

SpreaD3: Spatial Phylogenetic Reconstruction of Evolutionary Dynamics using Data-Driven Documents (D3)

Authors

- Filip Bielejec ([filip.bielejec\(sorry_spybots\)rega.kuleuven.be](mailto:filip.bielejec(sorry_spybots)rega.kuleuven.be))
- Guy Baele ([guy.baele\(sorry_spybots\)rega.kuleuven.be](mailto:guy.baele(sorry_spybots)rega.kuleuven.be))
- Bram Vrancken ([bram.vrancken\(sorry_spybots\)rega.kuleuven.be](mailto:bram.vrancken(sorry_spybots)rega.kuleuven.be))
- Marc A. Suchard ([msuchard\(sorry_spybots\)ucla.edu](mailto:msuchard(sorry_spybots)ucla.edu))
- Andrew Rambaut ([a.rambaut\(sorry_spybots\)ed.ac.uk](mailto:a.rambaut(sorry_spybots)ed.ac.uk))
- Philippe Lemey ([philippe.lemey\(sorry_spybots\)rega.kuleuven.be](mailto:philippe.lemey(sorry_spybots)rega.kuleuven.be))

Contact

Filip Bielejec ([filip.bielejec\(sorry_spybots\)rega.kuleuven.be](mailto:filip.bielejec(sorry_spybots)rega.kuleuven.be))

Download & run

SpreaD3 is available [here](#) and can be run in GUI (graphical user interface) mode by double-clicking on the jar file `spreaD3_v0.9.6.jar` or by running the following from command line:

```
$ java -jar spreaD3_v0.9.6.jar
```

Source code & scripting

The source code for SpreaD3 is freely available from: <https://github.com/phylogeography/SpreaD3>. You can also clone the project using Git by running:

```
$ git clone https://github.com/phylogeography/SpreaD3
```

cd into the base directory and compile SpreaD3 by running

```
$ ant jar
```

This builds an executable jar-file in the directory `./dist`, which can be run by double-clicking it.

Table of Contents

[Section 1: Introduction](#)

[Section 2: Software requirements](#)

[Section 3: geoJSON maps](#)

[Section 4: Usage examples](#)

[Subsection 4.1: Visualizing an MCC tree annotated with discrete traits](#)

[Parsing the data](#)

[Rendering and specifying the visualization options](#)

[Subsection 4.2: Identifying well-supported rates through Bayes factor tests](#)

[Parsing the data](#)

[Rendering and specifying the visualization options](#)

[Subsection 4.3: Visualizing an MCC tree annotated with continuous traits](#)

[Parsing the data](#)

[Rendering and specifying the visualization options](#)

[Subsection 4.4: Visualizing a distribution of trees annotated with continuous traits](#)

[Parsing the data](#)

[Rendering and specifying the visualization options](#)

[Subsection 4.5: Tips & tricks](#)

[Parsing the data](#)

[Rendering and specifying the visualization options](#)

[Subsection 4.6: KML rendering](#)

1 Introduction

SpreaD3 is a tool for analysing and visualising discrete and continuous trait evolutionary histories associated with phylogenies. It is designed primarily for use in conjunction with the popular Bayesian phylogenetic inference software BEAST (Drummond et al., 2012). However, it can also accommodate input generated by other phylogenetic inference tools (e.g. MrBayes and BEAST2 or maximum likelihood approaches), as long as the nodes and branches are annotated using the compatible syntax.

The user is offered great flexibility in and control over the visualisation process through two main innovations. First, each analysis is conceived as a two-step process. At the parsing step, the inputs are analysed and converted to a JavaScript Object Notation (JSON) compliant data file. Nodes are translated to points, branches to lines and both can have associated annotations. Next, the JSON output file is used for rendering the visualisations. By separating the parsing from the rendering, the user can quickly test various image setups without the need for repeatedly parsing the same input. Note that because JSON is a language-independent data format, the output of the parsing step can be forked to utilities based on several programming languages through dedicated packages and libraries. Second, much of the versatility comes from the capability to combine several layers of visualisation in one illustration.

On the one hand this lifts the restriction of projecting phylogenies on a pre-specified canvas, and gives the user detailed control on the type of map and the displayed attributes (e.g. province borders, altitude contours, population density, ...). At the same time this opens doors to mapping traits using any arbitrary coordinate system (see section [4.5](#) for an example). A third advantage of the layered approach to image building is the ease with which complex stratified images can be constructed (see section [4.5](#) for an example).

Alongside enhancing the flexibility in creating the visualisations, the other prime objective of SpreaD3 is to drastically facilitate the on-line publishing of interactive visualisations. This is achieved by rendering the figures using the JavaScript Data Driven Documents (D3) libraries (see www.d3js.org and Bostock et al. (2011)), to project the annotated phylogenies on a map in geoJSON data structure format (www.geojson.org).

To maintain consistency with the previous version of [SPREAD](#), and as an illustration of the branching to other renderers enabled by the versatile JSON data format as a go-between, SpreaD3 also supports re-interpreting parsed input data in the Keyhole Markup Language (KLM) for visualisations in virtual globe software like Google Earth (www.google.com/earth/).

In this tutorial we provide a detailed description of program functionalities, with an emphasis on using the Graphical User Interface (GUI), by presenting an example for the main genres of possible analyses. The data files used in the examples below can be found at <https://rega.kuleuven.be/cev/ecv/software/SpreaD3>. The user is assumed to know how to run discrete and continuous trait analyses in BEAST, and how to build a Maximum Clade Credibility (MCC) tree. Tutorials for this can be found at <http://beast.bio.ed.ac.uk/tutorials>.

2 Software requirements

SpreaD3 needs Java Runtime environment version 1.6 (at least) and a modern browser with built-in JavaScript to fully utilize its capabilities.

Google Chrome users may have to specifically set Chrome's permission to load a URL from file (other browsers do not require this). On Mac, this can for example be done by opening Chrome from a terminal window:

```
open -a 'Google Chrome' --args --allow-file-access-from-files
```

On Windows, this can be done by following these steps:

```
- start the Windows Command Prompt  
- go to your Chrome installation folder: cd C:\Users\your-user-name\AppData\Local\Google\Chrome\Application  
- type: chrome.exe --allow-file-access-from-files
```

On Linux (Ubuntu), Google Chrome can be started from the Terminal window as follows

```
/usr/bin/google-chrome --allow-access-from-files
```

3 geJSON maps

There are many places where ready-made geJSON maps are available, and various ways in which tailor-made geJSON maps can be made. We point out a few resources, but don't attempt to provide an exhaustive list. A good introduction to building geJSON maps can be found here: <http://bost.ocks.org/mike/map/>. Large repositories for geJSON maps can be found here: <http://grokbase.com/t/gg/d3-js/1372gq18j9/gejson-maps> and <http://data.okfn.org/data/datasets/geo-boundaries-world-110m>.

There are a number of easy-to-use utilities for creating geJSON maps starting from shape files. The latter can among others be found at <http://www.naturalearthdata.com/downloads/> and <http://www.gadm.org/>. An example of a conversion utility is <http://www.mapshaper.org/> or its command line version available at <https://github.com/mbloch/mapshaper>. Another alternative for converting shape files in geJSON format can be found at <http://ben.balter.com/2013/06/26/how-to-convert-shapefiles-to-geojson-for-use-on-github/>.

A handy feature of the geJSON format is that the latitude and longitude are treated as absolute values. Like this, the coordinates for a location (e.g. Leuven) remain valid when providing a more focused map (e.g. Belgium instead of the entire world). If no geJSON map is provided, a bounding box based on the provided coordinates will be created automatically.

4 Usage examples

Spread3 can process four distinct input types labelled MCC tree with DISCRETE traits (4.1), Log file from BSSVS analysis (4.2), MCC tree with CONTINUOUS traits (4.3), Tree distribution with CONTINUOUS traits (4.4). We provide an example for each of these basic analysis types, and also illustrate how more complex layered images can be constructed (e.g. 4.5). We conclude this tutorial with a brief overview of how to render parsed input in KML format (4.6).

The data files used in the examples below can be found at <https://rega.kuleuven.be/cev/ecv/software/Spread3>.

4.1 Visualizing an MCC tree annotated with discrete traits

In this example we turn to the recent Ebola virus outbreak in western Africa to show how an MCC tree summary created under a discrete diffusion model (Lemey et al., 2009) can be used to visualise a pathogen's spatiotemporal spread. Samples were collected from March to December 2014; BEAST (Drummond et al., 2012) was used to co-estimate the temporal and spatial history.

Parsing the data

The GUI opens in the 'Data' tab. Select the option 'MCC tree with DISCRETE traits' (Figure 1, nr.1), and browse to where you saved the example MCC tree to load it. Spread3 will now set the working directory to the one from which the MCC tree was loaded. This means that all output generated by Spread3 will be saved in this directory. To view the MCC tree in its geographic context, we have to associate each location with a particular latitude and longitude. To do this, indicate under which name the location trait is known (Figure 1, nr.2) and associate the appropriate coordinates with each location. For this you can either use the editor supplied with Spread3 (Figure 1, nr.3) or load a previously prepared tab-delimited file including each location, its latitude and longitude (the locations should be given in the same order of appearance as the ordering in which the location states are declared in the generalDataType block in the XML used for the analysis).

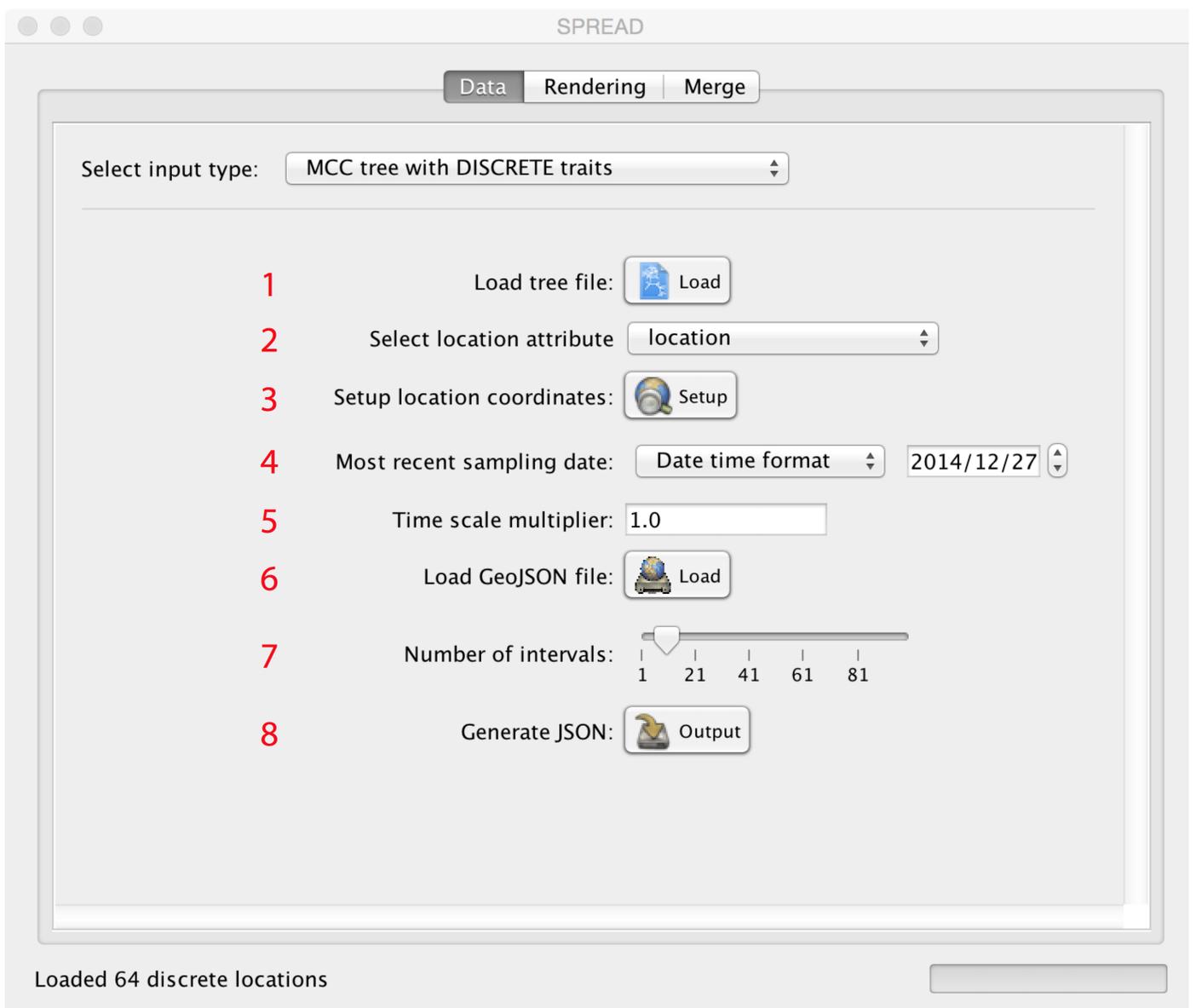


Figure 1 Parsing options for the MCC tree with DISCRETE traits analysis.

Spread3 sets the most recent sampling date to the current date. In this example, the date should be set to 27/12/2014 (Figure 1, nr.4). Leave the 'Time scale multiplier' (Figure 1, nr.5) to its default value of 1 (this option is useful when the time units should be rescaled, e.g. in molecular archeological work). Next, browse to the location where the geoJSON file is stored and load it (Figure 1, nr.6). The number of intervals equals the number of periods in which the phylogeny is partitioned (Figure 1, nr.7). Specifically, all lineages of the tree are virtually intersected at each transition time and for each trait state (e.g. location) the number of branches over which no trait state transition has occurred (e.g. no change in location between the branch's parent and child node) is summed. The reward counts ('Circular polygons') at slice n will be plotted in the animation starting from period n . Save the file as 'ebov.json' (Figure 1, nr.8).

Rendering and specifying the visualisation options

The 'Rendering' tab in the GUI by default opens the D3 renderer. Load 'ebov.json' and specify the name of the folder in which the rendered output will be stored. Following the name-giving in Figure 2, this creates a folder 'ebov_d3' with several files. The most important one is 'index.html', which can be opened by any modern internet browser by double-clicking.

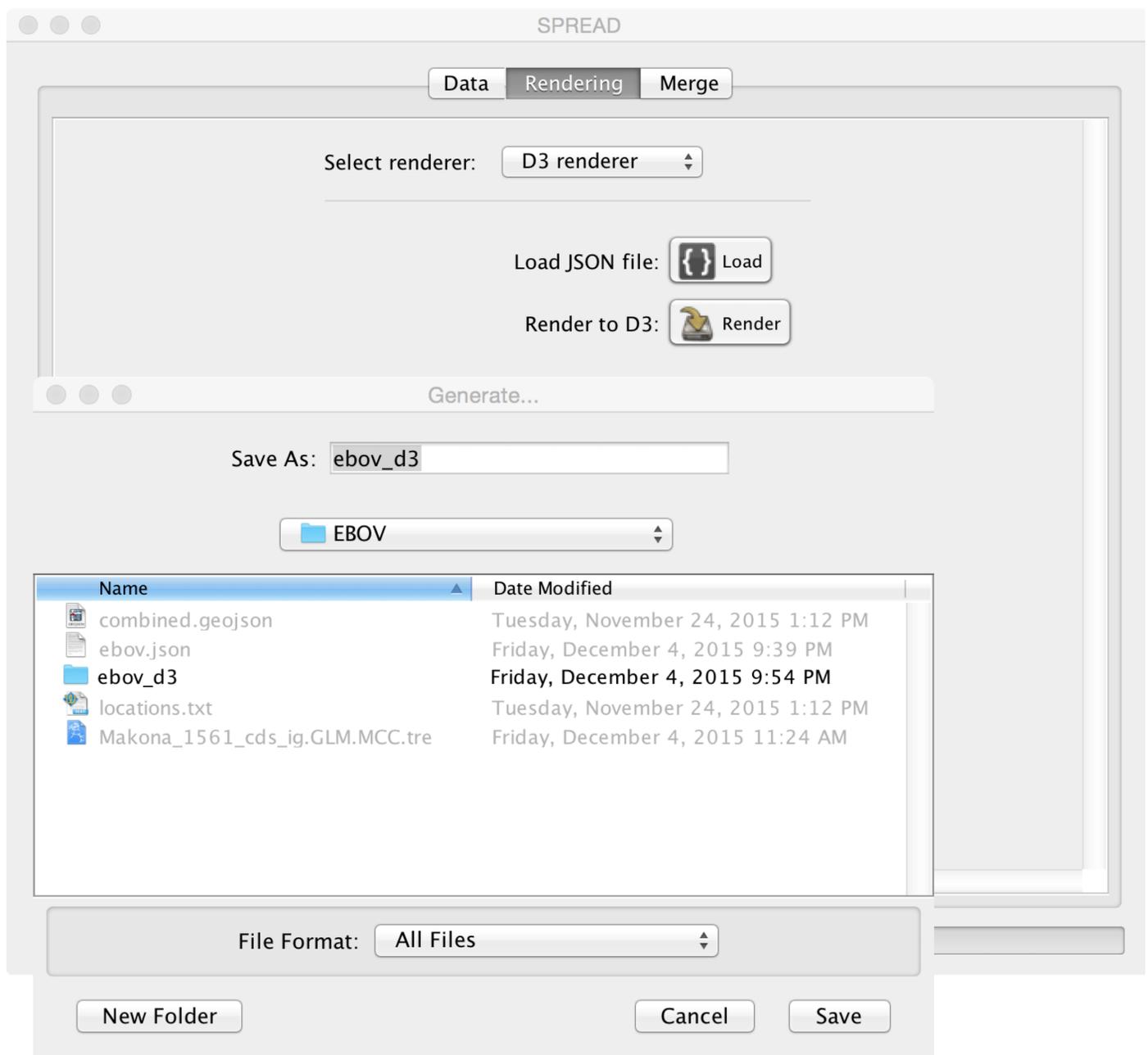


Figure 2 Rendering options for the MCC tree with DISCRETE traits analysis.

The visualisation options can be set via menu's in the left column of the window. The branches of the tree are represented by lines and the nodes by points. Each has particular properties (e.g. color, opacity, ...) that can either be given a fixed value, or have values assigned by corresponding traits.

Double-clicking 'index.html' opens a web-page showing the tree and the sampling locations mapped in their geographical context. Let's lay the visual focus for this example on the cross-country location transitions. Start by colouring the map by country. To concentrate on the pure geographical aspect of the picture, deselect the Counts and Lines layer under 'Toggle layer visibility'. Now set the 'Map color attribute' to 'ISO', which assigns a different color to Guinea, Sierra Leone and Liberia. Lower the 'Map fixed opacity' to 0.2 to enhance the contrast with the branch colors (cfr. infra). Set the size of the 'Points fixed radius' to 4 and also color the sampling locations by country via the 'Points color attribute'. Reselect the Lines layer. Setting the 'Lines color attribute' to 'country' colours the lines by destination location, and attracts the eye to cross-country movements. Reselect the Counts layer. Because the size of the polygons around a sampling location is proportional to the number of lineages that maintain that location, this captures the absolute and relative intensity of the local virus spread at any given point in time. For example, set the 'Counts fixed color' to the darkest red in combination with a low value for its opacity. After this, the figure should closely resemble Figure 1a in Bielejec et al. (2016) (Figure 3). Westward movements are depicted by lines with an upward curvature, while eastward movements are depicted by lines with a downward curvature.

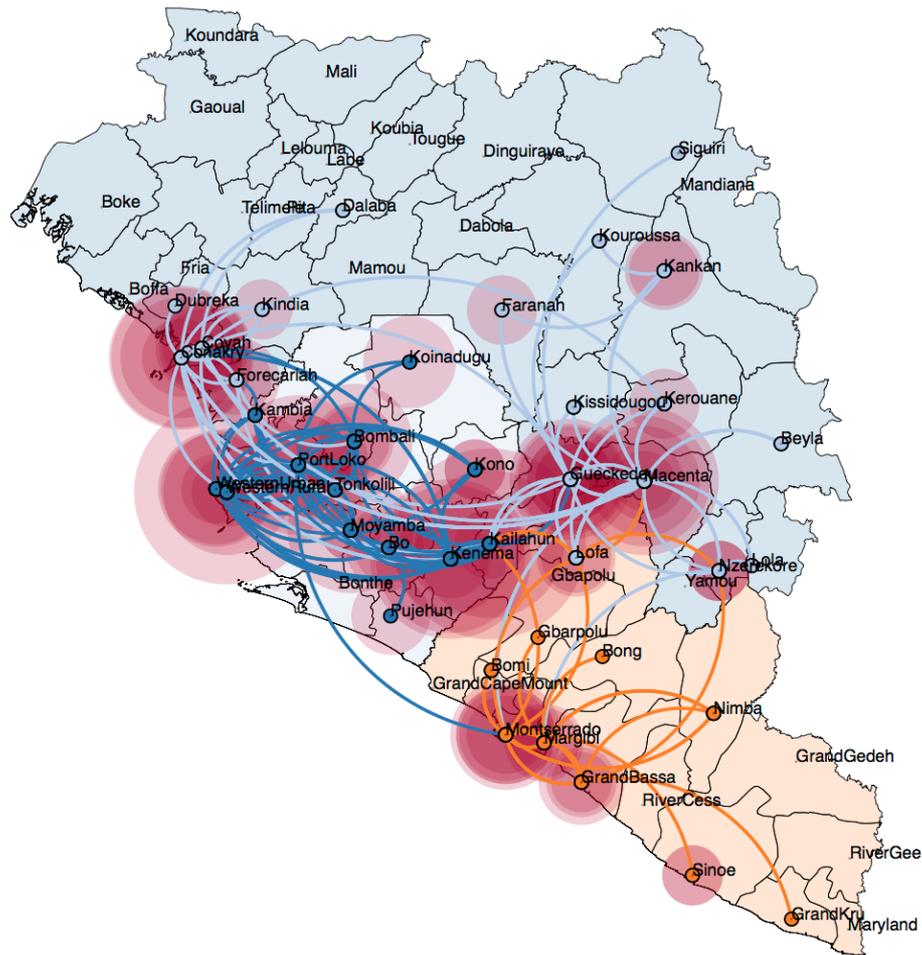


Figure 3 SpreaD3 generated image for the Ebola virus dataset.

SpreaD3 offers an export option to a vector graphics in a Scalable Vector Graphics (SVG) format. The easiest way to save the image to disk is by exporting (printing) the webpage in the required format (e.g. PDF/SVG) or using on-line tools like SVG crowbar (<https://nytimes.github.io/svg-crowbar/>). Any post-processing, if required, can then be handled by e.g. Adobe Illustrator or programs alike.

4.2 Identifying well-supported rates through Bayes factor tests

To identify well-supported rates between locations in standard discrete phylogeographic reconstructions we turn to a subset of the avian influenza type A H5N1 data that was also used by Lemey et al. (2009) in the original paper on Bayesian phylogeographic inference in BEAST. We will identify the subset of location exchange rates that dominate the diffusion process using the Bayesian stochastic search variable selection (BSSVS) procedure (Lemey et al., 2009).

In the BSSVS procedure, the standard phylogeographical model is augmented with a matrix of indicator variables $\delta_{i \rightarrow j}$. When the rate q between locations i and j is non-zero, the indicator variable $\delta_{i \rightarrow j}$ takes the value of 1, and otherwise $\delta_{i \rightarrow j}$ equals 0. The indicator variables are parameters of the model, and are hence also treated as random variables associated with statistical distributions (that is, the so-called priors). In the BSSVS procedure, the prior is a truncated Poisson prior with mean $\eta = \log 2$, which assigns 50% prior probability on the minimal rate configuration. For a model with K location states, this translates to a 50% prior probability on a model with $K - 1$ non-zero rates. Because of this, the prior expectation for each exchange rate $q_{i \rightarrow j}$ equals

$$q_k = (\eta + K - 1) / [K(K - 2) / 2].$$

Just as for other parameters in the model, the fraction of time the Markov chain Monte Carlo (MCMC) chain spends in a state $\delta_{i \rightarrow j} = 0$ or $\delta_{i \rightarrow j} = 1$ equals its steady state probability. In other words, the frequency with which $\delta_{i \rightarrow j}$ is included in the model in the sample of the stationary distribution equals the odds with which transitions between locations i and j help explain the observed distribution of locations at the tips of the tree, or equals the odds that $q_{i \rightarrow j}$ is non-zero. Intuitively it should be clear that if this frequency is higher than the prior expectation, location transitions between i and j help explain the diffusion process.

To distinguish between inclusion probabilities that are just a little higher than expected and those that indicate the concerning exchange rate really matters, we make use of Bayes factors.

Bayes factors provide a sense of how much the posterior (i.e. the outcome of the analysis) deviates from the prior beliefs, and are defined as the ratio of the posterior odds over the prior odds (Kass and Raftery, 1995). Due to the prior specification we can write:

$$\begin{aligned} & \text{Bayes factor for } q_{i \rightarrow j} = \text{posterior odds} / \text{prior odds} \\ & = [K(K - 2)/2](\text{posterior frequency } \delta_{i \rightarrow j} = 1)/(\eta + K - 1)(1 - \text{posterior frequency } \delta_{i \rightarrow j} = 1) \end{aligned}$$

which means we can directly obtain the Bayes factor support for each pairwise rate of diffusion between locations from samples collected from the posterior.

Parsing the data

To obtain the Bayes factors from a BSSVS analysis, select 'Log file from BSSVS analysis' in the Data tab (Figure 4, nr 1). Browse to the log file (H5N1_HA_discrete_rateMatrix.log) and open it to load it into SpreaD3 (Figure 4, nr 2). There are two options when specifying the location coordinates (Figure 4, nr 3). Either a set of longitude and latitude coordinates is passed, and these are afterwards coupled to the geoJSON formatted map that can be given along (Figure 4, nr 4), or the locations are represented by equally spaced points on a circle (this is most useful for traits that cannot be readily linked to another coordinate system) (Figure 4, nr 5). To set up the latter visualisation, select 'Generate' and enter the number of location states (Figure 4, nr 6), which is 7 for this example. The locations are now referred to as location1, location2, etc. Change this to the actual location names by updating the names in the same order as the location states are declared in the XML (the names should be passed in the same order as in the locationCoordinates_H5N1 file). Save as BF_circle.json. We'll also map the BSSVS outcome on a world map, for which you'll have to load 'locationCoordinates_H5N1.txt', and the provided geoJSON map (world.geojson). Save this as BF_world.json. In addition to the *.json files, SpreaD3 creates a text file with the Bayes factors for all possible transitions between locations. For this example these are named BF_circle.json.txt and BF_world.json.txt.

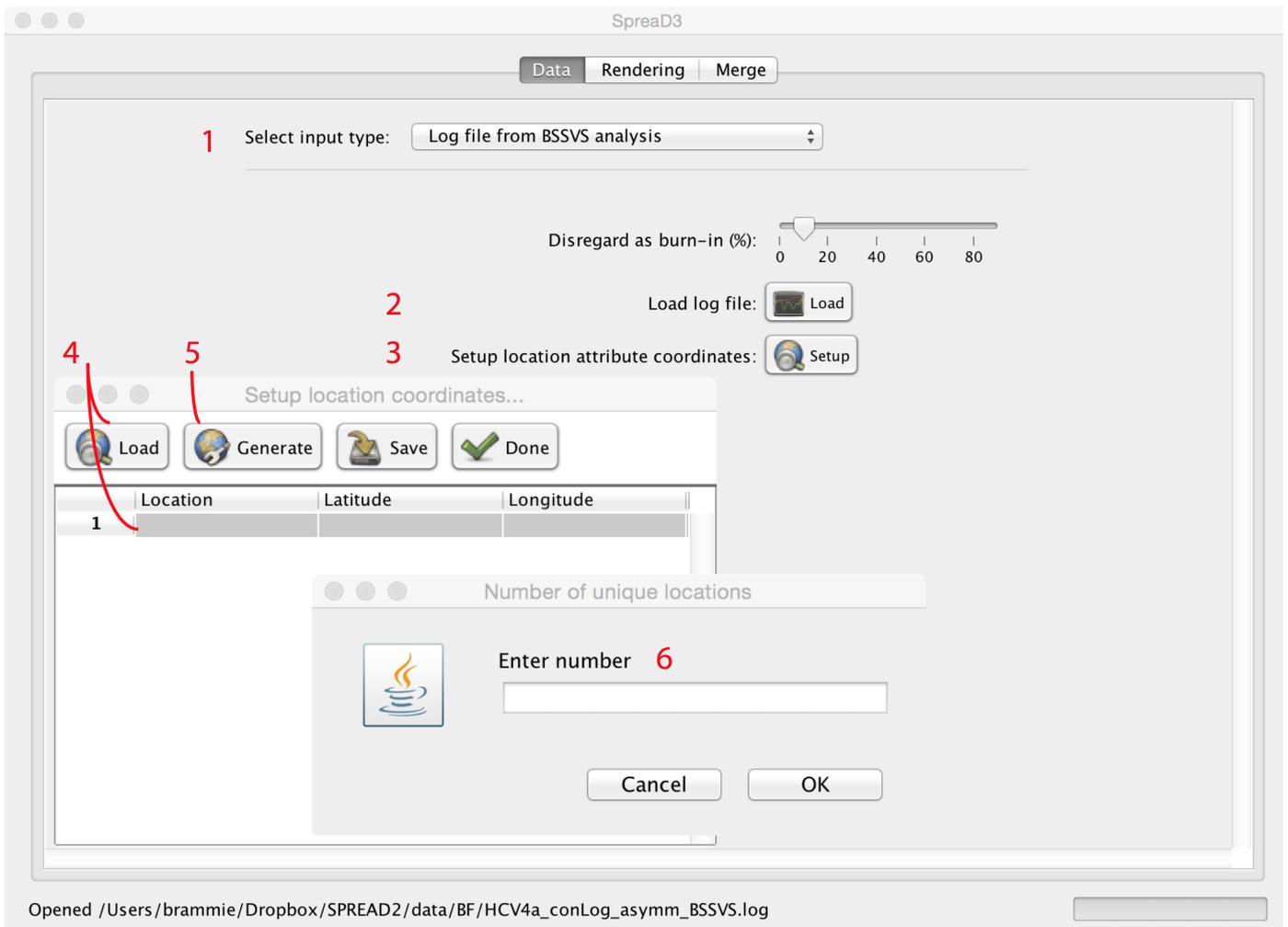


Figure 4 Parsing options for the BSSVS analysis.

Rendering and specifying the visualisation options

Open the Rendering tab and load BF_circle.json. Change the Lines color attribute start color to light blue (color code #abd9e9) and the end color to red (color code #d73027). The changes take effect after toggling between 'posteriorProbability' and 'bayesFactor'. The location dots can be better contrasted from the lines by setting the Points fixed color to dark blue (color code #4575b4), and adjusting the Points fixed radius to 5. By leaving the Lines cut-off to its default value it is possible to visualise the Bayes factor support for all location transitions. To include the legend in the image, print the page and copy-paste the legend in the image with e.g. Illustrator. In the default plot the location names may be overlaid with the lines; this too can be easily adjusted in e.g. Illustrator. The result should look similar to Figure 5.

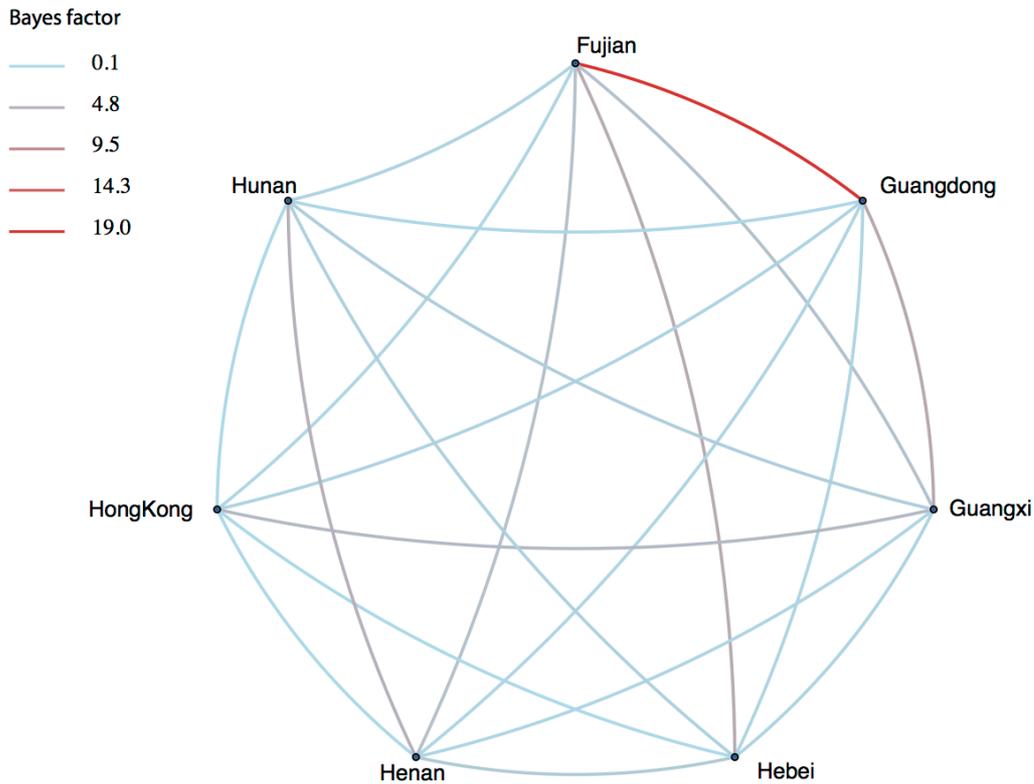


Figure 5 Circle representation of the Bayes factor support for the possible location transitions for the H5N1 dataset.

To plot the Bayes factor support on a geographical map, again start by selecting an appropriate background color. Give China a different than its neighbouring countries by setting the Map fill attribute to 'name'. For this example let's use the same colour scheme for the lines and points as above. It is possible to lay focus on the significant location transitions by changing the Line cut-off to e.g. 4. Including the legend in the same way as above should result in a similar image as Figure 6.

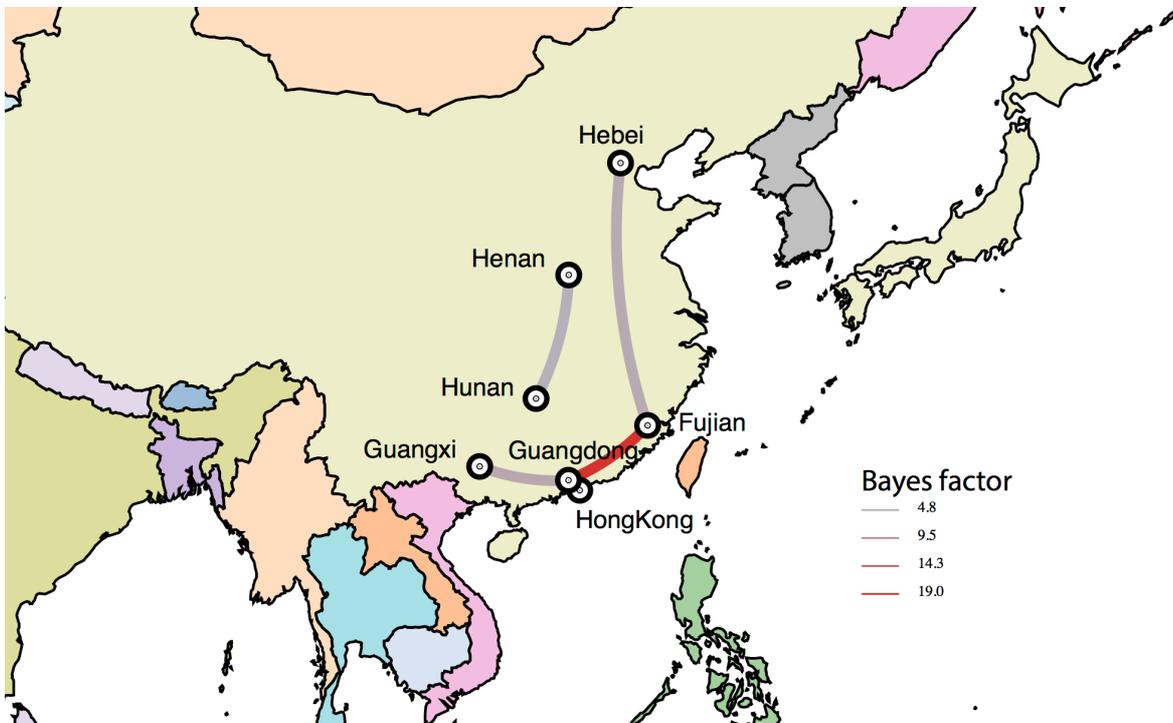


Figure 6 Visualisation of the most significant location transitions for the H5N1 dataset overlaid on a geographical map.

4.3 Visualizing an MCC tree annotated with continuous traits

For the example using an MCC tree annotated with continuous traits we revisit the work of Bouckaert et al. (2012) in which the origin and spread of Indo-European languages was investigated with well-established phylogenetic methods. We will here use the MCC tree summary to visualise their origin in and spread from Anatolia (a region in present day Turkey). This maps all branches of a continuous diffusion phylogeographic reconstruction and allows plotting the uncertainty of geographic coordinates at the internal nodes through their annotated highest posterior density contours.

Parsing the data

In the Data tab, select 'MCC tree with CONTINUOUS traits' and browse to where the example tree is saved (Figure 7, nr 1). Because the latitude and longitude are by default annotated as 'trait1' and 'trait2' by BEAST, we adhere to this ordering and first provide the y-coordinate (latitude, trait1), and then the x-coordinate (longitude, trait2) (Figure 7, nr 2). The option 'External annotations' is to be used when the child node trait values have an associated uncertainty (Figure 7, nr 3). Because the locations at the tips in this example cover the areas where the language is spoken, this option should be selected. On the other hand, one can safely ignore this option whenever the tips traits have no associated uncertainty (e.g. exact longitude and latitude). When the MCC tree is annotated with multiple levels of trait uncertainty (that is, multiple HPD levels), the level of interest can be chosen (Figure 7, nr 4). The provided example tree only has 1 HPD level annotation (80%). You can leave the most recent sampling date for this example to its default value, just as the Time scale multiplier (Figure 7, nr 5). To conclude the parsing step, select the world.geojson file and generate 'language.json' (Figure 7, nr 6 and 7).

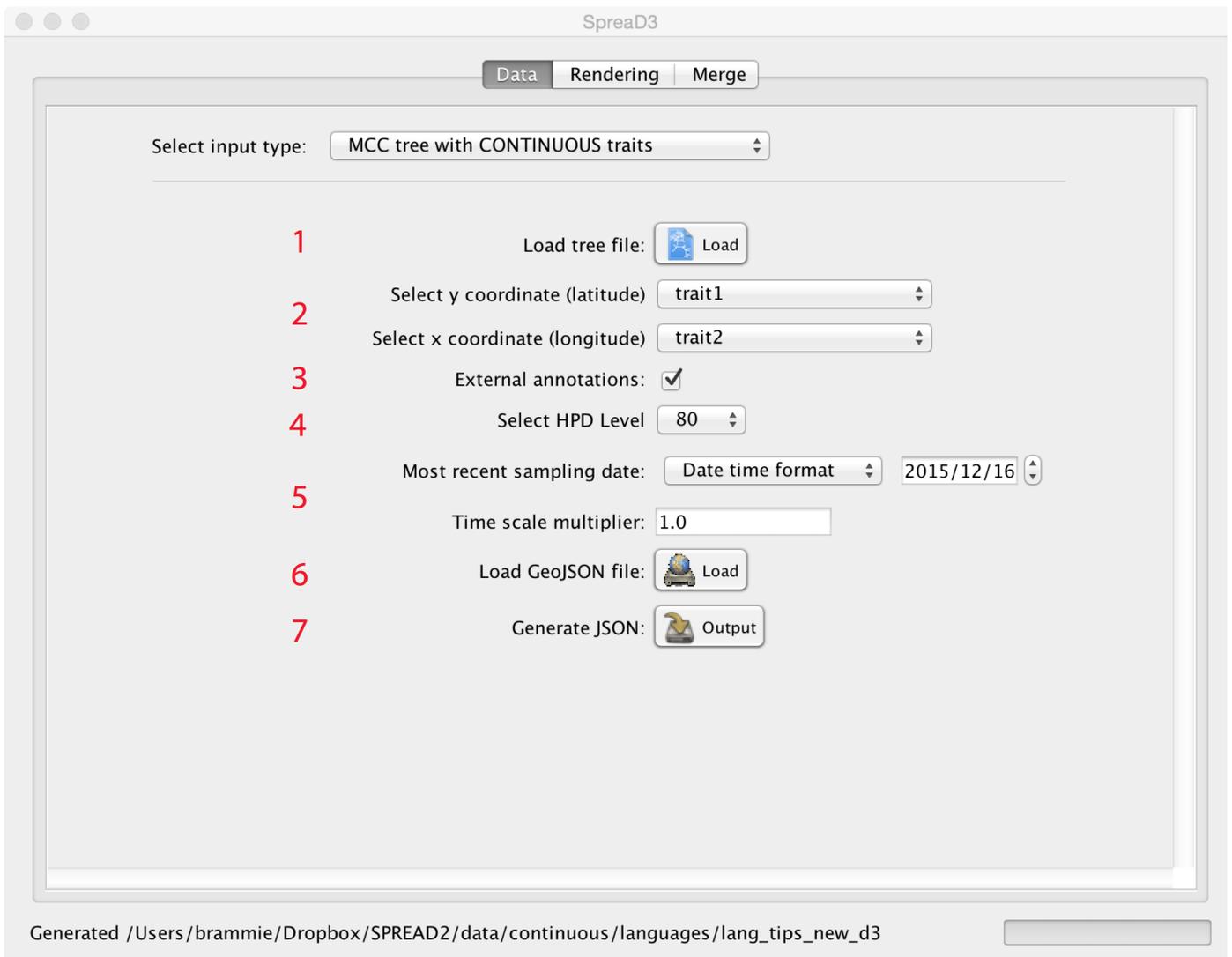


Figure 7 Parsing options for the 'MCC tree with continuous traits' analysis.

Rendering and specifying the visualisation options

Open the Rendering tab, load language.json and save as language_D3 (Figure 8). As before, we start by giving the geographical map on which the tree is projected an appropriate color. Let's keep the white color for the seas, and contrast this with grey for the land masses. For this, set the Map color to #000000 and the Map fixed opacity to 0.2. Note that, depending on the loaded geoJSON map, this may also be achieved by selecting the appropriate Map fixed attribute. Next, color the Points and Lines according to the time. For this, set the Point/Line color attribute to height (measured from the tips to the root), and the start and end color to respectively #fee090 and #74add1. Finally, let's visualise the uncertainty on the internal nodes by setting the Polygon color attribute to 'posterior'. By setting the start color to dark red (color code #a50026) and the end color to blue (color code #74add1), nodes with higher posterior support will be surrounded by progressively more intense blue polygons. To include the legend in the image, print the web page and copy-paste the legend in the image with e.g. Illustrator (Figure 9).

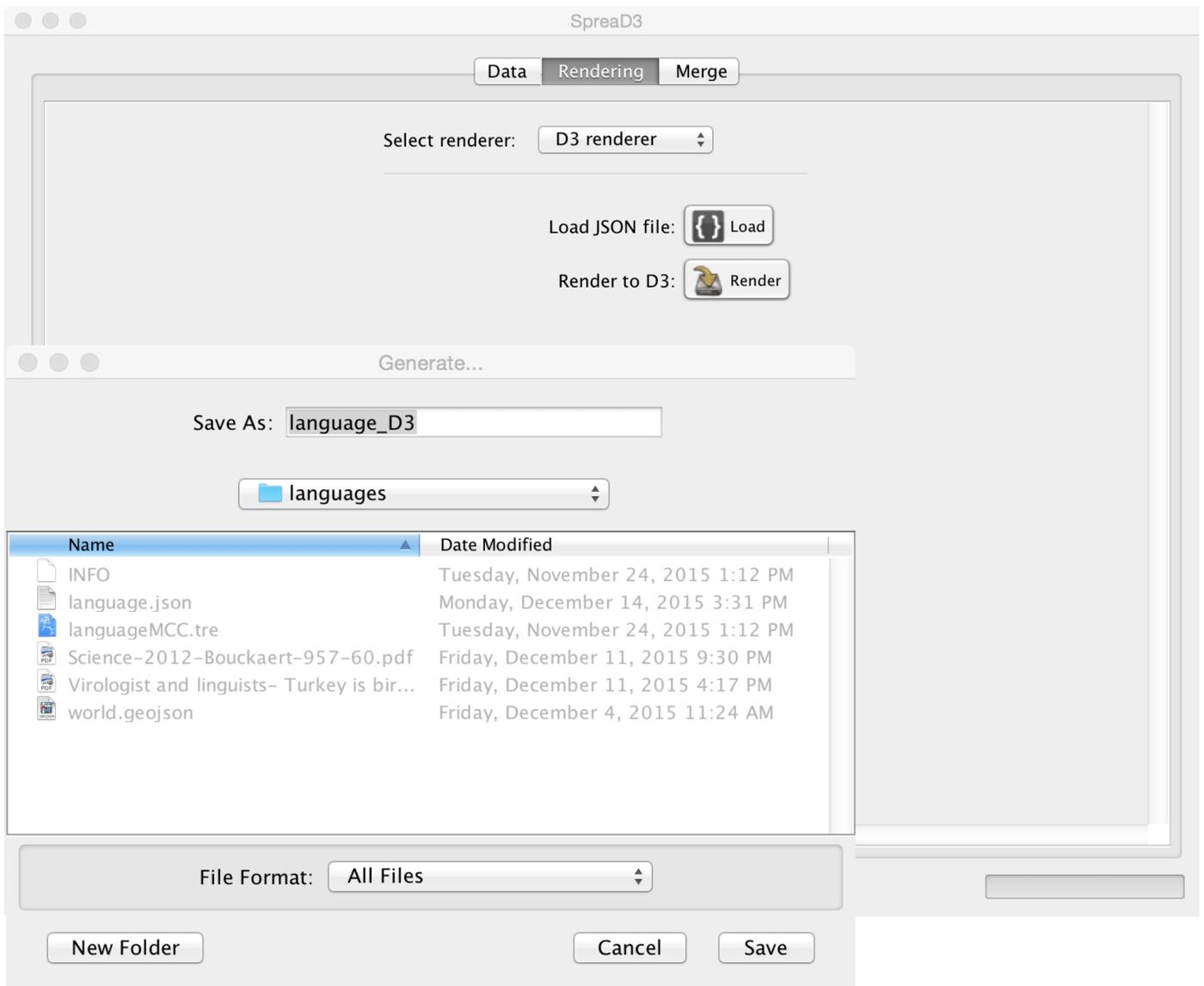


Figure 8 Rendering options for the 'MCC tree with continuous traits' analysis.

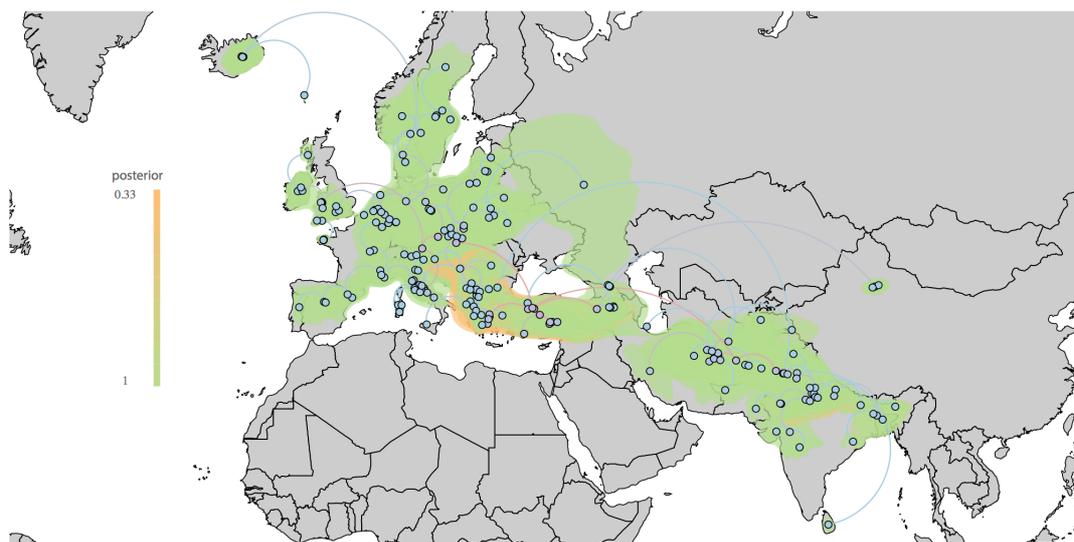


Figure 9 SpreaD3 generated image of the language dataset from Bouckaert et al. (2012).

4.4 Visualizing a distribution of trees annotated with continuous traits

For this example we use the West Nile virus dataset previously analysed by Pybus et al. (2012). This analysis allows one to summarise and visualise the full posterior distribution of trees obtained in continuous phylogeographic analysis. To achieve this, SpreaD3 creates a time line by slicing through each phylogeny at particular points in time, and imputes the unobserved descendant locations for those

ancestral infections. Uncertainty on these estimates is reflected by contouring them with polygons, which provides a natural representation of the uncertainty in these inferences.

Parsing the data

Select 'Tree distribution with CONTINUOUS traits' from the Data tab, load the trees, set the 2D trait to 'location' and load the MCC tree summary (Figure 10, nr 1, 2 and 3). The time slices are based on the MCC tree length, but it is also possible to supply a file with custom slice heights in a single column and in ascending order (from the tips to the root of the phylogeny) . For this, set 'time slices' to Custom (Figure 11, nr 1) and load the time slice file (Figure 11, nr 2). The most recent sampling time in this dataset is 2007.63 - reset the date format to decimal (Figure 10, nr 4). The trees distribution has been subsampled with LogCombiner (a tool distributed with BEAST) to reduce its size, at which stage the burnin was removed. Hence, set the burnin to zero (Figure 10, nr 5). Leave the other options at their default values, select the provided geojson map (Figure 10, nr 6) and generate wnv_trees.json. This may take a few moments for large trees-files.

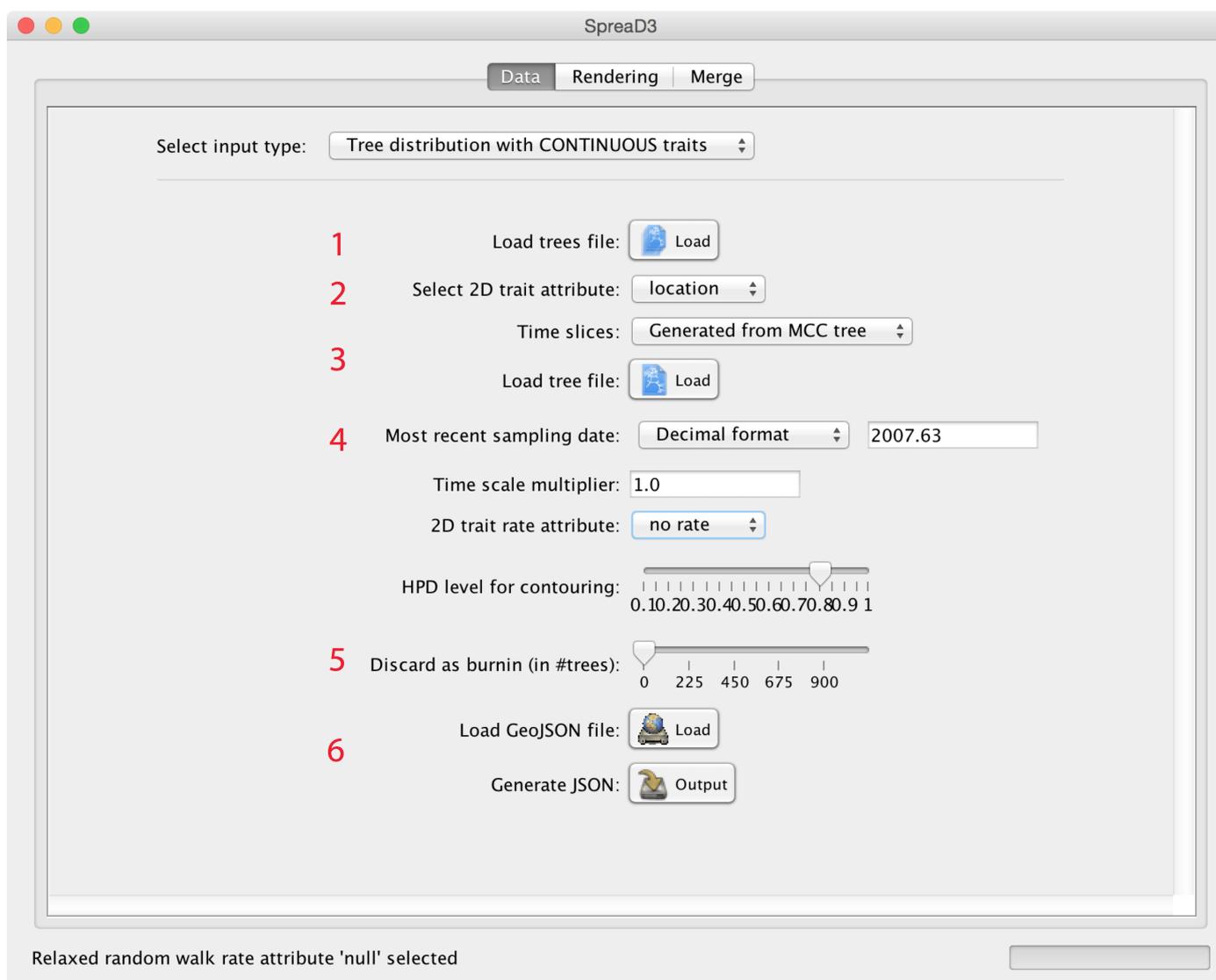


Figure 10 Parsing options for the 'Tree distribution with continuous traits' analysis.

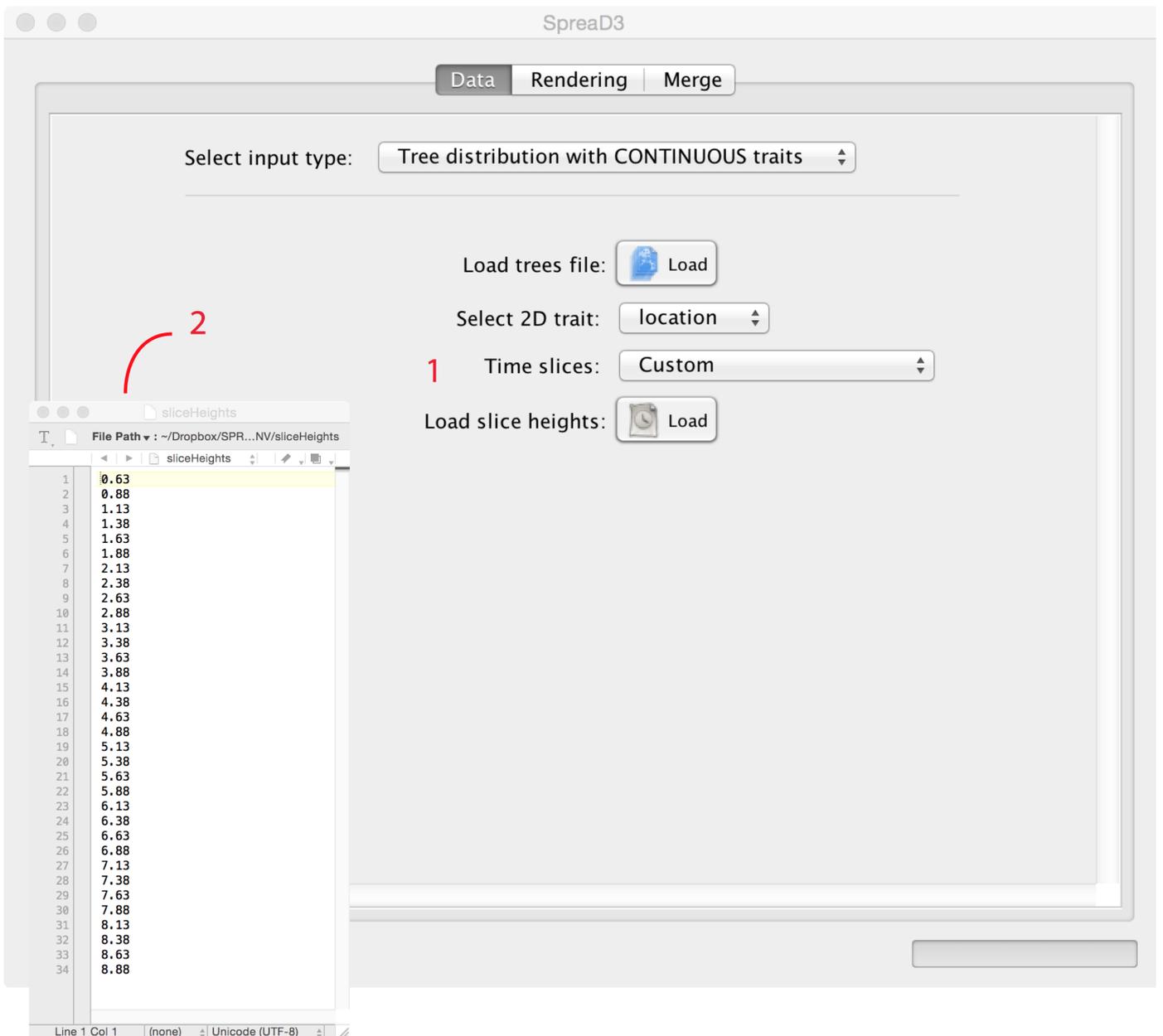


Figure 11 Specifying custom time slices.

We also visualise the MCC tree which gave rise to the time slices by drawing its branches. For this, parse the West Nile virus MCC tree as before and save as `wnv_mcc.json` (Figure 12). Before rendering, the time slice JSON file and the MCC tree JSON file need to be combined. Open the 'Merge' tab and double click in the 'Choose file. . . ' box to browse to and load the JSON files (Figure 13, nr 1). To add a JSON file, select the '+' button in the bottom left corner of the screen (Figure 13, nr 2). Keep the Points and Lines layers from `wnv_cc.json`, and the other layers from `wnv_trees.json` (Figure 13, nr 3). To save the merged file as `wnv_combined.json`, select File → Merge.

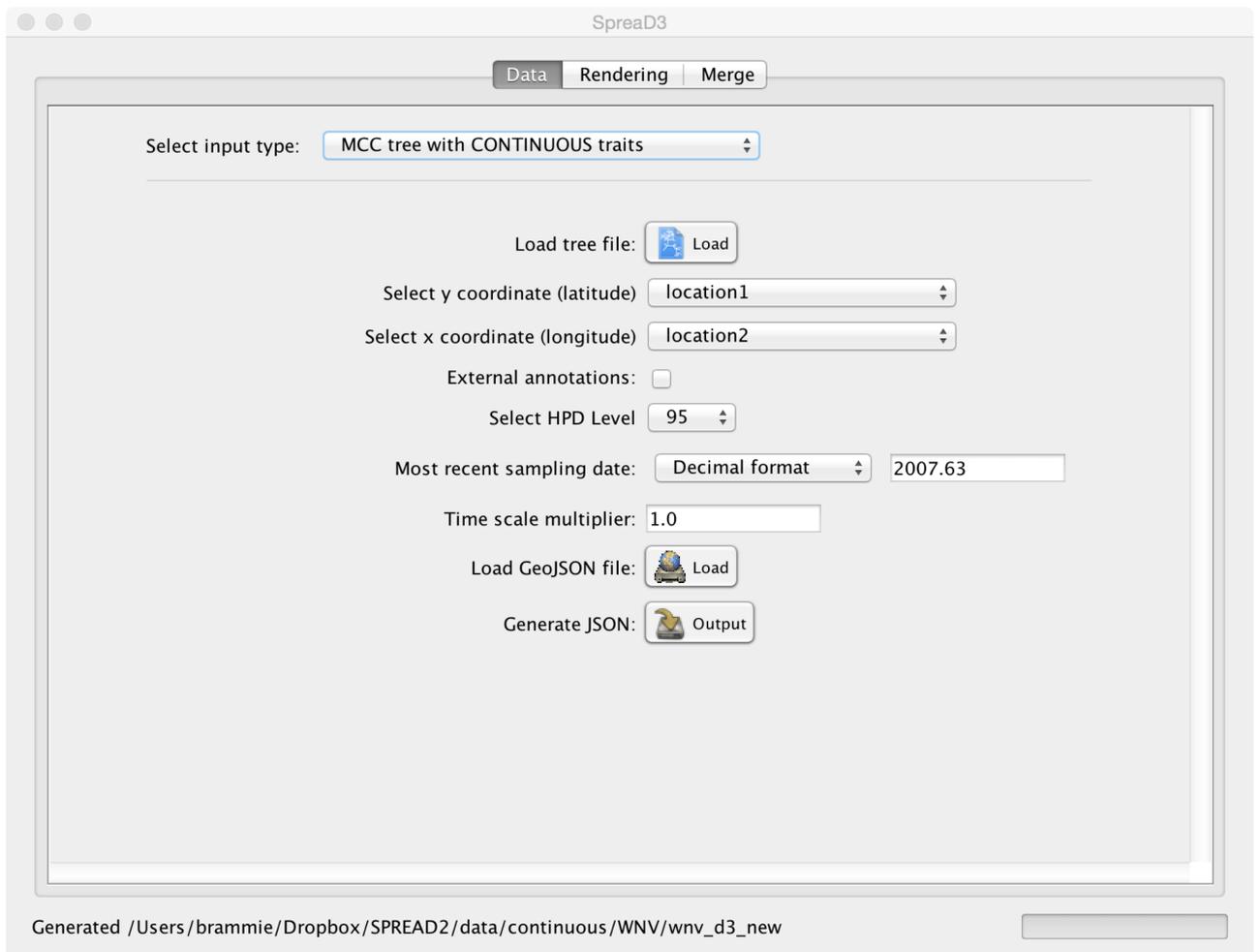


Figure 12 Parsing options for the 'MCC tree with continuous traits' analysis.

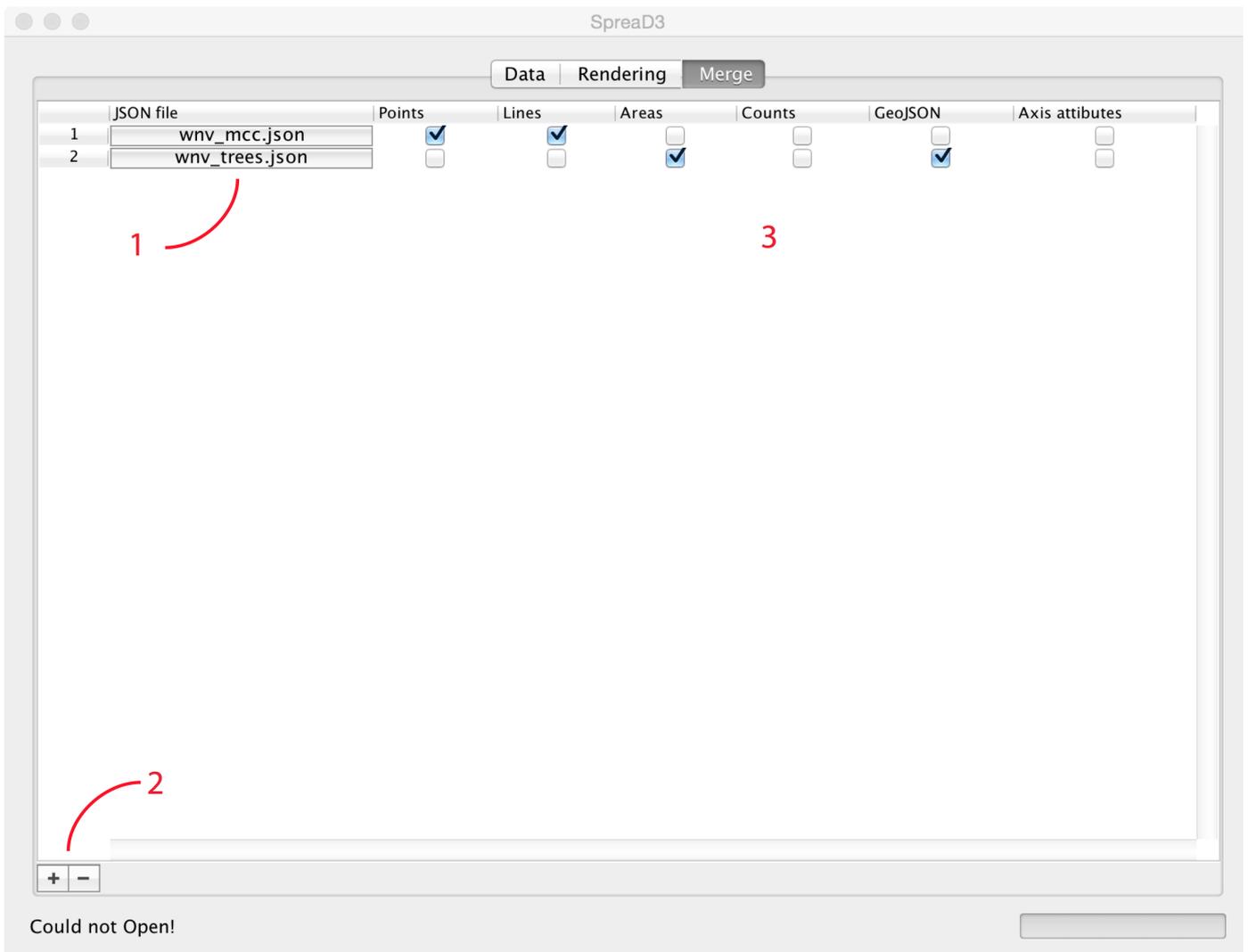


Figure 13 Example of merging the desired layers of various SpreaD3 parsing outputs.

Rendering and specifying the visualisation options

There are only a few steps required to arrive at the final image for this example (Figure 14). First give the seas a grey colour by setting the Background color to grey. Next, colour the land mass dark grey by setting the Map color to black (colour code #000000) and the Map fixed opacity to .5. Update the colour of the contours: set the Polygon fixed color to #74add1 and the Polygon opacity to .4. Finally, color the points and lines by the time by setting the Points/Lines color attribute to 'height', and their start and end colours to #ffffbf and #d73027 respectively. Finally, reset the Lines width to 0.5 and the Points fixed radius to 1.

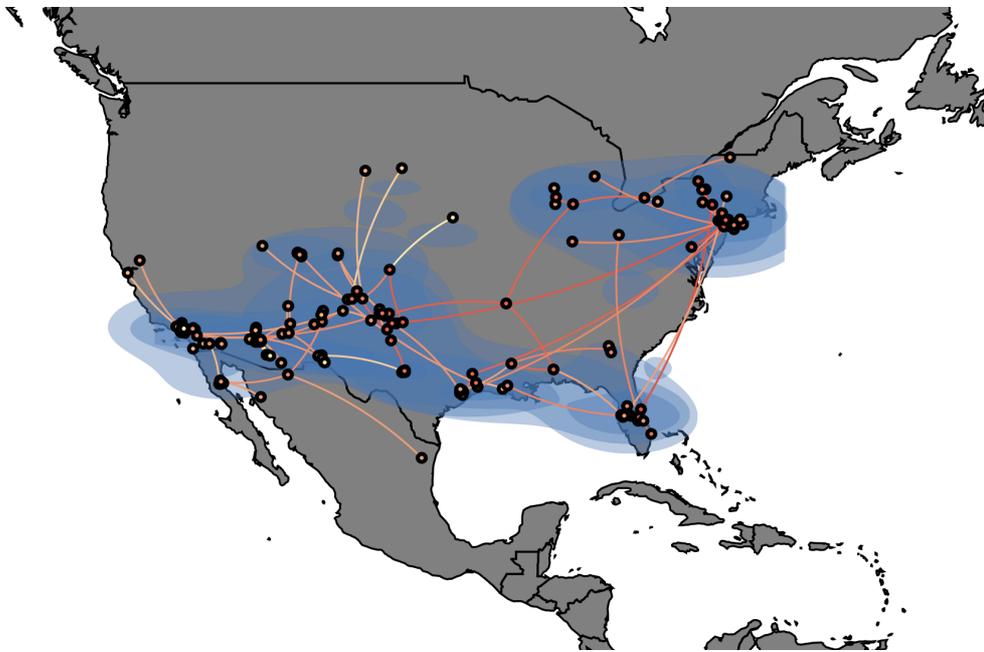


Figure 14 SpreaD3 generated image of the West Nile virus dataset from Pybus et al. (2012).

4.5 Tips & tricks

We illustrate the use of SpreaD3 for data mapped to other coordinates than longitude and latitude with an antigenic coordinate ancestral reconstruction for an influenza A/H3N2 virus dataset taken from Bedford et al. (2014). Antigenic distances are a quantitative measure of how antigenically dissimilar virus strains are, and are projected on a 2D map. When no underlying map is provided, SpreaD3 simply draws a bounding box around the data points. The idea here is to visualise three layers of uncertainty by combining the polygons of the 50%, 75% and 95% Highest Posterior Density interval.

Parsing the data

The parsing is fully analogous to the parsing explained in 4.3.1, with the exception that the x and y coordinates now are antigenic1 and antigenic2 respectively. Don't forget to select 'External annotations' (Figure 15, nr 1 and 2). Save the files as 'H3N2_95.json', 'H3N2_75.json' and 'H3N2_50.json' (one output file per HPD annotation). Now combine these files (Figure 16) and save as 'H3N2_combined.json'.

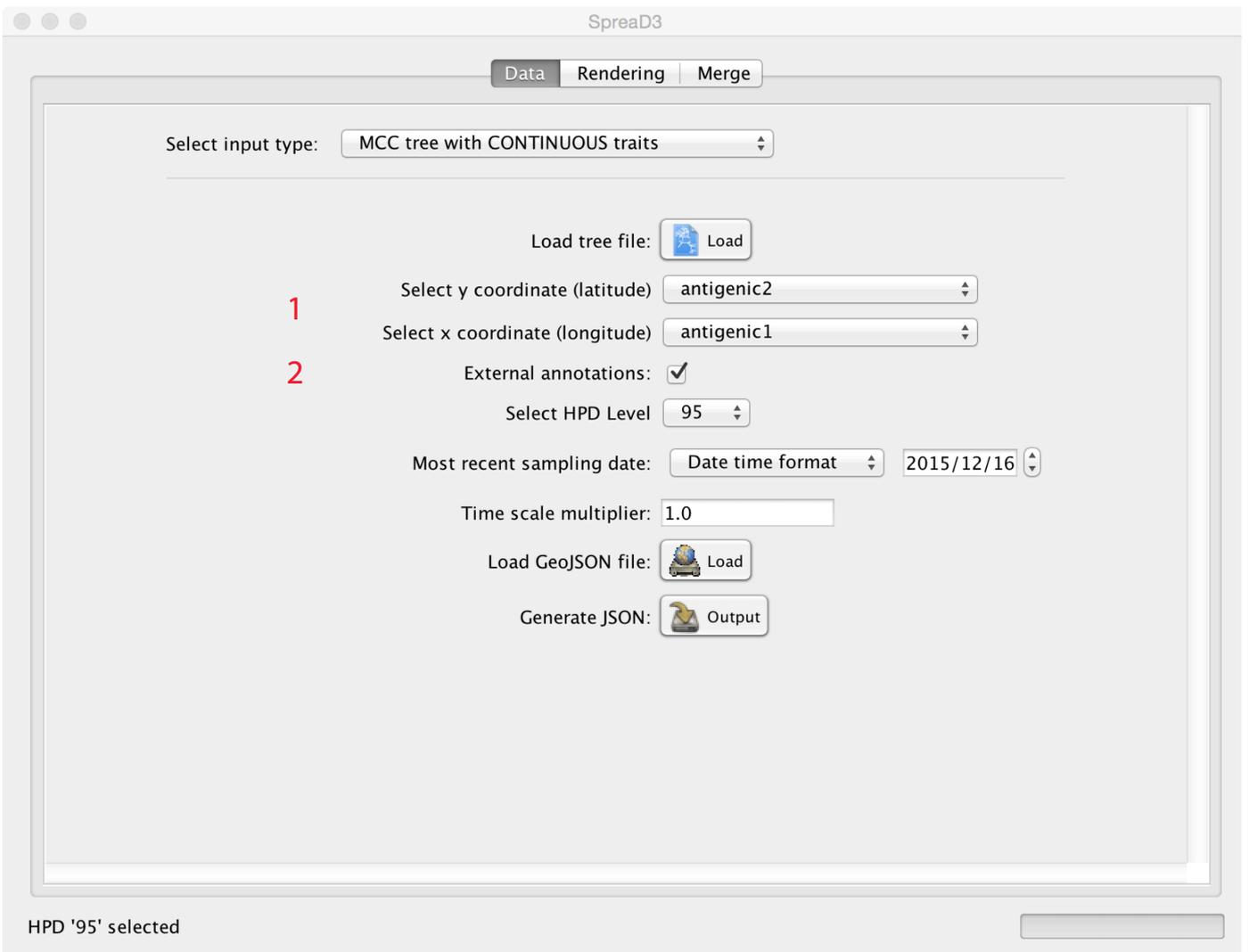


Figure 15 Parsing options for the 'Tree distribution with continuous traits' analysis.

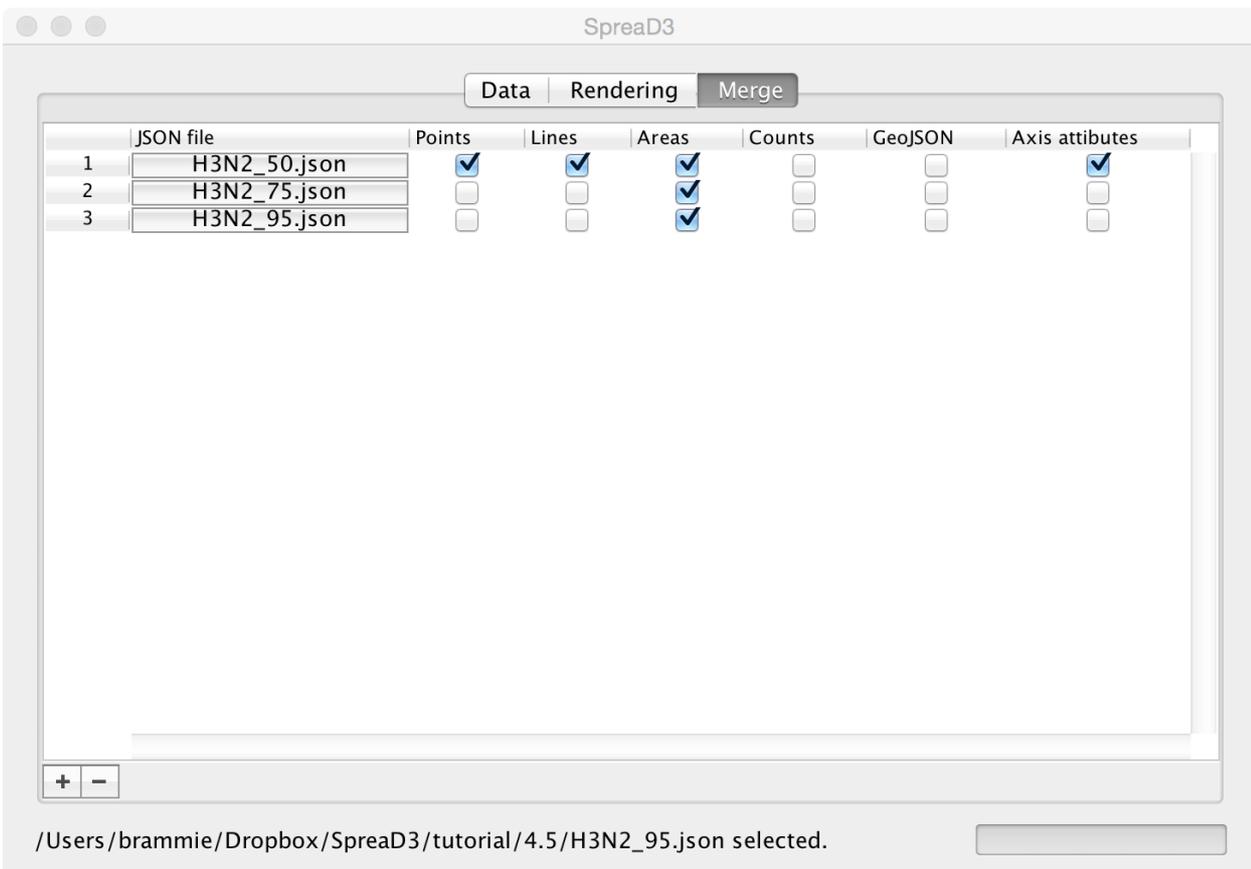


Figure 16 Example of a merging the desired layers of various SpreaD3 parsing outputs.

Rendering and specifying the visualisation options

Do the following to obtain Figure 17. Set the Line color attribute to antigenic1 and its start and end colors to #4575b4 and #f46d43 respectively. Do the same for the Points color attribute. The HPD layers, which take 3 discrete values, can be given a separate colour by setting the Polygon attribute color to HPD. Lower the Polygon opacity to .1 to visualise the contours of the wider HPDs.

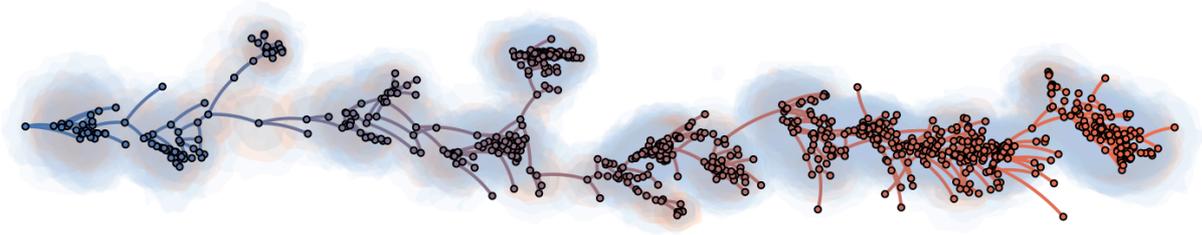


Figure 17 The antigenic coordinate map with three layers of uncertainty (50%, 75% and 90% HPD intervals).

4.6 KML rendering

The previously generated JSON files can also be used to render the images in KML. As an example, open ebov.json and select 'render as KML'. Several panels that enable the user to fine-tune the appearance of the elements used to represent all facets of the phylogeny and its associated traits appear (in some browsers the window needs to be resized a little to make these visualisation options appear) (Figure 19). The output of this rendering can be viewed in virtual globe software (e.g. Google Earth).

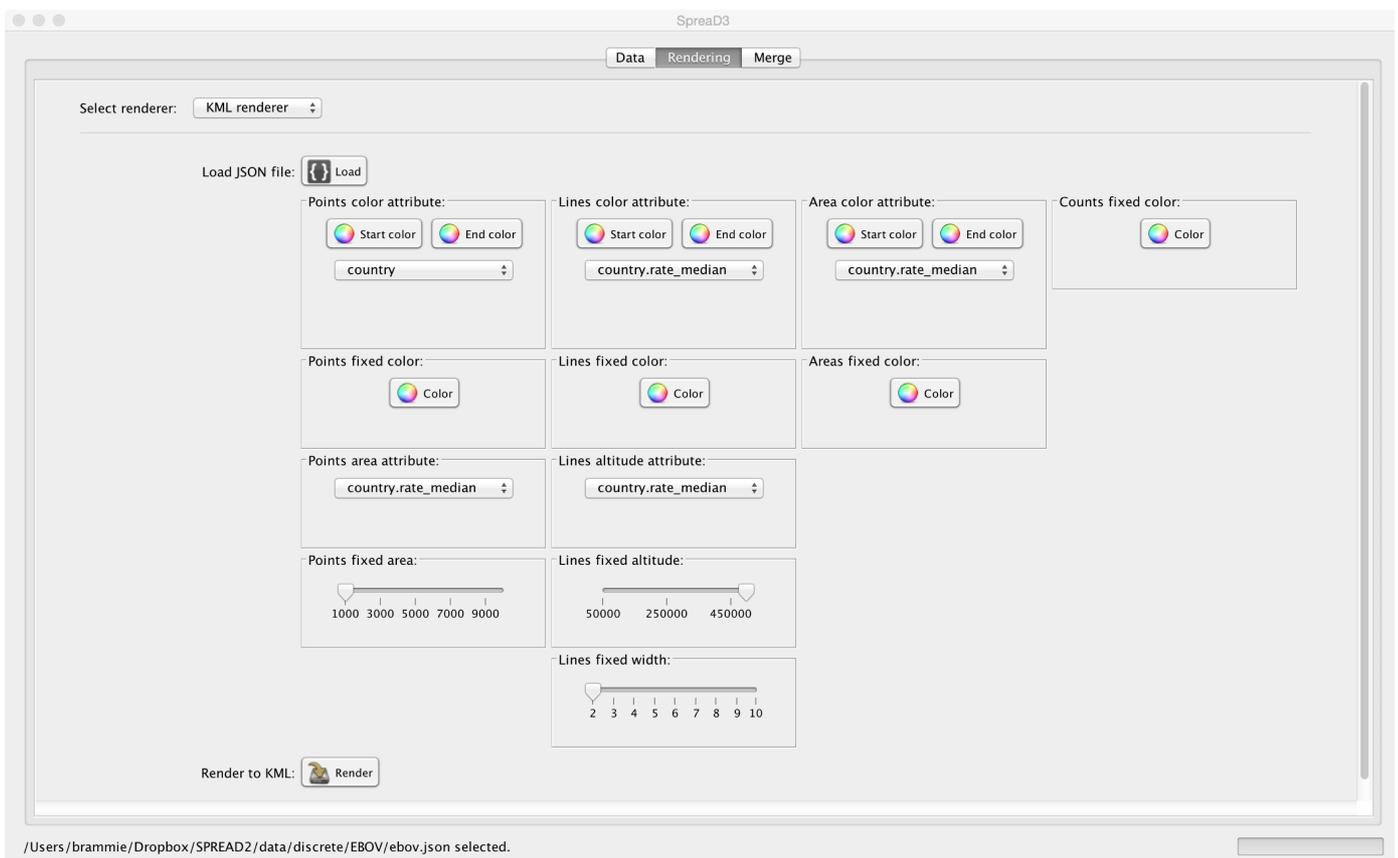


Figure 19 The visualisation options when rendering to KML.

4.7 Command line interface

We have found it easiest to use the command line interface with an alias added to .bash_profile:

```
alias spread='java -jar <absolute path to Spread3.jar>'
```

The alias will become active when you open another session in Terminal, or it can be used in the current session after executing

```
$ source ~/.bash_profile
```

in the current session [assuming your bash_profile resides at \$HOME].

To parse trees, the following syntax should be used:

A) discrete traits, with map:

```
$ spread -parse -locations locations.txt -header false -tree some.tree -locationTrait location -intervals 10 -mrsd 2015.8 -geojson some.geojson -output output.json
```

B) discrete traits, without map:

```
$ spread -parse -locations locations.txt -header false -tree some.tree -locationTrait location -intervals 10 -mrsd 2015.8 -output output.json
```

C) Bayes factor, with map:

```
$ spread -parse -locations locations.txt -log rateMatrix.log -burnin 10 -geojson some.geojson -output out-put.json
```

D) continuous trait MCC tree, without map:

```
$ spread -parse -tree some.tre -yCoordinate latitude -xCoordinate longitude -HPD 95 -externalAnnotations true -mrsd 2011.1 -output output.json
```

E) continuous trait MCC tree, with map:

```
$ spread -parse -tree some.tre -yCoordinate latitude -xCoordinate longitude -HPD 95 -externalAnnotations true -mrsd 2011.1 -geojson some.geojson -output output.json
```

F) continuous trait posterior tree distribution, with slices according to MCC tree:

```
$ spread -parse -trees some.trees -tree some.tre -trait location -externalAnnotations true -mrsd 2011.1 -geojson some.geojson -output output.json
```

G) continuous trait posterior tree distribution, with predefined slices:

```
$ spread -parse -trees some.trees -tree some.tre -trait location -sliceHeights.txt someHeights.txt -externalAnnotations true -mrsd 2011.1 -geojson some.geojson -output output.json  
#to override some default settings: -intervals 15 -burnin 5 -hpd 0.8
```

To merge JSON files, the following syntax should be used:

```
$ spread -merge -axisAttributes some.json -points some.json -lines some.json -areas some.json -geojson some.json -output merged.json  
#N.B. one or more JSON files can be specified for each option.
```

To render, the following syntax should be used:

```
# D3: $ spread -render d3 -json some.json -output rendered  
# KML: $ spread -render kml -json some.json -output rendered.kml
```

Parsing and rendering options can also be set via the command line (see Table 1 for an overview of the syntax and a concise description).

Table 1: Overview of available command line visualisation options

Points

color:

override default points color	-pointColor 0 255 255
map point colors to a continuous attribute	-pointColor 0 255 255 -

override default start-end colors

pointColorMapping
attribute

map point colors to a discrete attribute:

- same color for all states

-pointStartColor 250 0 0

- state-specific colors: via color sheet (a file with per line only the trait state name and its color code, separated by a tab; the colors should be supplied in RGB or RGBA values)

-pointEndColor 0 0 150

-pointColor 0 255 255 -

pointColorMapping
attribute

-pointcolors

traitColors.txt

pointColorMapping

attribute

area:

override default points area

-pointArea *number*

map point areas to a continuous attribute

-pointAreaMapping *attribute*

map point areas to a discrete attribute

-pointAreaMapping *attribute*

Lines

color:

adjust width

-lineWidth *number*

override default lines color

-lineColor 250 0 0

map line colors to a continuous attribute

-lineColorMapping
attribute

override default start-end colors

-lineStartColor 250 0 0 -

map line colors to a discrete attribute:

lineEndColor 0 0 150

- same color for all states

- state-specific colors: via color sheet (a file with per line only the trait state name and its color code, separated by a tab; the colors should be supplied in RGB or RGBA values)

-lineColor 250 0 0 -

lineColorMapping
attribute

-linecolors

traitColors.txt

lineColorMapping

attribute

width:

override default lines width -lineWidth *number*

altitude (curvature):

override default lines altitude

-lineAltitude *number*

map line altitude to a continuous attribute

-lineAltitudeMapping *attribute*

map line altitude to a discrete attribute

-lineAltitudeMapping *attribute*

Counts

color:

override default count color `-countColor 0 0 250`

Polygons

color:

override default area color	<code>-areaColor 0 0 250</code>
map area colors to a continuous attribute	<code>-areaColorMapping attribute</code>
override default start-end colors	<code>-areaStartColor 250 0 0 - areaEndColor 0 0 150</code>
map area colors to a discrete attribute: - same color for all states - state-specific colors: via color sheet (a file with per line only the trait state name and its color code, separated by a tab; the colors should be supplied in RGB or RGBA values)	<code>-areaColor 0 255 255 - areaColorMapping attribute -areacolors traitColors.txt - areaColorMapping attribute</code>

Citing SpreaD3

We have invested a lot of time and effort in creating SpreaD3, please cite it when using it in your research. BibTeX entry:

```
@article{Bielejec23042016,  
author = {Bielejec, Filip and Baele, Guy and Vrancken, Bram and Suchard, Marc A. and Rambaut, Andrew and Lemey, Philippe},  
title = {SpreaD3: interactive visualisation of spatiotemporal history and trait evolutionary processes},  
year = {2016},  
journal = {Mol. Biol. Evol.},  
doi = {10.1093/molbev/msw082},  
}
```

References

Bedford, T., Suchard, M. A., Lemey, P., Dudas, G., Gregory, V., Hay, A. J., McCauley, J. W., Russell, C. A., Smith, D. J., and Rambaut, A. (2014). Integrating influenza antigenic dynamics with molecular evolution. *Elife*, 3:e01914.

Bielejec, F., Baele, G., Vrancken, B., Suchard, M.A., Rambaut, A., and Lemey, P. (2016). SpreaD3: interactive visualisation of spatiotemporal history and trait evolutionary processes. *Mol. Biol. Evol.* (*in press*).

Bostock, M., Ogievetsky, V., and Heer, J. (2011). D3; data-driven documents. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2301–2309.

Bouckaert, R., Lemey, P., Dunn, M., Greenhill, S. J., Alekseyenko, A. V., Drummond, A. J., Gray, R. D., Suchard, M. A., and Atkinson, Q. D. (2012). Mapping the origins and expansion of the indo-european language

family. *Science*, 337(6097):957–60.

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.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *J. Am. Stat. Assoc.*, 90(430):773– 795.

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

Pybus, O. G., Suchard, M. A., Lemey, P., Bernardin, F. J., Rambaut, A., Crawford, F. W., Gray, R. R., Arinaminpathy, N., Stramer, S. L., Busch, M. P., and Delwart, E. L. (2012). Unifying the spatial epidemiology and molecular evolution of emerging epidemics. *Proc. Natl. Acad. Sci. USA*, 109(37):15066–71.
