Using BEAST to Reconstruct Pathogen Spread Incorporating Travel Data

Samuel L. Hong

October 23, 2020

Abstract

TBD

INTRODUCTION

Bayesian Evolutionary Analysis Sampling Trees (BEAST) is a software package [**?**] that attempts to provide a general framework for parameter estimation and hypothesis testing of evolutionary models from molecular sequence data [**???**]. The software uses a combination of complementary models to infer time-calibrated phylogenetic trees from an alignment of time-stamped sequences. BEAST has been widely used in the field of phylodynamics and molecular epidemiology of infectious to answer epidemiological questions from molecular data. Within this framework, it is possible to compare different evolutionary models to find which evolutionary hypothesis fits most with an epidemic processes observed.

A specific type of analysis that can be performed within this context is pathogen phylogeography. This analysis aims to answer the question of "how did an epidemic spread through space and time?" by jointly reconstructing the evolutionary and geographical history of a pathogen population as annotated time-scaled phylogenies conditioned on the observed locations at the tips. Bayesian phylogeographic analyses in BEAST have enjoyed wide success in uncovering the origins of viral lineages [**?**] and characterizing pathogen spread to inform public health response [**?**].

We can use BEAST to model the spatial spread of a pathogen between discrete locations using discrete trait analysis (DTA). This method estimates the probability of pathogen transmission between two locations using models analogous to those used for nucleotide substitution [**?**]. Recently, a new approach has been developed to integrate individual travel history data to DTA reconstructions in order to obtain more realistic reconstructions of pathogen spread that mitigate sampling bias [**?**].

In this unit, we provide four related protocols to reproduce the travel history aware phylogeographic reconstructions performed in Lemey et. al, 2020. Protocol 1 introduces the GISAID database and gives the steps necessary to construct a SARS-CoV-2 multiple sequence alignment from sequences in GISAID. In Protocol 2, we provide instructions on how to set up a generalized linear model analysis for discrete state phylogegraphic inference using BEAUti. Protocol 3 introduces an automated script to modify a BEAUti generated XML file to incorporate travel history data. Finally in Protocol 4, we guide the user through the process of visualizing individual geographic histories over the entire posterior of trees sampled with BEAST.

PROTOCOL 1: CONSTRUCTING A SARS-CoV-2 MSA FROM GISAID SEQUENCES

The first step in any phylogenetic analysis is to obtain a high-quality multiple sequence alignment (MSA). For SARS-CoV-2 analyses, the largest repostiory of genomes is available through the GISAID database. The GISAID initiative provides a plataform to openly share genomic data of influenza and SARS-CoV-2 viruses [**?**]. Access to the database is free (`https://www.gisaid.org/registration/register/`), but requires the user to register and agree to GISAID's terms for access.

This protocol describes how to construct a SARS-CoV-2 MSA from sequences downloaded from GISAID. In this example, we will construct an alignment for the 282-taxa dataset used in Lemey et. al, 2020.

Necessary Resources

Hardware

Standard workstation running Linux, MacOS, or Windows.

Software

A modern web browser (Google Chrome and Mozilla Firefox recommended).
The latest MAFFT [**?**] version (v7.453 used in this protocol)
The latest Aliview [**?**] version (v1.26 used in this protocol).

Files

A list of GISAID accessions desired (example available at `github`)

1. Search and download the desired sequences in the GISAID database

Log in into GISAID and click on `EpiCoV`™'s Browse tab to access a table with all available SARS-CoV-2 sequences in the database. To bulk download sequences by accession ID, click on the Fulltext▲ button, write your comma-separated list of accessions in the search box, and download the FASTA sequences using the download button. In this example, we will save the sequences as `gisaid_selection.fasta`.

> *We can also use the* `EpiCoV` *Browse portal to download a custom selection of genomes. On the header section of the table you will see multiple search fields and drop-down menus to filter sequences according to you desired criteria.*

2. Remove white-spaces from the FASTA file

```
sed -i.bkp "s/ /_/g" gisaid_selection.fasta
```

> *To avoid potential issues when parsing the FASTA headers, we replace all white-spaces in the file with underscores using* `sed`*. We use the* `-i` *flag to find and replace the file in-place while keeping a backup of the original file.*

3. Align the sequences

```
cat utr.fasta >> gisaid_selection.fasta
mafft --thread -1 --nomemsave gisaid_selection.fasta > gisaid_aln.fasta
```

To remove potential sequencing errors in the error-prone 5' and 3' ends of the virus, we include the reference sequences for the 5' and 3' untranslated regions (UTR) of the SARS-CoV-2 genome. We do this so that we can later trim these regions from the final alignment. We concatenate these sequences to the multi-fasta containing our genomes of interest, and align all sequences using MAFFT.

4. Manually trim UTR regions in the MSA using Aliview

In Aliview, visually identify the UTR sequences and manually select the corresponding sites. Remove the selected sites using the Edit menu . Remove the now-empty reference sequences and save the trimmed MSA.

Once we obtain our alignment from MAFFT, we visualize the MSA in Aliview to identify the sites corresponding to the UTR regions. Once this is done, we can remove the now empty reference sequences and save the trimmed MSA.

PROTOCOL 2: SETTING UP A DISCRETE TRAIT PHYLOGEOGRAPHIC RECONSTRUCTION IN BEAUTI

The travel history approach is an extension of traditional discrete trait analysis (DTA) available in BEAST. BEAUti is an interactive graphical application for designing your analysis and generating the a BEAST XML file which will be used to run the analysis.

This protocol describes how to set up a generalized linear model (GLM) variant of the DTA analysis using BEAST. In this example, we will use the MSA generated in Protocol 1 to set up a GLM phylogeographic reconstruction using a flight matrix and a distance matrix as covariates (See **?** for details).

Necessary Resources

Hardware

Standard workstation running Linux, MacOS, or Windows.

Software

Latest BEAUti version (v1.10.5)

Files

Multiple sequence alignment
Tab-delimited metadata file
Covariate matrices in CSV format

Example metadata and covariate files available at `gihutb`

1. Import the MSA into BEAUti

Load the MSA into BEAUti by selectin Import Data from the File menu. You can also do this by dragging the fasta file into the box in the Partitions tab.

2. Specify the tip sampling dates

Select the Tips tab and check the "Use tip dates" box. By default all taxa will show as having a date of zero (i.e. all sequences were sampled at the same time in the present). To specify each sampling date, select "Import Dates" and load the metadata file with the "Parse calendar dates with variable precision" option. This allows for taxon dates of different resolutions (e.g. Year-Month-Day vs Year-Month). Estimate the sampling dates for the taxa without day-level resolution by selecting Sampling uniformly from precision in the Tip date sampling menu at the bottom of the table.

It is also possible to specify the sampling dates without using a metadata file. This is done by parsing the FASTA headers of each sequence. To do this, click on Parse Dates, and specify the rules for delimiting the date from all taxon labels.

3. Specify the sampling location of each taxa as a discrete trait.

Click on the Traits tab. To associate each taxon with a sampling location, click on "Import traits" and select the metadata file. This will create a new trait for each column on the metadata file. Delete all non-relevant traits by clicking on the "-" button at the bottom-left of the page (keep only the "location" trait for this example). Select the desired trait and click on "create partition from trait". A new partition containing the trait data will be created under the Partitions tab (SCREENSHOT?).

It is also possible to add a discrete trait without using a metadata file. This is done by parsing the FASTA headers of each sequence. To do this, click on Add Trait to create a new trait with a corresponding data partition. Select all taxa and click on Guess trait values to parse the trait values from the taxon labels.

4. Set up the nucleotide and trait substitution models

Click on the "Sites— tab. Following Lemey et al. we will specify a HKY+Gamma nucleotide substitution model, and a GLM for the location trait. Load the covariates by clicking on Setup GLM and Import Predictors. Check the Log and Std boxes to log-transform and standardize the GLM predictors.

In this context, predictor and covariate are used interchangeably. By default each predictor name will be the same as the name of the file it originates from. You can specify new names by double clicking on each name. Non-pairwise covariates can also be setup as origin and destination predictors in this window.

5. Specify a clock model

Following Lemey et al., click on the "Clocks" tab to specify a strict clock model for both the nucleotide and trait partitions.

BEAST analyses operate under the assumption of a molecular clock. For this to hold valid, there needs to be enough temporal signal in the data (See Critical parameters and troubleshooting). For more information on the different molecular clock models available in BEAST see `http://beast.community/clocks`

.

6. Specify a demographic model

Click on the "Trees" tab to specify the Tree prior. Following Lemey et al., 2020 specify an exponential growth coalescent model parametrized with a Growth Rate parameter.

> *By default, BEAST will initialize the analysis with a randomly generated starting tree. It is also possible to start the analysis from a user-specified time-scaled phylogeny. This is usually done to reduce the burn-in time required for the tree topologies to converge. Specify a starting tree by clicking on Import Data under the File menu, to load a Nexus tree into BEAUti.*

7. Set up the ancestral state reconstruction for the location trait

Click on the "States" tab and select the location partition. Check the "Reconstruct state change counts" and "Reconstruct complete change history on tree" boxes to save complete realizations of the spatial spread process on the output trees.

8. Specify the priors

Click on the "Priors" tab. Following Lemey et al., we use default priors and only specify a Lognormal prior with `mu=0` and `sigma=10` for the effective population size, and a Laplace prior with `mean=0` and `scale=100` for the exponential growth rate parameter.

> *BEAST aims to offer sensible default priors when informative prior information is unavailable. A wide range of prior distributions is also available to customize each analysis as needed.*

9. Set up the operators

Click on the "Operators" tab. Identify the tip-date sampling operators on the table. These have the description "Uniform sample from precision of age of this tip". The parameters associated with these operators tend to converge rapidly and have good mixing. Decrease the weight of these operators to 0.25 to sample less frequently from this parameters so that the MCMC can get more samples from other parameters of interest.

> *Operators are used to propose new values for each parameter sampled from the posterior distribution. Different combinations of operators can be used to fix certain parameters and customize the analysis as needed (e.g. estimating spatial spread on a fixed tree).*

9. Set up the MCMC options and generate the BEAST XML file

Click on the "MCMC" tab. Set "Length of chain" to 100,000,000 states and "Log parameters every" to 50,000 states. This will thin the MCMC results so that only 2,000 samples are collected by the end of the run. Set your file name stem as the desired output file name (e.g. `282_GISAID_sarscov2`), and click on "Generate BEAST File" to create a BEAST XML file.

Thinning consists on discarding all but the nth sample from an MCMC run. This subsampling technique is done to decrease the autocorrelation in the samples collected and reduce the file size of the output. This is important since Bayesian phylogenetic analyses often require very long chains, and storing every single state would be computationally prohibitive.

PROTOCOL 3: PHYLOGEOGRAPHIC RECONSTRUCTION INCORPORATING TRAVEL HISTORY INFORMATION

Phylogeographic reconstruction using DTA has been show to be sensitive to spatiotemporal sampling bias. The ancestral reconstruction of locations will depend on the availability of samples from each location. In practice, this means that over/undersampling of sequences from a given location can greatly impact the ancestral locations being reconstructed. One way to mitigate sampling bias is by incorporating travel history information. Travel history data can be used to correct for gaps in sampling by allowing for ancestral nodes to be in a given location even when sequences are not available.

This protocol explains how augment a DTA analysis generated in BEAUti by incorporating travel history data. In this example, we will use the XML file generated in Protocol 2 and modify it to include individual travel history data (See Lemey et. al, 2020 for details).

Necessary Resources

Hardware

Standard workstation running Linux, MacOS, or Windows.
CUDA-compatible Graphics Processing Unit (optional but recommended)

Software

Latest BEAST jar file (v1.10.5)
Python v3.6+ with packages numpy and lxml
`add_travel_history.py` Python script

Files

XML file for a DTA analysis set up in BEAUti
Travel history metadata CSV file
Augmented covariate files

Example files are provided in the repository `github`*. The example in this protocol assumes an XML file for a GLM phylogeographic reconstruction with travel history locations without any sequence samples. Covariate files augmented to include the new locations are thus required to accommodate the increase in state space.*

1. Update the BEAST XML file to incorporate travel history data

```
python add_travel_history.py --xml 282_GISAID_sarscov2.xml
        --hist travel_metadata.csv
        --covariate augmented_flight_matrix.csv
        --covariate augmented_intra_cont_dist.csv
        --out 282_GISAID_sarscov2_travelHist.xml
```

Although this example uses a GLM-DTA analysis, the add_travel_history.py script also works for symmetric and asymmetric DTA analyses. For such cases, run the same command without the covariate flags. This script also requires for the travel history metadata to follow a specific format. The metadata file must contain the following columns: "name" (taxon name), "travelHistory" (travel location), "travelDays" (date of travel as days before to sampling date), "priorMean" and "priorSTDEV" (prior specifications for the mean and standard deviation of a Normal prior on travel dates when exact data is unavailable).

2. Start the BEAST MCMC sampler

```
java -cp beast.jar dr.app.beast.BeastMain -seed 2020
        -beagle_gpu
        -save_every 1000000
        -save_state travelHist.checkpoint
         282_GISAID_sarscov2_travelHist.xml
```

Here, we execute BEAST on the command-line using the latest jar file. We specify a starting seed with the -seed flag, and use the -beagle_cuda flag to accelerate the likelihood computations using a GPU. This option is recommended when available, as using a GPU reduces runtime by accelerating likelihood computations when performing phylogeographical analyses on large datasets. We also take advantage of the BEAST checkpointing functionality to save a snapshot of the MCMC run into the travelHist.checkpoint file every 1,000,000 states. This allows us to resume the analysis from the checkpoint in case the run becomes interrupted, or more iterations of the chain are required.

PROTOCOL 4: VISUALIZING TAXON-SPECIFIC SPATIAL TRAJECTORIES

Necessary Resources

Hardware

Standard workstation running Linux, MacOS, or Windows.

Software

Latest BEAST jar file (v1.10.5)
R with package MarkovJumpR

Files

Trees output from a BEAST travel history DTA

1. Extract all Markov jump histories for an isolate of interest

```
java -cp beast.jar dr.app.tools.TaxaMarkovJumpHistoryAnalyzer \
        -taxaToProcess "hCoV-19/Australia/NSW05/2020|EPI_ISL_412975|2020-02-28" \
        -stateAnnotation location \
        -burnin 100 \
        -msrd 2020.1748633879781 \
         282_GISAID_GLM.location.history.trees EPI_ISL_412975_MJhist.csv
```

The BEAST jar file packages a number of standalone tools that can be accessed using the -cp flag in Java. Here, we use the TaxaMarkovJumpHistoryAnalyzer tool to extract all Markov jump histories for isolate EPI_ISL_412975 , into a CSV file. This tool takes a trees file with complete state change history as an input, and outputs all spatial trajectories for some taxon/taxa. We specify the desired taxon labels through the -taxaToProcess flag, and specify the annotation name of the discrete trait which was reconstructed using -stateAnnotation. We can also remove a burn-in number of trees using the -burnin flag, and scale the output results to reflect chronological time instead of node heights using the -msrd flag by specifying the most recent sampling date. An example output file can be found in LINK

2. Load spatial trajectories into R

```
library(MarkovJumpR)
spatial_paths <- loadPaths(fileName = "EPI_ISL_412975_MJhist.csv")
```

3. Inspect spatial trajectories reconstructed

```
----
EXPLORE THE CSV OUTPUT IN R TO EXTRACT ALL LOCATIONS AND CUTOFF DATE
----
```

4. Set up plot colors

```
locations <- c("Wuhan","Italy","SEasia","Iran","Australia")
locationColors <-c("#E3272F","#31B186","#931ECF","#C695BD","#9DC7DD")
locationMap <- data.frame(location = locations,
                position = c(1, 2, 3, 4, 5))
```

5. Set up plot labels

```
dateLabels <- c("01-Dec-19", "15-Dec-19", "01-Jan-20", "15-Jan-20",
                "01-Feb-20", "15-Feb-20", "01-Mar-20" )
```

6. Plot path spatial trajectories

```
plotPaths(travelHistPaths$paths, locationMap = locationMap,
        yJitterSd = 0.1, alpha = 0.1, minTime = 2019.9,
        addLocationLine = TRUE,
        xAt = decimal_date(dmy(dateLabels)),
        xLabels = dateLabels,
        mustDisplayAllLocations = TRUE)
```

GUIDELINES FOR UNDERSTANDING RESULTS

The BEAST software package offers a flexible approach for combining demographic, molecular clock, nucleotide and trait models to infer time-scaled phylogenies. This is done under the Bayesian framework, using MCMC to sample trees along with their corresponding model parameters from the joint posterior. Protocols 2 and 3 are used to set up a model and collect samples using BEAST. Phylogenies drawn from the posterior are stored in a `.trees` file and samples from the remaining parameters are stored in a `.log` file. Here we present some of the standard tools used to interpret the outputs of a BEAST run.

Assessing convergence

Markov Chain theory states that MCMC samplers eventually converge to a stationary distribution. In the case of Bayesian inference, this stationary distribution is the posterior. Given enough time we know that the chain will converge, but it may take considerable time for this to happen. We can visually assess the convergence of a BEAST run by inspecting the sampled parameter values across an MCMC run. To do so, we load a `.log` file in Tracer (`https://beast.community/tracer`), and visually inspect the trace plot, which shows a time series of the parameter values drawn by the sampler. A detailed guide on how to do use Tracer can be found at `https://beast.community/tracer_convergence.html`

Parameter estimates and effective sample size

Loading a .log file in Tracer will also show the estimates and effective sample size (ESS) values for each model parameter. A characteristic of MCMC samplers is that samples collected tend to be correlated. This in turn poses a challenge, since having a large number of samples does not guarantee a considerable reduction of the uncertainty in our posterior estimates. A way to control for this is to look at the ESS values of an estimate. ESS is defined as the number of independently drawn samples equivalent to the MCMC results. Tracer will automatically calculate these values for all parameters in a log file, and flag values above 100 and 200. ESS values will increase as more and more samples are collected by the sampler. Larger ESS values will result in more precise posterior estimates but require higher computational resources. Tracer will automatically calculate the ESS for all parameters in a log file, and flag values above 100 (acceptable) and 200 (ideal).

Summarizing trees and phylogeographic estimates

Individually inspecting every tree from the posterior becomes an impractical way to interpret the MCMC results. We are thus required to summarize the distribution of trees sampled as a point-estimate with associated uncertainties. The TreeAnnotator tool in BEAST allows us to create a maximum clade credibility (MCC) tree to summarize the results for this purpose. For every tree in the distribution, a posterior clade probability for each node (i.e. the support for a node) is calculated by looking at the frequency of the clustering that is defined by the node in question. The MCC tree is then defined as the tree which maximizes the product of the posterior clade probabilities. Instructions on how to use TreeAnnotator can be found at `beast.community/second_tutorial`.

In some cases, the posterior support for all nodes in a tree is such that many topologies do not end up represented in the MCC tree. For such cases, a point-estimate of a tree is unable to capture the diverse phylogeographic histories compatible with the data. Protocol 4 allows us to inspect individual spatial trajectories by summarizing across all possible phylogenetic ancestries in the posterior. Each spatial trajectory in the posterior is represented by a stepwise curve, where vertical lines represent transitions between two locations, and horizontal lines time intervals where the pathogen remains in the same location. The relative density of lines reflects the posterior uncertainty in spatiotemporal ancestry.

COMMENTARY

Background Information

Methods for phylogeographic inference can be broadly categorized into two approaches depending on the assumptions used to model the spatial spread of a pathogen. Coalescent approaches model movement in terms of a migration matrix, and relate

migration rates with a location's population size under the structured coalescent [**??**]. On the other hand, diffusion approaches considers sampling locations as observed traits independent from the tree generating process, and model movement across space as a continuous-time Markov chain [**??**]. While coalescent approaches are theoretically a more robust approach, the lack of computationally efficient implementations (especially for larger datasets) has made CTMC methods more widely adopted. Currently BEAST v1.10.5 offers only implementations for CTMC models.

CTMC methods can be further divided into discrete and continuous depending on the spatial resolution being considered. Discrete locations are modeled using discrete trait analysis (DTA), with transition rates between locations in the form of a CTMC matrix analogous to those used for nucleotide substitution [**?**]. In contrast, continuous locations are modeled using Brownian diffusion based random-walk models [**?**]. Much of the genomic data collected has a spatial resolution coarser than latitude and longitude, which makes DTA the only practical approach to study the spread of an epidemic.

Under the DTA formulation, movement between K discrete locations is parameterized in terms of a $K \times K$ infinitesimal rate matrix Λ, where Λ_{ij} is the instantaneous movement rate from location i to j. This model has been extended to allow for symmetrical and asymmetrical transition rates, and uses Bayesian Stochastic Search Variable Selection (BSSVS) to limit the number of rates to only those that adequately explain the phylogenetic diffusion process. An alternative formulation of this model parameterizes the rate matrix using a generalized linear model [**?**]. Here, the transition rates are defined as a linear combination of P of potential explanatory predictors $(x_{ij1}, \ldots, x_{ijP})$, with corresponding coefficients $(\beta_1, ..., \beta_P)$ and indicator variables $(\delta_1, ..., \delta_P)$ such that $Log(\Lambda_{ij}) = \sum_{p=1}^{P} \beta_p \delta_p x_{ijp}$. This model specification allows us to use BSSVS to explore the space of 2^P predictor combinations and obtain a posterior probability on the indicator variables to determine the support for inclusion of reach predictor in the model. Furthermore, incorporating travel history data can be done to any of the DTA variants by augmenting augments the available dataset to include ancestral nodes associated with a known state but not with a known sequence [**?**]. Ambiguous ancestral locations can be allowed by integrating over the all possible locations with equal or user specified weights [**?**].

CRITICAL PARAMETERS AND TROUBLESHOOTING

A critical assumption for any BEAST analysis is that of measurably evolving populations. Measurably evolving populations [**??**] refer to time-stamped sequence data where a sufficient amount of molecular evolution has occurred throughout the sampling period to establish a statistical relationship between genetic divergence and time. A dataset conforming to this criteria is said to contain sufficient temporal signal. A lack of temporal signal will result in ultimately flawed analyses with unreliable divergence time estimates. Popular ways to assess temporal signal include doing a root-to-tip regression of genetic divergence and sampling times on a maximum likelihood tree [**?**] and permutating tip date labels through date-randomization [**?**]. Recently, a formal way to assess temporal signal has been developed under a Bayesian framework using model comparison through Bayes factors [**?**]. In cases where the temporal signal is not enough, one can resort to adding more data to increase temporal coverage or using prior knowledge to inform the molecular clock rate or the time to the most recent common ancestor.

Another commonly encountered issue is that low ESS values. At the end of a BEAST run you will generally notice that some parameters have much higher ESS values than others. One way to increase the ESS values of a parameter is to decrease the weight of other parameters with enough ESS to increase the sampling frequency of the parameter of interest (e.g.

Protocol 2 step 9). Other ways to obtain more samples include increasing the MCMC chain length and combining multiple independent BEAST runs.

SUGGESTIONS FOR FURTHER ANALYSIS

TBD

References

Biek, R., Pybus, O. G., Lloyd-Smith, J. O., and Didelot, X. (2015). Measurably evolving pathogens in the genomic era. *Trends Ecol. Evol.*, 30(6):306–313.

De Maio, N., Wu, C.-H., O'Reilly, K. M., and Wilson, D. (2015). New routes to phylogeography: A bayesian structured coalescent approximation. *PLoS Genet.*, 11(8):e1005421.

Drummond, A. J., Pybus, O. G., Rambaut, A., Forsberg, R., and Rodrigo, A. G. (2003). Measurably evolving populations. *Trends Ecol. Evol.*, 18(9):481–488.

Drummond, A. J. and Rambaut, A. (2007). BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evol. Biol.*, 7:214.

Drummond, A. J., Suchard, M. A., Xie, D., and Rambaut, A. (2012). Bayesian phylogenetics with BEAUti and the BEAST 1.7. *Mol. Biol. Evol.*, 29(8):1969–1973.

Duchene, S., Lemey, P., Stadler, T., Ho, S. Y. W., Duchene, D. A., Dhanasekaran, V., and Baele, G. (2020). Bayesian evaluation of temporal signal in measurably evolving populations. *Mol. Biol. Evol.*

Dudas, G., Carvalho, L. M., Bedford, T., Tatem, A. J., Baele, G., Faria, N. R., Park, D. J., Ladner, J. T., Arias, A., Asogun, D., Bielejec, F., Caddy, S. L., Cotten, M., D'Ambrozio, J., Dellicour, S., Di Caro, A., Diclaro, J. W., Duraffour, S., Elmore, M. J., Fakoli, L. S., Faye, O., Gilbert, M. L., Gevao, S. M., Gire, S., Gladden-Young, A., Gnirke, A., Goba, A., Grant, D. S., Haagmans, B. L., Hiscox, J. A., Jah, U., Kugelman, J. R., Liu, D., Lu, J., Malboeuf, C. M., Mate, S., Matthews, D. A., Matranga, C. B., Meredith, L. W., Qu, J., Quick, J., Pas, S. D., Phan, M. V. T., Pollakis, G., Reusken, C. B., Sanchez-Lockhart, M., Schaffner, S. F., Schieffelin, J. S., Sealfon, R. S., Simon-Loriere, E., Smits, S. L., Stoecker, K., Thorne, L., Tobin, E. A., Vandi, M. A., Watson, S. J., West, K., Whitmer, S., Wiley, M. R., Winnicki, S. M., Wohl, S., Wölfel, R., Yozwiak, N. L., Andersen, K. G., Blyden, S. O., Bolay, F., Carroll, M. W., Dahn, B., Diallo, B., Formenty, P., Fraser, C., Gao, G. F., Garry, R. F., Goodfellow, I., Günther, S., Happi, C. T., Holmes, E. C., Kargbo, B., Keïta, S., Kellam, P., Koopmans, M. P. G., Kuhn, J. H., Loman, N. J., Magassouba, N., Naidoo, D., Nichol, S. T., Nyenswah, T., Palacios, G., Pybus, O. G., Sabeti, P. C., Sall, A., Ströher, U., Wurie, I., Suchard, M. A., Lemey, P., and Rambaut, A. (2017). Virus genomes reveal factors that spread and sustained the ebola epidemic. *Nature*, 544(7650):309–315.

Elbe, S. and Buckland-Merrett, G. (2017). Data, disease and diplomacy: GISAID's innovative contribution to global health. *Glob Chall*, 1(1):33–46.

Katoh, K. and Standley, D. M. (2013). MAFFT multiple sequence alignment software version 7: improvements in performance and usability. *Mol. Biol. Evol.*, 30(4):772–780.

Larsson, A. (2014). AliView: a fast and lightweight alignment viewer and editor for large datasets. *Bioinformatics*, 30(22):3276–3278.

Lemey, P., Hong, S. L., Hill, V., Baele, G., Poletto, C., Colizza, V., O'Toole, Á., McCrone, J. T., Andersen, K. G., Worobey, M., Nelson, M. I., Rambaut, A., and Suchard, M. A. (2020). Accommodating individual travel history and unsampled diversity in bayesian phylogeographic inference of SARS-CoV-2. *Nat. Commun.*, 11(1):5110.

Lemey, P., Rambaut, A., Bedford, T., Faria, N., Bielejec, F., Baele, G., Russell, C. A., Smith, D. J., Pybus, O. G., Brockmann, D., and Suchard, M. A. (2014). Unifying viral genetics and human transportation data to predict the global transmission dynamics of human influenza H3N2. *PLoS Pathog.*, 10(2):e1003932.

Lemey, P., Rambaut, A., Drummond, A. J., and Suchard, M. A. (2009). Bayesian phylogeography finds its roots. *PLoS Comput. Biol.*, 5(9):e1000520.

Lemey, P., Rambaut, A., Welch, J. J., and Suchard, M. A. (2010). Phylogeography takes a relaxed random walk in continuous space and time. *Mol. Biol. Evol.*, 27(8):1877–1885.

Müller, N. F., Rasmussen, D., and Stadler, T. (2018). MASCOT: parameter and state inference under the marginal structured coalescent approximation. *Bioinformatics*, 34(22):3843–3848.

Rambaut, A., Lam, T. T., Max Carvalho, L., and Pybus, O. G. (2016). Exploring the temporal structure of heterochronous sequences using TempEst (formerly Path-O-Gen). *Virus Evol*, 2(1):vew007.

Ramsden, C., Holmes, E. C., and Charleston, M. A. (2009). Hantavirus evolution in relation to its rodent and insectivore hosts: no evidence for codivergence. *Mol. Biol. Evol.*, 26(1):143–153.

Scotch, M., Tahsin, T., Weissenbacher, D., O'Connor, K., Magge, A., Vaiente, M., Suchard, M. A., and Gonzalez-Hernandez, G. (2019). Incorporating sampling uncertainty in the geospatial assignment of taxa for virus phylogeography. *Virus Evol*, 5(1):vey043.

Suchard, M. A., Lemey, P., Baele, G., Ayres, D. L., Drummond, A. J., and Rambaut, A. (2018). Bayesian phylogenetic and phylodynamic data integration using BEAST 1.10. *Virus Evol*, 4(1):vey016.

Worobey, M., Watts, T. D., McKay, R. A., Suchard, M. A., Granade, T., Teuwen, D. E., Koblin, B. A., Heneine, W., Lemey, P., and Jaffe, H. W. (2016). 1970s and 'patient 0' HIV-1 genomes illuminate early HIV/AIDS history in north america. *Nature*, 539(7627):98–101.