JUMPER Enables Discontinuous Transcript Assembly in Coronaviruses

Palash Sashittal
University Of Illinois at Urbana-Champaign https://orcid.org/0000-0001-6581-4839
Chuanyi Zhang
University of Illinois at Urbana-Champaign https://orcid.org/0000-0002-6359-0904
Jian Peng
University of Illinois at Urbana Champaign https://orcid.org/0000-0002-1736-2978
Mohammed El-Kebir (melkebir@illinois.edu)
https://orcid.org/0000-0002-1468-2407

Article

Keywords: SARS-CoV-2, COVID-19, genomics, JUMPER

Posted Date: June 21st, 2021

DOI: https://doi.org/10.21203/rs.3.rs-600334/v1

Version of Record: A version of this preprint was published at Nature Communications on November 18th, 2021. See the published version at https://doi.org/10.1038/s41467-021-26944-y.

Jumper Enables Discontinuous Transcript Assembly in Coronaviruses

Palash Sashittal[1], Chuanyi Zhang[2], Jian Peng[1,3], and Mohammed El-Kebir[1,*]

[1]Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL 61801
[2]Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801
[3]College of Medicine, University of Illinois at Urbana-Champaign, IL, 61801
[*]Correspondence: melkebir@illinois.edu

Abstract

Genes in SARS-CoV-2 and other viruses in the order of *Nidovirales* are expressed by a process of discontinuous transcription mediated by the viral RNA-dependent RNA polymerase. This process is distinct from alternative splicing in eukaryotes and produces subgenomic RNAs that express different viral genes. Here, we introduce the DISCONTINUOUS TRANSCRIPT ASSEMBLY problem of finding transcripts $\mathcal{T}$ and their abundances $\mathbf{c}$ given an alignment $\mathcal{R}$ of paired end short reads under a maximum likelihood model that accounts for varying transcript lengths. Underpinning our approach is the concept of a segment graph, a directed acyclic graph that, distinct from the splice graph used to characterize alternative splicing, has a unique Hamiltonian path. We provide a compact characterization of solutions as subsets of non-overlapping edges in this graph, enabling the formulation of an efficient progressive heuristic that uses mixed integer linear program. We show using simulations that our method, JUMPER, drastically 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 well supported by direct evidence from long-read data, presence in multiple, independent samples or a conserved core sequence. 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.

Code availability: Software is available at `https://github.com/elkebir-group/Jumper`

1 Background

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[1]. Since the host ribosome processes mRNA starting at the 5' end, translation of the viral genome only generates the non-structural proteins. Expression of

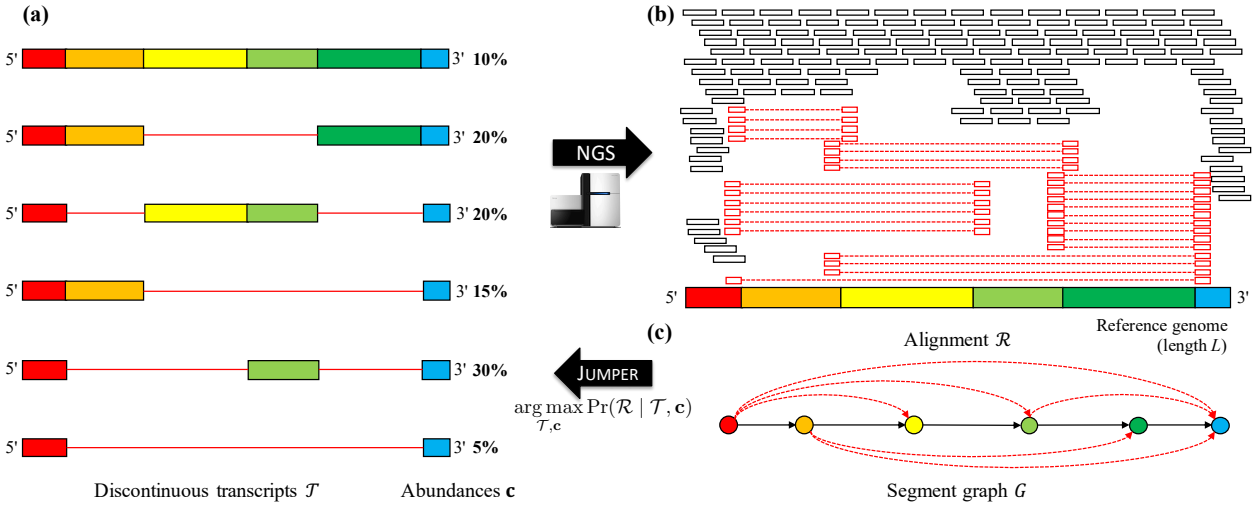

Fig. 1: (a) Coronaviruses generate a set $\mathcal{T}$ of discontinuous transcripts with varying abundances $\mathbf{c}$ during infection. (b) Next generation sequencing will produce an alignment $\mathcal{R}$ with two types of aligned reads: unphased reads that map to a contiguous genomic region (black) and phased 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.

the remaining genes is achieved by *discontinuous transcription* performed by the viral RNA-dependent RNA polymerase (RdRp)[2], 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 (Figure 1a). Several recent studies have analyzed SARS-CoV-2 sequencing samples, identifying 'split reads' — *i.e.* single reads that span non-contiguous parts of the viral geneome — that provide evidence for canonical discontinuous transcription events that produce an intact 3' open reading frame (ORF) as well as non-canonical discontinuous transcription events whose role is unclear[3–5]. 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[6–10] 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[11–15] generally achieve higher accuracy as they use the reference genome as a scaffold on which to align sequencing reads. Specifically, given an alignment $\mathcal{R}$, reference-based methods seek the set $\mathcal{T}$ of transcripts that comprise the transcriptome, enabling the subsequent quantification of their abundances $\mathbf{c}$ using separate tools[16,17].

While in this work we similarly seek to reconstruct transcripts $\mathcal{T}$ and their abundances $\mathbf{c}$ from an alignment $\mathcal{R}$ of coronavirus sequencing samples, there are critical differences between the processes of transcription in eukaryotes and coronaviruses. In eukaryotes, a gene may express multiple transcripts that differ in their

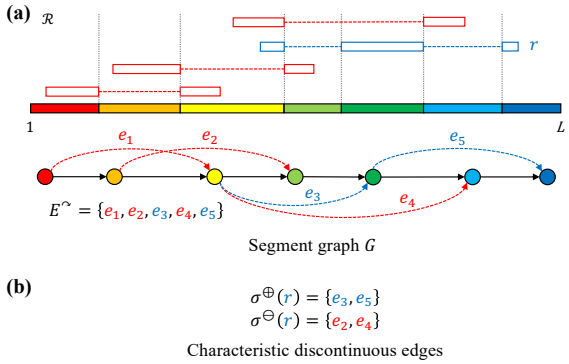

Fig. 2: (a) Phasing reads in an alignment $\mathcal{R}$ define a set of junctions, which in turn define the segment graph G. (b) Each phasing read has characteristic discontinuous edges indicating the set $\sigma^{\oplus}$ of discontinuous edges present in the read as well as conflicting/overlapping discontinuous edges $\sigma^{\ominus}$. Here, phasing 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 $\pi(r) = \{e_3, e_5\}$.

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 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. While conceptually the resulting discontinuous transcripts can be viewed as the result of alternative splicing of a single gene that corresponds to the complete viral genome, there are four key differences and constraints (Figure 1a). First, the genomes of coronaviruses are much smaller ($\sim$ 30kb) than eukaryotic genomes. Second, while alternative splicing sometimes involves shuffling of exons, this phenomenon is not observed in discontinuous transcription where the order of segments is fixed. Third, discontinuous transcripts have matching segments at the 5' and 3' ends, which is not necessarily true for eukaryotic transcripts of the same gene. Fourth, the complete viral genome, without any jumps, is always part of the transcriptome. Current transcript assembly methods are not optimized to leverage these four constraints that characterize coronavirus transcriptomes.

In this study, we introduce the DISCONTINUOUS TRANSCRIPT ASSEMBLY (DTA) problem of finding discontinuous transcripts $\mathcal{T}$ and their abundances $\mathbf{c}$ (Figure 1a) given an alignment $\mathcal{R}$ of paired end reads (Figure 1b). Underpinning our approach is the concept of a segment graph (Figure 1c), a directed acyclic graph that, distinct from the splice graph used to characterize alternative splicing, has a unique 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 drastically outperforms SCALLOP[11] and STRINGTIE[12], existing methods for reference-based transcript assembly in the general case. In real data[3], 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 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[18]. 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[19]. In summary, JUMPER enables detailed analyses of coronavirus transcriptomes under varying conditions.

2 Results

2.1 Discontinuous Transcription Problem

To formulate the DISCONTINUOUS TRANSCRIPT ASSEMBLY problem, we begin by defining discontinuous transcripts as follows.

Definition 1. *Given a reference genome, a* discontinuous transcript T *is a sequence* $\mathbf{v}_1, \ldots, \mathbf{v}_{|T|}$ *of segments where (i) each segment corresponds to a contiguous region in the reference genome, (ii) segment* $\mathbf{v}_i$ *precedes segment* $\mathbf{v}_{i+1}$ *in the reference genome for all* $i \in \{1, \ldots, |T| - 1\}$*, (iii) segment* $\mathbf{v}_1$ *contains the 5' end of the reference genome and (iv) segment* $\mathbf{v}_{|T|}$ *contains the 3' end of the reference genome.*

In the literature, discontinuous transcripts that differ from the genomic transcript T_0 are called *subgenomic transcripts*, which correspond to subgenomic RNAs (sgRNAs)[3]. 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 $\sum_{i=1}^{|\mathcal{T}|} c_i = 1$. While next-generation sequencing technologies provide high coverage of the viral genome of length L of about 10 to 30 Kbp, they are limited to short reads with fixed length ℓ ranging from 100 to 400 bp. For ease of exposition, we describe the formulation in context of single-end reads, but in practice we use the paired-end information if it is available.

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 ℓ, we use a splice-aware aligner such as `STAR`[20] 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 $\mathcal{T}$ of the transcripts is embedded in *phasing reads*, which are reads that align to multiple distinct regions in the reference genome (Figure 1b).

To make the relationship between $\mathcal{T}$, $\mathbf{c}$ and $\mathcal{R}$ clear, we introduce the *segment graph* G, which is obtained from the phasing reads in a alignment $\mathcal{R}$. As mentioned, each phasing 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 phasing read are separated by two positions v, w (where $w - v \geq 2$) in the reference genome called *junctions*. Thus, each phasing read contributes $2q - 2$ junctions. The collective set of junctions contributed by all phasing 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 (Figure 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 phasing read in $\mathcal{R}$ but not adjacent in the reference genome (*i.e.* $w^- - v^+ \geq 2$). Figure 1c shows an example of a segment graph.

Definition 2. *Given an alignment $\mathcal{R}$, 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 phasing 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. $w^- - v^+ \geq 2$ and there exists a phasing read where junctions v^+ and w^- are adjacent.*

We note that the segment graph G is closely related to the *splice graph* used in regular transcript assembly where transcripts correspond to varying sequences of exons due to alternative splicing. The key difference, however, is that an alignment $\mathcal{R}$ generated from reads obtained from discontinuous transcripts induces a segment graph G that is a directed acylic graph (DAG) with a unique Hamiltonian path. This is because, as stated in Definition 1, discontinuous transcripts $\mathcal{T}$ have matching 5' and 3' ends, and, although their comprising segments may vary, their order follows the reference genome.

Observation 1. *Segment graph G is a directed acyclic graph with a unique Hamiltonian path.*

The unique Hamiltonian path of G corresponds to the sequence of continuous edges $E^{\rightarrow}$. This path corresponds to the whole viral genome which is generated by the RdRp during the replication step[2]. Moreover, 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* π is a subset of edges E that can be ordered $(\mathbf{v}_1, \mathbf{w}_1), \ldots, (\mathbf{v}_{|\pi|}, \mathbf{w}_{|\pi|})$ such that $\mathbf{w}_i = \mathbf{v}_{i+1}$ for all $i \in [|\pi| - 1] = \{1, \ldots, |\pi| - 1\}$. While splice graphs are DAGs and typically have a unique source and sink node as well, they do not necessarily contain a Hamiltonian path[11,21–23].

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

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

Under an uninformative, flat prior, this is equivalent to maximizing the probability $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$. We use the segment graph G to compute the probability $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ of observing an alignment $\mathcal{R}$ given transcripts $\mathcal{T}$ and abundances $\mathbf{c}$. We follow the generative model which has been extensively used for transcription quantification[16,17,24]. The notations used in this paper best resemble the formulation described in[23]. Let $\mathcal{R}$ be composed of reads be $\{r_1, \ldots, r_n\}$ and the set $\mathcal{T}$ of transcripts be $\mathcal{T} = \{T_1, \ldots, T_k\}$ with lengths $L_1, \ldots, L_k$ and abundances $\mathbf{c} = [c_1, \ldots, c_k]$. 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 ℓ is much smaller than the length L_i of any transcript T_i. As such, we that $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ equals

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

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

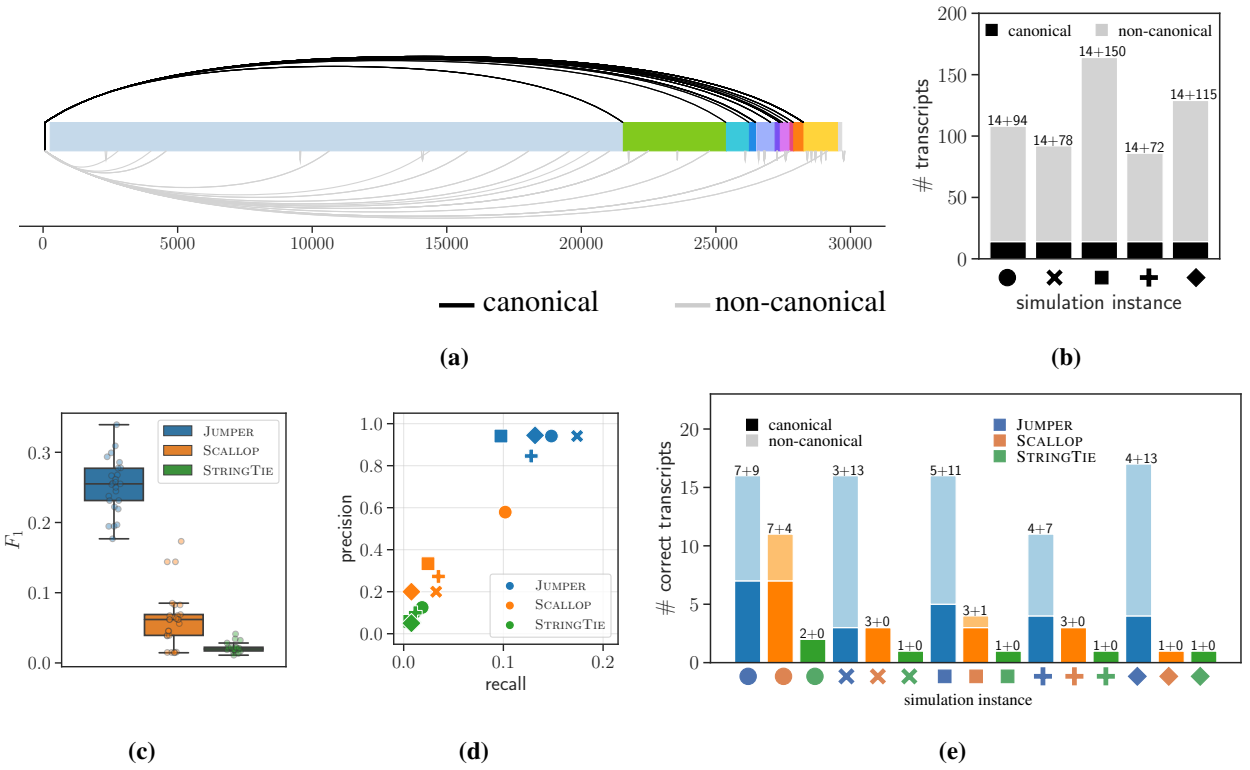

Fig. 3: JUMPER consistently outperforms SCALLOP[11] and STRINGTIE[12] in reconstruction of viral transcripts from simulated SARS-CoV-2 sequencing data. (a) Sashimi plot showing the canonical (black) and non-canonical (gray) discontinuous mappings supported by reads in short-read sample SRR11409417. (b) Number of canonical and non-canonical transcripts for 5 simulation instances of $(\mathcal{T}, \mathbf{c})$ generated under the negative-sense discontinuous transcription model. (c) F_1 score of the three methods (JUMPER, SCALLOP and STRINGTIE) for all the 25 simulated instances under the negative-sense discontinuous transcription model. (d) Precision and recall values of the three methods with one of sequencing experiment for each simulated instance of $(\mathcal{T}, \mathbf{c})$ 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).

necessary condition for transcript T to be a candidate transcript of origin of read r. Supplementary Section A gives the derivation of the above equation (Eq. (1)). Our goal is to find $\text{argmax}_{\mathcal{T},\mathbf{c}} \Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$, leading to the following problem.

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

The probability $P(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$, 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 $P(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ 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`[25] (version 9.0.3)

to solve the MILP and `pysam`[26] for reading and processing the input BAM file. JUMPER is available at `https://github.com/elkebir-group/Jumper`.

2.2 Experimental Evaluation

We begin by establishing terminology that will be used throughout this 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[1], and the first occurrence of 'AUG' downstream of the 3' junction w^- position coincides with the start codon of a known ORF, otherwise the discontinuous edge is called *non-canonical*. 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.

2.2.1 Simulations

We generated our simulation instances using a segment graph G obtained from a short-read sample (SRR11409417). Following Kim *et al.*[3], 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 Section C.1). Upon generating the transcripts, we simulated the generation and sequencing of RNA-seq data, and aligned the simulated reads using `STAR`[20]. We generated 5 independent pairs $(\mathcal{T}, \mathbf{c})$ of transcripts and abundances (Figure 3b). For each pair $(\mathcal{T}, \mathbf{c})$ we generated 5 paired-end short read sequencing simulations using `polyester`[28]. 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 junctions positions within a tolerance of 10 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 0.255 and range

[1]This range contains the TRS-L regions of the SARS-CoV-1[27], SARS-CoV-2[3] and MERS-CoV[27] genomes analyzed in this paper.

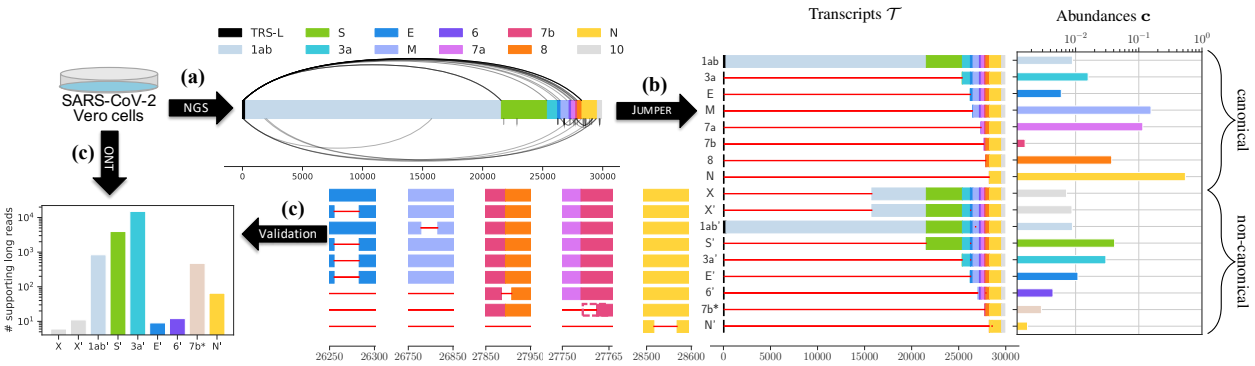

Fig. 4: Using short-read data of SARS-CoV-2 infected Vero cells[3], JUMPER identifies canonical and non-canonical transcripts that are well supported by long-read sequences of the same sample. (a) The segment graph for the short-read data contains both canonical (above) and non-canonical (below) edges. (b) JUMPER assembles 8 canonical transcripts and 9 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)

$[0.176, 0.339]$) 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 Figure C4 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 Λ on the performance of JUMPER, we ran our method on the simulated instances with $\Lambda \in \{10, 50, 100, 200\}$. Supplementary Figure C5 shows that JUMPER outperforms SCALLOP and STRINGTIE for all values of Λ, 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 $(\mathcal{T}, \mathbf{c})$. 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 Figure C6 shows that all three methods produce similar precision and recall values for different sequencing replicates of the same simulated instance of $(\mathcal{T}, \mathbf{c})$, demonstrating consistency in results. Finally, 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.

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 C2). We observe similar trends on simulated instances of a human gene (see Supplementary Section C.4).

2.2.2 Viral transcript assembly in SARS-CoV-2 infected Vero cells

Recently, Kim *et al.*[3] 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[29,30]. Subsequently, the authors aligned the resulting reads using splice-aware aligners, `STAR`[20] for the short-read sample (median depth of 1763) and `minimap2`[31] for the long-read sample (median depth of 6707 and mean length of 2875 bp). For both complementary sequencing techniques, the authors observed phasing reads that were indicative of canonical as well as non-canonical transcription events. While this previous work quantified the fraction of phasing 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[3]. 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 $|E^{\curvearrowright}| = 35$ most abundant discontinuous edges, 18 of which canonical and 17 non-canonical (Figure 4a), JUMPER identified 33 transcripts, 17 of which have an abundance of at least 0.001 as determined by SALMON (Figure 4b). A subset of 8 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[3,5] that did not find evidence for active transcription of ORF10.

As mentioned, JUMPER inferred 9 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 "GAACTTTAA" 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 Figure C7a,b). Strikingly, both matching sequences are conserved within the *Sarbecovirus* subgenus but not in other subgenera of the *Betacoronavirus* genus (Supplementary Figure C7a,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 Figure C8). 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 Figure C9). We note that although this discontinuous transcription event was also observed in[5], the authors found no evidence of this transcript leading to 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 ~ 27 nucleotides between positions 26256 and 26284. This is an in frame deletion inside ORF E, resulting in the loss of 9 amino acids that span the N-terminal domain (4 amino acids) and the transmembrane domain (5 amino acids) of the E protein[32]. A similar in-frame deletion of 24 nucleotides (from position 26259 to 26284) was observed by Finkel *et al.*[5] that resulted in the loss of a subset of 8 out of the 9 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.*[3] and Finkel *et al.*[5]. 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 9 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[3] 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[33,34] 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[3] 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 4 out of the 9 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

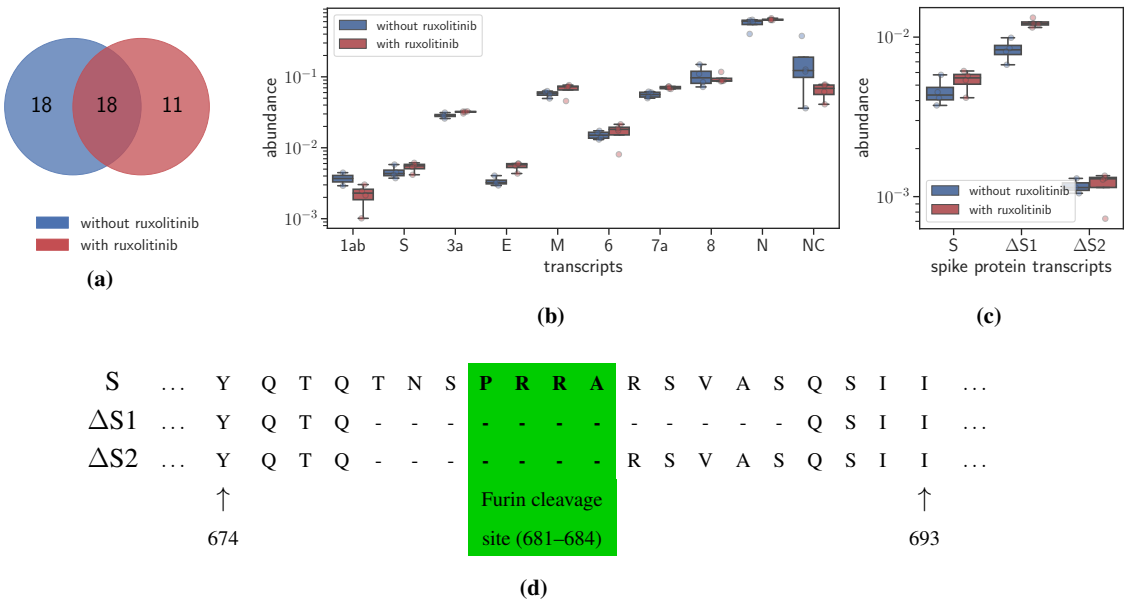

Fig. 5: JUMPER enables analysis of drug response in SARS-CoV-2 infected cells[19] at the transcript level. (a) A Venn diagram shows in the number of transcripts reconstructed from samples with and without treatment with ruxolitinib. Supplementary Figure C11 shows the distribution of the 18 transcripts that are common between samples with and without treatment while Supplementary Table C3 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 ΔS1 and ΔS2 whose structure is described in (d).

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 (Figure 4c). Moreover, we found support for the canonical transcripts as well (Supplementary Figure C10). 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 and their abundances, whose presence we subsequently validated on a long-read sequencing sample of the same cells.

2.2.3 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.[19] 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 hours of SARS-CoV-2 infection. There are a total of eight samples, four of which were pre-treated with ruxolitinib for 1 hour before the infection and the remaining four were untreated. Ruxolitinib is a JAK1 and 2 kinase inhibitor, which blocks type-I interferon (IFN-I) signaling necessary to engage cellular antiviral defenses[35,36]. 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[17].

SCALLOP, run with default parameters (Supplementary Section C2), 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 Figure C11a,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, Figure 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, Figure C11d in Appendix 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 ∇M, 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 C3, potentially mediating the jump made by the RdRp to generate these transcripts. Specifically, while transcript ∇M 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, ΔS1 and ΔS2, have in-frame deletions in the region that encodes for the spike protein.

ΔS1 contains an in-frame jump from position 23593 to 23630 resulting in a 12 amino-acid in-frame deletion, while ΔS2 contains a jump from position 23593 to 23615, which results in a 7 amino-acid in-frame

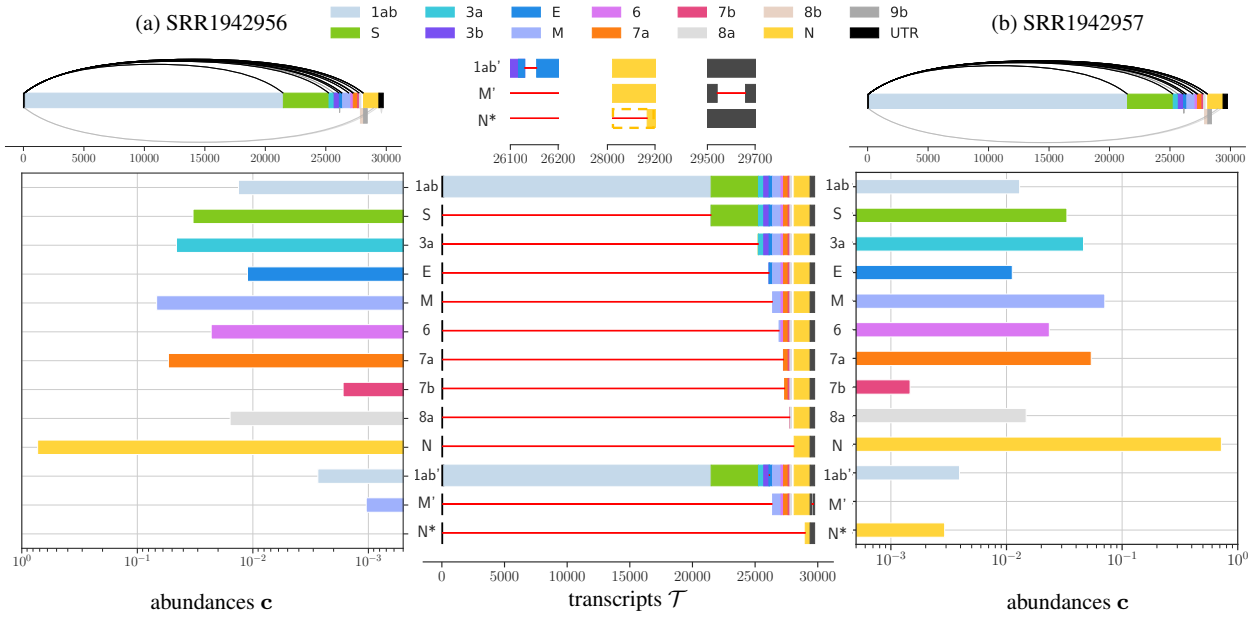

Fig. 6: JUMPER identifies canonical and non-canonical transcripts that recur in two short-read sequencing samples of SARS-CoV-1 infected Calu-3 cells[18]. For both the samples, we show the segment graph, with canonical (above) and non-canonical (below) discontinuous edges. We also show the predicted 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.

deletion in the spike protein (Figure 5d). Both these deletions overlap with the furin cleavage site (FCS), highlighted in Figure 5d, which has been the focus of several recent studies[4,37,38]. The authors of[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 ΔS1 and ΔS2 in infected A549 cell samples can be explained by the fact that Blanco et al.[19] 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), ΔS1 and ΔS2 (median increase from 0.0011 to 0.0012), with the increase being most significant for ΔS1 (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 variants 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.

2.2.4 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 Section C.5.

We analyzed two published samples of human Calu-3 cells infected with SARS-CoV-1[18], 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 hours 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, SCALLOP 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 (Figure 6).

SARS-CoV-1 has a genome of length 29751 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 (Figure 6). Notably, ORF8b and ORF9b share transcription regulating body sequences (TRS-B) with ORF8a and ORF N respectively[39]. 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[39]. 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[5]. 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 (Figure 6) and ranked in the same order (Supplementary Figure C12), 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 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 contain part of the C-terminal dimerization domain[33] 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 used to to reconstruct the transcriptome of all viruses in 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 Section C.5).

3 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[3]. 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[3,5]. Moreover, we predicted 9 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 8 samples of A549 lung alveolar cells infected by SARS-CoV-2, four of which were pre-treated with ruxolitinib for 1 hour before infection[19]. 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[18], 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. 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[40–45]. 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[23]. Finally, the approach presented in this paper can extended to the general transcript assembly problem. Although JUMPER can be used for transcript assembly of individual eukaryotic genes (see Supplementary Section 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.

4 Methods

4.1 Combinatorial Characterization of Solutions

Eq. (1) defines the probability $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ in terms of the observed reads r and their induced paths $\pi(r) \subseteq E(G)$ in the segment graph G. The authors in[23] 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* G 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 $\mathbf{s} - \mathbf{t}$ 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 $\mathbf{s}-\mathbf{t}$ path $\pi(T)$ of G the corresponding set of discontinuous edges is given by $\sigma(T) = \pi(T) \cap E^{\curvearrowright}$. Thus, we have the following proposition with the proof in Supplementary Section B.1.

Proposition 1. *There is a bijection between subsets of discontinuous edges that are pairwise non-overlapping and* $\mathbf{s} - \mathbf{t}$ *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.* $\sigma^{\oplus}(r) = \pi(r) \cap E^{\curvearrowright}$. 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 $E^{\curvearrowright} \setminus \sigma^{\oplus}$ that overlap with any 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 $E^{\curvearrowright}$ (see Figure 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. $\sigma^{\oplus}(r) = \pi(r) \cap E^{\curvearrowright}$, and $\sigma_i^{\ominus}$ is the set of discontinuous edges $(\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+]) \in E^{\curvearrowright} \setminus \sigma^{\oplus}(r)$ 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 Section 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 $\Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ as

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

where $X(\mathcal{T}, \sigma_j^{\oplus}, \sigma_j^{\ominus})$ be the subset of indices i corresponding to transcripts $T_i \in \mathcal{T}$ 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 splice graph wheres 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 $\mathcal{T}$ and abundances $\mathbf{c}$, Equations* (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.

4.2 JUMPER: a Heuristic for the DTA Problem

To solve the DTA problem, we use the results of the section on Combinatorial Characterization of Solutions to write a more concise form of the likelihood. Specifically, let $\mathcal{S} = \{(\sigma_1^{\oplus}, \sigma_1^{\ominus}), \ldots, (\sigma_m^{\oplus}, \sigma_m^{\ominus})\}$ be the set of characteristic discontinuous edges generated by the reads in alignment $\mathcal{R}$. Let $\mathbf{d} = \{d_1, \cdots, d_m\}$ be the number of reads that map to each pair in $\mathcal{S}$. Using that reads r with identical characteristic discontinuous edges $(\sigma^{\oplus}(r), \sigma^{\ominus}(r))$ have identical probabilities $\Pr(r \mid \mathcal{T}, \mathbf{c})$, we obtain the following mathematical program for the log-likelihood $\log \Pr(\mathcal{R} \mid \mathcal{T}, \mathbf{c})$ (see Supplementary Section A for derivation).

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

$$\text{s.t. } \pi(T_i) \text{ is an } \mathbf{s}-\mathbf{t} \text{ path} \tag{4}$$

$$\text{in the segment graph } G, \forall i \in [k],$$

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

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

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 Section 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, $\overline{c}_i(\mathbf{c}) = c_i D / \sum_{j=1}^{k} c_j L_j$ and $c_i(\overline{\mathbf{c}}) = \overline{c}_i / \sum_{j=1}^{k} \overline{c}_j$ for all $i \in [k]$. Then, $(\mathcal{T}, \mathbf{c} = [c_1(\overline{\mathbf{c}}), \ldots, c_k(\overline{\mathbf{c}})])$ is an optimal solution for* (3)-(6) *if and only if $(\mathcal{T}, \overline{\mathbf{c}} = [\overline{c}_1(\mathbf{c}), \ldots, \overline{c}_k(\mathbf{c})])$ is an optimal solution for*

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

$$\textit{s.t. } \pi(T_i) \textit{ is an } \mathbf{s}-\mathbf{t} \textit{ path} \tag{8}$$

$$\textit{in the segment graph } G, \forall i \in [k],$$

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

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

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 $\sum_{i \in X(\mathcal{T},\sigma_j^{\oplus},\sigma_j^{\ominus})} c_i$ 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 Section B.2 for (iii) and (iv).

Transcript composition. We begin modeling (8), which states that each transcript T_i must correspond to an $\mathbf{s}-\mathbf{t}$ path in the segment graph G. Using Proposition 1, we introduce binary variables $\mathbf{x} \in \{0,1\}^{|E^{\curvearrowright}| \times k}$ to encode the presence of discontinuous edges in each of the k $\mathbf{s}-\mathbf{t}$ paths corresponding to the k transcripts in $\mathcal{T}$. For any discontinuous edge $e = (\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$, let $I(e)$ denote the open interval (v^+, w^-) between the two segments $\mathbf{v}$ and $\mathbf{w}$. 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

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

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

Transcript abundance and length. We introduce non-negative continuous variables $\mathbf{c} = [c_1, \ldots, c_k]$ that encode the abundance of the k transcripts. The scale of these abundances depends on the choice of D. We choose $D = \ell^*$ where ℓ^* is the length of the shortest $\mathbf{s} - \mathbf{t}$ 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 $e = (\mathbf{v} = [v^-, v^+], \mathbf{w} = [w^-, w^+])$, let $L(e) = w^- - v^+$ be the length of the interval. Observe that

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

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

$$\begin{aligned} z_{e,i} &\leq c_i, \quad \forall i \in [k], \\ z_{e,i} &\leq x_{e,i}, \quad \forall e \in E^{\curvearrowright}, i \in [k], \\ z_{e,i} &\geq c_i + x_{e,i} - 1, \quad \forall e \in E^{\curvearrowright}, i \in [k]. \end{aligned}$$

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

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

The resulting formulation has $O(|E^{\curvearrowright}|k + |E^{\curvearrowright}|m + mh)$ variables, where h is the user-specified number of breakpoints used in the piecewise linear approximation of the log function. This number includes $|E^{\curvearrowright}|k$ 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 $k \cdot |E^{\curvearrowright}|$ 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 Section 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 $|E^{\curvearrowright}|k$ to $|E^{\curvearrowright}|$ 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 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 Section B.3 for details) with function SOLVELP to re-estimate the abundances $\mathbf{c}'$ of

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

$(\mathcal{T}, \mathbf{c}) \leftarrow (\emptyset, [])$
for $p \leftarrow 1$ **to** k **do**
    $T' \leftarrow$ SOLVEILP($\mathcal{T}$)
    $\mathcal{T}' \leftarrow \mathcal{T} \cup$ EXPAND(T')
    $\mathbf{c}' \leftarrow$ SOLVELP($\mathcal{T}'$)
    Sort $(\mathcal{T}', \mathbf{c}')$ s.t. $L_i c'_i \geq L_{i+1} c'_{i+1}$ for all $i \in \{1, \ldots, |\mathcal{T}'| - 1\}$
    $(\mathcal{T}', \mathbf{c}') \leftarrow (\{T_1, \ldots, T_p\}, [c'_1, \ldots, c'_p])$
    if $\mathcal{T}' \neq \mathcal{T}$ **then**
        $(\mathcal{T}, \mathbf{c}) \leftarrow (\mathcal{T}', \mathbf{c}')$
    else
        return $(\mathcal{T}, \mathbf{c})$
return $(\mathcal{T}, \mathbf{c})$

$\mathcal{T}'$, 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. 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 Section B.3.

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 Λ is provided in Supplementary Section B.4.

References

[1] De Vries, A. A., Horzinek, M. C., Rottier, P. J. & De Groot, R. J. The genome organization of the Nidovirales: similarities and differences between Arteri-, Toro-, and Coronaviruses. In *Seminars in VIROLOGY*, vol. 8, 33–47 (Elsevier, 1997).

[2] Maier, H. J., Bickerton, E., Britton, P. *et al. Coronaviruses: methods and protocols.* (Springer Berlin, 2015).

[3] Kim, D. *et al.* The architecture of SARS-CoV-2 transcriptome. *Cell* (2020).

[4] 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 medicine* **12**, 1–15 (2020).

[5] Finkel, Y. *et al.* The coding capacity of SARS-CoV-2. *Nature* **589**, 125–130 (2021).

[6] Robertson, G. *et al.* De novo assembly and analysis of RNA-seq data. *Nature Methods* **7**, 909–912 (2010).

[7] Grabherr, M. G. *et al.* Trinity: reconstructing a full-length transcriptome without a genome from RNA-Seq data. *Nature Biotechnology* **29**, 644 (2011).

[8] Xie, Y. *et al.* SOAPdenovo-Trans: de novo transcriptome assembly with short RNA-Seq reads. *Bioinformatics* **30**, 1660–1666 (2014).

[9] Chang, Z. *et al.* Bridger: a new framework for de novo transcriptome assembly using RNA-seq data. *Genome Biology* **16**, 30 (2015).

[10] 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).

[11] Shao, M. & Kingsford, C. Accurate assembly of transcripts through phase-preserving graph decomposition. *Nature Biotechnology* **35**, 1167–1169 (2017).

[12] Pertea, M. *et al.* StringTie enables improved reconstruction of a transcriptome from RNA-seq reads. *Nature Biotechnology* **33**, 290–295 (2015).

[13] Liu, J., Yu, T., Jiang, T. & Li, G. TransComb: genome-guided transcriptome assembly via combing junctions in splicing graphs. *Genome Biology* **17**, 213 (2016).

[14] Trapnell, C. *et al.* Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature Biotechnology* **28**, 511–515 (2010).

[15] Song, L. & Florea, L. CLASS: constrained transcript assembly of RNA-seq reads. In *BMC Bioinformatics*, vol. 14, S14 (Springer, 2013).

[16] Bray, N. L., Pimentel, H., Melsted, P. & Pachter, L. Near-optimal probabilistic RNA-seq quantification. *Nature Biotechnology* **34**, 525–527 (2016).

[17] Patro, R., Duggal, G., Love, M. I., Irizarry, R. A. & Kingsford, C. Salmon provides fast and bias-aware quantification of transcript expression. *Nature methods* **14**, 417–419 (2017).

[18] Zhang, X. *et al.* Competing endogenous RNA network profiling reveals novel host dependency factors required for MERS-CoV propagation. *Emerging microbes & infections* **9**, 733–746 (2020).

[19] Blanco-Melo, D. *et al.* Imbalanced host response to SARS-CoV-2 drives development of COVID-19. *Cell* (2020).

[20] Dobin, A. *et al.* STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**, 15–21 (2013).

[21] 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).

[22] 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 bioinformatics* **16**, 1–10 (2015).

[23] Zheng, H., Ma, C. & Kingsford, C. Deriving ranges of optimal estimated transcript expression due to non-identifiability. *bioRxiv* 2019–12 (2021).

[24] Li, B. & Dewey, C. N. RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. *BMC bioinformatics* **12**, 323 (2011).

[25] Gurobi Optimization, L. Gurobi optimizer reference manual (2020). URL `http://www.gurobi.com`.

[26] Li, H. *et al.* The sequence alignment/map format and SAMtools. *Bioinformatics* **25**, 2078–2079 (2009).

[27] Yang, D. & Leibowitz, J. L. The structure and functions of coronavirus genomic 3' and 5' ends. *Virus research* **206**, 120–133 (2015).

[28] 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).

[29] Gohl, D. M. *et al.* A rapid, cost-effective tailed amplicon method for sequencing SARS-CoV-2. *BMC genomics* **21**, 1–10 (2020).

[30] Quick, J. nCoV-2019 sequencing protocol v3 (LoCost). *protocols.io* (2020). Https://protocols.io/view/ncov-2019-sequencing-protocol-v3-locost-bh42j8ye.

[31] Li, H. Minimap2: pairwise alignment for nucleotide sequences. *Bioinformatics* **34**, 3094–3100 (2018).

[32] Mandala, V. S. *et al.* Structure and drug binding of the SARS-CoV-2 envelope protein transmembrane domain in lipid bilayers. *Nature Structural & Molecular Biology* **27**, 1202–1208 (2020). URL `http://www.nature.com/articles/s41594-020-00536-8`.

[33] Kang, S. *et al.* Crystal structure of SARS-CoV-2 nucleocapsid protein RNA binding domain reveals potential unique drug targeting sites. *Acta Pharmaceutica Sinica B* **10**, 1228–1238 (2020).

[34] Ye, Q., West, A. M., Silletti, S. & Corbett, K. D. Architecture and self-assembly of the SARS-CoV-2 nucleocapsid protein. *Protein Science* **29**, 1890–1901 (2020).

[35] Murira, A. & Lamarre, A. Type-I interferon responses: from friend to foe in the battle against chronic viral infection. *Frontiers in immunology* **7**, 609 (2016).

[36] Lee, J. S. & Shin, E.-C. The type I interferon response in COVID-19: implications for treatment. *Nature Reviews Immunology* **20**, 585–586 (2020).

[37] 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 transduction and targeted therapy* **5**, 1–3 (2020).

[38] Johnson, B. A. *et al.* Loss of furin cleavage site attenuates SARS-CoV-2 pathogenesis. *Nature* 1–10 (2021).

[39] Yang, Y., Yan, W., Hall, A. B. & Jiang, X. Characterizing Transcriptional Regulatory Sequences in Coronaviruses and Their Role in Recombination. *Molecular Biology and Evolution* (2020). URL `https://doi.org/10.1093/molbev/msaa281`. Msaa281.

[40] Sashittal, P., Luo, Y., Peng, J. & El-Kebir, M. Characterization of SARS-CoV-2 viral diversity within and across hosts. *bioRxiv* (2020).

[41] Rose, R. *et al.* Intra-host site-specific polymorphisms of SARS-CoV-2 is consistent across multiple samples and methodologies. *medRxiv* (2020).

[42] Ramazzotti, D. *et al.* Characterization of intra-host SARS-CoV-2 variants improves phylogenomic reconstruction and may reveal functionally convergent mutations. *bioRxiv* (2020).

[43] Shen, Z. *et al.* Genomic diversity of SARS-CoV-2 in coronavirus disease 2019 patients. *Clinical Infectious Diseases* (2020).

[44] Karamitros, T. *et al.* SARS-CoV-2 exhibits intra-host genomic plasticity and low-frequency polymorphic quasispecies. *bioRxiv* (2020).

[45] Tang, X. *et al.* On the origin and continuing evolution of SARS-CoV-2. *National Science Review* (2020).

Supplementary Files

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

- JumperNatCommsupp.pdf