Supplementary Material for “Accurate Identification of Transcription Regulatory Sequences and Genes in Coronaviruses”

Chuanyi Zhang1,*, Palash Sashittal2,3,*, Michael Xiang2, Yichi Zhang2, Ayesha Kazi2, and Mohammed El-Kebir2,†

1Department of Electrical & Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL
2Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, IL
3Present address: Department of Computer Science, Princeton University, Princeton, NJ
*Joint first authorship
†Correspondence: melkebir@illinois.edu

Contents

1 Supplementary Methods

1.1 Obtaining candidate regions for the TRS-ID problem
1.2 Constrained TRS-ID problem
1.3 Partitioning the genome into \( v_{\text{leader}} \) and \( v_{\text{body}} \) for the TRS-GENE-ID problem
1.4 Constrained TRS-GENE-ID problem
1.5 Practical considerations to solve the TRS-GENE-ID problem
1.6 Classifier to identify 5' truncated genomes with missing TRS-L

2 Supplementary Results

2.1 Evaluating TRS site identification methods
2.2 Evaluating gene finding methods
2.3 Tuning parameters for CORSID
2.4 Command-line arguments
2.5 Two anomalous coronavirus genomes
2.6 Non-AUG start codons
2.7 Scope and recommendations on using CORSID and CORSID-A

3 Supplementary Figures and Tables
1 Supplementary Methods

1.1 Obtaining candidate regions for the TRS-ID problem

Intuitively, a candidate region for gene \( x_i \) corresponds to the region \( w_i = [w_i^-, w_i^+] \) composed of positions \( w^- \leq p \leq w^+ \) such that any sgRNA starting at \( p \) will lead to the translation of ORF \( x_i \) by the ribosome. Note that the first ORF \( x_0 \) corresponds to \( ORF1ab \). Accordingly, we restrict \( w_0 = [w_0^-, w_0^+] \) to match exactly the leader region, spanning the start of the genome at the 5' end until the start codon of \( x_0 \), i.e. \( w_0^- = 1 \) and \( w_0^+ = x_0^- - 1 \). To define the remaining candidate regions \( w_1, \ldots, w_n \), we must take ribosomal leaky scanning into account, where the ribosome does not initiate translation at the first ‘ATG’ it encounters [11].

To model this, we make use of the fact that almost all coronavirus genes (3627 out of 3637; Fig. S15) have a length of at least 100 nucleotides. Specifically, when determining the candidate region of a gene, we skip over a previous ORF in case its length is less than 100. To that end, we introduce the following function.

**Definition 1.** Function \( \text{prev}(p) \) returns the first ORF \( x = [x^-, x^+] \) upstream of position \( p \) in the genome, i.e. for ORF \( x \) returned by \( \text{prev}(p) \) it holds that \( x^- < p \) and there exists no ORF \( y = [y^-, y^+] \) such that \( x^- < y^- < p \). If no such ORF \( x \) exists then \( \text{prev}(p) = [0, 0] \). Moreover, \( \text{prev}(0) = [0, 0] \).

Using this function, we define a TRS-B candidate region \( w \) of an ORF \( x \) as follows.

**Definition 2.** Let \( x = [x^-, x^+] \) be an ORF and let \( y = [y^-, y^+] = \text{prev}(x^-) \) and \( z = [z^-, z^+] = \text{prev}(y^-) \) be the previous two ORFs. The candidate region \( w = [w^-, w^+] \) of ORF \( x \) ends at a core sequence length \( \omega \) downstream of the start codon of \( x \), i.e. \( w^+ = x^- + \omega - 1 \), allowing the start codon to appear within the core sequence, and begins at the first position of the genome if \( x \) has no previous ORF or the only preceding ORF \( y \) has a length smaller than 100; \( w \) begins at the first ORF \( y \) if the length of \( y \) is at least 100; otherwise \( w \) begins at the second ORF \( z \) if the first ORF \( y \) has a length smaller than 100 nucleotides.

That is,

\[
\begin{align*}
w^- &= \begin{cases} 
1, & \text{if } y = [0, 0], \\
1, & \text{if } y \neq [0, 0], z = [0, 0] \text{ and } |y| < 100, \\
y^- + 3, & \text{if } y \neq [0, 0], |y| \geq 100, \\
z^- + 3, & \text{if } y \neq [0, 0], z \neq [0, 0] \text{ and } |y| < 100.
\end{cases}
\end{align*}
\]  

(1)

Let candidate regions \( w_0, \ldots, w_n \) be sorted by the 5' ends of their corresponding ORFs. To remove overlap among candidate regions \( w_0, \ldots, w_n \), we set \( w_i^- = w_{i-1}^+ + 1 \) if \( w_{i-1}^+ \geq w_i^- \) for all \( i \in \{1, \ldots, n\} \) as depicted in Fig. S16.
1.2 Constrained TRS-ID problem

Here, we introduce the following constrained version of the TRS-ID problem.

**Problem 1** (Constrained TRS Identification (TRS-ID-u)). Given non-overlapping sequences \(w_0, \ldots, w_n\), and a subsequence \(u\) of \(w_0\), find a TRS alignment \(A = [a_0, \ldots, a_n]^T\) such that (i) \(a_i\) corresponds to a subsequence in \(w_i\) for all \(i \in \{0, \ldots, n\}\), (ii) \(u\) is a subsequence of the core sequence \(c(A)\), and (iii) the alignment has maximum score \(s(A)\).

Let us focus on solving a single TRS-ID-u problem instance, where we are given non-overlapping sequences \(w_0, \ldots, w_n\) and a subsequence \(u\) of \(w_0\). Each such instance decomposes into \(n\) TRS-ID-u instances each with exactly two sequences. That is, for any \(i \in \{1, \ldots, n\}\), we seek a TRS alignment \(A_i = [a_0^i, a_i^i]^T\) of sequences \(w_0 = [w_{0,p}]\) and \(w_i = [w_{i,q}]\) such that the induced core sequence \(c(A_i)\) contains \(u = [u^-,u^+]\). This is a variant of local alignment [18] with three key differences: (i) alignment \(A_i\) may not contain gaps, (ii) \(A_i\) must span \(u^-\) and (iii) \(A_i\) must span \(u^+\). Letting \(\ell\) be the relative position of \(u^-\) in \(w_0\), we obtain

\[
s[p,q] = \begin{cases} 
0, & \text{if } p = 0 \text{ or } q = 0, \\
\max \{0, s[p-1,q-1] + \delta(w_{0,p}, w_{i,q})\}, & \text{if } 1 \leq p < \ell \text{ and } q \geq 1, \\
s[p-1,q-1] + \delta(w_{0,p}, w_{i,q}), & \text{if } p \geq \ell \text{ and } q \geq 1,
\end{cases}
\]

(2)

where for \(p < \ell\), \(s[p,q]\) indicates the optimal score of a TRS alignment between \(w_{0,1} \ldots w_{0,p}\) and \(w_{i,1} \ldots w_{i,q}\), and for \(p \geq \ell\), \(s[p,q]\) indicates the optimal score of a constrained TRS alignment between \(w_{0,1} \ldots w_{0,p}\) and \(w_{i,1} \ldots w_{i,q}\) that spans the entire subsequence \(u' = w_{0,\ell} \ldots w_{0,p}\). The desired solution is then taken from a trapezoidal region as follows (see Fig. S14 for an example).

\[
(p^*,q^*) = \arg \max_{\ell+|u|-1 \leq p \leq |w_0|, \ p-\ell+1 \leq q \leq |w_i|} s[p,q].
\]

(3)

Note that the recurrence (2) lacks the two cases of the Smith-Waterman [18] recurrence corresponding to a gap (i.e. \(s[p-1,|q|]\) and \(s[p,|q-1|]\)), thus satisfying constraint (i). Constraints (ii) and (iii) are satisfied because the first case, corresponding to initiating the alignment, is only enabled when \(p < \ell\) thus covering \(u^-\), and by (3), we have that the alignment contains \(u^+\). Similarly to local alignment, we can identify \((p^*,q^*)\) by filling out the table \(s[0,0], \ldots, s[|w_0|, |w_i|]\) using dynamic programming, and reconstruct the TRS alignment \(A_i = [a_0^i, a_i^i]^T\) using a backtrace from \((p^*,q^*)\). Letting \(L = \sum_{i=1}^n |w_i|\) be the total length of candidate regions \(w_1, \ldots, w_n\), solving the \(n\) pairwise TRS-ID-u problems and obtaining the pairwise TRS alignments \(A_1, \ldots, A_n\) that cover \(u\) takes \(O(|w_0|L)\) time.
Given these pairwise alignments $A_1, \ldots, A_n$ that each span $u$, we construct the final TRS alignment $A = [a_0, \ldots, a_n]^\top$ as follows. First, we exclude alignments $A_i$ that have a score less than the threshold $\tau$. Second, the TRS-L sequence $a_0$ equals the subsequence of $w_0$ that spans the positions covered by all pairwise alignments, i.e. $a_0$ spans exactly the positions of $w_0$ covered by $a_0^1, \ldots, a_0^n$. Third, we obtain the remaining gapped sequences $a_1, \ldots, a_n$ of $A$ by adding flanking gaps to each (ungapped) sequence $a_1^1, \ldots, a_n^n$ so as to match the unaligned letters of $a_0$ (main text Fig.5a). As flanking gaps do not incur a penalty, this operation will not change the total score, i.e. $s(A) = \sum_{i=1}^n s(A_i)$. The running time of computing alignments $A_1, \ldots, A_n$ and then subsequently merging them into $A$ is dominated by the first step.

1.3 Partitioning the genome into $v_{\text{leader}}$ and $v_{\text{body}}$ for the TRS-GENE-ID problem

To obtain these two sequences, $v_{\text{leader}}$ and $v_{\text{body}}$, for a given coronavirus genome, we developed a heuristic for identifying $\text{ORF1ab}$, the largest gene in coronavirus genomes. This heuristic begins by enumerating all ORFs $x_1, \ldots, x_m$ in the genome (main text Definition 5). As $\text{ORF1ab}$ is the result of a frameshift upstream of the stop codon of $\text{ORF1a}$, we extend each enumerated ORF $x_i$ by performing either a $-1$ or $-2$ frameshift and subsequently scanning for an in-frame stop codon. Rather than exclusively looking for a $-1$ frameshift, which is common in coronaviruses, we also consider a $-2$ frameshift to account for sequencing errors. We select the frameshift that results in the largest extended ORF, obtaining extended ORFs $y_1, \ldots, y_m$. We designate the largest ORF among this set as $\text{ORF1ab}$. Finally, we set $v_{\text{leader}}$ as the region from the start of the genome until the 5’ coordinate of $\text{ORF1ab}$. As the TRS-B of the first gene downstream of $\text{ORF1ab}$ may reside within $\text{ORF1ab}$, we set $v_{\text{body}}$ as the region starting from 200 nucleotides upstream of the 3’ coordinate of $\text{ORF1ab}$ until the 3’ end of the genome.

1.4 Constrained TRS-GENE-ID problem

Here, we introduce the constrained version of the TRS-GENE-ID problem.

**Problem 2** (CONSTRAINED TRS AND GENE IDENTIFICATION (TRS-GENE-ID-u)). Given leader region $v_{\text{leader}}$, body region $v_{\text{body}}$ and a subsequence $u$ of $v_{\text{leader}}$, find a TRS alignment $A = [a_i]$ such that (i) $a_0$ corresponds to a subsequence in $v_{\text{leader}}$, (ii) $a_i$ corresponds to a subsequence in $v_{\text{body}}$ for all $i \geq 1$, (iii) $u$ is a subsequence of the core sequence $c(A)$, (iv) $A$ is concordant, and (v) $A$ induces the set $\Gamma(A)$ of genes with maximum genome coverage $g(A)$ and subsequently has maximum score $s(A)$.

---

Footnote: Infectious bronchitis virus strain A2 (EU526388) genome has a deletion at position 7083 in genome, requiring a $-2$ frameshift to correctly identifying ORF1ab (530-11484).
We solve this problem in two steps. First, we use dynamic programming to compute \( s[p, q] \) for all values of \( 0 \leq p \leq |v_{\text{leader}}| \) and \( 0 \leq q \leq |v_{\text{body}}| \), i.e. the optimal score \( s[p, q] \) of a TRS alignment between \( v_{\text{leader},1} \ldots v_{\text{leader},p} \) and \( v_{\text{body},1} \ldots v_{\text{body},q} \) constrained to contain \( u \). The quantity \( s[p, q] \) is defined using the same recurrence as for the TRS-ID-\( u \) problem – Eq. (2) – and the complete table can be filled out using dynamic programming in \( O(|v_{\text{leader}}| |v_{\text{body}}|) \) time.

Second, let \( x_1, \ldots, x_m \) be the candidate ORFs in \( v_{\text{body}} \), each with a length of at least 100 nucleotides (Fig. S15). For each ORF \( x_i \) we find the position \( (p, q) \) that encodes the maximum scoring alignment \( A_i = [a_{i0}^-, a_{i0}^+] \top \) where \( a_{i0}^+ \) is associated with \( x_i \). We remove ORFs \( x_i \) whose maximum scoring alignment \( A_i \) has a score \( s(A_i) \) less than the user-specified score threshold \( \tau \). Let \( s^* \) indicate the maximum score among all TRS alignments \( A_1, \ldots, A_m \). Then, we construct a vertex-weighted interval graph \( G = (V, E) \) whose vertices \( V \) correspond to the intervals \( \{[x_1^-, x_1^+], \ldots, [x_m^-, x_m^+]\} \) of the candidate ORFs. There is an edge \((x_i, x_j)\) if and only if the two corresponding intervals overlap, i.e. \( [x_j^-, x_j^+] \cap [x_i^-, x_i^+] \neq \emptyset \). To capture the lexicographical ordering of the objective functions, i.e. first the genome coverage and then the score, each vertex/interval \( x_i \) is assigned weight

\[
w(x_i) = |x_i| + \frac{s(A_i)}{s^*}.
\]

In other words, among ORFs with the same length, we prefer those that have an associated TRS alignment with largest score. Finally, we solve a maximum-weight independent set (MWIS) problem, which can be done in \( O(|V|) \) time for interval graphs [9]. The maximum-weight independent set \( X = \{x_{\pi(1)}, \ldots, x_{\pi(|X|)}\} \) directly corresponds to the induced genes \( \Gamma(A) \) of the TRS alignment \( A \) that can be constructed by merging pairwise TRS alignments \( A_{\pi(1)}, \ldots, A_{\pi(|X|)} \) following the same procedure described in main text section 3.1. For each window \( u \) of fixed length \( \omega \), the first step takes \( O(|v_{\text{leader}}| |v_{\text{body}}|) \) time and the second step takes \( O(|v_{\text{leader}}| |v_{\text{body}}| + m) \) time.

### 1.5 Practical considerations to solve the TRS-GENE-ID problem

In this section we discuss practical considerations for identifying genes in coronaviruses and the way they are addressed in CORSID.

**Overlapping genes.** In practice, coronavirus genes may overlap. That is, the start codon of a gene can be located within another gene. To support such cases, we shrink candidate ORFs \( x = [x^-, x^+] \) by 5% prior to solving the MWIS problem, i.e. the interval graph \( G = (V, E) \) has shortened intervals \( x' = [x^- + \alpha, x^+ - \alpha] \) where \( \alpha = 0.05|\chi| \). The default shrinking percentage 5% was chosen after performing a grid search and cross validation. The details are explained in Section 2.3.
Progressive approach. To obtain biologically meaningful solutions, we solve the TRS-Gene-ID problem in a progressive manner. More specifically, given user-specified parameters \((\tau_{\text{min}}, \tau_{\text{max}})\), we start with setting \(\tau = \tau_{\text{max}}\) and solve the problem for a fixed window \(u\). Then, for every subsequent iteration we decrement \(\tau\) and require the solution to contain all ORFs that were identified in the previous iteration. The final iteration occurs when \(\tau = \tau_{\text{min}}\), yielding the final solution for window \(u\). We consider all sliding windows \(u\) within \(v_{\text{leader}}\) and return the solution with maximum genome coverage and subsequently maximum score. We show that the progressive approach performs better than solving the TRS-Gene-ID problem directly using \(\tau = \tau_{\text{min}}\) or \(\tau = \tau_{\text{max}}\) in Fig. S19. We set \(\tau_{\text{min}} = 2\) and \(\tau_{\text{max}} = 7\) after performing a grid search and cross validation, as described in Section 2.3.

1.6 Classifier to identify 5’-truncated genomes with missing TRS-L

In some cases the input genome to CORSID or CORSID-A is a truncated genome, whose 5’ UTR is incomplete or absent and the TRS-L is hence missing. As performance of CORSID and CORSID-A will be adversely affected in such cases, we trained a XGBoost [5] classifier to distinguish genomes with an intact TRS-L subsequence from truncated genomes that lack TRS-L. XGBoost is a tree boosting method that learns to classify samples by gradually adding decision trees into the ensemble by using gradient boosting technique. Specifically, we ran CORSID for each genome, obtaining the following 7 features for XGBoost based on the reported optimal solution and the identified ORF1ab.

1. Position of the identified leader core sequence.
2. Total TRS alignment score.
3. Distance between the leader core sequence and the start codon of ORF1ab.
4. Mean information content of the multiple sequence alignment of the core sequence\(^2\).
5. Mean alignment score for the TRS-Bs.
6. Genome coverage, i.e. the proportion of the genome that is covered by genes.
7. Position of the start codon of ORF1ab.

We labeled each genome to indicate if it contains the TRS-L which was determined by comparing the 5’ UTR of all genomes in the same subgenus using multiple sequence alignment (section 2.1).

\(^2\)The mean information content is \(\frac{1}{n} \sum_{i=1}^{n} (\log_2 4 - H_i)\), where \(n\) is the length of the core sequence, and \(H_i\) is the entropy of column \(i\) in the alignment.
There are only 35 truncated genomes without TRS-L among the original 505 genomes, which leads to an imbalance between positive and negative instances. In order to create a larger training set, we simulated additional incomplete genomes by either (1) truncating a region in the 5’ UTR containing the TRS-L, or (2) removing the entire sequence upstream of a random position among the first 500 bases of the ORF1ab of the 468 complete genomes. Recall that CORSID has a minimum score threshold $\tau_{\text{min}}$, which constrains each TRS-B sequence $\alpha_i$ to meet a minimum alignment score $s(\alpha_0, \alpha_i) \geq \tau_{\text{min}}$ with respect to TRS-L $\alpha_0$. As such, CORSID may not identify any solution for truncated genomes. Among the 505 genomes truncated in two different ways, CORSID identified solutions for 438 genomes where only TRS-L was truncated and 473 genomes where additionally ORF1ab was truncated. Thus, in addition to the original 505 genomes with 35 truncated genomes, we obtained an additional $438 + 473 = 911$ truncated genomes. Then we divided the $505 + 438 + 473 = 1416$ genomes into a 20% test set and 80% training set. We tuned hyperparameters for XGBoost (maximum tree depth, learning rate, and minimum weight for a child node) on the training set using 5-fold cross validation. In these preprocessing steps, we performed stratified splitting according to genus, such that the same ratio of the four genera was maintained in the training set, test set, and each fold of the cross validation. We selected the best combination of hyperparameters according to the average prediction accuracy in the testing fold. After selecting the best combination of parameters, our classifier achieved an accuracy of 0.972 on the test set. More specifically, the accuracy in each genus was 0.971 in Alphacoronavirus, 0.983 in Betacoronavirus, 0.973 in Gammacoronavirus, and 0.889 in Deltacoronavirus. The accuracy was 0.921 for the original 505 genomes, 1.00 for simulated incomplete genomes where only TRS-L was truncated, and 1.00 for simulated incomplete genomes where additionally ORF1ab was truncated. After applying this classifier with CORSID on the 468 complete genomes, we obtained an accuracy of 0.944, since the classifier returned 26 false positives. Among these 26 false positive genomes, CORSID predicted the wrong TRS-L in 17 genomes and was correct in the remaining 9. Strikingly, CORSID showed performance on these 26 misclassified genomes compared with the correctly classified 442 genomes. That is, the 26 misclassified genomes have a lower recall (median 0.80 vs 1.0), a lower precision (median 0.67 vs 0.90), a lower $F_1$ score (median 0.72 vs 0.92), and a lower TRS-L accuracy (0.35 vs 0.99) (see Fig. S17).

We incorporated the trained classifier into CORSID so that it outputs a warning when the genome is classified as potentially lacking TRS-L. This warning is also produced when CORSID fails to find any solution.
2 Supplementary Results

2.1 Evaluating TRS site identification methods

To evaluate methods for the TRS-ID problem, we established a ground-truth set of genes by processing the GFF annotation files and extracting a set of genes for each genome, removing duplicates and incomplete ORFs. In particular, we removed 10 ORFs (8 annotated as N and 2 annotated as ‘unknown’) that had duplicate names and were completely covered by another gene with the same name. This resulted in a median number of 8 genes per genome (min 3 and max 14, Fig. S1). We excluded two genomes that had no genes in their annotation (DQ288927 and EU526388).

We used the following approach to establish the ground truth for TRS-B sites. Following Definition 2, we identified a candidate region for each gene. If a method identified a TRS-B contained within any of the candidate regions, then we counted this as the method recalling the TRS-B for the corresponding gene.

We established ground-truth locations of TRS-Ls using the fact that these regulatory sequences occur between the second (SL2) and fourth stem loop (SL4) in the 5’ untranslated region (UTR) [13]. For Sarbecovirus genomes, a subgenus of the Betacoronavirus genus, we required TRS-Ls to occur in stem loop (SL3) [4]. Specifically, we analyzed leader regions upstream of ORF1ab, its location taken from the GFF annotation or determined by our own heuristic in case this gene was absent, and performed a multiple sequence alignment using ClustalW2 [12] of all sequences within each subgenus. We then superimposed secondary structure information from Rfam [8] onto each alignment to identify the relevant stem loops in each viral sequence, obtaining for each sequence a small range in which TRS-L may occur (Fig. S2).

2.2 Evaluating gene finding methods

Recognizing that many of the GFF files of the 468 genome sequences that we considered are incomplete, we used BLASTx [1] to transfer gene annotation from well-annotated genomes to poorly-annotated genomes. We initialized the set of ground-truth genes to contain all genes in the GFF files that meet the filtering criteria described previously. Then, we evaluated every predicted gene by all four methods (Glimmer3, Prodigal, VADR and CORSID) and added it to the ground truth provided that the alignment reported by BLASTx spans the entire query sequence (the predicted gene) and the hit sequence in the database, i.e. represents an exact match (detailed commands in Appendix 2.4). Applying this procedure, we increased the number of ground-truth genes by 148 genes starting from the original 3883 genes obtained from the GFF files (Fig. S1).

To assess the performance of gene finding methods, we say that a predicted gene \( x = [x^-, x^+] \) is correct provided there exists a ground-truth gene \( y = [y^-, y^+] \) in the same genome such that \( |x^- - y^-| \leq 3 \) and
\[ |x^+ - y^+| \leq 3. \] In other words, the start and end positions may be off by the length of at most one codon, accounting for variation in annotation (e.g. sometimes the stop codon is omitted from ORFs). Moreover, as ORF1ab is not a real ORF, i.e. the corresponding polypeptide 1ab results from a -1 frameshift, we treat this gene differently. That is, we say that a method correctly identified \( y = [y^-, y^+] = \text{ORF1ab} \) if it found two genes \( x_1 = [x_1^-, x_1^+] \) and \( x_2 = [x_2^-, x_2^+] \) such that the start position of \( x_1 \) is at most 3 nucleotides away from \( y^- \) and the stop position of \( x_2 \) is at most 3 nucleotides away from \( y^+ \). Note that CORSID will identify a single gene \( x \) matching \( \text{ORF1ab} \), in which case \( x = x_1 = x_2 \). Using this definition we classify each predicted and ground-truth gene as either a true positive (TP), i.e. the predicted gene matches a ground-truth gene; false positive (FP), i.e. the predicted gene does not match any ground-truth gene); or false negative (FN), i.e. the ground-truth gene has no matching predicted gene.

### 2.3 Tuning parameters for CORSID

We performed a 5-fold cross-validation to identify default values for the parameters used in CORSID. Specifically, we varied the minimum score threshold \( \tau_{\text{min}} \in \{2, 3, 4, 5, 6, 7\} \), gene shrinking percentage \( \alpha \in \{0, 0.01, 0.03, 0.05, 0.07\} \) and core sequence length \( \omega \in \{5, 6, 7\} \), amounting to a total of 90 combinations. We split the set of 468 complete coronavirus genomes into a training set of 375 genomes (80%) and a test set of 93 genomes (20%), stratified by genus. For each fold and parameter combination, we calculated the TRS-L accuracy (section 2.1) and median \( F_1 \) score for gene identification (section 2.2), taking the harmonic mean. Next, we computed the average harmonic mean among 5 folds. Parameter combination \( \tau_{\text{min}} = 2, \alpha = 0.05, \omega = 7 \), achieved the highest harmonic mean of 0.937 in the test set, with the highest \( F_1 \) score of 0.917 and the fourth highest TRS-L accuracy of 0.952 (Fig. S10). Fig. S11a shows that CORSID achieved a higher \( F_1 \) score and recall compared to Glimmer3, Prodigal, and VADR in the test set. Fig. S11b shows that CORSID’s TRS-L accuracy dropped slightly compared to CORSID-A (taking gene locations as input from public GFF annotation files), or when CORSID-A was run after gene prediction by VADR.

### 2.4 Command-line arguments

**MEME.** We ran MEME v5.3.0 in mode “zero or one occurrence per sequence” (zoops) and maximum width of 10.

```
1     python corsid-data/analysis/candidate_region.py -f ${input_fasta} \n2       -g ${input_gff} -o ${fasta} -m 10 --prev2ATG
3     meme ${fasta} -dna -maxw 10 -oc ${output_dir} \n4       -mod zoops -nmotifs 1
```

9
We used our own script `corsid-data/analysis/candidate_region.py` to prepare MEME input from the GFF and FASTA files, which is available at [https://github.com/elkebir-group/CORSID-data](https://github.com/elkebir-group/CORSID-data). This script extracts the same candidate region that CORSID-A uses from the FASTA files and store them for input to MEME.

**Glimmer3.** We followed the steps written in the script `g3-from-scratch.csh` provided in the Glimmer3 package.

```bash
1. long-orfs -n -t 1.15 $(input_fasta) $(long_orfs)
2. extract -t $(input_fasta) $(long_orfs) > $(train)
3. build-icm -r $(icm) < $(train)
4. glimmer3 -g 100 $(input_fasta) $(icm) $(dir)
```

**Prodigal.** We ran Prodigal v2.6.3 in metagenomic mode.

```bash
1. prodigal -i $(input_fasta) -p meta -f gff -o $(output_gff) -s $(output_gene)
```

**ClustalW2.** We used ClustalW2 v2.1 to align sequences.

```bash
1. clustalw2 -infile=$(fasta)
```

**BLASTx.** We ran BLASTx on a FASTA file containing FP genes with the following parameters. We used a subset of protein sequences from the official “nr” database (downloaded on October 7, 2021), containing all species under the taxonomic unit *Coronaviridae* (taxid:11118). From the BLASTx output, we extracted the top hit for each FP gene, and calculated the coverage between the aligned subsequence, and the query and hit sequences. In particular, the *hit coverage* is fraction of position in the hit sequence that are aligned. On the other hand, the *query coverage* is the fraction the query sequence that are aligned. We set the threshold to 95% for both hit and query coverage, restricting solutions to almost exact matches.

```bash
1. blastx -db $(database) -query $(fasta) -word_size 6 -gapopen 11 -gapextend 1 -comp_based_stats 2 -task blastx -max_target_seqs 15 -evalue 0.05 -num_threads 6 -outfmt 15 -out $(json)
```

**VADR.** We ran VADR using their official database for coronaviruses and the recommended parameters ([https://ftp.ncbi.nlm.nih.gov/pub/nawrocki/vadr-models/coronaviridae/1.3-3/vadr-models-corona-1.3-3.tar.gz](https://ftp.ncbi.nlm.nih.gov/pub/nawrocki/vadr-models/coronaviridae/1.3-3/vadr-models-corona-1.3-3.tar.gz)).
2.5 Two anomalous coronavirus genomes

CORSID-A was unsuccessful in identifying the correct TRS-L site in only two coronaviruses: MK211372 (from subgenus Pedacovirus of genus Alphacoronavirus) and MK472070 (unclassified subgenus of genus Alphacoronavirus).

To understand why CORSID-A failed on MK211372, we performed a multiple sequence alignment of the leader regions of all 45 genomes in the Pedacovirus subgenus. Inspecting the alignment, we see that MK211372 is an outlier, with multiple insertion/deletions in the TRS-L region compared to the other sequences (Fig. S3). This explains why CORSID-A was unable to accurately identify the TRS-L and TRS-B sites for this genome.

Since genome MK472070 has a known genus but unknown subgenus, we only aligned it to the covariance model of the alphacoronaviruses. From the alignment result we found a poor alignment in the TRS-L region. Based on the alignment, it resembles some nyctacoviruses, but it is still an outlier, as shown in Fig. S4. Specifically, the TRS-L consensus sequence, 5’-TCAACTAAAC-3’, differed significantly from the subsequence 5’-ACAATCTAAT-3’ of MK472070, with a Hamming distance of 5. Moreover, MEME also failed to identify TRS-L in this genome, and SuPER assigned a low confidence score to the identified TRS-Bs for important genes such as S and N.

In summary, we believe further investigation of genomes MK211372 and MK472070 is warranted in order to determine whether they harbor a TRS-L region or whether the deposited genome sequences are incomplete/incorrect.

2.6 Non-AUG start codons

Among 3637 annotated genes from the ground truth of 468 complete genomes, only 7 genes had non-AUG start codons (Table S3). We leave supporting non-AUG start codons as future work.
2.7 Scope and recommendations on using CORSID and CORSID-A

CORSID is a gene annotation method specifically designed for de novo identification of viral genes in coronaviruses using only the nucleotide sequence of the viral genome. It leverages the genomic structure of coronaviruses, specifically the transcription regulatory sequences (TRSs) located upstream of the genes in the genome, to perform accurate gene identification. CORSID-A, on the other hand, is a tool designed to identify the TRSs in a coronavirus genome annotated with gene locations. Since most coronavirus genes are preceded by TRSs, CORSID-A can also be used to remove putative viral genes that do not support a TRS with high confidence. As such, CORSID-A can be used to reduce false positives in the results of existing gene finding tools on coronavirus genomes. We explore the performance of this strategy and its comparison with CORSID in this section.

Specifically, we ran VADR [16] for a first-pass gene annotation of the genome. Then we ran CORSID-A to identify the TRSs and filter out the genes that do not support a TRS. We see that this strategy, which we refer to as VADR+CORSID-A, indeed achieved higher precision (median 0.909) compared to VADR in isolation (0.800) while having similar recall (0.889 vs 0.900). This led to an higher $F_1$ score VADR+CORSID-A as compared to VADR (mean is 0.892 vs 0.859, medians were the same at 0.909). However, CORSID still achieved the higher $F_1$ score (mean 0.892, median 0.923) with a precision of 0.889 and recall of 1.00. Looking at the accuracy of identifying TRS sites (Fig. S13), we see that VADR+CORSID-A has the highest accuracy (0.987, 462/468 genomes) while CORSID is a close second (0.957, 448/468 genomes). In summary, while we recommend the use of CORSID for de novo gene identification of coronavirus given only their genomes, our results show that CORSID-A should be used for identifying TRS sites in coronavirus genomes annotated with gene locations by existing gene-finding tools such as VADR.
3 Supplementary Figures and Tables

We have the following supplementary figures and tables.

• Fig. S1 shows the histogram of length of annotated genes from 468 genomes and the histogram of increment genes after the BLASTx adjustment.

• Fig. S2 shows the histogram of the length of the TRS-L regions, separated by four genera.

• Fig. S3 shows the MSA (multiple sequence alignment) of leader regions of pedacoviruses with genome MK211372 highlighted, indicating multiple INDELs (insertions and deletions) in the TRS-L region.

• Fig. S4 shows the MSA of leader regions of some alphacoronaviruses with genome MK472070 highlighted, indicating multiple INDELs in the TRS-L region.

• Fig. S5 shows CORSID finds the correct TRS-L in a Sarbecovirus, verified using a RNA-seq dataset.

• Fig. S6 shows RNA-seq data supports the TRS-B found by CORSID-A in SARS-CoV-1 (NC_004718), and CORSID identifies more genes than Glimmer3 and Prodigal in the same genome.

• Fig. S7 shows the number of coronaviruses with varying lengths of TRS-L regions broken down by the genus and the subgenus. Several coronaviruses of each genus have TRS-L regions much longer than the core sequences indicating recombination events.

• Fig. S8 shows the TRS alignment in genome NC_006577 with a TRS-L of 36 nucleotides long, indicative of a recombination and/or a TRS-L guided insertion event.

• Fig. S9 shows that the genomes with a TRS-L longer than 25 nt have a core sequence with a median actual length of 7 and a median Hamming distance of 1 between the core sequence within the longest TRS-B and the core sequence within TRS-L.

• Fig. S10 shows the comparison of the performance of varying hyperparameters of CORSID such as minimum alignment score threshold $\tau_{\text{min}}$, shrinkage percentage $\alpha$ and core sequence length $\omega$ in term of the harmonic mean of $F_1$ score for gene identification and TRS-L accuracy.

• Fig. S11 shows the CORSID results on a test set of 87 genomes with the best performing hyperparameters in Fig. S10.
• Fig. S12 shows that CORSID achieves a better precision and recall (also pooled precision and recall) compared to VADR and VADR+CORSID-A after excluding 48 genomes present in VADR’s reference database.

• Fig. S13 shows that CORSID shows modest reduction in TRS-L accuracy compared to CORSID-A and VADR+CORSID-A.

• Fig. S14 illustrates the recurrence for solving CONSTRAINED TRS AND GENE IDENTIFICATION problem and gives an example.

• Fig. S15 shows the histogram of length of annotated genes in coronaviruses.

• Fig. S16 illustrates the process of removing overlapping candidate regions.

• Fig. S17 shows that the classifier identifying 5’-truncated genomes with missing TRS-L failed on 26/468 genomes.

• Fig. S18 shows an example of TRS regions overlapping with the start codon of the corresponding genes in coronaviruses. Only 10 out of 3637 genes are shorter than 100 nt.

• Fig. S19 shows that CORSID achieves better performance when using the progressive approach rather than directly solving $\tau = \tau_{\text{max}} = 7$ or $\tau = \tau_{\text{min}} = 2$.

• Fig. S20 shows that CORSID achieves higher $F_1$ score compared to Glimmer3, Prodigal, VADR and the pipeline of VADR+CORSID-A for identifying genes in coronaviruses.

• In Table S1 we compare features of CORSID, CORSID-A, and other methods. We show that CORSID is the first method for simultaneous identification of TRS sites and genes in coronaviruses.

• Table S2 shows the number of genomes included and excluded in this study, grouped by genera and subgenera.

• Table S3 shows 7 genes with non-AUG start codons among 3637 annotated genes from the ground truth of 468 complete genomes.

References

[1] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990.
[2] Timothy L Bailey, Mikael Boden, Fabian A Buske, Martin Frith, Charles E Grant, Luca Clementi, Jingyuan Ren, Wilfred W Li, and William S Noble. MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Research*, 37(suppl 2):W202–W208, 2009.

[3] Pavel V Baranov, Clark M Henderson, Christine B Anderson, Raymond F Gesteland, John F Atkins, and Michael T Howard. Programmed ribosomal frameshifting in decoding the SARS-CoV genome. *Virology*, 332(2):498–510, 2005.

[4] Shih-Cheng Chen and René CL Olsthoorn. Group-specific structural features of the 5’-proximal sequences of coronavirus genomic RNAs. *Virology*, 401(1):29–41, 2010.

[5] Tianqi Chen and Carlos Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’16, pages 785–794, New York, NY, USA, 2016. ACM.

[6] Arthur L Delcher, Kirsten A Bratke, Edwin C Powers, and Steven L Salzberg. Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics*, 23(6):673–679, 2007.

[7] Alexander Dobin, Carrie A Davis, Felix Schlesinger, Jorg Drenkow, Chris Zaleski, Sonali Jha, Philippe Batut, Mark Chaisson, and Thomas R Gingeras. STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

[8] S. Griffiths-Jones. Rfam: an RNA family database. *Nucleic Acids Research*, 31(1):439–441, January 2003.

[9] Ju Yuan Hsiao, Chuan Yi Tang, and Ruay Shiung Chang. An efficient algorithm for finding a maximum weight 2-independent set on interval graphs. *Information Processing Letters*, 43(5):229–235, 1992.

[10] Doug Hyatt, Gwo-Liang Chen, Philip F LoCascio, Miriam L Land, Frank W Larimer, and Loren J Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, 11(1):1–11, 2010.

[11] Marilyn Kozak. Initiation of translation in prokaryotes and eukaryotes. *Gene*, 234(2):187–208, 1999.

[12] Mark A Larkin, Gordon Blackshields, Nigel P Brown, R Chenna, Paul A McGettigan, Hamish McWilliam, Franck Valentin, Iain M Wallace, Andreas Wilm, Rodrigo Lopez, et al. Clustal W and Clustal X version 2.0. *Bioinformatics*, 23(21):2947–2948, 2007.
[13] Ramakanth Madhugiri, Markus Fricke, Manja Marz, and John Ziebuhr. RNA structure analysis of alphacoronavirus terminal genome regions. *Virus Research*, 194:76–89, 2014.

[14] Helena Jane Maier, Erica Bickerton, and Paul Britton. Coronaviruses. *Methods and Protocols*, 2015.

[15] Christian Jean Michel, Claudine Mayer, Olivier Poch, and Julie Dawn Thompson. Characterization of accessory genes in coronavirus genomes. *Virology Journal*, 17(1):1–13, 2020.

[16] Alejandro A Schäffer, Eneida L Hatcher, Linda Yankie, Lara Shonkwiler, J Rodney Brister, Ilene Karsch-Mizrachi, and Eric P Nawrocki. VADR: validation and annotation of virus sequence submissions to GenBank. *BMC Bioinformatics*, 21:1–23, 2020.

[17] Aditi Shukla and Rolf Hilgenfeld. Acquisition of new protein domains by coronaviruses: analysis of overlapping genes coding for proteins N and 9b in SARS coronavirus. *Virus Genes*, 50(1):29–38, 2015.

[18] Temple F Smith, Michael S Waterman, et al. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.

[19] Ke Xu, Bo-Jian Zheng, Rong Zeng, Wei Lu, Yong-Ping Lin, Liang Xue, Li Li, Lei-Lei Yang, Chen Xu, Jie Dai, et al. Severe acute respiratory syndrome coronavirus accessory protein 9b is a virion-associated protein. *Virology*, 388(2):279–285, 2009.

[20] Yiyan Yang, Wei Yan, A Brantley Hall, and Xiaofang Jiang. Characterizing transcriptional regulatory sequences in coronaviruses and their role in recombination. *Molecular Biology and Evolution*, 38(4):1241–1248, 2021.
Figure S1: Number of genes in the original ground truth and increment after the BLASTx adjustment varies across the four genera of coronaviruses. (a) Number of genes extracted from GFF annotation file. Median: 8, min: 3, max: 14; (b) Number of genes added to ground truth after BLASTx. Median: 0, min: 40, max: 4.

Figure S2: Length of the TRS-L region varies across the four genera of coronaviruses.
**Figure S3:** MSA of leader regions of pedacoviruses. We highlight the MK211372 genome and indicate the TRS-L region with a red bar on top. MK211372 differs from other sequences since it contains multiple indels in the TRS-L region. As such, the TRS-L region cannot be found in this genome.

**Figure S4:** MSA of leader regions of some alphacoronaviruses. We highlight MK472070 genome and indicate the TRS-L region with red bar on top. MK472070 differs from others since it contains multiple indels in the TRS-L region. As such, the TRS-L region cannot be found in this genome.
**Figure S5:** CORSID-A finds the correct TRS-L site in a *Sarbecovirus* unlike SuPER. Although SuPER uses a hard-coded motif to identify core sequences, it incorrectly identified TRS-L (positions 19 – 25) in genome MN996532 (Bat coronavirus RaTG13). By contrast, CORSID-A found the TRS-L at a different location (54 – 60). We verified CORSID-A’s position using the corresponding RNA-seq data (SRR11085797). We aligned the reads to the reference genome using a splice-aware aligner, STAR [7]. The resulting alignment had a single split read, spanning positions 54 and 28,221, which matches CORSID-A’s TRS-L and the TRS-B for gene N. Moreover, TRS-L sites in sarbecoviruses occur in stem loop (SL3) [4], which coincides with the TRS-L site identified by CORSID-A.

|                      | TRS-L identification | TRS-B identification | Gene identification |
|----------------------|----------------------|----------------------|---------------------|
| CORSID               | ✓                    | ✓                    | ✓                   |
| CORSID-A             | ✓                    | ✓                    | ✓                   |
| SuPER [20]           | ✓                    | ✓                    | ✗                   |
| MEME [2]             | ✓                    | ✓                    | ✗                   |
| Glimmer3 [6]         | ✗                    | ✗                    | ✓                   |
| Prodigal [10]        | ✗                    | ✗                    | ✓                   |
| VADR [16]            | ✗                    | ✗                    | ✓                   |

**Table S1:** CORSID is the first method for simultaneous identification of TRS sites and genes in coronaviruses. This table shows the features of CORSID and CORSID-A along with four existing methods: MEME [2], Glimmer3 [6], Prodigal [10], and VADR [16].
Figure S6: RNA-seq data supports the TRS-B found by CORSID-A, and CORSID identifies more genes than Glimmer3 and Prodigal in a SARS-CoV-1 genome (NC_004718) (a) For SARS-CoV-1 (NC_004718), CORSID-A identified a TRS-B corresponding to ORF7b supported by 246 and 233 split reads in RNA-sequencing samples SRR1942956 and SRR1942957, respectively. For the same genome, SuPER identified a TRS-B site that is supported by only one read in each RNA-sequencing sample. (b) A Venn diagram of the genes correctly identified by four methods. (c) The TRS alignment of CORSID-A, and aligned core sequence with flanking regions of SuPER. "*": SuPER identified a TRS-B for 9b but its start codon is located at the second to fourth nucleotide of the core sequence, and the Hamming distance is 2. CORSID-A did not find this TRS-B as it occurs outside the candidate region of 9b. Moreover, in previous studies ORF9b has been hypothesized to be translated via a leaky scanning mechanism [15,17,19], explaining the absence of an associated TRS-B site.
Figure S7: Several coronaviruses of each genus have TRS-L regions much longer (median: 22 nt) than the core sequence length (6–7 nt) indicating 5' UTR insertion and/or recombination events.

Figure S8: Location of genes and the TRS alignment inferred by CORSID-A for Human coronavirus HKU1, (genome NC_006577). This genome has a TRS-B with a length of 36 nucleotides (without any mismatches) upstream of gene 4, and a TRS-B with a length of 27 nucleotides (with only 1 mismatch) upstream of gene HE, indicative of a recombination event and/or a TRS-L guided insertion.
Figure S9: Genomes with a TRS-L longer than 25 nt have a core sequence with a median actual length of 7 and a median Hamming distance of 1 between the core sequence within the longest TRS-B and the core sequence within TRS-L. The genomes for which CORSID-A identified a TRS-L longer than 25 nt has a core sequence with median actual length of 7 median Hamming distance of 1. We included this figure into the supplement as fig. S8. (a) Hamming distance between the core sequence in TRS-L and the core sequence of the gene with the longest TRS-B in each genome identified by CORSID-A. For clarity, we added a small jitter to dots in the scatter plot representing each genome. (b) The core sequence length of the core sequence for each genome. The actual core sequence length is the length of the maximum consecutive columns in the TRS alignment without gaps. By definition, the actual core sequence length is longer than our minimum core sequence length threshold $\omega = 7$.

Figure S10: In cross validation, the parameter combination with minimum score threshold $\tau_{\min} = 2$, shrinkage percentage $\alpha = 0.05$, and core sequence length $\omega = 7$ is optimal. Y axis: harmonic mean of averaged $F_1$ score and TRS-L accuracy in 5 folds for the parameter combination.
Figure S11: Results on a test set of 87 genomes where CORSID was run with parameter combination identified during 5-fold cross validation $\tau_{\text{min}} = 2, \alpha = 0.05, \omega = 7$ (Fig. S10). Note that we removed the 6 genomes that VADR used for training from the test set (93 genomes in total). (a) CORSID achieved the highest $F_1$ score with the optimal parameter combination. (b) On the other hand, TRS-L accuracy dropped slightly compared to CORSID-A (taking gene locations as input from public GFF annotation files), or when CORSID-A is run after gene prediction by VADR.
Figure S12: CORSID achieves higher $F_1$ score compared to Glimmer3, Prodigal, VADR and the pipeline of VADR+CORSID-A for identifying genes in coronaviruses. (a) Precision and recall of Glimmer3 [6], Prodigal [10], VADR [16], and CORSID for gene prediction in 420 genomes, excluding genomes used in VADR’s reference database. For clarity, we added a small jitter (drawn from $N(0, 2.5 \times 10^{-5})$) to the 2D contour plots of recall and precision. (b) Confusion matrices of the adjusted ground truth genes and the predicted genes by the each method. CORSID achieves higher scores in all pooled metrics. Panels (c) and (d) show results in the same format but are restricted to the 48 genome sequences that are also present in VADR’s reference database.
| Genus            | Subgenus     | # Included genomes | # Excluded genomes |
|------------------|--------------|--------------------|--------------------|
| Alphacoronavirus | Colacovirus  | 0                  | 2                  |
|                  | Decacovirus  | 12                 | 2                  |
|                  | Duvinacovirus| 5                  | 1                  |
|                  | Luchacovirus | 2                  | 0                  |
|                  | Minacovirus  | 4                  | 2                  |
|                  | Minunacovirus| 11                 | 0                  |
|                  | Myotacovirus | 1                  | 0                  |
|                  | Nyctacovirus | 6                  | 2                  |
|                  | Pedacovirus  | 39                 | 6                  |
|                  | Rhinacovirus | 5                  | 2                  |
|                  | Setracovirus | 4                  | 0                  |
|                  | Tegacovirus  | 45                 | 1                  |
|                  | Unclassified | 4                  | 1                  |
| **Subtotal**     |              | 138                | 19                 |
| Betacoronavirus  | Embecovirus  | 36                 | 1                  |
|                  | Hibecovirus  | 1                  | 0                  |
|                  | Merbecovirus | 27                 | 1                  |
|                  | Nobecovirus  | 11                 | 0                  |
|                  | Sarbecovirus | 34                 | 6                  |
| **Subtotal**     |              | 109                | 8                  |
| Deltacoronavirus | Andecovirus  | 1                  | 0                  |
|                  | Buldecovirus | 19                 | 0                  |
|                  | Herdecovirus | 1                  | 0                  |
| **Subtotal**     |              | 21                 | 0                  |
| Gammacoronavirus | Cegacovirus  | 3                  | 0                  |
|                  | Igacovirus   | 196                | 10                 |
|                  | Unclassified | 1                  | 0                  |
| **Subtotal**     |              | 200                | 10                 |
| **Totals**       |              | 468                | 37                 |

Table S2: Number of coronaviruses of each genus and subgenus included and excluded in this study.
Figure S13: CORSID shows a modest reduction in TRS-L accuracy compared to CORSID-A and VADR+CORSID-A. (a) TRS-L accuracy. (b) Venn diagram of the TRS-L identified by CORSID, CORSID-A, and VADR+CORSID-A compared to the ground truth. All methods failed on genomes MK211372 and MK472070 (discussed in Appendix 2.5). There are four samples where VADR+CORSID-A failed but others succeeded (NC_039207, NC_035191, KC545386, and MK423877), this failure was due to VADR.

Figure S14: Illustration of the recurrence for solving the CONSTRAINED TRS IDENTIFICATION problem. The green region containing pairs \((p, q)\) corresponding to feasible constrained TRS alignments is enclosed by \(\ell + |u| - 1 \leq p \leq |w_0|\) and \(p - \ell + 1 \leq q \leq |w_i|\). The blue region contains the cells that must be traversed by any feasible solution in the green region. Here, we set \(\ell = 2\) and \(|u| = 5\) so that \(p \geq 6\) and \(q \geq p - 1\).
Figure S15: Coronavirus genes are almost always longer than 100 nt. Histogram of annotated gene lengths from 468 genomes (ORF1ab not included). Only 10 out of the 3637 genes are shorter than 100 nt.

Figure S16: Illustration of removing overlap among candidate regions. When \( w_{i-1}^+ > w_i^- \), we move the 5’ end of \( w_i \) by setting \( w_i^- \) to the next base downstream of \( w_{i-1} \)'s 3’ end, i.e., \( w_i^- := w_{i-1}^+ + 1 \).
Figure S17: The classifier identifying 5’-truncated genomes with missing TRS-L failed on 26/468 genomes. For these 26 misclassified genomes, CORSID achieved lower (a) recall, (b) precision and (c) F1 score for genes as well as lower (d) TRS-L accuracy compared to the remaining 442 genomes.

Bat SARS-like coronavirus YNLF_31C (KP886808)

| ORF | score | ORF length | TRS alignment | Core sequence |
|-----|-------|-------------|---------------|---------------|
| L   | n/a   | n/a         | GTAGATCTTTCTTTGAAGAACCTTTAAAT | S 12 3723 |
| S   | 12    | 3723        | TTAGAGGAACAGAAATTTAAAT | 7a 14 366 |
| E   | 8     | 228         | TAAACGACTAACTAACTAAA | 8 9 366 |
| M   | 10    | 663         | TAAACGACTAACTAACTAAA | N 25 1263 |
| 7a  | 14    | 366         | TTAACGAACTTTAAATCTTTAAAT | 7a 14 366 |
| 7b  | 5     | 132         | ACTAACTAAA | 7b 5 132 |
| 8   | 9     | 366         | TAAACGAACTTTAAATCTTTAAAT | 8 9 366 |
| N   | 25    | 1263        | GTAGATCTTTCTTTGAAGAACCTTTAAAT | N 25 1263 |

Figure S18: TRS sites may overlap with the start codon of the corresponding genes in coronaviruses. The TRS alignment identified by CORSID when applied to a Sarbecovirus genome KP886808, showing the start codons of gene S, 7a, and 8 are partially contained in core sequences (highlighted in yellow). We note that MEME identified the same TRS-L as our method. As a side note, the TRS-B of gene N matches 25 nucleotides of the TRS-L, indicative of a possible recombination and/or TRS-L derived insertion event.
Figure S19: Using the progressive approach rather than directly solving $\tau = \tau_{\text{max}} = 7$ or $\tau = \tau_{\text{min}} = 2$ leads to better performance. (a) The $F_1$ score (harmonic mean between precision and recall) is shown in top left panel. We show the precision and recall in the top right and lower left panel, respectively, and the joint distribution in the lower right panel. (b) TRS-L accuracy of setting the minimum matching score threshold to $\tau = 2$ and $\tau = 7$, compared with the progressive approach. (c) Venn diagram of genome sets with correctly identified TRS-L by the three versions of CORSID.

| Genome    | Gene  | Start codon |
|-----------|-------|-------------|
| MK472067  | envelope | CUG        |
| MK472070  | envelope | CUG        |
| JF893452  | envelope | UUG        |
| JQ088078  | 2b      | CGG         |
| KT886454  | membrane | UUG        |
| NC_046965 | ORF10   | CUG         |
| NC_046965 | ORF11   | UUG         |

Table S3: Genes with non-AUG start codons among 3637 annotated genes from the ground truth of 468 complete genomes.
Figure S20: CORSID achieves higher $F_1$ score compared to Glimmer3, Prodigal, VADR and the pipeline of VADR+CORSID-A for identifying genes in coronaviruses. (a) Precision, recall, and $F_1$ for gene prediction in 468 genomes. For clarity, we added a small jitter (drawn from $N(0, 2.5 \times 10^{-5})$) to the 2D distribution plot. Compared to standalone VADR, VADR+CORSID-A increases the precision without a significant decrease in recall. The $F_1$ scores of both CORSID and VADR+CORSID-A are significantly higher than VADR’s ($P = 4.7 \times 10^{-3}$ and $P = 5.0 \times 10^{-5}$ respectively, one-sided Wilcoxon signed-rank test). (b) Confusion matrices of the ground truth genes and the predicted genes by each method. VADR+CORSID-A achieves the highest pooled precision while CORSID achieves the highest $F_1$ score.