Retrospective evaluation of variant forecasting models

Athanasios G. Bakis, Andrew F. Magee, Scott W. Olesen*

Center for Forecasting and Outbreak Analytics, Centers for Disease Control and Prevention
July 16, 2026

1 Key points

- Variant forecasting is potentially useful for predicting waves of infection and for informing medical countermeasure distribution.
- To avoid confusion, variant forecasters should distinguish between (1) genomic **sequences**, which are the data, (2) the **taxa** (e.g., evolutionary clades or lineages) to which sequences are assigned, and (3) the modeled **units** formed by aggregating taxa as motivated by real-life epidemiology and practical modeling constraints.
- Ensembling may be ineffective for variant forecasting because there are few extant classes of variant forecasting models. Rather than rely on a diversity of model types, modelers should consider *post hoc* model comparison to interrogate and improve the small number of extant models.
- Most variant forecasting models treat the relative prevalence of pathogen variants, not the per capita pathogen infection prevalence. This separation is acceptable for the moment but substantially reduces the utility of the resulting forecasts by decoupling variant dynamics from waves of infection.
- Variant forecasts are typically used to answer decision-oriented questions, like whether a variant will become dominant, while extant scoring metrics focus on a forecast's ability to make precise estimates day-by-day. Thus, there is a potential disconnect between the performance of models vis-à-vis traditional forecasting scores and their performance in answering the questions that variant forecasts are most often used for.

2 Introduction

2.1 Variant forecasting

Coordinated efforts to forecast disease transmission are not new to the scientific community; see, for example, the FluSight Challenge and COVID-19 Forecast Hub. However, improved disease surveillance and tooling are providing scientists with more granular viral genetic information, enabling forecasts not only of the incidence of a pathogen but also, in some cases, of the incidence of particular variants of that pathogen.

*Correspondence: ulp7@cdc.gov. The findings and conclusions in this report are those of the authors and do not necessarily represent the official position of the Centers for Disease Control and Prevention.

These types of forecasts have been performed for many pathogens, including SARS-CoV-2 [7], influenza [21], and dengue [24].

Here we use the term "variant forecasting" to narrowly refer to estimation of the future, relative prevalence of variants of a pathogen, that is, the proportion of all infections with a given pathogen that are infections with a given variant of that pathogen.

We use the term "future" to mean that the estimates are for dates after the most recent data provided to the forecasting model. This definition also encompasses what is often referred to as "nowcasting," since the forecast target dates might not be in the future relative to the time of the production of the forecast, due to long delays between sample collection and reporting. (We note that the SARS-CoV-2 Variant Nowcast Hub and the CDC [7] both use "nowcasting" to refer to what we here call "forecasting.") In infectious disease literature [6], "nowcasting" is also used to refer to forecasting that explicitly accounts for incomplete count data. Modeling incomplete counts is crucial for timely estimates of extensive quantities like numbers of hospitalizations, for which knowing the number of counts that will have been reported for a given date is the goal of forecasting. We do not address the problem of modeling incomplete counts here, in part because knowing the number of sequences that will have been reported typically does not provide meaningful information about the proportion of those sequences that will belong to each variant. (Explicitly modeling incomplete sequencing counts would be important if, for example, the delay from infection to the reporting of a sequence was systematically and substantively different for different variants.)

Note that this definition of "variant forecasting" refers only to the relative prevalence of variants, not necessarily whether there are many or few infections overall. This approach has been the standard for SARS-CoV-2; see the SARS-CoV-2 Variant Nowcast Hub.

Ideally, a variant forecasting model would take in all genomic and other surveillance data and return both per capita infection prevalence and relative variant prevalence. It might even predict where and when a new variant would evolve or emerge. While out of scope here, linking overall pathogen incidence with variant composition will be critical to models that demonstrate how the emergence of a variant can drive waves of disease.

The use of the term "variant" can lead to major confusions. "Variant," as a term of the art, refers to specific groupings, such as "variants of concern," "variants of interest," and "variants under monitoring." These officially-designated variants are typically defined for biological and epidemiological reasons. For example, the WHO defines a "variant of interest" as "a SARS-CoV-2 variant with genetic changes that are predicted or known to affect virus characteristics such as transmissibility, virulence, antibody evasion, susceptibility to therapeutics and detectability; AND identified to have a growth advantage over other circulating variants ... with increasing relative prevalence alongside increasing number of cases over time, or other apparent epidemiological impacts to suggest an emerging risk to global public health" [27]. Such variants are typically also clades, that is, monophyletic collections composed of an ancestor and all descendants.

2.2 Applications of SARS-CoV-2 variant forecasting

Variant forecasting has practical applications for SARS-CoV-2. First, rapid growth of a novel variant can be a predictive signal for a new wave of infections.

Second, the relative prevalence of variants can inform decisions about medical countermeasure distribution. For example, the monoclonal antibodies used to treat COVID-19 were not highly effective for all SARS-CoV-2 variants. To use the limited supply to greatest effect, and to ensure that treatments were effective, the FDA

authorized and revoked authorizations for particular monoclonal antibody treatments, and HHS changed its patterns of distribution of those treatments, depending on the relative prevalence of variants (Table 1). For example, FDA authorized bamlanivimab (later combined as bamlanivimab/etesevimab) to treat COVID-19 in November 2020. This was the first authorized monoclonal antibody treatment for COVID-19. In June 2021, distribution of bamlanivimab/etesevimab was paused [1] because prevalence of the Beta and Gamma variants was increasing, and bamlanivimab/etesevimab was "not active" against those variants. In September 2021, when it became clear that Beta and Gamma would not become dominant variants, distribution was restarted [2]. In October 2021, distribution to Hawaii was paused [4]. Hawaii was the only state where the relative prevalence of the Delta variant was greater than 5%, and it was suspected that bamlanivimab/etesevimab was not active against Delta. Distribution to Hawaii was restarted a few weeks later [3] when it was determined that bamlanivimab/etesevimab in fact retained activity against Delta. In January 2022, distribution ended permanently [10] because bamlanivimab/etesevimab was not active against Omicron and its descendants.

Similar patterns followed for the other monoclonal antibody treatments, whose use effectively ended with the emergence of variants against which the treatments were resistant. (Formal revocation of the emergency authorization often occurred years later.)

Treatment	Authorized	End of use
casirivimab/imdevimab (REGEN-COV)	Nov 2020[11]	Jan 2022[15]
sotrovimab	May 2021[11]	Mar 2022[14]
tixagevimab/cilgavimab (Evusheld)	Dec 2021[11]	Jan 2023[13]
bebtelovimab	Feb 2022[11]	Nov 2022[12]
pemivibart (Pemgarda)	Mar 2024[9]	(still in use)

Table 1: Dates of authorization and end of use for monoclonal antibody treatments for COVID-19.

Finally, variant forecasting could inform vaccine variant selection. Typically, the US Food and Drug Administration Vaccines and Related Biological Products Advisory Committee (VRBPAC) reviews variant circulation every year in May or June to make the formulation recommendation for September distribution.

2.3 Aggregating sequences into taxa and units

Confusingly, most variant forecasts do not forecast the prevalence of variants but rather certain *ad hoc* groups of genomic sequences. Typically, modelers will use bioinformatic software, in the form of a taxon assignment program, to assign each genomic sequence to a *taxon*. For SARS-CoV-2, there are two major taxonomic systems: Pango [23], with taxa like JN.1, and Nextstrain, with taxa like 24A. Modelers then aggregate those taxa into modeled *units*, groups of sequences that are treated as identical for purposes of the model (Figure 1).

Taxon assignment programs necessarily change over time, releasing new versions as new data are collected and new taxa are recognized (i.e., formally given names and added to the taxonomic system). Thus, most properly, one cannot speak of a taxon assignment program without reference to its version. We discuss some of the difficulties caused by ever-changing taxon assignment software in section 3.1.1.

Note that the modeled units are often not monophyletic and so are not clades *per se*. For example, a modeler might want to model Nextstrain taxon 24B (Pango taxon JN.1.11.1) separately from other sequences belonging to its ancestor 24A (JN.1). As another example, it is typical practice to aggregate multiple, disparate, low-prevalence taxa into a single "Other" unit.

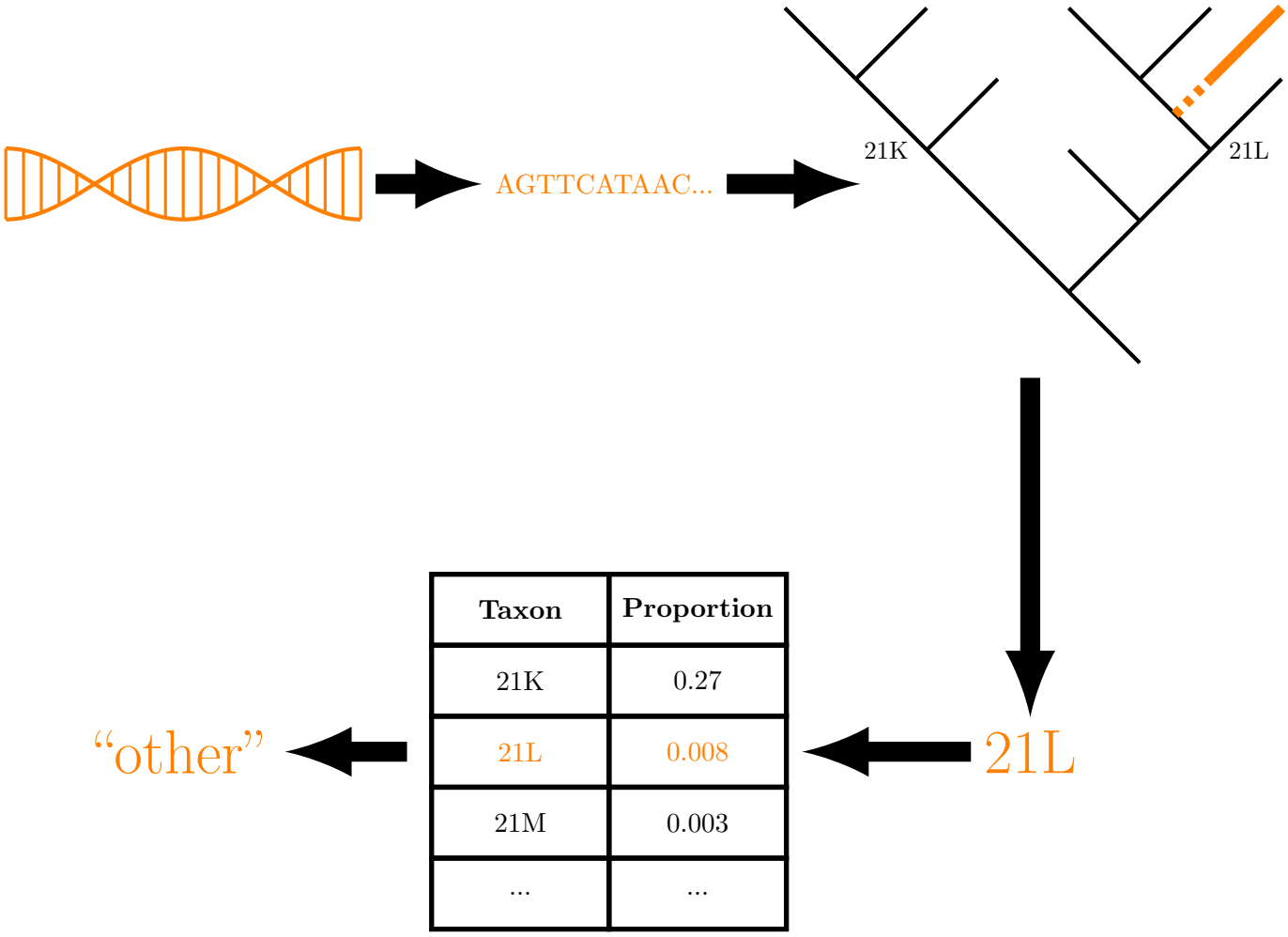

Figure 1: Grouping sequences for variant forecasting. First, viral genomic material collected from an infected individual is sequenced. Sequences are then placed on a phylogenetic tree. From the placement position, the sequence is assigned to a taxon (e.g., 21L). Then, typically using preliminary frequency data of unaggregated taxa, the modeler makes a decision about how to aggregate those taxa into units to be used in the model (e.g., "other").

We also note that taxonomy and nomenclature are intensely hand-curated processes. For example, in the context of SARS-CoV-2, taxonomy and nomenclature attempt to describe epidemiologically-relevant parts of the broader pathogen population, rather than simply biologically distinct groupings of related viruses. Consequently, variant nowcasting, as defined here, cannot predict the appearance of new taxa, not only because these models do not include pathogen evolution, but also because they are not designed to anticipate human decisions about what groupings of viruses will merit classification as new taxa.

2.4 Evaluation

For a forecasting model to provide actionable insights, we need to know how well it works. How accurate is it? Are estimates biased, and can estimation biases be reliably anticipated? Do estimates degrade in particular circumstances? Are the prediction intervals well-calibrated? These questions fall under the umbrella of *evaluation*. Absent evaluation, it is hard to know how trustworthy a forecast is and how to respond to it. In particular, we are interested in numerical and graphical means of evaluation.

Evaluating the performance of forecasts is also a crucial step in the process of iterative model development. A forecast can be evaluated on how well it fits to available training data, which we term the *training dataset*. However, the utility of a forecast lies in its ability to fill in gaps in situational awareness, predicting values which are unavailable, whether they are in the present (i.e., in a "nowcast") or the future. As such, it is typically more of interest to interrogate the performance of these predictions based on the more complete data that arrives later. We term the *evaluation dataset* to be all data available on the date evaluation is performed. The difference between the evaluation dataset and the training dataset is the out-of-sample data which was unavailable to the model when the forecast was produced as well as any data revision.

Two distinct kinds of forecasting evaluation coexist. *Prospective* evaluation is a real time process. Forecasts are produced, and when data becomes available for the dates covered by the predictions, their quality is examined. Modeling teams can, and do, react in real time to challenges that arise, adjusting components of the model or the data streams it ingests. As such, while it is a valuable tool for both model development and transparency, prospective evaluation is not an ideal approach for understanding how some particular model component affects performance in a systematic way. The SARS-CoV-2 Variant Nowcasting Hub is a recent effort to conduct prospective evaluation of variant forecasting models. Modelers submit probabilistic forecasts of unit proportions in the population, and predictions are scored according to appropriate metrics for such a target.

Retrospective evaluation is a backwards-looking process. A forecast is produced today for some *as of* date in the past, using only the data that would have been available then. Unlike prospective evaluation, retrospective evaluation requires either archived data snapshots or the ability to approximate this historically-available data, otherwise the model "cheats" by having access to overly-complete data. Retrospective evaluation can ease systematic interrogation of model performance. By putting models which are otherwise comparable head to head, a particular model component can be tested. By selecting a period of time where a known forecasting challenge applied, models can be empirically stress-tested against model violations. With appropriate care in designing the experimental setup to avoid overfitting, retrospective forecast performance is a useful proxy for how the model will perform in the future.

In this manuscript, we explore key concepts in retrospective evaluation of variant forecasting models, including challenges that arise in obtaining historical data, and demonstrate how to set up a framework to make systematic evaluation straightforward. We explore the performance of a selection of (1) multiple statistical models of increasing sophistication for each of (2) several epidemiologically significant points in COVID-19 history, using (3) a variety of evaluation metrics. We also share lessons learned about building a software

pipeline for this task, described in section A.

We note that for variant forecasting, there is an additional challenge posed by changes to taxon assignment software. We discuss this more in section 3.1.1.

3 Architecting a modular retrospective variant forecast evaluation pipeline

Our goal is to facilitate iteration on and comparison of forecasting models, even if the models belong to entirely different families. Thus, while existing approaches to variant forecasting models have focused on model-specific quantities such as growth rates [25], it is desirable to specify a common input/output format across models which is free of parameters specific to the structure of any particular model. In short, we define a variant forecasting model as something that (1) takes in a table of counts of modeling units in locations on dates, (2) takes in a target range of dates of interest, and then (3) returns samples of predictions of the true proportions of all units in all locations on all dates.

Variant forecasting models typically also assert unobserved, true proportions of infections that are associated with each modeling unit. While it is possible to envision some black-box machine learning model which does not have a notion of these true proportions, and while this model would meet the above specifications, a black box model has some design limitations. First, it is not obviously extensible. Separating the unobserved, true relative prevalences from observations allows for multiple types of observations. For example, a model initially might be designed to incorporate data only from clinical samples collected from hospitalized patients, making the incorrect but expedient assumption that those samples constitute a random sample of all infections. That model might later also incorporate data from community wastewater samples. The model's assertion of an unobserved, true relative prevalences provides a linkage between the two observed data streams.

Second, a black box model would make certain prediction problems more convoluted. For example, we might be interested in when a novel variant will first exceed 50% of infections. A black box model, having no notion of infections, could only predict the time when that variant would constitute 50% of *collected samples*. This is a noisier quantity and depends on the surveillance systems and how many samples they take. (We might also be interested in, say, the expected delay between the emergence of a variant and its first detection in some population, but that kind of question can and should be modeled separately from a variant forecast.)

We identify three components for a pipeline which can handle this in a general way:

1. generating training and evaluation tabular count data from line list infection data,
2. generating proportion predictions, and
3. evaluating predictions.

3.1 Generating tabular count data

Since our goal is to fit statistical models to the training data that would have been available on the forecast date, and then evaluate the forecasting predictions from that fit, we must handle a few complexities in data processing. First, however, we must define some key dates:

- *Forecast run date*: the date on which a forecast is produced.

- *Forecast date*: in prospective forecasting, this is the forecast run date. In retrospective forecasting, the model has access to training data that was available as of this date.
- *Event date*: the date a sample (which is later sequenced) is collected from an infected individual.
- *Report date*: the date that a sequence becomes available in a database or dataset.

Due to the time required to process (e.g., sequence) and report (submit) sequences, the *training* data (i.e., the data to which the model is fit to produce the forecast) should include only entries that were reported and included in the dataset by the forecast date, not merely sequences collected (i.e., having event dates) prior to the forecast date. As the data is in linelist form (each row corresponding to a single genetic sequence), this is achieved by filtering for rows whose report date falls before the forecast date, taking care not to accidentally filter instead on the event date. This filtering is performed only for training data and not for subsequent evaluation data, as the "future" data (relative to the forecast date) is the key to evaluating performance. Thus, during the creation of retrospective count data, two datasets must be forked from the initial input based on whether they will be used for fitting or evaluation, and then handled in parallel. In this way, we can approximate retrospectively what the SARS-CoV-2 Variant Nowcasting Hub does prospectively.

For simplicity in presentation, here we follow the SARS-CoV-2 Variant Nowcasting Hub in using Nextstrain taxa ("clades") as modeled units. While Nextstrain clades are generally large enough to preclude any intricate process of unit aggregation, we observed in practice that the presence of many units present at low frequencies in the population negatively impacts model fitting. As a result, we aggregate low-prevalence taxa into a catch-all unit named "Other." This requires manual curation or an automated cutoff, such as the one the SARS-CoV-2 Variant Nowcasting Hub uses. Separately for each analysis presented in this manuscript, we put all taxa which are not observed at a frequency of at least 1% in at least one week into the "Other" unit.

We track the first-level administrative division in which the sample was collected, which for our purposes is one of the 50 U.S. states, the District of Columbia, or Puerto Rico.

The result is a long-form table of counts per unit, division, and date.

3.1.1 The prospective relabeling problem

Evaluating a model's performance requires comparing the model output to the observed data. However, because the evaluation data is collected after the forecast data, it may include taxa that were not present in the forecast data. Novel sequences may have been observed and classified as a new taxon between the times the forecast and evaluation data, or extant sequences may have been reassigned to different taxa [23]. We refer to this as the *prospective relabeling problem*.

In practice, it is beyond the scope of most variant forecasting models to predict the emergence of new sequences or to predict human decisions to alter the existing taxonomy. Rather than penalize models for failing to anticipate novel taxa, all sequences included in both the forecast and evaluation datasets should be assigned to taxa and units according to the assignment program available as of the forecast date. The primary effect of this will be that as-yet unrecognized taxa (like 24I) will show up as their parent taxa (24A) —except, as we show in A.1.2, where recombinants are concerned— but other changes are possible given the changing accuracy of assignment over time.

Consider a prospective forecast for 2024-10-01. This is the forecast date for the "nomenclature shift" analysis in section 4.1.3. The forecast uses training data available as of 2024-10-01, and predictions extend

through 2024-10-15. The SARS-CoV-2 Variant Nowcast Hub creates the evaluation dataset on 2025-01-13, ninety days after the 2024-10-01 forecast date, using all data which has come in in that time. During the forecasting period, on 2024-10-04, clades 24F and 24G were recognized, becoming valid taxon labels with which assignment programs could label a sequence. Further new clades were named in November (24H and 24I on 2024-11-14) and December (24J on 2024-12-19). Thus, a sequence which was in the dataset on the forecast date might be assigned to a different taxon in an evaluation dataset (e.g., 24G instead of 24B). Moreover, a number of taxa will be available in data accessed during evaluation on 2025-01-13 for which there are no predictions, because they were unrecognized on the forecast date and thus could not be modeled.

The prospective relabeling problem also occurs in retrospective forecasting. Consider the same 2024-10-01 forecast date, but with a forecast run date of 2025-07-09. From the tabular count data available on the forecast run date, we make both the training dataset and the evaluation dataset. The unit assignments for both datasets should be what they would have been on the forecast date, that is, made using the assignment software available as of the forecast date. Using, for example, the assignments available on the forecast run date amounts to cheating, since this would recognize taxa and units that were obscured at the time of the forecast, unfairly avoiding the model violations discussed further below.

3.2 Generating proportion predictions

Since our focus is on probabilistic forecasting, a model must produce predictive samples of the proportions of each unit, in each division, on each date. This broad definition enables an evaluation pipeline to simultaneously score a wide range of possible models. Evaluation in this manner could resemble a forecasting Hub, where a set of models is compared; alternatively, a single model can be iterated on, with a focus on understanding how particular changes affect predictive performance. For illustration purposes, we create a small hierarchy of multinomial logistic regression models:

- a baseline model, in which each unit has a constant proportion throughout the modeling and forecasting periods;
- an independent divisions model, which uses multinomial logistic regression to model proportions of each unit over time, and which models each location separately; and
- a hierarchical divisions model, which is a partially pooled multinomial logistic regression model that shares information across locations.

Details of the mathematical and software implementation are in section A.2.

As of late July, 2025, the Variant Nowcasting Hub GitHub repository contains descriptions for six models submitted by various teams. Five are based on multinomial logistic regression; the other is a transformer-based machine learning model [8]. The multinomial family varies widely, including Bayesian and non-Bayesian approaches and different choices of hierarchical structure among divisions.

3.3 Evaluating predictions

The evaluation of variant forecasts presents a particular challenge: the key quantity of population proportions is not observable, and the primary evaluable observable of sampled unit counts is contingent on a nuisance parameter, the total number of samples collected. Consequently, we must either require that a model predict unit counts directly, or that the scoring methodology be equipped to score proportions, drawing counts as needed. The former requires that the nuisance parameter be modeled jointly with the unit proportions; this

is difficult, as it often varies for reasons unrelated to unit dynamics. We adopt the latter approach (following the SARS-CoV-2 Variant Nowcasting Hub) and consider evaluation to be contingent on the availability of the total number of samples. In practice, this means we determine the total number of samples on every division-date in the evaluation dataset, then draw predicted unit counts contingent on this quantity and the model-generated predicted proportions.

We thus define the evaluation routine as a mapping from predictive samples of unit proportions and observed unit counts, to some set of quantitative descriptors of the predictions. In this work, we use three evaluation metrics as examples of the broad classes of ways variant forecasts may be evaluated:

- *Energy score* [16]: A (strictly) proper scoring rule for multivariate forecasts which extends the continuous ranked probability score (CRPS).[1] Lower scores indicate better performance. This score is computed by first generating multinomially distributed counts from proportion predictions, then scoring these counts against the true observed counts.
- *L1-norm* of the difference between the model's population proportions of all units and the final observed fractions, for each division and date. Compared to the energy score, we opt out of performing multinomial sampling with the L1 norm, instead directly isolating performance of proportion predictions. We report the mean L1 norm, averaged over the posterior. Smaller values are better.
- *Uncovered proportion*: The proportion of units across all dates in all divisions for which the $(1-\alpha)\times 100\%$ prediction interval does not include the observed value. Like the energy score, this metric uses multinomial sampling: the predicted and observed values are counts. Unlike the other two metrics, this metric is *absolute*: the value should be $\alpha \times 100\%$, so we can tell how well a model does in isolation, and not simply relative to other models. We report uncoverage, rather than the more typical coverage, for better discrimination among (potentially small, on the absolute scale) differences in tail probabilities.

Unless otherwise noted, scores are computed for each metric and division-date, then summed within metrics across division-dates.

As an aside, the strategy of separating the model prediction and evaluation routines at unit proportions also theoretically allows for scoring approaches which leverage the mixture of parameters approach of Krüger et al. [19], a useful variance-reduction technique. However, given the multivariate discrete nature of the observation distribution, the utility may be restricted to simpler cases like scoring performance for a single unit [18].

4 Retrospective analyses: three models, three time points, three metrics

We illustrate the pipeline architecture and the axes of variation discussed in section 3 above in a small example of retrospective evaluation. Our goal is not a comprehensive and complete evaluation of model performance in the style of MacArthur et al. [20], but rather to examine how the models perform in a handful of scenarios. Specifically, we evaluate three models (see section A.2) and three evaluation metrics (see section A.3) for each of three forecast dates. Our examples are:

1. *Early 21L*; forecast date of February 14, 2022; examines how models perform early in a relatively typical wave, namely, Omicron 21L (BA.2);

[1] "A scoring rule is proper if the forecaster maximizes the expected score for an observation drawn from the distribution F if he or she issues the probabilistic forecast F, rather than $G \neq F$." Gneiting and Raftery [16]

2. *Variant soup*; forecast date June 1, 2023; examines how models perform in the absence of a rapid takeover (in progress or impending); and
3. *Nomenclature shift*; forecast date October 1, 2024; examines how model misspecification due to an impending change to taxonomy affects performance.

We ran all three models using Nextstrain open metadata accessed on 2025-07-09 (and using UShER metadata accessed on the same date for unit recoding; see section 3.1.1). In the following sections, we first discuss these scenarios in more detail and look at performance visually in two example states. Finally, we examine aggregated results.

4.1 Example fits

4.1.1 Early 21L

For our first example, we consider a forecast date of 2022-02-14, which was a point early in the Nextstrain clade 21L (BA.2) wave. The dynamics prior to and after this point were relatively typical of much of the history of SARS-CoV-2, with one "dominant" unit (Nextstrain clade 21K, Pango lineage BA.1) at high frequency being replaced by another (21L, BA.2).

Rather than attempting to visualize results for all divisions, we show only Arizona, as an example of (low to) moderate sampling intensity, and California, as an example of high sampling intensity. While the predictions of the baseline model are not particularly interesting, comparing the independent divisions model to the hierarchical divisions model is more enlightening. While the predictions for California do not change much between the two, pooling greatly increases the qualitative accuracy of the forecast for Arizona, correctly predicting that 21K will decrease while 21L will increase, a pattern the independent divisions model misses entirely (Figure 2). Situations like this are the motivation for partially-pooled models: predictions in regions with sparse data can be improved by leveraging overall patterns which are visible in aggregate (or in regions with more data).

4.1.2 Variant soup

For our second example, we consider a forecast date of 2023-06-01. After Nextstrain clade 23A (XBB.1.5) crossed below 50% prevalence in mid 2023, no single other clade exceeded 50% until 24A (JN.1) in early 2024. Thus, forecasting during this period of "variant soup" can be considered somewhat more of a stress-test of multinomial regression-style variant forecasting models, in which one unit will always become dominant (over a sufficiently long period). In this regime, we see much less difference in predictive performance between the models than for the early 21L example (Figure 3).

4.1.3 Nomenclature shift

Our third example uses a forecast date of 2024-10-01. As with the variant soup example, this is not an Omicron- or Delta-like period with one lineage sweeping to very high frequency, though Nextstrain clade 24E (KP.3.1.1) did exceed 50% frequency at its eventual peak. Unlike the variant soup example, however, this forecast date encompasses a change in nomenclature. During the forecast period, Nextstrain named the clades 24F (XEC) and 24G (KP.2.3), which were unrecognized as of the forecast date. As such, in the data

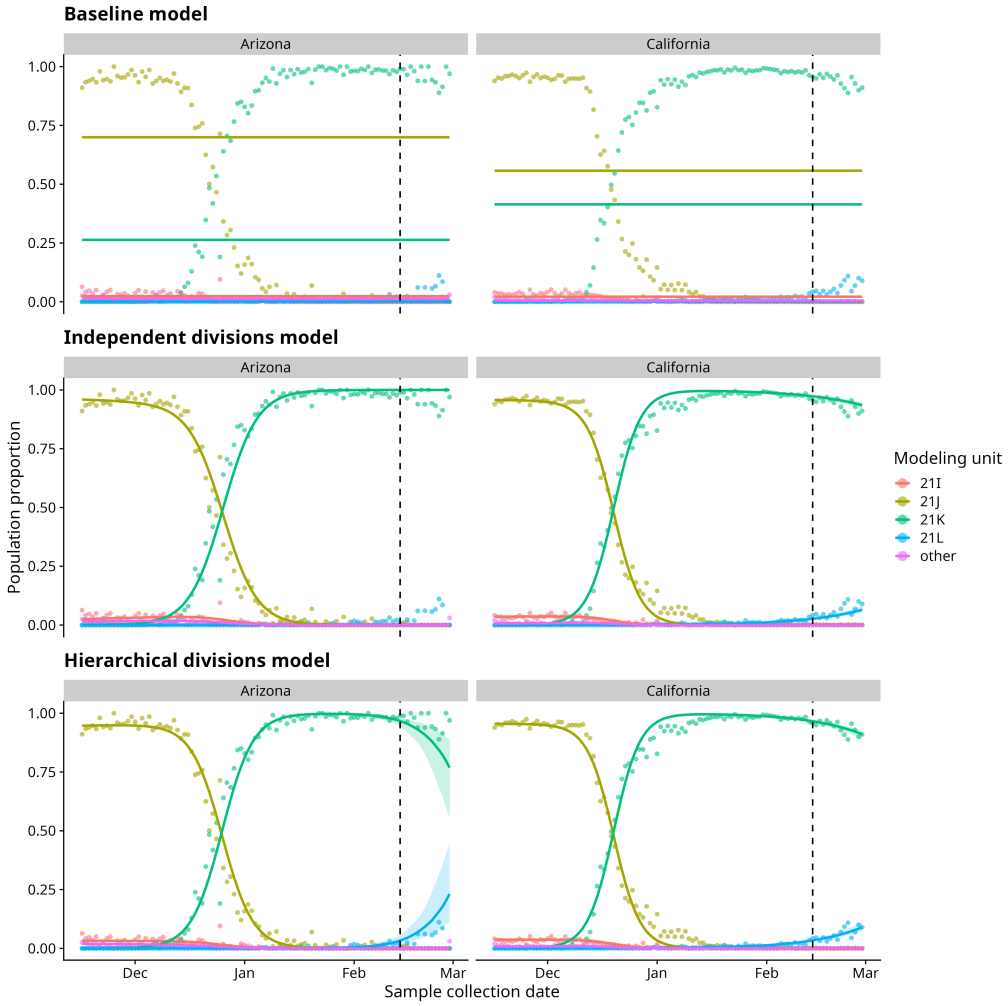

Figure 2: Data and modeling results for the early 21L scenario. Each row shows the evaluation data (circles; common across rows), the posterior median (solid line), posterior 50% credible interval (shaded region), and forecast date (vertical dashed line).

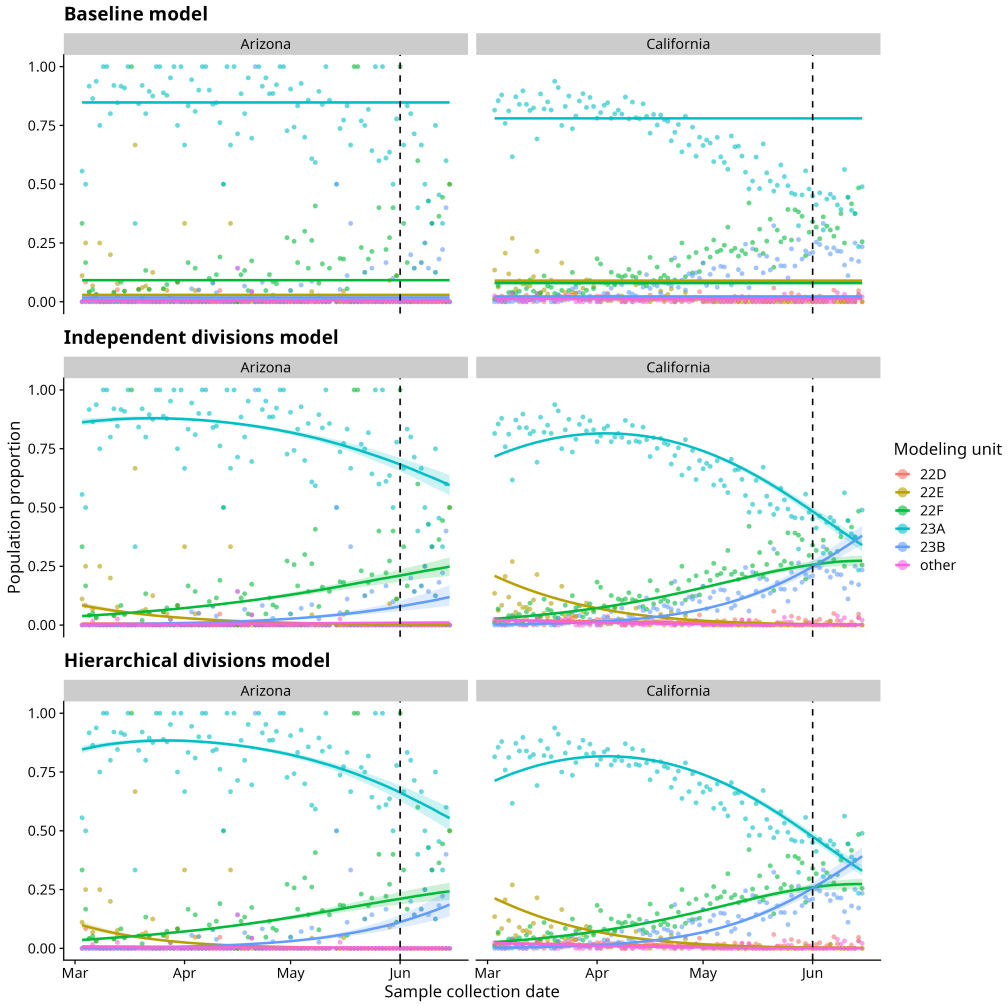

Figure 3: Data and modeling results for the variant soup scenario. Each row shows the evaluation data (circles; common across rows), the posterior median (solid line), posterior 50% credible interval (shaded region), and forecast date (vertical dashed line).

used here, samples that would later be labeled 24F and 24G were instead labeled as other units. (For more details on how this works in practice, see section A.1.1.) This scenario represents another sort of potential stress-test for models: the model treats all sequences in a unit as identical, but any parent lineage with an as-yet unsplit descendant is instead a mixture.

In this example, we see again that the hierarchical partial pooling benefits predictions in Arizona (Figure 4). The independent divisions model predicts a rather implausible rapid decrease in 24E and an even more implausible takeover by the grab-bag "other" unit.

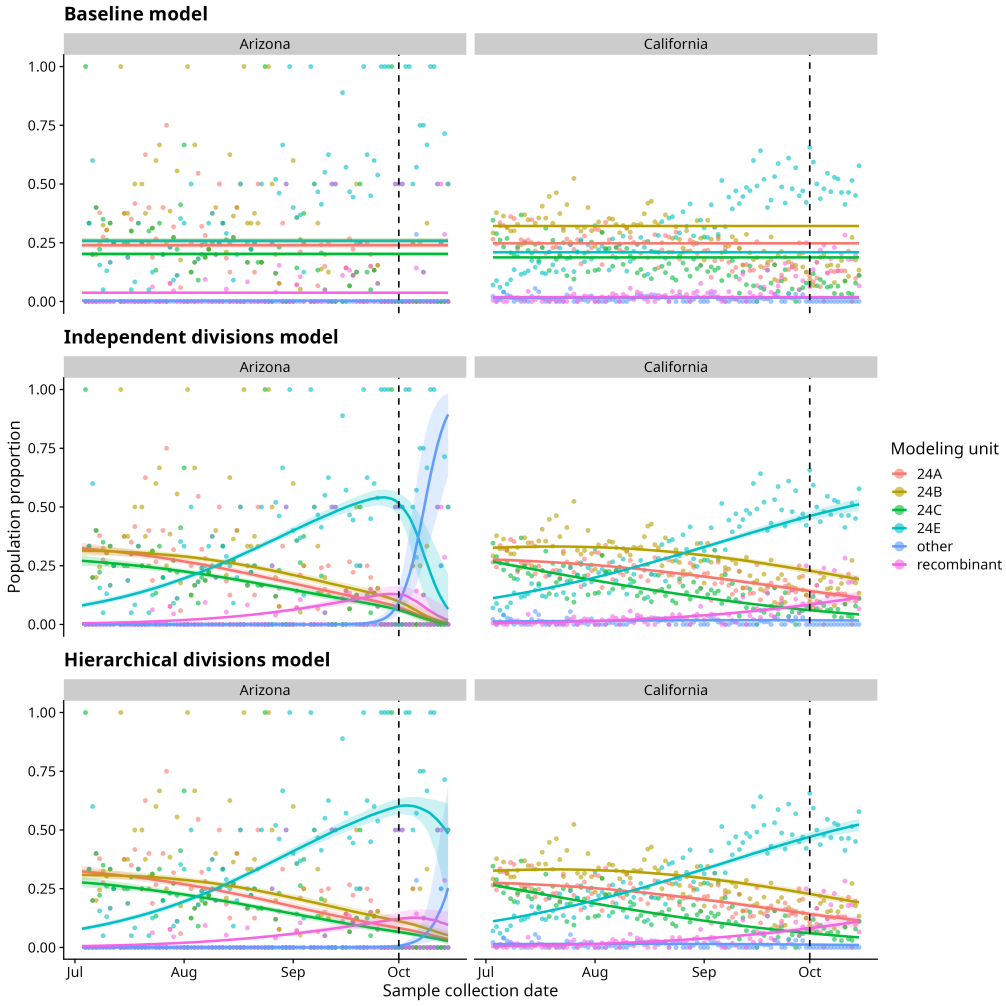

Figure 4: Data and modeling results for the nomenclature shift scenario. Each row shows the evaluation data (circles; common across rows), the posterior median (solid line), posterior 50% credible interval (shaded region), and forecast date (vertical dashed line).

4.1.4 Scoring the forecasts

The above examples are useful ways to understand how a specific model performs for a particular forecast date in a handful of locations. However, it is also useful to summarize a model's performance into a single number. Earlier, we proposed three evaluation metrics to use for scoring forecasts. We now examine the behaviors of these three metrics, aggregated over time and location, to understand how they inform us broadly of model performance (Figure 5).

One standing question is whether scores ought to be restricted or partitioned based on data completeness: should only days after the forecast date be scored, or might scores include periods with observed data? A related question is whether there is a relationship between model performance in-sample and out-of-sample: is a model's score over the training data predictive of its score in the future data? As a first pass at such questions, we consider two scores for each model, one based on all division-dates, and one based on only division-dates after the forecast date. For both settings, energy scores and L1 norms are summed over division-dates, then relative scores for each model are computed with respect to the baseline model's scores. Uncovered proportions are taken across all relevant division-dates, but not made relative to any baseline.

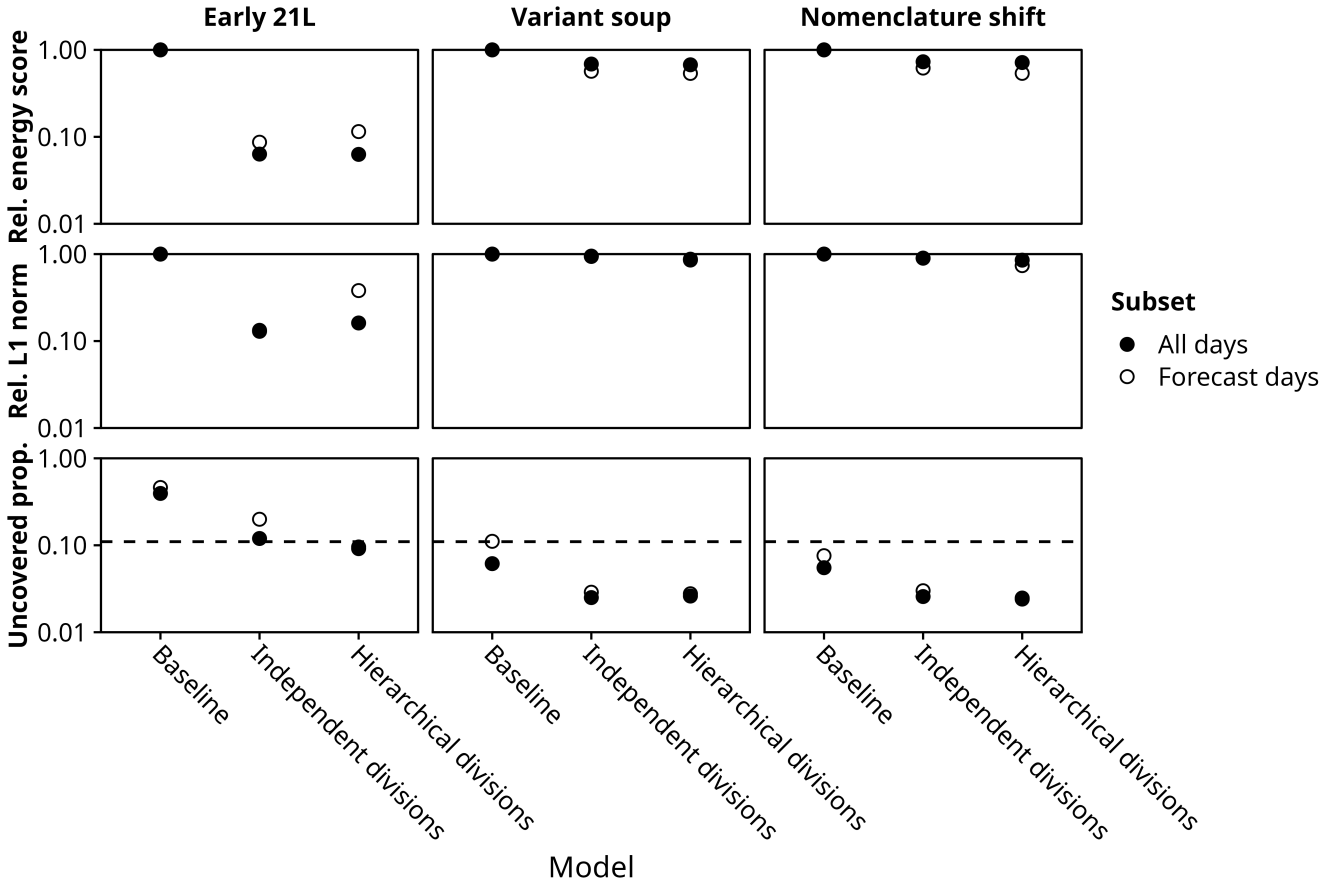

Figure 5: Values for all evaluation scenarios (columns), metrics (rows), and models (x-axis), for all division-dates (solid points) and all divisions but only dates in the forecasting period (open points). Energy scores and L1 norms are reported relative to the baseline model for interpretability. (The relative scores for the baseline model are therefore 1 by definition.) Smaller energy scores and L1 norms indicate better performance of the models. The ideal uncovered proportion is 11% (dashed line), given that 89% credible intervals are used.

There are likely multiple drivers of the differing performance of the models across scenarios. For example, the relative energy score and relative L1 norm are affected by the performance of the baseline model, which is particularly poor in the Early 21L scenario (Figure 2).

As another example, the relative energy score and uncovered proportion could be affected by data availability. Of our three scenarios, the ones that occur later in time have smaller total numbers of sequences collected, with an average of 3,943 sequences per date in the training dataset (across all divisions) in the 21L example, 466 in the variant soup example, and 323 in the nomenclature shift scenario. The relative energy scores for the independent and hierarchical divisions models are higher in the scenarios with less data, suggesting

that modeling becomes harder. That the relative L1 norm is smaller for the nomenclature shift scenario than the variant soup example may suggest that, in some sense, it is a harder modeling task, despite the nomenclature shift scenario's clear model violations. Concurrently, the uncovered proportion decreases, likely because smaller sample sizes lead to relatively wider prediction intervals. While this can have a positive effect on model performance if the intervals were otherwise too small, it can be detrimental if the resulting intervals are too wide. In the variant soup and nomenclature shift scenarios, the independent divisions model and hierarchical divisions model are substantially underconfident, with overwide intervals, that is, undershooting the target uncovered proportion.

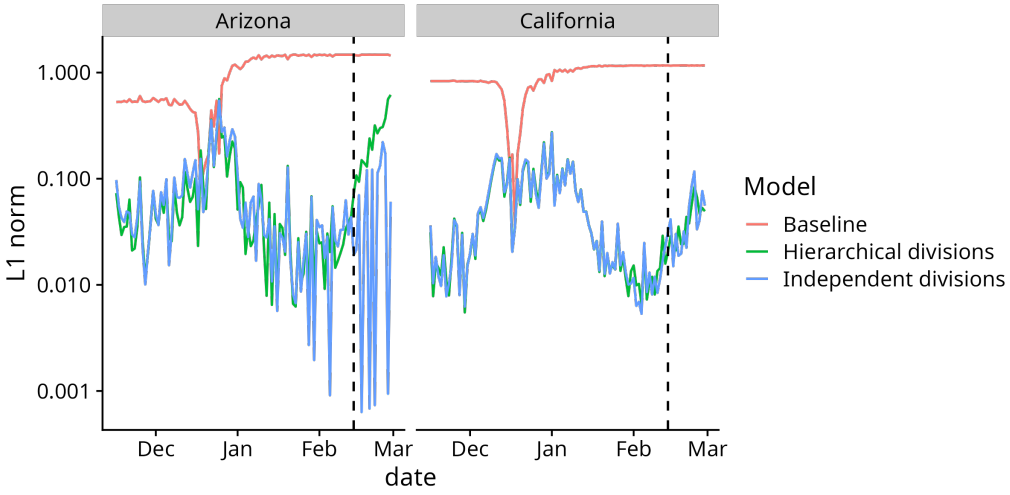

Figure 6: L1 norms for all three models for Arizona and California over time in the Early 21L scenario. The y-axis is log-scale. The spikes where the independent divisions model scores best are four dates in the evaluation period where the observed frequency of 21K was 1.0 (with sample counts ranging from 17 to 70).

It is important to remember that aggregating performance metrics across all divisions and dates can hide notable variability across divisions and dates. In Figure 6, we plot the L1 norm over time for Arizona and California for one of the example scenarios. Both the hierarchical and independent divisions models, which score better in aggregation than the baseline model, perform comparable to the baseline when the most rapid change is occurring. The baseline model does best in this range because it models only the average, which is approached (then passed) during the sweep.

5 Discussion

Here we examined how one might architect a framework for retrospective evaluation of variant forecasts, and showed how this can be used to understand model performance in multiple retrospective analyses using a variety of metrics. By defining model-agnostic, largely parallel model input and output formats, many different models of different families can be included. Despite the contingent nature of evaluation for variant forecasts (depending on the unknown number of samples to be collected on future dates), it is possible to make a similarly broad interface to evaluation routines, enabling the use of forecast scores (strictly proper or otherwise) as well as relative and absolute error metrics for evaluation. We note that, due to the shared challenges and related goals, the architecture discussed in this manuscript is similar in many ways to the SARS-CoV-2 Variant Nowcasting Hub.

We implemented three variant forecasting models, including two which share no information between regions and one that does. Based on the relative performance of the hierarchical model, we conclude that partial

pooling can be useful for predictions in data-poor regions but is not a silver bullet. More careful tuning of the strength of pooling could improve performance of such a model. Ideally, the strength of pooling would be an inferred model parameter, but by using a sufficiently broad set of retrospective analyses it could also be manually tuned. Indeed, in theory, an appropriate parameterization would span the independent divisions model and a model with complete pooling. However, in practice we found that the success of such models is not guaranteed. When we attempted to parameterize the strength of pooling directly, for both interpretability and ease of controlling the prior variance on slopes, the result appeared to be a posterior geometry sufficiently challenging for our posterior inference method (the No-U-Turn Sampler) that convergence was problematic. On the other hand, less-constrained parameterizations make controlling the prior distribution on slopes challenging, yielding prior predictive distributions which enable implausibly rapid sweeps.

We employed three evaluation metrics, including a strictly proper scoring rule, a relative error measure, and a measure of calibration (which can be interpreted in an absolute manner). The energy score, as a scoring rule, compares observables, in particular scoring counts drawn from a multinomial likelihood which incorporates (sometimes notable) sampling-based uncertainty. As a contrast, we chose our relative error metric, the L1 norm of proportion error, to focus on only posterior variation in the population proportions. That these two metrics do not always agree may reflect this difference, or it may indicate that the 1,000 posterior samples we used to estimate the scores is not sufficient to account for Monte Carlo error. Given the additional stochasticity induced by count-based evaluation, and the need to cover 5- and 6-dimensional simplices of proportions at each time step, this may not be entirely sufficient to accurately rank-order the hierarchical and independent divisions models in all cases. Where these first two metrics are relative, and only suitable for comparing between models, the uncovered proportion is an interpretable absolute metric of model adequacy. It revealed that for the later two example scenarios, the 89% credible intervals are severely conservative. It also suggested, interestingly, that in- and out-of-sample performance of the hierarchical divisions model might be more consistent than for the independent divisions model, as the uncovered proportion changes much less for the former than the latter.

We saw that the performance of different models is a nuanced issue, where different metrics can lead to different conclusions, and where looking at only overall scores can hide important information. Default approaches to comparing models, such as looking at overall metrics which equally weight all locations and time points, may not be appropriate for assessing model performance in useful ways. Perhaps more importantly, one should choose an evaluation metric which reflects what one wants the model to do. None of the metrics we examined, for example, address the suitability of the models for common variant-forecasting-adjacent questions: will this new and rare lineage come to predominate, and when? Consider that, in the Early 21L scenario, the independent divisions model completely misses the rise of 21L in Arizona (Figure 2). This qualitative failure would certainly contribute negatively to an overall summary score for that model, but there is no guarantee that a summary metric would reflect the models' performance with respect to any particular qualitative feature of interest.

We examined performance over a range of forecast dates selected to provide a variety of different lineage dynamics. This highlights that the difference between models in predictive performance is not a fixed quantity. The best model can change as the dynamics of lineage evolution change, and likely reflects at least in part the relative strength of model violations over time. For example, all our models assume constant growth rates, which is a simplification, but it is likely more wrong at some times than others, such as in the nomenclature shift scenario.

The analyses presented here were intended to be illustrative rather than exhaustive comparisons of model performance, and they should not be interpreted as such. First, only a single interval (89%) was used for coverage analysis. Second, only a single training window size was used, which limits the performance and interpretability of the baseline model.

Retrospective evaluation of variant forecasts is a challenging, but useful, exercise. By using the past as a guide, we can understand how different models perform in different circumstances, providing important context for interpreting real-time forecasting results. Further, we can iterate on model development rapidly and without having to wait weeks or months for sufficient data to come in to score prospective forecasts. In this way, we can either attempt to build more generally-useful models, or hone purpose-specific ones as needed.

A Architecture of `linmod`

Here we revisit the overall architecture described in section 3 above, discussing the specific approaches used in the Python package we built for this exploration, called `linmod`, available at https://doi.org/10.5281/zenodo.21396664.

A.1 Generating tabular count data in `linmod`

Following SARS-CoV-2 Variant Nowcasting Hub practices, the raw data used for our analyses are the metadata from the "open" Nextstrain SARS-CoV-2 workflow, providing information about sampled sequences of SARS-CoV-2 lineages. Given a forecast date, we process this data to create a table with columns:

- `date`, the sequence collection date, *i.e.* the event date;
- `fd_offset`, the number of days `date` occurs after the forecast date (a negative quantity if it occurs before), computed for convenience;
- `date_submitted`, the date on which the sequence for the collected sample was submitted to an online repository (namely, GenBank), *i.e.* the report date;
- `division`,
- `unit`, the name of the modeling unit, either the assigned Nextstrain clade or "other"; and
- `count`, the number of times the sequence was observed for the given tuple `(date, division, unit)`.

We keep entries whose collection `date` falls within 90 days before the forecast date (we term this period the model fitting period) or 14 days after the forecast date (the forecasting period). The result is a table like Table A.1.

date	fd_offset	date_submitted	division	unit	count
2024-05-07	-12	2024-05-10	Arizona	24A	1
2024-05-04	-15	2024-05-08	Pennsylvania	24A	2
...	...	...	...	...	...

There are a few notes on the data structure worth making. Perhaps most importantly, there are actually two tables for the two datasets. The training dataset contains only sequences available on the `forecast_date`, while the evaluation dataset contains all sequences in the raw data with collection dates in the fitting and forecast periods. Secondly, despite a fixed target set of divisions to forecast (here, all states, D.C., and Puerto Rico), the actual divisions seen in the data depend on the sampling intensities during the modeling and forecasting periods. We retain, in both datasets, any location which has evaluation data, as these are potentially scorable, and remove any location which does not, as it cannot be scored. This allows a modular decision on what level of input data is too low to produce useful forecasts.

A.1.1 Approximating historical unit assignments in `linmod`

The SARS-CoV-2 Variant Nowcasting Hub solves the prospective relabeling problem (see Section 3.1.1) using the utility Cladetime, which calls the Nextstrain lineage assignment pipeline on the evaluation data using the appropriate version to match the training data. However, changes in archiving over time limit

Cladetime's retrospective use to roughly May 2023 at the earliest extent, precluding its use for retrospective evaluation in the more distant past, such as our variant soup and early 21L analyses. On this basis, we adopt a procedure to *approximate* the data which would be used for a correct retrospective analysis (i.e., one which used Cladetime).

1. Start with a current Nextstrain dataset
2. Replace the sequence-level (line list) taxon labels using an archived dataset from `forecast_date` + 168 days
3. For all units which were not recognized as of the `forecast_date` replace the label with the most recent parent clade with a name which was recognized

We use the resulting line-list data to create both the modeling and evaluation datasets. The choice of 168 days represents a compromise between allowing sufficient completeness of the data during the evaluation period, while keeping the version of the assignment program from being too far out of sync with the forecast date. (As mentioned, accuracy of the assignment process varies over time.) In practice, for Step 2 we use UShER metadata [26] which is available monthly back to May 2021 (Nextstrain metadata is only archived back to September 2022). UShER archiving only stores data for the first of the month, so we use the most recent available UShER metadata which is no more recent than `forecast_date` + 168 days. For Step 3, we developed the Python package `cladecombiner`, described in section B.

Any sequence which cannot be re-coded according to the UShER metadata is discarded. This can result in a roughly 10% reduction in available data (the exact figure depends on the retrospective forecast date). However, we implement these steps modularly, such that Step 2 can be bypassed, using only the parental-lineage approximation of Step 3. Or, to understand the impact of "cheating," one can disable the process entirely.

If one wishes to only do retrospective evaluation of time periods where no data is needed prior to Cladetime's upper limit (May 2023), use of Cladetime should be preferred to this approximation. We use the approximation in all cases for consistency.

A.1.2 Recombinants

In general, when approximating historical unit assignment, one expects to find that a lineage is labeled as its parent. In the case of the nomenclature shift scenario, that means 24B for 24G. That is, lineages are more or less subtrees of the phylogeny of SARS-CoV-2, and 24G is a subtree within the subtree 24B. Indeed, examining the approximated historical taxon assignments of samples for the nomenclature shift scenario, we see that all sequences now labeled 24G were recoded as 24B.

Matters can become more complex when recombination comes into play and the underlying evolutionary history can no longer, in its entirety, be well-modeled by a phylogeny. Pango taxonomy notes that XEC, the (essentially) equivalent taxon to 24F, is a recombinant of KS.1.1 and KP.3.3. Neither of these Pango lineages has direct equivalents in Nextstrain clades: KS.1.1 falls into 24A, and KP.3.3 into 24C. The 24F genome is nominally a more than 2:1 ratio of 24A to 24C, so we might expect 24F to show up as 24A (which is the parent taxon Nextstrain lists), which is indeed the case in the nomenclature shift scenario.

A.2 Generating predictions in `linmod`

We denote the population proportion of unit ℓ at time t (`fd_offset`) in division g as $\phi_{\ell tg}$, and the proportions of all units as the vector $\boldsymbol{\phi}_{tg}$. We observe count $y_{\ell tg}$ of that unit at that time in that division (vector form $\mathbf{y}_{tg}$). All vectors have dimension L, the number of units being tracked in the data. It is the discrepancy between the key quantity for using a variant forecast being parameter $\boldsymbol{\phi}_{tg}$ and the primary evaluable quantity being $\mathbf{y}_{tg}$ that makes evaluating variant forecasts somewhat trickier than evaluating some other kinds of forecasts.

We now specify a tabular format for probabilistic forecasts of these unit proportions (Table A.2), similar in structure to the model input table (Table A.1). Columns `fd_offset`, `division`, and `unit` are carried directly from the input table. A new column `sample_index` indexes a probabilistic sample from the forecast distribution of the given unit proportion for the given date-division; thus, `fd_offset`, `division`, `unit`, and `sample_index` together uniquely identify each row. Finally, column `phi` contains the sampled proportion value.

fd_offset	division	unit	sample_index	phi
-30	Alabama	22B	1	0.000014979599
-30	Alabama	22B	2	9.945703e-7
...	...	...	...	...

As an example of iterative model development and comparison, we build a ladder of multinomial logistic regression models, each step increasing in complexity. Inference for each model is conducted using No-U-Turn Sampling (NUTS) [17] implemented in the `numpyro` software package [22, 5].

When producing forecasts, one must make a decision on when the amount of data included is insufficient for useful predictions. All currently implemented models in `linmod` are Bayesian and, as such, capable of producing a predictive distribution with or without available training data. Therefore, and for the sake of simplicity, the default is to produce forecasts for all regions where evaluation data is available.

A.2.1 Baseline model

This is a model assuming independence between divisions and specifying a constant proportion over time for each division-unit.

The parameters of this model are $\tilde{\phi}_g$, an L-dimensional vector of logit-scale population-level unit proportions, for each g. Note that the actual proportions live on an $(L-1)$-simplex, and as such our model specification has one extra dimension. The prior ensures identifiability of the posterior, and in practice we find that likelihood ridges are not seriously problematic for the NUTS sampler here. The model specification is:

$$\begin{aligned}
\tilde{\boldsymbol{\phi}}_g &\sim \mathcal{N}_L(\mathbf{0}, \sigma\mathbf{I}) \\
\boldsymbol{\phi}_g &:= \text{softmax}\left(\tilde{\boldsymbol{\phi}}_g\right) \\
\mathbf{y}_{tg}|\boldsymbol{\phi}_g &\sim \text{Multinomial}(N_{tg}, \boldsymbol{\phi}_g)
\end{aligned}$$

where $X \sim D$ indicates that random variable X is distributed according to the distribution D, $\mathcal{N}_L(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the normal distribution whose mean is the length-L vector $\boldsymbol{\mu}$ and whose covariance matrix is the $L \times L$ matrix $\boldsymbol{\Sigma}$, $\text{Multinomial}(n, \mathbf{p})$ is the multinomial distribution with size n and probabilities $\mathbf{p}$, and

$$(\text{softmax}\,(\mathbf{x}))_i = \frac{\exp(x_i)}{\sum_j \exp(x_j)}$$

In practice, we set $\sigma = 1$.

A.2.2 Independent divisions model

This is a model assuming independence between divisions and specifying an intercept and slope on time for each division-unit.

The parameters of this model are $\boldsymbol{\beta}_{0,g}$ and $\boldsymbol{\beta}_{1,g}$, the L-dimensional vectors of intercepts and slopes used to create the vector of population-level unit logits, for each g. In addition to the extra dimension in the intercepts here (as in the baseline model), the slopes also naturally live in $L-1$ dimensions. While the symmetry of this model specification (lacking a reference category) is attractive, likelihood ridges for the slopes can make sampling difficult without appropriate tuning of the mass matrix.

The model specification is as follows:

$$\begin{aligned}
\boldsymbol{\beta}_{0,g} &\sim \mathcal{N}_L(\mathbf{0}, \sigma_0 \mathbf{I}) \\
\boldsymbol{\beta}_{1,g} &\sim \mathcal{N}_L\left(\mathbf{0}, \sigma_1 \mathbf{I}\right) \\
\boldsymbol{\phi}_{tg} &:= \text{softmax}(\boldsymbol{\beta}_{0,g} + \boldsymbol{\beta}_{1,g} t) \\
\mathbf{y}_{tg} | \boldsymbol{\phi}_{tg} &\sim \text{Multinomial}(N_{tg}, \boldsymbol{\phi}_{tg})
\end{aligned}$$

In practice, we set $\sigma_0 = 1$ and $\sigma_1 = 1/16$. As such, the prior on the intercepts $\Pr(\boldsymbol{\beta}_{0,g})$ has the same covariance matrix as the prior $\Pr(\tilde{\boldsymbol{\phi}}_g)$ on the unchanging unit proportions in the baseline model.

A.2.3 Hierarchical divisions model

This is a model specifying an intercept and slope on time for each division-unit, with information sharing (partial pooling) over divisions. Thus, this model exerts a degree of regularization which tends to bring the slopes and intercepts for each division closer to an overall average term, but which allows for among-division variation in a data-driven fashion.

The parameters of this model are the vectors $\boldsymbol{\beta}_{0,g}$ and $\boldsymbol{\beta}_{1,g}$ from the independent divisions model as well as their corresponding L-dimensional mean vectors $\boldsymbol{\mu}_{\boldsymbol{\beta}_0}$ and $\boldsymbol{\mu}_{\boldsymbol{\beta}_1}$. As with the independent divisions model, the extra dimensionality requires appropriate mass matrix tuning [17].

The model specification is as follows:

$$
\begin{aligned}
\boldsymbol{\mu}_{\boldsymbol{\beta}_0} &\sim \mathcal{N}_L\left(\mathbf{0}, \sigma_{0,\boldsymbol{\mu}}\mathbf{I}\right) \\
\boldsymbol{\mu}_{\boldsymbol{\beta}_1} &\sim \mathcal{N}_L\left(\mathbf{0}, \sigma_{1,\boldsymbol{\mu}}\mathbf{I}\right) \\
\boldsymbol{\beta}_{0,g}|\boldsymbol{\mu}_{\boldsymbol{\beta}_0} &\sim \mathcal{N}_L\left(\boldsymbol{\mu}_{\boldsymbol{\beta}_0}, \sigma_{0,\boldsymbol{\beta}}\mathbf{I}\right) \\
\boldsymbol{\beta}_{1,g}|\boldsymbol{\mu}_{\boldsymbol{\beta}_1} &\sim \mathcal{N}_L\left(\boldsymbol{\mu}_{\boldsymbol{\beta}_1}, \sigma_{1,\boldsymbol{\beta}}\mathbf{I}\right) \\
\boldsymbol{\phi}_{tg} &:= \text{softmax}(\boldsymbol{\beta}_{0,g} + \boldsymbol{\beta}_{1,g}t) \\
\mathbf{y}_{tg}|\boldsymbol{\phi}_{tg} &\sim \text{Multinomial}(N_{tg}, \boldsymbol{\phi}_{tg})
\end{aligned}
$$

In practice, we set $\sigma_{0,\mu} = 1/2$ and $\sigma_{0,\beta} = 1/2$, which means that the prior $\Pr(\boldsymbol{\beta}_{0,g})$ has, marginalizing out $\boldsymbol{\mu}_{\boldsymbol{\beta}_0}$, the same covariance matrix as it does in the independent divisions model. Similarly, we set $\sigma_{1,\mu} = 3/16$ and $\sigma_{1,\beta} = 1/16$, matching the marginal covariance matrix in the prior $\Pr(\boldsymbol{\beta}_{1,g})$ to the independent divisions model. In this way, the hierarchical divisions model cannot outperform the independent divisions model simply because one has a better (implicit) prior on the (marginal) variance of the intercepts or slopes.

A.3 Evaluating predictions in `linmod`

For each model to be evaluated, we collect posterior samples $\boldsymbol{\phi}_{tg}^{(i)}$, where i denotes the `sample_index`. Both because $\boldsymbol{\phi}_{tg}$ is not observed, and because the sampling distribution of observables depends on the count of observed sequences, this probabilistic object of proportions is nontrivial to score. We therefore consider scoring it directly versus scoring samples from the posterior predictive distribution for observed counts. We note that the latter of these approaches is what has been adopted by the SARS-CoV-2 Variant Nowcasting Hub.

A.3.1 Proportion-based evaluation

One metric we consider is the expected L1-norm of proportion error, summed over all divisions and dates after the forecast date (i.e. `fd_offset` > 0):

$$
\sum_{t,g} \mathbb{E}||\mathbf{f}_{tg} - \boldsymbol{\phi}_{tg}||_1
$$

where $f_{\ell tg} = y_{\ell tg}/N_{tg}$ is the sample fraction of each unit and the expectation is taken over all samples of $\boldsymbol{\phi}_{tg}$ in the probabilistic forecast.

Note that division-dates where no units were sampled are not included in these sums, as the empirical proportions $\mathbf{f}_{tg}$ cannot be computed.

A.3.2 Count-based evaluation

Since we have access to the observed N_{tg} for dates after the forecast date, we can run an evaluation procedure in which forecasted unit *counts* are generated for each posterior sample of unit proportions. We can then score these counts in a similar fashion to the proportions themselves.

Let $\hat{y}_{tg}^{(i)} \sim \text{Multinomial}(N_{tg}, \phi_{tg}^{(i)})$ be these forecasted counts.

We consider the summed energy score:

$$\sum_{t,g} \left(\mathbb{E}||\mathbf{y}_{tg} - \hat{\mathbf{y}}_{tg}||_2 - \frac{1}{2}\mathbb{E}||\hat{\mathbf{y}}_{tg} - \hat{\mathbf{y}}'_{tg}||_2 \right)$$

In the latter expectation, $\hat{\boldsymbol{y}}'_{tg}$ is considered an independent and identically-distributed replicate of $\boldsymbol{\phi}_{tg}$.

We also consider the proportion of unit-division-dates for which the $(1-\alpha) \times 100\%$ prediction interval does not cover the observed count. Letting $q(\hat{\mathbf{y}}_{tg}, p)$ be the $p \times 100\%$ quantile of the samples $\hat{y}_{tg}^{(1)}, \hat{y}_{tg}^{(2)}, \ldots$, $\mathbb{I}$ the indicator function, and L, T, and G the number of units, evaluated time points, and divisions respectively, then this metric is:

$$\frac{1}{LTG} \sum_{\ell,t,g} \left[1 - \mathbb{I}\left(q\left(\hat{\mathbf{y}}_{\ell tg}, \frac{\alpha}{2}\right) < y_{\ell tg} < q\left(\hat{\mathbf{y}}_{\ell tg}, 1 - \frac{\alpha}{2}\right)\right)\right]$$

B Portable and reproducible taxon aggregation with `cladecombiner`

A taxon aggregation tool takes in a list of *input* taxa that have been observed in a dataset and a list of *output* taxa that are the desired modeling units, and outputs a mapping from each input taxon to each output taxon. It should be able to handle multiple distinct sets of input taxa and produce comparable and equivalent mappings for each. This can arise in, for example, prospective evaluation, when the evaluation dataset comes in separately after the forecasts have been produced.

Taxon aggregation need not be excessively burdensome, but there are some sharp edges which can make naive implementations brittle. We have written a Python package `cladecombiner`, available at https://github.com/CDCgov/cladecombiner, which handles evolving taxonomies in an explicitly phylogenetic and general way. Out of the box, `cladecombiner` supports both Pango lineages and Nextstrain clades for SARS-CoV-2. The implementation is general and extensible, however, such that adding functionality for additional taxonomic systems is straightforward. A tradeoff of handling aggregation in an explicitly phylogenetic manner is that non-monophyletic taxa (a pervasive feature in variant nowcasting) must be handled with care.

`cladecombiner` is explicitly phylogenetic, which helps ensure comparability of aggregations, that is, the portability of an aggregation from one dataset to another. Imagine we have two datasets. In one, a set of sequences is labeled as 24F; in the other they are labeled as 24B, which is the parent clade of 24F. If the user specifies that they want to use clades 24B, 24C, and 24G as the modeling units, then `cladecombiner` will assign the sequences labeled taxon 24F to unit 24B. This choice, which makes the two datasets comparable, is likely the one that would be made in manually curated aggregation, but without any *ad hoc*, hard-coded aggregation. `cladecombiner` would also, in this example, automatically assign taxa that are not descendants of 24B, 24C, or 24G as "other."

`cladecombiner` assigns every sequence to an output taxon that is a tip in the phylogeny of all recognized taxa. This ensures that every sequence is assigned to exactly one modeling unit and removes the need for special cases in aggregation algorithms. When needed, `cladecombiner` creates *ad hoc* tip taxa for sequences

that are not assigned to a tip taxon. Specifically, for each taxon T that consists of sequences that are more finely resolved and other sequences that are not more finely resolved, `cladecombiner` creates an *ad hoc* taxon T* that consists of the sequences in T that are not also in any descendant taxon of T. T* is treated as a descendant taxon of T and as a tip in the phylogeny of all recognized taxa. For example, imagine the user selects 24B, 24C, 24F, and 24G as the output taxa. While these clades are all monophyletic, they are nested and not disjoint: 24C, 24F, and 24G are all direct descendants of 24B, and there are sequences in 24B that are not in any of 24C, 24F, or 24G. In this situation, `cladecombiner` first assigns sequences in 24C, 24F, and 24G to their corresponding output taxa. It then creates a new output taxon 24B* consisting of all sequences labeled as input taxon 24B but not further resolved into any of the descendant taxa.

For Nextstrain clades and Pango lineages, `cladecombiner` is set up to access versioned lists of recognized taxa. In this way, input taxa may be phylogenetically mapped to their most recent recognized ancestor in a previous version of taxonomy. We note that, in the analyses presented in this manuscript, this historical aggregation is the only functionality of `cladecombiner` employed.

References

[1] Administration for Strategic Preparedness and Response. *Pause in the Distribution of Bamlanivimab/etesevimab*. June 25, 2021. URL: https://web.archive.org/web/20231108165618/https://aspr.hhs.gov/COVID-19/Therapeutics/updates/Pages/important-update-25June2021.aspx.

[2] Administration for Strategic Preparedness and Response. *Resumption in Use and Distribution of Bamlanivimab/Etesevimab in all U.S. States, Territories, and Jurisdictions*. Sept. 2, 2021. URL: https://web.archive.org/web/20231108142505/https://aspr.hhs.gov/COVID-19/Therapeutics/updates/Pages/important-update-02September2021.aspx.

[3] Administration for Strategic Preparedness and Response. *Shipment of Bamlanivimab/etesevimab Resumes to Hawaii*. Oct. 21, 2021. URL: https://web.archive.org/web/20230511082037/https://aspr.hhs.gov/COVID-19/Therapeutics/updates/Pages/important-update-21October2021.aspx.

[4] Administration for Strategic Preparedness and Response. *Shipment to Hawaii Paused for Bamlanivimab/etesevimab Monoclonal Antibody Therapeutic*. Oct. 8, 2021. URL: https://web.archive.org/web/20231108153725/https://aspr.hhs.gov/COVID-19/Therapeutics/updates/Pages/important-update-08October2021.aspx.

[5] Eli Bingham et al. "Pyro: Deep universal probabilistic programming". In: *Journal of Machine Learning Research* 20.28 (2019), pp. 1–6. URL: https://dl.acm.org/doi/10.5555/3322706.3322734.

[6] Centers for Disease Control and Prevention. *Nowcasting Data Research*. 2026. URL: https://www.cdc.gov/cfa-behind-the-model/php/data-research/nowcasting.html.

[7] Centers for Disease Control and Prevention. *Variants and Genomic Surveillance*. 2025. URL: https://www.cdc.gov/covid/php/variants/variants-and-genomic-surveillance.html.

[8] Yinan Feng et al. "CovTransformer: A transformer model for SARS-CoV-2 lineage frequency forecasting". In: *Virus Evolution* 10.1 (2024), veae086. DOI: 10.1093/ve/veae086.

[9] Food and Drug Administration. *Authorization of Emergency Use of a Drug Product During the COVID-19 Pandemic; Availability*. Federal Register 89 FR 46127. May 28, 2024. URL: https://www.federalregister.gov/documents/2024/05/28/2024-11640/authorization-of-emergency-use-of-a-drug-product-during-the-covid-19-pandemic-availability.

[10] Food and Drug Administration. *Coronavirus (COVID-19) update: FDA limits use of certain monoclonal antibodies to treat COVID-19 due to Omicron*. Jan. 24, 2022. URL: https://web.archive.org/web/20220124215636/https://www.fda.gov/news-events/press-announcements/coronavirus-covid-19-update-fda-limits-use-certain-monoclonal-antibodies-treat-covid-19-due-omicron.

[11] Food and Drug Administration. *Emergency Use Authorization Archived Information*. 2020. URL: https://www.fda.gov/emergency-preparedness-and-response/mcm-legal-regulatory-and-policy-framework/emergency-use-authorization-archived-information.

[12] Food and Drug Administration. *FDA Announces Bebtelovimab is Not Currently Authorized in Any US Region*. Nov. 30, 2022. URL: https://www.fda.gov/drugs/drug-safety-and-availability/fda-announces-bebtelovimab-not-currently-authorized-any-us-region.

[13] Food and Drug Administration. *FDA announces Evusheld is not currently authorized for emergency use in the U.S.* Jan. 26, 2023. URL: https://www.fda.gov/drugs/drug-safety-and-availability/fda-announces-evusheld-not-currently-authorized-emergency-use-us.

[14] Food and Drug Administration. *FDA updates Sotrovimab emergency use authorization*. Mar. 25, 2022. URL: https://web.archive.org/web/20220325225330/https://www.fda.gov/drugs/drug-safety-and-availability/fda-updates-sotrovimab-emergency-use-authorization.

[15] Food and Drug Administration. *Frequently Asked Questions on the Emergency Use Authorization of REGEN-COV (Casirivimab and Imdevimab)*. Jan. 31, 2022. URL: https://www.fda.gov/media/143894/download.

[16] Tilmann Gneiting and Adrian E Raftery. "Strictly proper scoring rules, prediction, and estimation". In: *Journal of the American Statistical Association* 102.477 (2007), pp. 359–378. DOI: 10.1198/016214506000001437.

[17] Matthew D Hoffman and Andrew Gelman. "The No-U-Turn sampler: adaptively setting path lengths in Hamiltonian Monte Carlo". In: *Journal of Machine Learning Research* 15.1 (2014), pp. 1593–1623. URL: https://jmlr.org/papers/v15/hoffman14a.html.

[18] Alexander Jordan, Fabian Krüger, and Sebastian Lerch. "Evaluating Probabilistic Forecasts with scoringRules". In: *Journal of Statistical Software* 90.12 (2019), pp. 1–37. DOI: 10.18637/jss.v090.i12. URL: https://www.jstatsoft.org/index.php/jss/article/view/v090i12.

[19] Fabian Krüger et al. "Predictive inference based on Markov chain Monte Carlo output". In: *International Statistical Review* 89.2 (2021), pp. 274–301. DOI: 10.1111/insr.12405.

[20] MacArthur et al. "Collaborative estimation and evaluation of SARS-CoV-2 variant nowcasting in the U.S." In preparation.

[21] Dylan H Morris et al. "Predictive modeling of influenza shows the promise of applied evolutionary biology". In: *Trends Microbiol* 26.2 (Feb. 2018), pp. 102–118. DOI: 10.1016/j.tim.2017.09.004.

[22] Du Phan, Neeraj Pradhan, and Martin Jankowiak. "Composable effects for flexible and accelerated probabilistic programming in NumPyro". In: *arXiv* (2019). DOI: 10.48550/arXiv.1912.11554.

[23] Andrew Rambaut et al. "A dynamic nomenclature proposal for SARS-CoV-2 lineages to assist genomic epidemiology". In: *Nature Microbiology* 5.11 (2020), pp. 1403–1407. DOI: 10.1038/s41564-020-0770-5.

[24] Robert C. Reiner et al. "Time-varying, serotype-specific force of infection of dengue virus". In: *PNAS* 111.26 (May 2014). DOI: 10.1073/pnas.1314933111.

[25] Zachary Susswein et al. "Leveraging global genomic sequencing data to estimate local variant dynamics". In: *medRxiv* (2023). DOI: 10.1101/2023.01.02.23284123.

[26] Yatish Turakhia et al. "Ultrafast Sample placement on Existing tRees (UShER) enables real-time phylogenetics for the SARS-CoV-2 pandemic". In: *Nature Genetics* 53.6 (2021), pp. 809–816. DOI: 10.1038/s41588-021-00862-7.

[27] World Health Organization. *Updated working definitions and primary actions for SARS-CoV-2 variants*. Oct. 4, 2023. URL: https://www.who.int/publications/m/item/updated-working-definitions-and-primary-actions-for--sars-cov-2-variants.