

A Bayesian approach to phylogeographic clustering

Ioanna Manolopoulou^{1,*}, Lorenza Legarreta^{2,†}, Brent C. Emerson³,
Steve Brooks⁴ and Simon Tavaré⁵

¹*Duke University, Durham, USA*

²*University of East Anglia, Norfolk, UK*

³*IPNA-CSIC, La Laguna, Tenerife, Spain*

⁴*Select Statistics, Exeter, UK*

⁵*University of Cambridge, Cambridge, UK*

Phylogeographic methods have attracted a lot of attention in recent years, stressing the need to provide a solid statistical framework for many existing methodologies so as to draw statistically reliable inferences. Here, we take a flexible fully Bayesian approach by reducing the problem to a clustering framework, whereby the population distribution can be explained by a set of migrations, forming geographically stable population clusters. These clusters are such that they are consistent with a fixed number of migrations on the corresponding (unknown) subdivided coalescent tree. Our methods rely upon a clustered population distribution, and allow for inclusion of various covariates (such as phenotype or climate information) at little additional computational cost. We illustrate our methods with an example from weevil mitochondrial DNA sequences from the Iberian peninsula.

Keywords: migration; coalescent; subdivided population; island model; Markov chain Monte Carlo; reversible jump

1. INTRODUCTION

Phylogeographic methods have attracted a lot of attention in recent years, stressing the need to provide a solid statistical framework for many existing methodologies so as to draw statistically reliable inferences. The variety of available methods reflects both the numerous different objectives at which each phylogeographic analysis aims [1–6], as well as the disagreements and misconceptions about the fundamental principles of model-based inference [7–11]. Several different paths may be taken in phylogeographic inference; some of the key choices include the evolutionary model to use for the taxa under study, which statistical frameworks allow for efficient inferences and how population and geographical distributions should be combined [12].

Here, we take a flexible fully Bayesian approach by reducing the problem to a clustering framework, whereby the population distribution can be explained by a set of migrations that form geographically stable population clusters. These clusters are such that they are consistent with a fixed number of migrations on the corresponding (unknown) subdivided coalescent tree. In other words, a simplified island migration model between geographically stable populations with equal migration rates and no back-migration is considered, with an unknown number of migrations. Although we consider a simplistic evolutionary model

in which only a finite, but very large, number of coalescent trees have positive probability (details in appendix A), the methods can easily be extended to include more sophisticated models, also allowing for ancestral inference [13]. Our methods rely upon a clustered, as opposed to clinal, population distribution [14], and allow for flexible inclusion of various covariates such as phenotype or climate information at little additional computational cost.

We illustrate our statistical model on a set of synthetic datasets and one real dataset, and describe algorithms for inferring the underlying parameters. Owing to the unknown number of migrations and the size of the discrete sample space of both the tree and the clustering, an efficient reversible jump Markov chain Monte Carlo (RJMCMC) [15] sampler is necessary in order to obtain posterior samples for the parameters. We implement our methods on an example from weevil mitochondrial DNA sequences from the Iberian peninsula.

2. UNCERTAINTY ABOUT THE HAPLOTYPE TREE

One of the challenges of comparative approaches lies in combining geographical and ancestral history information. In this paper, we focus on haplotype trees, which are known to be reliable only in situations with little homoplasy and low mutation rate [16]; however, our clustering methods are naturally applicable to other ancestral representations such as coalescent

*Author for correspondence (im30@stat.duke.edu).

†Present address: 61 Henley Way, Ely, Cambridgeshire, CB7 4YH, UK.

One contribution of 9 to a Theme Issue 'Inference in complex systems'.

trees. Many of the earlier approaches relied upon one, or very few, fixed haplotype trees representing ancestral histories [17,18]. However, there is often considerable uncertainty about the haplotype tree, and the space of such trees is discrete and infinite. At the same time, as has been identified previously [17,19], valuable information about the haplotype tree lies within the geographical information, which may be lost if inferences are drawn stepwise, by conditioning on a single tree first, rather than on the joint tree-geography space.

In the presence of homoplasy, the haplotype tree is unknown and the tree space is infinite. In order to allow for computationally feasible inferences, we describe how a *finite* set of ‘realistic’ (in terms of a relaxed parsimony criterion) haplotype trees Ω may be obtained from the sequence data \mathcal{S} . Algorithm A, which may be found in appendix A, uses a fixed mutational step limit d_s , and assumes that any pair of disconnected sequences that are d SNPs apart will be a maximum of $d + d_s$ actual mutations apart. The set of ‘realistic’ trees is constructed by cumulatively adding intermediate sequences following the relaxed parsimony assumption defined by d_s . In general, larger values of d_s (up to a maximum value) yield more inclusive (and hence realistic) sets Ω ; the choice of d_s is simply chosen according to computational power. A more sophisticated approach would allow d_s to vary according to a prior distribution, but implementation involves calculating normalization constants over the vast space of trees. For a fixed d_s , algorithm A inputs the DNA sequences at hand, and outputs a sequence network, including loops. The true haplotype tree is then assumed to be one of the subtrees of this network and can be obtained through the breaking of loops.

In this paper, we assume a uniform model over the space of such trees. This means that, in the absence of geographical information, all possible trees have equal probability. It can be shown [13] that there is a one-to-one correspondence between trees and loop-breaking edges, allowing for sampling and inferences on the space of trees to be computationally efficient.

3. PHYLOGEOGRAPHIC CLUSTERING

The underlying assumption of our methods relies on a simplified island migration model with equal migration rates and no back-migration, resulting in a coalescent model with subdivided populations. By projecting the coalescent onto a haplotype tree (although this projection is not one-to-one) and defining the geographical clusters conditional on this haplotype tree, the computational complexity of the algorithms is significantly reduced. Specifically, we assume that we are given a sample of N DNA sequences corresponding to N_h haplotypes from a haplotype tree, along with the corresponding geographical location where each sequence was sampled. Combined with a haplotype tree model, this provides a basis for simultaneous inferences on the joint haplotype tree and population clustering space.

We define population clusters that are consistent with the geographic and genetic information available. Our clustering model corresponds to a simplified island

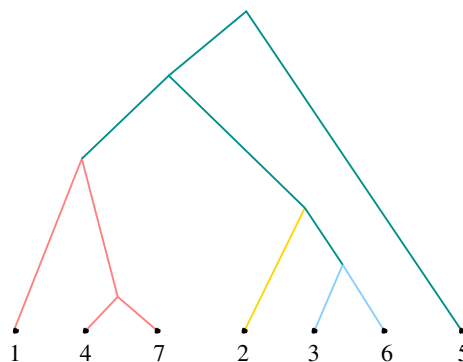


Figure 1. A coalescent tree with subdivided populations: green is the ancestral population, from which sequences subsequently migrated to found the pink, yellow and light blue populations.

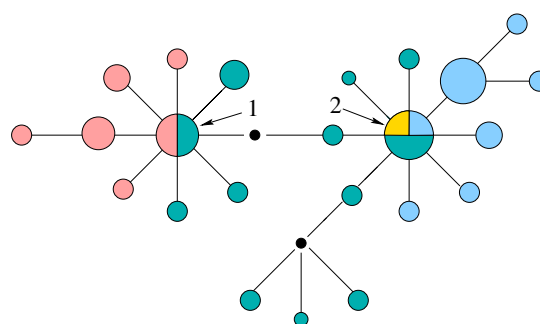


Figure 2. Example of a migration haplotype tree. The pink–green haplotype (1) is shared between the pink and green population clusters with half of its copies found in each, whereas the green–blue–yellow (2) is shared between the green, blue and yellow clusters with half of its copies found in the green cluster, and a quarter in each of the remaining two. In this case, the yellow cluster only contains copies of haplotype 2. Black dots represent unsampled (but known) haplotypes.

model with an unknown number of islands, such that several sampling locations and haplotypes may be grouped together into one homogeneous population; see Latter [20]. Although this migration model is less flexible than the model implemented in Sanmartín *et al.* [6], it allows for inference on the number and structure of the population clusters. The main assumption here is that each sequence (rather than each haplotype) is assumed to belong to a single geographical population cluster [21]. We thus aim to cluster individuals such that haplotypes may be shared across clusters owing to moving individuals following migration [18,19].

We develop a construction of phylogeographic clusters on haplotype trees that are consistent with migration island models, yielding shared haplotypes between populations. Using the migration model in De Iorio & Griffiths [21] consider a scenario where an ancestral population (depicted as green in figure 1) is the source for the colonization of another three populations, shown in yellow, pink and light blue. The migration model may be projected onto a haplotype tree. An example of (an extended version of) a haplotype tree consistent with the coalescent tree in figure 1 is shown in figure 2.

We introduce a general setting in which phylogeographic clusterings can be projected onto a haplotype tree. The clusters are seeded by K migrating haplotypes denoted m_1, \dots, m_K (not necessarily distinct), where K is fixed in this section, leading to $K + 1$ population clusters. Each of the migration events between two populations results in the migrating haplotype potentially being present in both populations (at some point in time). We denote the clusters that migrating haplotype m_k is shared between as the set $\mathcal{C}(m_k)$ (for example, pink and green for the case of haplotype 1 in figure 2), and use \mathbf{m} to denote the set of all migrating haplotypes. All sequences corresponding to a migrating haplotype m_k belong to one of the $|\mathcal{C}(m_k)|$ clusters.

Each migrating haplotype has a number of clades adjacent to it, which end either at a leaf node or at another migrating haplotype. All sequences contained in each of those clades are clustered together in one of the $|\mathcal{C}(m_k)|$ clusters. Intuitively, every single descendant will appear in exactly one of these populations unless another migration occurs. This implies that sequences (i.e. observations) corresponding to a haplotype that did not migrate are forced to belong to the same cluster, whereas sequences of a migrating haplotype may belong to different clusters. All such phylogeographic clusterings can be obtained using algorithm B, described in detail in appendix B.1, which describes a step-by-step method of constructing clusterings that are consistent with K migrating haplotypes based on a fixed haplotype tree of N_h haplotypes.

Once the complete clustering is determined, it is possible to separate all the observations into hard clusters. However, it is not possible directly to extract the historical information of the geographical movements. For example, in figure 2, we cannot distinguish whether the yellow cluster was formed before or after the light blue, and whether it was, for example, a migration from the pink or green cluster. It is only possible to make a (subjective) interpretation of the output, for example using the fact that smaller populations are more likely to be younger [22], or using external sources of information, for example about past glaciation of the area [23]. Devising a method that would directly infer historical events requires modelling complex phylogeographic phenomena and would greatly increase the complexity of the algorithms.

We denote the complete set of coordinates by \mathcal{Y} , such that y_{ij} refers to the coordinates (latitude–longitude) of the j th observation of haplotype i . The distribution of those coordinates is assumed to be Gaussian with mean μ_k and covariance Σ_k determined by each population cluster k . The location and shape of individual clusters is specified through independent Gaussian-Inverse Wishart priors, departing from the standard conjugate prior in order to decouple the dependence between the location and spread of individual clusters. The location coordinate data are standardized so that the mean is zero and the total (including both longitude and latitude) sample variance is one, keeping the North–South and East–West ratio fixed. We introduce the following priors for the clustering model, such that the phylogeographic clustering

model in full amounts to

$$\left. \begin{aligned} \mathcal{T} &\sim \mathcal{U}\{\Omega\}, \\ \mathcal{Y} | \mathbf{e}, \boldsymbol{\mu}, \boldsymbol{\Sigma} &\sim \mathcal{N}_p(\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i), \\ m_k &\sim \mathcal{U}\{1, \dots, n\} \text{ with replacement,} \\ \mathbf{c}, \mathbf{m} &\sim \text{Mult} \left\{ \prod_{k=1}^K \frac{\min(|m_k|, 1)}{N} \right. \\ &\quad \left. \times |\mathcal{C}(m_k)|^{-|m_k| - \text{deg}(m_k)} \right\}, \\ \boldsymbol{\mu}_k | \Sigma_k &\sim \mathcal{N}_p(0, V), \\ \Sigma_k &\sim \mathcal{IW}(\boldsymbol{\gamma}, \boldsymbol{\Psi}), \\ \text{and } \boldsymbol{\gamma} &\sim \mathcal{U}\{p + 1, \dots, g\}. \end{aligned} \right\} \quad (3.1)$$

Here p is the dimensionality of the data ($p = 2$ in the case of purely geographical, $p = 3$ with an additional covariate etc.), $|m_k|$ is the sample size of haplotype k and $\text{deg}(m_k)$ is the degree of haplotype k , i.e. the number of adjacent haplotypes. The parameter \mathcal{T} represents the haplotype tree (from the reduced set Ω), and $\boldsymbol{\gamma}$ is a hyperparameter which allows our model to borrow information about the spread of the population clusters from the data. We use the notation c_{ij} to represent the cluster of the j th data point corresponding to haplotype i , and \mathbf{c} to represent the set of all cluster memberships. The allocation parameter c_{ij} is forced to be the same for all j of haplotypes which are not shared, but is allowed to take different values for shared ones. The motivation for these priors is described in detail in appendix B.3. The distributions in model (3.1) give that

$$\boldsymbol{\mu}_k | \mathcal{Y}, \mathbf{e}, \Sigma_k \sim \mathcal{N}_p \left(\frac{\Sigma_k^{-1} n \bar{\mathbf{y}}}{V^{-1} + n \Sigma_k^{-1}}, \frac{1}{V^{-1} + n \Sigma_k^{-1}} \right) \quad (3.2)$$

and

$$\begin{aligned} \Sigma_k | \mathcal{Y}, \mathbf{e}, \boldsymbol{\gamma}, \boldsymbol{\mu} &\sim \mathcal{IW} \\ &\times \left(N + \boldsymbol{\gamma}, \boldsymbol{\Psi} + \sum_{j,l} \mathbb{I}_{\{c_{j,l}=k\}} (\mathbf{y}_{j,l} - \boldsymbol{\mu}_k)^\top (\mathbf{y}_{j,l} - \boldsymbol{\mu}_k) \right) \end{aligned} \quad (3.3)$$

The drawback of fixing the set Ω before the MCMC algorithm is that Ω may not include the true tree. Although it is generally true that evolution may follow the minimal path [24], this is not always the case, especially when data are deeply divergent or homoplasic. The parsimony assumption can alternatively be avoided by defining the clustering algorithms directly on coalescent trees, but that can be computationally intensive. In a coalescent tree setting, each clustering of a single haplotype tree requires particular permutations of coalescence events which allow for the order of migration events; in the haplotype tree framework, exploration of those permutations is decoupled from the inference about the tree, allowing for efficiently tuned proposal distributions.

In practice, we do not know the true number K of migrating haplotypes, and the number of clusters also needs to be inferred. We use a RJMCMC method similar to Richardson & Green [15], which allows moving between parameter spaces of different sizes. The model is augmented by adding a parameter K denoting

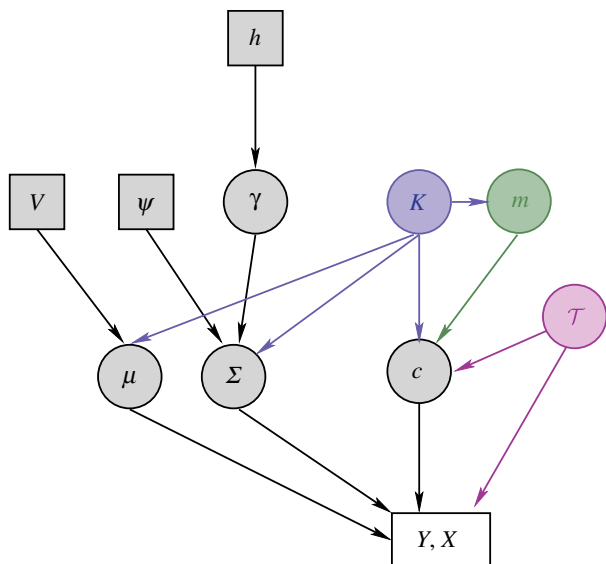


Figure 3. Here, we represent the parameters of our model as a directed acyclic graph (DAG). A DAG is interpreted as follows [25]: for any node v , conditioning on the value of its parent nodes (i.e. the nodes that have an arrow directed towards v), then no other nodes would be informative about v except its descendants. We adhere to the convention of representing fixed or observed quantities by squares, and circles for parameters that are estimated. The colours correspond to different types of analysis: the black represents the basic parameters of the Gaussian clustering model, the green for phylogeographic analysis, the blue for analysis with a variable number of clusters and the purple for the underlying haplotype tree.

the number of migration events, which is assumed to have a parsimonious Poisson prior $K \sim Po(K_0)$ for small K_0 . The hyperparameter γ in the degrees of freedom of the prior for the covariance matrices Σ is important because the number of clusters is heavily dependent upon the spread of each cluster. Thus, a prior for the covariance favouring small clusters will tend to result in a large K , and vice versa. The variable γ allows for the joint posterior of the number of clusters and their spread to be inferred. The hierarchical structure of the parameters is shown in figure 3.

The clustering of the observations is constrained by the structure of the haplotype tree; this implies that observations are not exchangeable given the haplotype tree, and the parameters of the clustering model cannot easily be sampled using, say, a Dirichlet process. Instead, in order to obtain samples from the posterior distribution, we construct a RJMCMC sampler with target distribution

$$\pi(\mathcal{T}, K, \mathbf{m}, \mathbf{c}, \gamma, \Sigma, \mu | \mathcal{Y}) \propto f(\mathcal{Y} | K, \mathbf{m}, \mathbf{c}, \Sigma, \mu) p(K) p(\mu) p(\gamma) p(\Sigma | \gamma) p(\mathbf{m}, \mathbf{c}).$$

For a detailed description of the Markov chain Monte Carlo updates, see appendix C.

4. SYNTHETIC DATA ANALYSIS

In order to test the validity of our methods, we generate a set of 10 replicate synthetic datasets and assess the

performance of our algorithm. For each dataset, we randomly draw an initial sequence of length $l = 500$, at an initial geographical location $y_{00} = (0,0)$ at the initial population cluster with centre $\mu_0 = (0,0)$. Subsequent sequences are generated according to a time-reversible Markov process, with possible events either splits or mutations at any of the available sites. We use a fixed rate of mutation equal to 1 and uniform across all possible mutations and sites, and a fixed rate of splitting w randomly drawn from $w \sim N(l,5)$ for each dataset. Each new sequence j of haplotype i then is assumed to belong to one of three possible locations: with probability 0.9 it stays in the geographical location of its ancestor a_{ij} such that $y_{ij} = y_{a_{ij}}$; otherwise, it either moves to a new location $y_{ij} = N(\mu_k, 0.1)$ but within the same geographical cluster $k = c_{a_{ij}}$ with probability 0.95, or it migrates to a new geographical cluster $\mu_{\text{new}} \sim N(\mu_k, 5)$ and creates a new location $y_{ij} \sim N(\mu_{\text{new}}, 0.1)$. The new sequence is forced to start a new location if the location of its ancestor contains 15 or more sequences. These tuning generative parameters were chosen in order for the synthetic datasets to match the real datasets at hand as much as possible. The iterative algorithm stops when it reaches 275 observed sequences (not including ones which are extinct in the process), corresponding to a variable number of haplotypes, locations and geographical clusters.

The RJMCMC sampler is then run on each dataset excluding extinct sequences (so that the true tree is unknown) for 10 seeds, each of 100 000 iterations. We set $V = 100\mathbb{I}, \Psi = 0.1 \times \mathbb{I}, \gamma \sim \mathcal{U}(3, \dots, 10)$. For each dataset, we compare the marginal maximum *a posteriori* cluster assignment to the true cluster assignment, and calculate the proportion of observations which are correctly assigned, yielding an average success rate over the 100 datasets of 82 per cent.

5. IMPLEMENTATION

We apply our algorithms to a mitochondrial DNA dataset of weevils in the Iberian peninsula. *Rhinusa vestita* is a seed parasite weevil feeding and reproducing on snapdragons. It is believed to have been present in Portugal, Spain, France and Italy. The complete nucleotide sequence for the mitochondrial COII gene (722 bp) was obtained for 275 *Rhinusa vestita* individuals. Previous studies investigating the association of weevils with three host plant species, combined with knowledge about the glaciation history of the Iberian peninsula [23], led to the biological prediction that the species originated from the Rhône valley to the east and west.

We implement our methods on the weevil dataset, taking the maximum parsimony level at $d_s = 3$, yielding 28 loops. Referring back to model 3.1 for $p = 2$, we set $V = 100\mathbb{I}, \Psi = 0.1 \times \mathbb{I}, \gamma \sim \mathcal{U}(3, \dots, 10)$ and $K_0 = 3$. The prior bound $g = 10$ was chosen so that large values of g result in a prior for the covariance which is very narrow compared with the typical distances between any two locations; it should be re-calibrated in cases where the posterior distribution of g shows significant support at the higher range of values. The posterior masses for the number of clusters from the RJMCMC sampler are shown in table 1. Although the

Table 1. The posterior masses for the number of migrations for the weevil dataset. The existence of four clusters is suggested, showing the highest posterior mass of 0.71.

no. of migrations	0	1	2	3	4	5	6
post. model prob.	0	0	0.28	0.71	0	0	0

data strongly inform the model about the posterior distribution of the phylogeographical clustering and the number of clusters, all haplotype trees consistent with each of those clusterings take equal posterior probability mass. The results of our method are shown below through one of maximum *a posteriori* estimate of the haplotype tree (figure 4) showing the unique MAP haplotype clustering, and a geographical contour plot of the clusters (figure 5). The results indicate that there is one large population cluster, shown in pink, which is geographically and genetically relatively homogeneous. Two clusters in the North-West are identified, which are geographically isolated, and are direct relatives of haplotype 2. Finally, the light blue cluster indicates an additional larger scale migration in the east-west direction. These findings agree with previous biological studies; additional inferences incorporating an evolutionary model can provide finer resolution into the timeline of the migration events.

In this case, the hyperparameter γ became important. Taking different values for Ψ yielded quite different clusterings for a fixed γ , especially in regard to the NW locations. Allowing γ to vary ensured robustness of the method, increasing the posterior mean of γ for larger values of Ψ .

6. DISCUSSION

We have presented a joint model for identifying a clustered geographical distribution consistent with an island migration model. By considering the corresponding subdivided population structure on the coalescent, we defined phylogeographic clusters consistent with a given haplotype tree. Introducing uncertainty about the haplotype tree, we provided a basis for joint inference of both the clusters and the tree in a flexible Bayesian framework. An interesting approach with similar features is presented by Sanmartín *et al.* [6], whereby the phylogeographic clusters are assumed fixed *a priori*, but a flexible model on the migration and evolutionary rates allows for a better representation of underlying processes. A combination of the two methods can simultaneously provide inferences on the structure of the phylogeographic clusters as well as evolutionary parameters.

Our methods can be improved and extended in several ways. Firstly, more sophisticated evolutionary models (such as a generalized time-reversible mutation model combined with a coalescent model) may be considered, taking into account individual population growth, population stationarity, variable mutation rates, panmixia, such as Huelsenbeck & Ronquist [26]. Some of those evolutionary model extensions have been implemented in Manolopoulou [13], yielding a non-equiprobable set of

trees; in practice, the posterior distribution is often dominated by the geographical information, so that the evolutionary dynamics only add to the computational cost. Recent advances in parallel computing with graphics processing units (GPUs) provide a powerful tool for inference and calculation on phylogenetic and coalescent trees [2]. Aside from providing more realistic evolutionary histories, this also allows for ancestral inference—although inference of individual ancestral haplotypes can yield particularly high variance estimates, inference of ancestral sampling locations can provide more stable posterior estimates [13].

Furthermore, more sophisticated migration models can be employed Sanmartín *et al.* [6], for example taking into account geographical distance to guide prior probabilities of individual migration events between phylogeographic clusters. To account for historical events such as population subdivision, the sample space of permissible clusterings can be extended to include those (see appendix B.1). Many of those extensions have been used in population genetics approaches such as Hey [4]. Alternative approaches also use a clinal rather than clustered geographical distribution such as described in Handley *et al.* [27].

As with most methods relying upon tree inference (whether phylogenetic, coalescent or haplotype), factors such as homoplasy, deep divergence and extinction can influence the reliability of the inferences. In such cases, the evolutionary history becomes a nuisance parameter without providing any additional information; other methods which do not draw inferences about the history may be more appropriate [3,28].

Finally, as is often the case with model-based approaches, the algorithms presented are heavily computational. Sophisticated inference and computer programming tools are necessary to ensure efficiency. The methods described are implemented in C built in an R-package that exports easily into Google Earth, available through <http://www.stat.duke.edu/~im30/software.html>.

APPENDIX A. UNCERTAINTY ABOUT THE HAPLOTYPE TREE

We describe the algorithm yielding a set of ‘feasible’ haplotype trees in the form of a network given a set of sequences.

ALGORITHM A

First, we pick a number of mutational steps d_s , which will be the number of mutations by which we relax the parsimony assumption for missing intermediate sequences. This means that we assume that if two sequences are k nucleotides apart, then they are at most $k + d_s$ mutational steps apart.

- (1) We connect any haplotypes that are one SNP apart, and count the number of disconnected groups of nodes. If the sequence data \mathcal{S} form a connected tree, then we assume that it is indeed the true haplotype tree \mathcal{T} so that $\Omega = \{\mathcal{T}\}$ and the algorithm terminates. For every pair of groups,

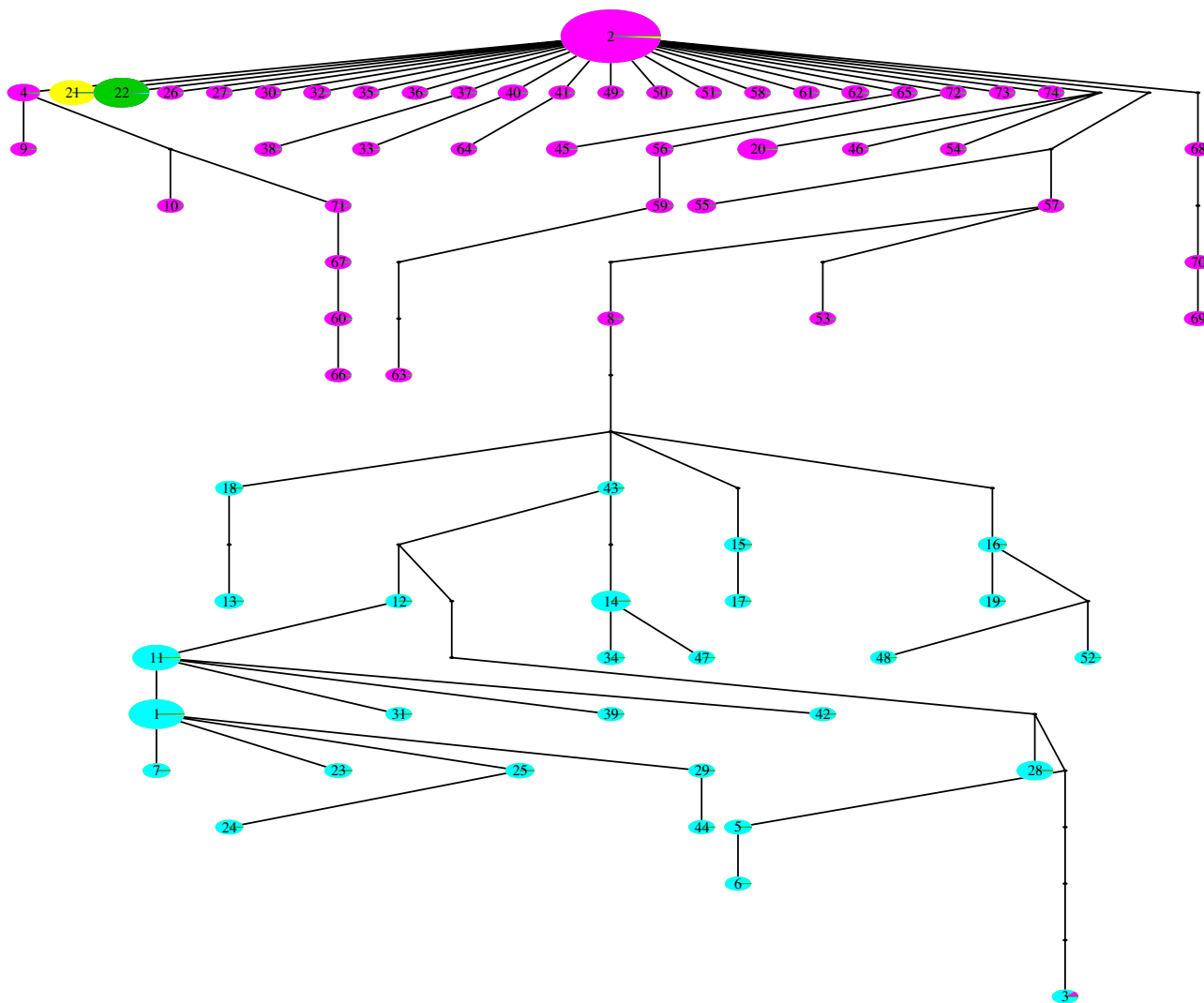


Figure 4. One of the non-unique MAP estimates of the haplotype tree using our approach, where colour corresponds to cluster and size to the number of individuals sampled with each sequence.

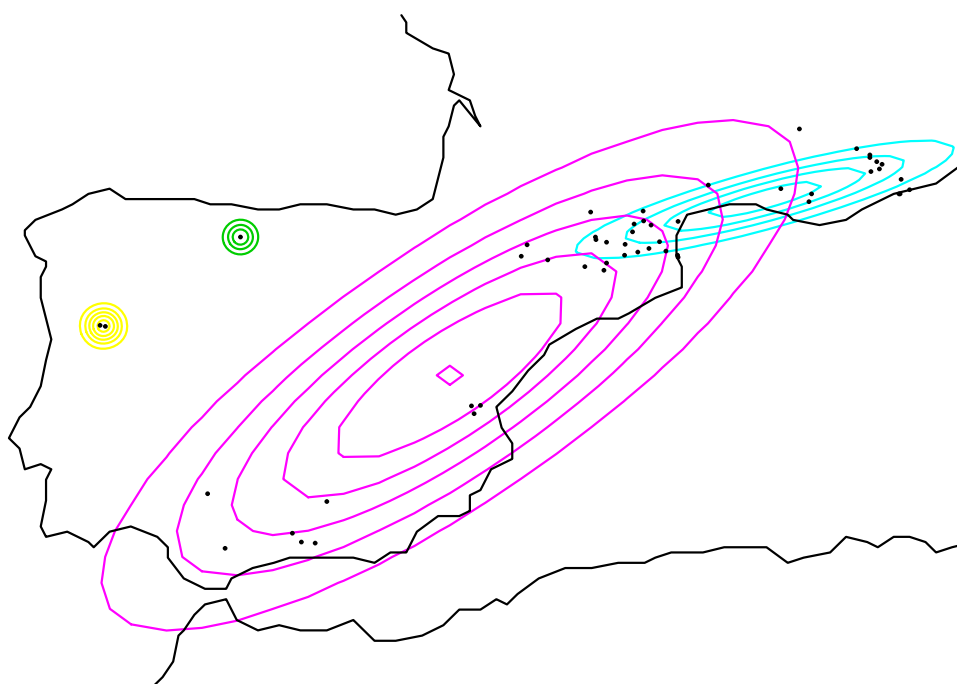


Figure 5. Corresponding bivariate normal contour plots evaluated at the posterior means for the weevil dataset. The black dots indicate sampling locations, and colours correspond to the clusters shown in figure 4.

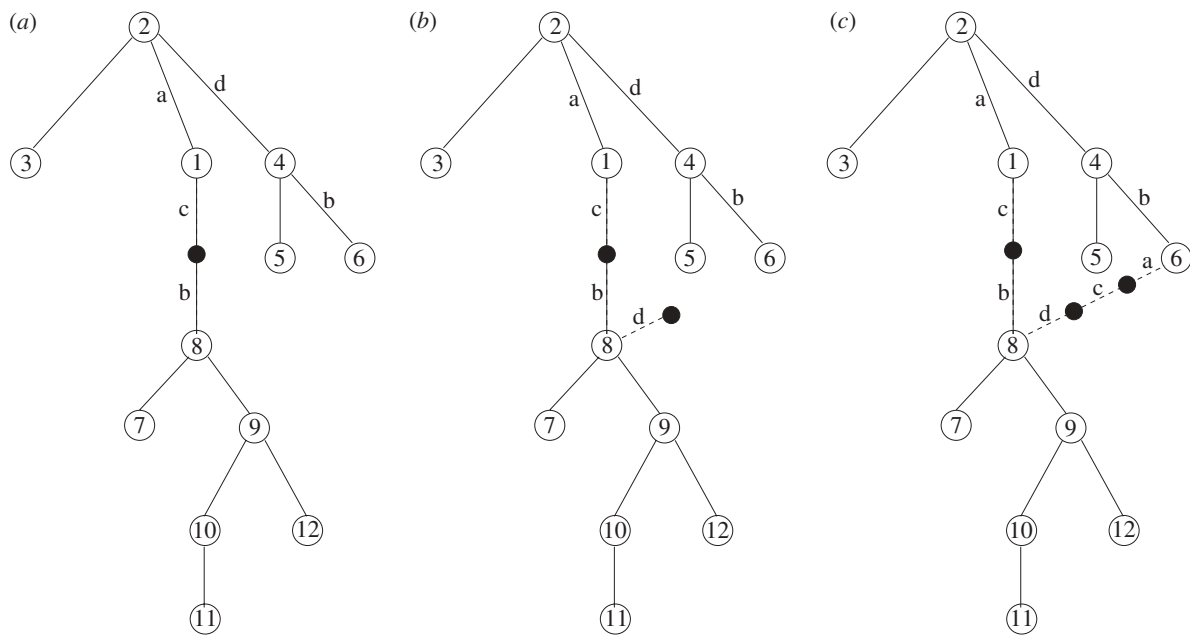


Figure 6. (a–c) The figure shows the three iterations for finding the two missing nodes. The letters on some of the edges represent the nucleotide position of each mutation. Here there is a back-mutation either at position a or at position b.

we find the closest distance between two nodes belonging to each group. We will refer to these pairs of nodes as the representatives between two groups (not necessarily unique). When no homoplasy is present, these are unique for each pair of groups. If the graph is connected (i.e. all sequences belong to the same group), then the algorithm terminates.

- (2) We then find d_{\min} , the minimum of these minimum distances.
- (3) We find all pairs of sequences (i, j) that belong to different groups and have distance (in terms of number of SNP mutations apart) $d(i, j) \leq d_{\min} + d_s$. If no such pair can be found, go to step 5 for the minimum pair of haplotypes.
- (4) For each pair (i, j) we then check if i has an adjacent node k which has $d(k, j) \leq d(i, j)$, and similarly for j . If either of these is true, then we repeat this step for the next pair of edges. Else we go to the next step.
- (5) We then find all the pairs of groups which have the reference node as one of their two representatives. We store the separating mutation positions between each one of these representatives and the reference node.
- (6) Then we find the separating mutation(s) which occurs most frequently between those pairs, and we pick one of them, which we call the ‘reference mutation’. This mutation has to be the one that occurred closest to the reference node, and so we create an extra node which is identical to the reference node except at the reference mutation position. When the reference mutation is not unique, without loss of generality we pick the first such nucleotide site. If any of these new nodes has already been created, we do not add the same sequence twice. We then go back to step 3 and repeat for the next pair of sequences.

This algorithm results in a haplotype *network*, implying that loops may appear. A key assumption of our approach is that the true haplotype tree is a subtree of the haplotype network obtained through algorithm A, which can be achieved by breaking the loops. Increasing d_s will generally result in disconnected groups of nodes being connected in more paths when homoplasy is present, implying that we can allow more and more possible haplotype trees. However, that does not imply that letting $d_s \rightarrow \infty$ will ensure that the network formed will include any possible mutational path. In fact, beyond a maximum value d_s^{\max} increasing d_s has no effect on the haplotype network.

A.1. Synthetic example

We use the haplotype tree shown in figure 6 and follow the steps described above in algorithm A in order to complete the missing nodes. Figure 6 shows a tree where nodes 1–12 are known haplotypes. We describe the algorithm for three cases: $d_s = 0$, $d_s = 1$ and $d_s > 1$.

- (1) Set $d_s = 0$. There are two disconnected groups of haplotypes: (1, 2, 3, 4, 5, 6), and (7, 8, 9, 10, 11, 12), with closest distance between nodes 1 and 8, which are two mutations apart at nucleotide positions b and c (step 1). Since there is only one pair of groups, immediately we obtain $d_{\min} = 2$ (step 2).

There are no other pairs of nodes from the two groups which are $d_{\min} + d_s = d_{\min}$ nucleotides apart (step 3), so we only need to connect the two nodes 1 and 8. Since there are only two groups available, they are the only ones involving the two missing mutations (step 4), so we insert the missing node 13 (referring to the original tree) and terminate (step 5). This single addition is shown in the left-hand panel of figure 6.

- (2) Set $d_s = 1$. There are two disconnected groups of haplotypes: (1, 2, 3, 4, 5, 6), and (7, 8, 9, 10, 11, 12), with closest distance between nodes 1 and 8 which are two mutations apart (step 1). Since there is only one pair of groups, immediately we obtain $d_{\min} = 2$ (step 2).

In this case, (1, 8) is the closest pair, but (2, 8) and (6, 8) are three nucleotides apart, which is indeed less than $\leq d_{\min} + d_s$ apart (step 3). Node 2 is adjacent to 1, which is closer to 8, so considering (2, 8) is implicit in the pair (1, 8), and hence redundant (step 4). On the other hand, no such adjacent nodes exist for the pair (6, 8), which has to be taken into account (step 4). For both pairs (1, 8) and (6, 8), there are only two groups involving the nucleotide changes (step 5), so both pairs are connected through their quickest route (step 5). In the case of (1, 8) this yields the same connection as before (figure 6a), but an extra branch is added on the right through two missing nodes, as shown figure 6b.

- (3) Set $d_s > 1$. In this case the exact network is obtained as in the case $d_s = 1$. This is because any extra pairs of sequences (i, j) which are obtained in step 3 actually have an adjacent node k which is closer to j , thus making the pair (i, j) redundant. The only pairs of sequences that reach step 5 are, as before, (1, 8) and (6, 8).

Lemma A.1. *When no homoplasy is present, algorithm A results in a unique haplotype tree (the true tree) up to rearrangement of strands of missing intermediate sequences for any value of d_s .*

Proof. In the absence of homoplasy, the following facts can be checked:

- the effective representatives of two groups are unique. This is true because in the absence of homoplasy, the mutational distance on the tree is always equal to the distance of the two sequences in terms of SNP mutations. If this were not the case, i.e. there exist two haplotypes that are closer in terms of their SNP distance than their tree distance, then at least one mutation would have had to be reversed, which contradicts the assumption of no homoplasy. This implies that there is only one pair of haplotypes which satisfies the condition in step 4 of the algorithm;
- each SNP mutation uniquely dichotomizes the sequences even in the absence of the tree. This means that it is not possible for two different pairs of groups which have the same minimum distance to involve the same mutation;
- if two pairs of groups with different minimum distances involve a common mutation, then the inferred mutations of both will coincide on the shorter branch.

Now assume that the inferred tree is not unique. This is possible in two ways: either two groups yield two effective pairs of representatives, or two different pairs of groups which involve a common mutation yield

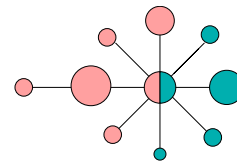


Figure 7. Example of a migrating haplotype shared between two populations, here green and pink. As before, nodes represent haplotypes, with the size of the circle representing the number of times that haplotype appears in the sample. The colour of each node shows the population cluster to which it belongs, with one haplotype being shared between the two clusters. Two haplotypes are connected by an edge if they are one mutation apart.

different intermediate sequences. None of these is possible, using the facts above, and hence the inferred tree is unique.

APPENDIX B. PHYLOGEOGRAPHIC CLUSTERING

B.1. The migration model

Consider the following example. Assume that haplotype i belongs to population A . If one of the individuals carrying haplotype i migrates from population cluster A to start a new population B , then haplotype i will be found in both clusters A and B . Assuming that no more migration (or other phylogeographic) events occurred, all of the descendants of i will either belong to cluster A (if their ancestral sequence belongs to cluster A) or cluster B (if their ancestral sequence belongs to cluster B). An example of the resulting haplotype tree is illustrated in figure 7, where as usual the colour indicates the cluster to which each sequence belongs.

We introduce a general setting in which phylogeographic clusterings can be projected onto a haplotype tree. The clusters are seeded by K migrating haplotypes denoted m_1, \dots, m_K (not necessarily distinct), where K is fixed in this section, leading to $K + 1$ population clusters. Each of the migration events between two populations results in the migrating haplotype being present in both populations (at some point in time). We denote the clusters that haplotype m_k is shared between as the set $\mathcal{C}(m_k)$. All sequences corresponding to a migrating haplotype m_k belong to one of the $|\mathcal{C}(m_k)|$ population clusters.

Based on the vector m_1, \dots, m_K , we describe how all the sequences are allocated to clusters given the set of migrating (and as a result shared) haplotypes \mathbf{m} . The main assumption of the clustering is that each sequence is allocated to precisely one cluster. All sequences corresponding to a migrating haplotype m_k belong to one of the $|\mathcal{C}(m_k)|$ clusters.

Each migrating haplotype has a number of clades starting from it, which end either at a leaf node or at another migrating haplotype. For each of those clades, all sequences within are clustered together in one of the $|\mathcal{C}(m_k)|$ clusters. Intuitively, a single mutation can occur in only one of the populations where it is present, so that every single descendant will appear in exactly one of these populations. This implies that sequences

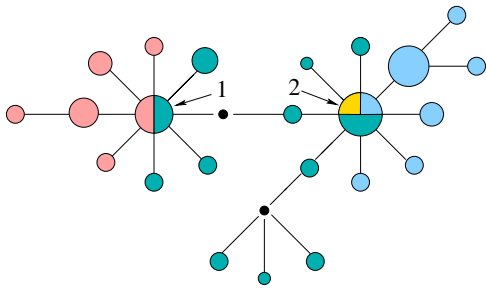


Figure 8. Example of a migration haplotype tree. The pink–green haplotype (1) is shared between the pink and green population clusters with half of its copies found in each, whereas the green–blue–yellow (2) is shared between the green, blue and yellow clusters with half of its copies found in the green cluster, and a quarter in each of the remaining two. In this case, the yellow cluster only contains copies of haplotype 2. Black dots represent unsampled (but known) haplotypes.

(i.e. observations) corresponding to a haplotype which did not migrate are forced to belong to the same cluster, whereas sequences of a migrating haplotype may belong to different clusters.

It is perhaps easier to think of clusterings seeded by the vector \mathbf{m} of migrating haplotypes in terms of migrations of the corresponding individuals. Taking the example in figure 8 and assuming the green cluster is ancestral, the migrating haplotypes are $\mathbf{m} = 1, 2, 2$ and they are shared between two and three clusters respectively, i.e. $|\mathcal{C}(m_1)| = 2$, $|\mathcal{C}(m_2)| = 3$. This corresponds to the three migration events of an individual migrating from the green cluster to a new population (pink), and individuals with haplotype 2 migrating to two new populations (yellow and light blue). It is thus clear that the number of times a specific haplotype occurs in \mathbf{m} is equal to the number of clusters it is shared between minus one.

Introducing K migrating haplotypes leads to the existence of $K + 1$ clusters. Each migrating haplotype represents a migration which introduces a new population cluster, thus K shared haplotypes result in $K + 1$ population clusters.

All such phylogeographic clusterings can be achieved by algorithm B, which describes a step-by-step method of constructing clusterings which are consistent with K migrating haplotypes based on a fixed haplotype tree of N_h haplotypes.

ALGORITHM B

- (1) Pick K of the N_h haplotypes with replacement, and denote them by m_1, \dots, m_K .
- (2) Pick one of the K migrating haplotypes m_k . The number of clusters that m_k is shared between is equal to the number of times it appears in the vector \mathbf{m} , plus 1. If the selected haplotype is shared between $|\mathcal{C}(m_k)|$ clusters, introduce clusters $1, \dots, |\mathcal{C}(m_k)|$ associated with that haplotype. Then iterate the following steps.
- (3a) Select one of the K haplotypes m_k that has at least one population cluster associated with it. If the clusters associated with it are fewer in number than the clusters it is shared by, introduce new clusters associated with this haplotype to complete the set.

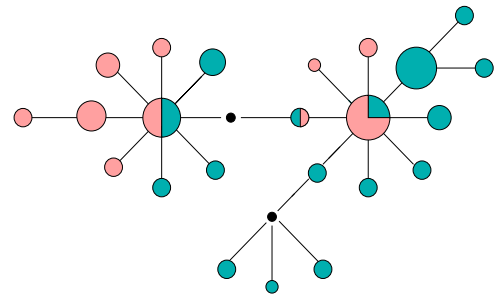


Figure 9. A subdivided population which is a result of past fragmentation. Initially one population was present, to which the three pink/green haplotypes belonged. The population was subsequently fragmented, so that the three haplotypes are found in both fragments. All their descendants after the split belong exclusively to one of the two populations.

- (3b) Allocate each of the data points of the chosen haplotype m_k to one of the associated clusters $\mathcal{C}(m_k)$.
- (3c) Allocate each of its adjacent nodes along with their branches (until either a leaf or another migrating haplotype is reached) to one of the associated clusters. If a migrating haplotype is reached, associate it with that cluster. Go back to step 3 until all haplotypes have been fully assigned to clusters.

Algorithm B is formed by following the properties of a phylogeographic clustering described in the current subsection, and as a result, any consistent clustering may be obtained.

B.2. Synthetic example

Using algorithm B we demonstrate how the clustering of figure 8 may be obtained from the haplotype tree in one of several ways.

- Start with step 1. The three migrating haplotypes are picked to be 1, 2, 2.
- Continue with step 2. Pick haplotype 1, which is shared between two clusters, and assume that the two clusters are 1 and 2 (in this case pink and green).
- Move on to step 3. Haplotype 1 is the only one which has any clusters associated with it, so pick haplotype 1.
- In step 4, allocate each of the data points of 1 to the pink or the green cluster one by one. In this case half of them are allocated to the pink and half to the green cluster, as indicated by the proportions of pink and green on the haplotype node.
- In step 5, allocate each of its adjacent branches to a cluster. All apart from one branch reach a leaf before reaching the other migrating haplotype. Those leaf branches are allocated to the pink or green clusters (in the figure the tree has been re-arranged so that all the pink ones lie on the left and all the green on the right; this need not be the case).

We allocate the branch connecting haplotype 1 and haplotype 2 to the green cluster, and thus assign one of the three clusters in which haplotype 2 is found to be the green one.

- Return to step 3 and select haplotype 2, which now has the green cluster assigned to it. We assign yellow and light blue for the remaining two.

- Continue with step 4 and assign each of the sequences of haplotype 2 into one of the three available clusters. In this case half of the sequences are allocated to the green cluster, a quarter to the light blue and a quarter to the yellow.
- Continue with step 5 and assign each of the adjacent branches which have not yet been allocated to a cluster into green, light blue or yellow. Note here that none of the adjacent branches is allocated to the yellow cluster.

The same clustering may be obtained for a number of different choices for the steps of algorithm B (e.g. if we select haplotype 2 in step 2).

We remark that the phylogeographic clustering does not explicitly account for past fragmentation events. Notice that if a population undergoes fragmentation, a number of haplotypes which originally belonged to the same population will subsequently belong to the two fragment populations. As a result, all their descendants will belong to only one of the two. The resulting haplotype tree may look like figure 9. The haplotype sharing construction described here does not allow for such a clustering, but would instead only identify the three migrating haplotypes as being shared between four clusters. The clustering construction could be extended to allow explicitly for such clusterings.

B.3. The clustering model

We use algorithm B to motivate a prior distribution for clustering constructions. Here we are assuming that *a priori*, any sequence is equally likely to correspond to a migrating haplotype. Referring back to the simplified migration setting described on page 8, this is equivalent to any individual being equally likely to migrate. This means that the probability of a haplotype being shared is proportional to the number of times it appears in the sample, yielding

$$p(\mathbf{m}') = \prod_{k=1}^K \frac{\min(|m_k|, 1)}{n},$$

where $|m_k|$ is the sample size of haplotype m_k . Note that here we correct $|m_k|$ by $\min(|m_k|, 1)$ to account for the fact that some haplotypes are extinct or unsampled, but may still have a non-zero probability of having migrated.

We use the notation c_{ij} to represent the cluster of the j th data point corresponding to haplotype i . In this case the allocation parameter c_{ij} is forced to be the same for all j for haplotypes which are not shared, but is allowed to take different values for shared ones. Assuming that the clusters chosen for each of the data points and branches of haplotype m_k in steps 3b and 3c of algorithm B are selected randomly from the $|\mathcal{C}(m_k)|$ clusters, the priors for the clustering \mathbf{c} can be written as

$$\begin{aligned} p(\mathbf{c}) &= p(\mathbf{c}, \mathbf{m}) = p(\mathbf{m})p(\mathbf{c}|\mathbf{m}) \\ &= \prod_{k=1}^K \frac{\min(|m_k|, 1)}{N} |\mathcal{C}(m_k)|^{-|m_k| - \deg(m_k)}, \end{aligned} \tag{B 1}$$

where $\deg(m_i)$ is the degree of haplotype m_k , i.e. the number of adjacent haplotypes. In other words, each of

the migrations (m_k for each migrating haplotype) is into one of the $|\mathcal{C}(m_k)|$ available clusters, resulting in $|\mathcal{C}(m_k)|^{-|m_k| - \deg(m_k)}$. This implies that, for large numbers of migrations and/or haplotypes, the number of combinations increases rapidly, requiring significantly larger computation times.

APPENDIX C. MARKOV CHAIN MONTE CARLO SAMPLER

We construct an MCMC sampler with target distribution

$$\begin{aligned} \pi(\mathcal{T}, K, \mathbf{m}, \mathbf{c}, \gamma, \Sigma, \mu | \mathcal{Y}) &\propto f(\mathcal{Y} | K, \mathbf{m}, \mathbf{c}, \Sigma, \mu) \\ & p(K)p(\mu)p(\gamma)p(\Sigma|\gamma)p(\mathbf{m}, \mathbf{c}). \end{aligned} \tag{C 1}$$

In order to achieve a computationally feasible sampler, devising efficient proposal kernels to move around the space of clusterings is key.

The nature of the phylogeographic clustering setting we are assuming implies a vast allocation parameter space. We develop a proposal kernel exploring the space of possible clusterings efficiently. Algorithm B describes a method by which phylogeographic clusterings can be achieved. In an MCMC setting, it can be modified so that the choices are made efficiently and allow mixing of the chains. To this end, we discuss some technical properties of the clustering algorithm.

Note that it is not easily possible to construct a local version of algorithm B; unless the algorithm is completed, the clustering cannot be updated, because the resulting clustering may be physically nonsensical and contradict the migrating haplotype structure. Hence, for each MCMC iteration, all clusters are initially empty, data points are gradually added using a variant of algorithm B until complete, and only then can the proposed move be accepted or rejected.

Here clusters are constrained by the phylogeographic clustering structure on the haplotype tree, which dictates that allocating an adjacent node to one of the clusters implies allocating a whole branch of the tree to that cluster. Algorithm C described below is a variant of algorithm B, using specific proposal distributions for each step which take into account the clustering of the previous iteration by allowing the proposal to extract information about the clusters using the allocation values which have been proposed so far within the same MCMC iteration. Population clusters are iteratively ‘filled’ with observations starting with initial local estimates and allowing those estimates to be updated depending on data point additions.

ALGORITHM C

During burn-in, for each iteration initially we set all clusters to be empty, with sample mean and covariances $\bar{\mu}, \bar{\Sigma}$ equal to their prior estimates. After burn-in, initially set all clusters involved with migrating haplotypes which have not been changed since the previous iteration to have mean, variance and sample size as in the previous iteration $(\bar{\mu}_i, \bar{\Sigma}_i, \bar{n}_i) = (\mu_i^{(t-1)}, \Sigma_i^{(t-1)}, n_i^{(t-1)})$ respectively.

Then carry out the following steps:

- (1) Select one of the migrating haplotypes of $\mathbf{m}^{(t-1)}$ uniformly at random from the previous iteration and change it to m'_k , proposing the new haplotype randomly.
- (2) For each of the migrating haplotypes m'_k shared by $|\mathcal{C}(m'_k)|$ clusters, if it was shared by $|\mathcal{C}(m_k)|^{(t-1)} \geq |\mathcal{C}(m'_k)|$ clusters at the previous iteration too, assign this migrating haplotype to be shared between the first $|\mathcal{C}(m_k)|^{(t-1)} - 1$ clusters of the set $\mathcal{C}(m_k)^{(t-1)}$, leaving the last cluster of $\mathcal{C}(m'_k)$ null.
- (3) Select at random one of the migrating haplotypes m'_k which has not been allocated to clusters; if none such exist, the algorithm has completed. If it was previously a migrating haplotype with at least $|\mathcal{C}(m'_k)|$ available clusters, then the last cluster of $\mathcal{C}(m_k)$ is set to same one as the previous iteration. Otherwise the next available cluster from the list of all clusters is chosen.
- (4) Select at random one of the observations j of the migrating haplotype m_k which has not been assigned to a cluster, and assign it to one of the available clusters $m \in \mathcal{C}(m_k)$ with probability

$$\propto p(c_{mkj} = m | \mathcal{Y}, \bar{\Sigma}, \bar{\mu}, \mathbf{m})$$

$$\propto |\bar{\Sigma}_m|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y}_{mkj} - \bar{\mu}_m)^T \bar{\Sigma}_m^{-1} (\mathbf{y}_{mkj} - \bar{\mu}_m)\right).$$

Update the sample means and covariances $\bar{\mu}_{c_{mkj}}, \bar{\Sigma}_{c_{mkj}}$. If all data points of m_k have been assigned to a cluster, move on to the next step, else repeat this step.

- (5) Select one of the adjacent nodes l of m_k which has not been assigned to a cluster yet. Each adjacent node defines a branch, which starts at the adjacent node and ends either at a leaf node, or at another migrating haplotype. Assign all data points j of all the haplotypes i along the branch to one of the clusters $m \in \mathcal{C}(m_k)$, with probability

$$\propto p(\cup_{i,j \in \text{branch}} c_{ij} = m | \mathcal{Y}, \bar{\Sigma}, \bar{\mu}, \mathbf{m})$$

$$\propto \prod_{i,j} |\bar{\Sigma}_m|^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{y}_{ij} - \bar{\mu}_m)^T \bar{\Sigma}_m^{-1} (\mathbf{y}_{ij} - \bar{\mu}_m)\right),$$

where the product is taken over all data points of all haplotypes along the branch. If the branch ends at a migrating haplotype, then assign one of its associated clusters to be m . If all adjacent branches have been allocated to clusters, go back to step 3. Else repeat this step.

Using algorithm C, we adapt the MCMC algorithm described in previous sections for the phylogeographic data. The chain is initialized by generating $\mathbf{m}^{(0)}, \mathbf{c}^{(0)}, \boldsymbol{\gamma}^{(0)}, \boldsymbol{\mu}^{(0)}, \boldsymbol{\Sigma}^{(0)}$ from the prior distributions. Subsequently iterate the following steps:

- (C1a) Split sequences into clusters using algorithm C.
- (C1b) Propose a new value $\boldsymbol{\gamma}'$ from its prior $\mathcal{U}\{p+1, g\}$.
- (C1c) Propose new covariance matrices $\boldsymbol{\Sigma}'_k$ from the conjugate approximation of $\boldsymbol{\Sigma}_k | \mathcal{Y}, \mathbf{m}', \mathbf{c}', \boldsymbol{\gamma}'$

given by

$$q(\boldsymbol{\Sigma}_k | \mathcal{Y}, \mathbf{e}, \boldsymbol{\gamma}) \sim \mathcal{IW}(n_k + \boldsymbol{\gamma}, \boldsymbol{\Psi} + \sum_{j,l} \mathbb{I}_{\{c_{jl}=k\}} \mathbf{y}_{jl} \mathbf{y}_{jl}^T - n_k \bar{\mathbf{y}}_k \bar{\mathbf{y}}_k^T). \quad (\text{C1})$$

- (C1d) Propose $\boldsymbol{\mu}'_k$ from $\boldsymbol{\mu}_k | \mathcal{Y}, \boldsymbol{\Sigma}'_k, \mathbf{m}', \mathbf{c}'$ given in equation (3.2).
- (C1e) Calculate

$$A_C = \frac{f(\mathcal{Y} | \mathbf{m}', \mathbf{c}', \boldsymbol{\mu}', \boldsymbol{\Sigma}') p(\mathbf{m}', \mathbf{c}') p(\boldsymbol{\Sigma}') p(\boldsymbol{\mu}')}{f(\mathcal{Y} | \mathbf{m}, \mathbf{c}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) p(\mathbf{m}, \mathbf{c}) p(\boldsymbol{\Sigma}) p(\boldsymbol{\mu})}$$

$$\times \frac{q(\mathbf{m}', \mathbf{c}' \rightarrow \mathbf{m}, \mathbf{c}) \pi(\boldsymbol{\mu} | \mathbf{m}, \mathbf{c}, \boldsymbol{\Sigma}) q(\boldsymbol{\Sigma} | \mathcal{Y}, \mathbf{m}, \mathbf{c})}{q(\mathbf{m}, \mathbf{c} \rightarrow \mathbf{m}', \mathbf{c}') \pi(\boldsymbol{\mu}' | \mathbf{m}', \mathbf{c}', \boldsymbol{\Sigma}') q(\boldsymbol{\Sigma}' | \mathcal{Y}, \mathbf{m}', \mathbf{c}')}.$$

Accept the move with probability $\min(1, A_C)$ and set

$$(\mathbf{m}^{(t+1)}, \mathbf{c}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\mu}'', \boldsymbol{\Sigma}'')$$

$$= (\mathbf{m}', \mathbf{c}', \boldsymbol{\gamma}', \boldsymbol{\mu}', \boldsymbol{\Sigma}'),$$

otherwise set $(\mathbf{m}^{(t+1)}, \mathbf{c}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\mu}'', \boldsymbol{\Sigma}'') = (\mathbf{m}^{(t)}, \mathbf{c}^{(t)}, \boldsymbol{\gamma}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})$

- (C2) Generate $\boldsymbol{\Sigma}^{(t+1)}$ directly from the posterior conditional

$$\boldsymbol{\Sigma} | \mathcal{Y}, \mathbf{m}^{(t+1)}, \mathbf{c}^{(t+1)}, \boldsymbol{\gamma}^{(t+1)}, \boldsymbol{\mu}''$$

given in equation (3.3).

- (C3) Generate $\boldsymbol{\mu}^{(t+1)}$ directly from the posterior conditional distribution

$$\boldsymbol{\mu} | \mathcal{Y}, \mathbf{m}^{(t+1)}, \mathbf{c}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)}$$

of equation (3.2). Go back to step C1.

- (C4a) Propose to add or subtract a migrating haplotype m_k with probabilities p_{split} and p_{merge} .

- (a) For a merging move, select two of the clusters $\mathcal{C}(m_k)$ between which m_k is shared, say k_1 and k_2 , and merge them into one cluster k' . The probability of this move becomes

$$q(\mathbf{m}, \mathbf{c} \rightarrow \mathbf{m}', \mathbf{c}') = \frac{1}{K^{(t)} \times \binom{|\mathcal{C}(m_k)|}{2}}. \quad (\text{C2})$$

- (b) For a splitting move, add one of the N_h haplotypes to the vector \mathbf{m} . All of the data points and adjacent haplotypes of the added node then have to be inserted to one of the available clusters. We start with \mathbf{m}' and reallocate all the data points of all the haplotypes to clusters according to Algorithm C. The probability of this move is equal to

$$q(\mathbf{m}, \mathbf{c} \rightarrow \mathbf{m}', \mathbf{c}') = \frac{1}{N_h} q(\mathbf{c}' | \mathbf{m}'), \quad (\text{C3})$$

where $q(\mathbf{c}' | \mathbf{m}')$ is calculated iteratively through algorithm C.

(C4b) We propose values for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the new clusters formed.

(a) If we decide to merge two clusters k_1 and k_2 into k' , then we propose $\boldsymbol{\Sigma}'_{k'} | \mathcal{Y}, \mathbf{e}', \boldsymbol{\gamma}$ (see equation (3.2)), and $\boldsymbol{\mu}'_{k'} | Y, \mathbf{e}, \boldsymbol{\Sigma}'_{k'}$ (see equation (2.2)). The remaining covariances of clusters which are not affected by the move are left unchanged.

(b) Similarly, if we decide to split one of the existing $K^{(t)} + 1$ clusters, then we propose $\boldsymbol{\Sigma}'_{k_1}, \boldsymbol{\Sigma}'_{k_2}, \boldsymbol{\mu}'_{k_1}, \boldsymbol{\mu}'_{k_2}$ from the distributions given in equations (C1), (3.2). The remaining covariances of clusters which are not affected by the move are left unchanged.

(C4c) The acceptance probability of a merging move becomes $\alpha = \min(1, A_E)$ where

$$A_E = \frac{f(\mathcal{Y} | \mathbf{m}', \mathbf{c}', \boldsymbol{\mu}', \boldsymbol{\Sigma}')}{f(\mathcal{Y} | \mathbf{m}, \mathbf{c}, \boldsymbol{\mu}, \boldsymbol{\Sigma})} \frac{p(\mathbf{m}', \mathbf{c}')}{p(\mathbf{m}, \mathbf{c})} \frac{p(\boldsymbol{\mu}'_{k'})}{p(\boldsymbol{\mu}'_{k_1}) p(\boldsymbol{\mu}'_{k_2})} \\ \times \frac{p(\boldsymbol{\Sigma}'_{k'})}{p(\boldsymbol{\Sigma}'_{k_1}) p(\boldsymbol{\Sigma}'_{k_2})} \\ \times \frac{q(\mathbf{m}', \mathbf{c}' \rightarrow \mathbf{m}, \mathbf{c})}{q(\mathbf{m}, \mathbf{c} \rightarrow \mathbf{m}', \mathbf{c}')} \frac{q(\boldsymbol{\mu}_{k_1}) q(\boldsymbol{\mu}_{k_2})}{q(\boldsymbol{\mu}'_{k'})} \frac{q(\boldsymbol{\Sigma}_{k_1}) q(\boldsymbol{\Sigma}_{k_2})}{q(\boldsymbol{\Sigma}'_{k'})} \\ \times \frac{p_{\text{split}}}{p_{\text{merge}}} |J|,$$

using equations (C1), (3.2), (C2) and (C3). As before, $|J| = 1$.

Similarly, the acceptance probability of a splitting move becomes $\alpha = \min(1, A_E^{-1})$. We decide to accept or reject the proposed move, with some terms replaced appropriately.

(C4d) If we accept, then we set $(\mathbf{K}^{(t+1)}, \mathbf{m}^{(t+1)}, \mathbf{c}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)}) = (\mathbf{K}', \mathbf{m}', \mathbf{c}', \boldsymbol{\mu}', \boldsymbol{\Sigma}')$, otherwise $(\mathbf{K}^{(t+1)}, \mathbf{m}^{(t+1)}, \mathbf{c}^{(t+1)}, \boldsymbol{\mu}^{(t+1)}, \boldsymbol{\Sigma}^{(t+1)}) = (\mathbf{K}^{(t)}, \mathbf{m}^{(t)}, \mathbf{c}^{(t)}, \boldsymbol{\mu}^{(t)}, \boldsymbol{\Sigma}^{(t)})$.

Cycling through steps C1–C4 produces an irreducible chain with stationary distribution (C 1).

Lemma C.1. *Algorithm C preserves irreducibility and aperiodicity of the chain.*

Proof. Clearly, it is always possible to change one of the migrating haplotypes to be haplotype 1, without loss of generality. Similarly, we may repeat the same, until haplotype 1 is the only migrating haplotype. Hence, we can get to this clustering from any other clustering, so the chain is irreducible. Aperiodicity is guaranteed because there is always a positive probability of staying in the same state during steps C1–C3.

Lemma C.2. *Randomizing the order in which data points and branches are clustered in steps 3 and 4 of algorithm C described above preserves time-reversibility of the chain.*

Proof. Notice first that the move $\mathbf{c}^{(t-1)} \rightarrow \mathbf{c}^{(t)}$ may be achieved in a number of different combinations of steps in the algorithm, depending on the order in which we choose to propose the migrating haplotypes and their data points; remember that the move can only be

accepted or rejected once they have all been proposed. Randomizing the order in which the migrating haplotypes are proposed is equivalent to having a pool of proposals q_i and randomly selecting one [29,30].

In the standard MCMC setting, the ratio of the proposal distributions would then be equal to:

$$\frac{q(\mathbf{c}^{(t)} \rightarrow \mathbf{c}^{(t-1)})}{q(\mathbf{c}^{(t-1)} \rightarrow \mathbf{c}^{(t)})} = \frac{\sum_i q_i(\mathbf{c}^{(t)} \rightarrow \mathbf{c}^{(t-1)})}{\sum_i q_i(\mathbf{c}^{(t-1)} \rightarrow \mathbf{c}^{(t)})},$$

where the sum is taken over all the possible step combinations which may lead to the same clustering.

However, in this setting we use the order of the update as an extra parameter, say \mathbf{z}_c , and assume that all step combinations have equal probability *a priori*. At each iteration we propose a step combination and then update the clustering using the proposal distribution

$$q(\mathbf{c}^{(t-1)} \rightarrow \mathbf{c}^{(t)}) = \sum_i \mathbb{I}_{i=\mathbf{z}_c} q_i(\mathbf{c}^{(t-1)} \rightarrow \mathbf{c}^{(t)}).$$

Clearly, q is a distribution, since all but one term will be zero, and q_i is a distribution. This means that the overall proposal ratio becomes simply

$$\frac{q(\mathbf{z}_c^{(t)} \rightarrow \mathbf{z}_c^{(t-1)})}{q(\mathbf{z}_c^{(t-1)} \rightarrow \mathbf{z}_c^{(t)})} \frac{q(\mathbf{c}^{(t)} \rightarrow \mathbf{c}^{(t-1)})}{q(\mathbf{c}^{(t-1)} \rightarrow \mathbf{c}^{(t)})} \\ = \frac{q(\mathbf{z}_c^{(t)} \rightarrow \mathbf{z}_c^{(t-1)})}{q(\mathbf{z}_c^{(t-1)} \rightarrow \mathbf{z}_c^{(t)})} \frac{q_{\mathbf{z}^{(t-1)}}(\mathbf{c}^{(t)} \rightarrow \mathbf{c}^{(t-1)})}{q_{\mathbf{z}^{(t)}}(\mathbf{c}^{(t-1)} \rightarrow \mathbf{c}^{(t)})},$$

and this can be treated as a standard time-reversible MCMC sampler.

REFERENCES

- 1 Templeton, A. 2008 Nested clade analysis: an extensively validated method for strong phylogeographic inference. *Mol. Ecol.* **17**, 1877–1880. (doi:10.1111/j.1365-294X.2008.03731.x)
- 2 Lemey, P., Rambaut, A., Drummond, A. & Suchard, M. 2009 Bayesian phylogeography finds its roots. *PLoS Comput. Biol.* **5**, e1000520. (doi:10.1371/journal.pcbi.1000520)
- 3 Minin, V., Bloomquist, E. & Suchard, M. 2008 Smooth skyride through a rough skyline: Bayesian coalescent-based inference of population dynamics. *Mol. Biol. Evol.* **25**, 1459–1471. (doi:10.1093/molbev/msn090)
- 4 Hey, J. 2010 Isolation with migration models for more than two populations. *Mol. Biol. Evol.* **27**, 905–920. (doi:10.1093/molbev/msp296)
- 5 Beerli, P. & Palczewski, M. 2010 Unified framework to evaluate panmixia and migration direction among multiple sampling locations. *Genetics* **185**, 313–326. (doi:10.1534/genetics.109.112532)
- 6 Sanmartín, I., Anderson, C. L., Alarcon, M., Ronquist, F. & Aldasoro, J. J. 2010 Bayesian island biogeography in a continental setting: the rand flora case. *Biol. Lett.* **6**, 703–707. (doi:10.1098/rsbl.2010.0095)
- 7 Beaumont, M. & Panchal, M. 2008 On the validity of nested clade phylogeographical analysis. *Mol. Ecol.* **17**, 2563–2565. (doi:10.1111/j.1365-294X.2008.03786.x)
- 8 Beaumont, M. *et al.* 2010 In defence of model-based inference in phylogeography. *Mol. Ecol.* **19**, 436–446. (doi:10.1111/j.1365-294X.2009.04515.x)

- 9 Knowles, L. & Rausher, M. 2008 Why does a method that fails continue to be used? *Evolution* **62**, 2713–2717. (doi:10.1111/j.1558-5646.2008.00481.x)
- 10 Templeton, A. 2009 Why does a method that fails continue to be used: the answer. *Evolution* **63**, 807–812. (doi:10.1111/j.1558-5646.2008.00600.x)
- 11 Templeton, A. 2010 Coherent and incoherent inference in phylogeography and human evolution. *Proc. Natl Acad. Sci. USA* **107**, 6376–6381. (doi:10.1073/pnas.0910647107)
- 12 Bloomquist, E., Lemey, P. & Suchard, M. 2010 Three roads diverged? Routes to phylogeographic inference. *Trends Ecol. Evol.* **25**, 626–632. (doi:10.1016/j.tree.2010.08.010)
- 13 Manolopoulou, I. 2009 A Bayesian approach to nested clade analysis. PhD thesis, University of Cambridge, Cambridge, UK.
- 14 Balloux, F., Handley, L., Jombart, T., Liu, H. & Manica, A. 2009 Climate shaped the worldwide distribution of human mitochondrial DNA sequence variation. *Proc. R. Soc. B* **276**, 3447–3455. (doi:10.1098/rspb.2009.0752)
- 15 Richardson, S. & Green, P. 1997 On Bayesian analysis of mixtures with an unknown number of components. *J. R. Statist. Soc. Ser. B (Methodol.)* **59**, 731–792. (doi:10.1111/1467-9868.00095)
- 16 Swofford, D. L., Olsen, G. J., Waddell, P. J. & Hillis, D. M. 1996 Phylogenetic inference. In *Molecular systematics* (eds D. M. Hillis, C. Moritz & B. K. Mable), pp. 407–514, 2nd edn. Sunderland, MA: Sinauer Associates.
- 17 Templeton, A. & Sing, C. 1993 A cladistic analysis of phenotypic associations with haplotypes inferred from restriction endonuclease mapping. IV. Nested analyses with cladogram uncertainty and recombination. *Genetics* **134**, 659–669.
- 18 Templeton, A. 1998 Nested clade analysis of phylogeographic data: testing hypotheses about gene flow and population history. *Mol. Ecol.* **7**, 381–397. (doi:10.1046/j.1365-294x.1998.00308.x)
- 19 Avise, J. 2000 *Phylogeography: the history and formation of species*. Harvard, MA: Harvard University Press.
- 20 Latter, B. 1973 The island model of population differentiation: a general solution. *Genetics* **73**, 147–157.
- 21 De Iorio, M. & Griffiths, R. 2004 Importance sampling on coalescent histories. II. Subdivided population models. *Adv. Appl. Probab.* **36**, 434–454. (doi:10.1239/aap/1086957580)
- 22 Emerson, B. & Hewitt, G. 2005 Phylogeography. *Curr. Biol.* **15**, 367–371. (doi:10.1016/j.cub.2005.05.016)
- 23 Hewitt, G. 2000 The genetic legacy of the quaternary ice ages. *Nature* **405**, 907–913. (doi:10.1038/35016000)
- 24 Sankoff, D. 1975 Minimal mutation trees of sequences. *SIAM J. Appl. Math.* **28**, 35–42. (doi:10.1137/0128004)
- 25 Gilks, W., Richardson, S. & Spiegelhalter, D. 1995 *Markov chain Monte Carlo in practice*. London, UK: Chapman & Hall.
- 26 Huelsenbeck, J. & Ronquist, F. 2001 MrBayes: Bayesian inference on phylogenetic trees. *Bioinformatics* **17**, 754–755. (doi:10.1093/bioinformatics/17.8.754)
- 27 Handley, L., Manica, A., Goudet, J. & Balloux, F. 2007 Going the distance: human population genetics in a clinal world. *Trends Genet.* **23**, 432–439. (doi:10.1016/j.tig.2007.07.002)
- 28 Drummond, A., Rambaut, A., Shapiro, B. & Pybus, O. 2005 Bayesian coalescent inference of past population dynamics from molecular sequences. *Mol. Biol. Evol.* **22**, 1185–1192. (doi:10.1093/molbev/msi103)
- 29 Geyer, C. 1992 Practical Markov chain Monte Carlo. *Statist. Sci.* **7**, 473–483. (doi:10.1214/ss/1177011137)
- 30 Geyer, C. 1991 Markov chain Monte Carlo maximum likelihood. In *Computing science and statistics: proceedings of 23rd Symposium on the Interface* (ed. E. M. Keramidas), pp. 156–163. Fairfax Station, VA: Interface Foundation.