUOUS TRANSCRIPT ASSEMBLY problem, we define discontinuous transcripts as sequences of segments whose order matches the reference genome. More formally, we have the following definition.

**Definition 1.** Given a reference genome, a *discontinuous transcript*  $T$  is a sequence  $v_1, \dots, v_{|T|}$  of segments where (i) each segment corresponds to a contiguous region in the reference genome, (ii) segment  $v_i$  precedes segment  $v_{i+1}$  in the reference genome for all  $i \in \{1, \dots, |T| - 1\}$ , (iii) segment  $v_1$  contains the 5' end of the reference genome and (iv) segment  $v_{|T|}$  contains the 3' end of the reference genome.

While the genomic transcript  $T_0$  matches the reference genome<sup>2</sup>, subgenomic transcripts contain jumps and correspond to subgenomic RNAs (sgRNAs)<sup>3</sup>. Discontinuous transcripts  $\mathcal{T} = \{T_i\}$  occur in abundances  $\mathbf{c} = [c_i]$  where  $c_i \geq 0$  is the relative abundance of transcript  $T_i$  such that  $\sum_{i=1}^{|\mathcal{T}|} c_i = 1$ . In this work, we focus on coronavirus sequencing samples obtained using Illumina sequencing, where reads originate from the reference genome of length  $L$  of about 10–30 Kbp and have a fixed length  $\ell$  ranging from 100 to 400 bp. We refer to Supplementary Note A.3 for a discussion on why transcript assembly remains relevant for such samples in light of the availability long-read sequencing samples. For ease of exposition, we describe the formulation in the context of single-end reads, but in practice we use the paired-end information if it is available. We refer to Supplementary Note B.5 for details on the paired-end formulation.

As  $\ell \ll L$ , the identity of the transcript of origin for a given read is ambiguous. Therefore, we need to use computational methods to reconstruct the transcripts and their abundances from the sequencing reads. Specifically, given a coronavirus reference genome of length  $L$  and reads of a fixed length  $\ell$ , we use a splice-aware aligner such as STAR<sup>26</sup> to obtain an alignment  $\mathcal{R}$ . This alignment provides information about the abundance  $\mathbf{c}$  and composition of the underlying transcripts  $\mathcal{T}$  in the following two ways. First, the depth, or the number of reads along the genome is informative for quantifying the abundance  $\mathbf{c}$  of the transcripts. Second, the composition of the transcripts  $\mathcal{T}$  is embedded in split reads, which are reads that align to multiple distinct regions in the reference genome (Fig. 1b). Since the alignment  $\mathcal{R}$  is composed of reads from discontinuous transcripts, the alignment satisfies the following two properties. First, the genomic regions induced by any read in  $\mathcal{R}$ , including split reads, are ordered from the 5' to 3' direction of the reference genome. Second, due to the presence of the genomic transcript  $T_0$ , every position in the reference genome can be expected to be covered by a read.

To infer  $\mathcal{T}$  and  $\mathbf{c}$  from  $\mathcal{R}$ , most reference-based transcript assembly methods employ a splice graph<sup>11,12,18</sup>. Informally, the nodes of this graph correspond to contiguous segments of the genome (i.e. are not separated by any split read) and directed edges correspond to pairs of segments that are spanned by the same read. Due to the aforementioned properties of an alignment  $\mathcal{R}$  of reads from discontinuous transcripts, the edges of the corresponding splice graph can be partitioned into two sets. First,



**Fig. 2 Schematic describing split reads and characteristic discontinuous edges.** **a** Split reads in an alignment  $\mathcal{R}$  define a set of junctions, which in turn define the segment graph  $G$ . **b** Each split read has characteristic discontinuous edges indicating the set  $\sigma^{\oplus}$  of discontinuous edges present in the read as well as conflicting/overlapping discontinuous edges  $\sigma^{\ominus}$ . Here, split read  $r$  (blue), has  $\sigma^{\oplus}(r) = \{e_3, e_5\}$  and  $\sigma^{\ominus}(r) = \{e_2, e_4\}$ . Note that  $e_1$  is not included in  $\sigma^{\ominus}(r)$  as it does not overlap with  $\pi(r) = \{e_3, e_5\}$ .

continuous edges correspond to edges between segments that are adjacent in the reference genome. Conversely, due to the presence of reads from the genomic transcript  $T_0$ , every pair of adjacent segments in the reference genome will be connected by a continuous edge. Second, discontinuous edges connect non-adjacent segments, which indicate the jumps made by the viral RdRp during discontinuous transcription. Both types of directed edges connect segments in the same as the reference genome. Thus, the splice graph obtained from  $\mathcal{R}$  is a directed acyclic graph (DAG) with a Hamiltonian path composed of the continuous edges.

We now show an alternative, more efficient construction of the same graph using only the split reads in an alignment  $\mathcal{R}$  of reads from discontinuous transcripts. Each split read  $r \in \mathcal{R}$  maps to  $q \geq 2$  distinct regions in the reference genome. Each pair of regions that are adjacent in the split read are separated by two positions  $v, w$  (where  $w - v \geq 2$ ) in the reference genome called junctions. Thus, each split read contributes  $2q - 2$  junctions. The collective set of junctions contributed by all split reads in  $\mathcal{R}$  in combination with positions  $\{1, L\}$  induces a partition of the reference genome into closed intervals  $[v^-, v^+]$  of junctions that are consecutive in the reference genome (i.e. there exists no other junction that occurs in between  $v^-$  and  $v^+$ ). The resulting set of segments equals the node set  $V$  of segment graph  $G$  (Fig. 2a). The edge set  $E$  of segment graph  $G$  is composed of continuous edges  $E^{\rightarrow}$  and discontinuous edges  $E^{\curvearrowright}$ . Continuous edges  $E^{\rightarrow}$  are composed of ordered pairs  $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$  of nodes that correspond to segments that are adjacent in the reference genome, i.e. where  $v^+ = w^-$ . On the other hand, discontinuous edges  $E^{\curvearrowright}$  are composed of ordered pairs  $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$  of nodes that corresponds to segments that are adjacent in at least one split read in  $\mathcal{R}$  but not adjacent in the reference genome (i.e.  $w^- - v^+ \geq 2$ ). Fig. 1c shows an example of a segment graph.

**Definition 2.** Given an alignment  $\mathcal{R}$  of reads from discontinuous transcripts, the corresponding *segment graph*  $G = (V, E^{\rightarrow} \cup E^{\curvearrowright})$  is a directed graph whose node set  $V$  equals the set of segments induced by the junctions of split reads in  $\mathcal{R}$  and whose edge set  $E = E^{\rightarrow} \cup E^{\curvearrowright}$  is composed of edges  $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$  that are either continuous, i.e.  $v^+ = w^-$ , or discontinuous, i.e.  $w^- - v^+ \geq 2$ .

–  $v^+ \geq 2$  and there exists a split read where junctions  $v^+$  and  $w^-$  are adjacent.

Thus, by definition any segment graph will have a Hamiltonian path induced by the continuous edges  $E^\rightarrow$ . Moreover, the segment graph obtained from an alignment of reads from discontinuous transcripts will be a DAG.

**Observation 1.** The segment graph  $G$  obtained from an alignment of reads from discontinuous transcripts is a directed acyclic graph with a (unique) Hamiltonian path.

By the above observation,  $G$  has a unique source node  $\mathbf{s}$  and sink node  $\mathbf{t}$ . Importantly, each transcript  $T \in \mathcal{T}$  that is compatible with an alignment  $\mathcal{R}$  corresponds to an  $\mathbf{s} - \mathbf{t}$  path  $\pi(T)$  in  $G$ . Here, a path  $\pi$  is a subset of edges  $E$  that can be ordered  $(\mathbf{v}_1, \mathbf{w}_1), \dots, (\mathbf{v}_{|\pi|}, \mathbf{w}_{|\pi|})$  such that  $\mathbf{w}_i = \mathbf{v}_{i+1}$  for all  $i \in [|\pi| - 1] = \{1, \dots, |\pi| - 1\}$ . While splice graphs of general alignments are DAGs and typically have a unique source and sink node as well, they do not necessarily contain a Hamiltonian path<sup>11,19,27,28</sup>.

Our goal is to find a set  $\mathcal{T}$  of transcripts and their abundances  $\mathbf{c}$  that maximize the posterior probability

$$\Pr(\mathcal{T}, \mathbf{c} | \mathcal{R}) \propto \Pr(\mathcal{R} | \mathcal{T}, \mathbf{c}) \Pr(\mathcal{T}, \mathbf{c}).$$

Under an uninformative, flat prior  $\Pr(\mathcal{T}, \mathbf{c})$ , this is equivalent to maximizing the probability  $\Pr(\mathcal{R} | \mathcal{T}, \mathbf{c})$ . We use the segment graph  $G$  to compute the probability  $\Pr(\mathcal{R} | \mathcal{T}, \mathbf{c})$  of observing an alignment  $\mathcal{R}$  given transcripts  $\mathcal{T}$  and abundances  $\mathbf{c}$ . We follow the same generative model which has been extensively used for transcription quantification<sup>22,23,29</sup>. The notations used in this paper best resemble the formulation described in ref.<sup>28</sup>. Let  $\mathcal{R}$  be composed of reads  $\{r_1, \dots, r_n\}$  and the set  $\mathcal{T}$  of transcripts be  $\{T_1, \dots, T_k\}$  with lengths  $L_1, \dots, L_k$  and abundances  $\mathbf{c} = [c_1, \dots, c_k]$ . In line with current literature, reads  $\mathcal{R}$  are generated independently from transcripts  $\mathcal{T}$  with abundances  $\mathbf{c}$ . Further, we must marginalize over the set of transcripts  $\mathcal{T}$  as the transcript of origin of any given read is typically unknown, since  $\ell \ll L$ . Moreover, we assume that the fixed read length  $\ell$  is much smaller than the length  $L_i$  of any transcript  $T_i$ . As such, we have that  $\Pr(\mathcal{R} | \mathcal{T}, \mathbf{c})$  equals

$$\begin{aligned} \Pr(\mathcal{R} | \mathcal{T}, \mathbf{c}) &= \prod_{j=1}^n \Pr(r_j | \mathcal{T}, \mathbf{c}) \\ &= \prod_{j=1}^n \frac{1}{\sum_{b=1}^k c_b L_b} \sum_{i: \pi(T_i) \supseteq \pi(r_j)} c_i, \end{aligned} \quad (1)$$

where  $\pi(T) \subseteq E$  is the  $\mathbf{s} - \mathbf{t}$  path corresponding to transcript  $T$  and  $\pi(r) \subseteq E$  is the path induced by the ordered sequence of segments (or nodes of  $G$ ) spanned by read  $r$ . By construction,  $\pi(T) \supseteq \pi(r)$  is a necessary condition for transcript  $T$  to be a candidate transcript of origin of read  $r$ . Supplementary Note A.2 gives the derivation of the above equation (Eq. (1)). Our goal is to find  $\operatorname{argmax}_{\mathcal{T}, \mathbf{c}} \Pr(\mathcal{R} | \mathcal{T}, \mathbf{c})$ , leading to the following problem.

**Problem 1.** (DISCONTINUOUS TRANSCRIPT ASSEMBLY(DTA)). Given alignment  $\mathcal{R}$  and integer  $k$ , find discontinuous transcripts  $\mathcal{T} = \{T_1, \dots, T_k\}$  and abundances  $\mathbf{c} = [c_1, \dots, c_k]$  such that (i) each transcript  $T_i \in \mathcal{T}$  is an  $\mathbf{s} - \mathbf{t}$  path in segment graph  $G$ , and (ii)  $\Pr(\mathcal{R} | \mathcal{T}, \mathbf{c})$  is maximum.

In practice, we set the value of  $k$  to a large number (e.g.  $k = 50$ ) and restrict the subsequent analyses to the set of transcripts whose abundance exceeds a threshold value (e.g.  $\geq 0.001$ ). The probability  $P(\mathcal{R} | \mathcal{T}, \mathbf{c})$ , in Eq. (1), is expressed in terms of the observed reads and their induced paths  $\pi(r) \subseteq E(G)$  in the segment graph  $G$ . In the ‘Methods’ section, we describe a more concise way of expressing the probability  $P(\mathcal{R} | \mathcal{T}, \mathbf{c})$  using the fact that the segment graph  $G$  is a DAG with a unique Hamiltonian path. This concise characterization enables us to design a

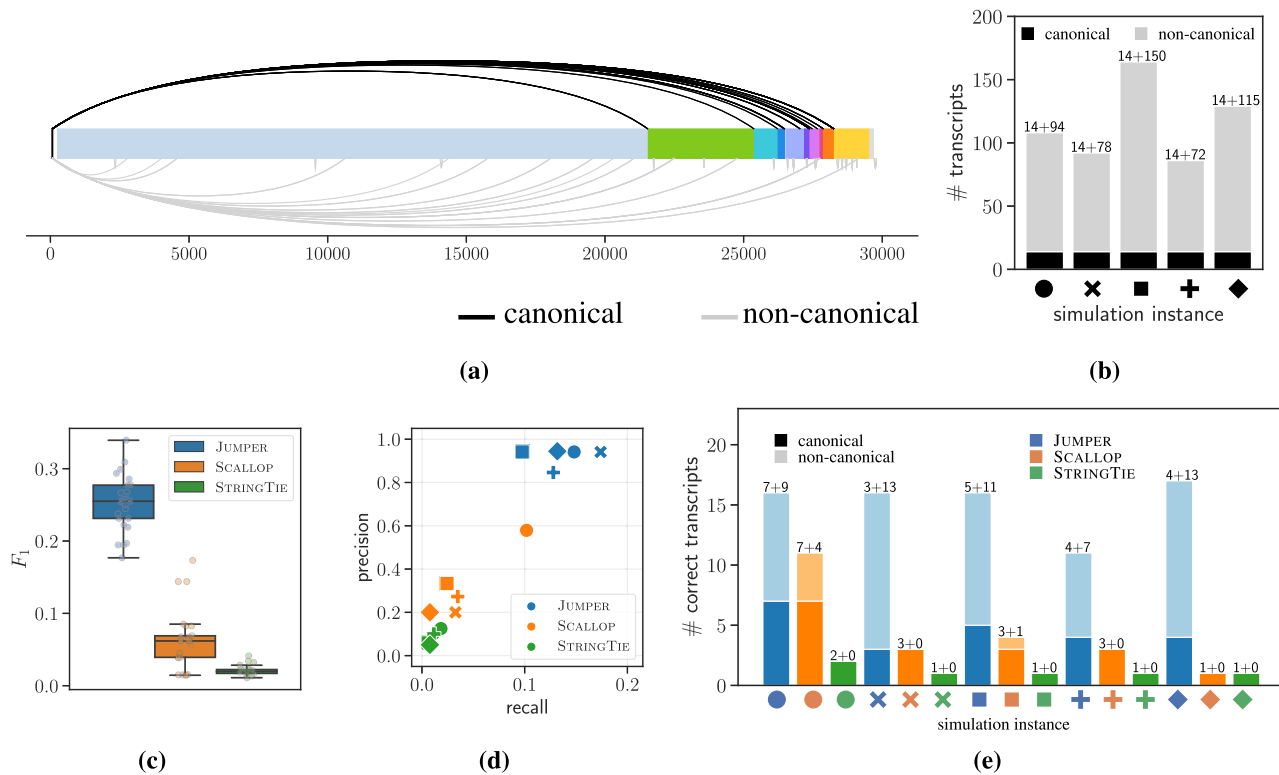
progressive heuristic that incorporates an efficient mixed linear integer program (MILP) to solve the DTA problem (details are in the ‘Methods’ section). Our resulting method, JUMPER, is implemented in Python 3 using Gurobi<sup>30</sup> (version 9.0.3) to solve the MILP and pysam<sup>31</sup> for reading and processing the input BAM file. JUMPER is available at <https://github.com/elkebir-group/Jumper>.

**Experimental evaluation.** We begin by establishing terminology that will be used in the rest of the section. A discontinuous edge  $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$  is canonical provided its 5' junction  $v^+$  occurs in the transcription regulating leader sequence (TRS-L), i.e. between positions 50 and 85, and the first occurrence of ‘AUG’ downstream of the 3' junction  $w^-$  position coincides with the start codon of a known open reading frame (ORF), otherwise the discontinuous edge is called non-canonical. Note that the range 50–85 is chosen since it contains the TRS-L regions of the SARS-CoV-1<sup>32</sup>, SARS-CoV-2<sup>3</sup> and MERS-CoV<sup>32</sup> genomes analyzed in this paper. In a similar vein, a transcript is canonical if it contains at most one canonical and no non-canonical discontinuous edges, otherwise the transcript is non-canonical. We ran all experiments on a server with two 2.6 GHz CPUs and 512 GB of RAM.

**Simulations.** We generated our simulation instances using a segment graph  $G$  obtained from a short-read sample (SRR11409417). Following Kim et al.<sup>3</sup>, we used *fastp* to trim short reads (trimming parameter set to 10 nucleotides), which were input to STAR run in two-pass mode yielding an alignment  $\mathcal{R}$ . Figure 3a shows the sashimi plot of the canonical and the non-canonical discontinuous edges (mappings) supported by the reads in the sample. From  $\mathcal{R}$ , we obtained  $G$  by only including discontinuous edges supported by at least 20 reads. The segment graph  $G$  has  $|V| = 39$  nodes and  $|E| = 67$  edges, which include  $|E^\curvearrowright| = 29$  discontinuous edges and  $|E^\rightarrow| = 38$  continuous edges. The discontinuous edges are subdivided into 14 canonical discontinuous edges that produce a known ORF and 15 non-canonical discontinuous edges. Next, we generated transcripts  $\mathcal{T}$  and their abundances  $\mathbf{c}$  from  $G$  using the negative-sense discontinuous transcription model (described in Supplementary Note C.1). Upon generating the transcripts, we simulated the generation and sequencing of RNA-seq data, and aligned the simulated reads using STAR<sup>26</sup>. We generated five independent pairs  $(\mathcal{T}, \mathbf{c})$  of transcripts and abundances (Fig. 3b). For each pair  $(\mathcal{T}, \mathbf{c})$  we generated five paired-end short-read sequencing simulations using polyester<sup>33</sup>. Thus, in total we generated  $5 \times 5 = 25$  simulation instances.

We compare the performance of our method JUMPER with two other reference-based transcript assembly methods, SCALLOP and STRINGTIE. Note that our method, JUMPER, does not use prior knowledge about the underlying negative-sense discontinuous transcription model to infer the viral transcripts from the simulated data. To avoid including false-positive discontinuous edges, we set  $\Lambda = 100$  so that JUMPER discards discontinuous edges with fewer than 100 supporting reads. For SCALLOP and STRINGTIE, we performed a sweep on their input parameters and report the best results here. We begin by comparing the transcripts predicted by the three methods to the ground-truth transcripts. Specifically, a predicted transcript is correct if there exists a transcript in the ground truth whose junction positions match the predicted junction positions within a tolerance of ten nucleotides.

Figure 3c shows the  $F_1$  score (harmonic mean of recall and precision) of the three methods for all the simulation instances, showing that JUMPER achieves a higher  $F_1$  score (median of



**Fig. 3 JUMPER consistently outperforms SCALLOP<sup>11</sup> and STRINGTIE<sup>12</sup> in reconstruction of viral transcripts from simulated SARS-CoV-2 sequencing data.** **a** Sashimi plot showing the canonical (black) and non-canonical (gray) discontinuous mappings supported by reads in short-read sample SRR11409417. **b** Number of canonical and non-canonical transcripts for five simulation instances of ( $T$ , **c**) generated under the negative-sense discontinuous transcription model. **c**  $F_1$  score of the three methods (JUMPER, SCALLOP, and STRINGTIE) for all the  $5 \times 5 = 25$  simulated instances (i.e. five technical replicates for each of the five simulated transcriptomes) under the negative-sense discontinuous transcription model. Box plots show the median and the interquartile range (IQR), and the whiskers denote the lowest and highest values within 1.5 times the IQR from the first and third quartiles, respectively. **d** Precision and recall values of the three methods with one of sequencing experiment for each simulated instance of ( $T$ , **c**) under the negative-sense discontinuous transcription model as input. **e** Total number of canonical and non-canonical transcripts recalled by the three methods for the simulated instances shown in panel (**d**).

0.255 and range [0.176, 0.339]) compared to SCALLOP (median of 0.062 and range [0.0145, 0.173]) and STRINGTIE (median of 0.019 and range [0.0114, 0.0412]). Supplementary Fig. 5 shows that JUMPER’s improved performance holds for both the recall and the precision with running times comparable to the SCALLOP and STRINGTIE. To investigate the effect of threshold parameter  $\Lambda$  on the performance of JUMPER, we ran our method on the simulated instances with  $\Lambda \in \{10, 50, 100, 200\}$ . Supplementary Fig. 6 shows that JUMPER outperforms SCALLOP and STRINGTIE for all values of  $\Lambda$ , although it incurs significantly more runtime for  $\Lambda = 10$ .

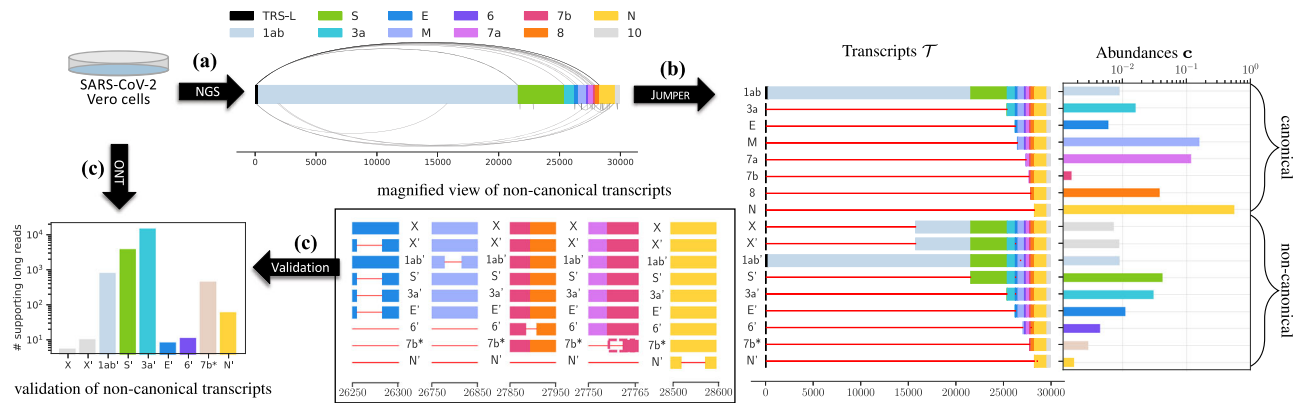
To better understand the tradeoff between precision and recall, we zoom in on five simulation instances with distinct pairs ( $T$ , **c**). Figure 3d shows the precision and recall achieved by each method for each of these five simulation instances, demonstrating that JUMPER consistently outperforms both SCALLOP and STRINGTIE. On average, JUMPER recalls 5 times more transcripts than SCALLOP and 11 times more transcripts than STRINGTIE while also having higher precision in all simulated cases. Supplementary Fig. 7 shows that all three methods produce similar precision and recall values for different sequencing replicates of the same simulated instance of ( $T$ , **c**), demonstrating consistency in results. Figure 3e shows the number of canonical and non-canonical transcripts generated by the three methods that match the ground truth for each simulated instance, with JUMPER consistently recalling a larger number of ground-truth canonical and non-canonical transcripts. To assess the accuracy of JUMPER’s

estimation of the abundances **c**, we computed the Pearson correlation between the abundances of the correctly recalled transcripts and their ground-truth abundances. We find that JUMPER achieves a median Pearson correlation of 0.979, and that the use of SALMON to re-estimate abundances improves the median correlation to 0.985 (Supplementary Fig. 8).

In summary, we found that JUMPER correctly predicts higher number of both canonical and non-canonical transcripts compared to SCALLOP and STRINGTIE for all the simulated instances (summarized in Supplementary Table 3). We observe similar trends on simulated instances of a human gene (see Supplementary Note C.4).

**Viral transcript assembly in SARS-CoV-2-infected Vero cells.**

Recently, Kim et al.<sup>3</sup> explored the transcriptomic architecture of SARS-CoV-2 by performing short-read as well as long-read sequencing of Vero cells infected by the virus. The authors used oligo(dT) amplification, which targets the poly(A) tail at the 3’ end of messenger RNAs, thus limiting positional bias that would occur when using SARS-CoV-2-specific primers<sup>34,35</sup>. Subsequently, the authors aligned the resulting reads using splice-aware aligners, STAR<sup>26</sup> for the short-read sample (median depth of 1763) and minimap<sup>236</sup> for the long-read sample (median depth of 6707 and mean length of 2875 bp). For both complementary sequencing techniques, the authors observed split reads that were indicative of canonical as well as non-canonical transcription events. While this previous work quantified the fraction of split



**Fig. 4** Using short-read data of SARS-CoV-2-infected Vero cells<sup>3</sup>, JUMPER identifies canonical and non-canonical transcripts that are well supported by long-read sequences of the same sample. **a** The segment graph for the short-read data contains both canonical (above) and non-canonical (below) edges. **b** JUMPER assembles eight canonical transcripts and nine non-canonical transcripts and estimates their abundances with zoomed-in view of the non-canonical transcripts X, X', 1ab', S', 3a', E', 6', 7b\*, and N'. **c** All non-canonical transcripts predicted by JUMPER are well supported by long-read data. NGS next-generation sequencing, ONT Oxford Nanopore Technologies.

reads supporting each discontinuous transcription event, it did not attempt to assemble complete viral transcripts.

We used JUMPER to reconstruct the SARS-CoV-2 transcriptome of the short-read sequencing sample using the BAM file obtained by running Kim et al.'s pipeline<sup>3</sup>. This was followed by running SALMON to identify precise transcript abundances. We note that running SCALLOP on the short-read data resulted in only a single, complete canonical transcript (corresponding to 'N') but required subsampling of the BAM file (to 20%) due to memory constraints, whereas STRINGTIE produced two incomplete transcripts ('ORF3a' and a non-canonical transcript with low support). On a segment graph with  $|V| = 59$  nodes and  $|E| = 93$  edges comprised of  $|E^c| = 35$  most abundant discontinuous edges, 18 of which canonical and 17 non-canonical (Fig. 4a), JUMPER identified 33 transcripts, 17 of which have an abundance of at least 0.001 as determined by SALMON (Fig. 4b). A subset of eight transcripts are canonical, containing at most one discontinuous edge with the 5' junction in TRS-L and the first ATG downstream of the 3' junction coinciding with the start codon of a known ORF. These canonical transcripts correspond to ORF1ab, ORF3a, E, M, ORF7a, ORF7b, ORF8, N. In particular, ORF1ab (abundance of 0.008) corresponds to the complete viral genome, necessary for viral replication. Notably, ORF10 is the only missing ORF in the identified transcriptome, which is in line with previous studies<sup>3,5</sup> that did not find evidence for active transcription of ORF10.

As mentioned, JUMPER inferred nine non-canonical transcripts, denoted as X, X', 1ab', S', 3a', 6', E', 7b\* and N'. Among these, transcripts 1ab', S', 3a' and 6' encode for the 1ab polypeptide, spike protein S, accessory protein 3a and accessory protein 6, respectively. Transcripts X and X' both contain the discontinuous edge going from position 68 to 15774, with the latter containing an additional discontinuous edge from position 26256 to 26284. The 5' end of the common discontinuous edge occurs within TRS-L, whereas the 3' end occurs in the middle of ORF1b but is out of frame with respect to the starting position of ORF1b (13468). Specifically, the start codon 'ATG' downstream of the 3' end is located at position 15812 and occurs within nsp12 (RdRp) and the first stop codon is located at position 15896, encoding for a peptide sequence of 28 amino acids. Interestingly, when we examined the reference genome, we observed matching sequences 'GAACCTTAA' near the 5' and 3' junctions of the discontinuous edge common to X and X', possibly explaining why the viral RdRp generated this jump (Supplementary Fig. 9a, b). Strikingly, both matching sequences are conserved within the

*Sarbecovirus* subgenus but not in other subgenera of the *Betacoronavirus* genus (Supplementary Fig. 9a, c). To further corroborate this transcript, we examined short- and long-read SARS-CoV-2 sequencing samples from the NCBI Sequence Read Archive (SRA). Specifically, we looked for the presence of reads potentially originating from transcript X focusing on high-quality samples with 100 or more leader-spanning reads (reads whose 5' end maps to the TRS-L region). We say a read  $r$  supports a transcript  $T$  if the discontinuous edges of  $r$  exactly match those of  $T$ , i.e.  $\pi(r) \subseteq \pi(T)$  and  $|\sigma^\oplus(r)| = |\sigma(T)|$  (Supplementary Fig. 10). We found ample support for transcript X in both short- and long-read samples on SRA, with 100 out of 351 short-read samples and 81 out of 653 long-read samples having more than 0.1% of leader-spanning reads supporting transcript X (Supplementary Fig. 11). We note that although this discontinuous transcription event was also observed in ref.<sup>5</sup>, the authors found no evidence of this transcript leading to a protein product in the ribo-seq data. Further research into a potentially regulatory function of this transcript is required.

As stated, the difference between transcripts X and X' is that the latter includes an additional discontinuous edge, corresponding to a short jump of  $\sim 27$  nucleotides between positions 26256 and 26284. This is an in-frame deletion inside ORF E, resulting in the loss of nine amino acids that span the N-terminal domain (four amino acids) and the transmembrane domain (five amino acids) of the E protein<sup>37</sup>. A similar in-frame deletion of 24 nucleotides (from position 26259 to 26284) was observed by Finkel et al.<sup>5</sup> that resulted in the loss of a subset of eight out of the nine amino acids in the deletion that we observed. Furthermore, it is possible that this common deletion is being selected for during passage in Vero E6 cells, which were used by both Kim et al.<sup>3</sup> and Finkel et al.<sup>5</sup>. Non-canonical transcripts S', 3a' and E' also contain the same discontinuous edge from position 26256 to 26284. While transcript E' produces a version of protein E with nine missing amino acids, transcripts S' and 3a' produce complete viral proteins S and 3a, respectively. Non-canonical transcript 6' differs from the canonical transcript 6, containing a jump from position 27886 to 27909. This jump is downstream of ORF6 and therefore does not disrupt the translation of accessory protein 6. Similarly, transcript 1ab' has a single jump from position 26779 to 26817, which is downstream of the ORF1ab gene and therefore will yield the complete polypeptide 1ab. Transcript 7b\*, on the other hand, has a single discontinuous edge from position 71 to 27762. The start codon 'ATG' downstream of the 3' end occurs at position 27825, maintaining the frame of 7b, and thus leading to

an N-terminal truncation<sup>3</sup> of 23 amino acids. Interestingly, transcript 7b and transcript 7b\* appear with similar abundances in our solution. Finally, transcript N' has one canonical discontinuous edge from TRS-L (position 65) to the transcription regulating body sequence (TRS-B) region corresponding to ORF N (position 28255) and an additional jump from position 28525 to 28577, which leads to an in-frame deletion of 17 amino acids in the N-terminal RNA-binding domain<sup>38,39</sup> of ORF N. Thus, with the exception of transcripts X and X', the non-canonical transcripts identified by JUMPER either produce complete viral proteins (1ab', S', 3a', 6'), contain in-frame deletions in the middle of known proteins (E', N') or produce N-terminally truncated proteins (7b\*).

One of the major findings of the Kim et al. paper<sup>3</sup> is that the SARS-CoV-2 transcriptome is highly complex owing to numerous non-canonical discontinuous transcription events. Strikingly, our results show that these non-canonical transcription events do not significantly change the resulting proteins. Indeed, we find that four out of the nine non-canonical transcripts produce a complete known viral protein and the total abundance of the predicted transcripts that produce a complete known viral protein is 0.968. Moreover, these predicted transcripts account for more than 90% of the reads in the sample according to the estimates provided by SALMON.

Typically, reads from short-read sequencing samples are not long enough to contain more than one discontinuous edge. As a result, short-read data can only provide direct evidence for transcripts with closely spaced discontinuous edges. For instance, we observed ample support (63485 short reads) for the predicted non-canonical transcript E', which has two discontinuous edges (69,26237) and (26256,26284), in short-read data due to the close proximity of the two discontinuous edges (i.e. the discontinuous edges are only  $26256 - 26237 = 19$  nucleotides apart). The other non-canonical transcripts with multiple discontinuous edges, i.e. X', S', 3a', 6' and N', have edges that are too far apart to be spanned by a single short read. Using the long-read sequencing data of this sample, we detected supporting long reads that span the exact set of discontinuous edges of all 9 non-canonical transcripts (Fig. 4c). Moreover, we found support for the canonical transcripts as well (Supplementary Fig. 12). Thus, all transcripts identified by JUMPER from the short-read data are supported by direct evidence in the long-read data.

In summary, using JUMPER, we reconstructed a detailed picture of the transcriptome of a short-read sequencing sample of Vero cells infected by SARS-CoV-2. While existing methods failed to recall even the reference transcriptome, JUMPER identified transcripts encoding for all known viral protein products. In addition, our method predicted non-canonical transcripts, whose presence we subsequently validated on a long-read sequencing sample of cells from the same cell line.

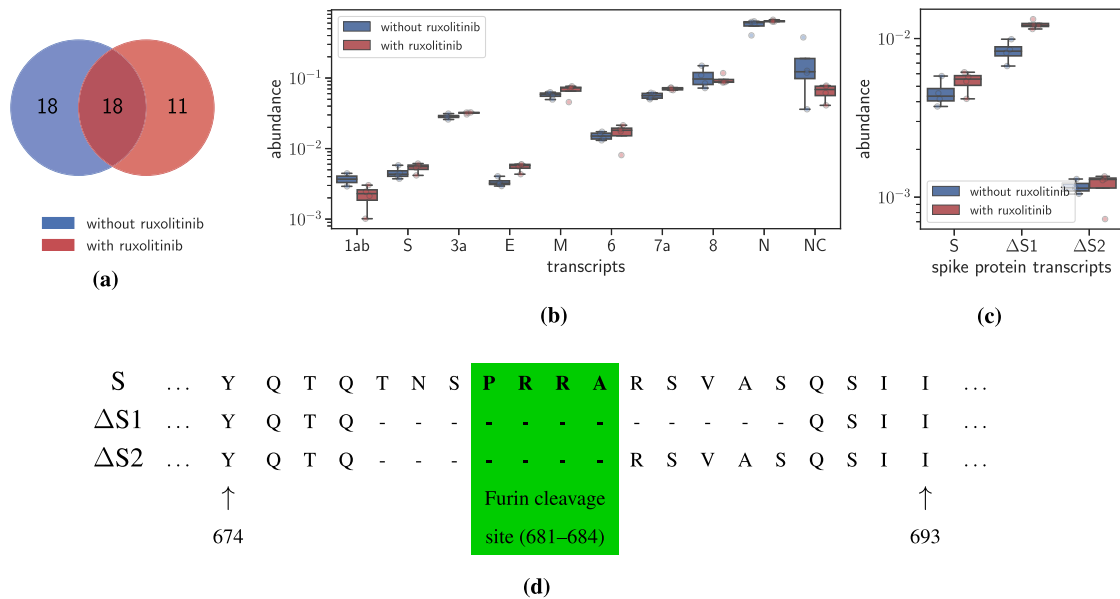
**Viral transcript assembly in SARS-CoV-2-infected A549 cells with and without treatment.** To demonstrate that JUMPER can be used to understand the effect of drugs on the viral transcriptome, we analyzed a recent dataset by Blanco et al.<sup>25</sup> who studied the host transcriptional response to SARS-CoV-2 and other viral infections using various cell lines. We focused on A549 lung alveolar cell line samples that were sequenced after 24 h of SARS-CoV-2 infection. There are a total of eight samples, four technical replicates that were pre-treated with ruxolitinib for 1 h before the infection and four technical replicates that were untreated. Ruxolitinib is a JAK1 and 2 kinase inhibitor, which blocks type-I interferon (IFN-I) signaling necessary to engage cellular antiviral defenses<sup>40,41</sup>. Specifically, the four samples without treatment are [SRR11573904](#) (median depth of 86),

[SRR11573905](#) (median depth of 85), [SRR11573906](#) (median depth of 89), and [SRR11573907](#) (median depth of 89), and the four samples treated with ruxolitinib are [SRR11573924](#) (median depth of 90), [SRR11573925](#) (median depth of 91), [SRR11573926](#) (median depth of 91), and [SRR11573927](#) (median depth of 92). We used `fastp` to trim the short reads (trimming parameter set to 10 nucleotides) and we aligned the resulting reads using STAR in two-pass mode. We ran JUMPER with the 35 most abundant discontinuous edges in the segment graph. Similarly to the previous analysis, we restricted our attention to transcripts identified by JUMPER that have more than 0.001 abundance as estimated by SALMON<sup>23</sup>.

SCALLOP, run with default parameters (Supplementary Note C.2), identified at most two transcripts for each sample encoding for different variants of ORF N. JUMPER identified a total of 47 transcripts across the eight samples, with 18 of these transcripts present in both ruxolitinib treated and untreated samples (Supplementary Fig. 13a, c). We observed that samples with pre-treatment of ruxolitinib cumulatively have fewer transcripts compared to the number of transcripts from samples without any treatment (29 vs. 36 transcripts, Fig. 5a). Strikingly, all the transcripts that are present in two or more samples were also present across the two groups of samples (treated and untreated). Focusing on the 18 common transcripts, Supplementary Fig. 11d shows the total number of samples that contain each of these 18 transcripts. A subset of 13 out of these 18 transcripts produce all known canonical viral proteins except 7b. Figure 5b shows the abundance of the transcripts yielding functional proteins in the samples along with 'NC' depicting the abundance of transcripts producing either non-canonical or non-functional viral proteins. The abundance of the canonical transcripts, except 1ab, is slightly higher in samples with treatment compared to the samples without treatment. Consequentially, the abundance of non-canonical transcripts is lower in samples with treatment compared to samples without treatment.

There are five non-canonical transcripts, including VM, NC1, and NC2, which do not encode for known SARS-CoV-2 proteins but are explained by matching motifs near the 5' and 3' ends of the non-canonical discontinuous edges, described in Supplementary Table 4, potentially mediating the jump made by the RdRp to generate these transcripts. Specifically, while transcript VM contains a canonical discontinuous edge from the leader to the known TRS-B region of M, it also contains an out-of-frame deletion such that the transcript yields a 116 amino acids long protein which matches the M protein for the first 87 amino acids (total length of protein M is 222 amino acids). Both transcripts NC1 and NC2 contain only one jump with the 5' end within ORF1a. The 3' end of the jump lies within ORF7b and ORF N for transcript NC1 and transcript NC2, respectively. The remaining two non-canonical transcripts,  $\Delta$ S1 and  $\Delta$ S2, have in-frame deletions in the region that encodes for the spike protein.

$\Delta$ S1 contains an in-frame jump from position 23593 to 23630 resulting in a 12 amino-acid in-frame deletion, while  $\Delta$ S2 contains a jump from position 23593 to 23615, which results in a 7 amino-acid in-frame deletion in the spike protein (Fig. 5d). Both these deletions overlap with the furin cleavage site (FCS), highlighted in Fig. 5d, which has been the focus of several recent studies<sup>4,42,43</sup>. The authors of ref. 4 deduced that the deletion of the FCS enhances the ability of the virus to enter Vero cells and is selected for during passage in Vero E6 cells, a cell line that lacks a working type-I interferon response. The observation of  $\Delta$ S1 and  $\Delta$ S2 in infected A549 cell samples can be explained by the fact that Blanco et al.<sup>25</sup> propagated SARS-CoV-2 in Vero E6 cells prior to the infection of the A549 cells. Figure 5c shows that pre-treatment with ruxolitinib leads to an increase in the abundance of the three transcripts, S (median increase from 0.004 to 0.005),



**Fig. 5** JUMPER enables analysis of drug response in SARS-CoV-2-infected cells<sup>25</sup> at the transcript level. **a** A Venn diagram showing the number of transcripts reconstructed from four samples with and four samples without treatment with ruxolitinib (i.e. two groups of four technical replicates). Supplementary Fig. 13 shows the distribution of the 18 transcripts that are common between samples with and without treatment while Supplementary Table 4 describes these transcripts. **b** Abundance of the transcripts yielding canonical proteins in the samples along with 'NC' depicting the abundance of the non-canonical transcripts. **c** Abundance of the transcripts yielding the spike protein (S) and its variants  $\Delta S1$  and  $\Delta S2$  whose structure is described in **(d)**. Box plots show the median and the interquartile range (IQR), and the whiskers denote the lowest and highest values within 1.5 times the IQR from the first and third quartiles, respectively.

$\Delta S1$  and  $\Delta S2$  (median increase from 0.0011 to 0.0012), with the increase being most significant for  $\Delta S1$  (median increase from 0.008 to 0.012) with a  $p$  value of 0.015 with the Mann–Whitney  $U$  test. This shows that the response of different transcripts of the virus to treatment of drugs can differ significantly. In summary, we find that JUMPER enables transcript-level analysis of the viral response to drug treatments.

**Viral transcript assembly in SARS-CoV-1- and MERS-CoV-infected cells.** To show the generalizability of our method, we considered two other coronaviruses, SARS-CoV-1 and MERS-CoV. We describe the results for two SARS-CoV-1-infected cell samples here and the analysis of three MERS-CoV-infected cell samples is described in Supplementary Note C.5.

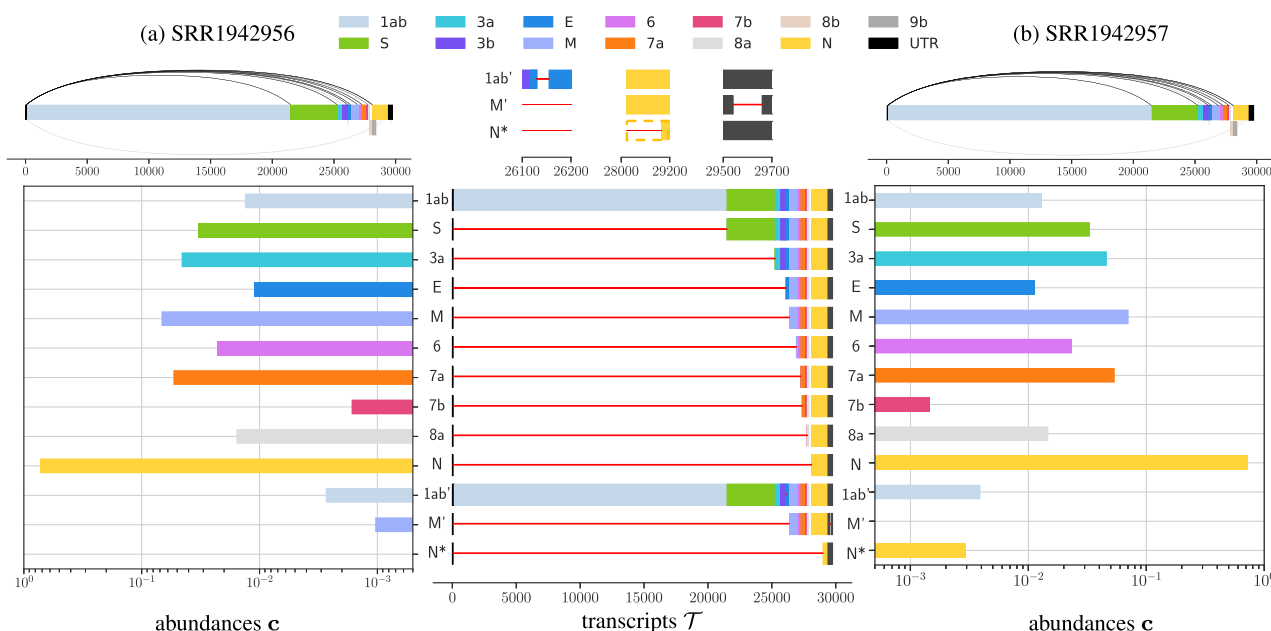
We analyzed two published samples of human Calu-3 cells infected with SARS-CoV-1<sup>24</sup>, [SRR1942956](#) and [SRR1942957](#), with a median depth of 21,358 and 20,991, respectively. These two samples originate from the same SRA project ('PRJNA279442') whose metadata states that both samples were sequenced 24 h after infection. We used *fastp* to trim the short reads (trimming parameter set to 10 nucleotides) and we aligned the resulting reads using *STAR* in two-pass mode. We ran JUMPER with the 35 most abundant discontinuous edges in the segment graph. As observed previously, SCALOP only identified a single transcript corresponding to ORF N in both the samples. By contrast, JUMPER reconstructed 25 transcripts in sample [SRR1942956](#) and 26 transcripts for sample [SRR1942957](#). Similarly to the previous analysis, we discuss the transcripts identified by JUMPER that have more than 0.001 abundance as estimated by SALMON. There are 13 such transcripts for sample [SRR1942956](#) and 13 such transcripts for sample [SRR1942957](#) (Fig. 6).

SARS-CoV-1 has a genome of length 29,751 bp, and consists of 13 ORFs (1ab, S, 3a, 3b, E, M, 6, 7a, 7b, 8a, 8b, N and 9b), two more than SARS-CoV-2. For both samples, JUMPER identified

canonical transcripts corresponding to all the ORFs of SARS-CoV-1 except ORF3b, ORF8b and ORF9b (Fig. 6). Notably, ORF8b and ORF9b share transcription regulating body sequences (TRS-B) with ORF8a and ORF N respectively<sup>44</sup>. More specifically, ORF9b (from position 28130 to 28426) is nested within ORF N (from position 28120 to 29388) with start codons only 10 nucleotides apart and consequently shares the same TRS-B as ORF N. ORF8b (from position 27864 to 28118) intersects with ORF8a (from 27779 to 27898) and previous studies have failed to validate a TRS-B region for ORF8b<sup>44</sup>. One possible way that these ORFs are translated is due to ribosome leaky scanning, which was also hypothesized to lead to ORF7b translation in SARS-CoV-2<sup>5</sup>. This explains why JUMPER was unable to identify transcripts that directly encode for 8b and 9b. Regarding ORF3b, JUMPER did identify a canonical transcript corresponding to 3b in both samples, but the SALMON estimated abundances (0.00044 for [SRR1942956](#) and 0.0005 for [SRR1942957](#)) for these transcripts were below the cut-off value of 0.01. Finally, we note that the relative abundances of the canonical transcripts are consistent for the two samples (Fig. 6) and ranked in the same order (Supplementary Fig. 14), with ORF7b being the least abundant and ORF N having the largest abundance, in line with the observations in SARS-CoV-2-infected cells described in the previous sections.

Figure 6 shows the three non-canonical transcripts predicted by JUMPER in the two SARS-CoV-1 samples, designated as 1ab', M' and N\*. Since these non-canonical transcripts are in very low abundance, we see some discrepancy in the prediction between the two samples. The first non-canonical transcript 1ab' with a single short discontinuous edge from position 26131 to 26156 is detected in both samples and has a very low abundance compared to the canonical transcript 1ab (0.0133 for 1ab vs. 0.002 for 1ab' in [SRR1942956](#), and 0.013 for 1ab vs. 0.0039 for 1ab' in [SRR1942956](#)). Since the discontinuous edge occurs downstream of the stop codon of 1ab (position 21492), the 1ab' transcript





**Fig. 6** JUMPER identifies canonical and non-canonical transcripts that recur in two short-read sequencing samples of SARS-CoV-1-infected Calu-3 cells<sup>24</sup>. For both the samples, **a** SRR1942956 and **b** SRR1942957, we show the segment graph, with canonical (above) and non-canonical (below) discontinuous edges. We also show the predicted transcripts and their abundances in the two samples with a zoomed-in view of the non-canonical transcripts 1ab', M' and N\*. UTR: untranslated region.

encodes for the complete polypeptide 1ab. The second non-canonical transcript M' has two discontinuous edges: a canonical discontinuous edge from TRS-L (position 65) to TRS-B of ORF M (position 26351) and a non-canonical discontinuous edge from 29542 to 29661 in the 3' untranslated region (UTR). As such, this transcript encodes for the complete M protein. This transcript is detected in SRR1942956 with a very low abundance of 0.001 and is detected at an even lower abundance of 0.0008 in SRR1942957, which is below the cut-off threshold of 0.001. The third non-canonical transcript, denoted by N\*, has a single discontinuous edge from position 65 to 29003. While JUMPER and SALMON detected this transcript only in sample SRR1942957 with a low abundance of 0.003, we do observe 119 reads in SRR1942956 (compared to 151 reads in SRR1942957) that support this edge, suggesting that N\* might be present in the latter sample at too small of an abundance to be detected. Transcript N\* is interesting because the first 'ATG' downstream of the 3' end of its discontinuous edge occurs at position 29071 maintaining the frame of N (which starts at position 28120). Thus transcript N\* encodes for an N-terminally truncated version of protein N with 105 amino acids (while protein N is composed of 422 amino acids) and only contains part of the C-terminal dimerization domain<sup>38</sup> of protein N. This is similar to transcript 7b\* in the SARS-CoV-2-infected Vero cell sample, which yields a N-terminal truncated version of protein 7b. Detection of non-canonical transcripts such as E' and 7b\* in SARS-CoV-2 and N' in SARS-CoV-1 suggests that generation of N-terminally truncated proteins might be a common feature in coronaviruses.

In summary, JUMPER can be used to reconstruct the transcriptome of all viruses in *Nidovirales* and lead to discovery of novel viral transcripts and corresponding viral proteins. While this section focused on SARS-CoV-1, we observed similar results for MERS-CoV samples, where JUMPER reconstructed transcripts corresponding to all the ORFs with well-supported TRS-B sites along with consistent abundances across the three samples (see Supplementary Note C.5).

## Discussion

In this paper, we formulated the DISCONTINUOUS TRANSCRIPT ASSEMBLY (DTA) problem of reconstructing viral transcripts from short-read RNA-seq data of coronaviruses. The discontinuous transcription process exhibited by the viral RNA-dependent RNA polymerase (RdRp) is distinct from alternative splicing observed in eukaryotes. Our proposed method, JUMPER, is specifically designed to reconstruct the viral transcripts generated by discontinuous transcription and is therefore able to outperform existing transcript assembly methods such as SCALLOP and STRINGTIE, as we have shown in both simulated and real data.

For real-data analysis, we used publicly available short-read and long-read sequencing data of the same sample of SARS-CoV-2-infected Vero cells<sup>3</sup>. We performed transcript assembly using the short-read sequencing data and used the long-read data for validation. JUMPER was able to identify transcripts encoding for all known viral proteins except ORF10, which has been shown to have little support of active transcription in previous studies<sup>3,5</sup>. Moreover, we predicted nine non-canonical transcripts that are well supported by long-read sequencing data.

Furthermore, we demonstrated that JUMPER enables transcript-level quantitative analysis of viral response to treatment with drugs. More specifically, we analyzed eight samples of A549 lung alveolar cells infected by SARS-CoV-2, four of which were pre-treated with ruxolitinib for 1 h before infection<sup>25</sup>. JUMPER identified one variant of the spike protein, with a 12 amino acid deletion overlapping with the furin cleavage site, that showed statistically significant increase in expression in samples that were pre-treated with ruxolitinib. We also showed the versatility of JUMPER by considering two additional coronaviruses, SARS-CoV-1 and MERS-CoV. For two samples of Calu-3 cells infected by SARS-CoV-1 and three samples of Calu-3 cells infected by MERS-CoV<sup>24</sup>, JUMPER reconstructed all the canonical transcripts with distinct TRS-B regions and additionally predicted the presence of non-canonical transcripts encoding for either complete or truncated versions of known viral proteins.

There are several avenues for future work. First, JUMPER currently is only applicable to data obtained using technologies that limit positional bias such as oligo(dT) amplification, which targets the poly(A) tail at the 3' end of messenger RNAs. We plan to extend our current model to account for positional and sequencing biases in the data. Doing so will enable us to assemble transcriptomes from sequencing samples that used SARS-CoV-2-specific primers, which form the majority of currently available data. Second, we currently make the assumption of a fixed read length that is much smaller than the length of viral transcripts. We will relax this assumption in order to support long-read sequencing data that have variable read lengths, similar to previous methods such as Bayessembler<sup>16</sup> and Scallop-LR<sup>45</sup>. Third, we plan to study the effect of mutations (including single-nucleotide variants as well as indels) on the transcriptome. Along the same lines, there is evidence of within-host diversity in COVID-19 patients<sup>46–51</sup>. It will be interesting to study whether this diversity translates to distinct sets of transcripts and abundances within the same host. Fourth, there are possibly multiple optimal solutions to the DTA problem that present equally likely viral transcripts with different relative abundances in the sample. A useful direction of future work is to explore the space of optimal solutions similar to the work done in ref. <sup>28</sup>. Finally, the approach presented in this paper can be extended to the general transcript assembly problem. Although JUMPER can be used for transcript assembly of individual eukaryotic genes (see Supplementary Note C.4), it does not currently support assembly across multiple genes. The extension of the current approach can be facilitated by using the topological ordering of the nodes in a general splice graph that does not have a unique Hamiltonian path, unlike the segment graph considered in the DTA problem. We envision this will facilitate efficient use of combinatorial optimization tools such as integer linear programming to transcript assembly problems.

**Methods**

**Combinatorial characterization of solutions.** Equation (1) defines the probability  $\Pr(\mathcal{R}|T, \mathbf{c})$  in terms of the observed reads  $r$  and their induced paths  $\pi(r) \subseteq E(G)$  in the segment graph  $G$ . The authors in ref. <sup>28</sup> use this characterization of reads as paths in a general splice graph to account for ambiguity in the transcript of origin for the reads. For a general splice graph, such a characterization is required to capture all the possible observed reads. However, in our setting, where the segment graph  $G$  is a DAG with a unique Hamiltonian path, it is possible to describe each read and each transcript uniquely in a more concise form. Each path in the segment graph is characterized by a set of non-overlapping discontinuous edges. To describe this, we introduce the following definition.

**Definition 3.** Two edges  $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$  and  $(\mathbf{x} = [x^-, x^+], \mathbf{y} = [y^-, y^+])$  of *Overlap* if the open intervals  $(v^+, w^-)$  and  $(x^+, y^-)$  intersect, i.e.  $(v^+, w^-) \cap (x^+, y^-) \neq \emptyset$ .

For any transcript  $T$  corresponding to an  $s - t$  path in  $G$ , for which we are only given its discontinuous edges  $\sigma(T)$ , the continuous edges of  $T$  are uniquely determined by  $G$  and  $\sigma(T)$ . That is, the continuous edges of  $T$  equal precisely the subset of continuous edges  $E^{\rightarrow}$  that do not overlap with any of the discontinuous edges in  $\sigma(T)$ . Conversely, given an  $s - t$  path  $\pi(T)$  of  $G$  the corresponding set of discontinuous edges is given by  $\sigma(T) = \pi(T) \cap E^{\leftarrow}$ . Thus, we have the following proposition with the proof in Supplementary Note B.1.

**Proposition 1.** *There is a bijection between subsets of discontinuous edges that are pairwise non-overlapping and  $s - t$  paths in  $G$ .*

In a similar vein, rather than characterizing a read  $r$  by its induced path  $\pi(r) \subseteq E$  in the segment graph, we characterize a read  $r$  by a pair  $(\sigma^{\oplus}(r), \sigma^{\ominus}(r))$  of characteristic discontinuous edges. Here,  $\sigma^{\oplus}(r)$  is the set of discontinuous edges that must be present in any transcript that could generate read  $r$ , i.e.  $\sigma^{\oplus}(r) = \pi(r) \cap E^{\leftarrow}$ . Conversely,  $\sigma^{\ominus}(r)$  is the set of discontinuous edges that must be absent in any transcript that could generate read  $r$  due to the unidirectional nature of RdRp transcription. Thus, the set  $\sigma^{\ominus}(r)$  consists of discontinuous edges  $E^{\leftarrow} \setminus \sigma^{\oplus}(r)$  that overlap with an edge in  $\pi(r)$ . Clearly, while  $\sigma^{\oplus}(r) \cap \sigma^{\ominus}(r) = \emptyset$ , it need not hold that  $\sigma^{\oplus}(r) \cup \sigma^{\ominus}(r)$  equals  $E^{\leftarrow}$  (see Fig. 2b). Formally, we define  $(\sigma^{\oplus}(r), \sigma^{\ominus}(r))$  as follows.

**Definition 4.** The *characteristic discontinuous edges* of a read  $r$  are a pair  $(\sigma^{\oplus}(r), \sigma^{\ominus}(r))$  where  $\sigma^{\oplus}(r)$  is the set of discontinuous edges present in read  $r$ , i.e.  $\sigma^{\oplus}(r) = \pi(r) \cap E^{\leftarrow}$ ,

and  $\sigma^{\ominus}(r)$  is the set of discontinuous edges  $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+]) \in E^{\leftarrow} \setminus \sigma^{\oplus}(r)$  that overlaps with an edge  $(\mathbf{x} = [x^-, x^+], \mathbf{y} = [y^-, y^+])$  in  $\pi(r)$ .

We have the following result with the proof given in Supplementary Note B.1.

**Proposition 2.** *Let  $G$  be a segment graph,  $T$  be a transcript and  $r$  be a read. Then,  $\pi(T) \supseteq \pi(r)$  if and only if  $\sigma(T) \supseteq \sigma^{\oplus}(r)$  and  $\sigma(T) \cap \sigma^{\ominus}(r) = \emptyset$ .*

Hence, we may rewrite the likelihood  $\Pr(\mathcal{R}|T, \mathbf{c})$  as

$$\prod_{j=1}^n \frac{1}{\sum_{b=1}^k c_b L_b} \sum_{i \in X(T, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i \tag{2}$$

where  $\mathcal{R} = \{r_1, \dots, r_n\}$ ,  $T = \{T_1, \dots, T_k\}$ ,  $\mathbf{c} = [c_1, \dots, c_k]$ , and where  $X(T, \sigma_j^{\oplus}, \sigma_j^{\ominus})$  be the subset of indices  $i$  corresponding to transcripts  $T_i \in T$  where  $\sigma(T_i) \supseteq \sigma_j^{\oplus}$  and  $\sigma(T_i) \cap \sigma_j^{\ominus} = \emptyset$ . Note that the only difference between Eq. (2) and the formulation in Eq. (1) is the way that the candidate transcripts of origin for a given read are described. In Eq. (1), they are described as paths in the segment graph whereas in Eq. (2), they are described by sets of pairwise non-overlapping discontinuous edges in the segment graph. This leads to the following theorem.

**Theorem 1.** *For any alignment  $\mathcal{R}$ , transcripts  $T$  and abundances  $\mathbf{c}$ , Eqs. (1) and (2) are identical.*

Although we have described the formulation for single-end reads, this characterization is applicable to paired-end and even synthetic long reads. Moreover, our implementation provides support for both single-end and paired-end read samples with a fixed read length. The above characterization using discontinuous edges allows us to reduce the number of terms in the likelihood function since multiple reads can be characterized by the same characteristic discontinuous edges. We describe this in detail in the next section.

**JUMPER: a progressive heuristic for the DTA problem.** To solve the DTA problem, we use the results of the previous section to write a more concise form of the likelihood. Specifically, let  $S = \{(\sigma_1^{\oplus}, \sigma_1^{\ominus}), \dots, (\sigma_m^{\oplus}, \sigma_m^{\ominus})\}$  be the set of pairs of characteristic discontinuous edges generated by the reads in alignment  $\mathcal{R}$ . Let  $\mathbf{d} = \{d_1, \dots, d_m\}$ , where  $d_i$  is the number of reads that map to pair  $(\sigma_i^{\oplus}, \sigma_i^{\ominus}) \in S$ . Using that reads  $r$  with identical characteristic discontinuous edges  $(\sigma^{\oplus}(r), \sigma^{\ominus}(r))$  have identical probabilities  $\Pr(r|T, \mathbf{c})$ , we obtain the following mathematical program for the log-likelihood  $\log \Pr(\mathcal{R}|T, \mathbf{c})$  (see Supplementary Note A.2 for derivation).

$$\max_{T, \mathbf{c}} \sum_{j=1}^m d_j \log \sum_{i \in X(T, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i - m \log \sum_{b=1}^k c_b L_b \tag{3}$$

$$\text{s.t.} \quad \pi(T_i) \text{ is an } s - t \text{ path} \\ \text{in the segment graph } G, \forall i \in [k], \tag{4}$$

$$\sum_{i=1}^k c_i = 1, \tag{5}$$

$$c_i \geq 0, \quad \forall i \in [k]. \tag{6}$$

Observe that the first sum (over reads) is concave and the second sum (over transcripts) is convex. Since we are maximizing, our objective function would ideally be concave. In Supplementary Note B.1, we prove the following lemma, which enables us to remove the second term using a scaling factor for the relative abundances  $\mathbf{c}$  that does not alter the solution space.

**Lemma 1.** *Let  $D > 0$  be a constant,  $\bar{c}_i(\mathbf{c}) = c_i D / \sum_{j=1}^k c_j L_j$  and  $c_i(\bar{\mathbf{c}}) = \bar{c}_i / \sum_{j=1}^k \bar{c}_j$  for all  $i \in [k]$ . Then,  $(T, \mathbf{c} = [c_1(\bar{\mathbf{c}}), \dots, c_k(\bar{\mathbf{c}})])$  is an optimal solution for (3)–(6) if and only if  $(T, \bar{\mathbf{c}} = [\bar{c}_1(\mathbf{c}), \dots, \bar{c}_k(\mathbf{c})])$  is an optimal solution for*

$$\max_{T, \bar{\mathbf{c}}} \sum_{j=1}^m d_j \log \sum_{i \in X(T, \sigma_j^{\oplus}, \sigma_j^{\ominus})} \bar{c}_i \tag{7}$$

$$\text{s.t.} \quad \pi(T_i) \text{ is an } s - t \text{ path} \\ \text{in the segment graph } G, \forall i \in [k], \tag{8}$$

$$\sum_{i=1}^k \bar{c}_i L_i = D, \tag{9}$$

$$\bar{c}_i \geq 0, \quad \forall i \in [k]. \tag{10}$$

We formulate the mathematical program given in Lemma 1 as a mixed integer linear program. More specifically, we encode (i) the composition of each transcript  $T_i$  as a set  $\sigma(T_i)$  of non-overlapping discontinuous edges, (ii) the abundance  $c_i$  and length  $L_i$  of each transcript  $T_i$ , (iii) the total abundance  $\sum_{i \in X(T, \sigma_j^{\oplus}, \sigma_j^{\ominus})} c_i$  of transcripts supported by characteristic discontinuous edges  $(\sigma_j^{\oplus}, \sigma_j^{\ominus})$ , and (iv) a piecewise linear approximation of the log function using a user-specified number  $h$  of breakpoints. We will describe (i) and (ii) in the following and refer to Supplementary Note B.2 for (iii) and (iv).

**Transcript composition.** We begin modeling (8), which states that each transcript  $T_i$  must correspond to an  $s - t$  path in the segment graph  $G$ . Using Proposition 1, we introduce binary variables  $x_e \in \{0, 1\}^{E^{\gamma} \times k}$  to encode the presence of discontinuous edges in each of the  $k$   $s - t$  paths corresponding to the  $k$  transcripts in  $\mathcal{T}$ . For any discontinuous edge  $e = (v = [v^-, v^+], w = [w^-, w^+])$ , let  $I(e)$  denote the open interval  $(v^+, w^-)$  between the two segments  $v$  and  $w$ . By Proposition 1, it must hold that  $I(e) \cap I(e') = \emptyset$  for any two distinct discontinuous edges  $e$  and  $e'$  assigned to the same transcript. To encode this, we impose

$$x_{e,i} + x_{e',i} \leq 1, \quad \forall i \in [k], e, e' \in E^{\gamma}$$

$$\text{s.t. } e \neq e', I(e) \cap I(e') \neq \emptyset.$$

**Transcript abundance and length.** We introduce non-negative continuous variables  $c = [c_1, \dots, c_k]$  that encode the abundance of the  $k$  transcripts. The scale of these abundances depends on the choice of  $D$ . We choose  $D = \ell^*$  where  $\ell^*$  is the length of the shortest  $s - t$  path in the segment graph  $G$ . Substituting  $D = \ell^*$  into (9) yields  $\sum_{i=1}^k c_i L_i = \ell^*$ .

Since  $c_i L_i \leq \sum_{j=1}^k c_j L_j = \ell^*$  and  $L_i \geq \ell^*$ , we have that  $c_i \leq 1$ . To model the product  $c_i L_i$  of the length  $L_i$  of a transcript  $T_i$  and its abundance  $c_i$ , we focus on individual discontinuous edges  $e$ . For any discontinuous edge  $e = (v = [v^-, v^+], w = [w^-, w^+])$ , let  $L(e) = w^- - v^+$  be the length of the interval. Observe that

$$c_i L_i = c_i L - c_i \sum_{e \in \sigma(T_i)} L(e) = c_i L - \sum_{e \in E^{\gamma}} c_i x_{e,i} L(e).$$

We introduce continuous variables  $z_e \in [0, 1]^k$  and encode the product  $z_{e,i} = c_i x_{e,i}$  for all  $e \in E^{\gamma}$  as

$$z_{e,i} \leq c_i, \quad \forall i \in [k],$$

$$z_{e,i} \leq x_{e,i}, \quad \forall e \in E^{\gamma}, i \in [k],$$

$$z_{e,i} \geq c_i + x_{e,i} - 1, \quad \forall e \in E^{\gamma}, i \in [k].$$

Therefore, we may represent  $\sum_{i=1}^k c_i L_i = \ell^*$  as

$$\sum_{i=1}^k c_i L - \sum_{i=1}^k \sum_{e \in E^{\gamma}} z_{e,i} L(e) = \ell^*. \quad (11)$$

The resulting formulation has  $O(|E^{\gamma}|k + |E^{\gamma}|m + mh)$  variables, where  $h$  is the user-specified number of breakpoints used in the piecewise linear approximation of the log function. This number includes  $|E^{\gamma}|k$  binary variables. The number of constraints is  $O(k|E|^2 + |E|km)$ .

**Progressive heuristic.** In practice, the number of discontinuous edges in the segment graph is inflated due to ambiguity in the exact location at which the RdRp jumps as well as sequencing and alignment errors. This leads to large number of binary variables in our MILP (we have  $k \cdot |E^{\gamma}|$  binary variables) which can make the MILP intractable. In order to approximately solve the problem with large values of  $k$ , we implement a progressive heuristic. Our heuristic takes as input the alignment  $\mathcal{R}$  and an integer  $k$ , which is the maximum number of transcripts in the solution. At each iteration  $p \leq k$ , we are given a set  $\mathcal{T}$  of  $p - 1$  previously computed transcripts and seek a new transcript  $T'$  by solving the MILP (see Supplementary Note B.3 for details) using function SOLVEILP with additional constraints to fix the values of the variables that encode the presence/absence of discontinuous edges for the transcripts in  $\mathcal{T}$ . The resulting reduction in number of binary variables from  $|E^{\gamma}|k$  to  $|E^{\gamma}|$  improves the running time of the MILP. As an additional optimization, we re-estimate the abundances of a new set  $\mathcal{T}'$  of transcripts. This set contains all transcripts in  $\mathcal{T}$  as well as additional transcripts corresponding to all possible subsets of discontinuous edges  $\sigma(T')$  of the newly identified transcript  $T'$ , identified by the function EXPAND. We solve a linear program (see Supplementary Note B.3 for details) with function SOLVELP to re-estimate the abundances  $c'$  of  $\mathcal{T}'$ , retaining only the top  $p$  transcripts  $T_i$  from  $\mathcal{T}'$  with the largest abundances  $c_i L_i$ . We terminate upon convergence, i.e. if  $\mathcal{T} = \mathcal{T}'$ , or if the number  $p$  of iterations reaches the number  $k$ . We note that a segment graph  $G$  with  $|E^{\gamma}|$  discontinuous edges induces a space of  $2^{|E^{\gamma}|}$ , thus providing a theoretical upper bound for  $k$ . However, in practice, we typically restrict our attention to the set of transcripts that exceed a minimum abundance threshold, resulting in a much smaller value for  $k$ . Algorithm 1 provides the pseudo code of the progressive heuristic implemented in JUMPER. The details of the subproblems SOLVEILP and SOLVELP are given in Supplementary Note B.3.

**Algorithm 1.** JUMPER( $\mathcal{R}, k$ )

```

1  $(\mathcal{T}, c) \leftarrow (\emptyset, [])$ 
2 for  $p \leftarrow 1$  to  $k$  do
3    $T' \leftarrow \text{SOLVEILP}(\mathcal{T})$ 
4    $\mathcal{T}' \leftarrow \mathcal{T} \cup \text{EXPAND}(T')$ 
5    $c' \leftarrow \text{SOLVELP}(\mathcal{T}')$ 
6   Sort  $(\mathcal{T}', c')$  s.t.  $L_i c'_i \geq L_{i+1} c'_{i+1}$  for all  $i \in \{1, \dots, |\mathcal{T}'| - 1\}$ 
7    $(\mathcal{T}, c) \leftarrow ((T_1, \dots, T_p), [c'_1, \dots, c'_p])$ 
8   if  $\mathcal{T}' \neq \mathcal{T}$  then
9      $(\mathcal{T}, c) \leftarrow (\mathcal{T}', c')$ 
10  else
11    return  $(\mathcal{T}, c)$ 
12 return  $(\mathcal{T}, c)$ 

```

**Implementation details.** Matching core sequences that mediate the discontinuous transcription by RdRp lead to ambiguity in precise location of breakpoint during alignment of spliced reads. Therefore, in practice we observe multiple discontinuous edges with closely spaced 5' and 3' breakpoints. Moreover, false-positive discontinuous edges are introduced due to sequencing and alignment errors. We use a threshold on the number of spliced reads supporting a discontinuous edge to filter false-positive edges with low support. This parameter can also be used to reduce computational burden and focus on the highly expressed transcripts in the sample. A discussion on the choice of the thresholding parameter  $\Lambda$  is provided in Supplementary Note B.4.

**Reporting summary.** Further information on research design is available in the Nature Research Reporting Summary linked to this article.

**Data availability**

The sequencing data deposited by Kim et al.<sup>3</sup> into the Open Science Framework (OSF) at <https://doi.org/10.17605/OSF.IO/8F6N9> were analyzed. The accession numbers of data available on SRA analyzed in this study are—SRR11573904, SRR11573905, SRR11573906, SRR11573907, SRR11573924, SRR11573925, SRR11573926, SRR11573927, SRR1942956, SRR1942957, SRR10357372, SRR10357373, and SRR10357374. All the simulated data generated in this study have been deposited to the Illinois Databank and are available at <https://databank.illinois.edu/datasets/IDB-6667667>. The analyzed and processed real and simulated data and results are available at <https://github.com/elkebir-group/Jumper-data>.

**Code availability**

The code has been deposited on Github at <https://github.com/elkebir-group/Jumper52>.

Received: 7 June 2021; Accepted: 20 October 2021;

Published online: 18 November 2021

**References**

- Vries, Antoine A. F. de, Marian C. Horzinek, Peter J. M. Rottier, and Raoul J. de Groot. "The Genome Organization of the Nidovirales: Similarities and Differences between Arteri-, Toro-, and Coronaviruses." *Seminars in Virology* 8, no. 1 (February 1997): 33–47. <https://doi.org/10.1006/smvy.1997.0104>.
- Maier, H. J. et al. *Coronaviruses: Methods and Protocols* (Springer Berlin, 2015).
- Kim, D. et al. The architecture of SARS-CoV-2 transcriptome. *Cell* 181, 914–921 (2020).
- Davidson, A. D. et al. Characterisation of the transcriptome and proteome of SARS-CoV-2 reveals a cell passage induced in-frame deletion of the furin-like cleavage site from the spike glycoprotein. *Genome Med.* 12, 1–15 (2020).
- Finkel, Y. et al. The coding capacity of SARS-CoV-2. *Nature* 589, 125–130 (2021).
- Robertson, G. et al. De novo assembly and analysis of RNA-seq data. *Nat. Methods* 7, 909–912 (2010).
- Grabherr, M. G. et al. Trinity: reconstructing a full-length transcriptome without a genome from RNA-Seq data. *Nat. Biotechnol.* 29, 644 (2011).
- Xie, Y. et al. SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads. *Bioinformatics* 30, 1660–1666 (2014).
- Chang, Z. et al. Bridger: a new framework for de novo transcriptome assembly using RNA-seq data. *Genome Biol.* 16, 30 (2015).
- Schulz, M. H., Zerbino, D. R., Vingron, M. & Birney, E. Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. *Bioinformatics* 28, 1086–1092 (2012).
- Shao, M. & Kingsford, C. Accurate assembly of transcripts through phase-preserving graph decomposition. *Nat. Biotechnol.* 35, 1167–1169 (2017).
- Pertea, M. et al. StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nat. Biotechnol.* 33, 290–295 (2015).
- Liu, J., Yu, T., Jiang, T. & Li, G. TransComb: genome-guided transcriptome assembly via combing junctions in splicing graphs. *Genome Biol.* 17, 213 (2016).
- Trapnell, C. et al. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat. Biotechnol.* 28, 511–515 (2010).
- Song, L., Florea, L. CLASS: constrained transcript assembly of RNA-seq reads. *BMC Bioinformatics* 14, S14 <https://doi.org/10.1186/1471-2105-14-S5-S14> (2013).
- Marett, L., Sibbesen, J. A. & Krogh, A. Bayesian transcriptome assembly. *Genome Biol.* 15, 1–11 (2014).
- Behr, J. et al. Mitie: simultaneous rna-seq-based transcript identification and quantification in multiple samples. *Bioinformatics* 29, 2529–2538 (2013).

18. Zhao, J., Feng, H., Zhu, D. & Lin, Y. Multitrans: an algorithm for path extraction through mixed integer linear programming for transcriptome assembly. *IEEE/ACM Trans. Comput. Biol. Bioinform.* <https://doi.org/10.1109/TCBB.2021.3083277> (2021).
19. Bernard, E., Jacob, L., Mairal, J. & Vert, J.-P. Efficient RNA isoform identification and quantification from RNA-Seq data with network flows. *Bioinformatics* **30**, 2447–2455 (2014).
20. Roberts, A., Pimentel, H., Trapnell, C. & Pachter, L. Identification of novel transcripts in annotated genomes using rna-seq. *Bioinformatics* **27**, 2325–2329 (2011).
21. Li, W., Feng, J. & Jiang, T. Isolasso: a lasso regression approach to rna-seq based transcriptome assembly. *J. Comput. Biol.* **18**, 1693–1707 (2011).
22. Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.* **34**, 525–527 (2016).
23. Patro, R., Duggal, G., Love, M. I., Irizarry, R. A. & Kingsford, C. Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods* **14**, 417–419 (2017).
24. Zhang, X. et al. Competing endogenous RNA network profiling reveals novel host dependency factors required for MERS-CoV propagation. *Emerg. Microbes Infect.* **9**, 733–746 (2020).
25. Blanco-Melo, D. et al. Imbalanced host response to SARS-CoV-2 drives development of COVID-19. *Cell* **181**, 1036–1045.e9 (2020).
26. Dobin, A. et al. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).
27. Bernard, E., Jacob, L., Mairal, J., Viara, E. & Vert, J.-P. A convex formulation for joint RNA isoform detection and quantification from multiple RNA-seq samples. *BMC Bioinform.* **16**, 1–10 (2015).
28. Zheng, H., Ma, C. & Kingsford, C. Deriving ranges of optimal estimated transcript expression due to non-identifiability. Preprint at *bioRxiv* <https://doi.org/10.1101/2019.12.13.875625> (2021).
29. Li, B. & Dewey, C. N. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC Bioinform.* **12**, 323 (2011).
30. Gurobi Optimization, L. *Gurobi Optimizer Reference Manual* <http://www.gurobi.com> (2020).
31. Li, H. et al. The sequence alignment/map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).
32. Yang, D. & Leibowitz, J. L. The structure and functions of coronavirus genomic 3' and 5' ends. *Virus Res.* **206**, 120–133 (2015).
33. Frazee, A. C., Jaffe, A. E., Langmead, B. & Leek, J. T. Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics* **31**, 2778–2784 (2015).
34. Gohl, D. M. et al. A rapid, cost-effective tailed amplicon method for sequencing SARS-CoV-2. *BMC Genomics* **21**, 1–10 (2020).
35. Quick, J. nCoV-2019 sequencing protocol v3 (LoCost). *protocols.io*. <https://protocols.io/view/ncov-2019-sequencing-protocol-v3-locost-bh42j8ye> (2020).
36. Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).
37. Mandala, V. S. et al. Structure and drug binding of the SARS-CoV-2 envelope protein transmembrane domain in lipid bilayers. *Nat. Struct. Mol. Biol.* **27**, 1202–1208 (2020).
38. Kang, S. et al. Crystal structure of SARS-CoV-2 nucleocapsid protein RNA binding domain reveals potential unique drug targeting sites. *Acta Pharmaceut. Sin. B* **10**, 1228–1238 (2020).
39. Ye, Q., West, A. M., Silletti, S. & Corbett, K. D. Architecture and self-assembly of the SARS-CoV-2 nucleocapsid protein. *Protein Sci.* **29**, 1890–1901 (2020).
40. Murira, A. & Lamarre, A. Type-I interferon responses: from friend to foe in the battle against chronic viral infection. *Front. Immunol.* **7**, 609 (2016).
41. Lee, J. S. & Shin, E.-C. The type I interferon response in COVID-19: implications for treatment. *Nat. Rev. Immunol.* **20**, 585–586 (2020).
42. Xia, S. et al. The role of furin cleavage site in SARS-CoV-2 spike protein-mediated membrane fusion in the presence or absence of trypsin. *Signal Transduct. Target. Ther.* **5**, 1–3 (2020).
43. Johnson, B. A. et al. Loss of furin cleavage site attenuates SARS-CoV-2 pathogenesis. *Nature* **591**, 293–299 (2021).
44. Yang, Y., Yan, W., Hall, A. B. & Jiang, X. Characterizing transcriptional regulatory sequences in coronaviruses and their role in recombination. *Mol. Biol. Evol.* <https://doi.org/10.1093/molbev/msaa281> (2020).
45. Tung, L. H., Shao, M. & Kingsford, C. Quantifying the benefit offered by transcript assembly with Scallop-LR on single-molecule long reads. *Genome Biol.* **20**, 1–18 (2019).
46. Sashittal, P., Luo, Y., Peng, J. & El-Kebir, M. Characterization of SARS-CoV-2 viral diversity within and across hosts. Preprint at *bioRxiv* <https://doi.org/10.1101/2020.05.07.083410> (2020).
47. Rose, R. et al. Intra-host site-specific polymorphisms of SARS-CoV-2 is consistent across multiple samples and methodologies. Preprint at *medRxiv* <https://doi.org/10.1101/2020.04.24.20078691> (2020).
48. Ramazzotti, Daniele, et al. "VERSO: a comprehensive framework for the inference of robust phylogenies and the quantification of intra-host genomic diversity of viral samples." *Patterns* **2.3**, 100212 (2021).
49. Shen, Z. et al. Genomic diversity of SARS-CoV-2 in coronavirus disease 2019 patients. *Clin. Infect. Dis.* <https://doi.org/10.1093/cid/ciaa203> (2020).
50. Karamitros, T. et al. SARS-CoV-2 exhibits intra-host genomic plasticity and low-frequency polymorphic quasispecies. *J. Clin. Virol.* **131**, 104585 (2020).
51. Tang, X. et al. On the origin and continuing evolution of SARS-CoV-2. *Nat. Sci. Rev.* **7**, 1012–1023 (2020).
52. Sashittal, P., Zhang, C. & El-Kebir, M. Jumper. <https://zenodo.org/badge/latestdoi/309318448> (2021).

## Acknowledgements

This material is based upon work supported by the National Science Foundation under award numbers DBI-1652815, CCF-1850502, CCF-2027669, and CCF-2046488.

## Author contributions

P.S. and M.E.-K. conceived the project, and developed the theory and algorithms. M.E.-K. and J.P. supervised the project. P.S., C.Z. and M.E.-K. wrote the paper. P.S. implemented the algorithms, and P.S. and C.Z. performed the analyses. All authors read and approved the final manuscript.

## Competing interests

The authors declare no competing interests.

## Additional information

**Supplementary information** The online version contains supplementary material available at <https://doi.org/10.1038/s41467-021-26944-y>.

**Correspondence** and requests for materials should be addressed to Mohammed El-Kebir.

**Peer review information** *Nature Communications* thanks Guojun Li and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

**Reprints and permission information** is available at <http://www.nature.com/reprints>

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

© The Author(s) 2021