Methods

SEARCH-MaP of the 2,924 nt segment of SARS-CoV-2 genomic RNA

A DNA template of the 2,924 nt segment of the SARS-CoV-2 genome, plus an upstream T7 promoter, was PCR-amplified from our previously constructed pmirGLO plasmid **?** with 250 nM primers TAATACGACTCACTATAGAATAATGAGCTTAGTCCTGTTGCACTACG and TAAATTGCGGACATACTTATCGGCAATTTTGTTACC (Thermo Fisher) using 2X CloneAmp HiFi PCR Premix (Takara) in a 50 µl volume with initial denaturation at 98°C for 60 s; 35 cycles of 98°C for 10 s, 65°C for 10 s, and 72°C for 15 s; and final extension at 72°C for 60 s. The 50 µl PCR product was mixed with 10 µl of 6X Purple Loading Dye (New England Biolabs) alongside 10 µl of 0.1X 1 kb DNA Ladder (New England Biolabs) and electrophoresed through a 1% agarose gel – 50 ml of 1X tris-acetate-EDTA buffer (Boston BioProducts), 0.5 g of agarose powder, and 5 µl of 10,000X SYBR Safe (Thermo Fisher) – in 1X tris-acetate-EDTA buffer (Boston BioProducts) within a Mini-Sub cell GT (Bio-Rad) at 60 V for 60 min. The band at roughly 3 kb was excised and the DNA purified using a Zymoclean Gel DNA Recovery Kit (Zymo Research) according to the manufacturer's protocol; samples were eluted in 10 µl of nuclease-free water (Fisher Bioreagents) and measured with a NanoDrop (Thermo Fisher). To increase the DNA yield, the gel-extracted DNA was amplified by a second round of PCR followed by gel extraction, using the same protocol as above. To remove contaminants after the second gel extraction, the DNA was further purified using a DNA Clean & Concentrator-5 kit (Zymo Research) according to the manufacturer's protocol; samples were eluted in 10 µl of nuclease-free water (Fisher Bioreagents) and measured with a NanoDrop (Thermo Fisher).

RNA was transcribed using the MEGAscript T7 Transcription Kit (Thermo Fisher) according to the manufacturer's protocol. Briefly, 1 µl (150 ng) of DNA template from the previous step was diluted in 7 ul of nuclease-free water (Fisher

Bioreagents), mixed with 2 µl of each 10X ribonucleotide solution followed by 2 µl of the 10X reaction buffer and 2 µl of the 10X enzyme mix, and then incubated at 37°C for 3 hr. The DNA template was then degraded by adding 1 µl of TURBO DNase (Thermo Fisher) and incubating at 37°C for 15 min. The RNA transcript was purified using an RNA Clean & Concentrator-5 kit (Zymo Research) according to the manufacturer's protocol; samples were eluted in 20 µl of nuclease-free water (Fisher Bioreagents) and measured with a NanoDrop (Thermo Fisher). DID YOU CONFIRM THE RNA HAS NO OFF-TARGET BANDS?

Antisense oligonucleotides (ASOs) were ordered from Integrated DNA Technologies in a 96-well PCR plate, each ASO already resuspended to 10 µM in 1X IDTE buffer (10 mM Tris, 0.1 mM EDTA). Each ASO pool (of up to 5 ASOs) was made by mixing 2.5 µl (25 pmol) of each constituent ASO; the volume of each pool was adjusted to 12.5 µl by adding 10 mM Tris (SUPPLIER) with 0.1 mM EDTA (SUPPLIER). Each 12.5 µl ASO pool was then mixed with 1 µl (425 ng, 453 fmol) of RNA (55X molar excess of each ASO) in a PCR tube. The tubes were heated to 95°C for 60 seconds to denature the RNA, then placed into an ice block for several minutes. The RNA was transferred to 1.5 ml tubes; to each, 35 µl of 1.4X refolding buffer comprising 400 mM sodium cacodylate (Electron Microscopy Sciences) and 6 mM magnesium chloride (SUPPLIER) was added, followed by incubation at 37°C for 25 min to allow the RNA to refold and bind the ASOs. No-ASO control 1 was handled in the same manner but with 12.5 µl of TE buffer in lieu of an ASO pool. For no-ASO control 2, 12.5 µl of TE buffer was added after placing on ice and before adding refolding buffer, to confirm that the timing of adding TE buffer would not alter the RNA structure.

For chemical probing, 1.5 µl of neat DMS (MilliporeSigma) was added to each tube for a total volume of 50 µl including 320 mM DMS, 280 mM cacodylate, and 9.1 nM RNA. The DMS was mixed in by swirling the pipette tip and kept resuspended by shaking at 500 rpm in a thermomixer (Eppendorf) throughout the treatment at 37°C for 5 min. The reactions were quenched by adding 30 µl neat beta-mercaptoethanol (SUPPLIER) and mixing thoroughly. Each sample of DMS-

modified RNA was purified using an RNA Clean & Concentrator-5 kit (Zymo Research) according to the manufacturer's protocol; samples were eluted in 10 µl of nuclease-free water (Fisher Bioreagents) and measured with a NanoDrop (Thermo Fisher).

ASOs were removed from each sample using TURBO DNase (Thermo Fisher) according to the manufacturer's protocol. Briefly, 4 µl of each DMS-modified RNA was mixed with 4 µl of nuclease-free water (Fisher Bioreagents), 1 µl of 10X TURBO DNase Buffer, and 1 µl of TURBO DNase in a PCR tube; and then incubated at 37°C for 30 min. To stop each reaction, 2 µl of DNase Inactivation Reagent was mixed in and incubated at room temperature for 10 min, and mixed throughout by flicking several times. The DNase Inactivation Reagent was precipitated by spinning the tubes on a benchtop PCR tube centrifuge for 10 min, then transferring 4 µl of each supernatant to a new tube.

Correcting observer bias due to drop-out of reads

Let N reads from K clusters align to a reference sequence of length L. Let the proportion of reads whose 5' and 3' ends align, respectively, to coordinates a and b ($1 \leq a \leq b \leq L$) be η_{ab} (assuming these proportions are equal for all clusters). Let the mutation rate of base j ($1 \leq j \leq L$) in cluster k ($1 \leq k \leq K$) be μ_{jk}. Let the proportion of cluster k in the ensemble be π_k. To express these quantities as probabilities, let C_k be the event that a read comes from cluster k; let E_{ab} be the event that a read aligns with 5' and 3' coordinates a and b, respectively; let S_j be the event that a read contains position j (i.e. its alignment coordinates a and b satisfy $1 \leq a \leq j \leq b \leq L$); let M_j be the event that a read has a mutation at position j; and let G_g be the event that a read has no two mutations separated by fewer than g non-mutated bases.

Deriving mutation rates of reads with no two mutations too close

In terms of these events, the total mutation rates (μ_{jk}) are $P(M_j|S_jC_k)$, i.e. the probability that a read would have a mutation at position j given that it contained position j and came from cluster k; and the observable mutation rates (m_{jk}) are $P(M_j|S_jC_kG_g)$, i.e. the probability that a read would have a mutation at position j given that it contained position j, came from cluster k, and had no two mutations closer than g bases. Using these definitions and Bayes' theorem yields a probabilistic formula for m_{jk}:

$$m_{jk} = P(M_j|S_jC_kG_g) = P(M_j|S_jC_k)\frac{P(G_g|S_jM_jC_k)}{P(G_g|S_jC_k)} = \mu_{jk}\frac{P(G_g|S_jM_jC_k)}{P(G_g|S_jC_k)}$$

The term $P(G_g|S_jC_k)$ is the probability that a read would have no two mutations closer than g bases given that it contained position j and came from cluster k. It can be computed using $P(G_g|E_{ab}C_k)$ (abbreviated d_{abk}): the probability that a read would contain no two mutations closer than g bases given that its 5' and 3' coordinates are a and b, repectively ($1 \leq a \leq b \leq L$), and that it came from cluster k. If position b were mutated (probability μ_{bk}), then the read would contain no two mutations closer than g bases if and only if none of the g bases preceding b (i.e. positions $b-g$ to $b-1$, inclusive) were mutated (probability $\prod_{j'=\max(b-g,a)}^{b-1}(1-\mu_{j'k})$, abbreviated $w_{\max(b-g,a),b-1,k}$) and two no mutations between positions a and $b-(g+1)$, inclusive, were too close (probability $d_{a,\max(b-(g+1),a),k}$)). If position b were not mutated (probability $1-\mu_{bk}$), then the read would contain no two mutations closer than g bases if and only if no mutations between positions a and $b-1$, inclusive, were too close (probability $d_{a,\max(b-1,a),k}$). These two possibilities generate a recurrence relation:

$$d_{abk} = \mu_{bk}w_{\max(b-g,a),b-1,k}d_{a,\max(b-(g+1),a),k} + (1-\mu_{bk})d_{a,\max(b-1,a),k}$$

The base case is $d_{abk} = 1$ when $a = b$ because such a read would contain one position and thus be guaranteed to have no two mutations too close. Then, $P(G_g|S_jC_k)$ is the average of d_{abk} over every read that contains position j, weighted by the proportions η_{ab}:

$$P(G_g|S_jC_k) = \frac{\sum_{a=1}^{j}\sum_{b=j}^{L}\eta_{ab}d_{abk}}{\sum_{a=1}^{j}\sum_{b=j}^{L}\eta_{ab}}$$

The term $P(G_g|S_jM_jC_k)$ is the probability that a read would have no two mutations too close given that it contained a mutation at position j and came from cluster k. It can be computed using $P(G_g|M_jE_{ab}C_k)$ (abbreviated f_{abjk}): the probability that a read would contain no two mutations too close given that position j is mutated ($1 \leq a \leq j \leq b \leq L$), that its 5' and 3' coordinates are a and b (respectively), and that it came from cluster k. Because position j is mutated, having no two mutations too close requires that none of the g bases on both sides of position j be mutated. The probability that none of the preceding g positions ($j-g$ to $j-1$) is mutated is $w_{\max(j-g,a),j-1,k}$, while that of the following g positions ($j+1$ to $j+g$) is $w_{j+1,\min(j+g,b),k}$. Upstream of the g bases flanking position j (i.e. positions a to $j-(g+1)$), the probability that no two mutations are too close is $d_{a,\max(j-(g+1),a),k}$; downstream (i.e. positions $j+(g+1)$ to b), the probability is $d_{\min(j+(g+1),b),b,k}$. Since mutations in these four sections are independent, the probability that the read contains no two mutations too close is the product:

$$f_{abjk} = d_{a,\max(j-(g+1),a),k}w_{\max(j-g,a),j-1,k}w_{j+1,\min(j+g,b),k}d_{\min(j+(g+1),b),b,k}$$

Then, $P(G_g|S_jM_jC_k)$ is the average of f_{abjk} over every read that contains position j, weighted by the proportions η_{ab}.

$$P(G_g|S_jM_jC_k) = \frac{\sum_{a=1}^{j}\sum_{b=j}^{L}\eta_{ab}f_{abjk}}{\sum_{a=1}^{j}\sum_{b=j}^{L}\eta_{ab}}$$

Combining the above results yields an explicit formula for m_{jk}:

$$m_{jk} = \mu_{jk}\frac{\sum_{a=1}^{j}\sum_{b=j}^{L}\eta_{ab}f_{abjk}}{\sum_{a=1}^{j}\sum_{b=j}^{L}\eta_{ab}d_{abk}}$$

Deriving end coordinate proportions of reads with no two mutations too close

The total proportions (η_{ab}) of reads aligned to 5' and 3' coordinates a and b, respectively, are $P(E_{ab})$; and the proportions of reads with no two mutations too close that align with coordinates a and b (e_{abk}) are $P(E_{ab}|G_gC_k)$. Note that, while reads are assumed to come from the same distribution of coordinates (η_{ab}) regardless of their cluster k, the observable distribution of coordinates (e_{abk}) varies by cluster because $\mathrm{P}(G_gC_k)$ depends on k. Using these definitions and Bayes' theorem yields a probabilistic formula for e_{abk}:

$$e_{abk} = P(E_{ab}|G_gC_k) = P(G_g|E_{ab}C_k)\frac{P(E_{ab}|C_k)}{P(G_g|C_k)} = d_{abk}\frac{\eta_{ab}}{P(G_g|C_k)}$$

The term $P(G_g|C_k)$ is the probability that a read would have no two mutations too close given that it came from cluster k. It can be computed as an average of $P(G_g|E_{ab}C_k)$ (i.e. d_{abk}) over all coordinates a and b (such that $1 \leq a \leq b \leq L$), weighted by the proportion of each coordinate, $P(E_{ab})$ (i.e. η_{ab}):

$$P(G_g|C_k) = \frac{\sum_{a=1}^{L}\sum_{b=a}^{L}\eta_{ab}d_{abk}}{\sum_{a=1}^{L}\sum_{b=a}^{L}\eta_{ab}} = \sum_{a=1}^{L}\sum_{b=a}^{L}\eta_{ab}d_{abk}$$

This expression is already normalized because $\sum_{a=1}^{L}\sum_{b=a}^{L}\eta_{ab} = 1$, by definition.

Combining the above results yields an explicit formula for e_{abk}:

$$e_{abk} = \frac{\eta_{ab}d_{abk}}{\sum_{a'=1}^{L}\sum_{b'=a'}^{L}\eta_{a'b'}d_{a'b'k}}$$

Deriving cluster proportions of reads with no two mutations too close

The proportion of total reads in cluster k is $\pi_k = P(C_k)$. The proportion among only reads with no two mutations closer than g bases is

$$p_k = P(C_k|G_g) = P(G_g|C_k)\frac{P(C_k)}{P(G_g)} = \pi_k\frac{\sum_{a=1}^{L}\sum_{b=a}^{L}\eta_{ab}d_{abk}}{P(G_g)}$$

The term $P(G_g)$ is the probability that a read from any cluster would have no two mutations closer than g bases and can be solved for by leveraging that the cluster proportions (p_k) must sum to 1:

$$1 = \sum_{k=1}^{K} p_k = \sum_{k=1}^{K} \pi_k \frac{\sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk}}{P(G_g)} = \frac{1}{P(G_g)} \sum_{k=1}^{K} \pi_k \sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk}$$

$$P(G_g) = \sum_{k=1}^{K} \pi_k \sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk}$$

The result is an explicit formula for p_k:

$$p_k = \frac{\pi_k \sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk}}{\sum_{k'=1}^{K} \pi_{k'} \sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk'}}$$

Solving total mutation rates and cluster and coordinate proportions

The observed mutation rates (m_{jk}), end coordinate proportions (e_{abk}), and cluster proportions (p_k) can be calculated as weighted averages over the N reads with no two mutations too close:

$$m_{jk} = \frac{\sum_{i=1}^{N} z_{ik} x_{ij}}{\sum_{i=1}^{N} z_{ik}}$$

$$e_{abk} = \frac{\sum_{i=1}^{N} z_{ik} y_{abi}}{\sum_{i=1}^{N} z_{ik}}$$

$$p_k = \frac{\sum_{i=1}^{N} z_{ik}}{N}$$

where x_{ij} is 1 if read i has a mutation at position j, otherwise 0; y_{abi} is 1 if read i aligns to coordinates a and b, otherwise 0; and z_{ik} is the probability that read i came from cluster k.

The original parameters μ_{jk}, η_{abk}, and π_k can be solved by setting the two formula each for m_{jk}, e_{abk}, and p_k equal to each other, creating a system of equations:

$$\mu_{jk} \frac{\sum_{a=1}^{j} \sum_{b=j}^{L} \eta_{ab} f_{abjk}}{\sum_{a=1}^{j} \sum_{b=j}^{L} \eta_{ab} d_{abk}} = m_{jk} = \frac{\sum_{i=1}^{N} z_{ik} x_{ij}}{\sum_{i=1}^{N} z_{ik}}$$

$$\eta_{ab}\frac{d_{abk}}{\sum_{a'=1}^{L}\sum_{b'=a'}^{L}\eta_{a'b'}d_{a'b'k}} = e_{ab} = \frac{\sum_{i=1}^{N} z_{ik}y_{abi}}{\sum_{i=1}^{N} z_{ik}}$$

$$\pi_k\frac{\sum_{a=1}^{L}\sum_{b=a}^{L}\eta_{ab}d_{abk}}{\sum_{k'=1}^{K}\pi_{k'}\sum_{a=1}^{L}\sum_{b=a}^{L}\eta_{ab}d_{abk'}} = p_k = \frac{\sum_{i=1}^{N} z_{ik}}{N}$$

Solving this entire system at once has proven computationally impractical for all but extremely short sequences. A more feasible approach is to first solve for μ_{jk} given an initial guess for η_{ab}, next solve for η_{ab} given the updated μ_{jk}, then solve for π_k given the updated μ_{jk} and η_{ab}, and iterate until all three sets of parameters converge.

Even assuming every η_{ab} is a constant, these equations are still too complex to solve for μ_{jk} analytically because d_{abk} and f_{abjk} also depend on μ_{jk} (as well as on other μ variables). Thus, every μ_{jk} is solved for numerically by rearranging each equation to

$$\mu_{jk}\frac{\sum_{a=1}^{j}\sum_{b=j}^{L}\eta_{ab}f_{abjk}}{\sum_{a=1}^{j}\sum_{b=j}^{L}\eta_{ab}d_{abk}} - m_{jk} = 0$$

and applying the Netwon-Krylov method **?** implemented in SciPy **?**.

Once every μ_{jk} has been solved for, every η_{ab} can be updated. Because d_{abk} does not depend on η_{ab} (except indirectly through the μ_{jk} parameters, which are now assumed to be constants), each equation can be rearranged to

$$\eta_{ab} = \frac{e_{ab}}{d_{abk}}\sum_{a'=1}^{L}\sum_{b'=a'}^{L}\eta_{a'b'}d_{a'b'k}$$

Leveraging that $\sum_{a=1}^{L}\sum_{b=a}^{L}\eta_{ab} = 1$, by definition, leads to

$$\sum_{a=1}^{L}\sum_{b=a}^{L}\frac{e_{ab}}{d_{abk}}\sum_{a'=1}^{L}\sum_{b'=a'}^{L}\eta_{a'b'}d_{a'b'k} = 1$$

$$\sum_{a'=1}^{L}\sum_{b'=a'}^{L}\eta_{a'b'}d_{a'b'k} = \frac{1}{\sum_{a=1}^{L}\sum_{b=a}^{L}\frac{e_{ab}}{d_{abk}}}$$

and finally a closed-form expression for each η_{ab} given μ_{jk} (and hence d_{abk}) and e_{abk}:

$$\eta_{ab} = \frac{\frac{e_{ab}}{d_{abk}}}{\sum_{a'=1}^{L}\sum_{b'=a'}^{L}\frac{e_{a'b'}}{d_{a'b'k}}}$$

This equation should theoretically yield the same value of η_{ab} for every k. In practice, the values will differ due to inexactness in floating-point arithmetic. Thus, the consensus value of η_{ab} is taken to be the average η_{ab} over every k, weighted by π_k:

$$\eta_{ab} = \sum_{k=1}^{K} \pi_k \frac{\frac{e_{ab}}{d_{abk}}}{\sum_{a'=1}^{L} \sum_{b'=a'}^{L} \frac{e_{a'b'}}{d_{a'b'k}}}$$

With updated values of μ_{jk} and η_{ab}, π_k can also be solved. The above equations can be rearranged to

$$\pi_k = p_k \frac{\sum_{k'=1}^{K} \pi_{k'} \sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk'}}{\sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk}}$$

Given that $\sum_{k=1}^{K} \pi_k = 1$, by definition:

$$\sum_{k=1}^{K} p_k \frac{\sum_{k'=1}^{K} \pi_{k'} \sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk'}}{\sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk}} = 1$$

$$\sum_{k'=1}^{K} \pi_{k'} \sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk'} = \frac{1}{\sum_{k=1}^{K} \frac{p_k}{\sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk}}}$$

which leads to a closed-form expression for each π_k given μ_{jk} (and hence d_{abk}), η_{ab}, and p_k:

$$\pi_k = \frac{\frac{p_k}{\sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk}}}{\sum_{k'=1}^{K} \frac{p_{k'}}{\sum_{a=1}^{L} \sum_{b=a}^{L} \eta_{ab} d_{abk'}}}$$

Clustering reads with the expectation-maximization algorithm

Let N reads from K clusters align to a reference sequence of length L. Let the proportion of reads whose 5' and 3' ends align, respectively, to coordinates a and b ($1 \leq a \leq b \leq L$) be η_{ab} (assuming these proportions are equal for all clusters). Let the mutation rate of base j ($1 \leq j \leq L$) in cluster k ($1 \leq k \leq K$) be μ_{jk}. Let the proportion of cluster k in the ensemble be π_k.

Maximization step

The maximization step updates the parameters (μ_{jk}, η_{ab}, and π_k) using the current cluster memberships (z_{ik}). The observed estimates of the parameters m_{jk}, e_{ab}, and p_k are first computed; then, the underlying parameters μ_{jk}, η_{ab}, and π_k are solved for as described in 1.2.4.

Expectation step

The expectation step updates the cluster memberships (z_{ik}) and the likelihood function (L) using the current parameters (μ_{jk}, η_{ab}, and π_k). Each cluster membership is defined as the probability that read i came from cluster k given its 5'/3' end coordinates (E_{ab}) and mutations (M) and given that no two mutations are too close (G_g): $z_{ik} = P(C_k|E_{ab}MG_g)$. The likelihood of the model (L) is the product of the marginal probability (L_i) of observing each read i from any cluster: $L_i = P(E_{ab}M|G_g)$. Both L_i and z_{ik} can be expressed in terms of the joint probability ($L_{ik} = P(E_{ab}MC_k|G_g)$) of observing each read i from each cluster k:

$$L_i = P(E_{ab}M|G_g) = \sum_{k=1}^{K} P(E_{ab}MC_k|G_g) = \sum_{k=1}^{K} L_{ik}$$

$$z_{ik} = P(C_k|E_{ab}MG_g) = \frac{P(E_{ab}MC_kG_g)}{P(E_{ab}MG_g)} = \frac{P(E_{ab}MC_k|G_g)}{P(E_{ab}M|G_g)} = \frac{L_{ik}}{L_i}$$

To derive a formula for L_{ik}, it can be factored into three parts using the chain rule for probability:

$$L_{ik} = P(E_{ab}MC_k|G_g) = \frac{P(E_{ab}MC_kG_g)}{P(G_g)} = P(M|E_{ab}C_kG_g)P(E_{ab}|C_kG_g)P(C_k|G_g)$$

The first part – the probability that a read would have the specific mutations x_{ij} given that its 5'/3' end coordinates are a and b (respectively), it comes from cluster k, and no two mutations are too close – is the product over every position j from a to b of the probability of a mutation (μ_{jk}) if read i is mutated at position j ($x_{ij} = 1$), otherwise ($x_{ij} = 0$) the probability of no mutation ($1 - \mu_{jk}$), normalized by the

probability that no two mutations would be too close (d_{abk}):

$$P(M|E_{ab}C_kG_g) = \frac{1}{d_{abk}} \prod_{j=a}^{b} \mu_{jk}^{x_{ij}} (1-\mu_{jk})^{(1-x_{ij})}$$

The second part, $P(E_{ab}|C_kG_g) = e_{abk}$, can be calculated from the parameters μ_{jk}, η_{ab}, and π_k, as explained in 1.2.2. Likewise, the third part, $P(C_k|G_g) = p_k$, can also be calculated from the parameters, as explained in 1.2.3. Combining all parts yields a formula for L_{ik} in terms of the parameters μ_{jk}, η_{ab}, and π_k and of their derived values d_{abk}, e_{abk}, and p_k:

$$L_{ik} = p_k \frac{e_{abk}}{d_{abk}} \prod_{j=a}^{b} \mu_{jk}^{x_{ij}} (1-\mu_{jk})^{(1-x_{ij})}$$

The formula for the total likelihood of the model and its parameters follows:

$$L(\mu, \eta, \pi) = \prod_{i=1}^{N} L_i = \prod_{i=1}^{N} \sum_{k=1}^{K} p_k \frac{e_{abk}}{d_{abk}} \prod_{j=a}^{b} \mu_{jk}^{x_{ij}} (1-\mu_{jk})^{(1-x_{ij})}$$

Screening coronavirus long-range interactions computationally

All coronaviruses with reference genomes in the NCBI Reference Sequence Database **?** were searched for using the following query:

```
refseq[filter] AND ("Alphacoronavirus"[Organism] OR
                    "Betacoronavirus"[Organism] OR
                    "Gammacoronavirus"[Organism] OR
                    "Deltacoronavirus"[Organism])
```

The complete record of every reference genome was downloaded both in FASTA format (for the reference sequence) and in Feature Table format (for feature locations). The location of the frameshift stimulating element (FSE) in each genome was estimated from the feature table, and the nearest instance of `TTTAAAC` was used as the slippery site, using a custom Python script. The 2,000 nt segment beginning 100 nt upstream of and ending 1,893 nt downstream of the slippery site was

used for predicting long-range interactions involving the FSE. Genomes with ambiguous nucleotides (e.g. `N`) in this segment were discarded. For each coronavirus genome, up to 100 secondary structure models of the 2,000 nt segment were generated using Fold version 6.3 from RNAstructure **?** with `-M 100` and otherwise default parameters. Then, for each position, the fraction of models for the coronavirus in which the base at the position paired with any other base between positions 101 (the first base of the slippery sequence) and 250 was calculated using a custom Python script. The coronaviruses were clustered by their fraction vectors using the unweighted pair group method with arithmetic mean (UPGMA) and a euclidean distance metric, implemented in Seaborn version 0.11 **?** and SciPy version 1.7 **?**. The resulting hierarchically-clustered heatmap was examined manually to select coronaviruses based on the prominence of potential long-range interactions with the FSE (relatively large fractions far from positions 101-250).