

 Open access • Posted Content • DOI:10.1101/2020.08.07.242214

Fast and Flexible Estimation of Effective Migration Surfaces — [Source link](#)

Joseph Marcus, Wooseok Ha, Rina Foygel Barber, John Novembre

Institutions: University of Chicago, University of California, Berkeley

Published on: 07 Aug 2020 - bioRxiv (Cold Spring Harbor Laboratory)

Topics: Population

Related papers:

- [Modified Linear Projection for Large Spatial Data Sets](#)
- [Bayesian Nonstationary Spatial Modeling for Very Large Datasets](#)
- [A memory-free spatial additive mixed modeling for big spatial data](#)
- [Multi-scale shotgun stochastic search for large spatial datasets](#)
- [Efficient Spatial Pattern Analysis for Variation Decomposition Via Robust Sparse Regression](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/fast-and-flexible-estimation-of-effective-migration-surfaces-1ibdr0gn8z>

1 Fast and Flexible Estimation of Effective Migration Surfaces

2 Joseph H. Marcus^{1,*,}, Wooseok Ha^{2,*,}, Rina Foygel Barber^{3,†,} and John

3 Novembre^{1,4,†,}


4 ¹Department of Human Genetics, University of Chicago, Chicago, IL.

5 ²Department of Statistics, University of California, Berkeley, CA.

6 ³Department of Statistics, University of Chicago, Chicago, IL.

7 ⁴Department of Ecology and Evolution, University of Chicago, Chicago, IL.

8 contact: J.H.M.: jhmarcus@uchicago.edu, W.H.: haywse@berkeley.edu, R.F.B.:
9 rina@uchicago.edu, J.N.: jnovembre@uchicago.edu

10
11
12 * denotes co-first-authorship, † denotes co-mentorship, and  denotes corresponding
13 authorship.

14 **Abstract**

15 An important feature in spatial population genetic data is often “isolation-by-distance,” where
16 genetic differentiation tends to increase as individuals become more geographically distant. Re-
17 cently, [Petkova et al. \(2016\)](#) developed a statistical method called Estimating Effective Migration
18 Surfaces (EEMS) for visualizing spatially heterogeneous isolation-by-distance on a geographic
19 map. While EEMS is a powerful tool for depicting spatial population structure, it can suffer
20 from slow runtimes. Here we develop a related method called Fast Estimation of Effective Migra-
21 tion Surfaces (FEEMS). FEEMS uses a Gaussian Markov Random Field in a penalized likelihood
22 framework that allows for efficient optimization and output of effective migration surfaces. Fur-
23 ther, the efficient optimization facilitates the inference of migration parameters per edge in the
24 graph, rather than per node (as in EEMS). When tested with coalescent simulations, FEEMS
25 accurately recovers effective migration surfaces with complex gene-flow histories, including those
26 with anisotropy. Applications of FEEMS to population genetic data from North American gray
27 wolves shows it to perform comparably to EEMS, but with solutions obtained orders of mag-
28 nitude faster. Overall, FEEMS expands the ability of users to quickly visualize and interpret
29 spatial structure in their data.

30 Introduction

31 The relationship between geography and genetics has had enduring importance in evolutionary
32 biology (see [Felsenstein, 1982](#)). One fundamental consideration is that individuals who live near
33 one another tend to be more genetically similar than those who live far apart ([Kimura, 1953](#),
34 [Kimura and Weiss, 1964](#), [Malécot, 1948](#), [Wright, 1943, 1946](#)). This phenomenon is often referred
35 to as “isolation-by-distance” (IBD) and has been shown to be a pervasive feature in spatial popu-
36 lation genetic data across many species ([Dobzhansky and Wright, 1943](#), [Meirmans, 2012](#), [Slatkin,](#)
37 [1985](#)). Statistical methods that use both measures of genetic variation and geographic coordi-
38 nates to understand patterns of IBD have been widely applied ([Battey et al., 2020](#), [Bradburd](#)
39 [and Ralph, 2019](#)). One major challenge in these approaches is that the relationship between ge-
40 ography and genetics can be complex. Particularly, geographic features can influence migration
41 in localized regions leading to spatially heterogeneous patterns of genetic covariation ([Bradburd](#)
42 [and Ralph, 2019](#)).

43 Multiple approaches have been introduced to model non-homogeneous IBD in spatial popu-
44 lation genetic data ([Al-Asadi et al., 2019](#), [Bradburd et al., 2018](#), [Duforet-Frebourg and Blum,](#)
45 [2014](#), [Hanks and Hooten, 2013](#), [McRae, 2006](#), [Petkova et al., 2016](#), [Ringbauer et al., 2018](#), [Safner](#)
46 [et al., 2011](#)). Particularly relevant to our proposed approach is the work of [Petkova et al. \(2016\)](#)
47 and [Hanks and Hooten \(2013\)](#). Both approaches model genetic distance using the “resistance
48 distance” on a weighted graph. This distance metric is inspired by concepts of effective resistance
49 in circuit theory models, or alternatively understood as the commute time of a random walk on
50 a weighted graph or as a Gaussian graphical model (specifically a conditional auto-regressive
51 process) ([Chandra et al., 1996](#), [Hanks and Hooten, 2013](#), [Rue and Held, 2005](#)). Additionally,
52 the resistance distance approach is a computationally convenient and accurate approximation
53 to spatial coalescent models ([McRae, 2006](#)), though it has limitations in asymmetric migration
54 settings ([Lundgren and Ralph, 2019](#)).

55 [Hanks and Hooten \(2013\)](#) introduced a Bayesian model that uses measured ecological co-
56 variates, such as elevation, to help predict genetic distances across sub-populations. Specifically,
57 they use a graph-based model for genotypes observed at different spatial locations. Expected
58 genetic distances across sub-populations in their model are given by resistance distances com-
59 puted from the edge weights. They parameterize the edge weights of the graph to be a function
60 of known biogeographic covariates, linking local geographic features to genetic variation across
61 the landscape.

62 Concurrently, the Estimating Effective Migration Surfaces (EEMS) method was developed
63 to help interpret and visualize non-homogeneous gene-flow on a geographic map ([Petkova et al.,](#)
64 [2016](#), [Petkova, 2013](#)). EEMS uses resistance distances to approximate the between-sub-population
65 component of pairwise coalescent times in a “stepping-stone” model of migration and genetic drift
66 ([Kimura, 1953](#), [Kimura and Weiss, 1964](#)). EEMS models the within-sub-population component of
67 pairwise coalescent times, with a node-specific parameter. Instead of using known biogeographic
68 covariates to connect geographic features to genetic variation as in [Hanks and Hooten \(2013\)](#),
69 EEMS infers a set of edge weights (and diversity parameters) that explain the genetic distance
70 data. The inference is based on a hierarchical Bayesian model and a Voronoi-tessellation-based
71 prior to encourage piece-wise constant spatial smoothness in the fitted edge weights.

72 EEMS uses Markov Chain Monte Carlo (MCMC) and outputs a visualization of the posterior
73 mean for effective migration and a measure of genetic diversity for every spatial position of
74 the focal habitat. Regions with relatively low effective migration can be interpreted to have
75 reduced gene-flow over time whereas regions with relatively high migration can be interpreted
76 as having elevated gene-flow. EEMS has been applied to multiple systems to describe spatial
77 genetic structure, but despite EEMS’s advances in computational tractability with respect to the

78 previous work, the MCMC algorithm it uses can be slow to converge, in some cases leading to
79 days of computation time for large datasets (Peter et al., 2018).

80 These inference problems from spatial population genetics are related to a growing area of
81 interest in the graph signal processing literature referred to as “graph learning” (Dong et al.,
82 2019, Mateos et al., 2019). In graph learning, a noisy signal is measured as a scalar value at
83 a set of nodes from the graph, and the aim is then to infer non-negative edge weights that
84 reflect how spatially “smooth” the signal is with respect to the graph topology (Kalofolias, 2016).
85 In population genetic settings, this scalar could be an allele frequency measured at locations
86 in a discrete spatial habitat with effective migration rates between sub-populations. Like the
87 approach taken by Hanks and Hooten (2013), one widely used representation of smooth graph
88 signals is to associate the smoothness property with a Gaussian graphical model where the
89 precision matrix has the form of a graph Laplacian (Dong et al., 2016, Egilmez et al., 2016). The
90 probabilistic model defined on the graph signal then naturally gives rise to a likelihood for the
91 observed samples, and thus much of the literature in this area focuses on developing specialized
92 algorithms to efficiently solve optimization problems that allow reconstruction of the underlying
93 latent graph. For more information about graph learning and signal processing in general see
94 the excellent survey papers of Dong et al. (2019) and Mateos et al. (2019).

95 To position the present work in comparison to the “graph learning” literature, our contribu-
96 tions are twofold:

- 97 • In population genetics, it is impossible to collect individual genotypes across all the geo-
98 graphic locations and, as a result, we often work with many, often the majority, of nodes
99 having missing data. As far as we are aware, none of the work in graph signal processing
100 considers this scenario and thus their algorithms are not directly applicable to our setting.
101 In addition, if the number of the observed nodes is much smaller than the number of nodes
102 of a graph, one can project the large matrices associated with the graph to the space of
103 observed nodes, therefore allowing for fast and efficient computation.
- 104 • On the other hand, highly missing nodes in the observed signals can result in significant
105 degradation of the quality of the reconstructed graph unless it is regularized properly.
106 Motivated by the Voronoi-tessellation-based prior adopted in EEMS (Petkova et al., 2016),
107 we propose regularization that encourages spatial smoothness in the edge weights.

108 Building on advances in graph learning, we introduce a method, Fast Estimation of Effective
109 Migration Surfaces (FEEMS), that uses optimization rather than MCMC to obtain penalized-
110 likelihood-based estimates of effective migration parameters. In contrast to EEMS which uses a
111 node-specific parameterization of effective migration, we optimize over edge-specific parameters
112 allowing for more flexible migration processes to be fit, such as spatial anisotropy, in which
113 the migration process is not invariant to rotation of the coordinate system (e.g., migration is
114 more extensive along a particular axis). We develop a fast quasi-Newton optimization algorithm
115 (Nocedal and Wright, 2006) and apply it to a dataset of gray wolves from North America. The
116 output is comparable to the results of EEMS but is provided in orders of magnitude less time.
117 With this improvement in speed, FEEMS opens up the ability to perform fast exploratory and
118 iterative data analysis of spatial population structure.

119 Results

120 Overview of FEEMS

121 Figure 1 shows a visual schematic of the FEEMS method. The input data are genotypes and spa-
122 tial locations (e.g., latitudes and longitudes) for a set of individuals sampled across a geographic

123 region. We construct a dense spatial grid embedded in geographic space where nodes represent
124 sub-populations, and we assign individuals to nodes based on spatial proximity (see Supp. Fig. 1
125 for a visualization of the grid construction and node assignment procedure). The density of the
126 grid is user defined and must be explored to appropriately balance model-mis-specification and
127 computational burden. As the density of the lattice increases, the model is similar to discrete
128 approximations used for continuous spatial processes, but the increased density comes at the cost
129 of computational complexity.

130 We assume exchangeability of individuals within each sub-population and estimate allele
131 frequencies, $\hat{f}_j(k)$, for each sub-population, indexed by k , and single nucleotide polymorphism
132 (SNP), indexed by j , under a simple Binomial sampling model. We also use the recorded sample
133 sizes at each node to model the precision of the estimated allele frequency. The use of allele
134 frequencies allows a number of advantages in this context: (1) Allele frequencies can be more
135 easily shared between researchers than individual genotypes due to privacy concerns, which is
136 especially relevant in human population genetic studies; (2) We usually gain large computational
137 savings in memory and speed because in most population genetic studies the number of observed
138 locations, in which allele frequencies are estimated, is smaller than the total number of individuals
139 sampled i.e. many individuals are sampled from the same spatial location.

140 With the estimated allele frequencies in hand, we model the data at each SNP using an
141 approximate Gaussian model whose covariance is shared across all SNPs, in other words we
142 assume that the observed frequencies at each SNP is an independent realization of the same
143 spatial process after rescaling by SNP-specific variation factors. The latent frequency variables,
144 $f_j(k)$, are modeled as a Gaussian Markov Random Field (GMRF) with a sparse precision matrix
145 determined by the graph Laplacian and a set of residual variances. The graph's weighted edges,
146 denoted by w_{ij} between nodes i and j , represent gene-flow between the sub-populations (Fried-
147 man et al., 2008, Hanks and Hooten, 2013, Petkova et al., 2016). We analytically marginalize out
148 the latent frequency variables and use penalized restricted maximum likelihood to estimate the
149 edge weights of the graph after removing the SNP-specific mean allele frequencies by projecting
150 the data onto contrasts (Felsenstein, 1982, Hanks and Hooten, 2013, Petkova et al., 2016). Our
151 overall goal is to solve the following optimization problem:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{l} \leq \mathbf{w} \leq \mathbf{u}} \ell(\mathbf{w}) + \phi_{\lambda, \alpha}(\mathbf{w}),$$

152 where \mathbf{w} is a vector that stores all the unique elements of the weighted adjacency matrix, \mathbf{l} and
153 \mathbf{u} are element-wise non-negative lower and upper bounds for \mathbf{w} , and $\ell(\mathbf{w})$ is the negative log-
154 likelihood function that comes from the GMRF model described above. The penalty, $\phi_{\lambda, \alpha}(\mathbf{w})$,
155 controls how constant or smooth the output migration surface will be and is controlled by the
156 hyperparameters λ and α . Specifically, the hyperparameters determine a penalty function based
157 on the squared differences between edge weights for pairs of edges that share a common node,

$$\phi_{\lambda, \alpha}(\mathbf{w}) = \frac{\lambda}{2} \|\Delta(\mathbf{w} + \alpha \log(\mathbf{w}))\|_2^2,$$

158 where Δ is a signed graph incidence matrix indicating if two edges are connected to the same
159 node. Note that λ controls the overall strength of the penalization placed on the output of
160 migration surface while α controls the relative strength of the penalization on the logarithmic
161 scale. Thus, if the model is highly penalized, the fitted surface will favor a homogeneous spatial
162 process on the graph across orders of magnitude of edge weights and if the penalty is low,
163 more flexible graphs can be fit, but are potentially prone to over-fitting. Akin to the choice in
164 admixture models of the number of latent ancestral populations or clusters (K), inspecting the

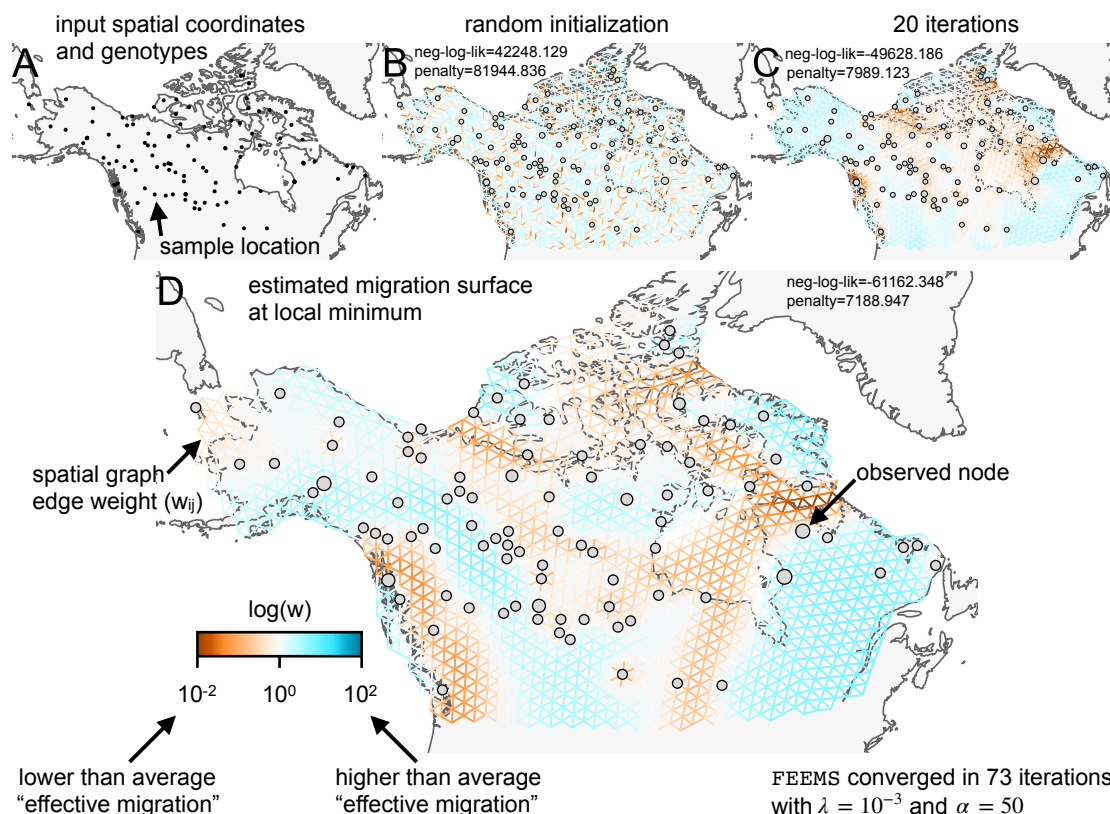


Figure 1: Schematic of the FEEMS model: The full panel shows a schematic of going from the raw data (spatial coordinates and genotypes) through optimization of the edge weights, representing effective migration, to convergence of FEEMS to a local optima. (A) Map of sample coordinates (black points) from a dataset of gray wolves from North America (Schweizer et al., 2016). The input to FEEMS are latitude and longitude coordinates as well as genotype data for each sample. (B) The spatial graph edge weights after random initialization uniformly over the graph to begin the optimization algorithm. (C) The edge weights after 20 iterations of running FEEMS, when the algorithm has not converged yet. (D) The final output of FEEMS after the algorithm has fully converged. The output is annotated with important features of the visualization.

165 outputs across a series of λ and α values is recommended and demonstrated (below). We use
166 sparse linear algebra routines to efficiently compute the objective function and gradient of our
167 parameters, allowing the use of widely applied quasi-Newton optimization algorithms (Nocedal
168 and Wright, 2006) implemented in standard numerical computing libraries like `scipy` (Virtanen
169 et al., 2020). See the materials and methods section for a detailed description of the statistical
170 models and algorithms used.

171 Evaluating FEEMS on “out of model” coalescent simulations

172 While our statistical model is not directly based on a population genetic process, it is useful
173 to see how it performs on simulated data under the coalescent stepping stone model. In these

174 simulations we know, by construction, the model we fit (FEEMS) is different from the true
175 model we simulate data under (the coalescent), allowing us to assess the robustness of the fit to
176 a controlled form of model mis-specification. In Figure 2 we use `msprime` (Kelleher et al., 2016) to
177 recapitulate and extend the results of Petkova et al. (2016), simulating data under the coalescent
178 in three simple migration scenarios with two different spatial sampling designs. Note that in
179 Supp. Fig. 2 we display a larger set of simulations with additional sampling configurations. For
180 brevity, here we only show results for $\lambda = .001$ and $\alpha = 50$, based on values that performed well
181 after experimental tuning. In Supp. Fig. 3 and Supp. Fig. 4, we also show results varying λ and
182 α for two migration scenarios with one particular sampling design.

183 The first migration scenario (Figure 2A-C) is a spatially homogeneous model where all the
184 migration rates are set to be a constant value on the graph, this is equivalent to simulating data
185 under an homogeneous isolation-by-distance model. In the second migration scenario (Figure 2D-
186 E) we simulate a non-homogeneous process by representing a geographic barrier to migration,
187 lowering the migration rates by a factor of 10 in the center of the habitat relative to the left and
188 right regions of the graph. Finally, in the third migration scenario (Figure 2G-I) we simulate
189 a pattern which corresponds to anisotropic migration with edges that point east/west being as-
190 signed to a five-fold higher migration rate than edges pointing north/south. For each migration
191 scenario we simulate two sampling designs. In the first “dense-sampling” sampling design (Fig-
192 ure 2B,E,I) we sample individuals for every node of the graph. Next, in the “sparse-sampling”
193 sampling design (Figure 2C,F,J) we randomly sample individuals for only 20% of the nodes.

194 As expected, FEEMS performs best when all the nodes are sampled on the graph, i.e. when
195 there is no missing data (Figure 2B,E,H). Interestingly, in the simulated scenarios with many
196 missing nodes, FEEMS can still partly recover the migration history, including the presence of
197 anisotropic migration (Figure 2F). A sampling scheme with a central gap leads to a slightly
198 narrower barrier in the heterogeneous migration scenario (Supp. Fig. 2I) and for the anisotropic
199 scenario, a degree of over-smoothness in the northern and southern regions of the center of the
200 graph (Supp. Fig. 2N). For the missing at random sampling design, FEEMS is able to recover
201 the relative edge weights surprisingly well for all scenarios, with the inference being the most
202 challenging when there is anisotropic migration. We emphasize that the potential for FEEMS
203 to recover anisotropic migration is novel relative to EEMS, which was parameterized for fitting
204 non-stationary isotropic migration histories and produces banding patterns perpendicular to the
205 axis of migration when applied to data from anisotropic coalescent simulations (see Petkova
206 et al. (2016) supplementary figure 2; see also Supp. Note “*Edge versus node parameterization*”
207 for a related discussion). Overall, even with sparsely sampled graphs, FEEMS is able to pro-
208 duce visualizations that qualitatively capture the migration history in “out of model” coalescent
209 simulations.

210 Application of FEEMS to genotype data from North American gray 211 wolves

212 To assess the performance of FEEMS on real data we used a previously published dataset of
213 111 gray wolves sampled across North America typed at 17,729 SNPs (Schweizer et al., 2016),
214 Supp. Fig. 5). This dataset has a number of advantageous features that make it a useful test case
215 for evaluating FEEMS: (1) The broad sampling range across North America includes a number of
216 relevant geographic features that, a priori, could conceivably lead to restricted gene-flow averaged
217 throughout the population history. These geographic features include mountain ranges, lakes and
218 island chains. (2) The scale of the data is consistent with many studies for non-model systems
219 whose spatial population structure is of interest. For instance, the relatively sparse sampling
220 leads to a challenging statistical problem where there is the potential for many unobserved

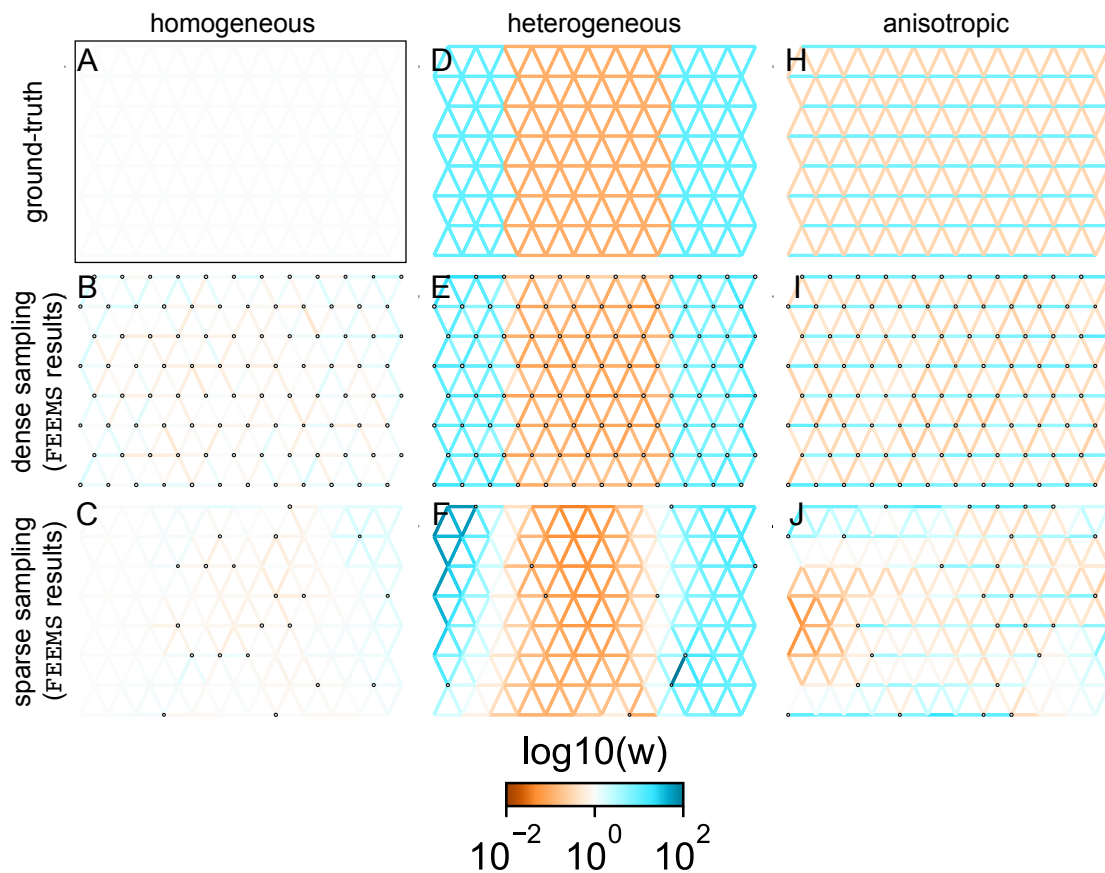


Figure 2: FEEMS fit to coalescent simulations: We run FEEMS on coalescent simulations, varying the migration history (columns) and sampling design (rows). The first column (A-C) shows the ground-truth and fit of FEEMS to coalescent simulations with a homogeneous migration history i.e. a single migration parameter for all edge weights. Note that the ground-truth simulation figures (A,D,F) display coalescent migration rates, not fitted effective migration rates output by FEEMS. The second column (D-F) shows the ground truth and fit of FEEMS to simulations with a heterogeneous migration history i.e. reduced gene-flow, with 10 fold lower migration, in the center of the habitat. The third column (H-J) shows the ground truth and fit of FEEMS to an anisotropic migration history with edge weights facing east-west having five fold higher migration than north-south. The second row (B,E,H) shows a sampling design with no missing observations on the graph. The final row (C,F,I) shows a sampling design with 80% of nodes missing at random.

221 nodes (sub-populations), depending the density of the grid chosen. Before applying FEEMS, we
 222 confirmed a signature of spatial structure in the data through regressing genetic distances on
 223 geographic distances and top genetic PCs against geographic coordinates (Supp. Fig. 6, 7, 8, 9).

224 We ran FEEMS with four different values of the smoothness parameter, λ (from large $\lambda =$
 225 10 to small $\lambda = 10^{-5}$), while setting the tuning parameter α to a value that we found that
 226 worked for multiple data applications and simulations ($\alpha = 50$, Figure 3). One interpretation
 227 of our regularization penalty is that it encourages fitting models of homogeneous and isotropic
 228 migration. When λ is very large (Figure 3A), we see FEEMS fits a model where all of the edge

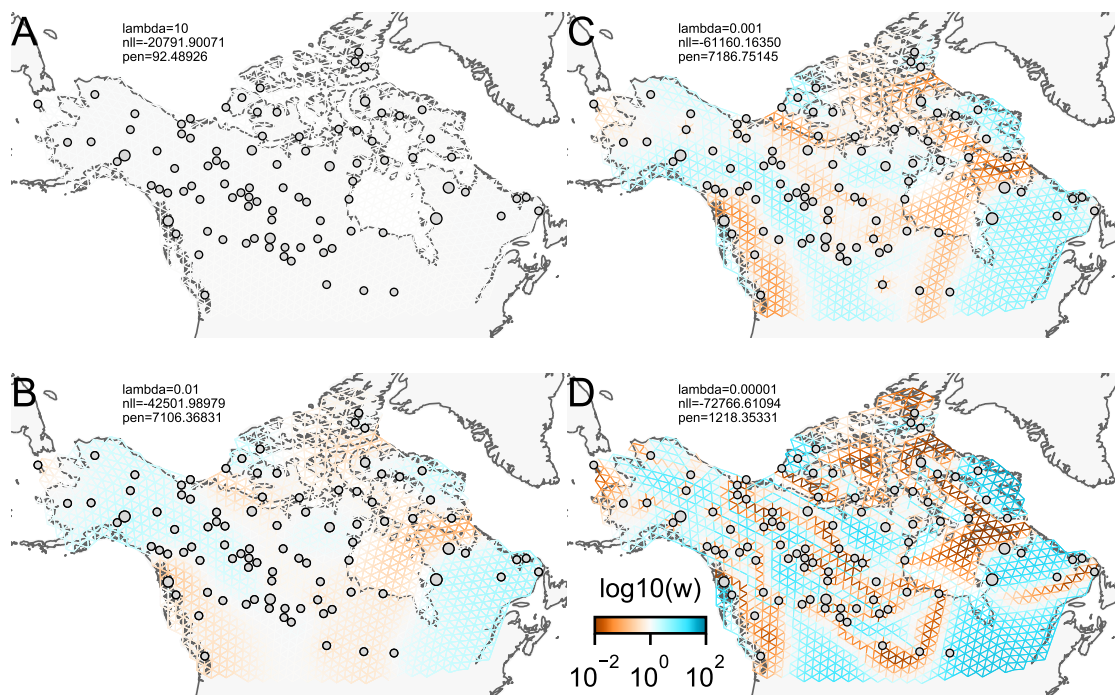


Figure 3: The fit of FEEMS to the North American gray wolf dataset for different choices of the smoothing regularization parameter λ : (A) $\lambda = 10$, (B) $\lambda = 10^{-2}$, (C) $\lambda = 10^{-3}$, and (D) $\lambda = 10^{-5}$. As expected, when λ decreases from large to small (A-D), the fitted graph becomes less smooth and presumably eventually over-fits to the data, revealing a patchy surface in (D), whereas, earlier in the regularization path FEEMS fits a completely homogeneous surface with all edge weights having the same fitted value, like in (A).

229 weights on the graph nearly equal the mean value, hence all the edge weights are colored white
230 in the relative log-scale. In this case, FEEMS is fitting a completely homogeneous migration
231 model where all the estimated edge weights get assigned the same value on the graph. Next,
232 as we sequentially lower the penalty parameter and (Figure 3B,C,D) the fitted graph begins to
233 appear more complex and heterogeneous as expected (discussed further below).

234 We also ran multiple replicates of ADMIXTURE for $K = 2$ to $K = 8$, selecting for each
235 K the highest likelihood run among replicates to visualize (Supp. Fig. 10). As expected in
236 a spatial genetic dataset, nearby samples have similar admixture proportions and continuous
237 gradients of changing ancestries are spread throughout the map (Bradburd et al., 2018). Whether
238 such gradients in admixture coefficients are due to isolation by distance or specific geographic
239 features that enhance or diminish the levels of genetic differentiation is an interpretive challenge.
240 Explicitly modeling the spatial locations and genetic distance jointly using a method like EEMS
241 or FEEMS is exactly designed to explore and visualize these types of questions in the data
242 (Petkova et al., 2016, Petkova, 2013).

243 Once we have run FEEMS for a grid of regularization parameters it is helpful to look more
244 closely at particular solutions that find a balance between spatial homogeneity and complexity
245 (Figure 4). Spatial features in the FEEMS visualization qualitatively matches the structure
246 plot output from ADMIXTURE using $K = 6$ (Supp. Fig. 10). We add labels on the figure to
247 highlight a number of pertinent features: (A) St. Lawrence Island, (B) the coastal islands and
248 mountain ranges in British Columbia, (C) The boundary of Boreal Forest and Tundra eco-regions

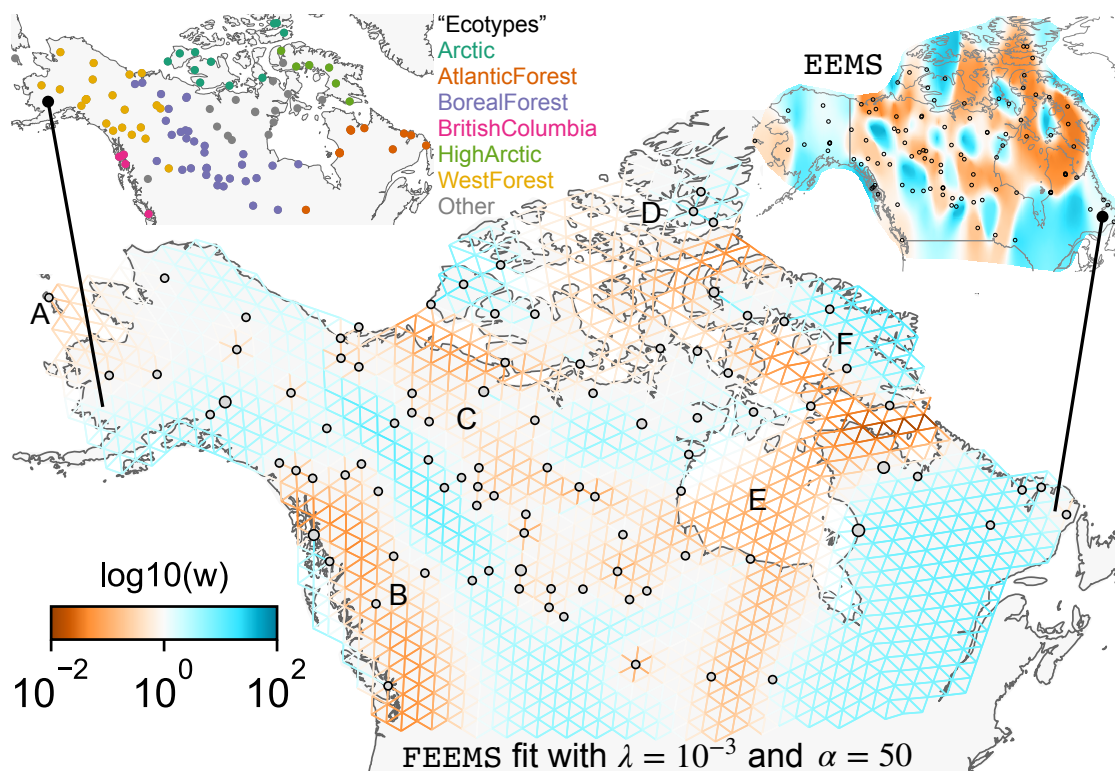


Figure 4: FEEMS applied to a population genetic dataset of North American gray wolves: We show the fit of FEEMS to a previously published dataset of North American gray wolves. This fit corresponds to a setting of tuning parameters at $\lambda = 10^{-3}$, $\alpha = 50$. We show the fitted parameters in log-scale with lower effective migration shown in orange and higher effective migration shown in blue. The bold text letters highlights a number of known geographic features that could have plausibly influenced Wolf migration over time: (A) St. Lawrence Island, (B) Coastal mountain ranges in British Columbia, (C) The boundary of Boreal Forest and Tundra eco-regions in the Shield Taiga, (D) Queen Elizabeth Islands, (E) Hudson Bay, and (F) Baffin Island. We also display two insets to help interpret the results and compare them to EEMS. In the top left inset we show a map of sample coordinates colored by an ecotype label provided by Schweizer et al. (2016). These labels were devised using a combination of genetic and ecological information for 94 “un-admixed” gray wolf samples, and the remaining samples were labeled “Other”. We can see these ecotype labels align well with the visualization output provided by FEEMS. In the right inset we display a visualization of the posterior mean effective migration rates from EEMS.

249 in the Shield Taiga, (D) Queen Elizabeth Islands, (E) Hudson Bay, and (F) Baffin Island. Many
250 of these features were described in Schweizer et al. (2016) by interpretation of ADMIXTURE,
251 PCA, and F_{ST} statistics. FEEMS is able to succinctly provide an interpretable view of these
252 data in a single visualization. Indeed many of these geographic features plausibly impact gray
253 wolf dispersal and population history (Schweizer et al., 2016).

Method	Sparse Grid (run-time)	Dense Grid (run-time)
EEMS	27.43hrs	N/A
FEEMS (total)	13.02s	3.54min
FEEMS (init)	8.25s	2min 11s
FEEMS ($\lambda = 10$)	604ms	10.7s
FEEMS ($\lambda = 10^{-2}$)	442ms	7.78s
FEEMS ($\lambda = 10^{-3}$)	917ms	9.18s
FEEMS ($\lambda = 10^{-5}$)	2.81s	53.9s

Table 1: Runtimes for FEEMS and EEMS on the North American gray wolf dataset: We show a table of runtimes for FEEMS and EEMS for two different grid densities, a sparse grid with 307 nodes and a dense grid with 1207 nodes. In the first two rows we show the total runtimes for both EEMS and FEEMS. In the following rows we show the total runtime for FEEMS, broken down into multiple components i.e. initialization time and the time to fit four solutions with different smoothing parameters.

254 Comparison to EEMS

255 We also ran EEMS on the same gray wolf dataset described throughout this manuscript. We
256 used default parameters provided by EEMS but set the number of burn-in iterations to 20×10^6 ,
257 MCMC iterations to 50×10^6 , and thinning intervals to 2000. We were unable to run EEMS in
258 a reasonable run time (≤ 3 days) for the dense spatial grid of 1207 nodes so we ran EEMS and
259 FEEMS on a sparser graph with 307 nodes.

260 We find that FEEMS is multiple orders of magnitude faster than EEMS, even when running
261 multiple runs of FEEMS for different regularization settings on both the sparse and dense graphs
262 (Table 1). The total FEEMS run-times in Table 1 also include the time needed to construct
263 relevant graph data structures and initialization. We note that constructing the graph and fitting
264 the model with very low regularization parameters are the most computationally demanding steps
265 in running FEEMS.

266 We find that many of the same geographic features that have reduced or enhanced gene-flow
267 are concordant between the two methods. The EEMS visualization, qualitatively, best matches
268 solutions of FEEMS with lower regularization penalties (Figure 4, Supp. Fig. 11); however, based
269 on the ADMIXTURE results and visual inspection in relation to known geographical features,
270 we find these solutions to be less satisfying compared to those with higher penalties and believe
271 the solutions output from lower penalties are likely overfitting the data. Indeed, we only see a
272 small gain in the R^2 when comparing observed and fitted distances computed from the output
273 graphs of Figure 3C and Figure 3D (Supp. Fig. 6). We note that in many of the EEMS runs the
274 MCMC appears to not have converged (based on visual inspection of trace plots) even after a
275 large number of iterations.

276 Discussion

277 FEEMS is a fast approach that provides an interpretable view of spatial population structure
278 in real datasets and simulations. We want to emphasize that beyond being a fast optimization
279 approach for inferring population structure, our parameterization of the likelihood opens up a
280 number of exciting new directions for improving spatial population genetic inference. Notably,
281 one major difference between EEMS and FEEMS is that in FEEMS each edge weight is assigned
282 its own parameter to be estimated whereas, in EEMS, each node is assigned a parameter and

283 each edge is constrained to be the average effective migration between the nodes it connects (see
284 Materials and Methods and Supp. Note “*Edge versus node parameterization*” for details). The
285 node-based parameterization in EEMS makes it difficult to incorporate anisotropy and asymme-
286 teric migration (Lundgren and Ralph, 2019). As we have shown here, FEEMS’s simple and novel
287 parameterization already has potential to fit anisotropic migration (as shown in coalescent sim-
288 ulations) and may be extendable to other more complex migration processes (such as long-range
289 migration, see below).

290 FEEMS estimates one set of graph edge weights for each setting of the tuning parameters λ
291 and α which control the smoothness of the fitted edge-weights. One general challenge, which is
292 not unique to this method, is selecting a particular set of tuning parameters. A natural approach
293 is to use cross-validation, which estimates the out-of-sample fit of FEEMS for a particular model
294 (selection of λ and α). While cross-validation might be useful for assessing the choice of tuning
295 parameters, in preliminary experiments applying cross validation by holding out individuals or
296 observed nodes, and assessing performance via the model-likelihood, we found too much variation
297 across cross-validation folds to reliably tune λ and α (results not shown). In order to reduce the
298 variation across different folds, we also applied cross-validation with standardization (Bradburd
299 et al., 2018), where the model-likelihood is standardized for each fold, and approximate leave-one-
300 out cross-validation Wilson et al. (2020), where the leave-one-out CV likelihood is approximated
301 with a few steps of the quasi-Newton algorithm warm-started from the full training set migration
302 surfaces. Neither of these approaches were promising for reliable model selection. We suspect
303 this poor performance is due to spatial dependency of allele frequencies and the large fraction
304 of unobserved nodes. In unsupervised learning settings like this one, it is not obvious that
305 estimates of out of sample fit will always lead to the most biologically interpretable models and
306 sometimes other metrics can be preferable, such as those based on the stability of the solution to
307 perturbations like variable initialization (Wu et al., 2016). Stability-based approaches for model
308 selection could be a fruitful future direction to develop a formal procedure for tuning. Currently,
309 we recommend fitting FEEMS with several values of the tuning parameters and interpreting the
310 results in an integrative fashion with other analyses.

311 We find it useful to fit FEEMS to a sequential grid of regularization parameters and to look at
312 what features are consistent and vary across multiple fits. Informally, one can gain an indication
313 of the strongest features in the data by looking at the order they appear in the regularization
314 path i.e. what features overcome the strong penalization of smoothness in the data and that
315 are highly supported by the likelihood. For example, early in the regularization path, we see
316 regions of reduced gene-flow occurring in the west coast of Canada that presumably correspond to
317 Coastal mountain ranges and islands in British Columbia (Figure 3B) and this reduced gene-flow
318 appears throughout more flexible fits with lower λ .

319 Beyond tuning the unknown parameters, we encountered other challenges when solving this
320 difficult optimization problem. Notably, the objective function we optimize is non-convex so any
321 visualization output by FEEMS should be considered a local optimum and, as a result, with
322 different initialization we could get different results. Overall, we found the output visualization
323 was not sensitive to initialization, and thus our default setting is constant initialization fitted
324 under an homogeneous isolation by distance model (See Materials and Methods).

325 When comparing to EEMS, we found FEEMS to be much faster (Table 1). While this is
326 encouraging, care must be taken because the goals and outputs of FEEMS and EEMS have
327 a number of differences. FEEMS fits a sequential grid of solutions for different regularization
328 parameters whereas EEMS infers a posterior distribution and outputs the posterior mean as
329 a point estimate. So in order to compare the results, in principal, one must compare many
330 FEEMS visualizations to a single EEMS visualization. FEEMS is not a Bayesian method and
331 unlike EEMS, which explores the entire landscape of the posterior distribution, FEEMS returns

332 a particular point estimate: a local minimum point of the optimization landscape. Setting the
333 prior hyper-parameters in EEMS act somewhat like a choice of tuning parameters, except that
334 EEMS uses hierarchical priors that in principle allow for exploration of multiple scales of spatial
335 structure in a single run; this arguably results in less sensitivity to user-based settings but requires
336 potentially long computation times for adequate MCMC convergence.

337 One natural extension to FEEMS, pertinent to a number of biological systems, is incorpo-
338 rating long-range migration (Bradburd et al., 2016, Pickrell and Pritchard, 2012). In this work
339 we have used a triangular lattice embedded in geographic space and enforced smoothness in
340 nearby edge weights through penalizing their squared differences (see Materials and Methods).
341 We could imagine changing the structure of the graph by adding edges to allow for long-range
342 connection; however our current regularization scheme would not be appropriate for this setting.
343 Instead, we could imagine adding an additional penalty to the objective, which would only allow
344 a few long range connections to be tolerated. This could be considered to be a combination of
345 two existing approaches for graph-based inference, graphical lasso (GLASSO) and graph Lapla-
346 cian smoothing, combining the smoothness assumption for nearby connections and the sparsity
347 assumption for long-range connections (Friedman et al., 2008, Wang et al., 2016). Another po-
348 tential methodological avenue to incorporate long-range migration is to use a “greedy” approach.
349 We could imagine adding long-range edges one a time, guided by re-fitting the spatial model
350 and taking a data driven approach to select particular long-range edges to include. The pro-
351 posed greedy approach could be considered to be a spatial graph analog of TreeMix (Pickrell and
352 Pritchard, 2012).

353 Another interesting extension would be to incorporate asymmetric migration into the frame-
354 work of resistance distance and Gaussian Markov Random Field based models. Recently, Hanks
355 (2015) developed a promising new framework for deriving the stationary distribution of a con-
356 tinuous time stochastic process with asymmetric migration on a spatial graph. Interestingly, the
357 expected distance of this process has a similar “flavor” to the resistance distance based models,
358 in that it depends on the pseudo-inverse of a function of the graph Laplacian. Hanks (2015) used
359 MCMC to estimate the effect of known covariates on the edge weights of the spatial graph. Future
360 work could adapt this framework into the penalized optimization approach we have considered
361 here, where adjacent edge weights are encouraged to be smooth.

362 Finally, when interpreted as mechanistic rather than statistical models, both EEMS and
363 FEEMS implicitly assume time-stationarity, so the estimated migration parameters should be
364 considered to be “effective” in the sense of being averaged over time in a reality where migration
365 rates are dynamic and changing (Pickrell and Reich, 2014). The MAPS method is one recent
366 advance that utilizes long stretches of shared haplotypes between pairs of individuals to perform
367 Bayesian inference of time varying migration rates and population sizes (Al-Asadi et al., 2019).
368 With the growing ability to extract high quality DNA from ancient samples, another exciting
369 future direction would be to apply FEEMS to ancient DNA datasets over different time transects
370 in the same focal geographic region to elucidate changing migration histories (Mathieson et al.,
371 2018). There are a number of technical challenges in ancient DNA data that make this a difficult
372 problem, particularly high levels of missing and low-coverage data. Our modeling approach could
373 be potentially more robust, in that it takes allele frequencies as input, which may be estimable
374 from dozens of ancient samples at the same spatial location, in spite of high degrees of missingness
375 (Korneliussen et al., 2014).

376 In closing, we look back to a review titled “How Can We Infer Geography and History from
377 Gene Frequencies?” published in 1982 (Felsenstein, 1982). In this review, Felsenstein laid out
378 fundamental open problems in statistical inference in population genetic data, a few of which we
379 restate as they are particularly motivating for our work:

- 380 • “For any given covariance matrix, is there a corresponding migration matrix which would

381 be expected to lead to it? If so, how can we find it?”

382 • “How can we characterize the set of possible migration matrices which are compatible with
383 a given set of observed covariances?”

384 • “How can we confine our attention to migration patterns which are consistent with the
385 known geometric co-ordinates of the populations?”

386 • “How can we make valid statistical estimates of parameters of stepping stone models?”

387 The methods developed here aim to help address these longstanding problems in statistical
388 population genetics and to provide a foundation for future work to elucidate the role of geography
389 and dispersal in ecological and evolutionary processes.

390 Materials and Methods

391 Model description

392 See Supp. Note “*Mathematical notation*” for a detailed description of the notation used to describe
393 the model. To visualize and model spatial patterns in a given population genetic dataset, FEEMS
394 uses an undirected graph, $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = d$, where nodes represent sub-populations and
395 edge weights $(w_{ij})_{(i,j) \in \mathcal{E}}$ represent the level of gene-flow between sub-populations i and j . For
396 computational convenience, we assume \mathcal{G} is a highly sparse graph, specifically a triangular grid
397 that is embedded in geographic space around the sample coordinates. We observe a genotype
398 matrix, $\mathbf{Y} \in R^{n \times p}$, with n rows representing individuals and p columns representing SNPs. We
399 imagine diploid individuals are sampled on the nodes of \mathcal{G} so that $y_{ij}(k) \in \{0, 1, 2\}$ records the
400 count of some arbitrarily predefined allele in individual i , SNP j , on node $k \in \mathcal{V}$. We assume a
401 commonly used simple Binomial sampling model for the genotypes:

$$y_{ij}(k) | f_j(k) \sim \text{Binomial}(2, f_j(k)), \quad (1)$$

402 where conditional on $f_j(k)$ for all j, k , the $y_{ij}(k)$'s are independent. We then estimate an allele
403 frequency at each node and SNP by maximum likelihood:

$$\hat{f}_j(k) = \frac{\sum_{i=1}^{n_k} y_{ij}(k)}{2n_k},$$

404 where n_k is the number of individuals sampled at node k . We estimate allele frequencies at o of
405 the observed nodes out of d total nodes on the graph. From (1), the estimated frequency in a
406 particular sub-population, conditional on the latent allele frequency, will approximately follow a
407 Gaussian distribution:

$$\hat{f}_j(k) | f_j(k) \sim \mathcal{N}\left(f_j(k), \frac{f_j(k)(1 - f_j(k))}{2n_k}\right).$$

408 Using vector notation, we represent the joint model of estimated allele frequencies as:

$$\hat{\mathbf{f}}_j | \mathbf{f}_j \sim \mathcal{N}_o(\mathbf{A}\mathbf{f}_j, \text{diag}(\mathbf{d}_{\mathbf{f}, \mathbf{n}})), \quad (2)$$

409 where $\hat{\mathbf{f}}_j$ is a $o \times 1$ vector of estimated allele frequencies at observed nodes, \mathbf{f}_j is a $d \times 1$ vector
410 of latent allele frequencies at all the nodes (both observed and unobserved), and \mathbf{A} is a $o \times d$

411 node assignment matrix where $\mathbf{A}_{k\ell} = 1$ if the k th estimated allele frequency comes from sub-
 412 population ℓ and $\mathbf{A}_{k\ell} = 0$ otherwise; and $\text{diag}(\mathbf{d}_{\mathbf{f}, \mathbf{n}})$ denotes a $o \times o$ diagonal matrix whose
 413 diagonal elements corresponds to the appropriate variance term at observed nodes.

414 To summarize, we estimate allele frequencies from a subset of nodes on the graph and define
 415 latent allele frequencies for all the nodes of the graph. The assignment matrix \mathbf{A} maps these
 416 latent allele frequencies to our observations. Our summary statistics (the data) are thus $(\hat{\mathbf{F}}, \mathbf{n})$
 417 where $\hat{\mathbf{F}}$ is a $o \times p$ matrix of estimated allele frequencies and \mathbf{n} is a $o \times 1$ vector of sample sizes
 418 for every observed node. We assume the latent allele frequencies come from a Gaussian Markov
 419 Random Field:

$$\mathbf{f}_j \sim \mathcal{N}_a\left(\mu_j \mathbf{1}, \mu_j(1 - \mu_j) \mathbf{L}^\dagger\right), \quad (3)$$

420 where \mathbf{L} is the graph Laplacian and μ_j represents the average allele frequency across all of the
 421 sub-populations. Note that the multiplication by the SNP-specific factor $\mu_j(1 - \mu_j)$ ensures that
 422 the variance of the latent allele frequencies vanishes as the average allele frequency approaches
 423 to 0 or 1. One interpretation of this model is that the expected squared Euclidean distance
 424 between latent allele frequencies on the graph, after being re-scaled by $\mu_j(1 - \mu_j)$, is exactly the
 425 resistance distance of an electrical circuit (Hanks and Hooten, 2013, McRae, 2006):

$$r_{j,ik} = \frac{E\left[\left(f_j(i) - f_j(k)\right)^2\right]}{\mu_j(1 - \mu_j)} = (\mathbf{o}_i - \mathbf{o}_k)^\top \mathbf{L}^\dagger (\mathbf{o}_i - \mathbf{o}_k) = \mathbf{L}_{ii}^\dagger - 2\mathbf{L}_{ik}^\dagger + \mathbf{L}_{kk}^\dagger,$$

426 where \mathbf{o}_i is a one-hot vector (i.e., storing a 1 in element i and zeros elsewhere). It is known that
 427 the resistance distance is equivalent to the expected commute time between nodes i and k of a
 428 random walker on the weighted graph \mathcal{G} (Chandra et al., 1996). Additionally, the model (3) forms
 429 a Markov random field, and thus any latent allele frequency $f_j(i)$ is conditionally independent of
 430 all other allele frequencies given its neighbors which are encoded by nonzero elements of \mathbf{L} (Koller
 431 and Friedman, 2009, Lauritzen, 1996).¹

432 Using the law of total variance formula, we can derive from (2), (3) an analytic form for the
 433 marginal likelihood. Before proceeding, however, we further approximate the model by assuming
 434 $\frac{1}{2}f_j(k)(1 - f_j(k)) \approx \sigma^2\mu_j(1 - \mu_j)$ for all j and k . This assumption is mainly for computational
 435 purposes and may be a coarse approximation in general. On the other hand, the assumption is
 436 not too strong if we exclude SNPs with extremely rare allele frequencies, and more importantly,
 437 we find it leads to a good empirical performance, both statistically and computationally. With
 438 this approximation the residual variance parameter σ^2 is still unknown and needs to be estimated.

439 With the above considerations, we arrive at the following marginal likelihood:²

$$\hat{\mathbf{f}}_j \sim \sqrt{\mu_j(1 - \mu_j)} \cdot \mathcal{N}_o\left(\mu_j \mathbf{1}, \mathbf{A} \mathbf{L}^\dagger \mathbf{A}^\top + \sigma^2 \text{diag}(\mathbf{n}^{-1})\right), \quad (4)$$

440 where $\text{diag}(\mathbf{n}^{-1})$ is a $o \times o$ diagonal matrix computed from the sample sizes at observed nodes.
 441 To remove the SNP means we transform the estimated frequencies by a contrast matrix, $\mathbf{C} \in$
 442 $\mathbb{R}^{(o-1) \times o}$, that is orthogonal to the one-vector:

¹Specifically, since we use a triangular grid embedded in geographic space to define the graph \mathcal{G} , the pattern of nonzero elements is prefixed by the structure of the sparse triangular grid.

²To be more precise, under (2), (3), the law of total variance formula leads to specific formulas for the mean and variance structure as given in (4), whereas the marginal distribution of \hat{f}_j is not necessarily a Gaussian distribution. We simply chose the Gaussian distribution here to enable easy calculation for the data likelihood. We believe the specific choice of the likelihood is not that critical as long as the first two moments of the distribution can be matched closely.

$$\mathbf{C}\hat{\mathbf{f}}_j \sim \sqrt{\mu_j(1-\mu_j)} \cdot \mathcal{N}_{o-1} \left(\mathbf{0}, \mathbf{C}\mathbf{A}\mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \sigma^2 \mathbf{C}\text{diag}(\mathbf{n}^{-1})\mathbf{C}^\top \right). \quad (5)$$

443 Letting $\hat{\Sigma} = \frac{1}{p} \widehat{\mathbf{F}}_s \widehat{\mathbf{F}}_s^\top$ be the $o \times o$ sample covariance matrix of estimated allele frequencies after
 444 rescaling, i.e. $\widehat{\mathbf{F}}_s$ is a matrix formed by rescaling the columns of $\widehat{\mathbf{F}}$ by $\sqrt{\hat{\mu}_j(1-\hat{\mu}_j)}$, where $\hat{\mu}_j$ is
 445 an estimate of the average allele frequency (see above). We can then express the model in terms
 446 of the transformed sample covariance matrix:

$$p \cdot \mathbf{C}\hat{\Sigma}\mathbf{C}^\top \sim \mathcal{W}_{o-1} \left(\mathbf{C}\mathbf{A}\mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \sigma^2 \mathbf{C}\text{diag}(\mathbf{n}^{-1})\mathbf{C}^\top, p \right), \quad (6)$$

447 where \mathcal{W}_p denotes a Wishart distribution with p degrees of freedom.³ Note we can equivalently
 448 use the sample squared Euclidean distance (often referred to as a genetic distance) as a summary
 449 statistic: letting $\hat{\mathbf{D}}$ be the genetic distance matrix with $\mathbf{D}_{ik} = \sum_{j=1}^p (\hat{f}_j(i) - \hat{f}_j(k))^2 / p \cdot \hat{\mu}_j(1-\hat{\mu}_j)$,
 450 we have

$$\hat{\mathbf{D}} = \mathbf{1}\text{diag}(\hat{\Sigma})^\top + \text{diag}(\hat{\Sigma})\mathbf{1}^\top - 2\hat{\Sigma},$$

451 and so

$$\mathbf{C}\hat{\mathbf{D}}\mathbf{C}^\top = -2\mathbf{C}\hat{\Sigma}\mathbf{C}^\top,$$

452 using the fact that the contrast matrix \mathbf{C} is orthogonal to the one-vector. Thus we can use the
 453 same spatial covariance model implied by the allele frequencies once we project the distances on
 454 to the space of contrasts:⁴

$$-\frac{p}{2} \cdot \mathbf{C}\hat{\mathbf{D}}\mathbf{C}^\top \sim \mathcal{W}_{o-1} \left(\mathbf{C}\mathbf{A}\mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \sigma^2 \mathbf{C}\text{diag}(\mathbf{n}^{-1})\mathbf{C}^\top, p \right).$$

455 Overall, the negative log-likelihood function implied by our spatial model is (ignoring constant
 456 terms):

$$\begin{aligned} \ell(\mathbf{w}, \sigma^2; \mathbf{C}\hat{\Sigma}\mathbf{C}^\top) &= p \cdot \text{tr} \left(\left(\mathbf{C}\mathbf{A}\mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \sigma^2 \mathbf{C}\text{diag}(\mathbf{n}^{-1})\mathbf{C}^\top \right)^{-1} \mathbf{C}\hat{\Sigma}\mathbf{C}^\top \right) \\ &\quad - p \cdot \log \det \left(\mathbf{C}\mathbf{A}\mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \sigma^2 \mathbf{C}\text{diag}(\mathbf{n}^{-1})\mathbf{C}^\top \right)^{-1}, \quad (7) \end{aligned}$$

³Our model (6) says that the p SNPs are independent. This assumption is unlikely to hold when SNPs are in close chromosomal proximity and analyzed due to linkage disequilibrium. In (Petkova et al., 2016), they introduce the effective degree of freedom $\nu \in [o-1, p]$ to account for such dependency and instead consider the model $\nu \cdot \mathbf{C}\hat{\Sigma}\mathbf{C}^\top \sim \mathcal{W}_{o-1}(\mathbf{C}\mathbf{A}\mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \sigma^2 \mathbf{C}\text{diag}(\mathbf{n}^{-1})\mathbf{C}^\top, \nu)$ with ν being estimated alongside other model parameters. In FEEMS, we note that the degree of freedom parameter does not affect the point estimate produced by our algorithm.

⁴We remark that besides the effective degree of freedom and the SNP-specific re-scaling by $\mu_j(1-\mu_j)$, the EEMS (Petkova et al., 2016) and FEEMS likelihoods are equivalent up to constant factors, as long as only one individual is observed per node and the residual variance σ^2 is allowed to vary across nodes—See Supp. Note “Jointly estimating the residual variance and edge weights” for details. In addition, constant factors are effectively absorbed into the unknown model parameters \mathbf{L} and σ^2 and therefore it does not affect the estimation of effective migration rates, up to constant factors.

457 where $\mathbf{w} \in R^m$ is a vectorized form of the non-zero lower-triangular entries of the weighted
 458 adjacency matrix \mathbf{W} (recall that the graph Laplacian is completely defined by the edge weights
 459 $\mathbf{L} = \text{diag}(\mathbf{W}\mathbf{1}) - \mathbf{W}$ so there is an implicit dependency here). Since the graph is a triangular
 460 lattice, we only need to consider the non-zero entries to save computational time, i.e. not all
 461 sub-populations are connected to each other.

462 One key difference between EEMS (Petkova et al., 2016) and FEEMS is how the edge weights
 463 are parameterized. In EEMS, each node is given an effective migration parameter m_i for node
 464 $i \in \mathcal{V}$ and the edge weight is parameterized as the average between the nodes it connects, i.e. $w_{ij} =$
 465 $(m_i + m_j)/2$ for $(i, j) \in \mathcal{E}$. FEEMS, on the other hand, assigns a parameter to every nonzero
 466 edge-weight. The former has fewer parameters, with the specific consequence that it only allows
 467 isotropy and imposes an additional degree of similarity among edge weights; instead, in the
 468 latter, the edge weights are free to vary apart from the regularization imposed by the penalty.
 469 See Supp. Note “Edge versus node parameterization” and Supp. Fig. 16 for more details.

470 Penalty description

471 As mentioned previously we would like to encourage that nearby edge weights on the graph
 472 have similar values to each other. This can be performed by penalizing the squared differences
 473 between all edges connected to the same node, i.e. spatially adjacent edges:

$$\phi_{\lambda, \alpha}(\mathbf{w}) = \frac{\lambda}{2} \sum_{i \in \mathcal{V}} \sum_{k, \ell \in \mathcal{E}(i)} \left((w_{ik} + \alpha \log(w_{ik})) - (w_{i\ell} + \alpha \log(w_{i\ell})) \right)^2,$$

474 where $\phi_{\lambda, \alpha}$ is our penalty function that represents the total amount of smoothness on the graph
 475 and $\mathcal{E}(i)$ denotes the set of edges that connected to node i . Here we penalize a weighted combina-
 476 tion of the edge weights on the original scale and logarithmic-scale where α , a tuning parameter,
 477 controls how strong the penalization is placed on the logarithmic scale—in the special case that
 478 $\alpha = 0$, it reduces to the commonly used Laplacian smoothing-type penalty. Adding a logarithmic
 479 scale leads to smooth graphs for small edge values and thus allow for an additional degree of
 480 flexibility across orders of magnitude of edge weights. The smoothness parameter, λ , controls
 481 the overall contribution of the penalty to the objective function. It is convenient to write the
 482 penalty in matrix-vector form which we will use throughout:

$$\phi_{\lambda, \alpha}(\mathbf{w}) = \frac{\lambda}{2} \|\Delta(\mathbf{w} + \alpha \log(\mathbf{w}))\|_2^2, \quad (8)$$

483 where Δ is a signed graph incidence matrix derived from a unweighted graph denoting if pairs
 484 of edges are connected to the same node. This penalty function (8) is also scale invariant, in the
 485 sense that for any $c > 0$, $\phi_{\lambda, \alpha}(\mathbf{w}) = \phi_{c^{-2}\lambda, c\alpha}(c\mathbf{w})$.

486 One might wonder whether it is possible to use the ℓ_1 norm in the penalty form (8) in place
 487 of the ℓ_2 norm. While it is known that the ℓ_1 norm might increase local adaptivity and better
 488 capture the sharp changes of the underlying structure of the latent allele frequencies, (e.g. Wang
 489 et al., 2016), in our case, we found an inferior performance when using the ℓ_1 norm over the ℓ_2
 490 norm—in particular, our primary application of interest is the regime of highly missing nodes,
 491 i.e. $o \ll d$, in which case the global smoothing seems somewhat necessary to encourage stable
 492 recovery of the edge weights at regions with sparsely observed nodes (see Supp. Note “Smooth
 493 penalty with ℓ_1 norm”). In addition, adding the penalty $\phi_{\lambda, \alpha}(\mathbf{w})$ allows us to implement faster
 494 algorithms to solve the optimization problem due to the differentiability of the ℓ_2 norm, and as
 495 a result, it leads to better overall computational savings and a simpler implementation.

496 Optimization

497 Putting (7) and (8) together, we infer the migration edge weights $\hat{\mathbf{w}}$ by minimizing the following
 498 penalized negative log-likelihood function:

$$\begin{aligned} \hat{\mathbf{w}} &= \arg \min_{\mathbf{l} \leq \mathbf{w} \leq \mathbf{u}} \ell(\mathbf{w}, \sigma^2; \mathbf{C}\hat{\Sigma}\mathbf{C}^\top) + \phi_{\lambda, \alpha}(\mathbf{w}) \\ &= \arg \min_{\mathbf{l} \leq \mathbf{w} \leq \mathbf{u}} \left[p \cdot \text{tr} \left(\left(\mathbf{C}\mathbf{A}\mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \sigma^2 \mathbf{C} \text{diag}(\mathbf{n}^{-1}) \mathbf{C}^\top \right)^{-1} \mathbf{C}\hat{\Sigma}\mathbf{C}^\top \right) \right. \\ &\quad \left. - p \cdot \log \det \left(\mathbf{C}\mathbf{A}\mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \sigma^2 \mathbf{C} \text{diag}(\mathbf{n}^{-1}) \mathbf{C}^\top \right)^{-1} + \frac{\lambda}{2} \|\Delta(\mathbf{w} + \alpha \log(\mathbf{w}))\|_2^2 \right], \end{aligned} \quad (9)$$

499 where $\mathbf{l}, \mathbf{u} \in R_+^m$ represent respectively the entrywise lower- and upper bounds on \mathbf{w} , i.e. we
 500 constrain the lower- and upper bound of the edge weights to \mathbf{l} and \mathbf{u} throughout the optimization.
 501 When no prior information is available on the range of the edge weights, we often set $\mathbf{l} = \mathbf{0}$ and
 502 $\mathbf{u} = +\infty$.

503 One advantage of the formulation of (9) is the use of the vector form parameterization $\mathbf{w} \in R_+^m$
 504 of the symmetric weighted adjacency matrix $\mathbf{W} \in R_+^{d \times d}$. In our triangular graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$,
 505 the number of non-zero lower-triangular entries is $m = \mathcal{O}(d) \ll d^2$, so working directly on the
 506 space of vector parameterization saves computational cost. In addition, this avoids the symmetry
 507 constraint imposed on the adjacency matrix \mathbf{W} , hence making optimization easier (Kalofolias,
 508 2016).

509 We solve the optimization problem using a constrained quasi-Newton optimization algorithm,
 510 specifically L-BFGS implemented in `scipy` (Byrd et al., 1995, Virtanen et al., 2020).⁵ Since our
 511 objective (9) is non-convex, the L-BFGS algorithm is guaranteed to converge only to a local
 512 minimum. Even so, we empirically observe that local minima starting from different initial
 513 points are qualitatively similar to each other across many datasets. The L-BFGS algorithm
 514 requires gradient and objective values as inputs. Note the naive computation of the objective (9)
 515 is computationally prohibitive since inverting the graph Laplacian has complexity $\mathcal{O}(d^3)$. We take
 516 advantage of the sparsity of the graph and specific structure of the problem to efficiently compute
 517 gradient and objective values. In theory, our implementation has computational complexity of
 518 $\mathcal{O}(do + o^3)$ per iteration which, in the setting of $o \ll d$, is substantially smaller than $\mathcal{O}(d^3)$.⁶

519 Estimating the residual variance and edge weights under the null model

520 For estimating the residual variance parameter σ^2 , we first estimate it via maximum likelihood
 521 assuming homogeneous isolation by distance. This corresponds to the scenario where every edge-
 522 weight in the graph is given the exact same unknown parameter value w_0 . Under this model
 523 we only have two unknown parameters w_0 and the residual variance σ^2 . We estimate these
 524 two parameters by jointly optimizing the marginal likelihood using a Nelder-Mead algorithm
 525 implemented in `scipy` (Virtanen et al., 2020). This requires only likelihood computations which
 526 are efficient due to the sparse nature of the graph. This optimization routine outputs an estimate

⁵We solve using linearized ADMM when the penalty function is ℓ_1 norm, i.e. $\lambda \|\Delta((w) + \alpha \log((w)))\|_1$ (Boyd et al., 2011).

⁶More precisely, it is possible to achieve $\mathcal{O}(do + o^3)$ per-iteration complexity if one employs a solver that is specially designed for sparse Laplacian system. In our work we use sparse Cholesky factorization which may slightly slow down the per-iteration complexity. See Supp. Material for the details of the gradient and objective computation.

527 of the residual variance $\hat{\sigma}^2$ and the null edge weight \hat{w}_0 , which can be used to construct $\mathbf{W}(\hat{w}_0)$
528 and in turn $\mathbf{L}(\hat{w}_0)$.

529 One strategy we found effective is to fit the model of homogeneous isolation by distance and
530 then fix the estimated residual variance $\hat{\sigma}^2$ throughout later fits of the more flexible penalized
531 models—See Supp. Note “*Jointly estimating the residual variance and edge weights*”. Additionally
532 we find that initializing the edge weights to \hat{w}_0 to be a useful and intuitive strategy to set the
533 initial values for the entries of \mathbf{w} to the correct scale.

534 Data description and quality control

535 We analyzed a population genetic dataset of North American gray wolves previously published in
536 [Schweizer et al. \(2016\)](#). For this, we downloaded plink formatted files and spatial coordinates from
537 <https://doi.org/10.5061/dryad.c9b25>. We removed all SNPs with minor allele frequency less
538 than 5% and with missingness greater than 10% resulting in a final set of 111 individuals and
539 17,729 SNPs.

540 Population structure analyses

541 We fit the Pritchard, Donnelly, and Stephens model (PSD) and ran principal components analysis
542 on the genotype matrix of North American gray wolves ([Price et al., 2006](#), [Pritchard et al., 2000](#)).
543 For the PSD model we used the ADMIXTURE software on the un-normalized genotypes, running
544 5 replicates per choice of K , from $K = 2$ to $K = 8$ ([Alexander et al., 2009](#)). For each K we
545 choose the one that achieved the highest likelihood to visualize. For PCA, we centered and scaled
546 the genotype matrix and then ran `sklearn` implementation of PCA, truncated to compute 50
547 eigenvectors.

548 Grid construction

549 To create a dense triangular lattice around the sample locations, we first define an outer boundary
550 polygon. As a default, we construct the lattice by creating a convex hull around the sample points
551 and manually trimming the polygon to adhere to the geography of the study organism and
552 balancing the sample point range with the extent of local geography using the following website
553 <https://www.keene.edu/campus/maps/tool/>. We often do not exclude internal "holes" in
554 the habitat (e.g. water features for terrestrial animals), and let the model instead fit effective
555 migration rates for those features to the extent they lead to elevated differentiation. We also
556 emphasize the importance of defining the lattice for FEEMS as well as EEMS and suggest this
557 should be carefully curated with prior biological knowledge about the system.

558 To ensure edges cover an equal area over the entire region we downloaded and intersected a
559 uniform grid defined on the spherical shape of earth ([Sahr et al., 2003](#)). These defined grids are
560 pre-computed at a number of different resolutions, allowing a user to test FEEMS at different
561 grid densities which is an important feature to explore.

562 Code Availability

563 The code to reproduce the results of this paper and more can be found in [https://github.com/](https://github.com/jhmarcus/feems-paper)
564 [jhmarcus/feems-paper](https://github.com/jhmarcus/feems-paper). A python package implementing the method can be found in [https://](https://github.com/jhmarcus/feems)
565 github.com/jhmarcus/feems with documentation found in <http://jhmarcus.com/feems/>.

566 Data Availability

567 We included a processed version of the dataset used in this manuscript in the `feems` package
568 found here: <https://github.com/jhmarcus/feems>. An example tutorial on how to access the
569 data the can be found here: <http://jhmarcus.com/feems/notebooks/getting-started.html>.

570 Acknowledgements

571 We thank Rena Schweizer for helping us download and process the gray wolf dataset used in the
572 paper, Ben Peter for providing feedback and code for helping to construct the discrete global grids
573 and preparing the human genetic dataset, and Hussein Al-Asadi, Peter Carbonetto, Dan Rice
574 for helpful conversations about the optimization and modeling approach. We also acknowledge
575 helpful feedback from Arjun Biddanda, Anna Di Rienzo, Matthew Stephens, the Stephens Lab,
576 the Novembre Lab, and the University of Chicago 4th floor computational biology group.

577 This study was supported in part by the National Science Foundation via fellowship DGE-
578 1746045 and the National Institute of General Medical Sciences via training grant T32GM007197
579 to J.H.M and R01GM132383 to J.N. W.H. was partially supported by the NSF via the TRIPODS
580 program and by the Berkeley Institute for Data Science. R.F.B. was supported by the National
581 Science Foundation via grant DMS-1654076, and by the Office of Naval Research via grant
582 N00014-20-1-2337.

583 Author Contributions

584 J.H.M and J.N. conceived of the project. J.H.M. and W.H. developed the statistical methodology
585 with guidance from R.F.B. (lead) and J.N. (supporting). J.H.M, W.H. carried out method testing
586 and application with guidance from J.N. (lead) and R.F.B. (supporting). J.H.M. and W.H.
587 developed the software. J.H.M. and W.H. wrote the paper with edits from R.F.B. and J.N.

588 Competing interests

589 The authors have no competing interests to declare.

590 References

- 591 Al-Asadi, H., Petkova, D., Stephens, M., and Novembre, J. (2019). Estimating recent migration
592 and population-size surfaces. *PLoS Genetics*, 15(1):e1007908.
- 593 Alexander, D. H., Novembre, J., and Lange, K. (2009). Fast model-based estimation of ancestry
594 in unrelated individuals. *Genome Research*, 19(9):1655–1664.
- 595 Battey, C., Ralph, P. L., and Kern, A. D. (2020). Space is the place: Effects of continuous spatial
596 structure on analysis of population genetic data. *Genetics*, 215(1):193–214.
- 597 Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. (2011). Distributed optimization
598 and statistical learning via the alternating direction method of multipliers. *Foundations and
599 Trends in Machine learning*, 3(1):1–122.
- 600 Bradburd, G. S., Coop, G. M., and Ralph, P. L. (2018). Inferring continuous and discrete
601 population genetic structure across space. *Genetics*, 210(1):33–52.

- 602 Bradburd, G. S. and Ralph, P. L. (2019). Spatial population genetics: It’s about time. *Annual*
603 *Review of Ecology, Evolution, and Systematics*, 50:427–449.
- 604 Bradburd, G. S., Ralph, P. L., and Coop, G. M. (2016). A spatial framework for understanding
605 population structure and admixture. *PLoS Genetics*, 12(1).
- 606 Byrd, R. H., Lu, P., Nocedal, J., and Zhu, C. (1995). A limited memory algorithm for bound
607 constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208.
- 608 Chandra, A. K., Raghavan, P., Ruzzo, W. L., Smolensky, R., and Tiwari, P. (1996). The elec-
609 trical resistance of a graph captures its commute and cover times. *Computational Complexity*,
610 6(4):312–340.
- 611 Dobzhansky, T. and Wright, S. (1943). Genetics of natural populations. x. dispersion rates in
612 drosophila pseudoobscura. *Genetics*, 28(4):304.
- 613 Dong, X., Thanou, D., Frossard, P., and Vandergheynst, P. (2016). Learning laplacian matrix in
614 smooth graph signal representations. *IEEE Transactions on Signal Processing*, 64(23):6160–
615 6173.
- 616 Dong, X., Thanou, D., Rabbat, M., and Frossard, P. (2019). Learning graphs from data: A
617 signal representation perspective. *IEEE Signal Processing Magazine*, 36(3):44–63.
- 618 Duforet-Frebourg, N. and Blum, M. G. (2014). Nonstationary patterns of isolation-by-distance:
619 inferring measures of local genetic differentiation with bayesian kriging. *Evolution*, 68(4):1110–
620 1123.
- 621 Egilmez, H. E., Pavez, E., and Ortega, A. (2016). Graph learning from data under structural
622 and laplacian constraints. *arXiv preprint arXiv:1611.05181*.
- 623 Felsenstein, J. (1982). How can we infer geography and history from gene frequencies? *Journal*
624 *of Theoretical Biology*, 96(1):9–20.
- 625 Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with
626 the graphical lasso. *Biostatistics*, 9(3):432–441.
- 627 Hanks, E. M. (2015). A constructive spatio-temporal approach to modeling spatial covariance.
628 *arXiv preprint arXiv:1506.03824*.
- 629 Hanks, E. M. and Hooten, M. B. (2013). Circuit theory and model-based inference for landscape
630 connectivity. *Journal of the American Statistical Association*, 108(501):22–33.
- 631 Kalofolias, V. (2016). How to learn a graph from smooth signals. In *Artificial Intelligence and*
632 *Statistics*, pages 920–929.
- 633 Kelleher, J., Etheridge, A. M., and McVean, G. (2016). Efficient coalescent simulation and
634 genealogical analysis for large sample sizes. *PLoS Computational Biology*, 12(5).
- 635 Kimura, M. (1953). Stepping stone model of population. *Annual Report of the National Institute*
636 *of Genetics Japan*, 3:62–63.
- 637 Kimura, M. and Weiss, G. H. (1964). The stepping stone model of population structure and the
638 decrease of genetic correlation with distance. *Genetics*, 49(4):561.
- 639 Koller, D. and Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*.
640 MIT Press.

- 641 Korneliussen, T. S., Albrechtsen, A., and Nielsen, R. (2014). Angsd: analysis of next generation
642 sequencing data. *BMC bioinformatics*, 15(1):356.
- 643 Lauritzen, S. L. (1996). *Graphical models*, volume 17. Clarendon Press.
- 644 Lundgren, E. and Ralph, P. L. (2019). Are populations like a circuit? comparing isolation by
645 resistance to a new coalescent-based method. *Molecular ecology resources*, 19(6):1388–1406.
- 646 Malécot, G. (1948). *Les mathématiques de l’hérédité*. masson et cie. *Paris, France*.
- 647 Mateos, G., Segarra, S., Marques, A. G., and Ribeiro, A. (2019). Connecting the dots: Identifying
648 network structure via graph signal processing. *IEEE Signal Processing Magazine*, 36(3):16–43.
- 649 Mathieson, I., Alpaslan-Roodenberg, S., Posth, C., Szécsényi-Nagy, A., Rohland, N., Mallick, S.,
650 Olalde, I., Broomandkoshbacht, N., Candilio, F., Cheronet, O., et al. (2018). The genomic
651 history of southeastern europe. *Nature*, 555(7695):197–203.
- 652 McCullagh, P. (2009). Marginal likelihood for distance matrices. *Statistica Sinica*, pages 631–649.
- 653 McRae, B. H. (2006). Isolation by resistance. *Evolution*, 60(8):1551–1561.
- 654 Meirmans, P. G. (2012). The trouble with isolation by distance. *Molecular ecology*, 21(12):2839–
655 2846.
- 656 Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer Science & Business Media.
- 657 Patterson, N., Price, A. L., and Reich, D. (2006). Population structure and eigenanalysis. *PLoS*
658 *Genetics*, 2(12).
- 659 Peter, B. M., Petkova, D., and Novembre, J. (2018). Genetic landscapes reveal how human
660 genetic diversity aligns with geography. *BioRxiv*, page 233486.
- 661 Petkova, D., Novembre, J., and Stephens, M. (2016). Visualizing spatial population structure
662 with estimated effective migration surfaces. *Nature Genetics*, 48(1):94.
- 663 Petkova, D. I. (2013). *Inferring effective migration from geographically indexed genetic data*. The
664 University of Chicago.
- 665 Pickrell, J. and Pritchard, J. (2012). Inference of population splits and mixtures from genome-
666 wide allele frequency data. *Nature Precedings*, pages 1–1.
- 667 Pickrell, J. K. and Reich, D. (2014). Toward a new history and geography of human genes
668 informed by ancient dna. *Trends in Genetics*, 30(9):377–389.
- 669 Price, A. L., Patterson, N. J., Plenge, R. M., Weinblatt, M. E., Shadick, N. A., and Reich, D.
670 (2006). Principal components analysis corrects for stratification in genome-wide association
671 studies. *Nature Genetics*, 38(8):904–909.
- 672 Pritchard, J. K., Stephens, M., and Donnelly, P. (2000). Inference of population structure using
673 multilocus genotype data. *Genetics*, 155(2):945–959.
- 674 Ringbauer, H., Kolesnikov, A., Field, D. L., and Barton, N. H. (2018). Estimating barriers to
675 gene flow from distorted isolation-by-distance patterns. *Genetics*, 208(3):1231–1245.
- 676 Rue, H. and Held, L. (2005). *Gaussian Markov random fields: theory and applications*. CRC
677 press.

- 678 Safner, T., Miller, M. P., McRae, B. H., Fortin, M.-J., and Manel, S. (2011). Comparison of
679 bayesian clustering and edge detection methods for inferring boundaries in landscape genetics.
680 *International Journal of Molecular Sciences*, 12(2):865–889.
- 681 Sahr, K., White, D., and Kimerling, A. J. (2003). Geodesic discrete global grid systems. *Car-*
682 *tography and Geographic Information Science*, 30(2):121–134.
- 683 Schweizer, R. M., Vonholdt, B. M., Harrigan, R., Knowles, J. C., Musiani, M., Coltman, D.,
684 Novembre, J., and Wayne, R. K. (2016). Genetic subdivision and candidate genes under
685 selection in north american grey wolves. *Molecular Ecology*, 25(1):380–402.
- 686 Slatkin, M. (1985). Gene flow in natural populations. *Annual review of ecology and systematics*,
687 16(1):393–430.
- 688 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,
689 Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wil-
690 son, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson,
691 E., Carey, C., Polat, İ., Feng, Y., Moore, E. W., Vand erPlas, J., Laxalde, D., Perktold, J.,
692 Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H.,
693 Pedregosa, F., van Mulbregt, P., and Contributors, S. . . (2020). SciPy 1.0: Fundamental
694 Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.
- 695 Vishnoi, N. K. et al. (2013). $L_x = b$. *Foundations and Trends in Theoretical Computer Science*,
696 8(1–2):1–141.
- 697 Wang, Y.-X., Sharpnack, J., Smola, A. J., and Tibshirani, R. J. (2016). Trend filtering on graphs.
698 *The Journal of Machine Learning Research*, 17(1):3651–3691.
- 699 Wilson, A., Kasy, M., and Mackey, L. (2020). Approximate cross-validation: Guarantees for
700 model assessment and selection. *arXiv preprint arXiv:2003.00617*.
- 701 Wright, S. (1943). Isolation by Distance. *Genetics*, 28(2):114.
- 702 Wright, S. (1946). Isolation by distance under diverse systems of mating. *Genetics*, 31(1):39.
- 703 Wu, S., Joseph, A., Hammonds, A. S., Celniker, S. E., Yu, B., and Frise, E. (2016). Stability-
704 driven nonnegative matrix factorization to interpret spatial gene expression and build local
705 gene networks. *Proceedings of the National Academy of Sciences*, 113(16):4290–4295.
- 706 Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal*
707 *of the royal statistical society: series B (statistical methodology)*, 67(2):301–320.

Supplementary Materials

Mathematical notation

We denote matrices using bold capital letters \mathbf{A} . Bold lowercase letters are vectors \mathbf{a} , and non-bold lowercase letters are scalars a . We denote by \mathbf{A}^{-1} and \mathbf{A}^\dagger the inverse and (Moore-Penrose) pseudo-inverse of \mathbf{A} respectively. We use $\mathbf{y} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to express that the random vector \mathbf{y} is modeled as a p -dimensional multivariate Gaussian distribution with fixed parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ and use the conditional notation $\mathbf{y}|\boldsymbol{\mu} \sim N_p(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ if $\boldsymbol{\mu}$ is random.

A graph is a pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where \mathcal{V} denotes a set of nodes or vertices and $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ denotes a set of edges. Throughout we assume the graph \mathcal{G} is undirected, weighted, and contains no self loops, i.e. $(i, j) \in \mathcal{E} \iff (j, i) \in \mathcal{E}$ and $(i, i) \notin \mathcal{E}$ and each edge $(i, j) \in \mathcal{E}$ is given a weight $w_{ij} = w_{ji} > 0$. We write \mathbf{W} to indicate the symmetric weighted adjacency matrix, i.e.

$$\mathbf{W}_{ij} = \begin{cases} w_{ij}, & \text{if } (i, j) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases}$$

$\mathbf{w} \in R^m$ is a vectorized form of the non-zero lower-triangular entries of \mathbf{W} where $m = |\mathcal{E}|/2$ is the number of non-zero lower triangular elements. We denote by $\mathbf{L} = \text{diag}(\mathbf{W}\mathbf{1}) - \mathbf{W}$ the graph Laplacian.

Gradient computation

In practice, we make a change of variable from $\mathbf{w} \in R_+^m$ to $\mathbf{z} = \log(\mathbf{w}) \in R^m$ and the algorithm is applied to the transformed objective function:

$$\ell(\exp(\mathbf{z}), \sigma^2; \mathbf{C}\hat{\boldsymbol{\Sigma}}\mathbf{C}^\top) + \phi_{\lambda, \alpha}(\exp(\mathbf{z})) = \tilde{\ell}(\mathbf{z}, \sigma^2; \mathbf{C}\hat{\boldsymbol{\Sigma}}\mathbf{C}^\top) + \tilde{\phi}_{\lambda, \alpha}(\mathbf{z}).$$

After the change of variable, the objective value remains the same whereas it follows from the chain rule that $\nabla(\tilde{\ell}(\mathbf{z}) + \tilde{\phi}_{\lambda, \alpha}(\mathbf{z})) = \nabla(\ell(\mathbf{w}) + \phi_{\lambda, \alpha}(\mathbf{w})) \odot \mathbf{w}$ where \odot indicates the Hadamard product or elementwise product—for notational convenience, we drop the dependency of ℓ on the quantities σ^2 and $\mathbf{C}\hat{\boldsymbol{\Sigma}}\mathbf{C}^\top$. Furthermore, the computation of $\nabla\phi_{\lambda, \alpha}(\mathbf{w})$ is relatively straightforward, so in the rest of this section, we discuss only the computation of the gradient of the negative log-likelihood function with respect to \mathbf{w} , i.e. $\nabla\ell(\mathbf{w})$.

Recall, by definition, the graph Laplacian \mathbf{L} implicitly depends on the variable \mathbf{w} through $\mathbf{L} = \text{diag}(\mathbf{W}\mathbf{1}) - \mathbf{W}$. Throughout we assume the first o rows and columns of \mathbf{L} correspond to the observed nodes. With this assumption, our node assignment matrix has block structure $\mathbf{A} = [\mathbf{I}_{o \times o} \mid \mathbf{0}_{o \times (d-o)}]$. To simplify some of the equations appearing later, we introduce the notation: we define

$$\mathbf{L}_{\text{full}} := \mathbf{L} + \frac{\mathbf{1}\mathbf{1}^\top}{d}, \quad \boldsymbol{\Sigma} := \mathbf{A}\mathbf{L}_{\text{full}}^{-1}\mathbf{A}^\top + \sigma^2\text{diag}(\mathbf{n}^{-1}), \quad (10)$$

and

$$\mathbf{M} := \mathbf{C}^\top \left((\mathbf{C}\boldsymbol{\Sigma}\mathbf{C})^{-1}(\mathbf{C}\hat{\boldsymbol{\Sigma}}\mathbf{C}) - (\mathbf{C}\boldsymbol{\Sigma}\mathbf{C})^{-1} \right) \mathbf{C}.$$

Applying the chain rule and matrix derivatives, we can calculate:

$$\nabla\ell(\mathbf{w}) = \frac{\partial\ell(\mathbf{w})}{\partial\text{vec}(\mathbf{L})} \cdot \frac{\partial\text{vec}(\mathbf{L})}{\partial\mathbf{w}^\top},$$

where vec is the vectorization operator and $\partial\ell/\partial\text{vec}(\mathbf{L})$ and $\partial\text{vec}(\mathbf{L})/\partial\mathbf{w}^\top$ are $1 \times d^2$ vector and $d^2 \times d$ matrix, respectively, given by

$$\frac{\partial\ell(\mathbf{w})}{\partial\text{vec}(\mathbf{L})} = p \cdot \text{vec} \left(\mathbf{L}_{\text{full}}^{-1} \mathbf{A}^\top \mathbf{M} \mathbf{A} \mathbf{L}_{\text{full}}^{-1, \top} \right), \quad \frac{\partial\text{vec}(\mathbf{L})}{\partial\mathbf{w}^\top} = \mathbf{S} - \mathbf{T}. \quad (11)$$

737 Here \mathbf{S} and \mathbf{T} are linear operators that satisfy $\mathbf{S}\mathbf{w} = \text{diag}(\mathbf{W}\mathbf{1})$ and $\mathbf{T}\mathbf{w} = \mathbf{W}$. Note \mathbf{S} and
 738 \mathbf{T} both have $\mathcal{O}(d)$ many nonzero entries, so we can perform sparse matrix multiplication to
 739 efficiently compute the matrix-vector multiplication $\partial\ell/\partial\text{vec}(\mathbf{L}) \cdot (\mathbf{S} - \mathbf{T})$. On the other hand,
 740 the computation of $\partial\ell/\partial\text{vec}(\mathbf{L})$ is more challenging as it requires inverting the full $d \times d$ matrix
 741 \mathbf{L}_{full} . Next we develop a procedure that efficiently computes $\partial\ell/\partial\text{vec}(\mathbf{L})$. We proceed by dividing
 742 the task into multiple steps.

743 **1. Computing Σ^{-1}** Recalling the block structure $\mathbf{A} = [\mathbf{I}_{o \times o} \mid \mathbf{0}_{o \times (d-o)}]$ of the node assign-
 744 ment matrix, we can write Σ as:

$$\Sigma = (\mathbf{L}_{\text{full}}^{-1})_{o \times o} + \sigma^2 \text{diag}(\mathbf{n}^{-1}),$$

745 where $(\mathbf{L}_{\text{full}}^{-1})_{o \times o}$ denotes the $o \times o$ upper-left block of $\mathbf{L}_{\text{full}}^{-1}$. Following [Petkova et al. \(2016\)](#), the
 746 inverse Σ^{-1} has the form

$$\Sigma^{-1} = \mathbf{X} + \sigma^{-2} \text{diag}(\mathbf{n}), \quad (12)$$

for some matrix $\mathbf{X} \in R^{o \times o}$. Equating $\Sigma\Sigma^{-1} = \mathbf{I}$, it follows that

$$\begin{aligned} & \left[(\mathbf{L}_{\text{full}}^{-1})_{o \times o} + \sigma^2 \text{diag}(\mathbf{n}^{-1}) \right] (\mathbf{X} + \sigma^{-2} \text{diag}(\mathbf{n})) = \mathbf{I} \\ \iff & \left[(\mathbf{L}_{\text{full}}^{-1})_{o \times o} + \sigma^2 \text{diag}(\mathbf{n}^{-1}) \right] \mathbf{X} = -\sigma^{-2} (\mathbf{L}_{\text{full}}^{-1})_{o \times o} \text{diag}(\mathbf{n}). \end{aligned} \quad (13)$$

747 Therefore, Σ^{-1} can be obtained by solving the $o \times o$ linear system (13) and plugging the solution
 748 into (12). The challenge here is to compute $(\mathbf{L}_{\text{full}}^{-1})_{o \times o}$ without matrix inversion of the full-
 749 dimensional \mathbf{L}_{full} .

2. Computing $(\mathbf{L}_{\text{full}}^{-1})_{o \times o}$ Let $\mathbf{L}_{\text{full}, o \times o}$ be the $o \times o$ block matrix corresponding to the ob-
 served nodes of \mathbf{L}_{full} , and similarly let $\mathbf{L}_{\text{full}, (d-o) \times (d-o)}$ and $\mathbf{L}_{\text{full}, o \times (d-o)} = \mathbf{L}_{\text{full}, (d-o) \times o}^\top$ be the
 corresponding block matrices of \mathbf{L}_{full} respectively. The inverse of $(\mathbf{L}_{\text{full}}^{-1})_{o \times o}$ is then given by the
 Schur complement of $\mathbf{L}_{\text{full}, (d-o) \times (d-o)}$ in \mathbf{L} :

$$\left[(\mathbf{L}_{\text{full}}^{-1})_{o \times o} \right]^{-1} = \mathbf{L}_{\text{full}, o \times o} - \mathbf{L}_{\text{full}, o \times (d-o)} (\mathbf{L}_{\text{full}, (d-o) \times (d-o)})^{-1} \mathbf{L}_{\text{full}, (d-o) \times o}. \quad (14)$$

See also [Hanks and Hooten \(2013\)](#), [Petkova et al. \(2016\)](#). Since every term in (14) has sparse
 + rank-1 structure, the matrix multiplications can be performed fast. In addition, for the term
 $(\mathbf{L}_{\text{full}, (d-o) \times (d-o)})^{-1}$, we can use the Sherman-Morrison formula so that the inverse is given
 explicitly by

$$\begin{aligned} (\mathbf{L}_{\text{full}, (d-o) \times (d-o)})^{-1} &= \left(\mathbf{L}_{(d-o) \times (d-o)} + \frac{\mathbf{1}\mathbf{1}^\top}{d} \right)^{-1} \\ &= \mathbf{L}_{(d-o) \times (d-o)}^{-1} - \frac{1}{d + \mathbf{1}^\top \mathbf{L}_{(d-o) \times (d-o)}^{-1} \mathbf{1}} \mathbf{L}_{(d-o) \times (d-o)}^{-1} \mathbf{1}\mathbf{1}^\top \mathbf{L}_{(d-o) \times (d-o)}^{-1}. \end{aligned}$$

750 Hence, in order to compute $(\mathbf{L}_{\text{full},(d-o)\times(d-o)})^{-1} \mathbf{L}_{\text{full},(d-o)\times o}$, we need to solve two systems of
751 linear equations:

$$\mathbf{L}_{(d-o)\times(d-o)} \mathbf{U} = \mathbf{L}_{\text{full},(d-o)\times o} \text{ and } \mathbf{L}_{(d-o)\times(d-o)} \mathbf{u} = \mathbf{1}.$$

752 Note that the matrix $\mathbf{L}_{(d-o)\times(d-o)}$ is sparse, so both systems can be solved efficiently by per-
753 forming sparse Cholesky factorization on $\mathbf{L}_{(d-o)\times(d-o)}$ (Hanks and Hooten, 2013). Alternatively,
754 one can implement fast Laplacian solvers (Vishnoi et al., 2013) that solve the Laplacian system
755 in time nearly linear in the dimension $\mathcal{O}(d)$. After we obtain $\left[(\mathbf{L}_{\text{full}}^{-1})_{o\times o} \right]^{-1}$ via sparse + rank-1
756 matrix multiplication and sparse Cholesky factorization, we can invert the $o \times o$ matrix to get
757 $(\mathbf{L}_{\text{full}}^{-1})_{o\times o}$.

758 **3. Computing $(\mathbf{L}_{\text{full}}^{-1})_{d\times o}$** Write

$$(\mathbf{L}_{\text{full}}^{-1})_{d\times o} = \begin{bmatrix} (\mathbf{L}_{\text{full}}^{-1})_{o\times o} \\ (\mathbf{L}_{\text{full}}^{-1})_{(d-o)\times o} \end{bmatrix}.$$

759 Using the inversion of the matrix in a block form, the $(d - o) \times o$ block component is given by

$$(\mathbf{L}_{\text{full}}^{-1})_{(d-o)\times o} = - \underbrace{(\mathbf{L}_{\text{full},(d-o)\times(d-o)})^{-1} \mathbf{L}_{\text{full},(d-o)\times o}}_{(A)} \underbrace{(\mathbf{L}_{\text{full}}^{-1})_{o\times o}}_{(B)}. \quad (15)$$

760 Since each of the two terms (A) and (B) has been already computed in the previous step, there
761 is no need to recompute them. In total, it requires a $(d - o) \times o$ matrix and $o \times o$ matrix
762 multiplication.

763 **4. Computing the full gradient** Going back to the expression of $\nabla \ell(\mathbf{w})$ in (11), and noting
764 the block structure of the assignment matrix \mathbf{A} , we have:

$$\frac{\partial \ell(\mathbf{w})}{\partial \text{vec}(\mathbf{L})} = p \cdot \text{vec} \left((\mathbf{L}_{\text{full}}^{-1})_{d\times o} \mathbf{M} (\mathbf{L}_{\text{full}}^{-1})_{d\times o}^{\top} \right).$$

Let $\Pi_1 = \mathbf{1} (\mathbf{1}^{\top} \Sigma^{-1} \mathbf{1})^{-1} \mathbf{1}^{\top} \Sigma^{-1}$ be projection to the space of constant vectors with respect to the
inner product $\langle \mathbf{x}, \mathbf{y} \rangle = \mathbf{x}^{\top} \Sigma^{-1} \mathbf{y}$. Using the identity $\mathbf{I} - \Pi_1 = \Sigma \mathbf{C}^{\top} (\mathbf{C} \Sigma \mathbf{C}^{\top})^{-1} \mathbf{C}$ (McCullagh,
2009), then we can write \mathbf{M} in terms of Π_1 :

$$\mathbf{M} = \Sigma^{-1} (\mathbf{I} - \Pi_1) \widehat{\Sigma} \Sigma^{-1} (\mathbf{I} - \Pi_1) - \Sigma^{-1} (\mathbf{I} - \Pi_1). \quad (16)$$

765 Since Π_1 is a rank-1 matrix, this expression of \mathbf{M} allows easier computation. Finally we can
766 put together (12), (13), (15), and (16), to compute the gradient of the negative log-likelihood
767 function with respect to the graph Laplacian.

768 Objective computation

769 The graph Laplacian \mathbf{L} is orthogonal to the one vector $\mathbf{1}$, so using the notation introduced in (10),
770 we can express our objective function as

$$\ell(\mathbf{w}) + \phi_{\lambda, \alpha}(\mathbf{w}) = p \cdot \text{tr} \left((\mathbf{C} \Sigma \mathbf{C}^{\top})^{-1} \mathbf{C} \widehat{\Sigma} \mathbf{C}^{\top} \right) - p \cdot \log \det (\mathbf{C} \Sigma \mathbf{C})^{-1} + \frac{\lambda}{2} \|\Delta(\mathbf{w} + \alpha \log(\mathbf{w}))\|_2^2.$$

With the identity $\mathbf{I} - \Pi_1 = \Sigma \mathbf{C}^\top (\mathbf{C} \Sigma \mathbf{C}^\top)^{-1} \mathbf{C}$, the trace term is:

$$\text{tr} \left((\mathbf{C} \Sigma \mathbf{C}^\top)^{-1} \mathbf{C} \hat{\Sigma} \mathbf{C}^\top \right) = \text{tr} \left(\mathbf{C}^\top (\mathbf{C} \Sigma \mathbf{C}^\top)^{-1} \mathbf{C} \hat{\Sigma} \right) = \text{tr} \left(\Sigma^{-1} (\mathbf{I} - \Pi_1) \hat{\Sigma} \right).$$

771 The matrix inside the trace has been constructed in the gradient computation, see equation (16).
 772 In terms of the determinant, we use the same approach considered in Petkova et al. (2016)—in
 773 particular, concatenating \mathbf{C}^\top and $\mathbf{1}$, the matrix $[\mathbf{C}^\top \mid \mathbf{1}]$ is orthogonal, so it can be shown that

$$\det(\Sigma) = \frac{\det(\mathbf{1}^\top \mathbf{1}) \det(\mathbf{C} \Sigma \mathbf{C}^\top)}{\det(\mathbf{C} \mathbf{C}^\top) \det(\mathbf{1}^\top \Sigma^{-1} \mathbf{1})}.$$

774 Rearranging terms and using the fact $\det(\mathbf{U}^{-1}) = \det(\mathbf{U})^{-1}$ for any matrix \mathbf{U} , we obtain:

$$\det(\mathbf{C} \Sigma \mathbf{C}^\top)^{-1} = \frac{\det(\mathbf{1}^\top \mathbf{1}) \det(\Sigma^{-1})}{\det(\mathbf{C} \mathbf{C}^\top) \det(\mathbf{1}^\top \Sigma^{-1} \mathbf{1})} = \frac{o}{\mathbf{1}^\top \Sigma^{-1} \mathbf{1}} \det(\Sigma^{-1}).$$

775 We have computed Σ^{-1} in equation (12), so each of the terms above can be computed without any
 776 additional matrix multiplications. Finally, the signed graph incidence matrix Δ defined on the
 777 edges of the graph is, by construction, highly sparse with $\mathcal{O}(d)$ many nonzero entries. Hence we
 778 implement sparse matrix multiplication to evaluate the penalty function $\phi_{\lambda, \alpha}(\mathbf{w})$ while avoiding
 779 the full-dimensional matrix-vector product.

780 Estimating the edge weights under the exact likelihood model

781 Recall that, when describing our data model, we employed the approximation $\frac{1}{2} f_j(k)(1 - f_j(k)) \approx$
 782 $\sigma^2 \mu_j(1 - \mu_j)$ for all SNPs j and nodes k (see equation (4)) and estimated the residual variance σ^2
 783 under the homogeneous isolation by distance model. Here we examine whether this approxima-
 784 tion results in a significant difference with respect to the estimation quality of the edge weights
 785 of the graph.

786 Without approximation, we can calculate the exact analytical form for the marginal likelihood
 787 of the estimated frequency as follows (after removing the SNP means):

$$\mathbf{C} \hat{\mathbf{f}}_j \sim \sqrt{\mu_j(1 - \mu_j)} \cdot \mathcal{N}_{o-1} \left(\mathbf{0}, \mathbf{C} \mathbf{A} \mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \mathbf{C} \text{diag}(\mathbf{n}^{-1}) \mathbf{A} \text{diag} \left(\left\{ \frac{1 - L_{kk}^\dagger}{2} \right\}_{k=1}^d \right) \mathbf{A}^\top \mathbf{C}^\top \right), \quad (17)$$

788 where $\{a_k\}_{k=1}^d$ represents the vector $\mathbf{a} = (a_1, \dots, a_d)$. We then consider estimating the edge
 789 weights with the likelihood based on (17) and without relying on approximating the residual
 790 variance. In particular, comparing to the model (5), this formulation does not introduce the
 791 unknown residual variance parameter σ^2 but rather it is given implicitly by $(1 - L_{kk}^\dagger)/2$. This
 792 means that the model (17) is well-defined only when $L_{kk}^\dagger \leq 1$ for all nodes k , hence leading to
 793 the following constrained optimization problem:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{l} \leq \mathbf{w} \leq \mathbf{u}} \left\{ \ell_{\text{exact}}(\mathbf{w}; \mathbf{C} \hat{\Sigma} \mathbf{C}^\top) + \phi_{\lambda, \alpha}(\mathbf{w}) : L_{kk}^\dagger \leq 1 \text{ for all } k \in \mathcal{V} \right\}, \quad (18)$$

where ℓ_{exact} is the negative log-likelihood function implied by the model (17) and $\phi_{\lambda, \alpha}$ is our
 smooth penalty function. The main difficulty of solving (18) is that enforcing the constraint
 $L_{kk}^\dagger \leq 1$ for all nodes $k \in \mathcal{V}$, requires full computation of the pseudo-inverse of a $d \times d$ matrix
 \mathbf{L} whereas in order to evaluate the likelihood, we only need to calculate \mathbf{L}^\dagger on the observed

nodes. To overcome this computational challenge, we may relax the constraint and consider the following form as a proxy for optimization (18):

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{l} \leq \mathbf{w} \leq \mathbf{u}} \left\{ \ell_{\text{exact}}(\mathbf{w}; \mathbf{C} \hat{\Sigma} \mathbf{C}^\top) + \phi_{\lambda, \alpha}(\mathbf{w}) : L_{kk}^\dagger \leq 1 \text{ for all observed nodes } k \right\}. \quad (19)$$

794 We can solve this problem efficiently using a gradient-based algorithm where the gradient of
795 ℓ_{exact} with respect to \mathbf{L} is given by

$$\frac{\partial \ell_{\text{exact}}(\mathbf{w})}{\partial \text{vec}(\mathbf{L})} = p \cdot \text{vec} \left(\mathbf{L}_{\text{full}}^{-1} \mathbf{A}^\top \mathbf{M} \mathbf{A} \mathbf{L}_{\text{full}}^{-1, \top} \right) - p \cdot \text{diag}(\mathbf{M})^\top \text{diag}((2\mathbf{n})^{-1}) \mathbf{N},$$

796 where \mathbf{M} is a $o \times o$ matrix defined in (16), while \mathbf{N} is a $o \times d^2$ matrix whose rows correspond to
797 the observed subsets of the rows of the $d^2 \times d^2$ matrix $\mathbf{L}_{\text{full}}^{-1} \otimes \mathbf{L}_{\text{full}}^{-1}$.

798 Overall, when we implement the penalized restricted maximum likelihood procedure in (19),
799 we find that it does not make much of a difference and output qualitatively comparable results to
800 FEEMS—for example, Supp. Fig. 12 shows one such fit with a setting of $\lambda = 10^{-3}$ and $\alpha = 50$.
801 Unfortunately, this approach has a drawback that after the algorithm reaches the solution, the
802 term $1 - L_{kk}^\dagger$ is not guaranteed to be positive for the unobserved nodes, since, due to the
803 computational efficiency, the constraints $L_{kk}^\dagger \leq 1$ are only placed on the observed nodes. This,
804 in principle, results in an ill-defined model if we would like interpretable results at unobserved as
805 well as observed nodes, and therefore we replace the calculation (17) with the approximation (5)
806 to avoid this issue. In addition, by decoupling the residual variance parameter σ^2 from the
807 graph-related weighted edges \mathbf{w} , the model (6) has more resemblance to spatial coalescent model
808 used in EEMS (Petkova et al., 2016).

809 Jointly estimating the residual variance and edge weights

810 One simple strategy we have used throughout the paper was to fit σ^2 first under a model of
811 homogeneous isolation by distance and prefix the estimated residual variance to the resulting $\hat{\sigma}^2$
812 for later fits of the effective migration rates. Alternatively, one might come up with a strategy
813 to estimate the unknown residual variance jointly with the edge weights, instead of prefixing it
814 from the estimation of the null model—the hope here is to simultaneously correct the model
815 misspecification and allow for improving model fit to the data.

816 As it turns out, given such a small fraction of sampled spatial locations in the data, the
817 strategy of jointly optimizing the marginal likelihood with respect to both variables has the
818 tendency to overfit to the data unless it is properly regularized. Specifically, we can consider the
819 model that generalizes (6), namely

$$p \cdot \mathbf{C} \hat{\Sigma} \mathbf{C}^\top \sim \mathcal{W}_{o-1} \left(\mathbf{C} \mathbf{A} \mathbf{L}^\dagger \mathbf{A}^\top \mathbf{C}^\top + \mathbf{C} \text{diag}(\mathbf{n}^{-1}) \mathbf{A} \text{diag}(\boldsymbol{\sigma}^2) \mathbf{A}^\top \mathbf{C}^\top, p \right),$$

820 where $\boldsymbol{\sigma}^2$ is a $d \times 1$ vector of node specific residual variances, i.e. each deme has its own residual
821 parameter σ_k for all nodes k . If the node specific parameters σ_k 's are assumed to be same across
822 all nodes, this reduces to the model (6). Supp. Fig. 13 shows the results of different strategies of
823 estimating the residual variances. As expected, when the model has a single residual variance σ^2 ,
824 either prefixing it from the null model (Figure 4) or estimating it jointly with the edge weights
825 (Supp. Fig. 13A) lead to similar and comparable outputs. The major difference is the high
826 migration edge forming long path appearing in Supp. Fig. 13A to separate the reduced gene-
827 flows in the middle, which tends to disappear as α increases. Whereas, if the residual variances

828 are allowed to be node specific, the fitted σ_k^2 's are highly variable and as a result the estimated
829 graph misses some geographic features present in the data, such as reduced effective migration
830 around St. Lawrence Island (Supp. Fig. 13B). Presumably this is attributed to overfitting,
831 due to the absence of data in many unobserved demes. In EEMS, in order to estimate the
832 genetic diversity parameters for every spatial position, which play a similar role as the residual
833 variance in FEEMS, a Voronoi-tessellation prior is placed to encourage sharing of information
834 across adjacent nodes and prevent over-fitting. While we can similarly estimate the node specific
835 residual variances on every node of the graph with our penalty function ($\phi_{\lambda,\alpha}$ defined on the
836 variable σ^2), we do not find it substantially improves the extent to which the model suits the
837 data. Thus, we take the approach of fitting the single residual variance σ^2 under the null model
838 and prefixing it as a simple but effective strategy with apparent good empirical performance.

839 Edge versus node parameterization

840 One of the novel features of FEEMS is its ability to directly find the edge weights of the graph that
841 best suit the data. This direct edge parameterization may increase the risk of model's overfitting,
842 but also allows for more flexible estimation of migration histories. Furthermore, as seen in
843 Figure 2 and Supp. Fig. 2, it has potential to recover anisotropic migration processes. This is in
844 contrast to EEMS wherein every spatial node is assigned an effective migration parameter m_k and
845 a migration rate on each edge joining nodes k and k' is given by the average effective migration
846 $w_{kk'} = (m_k + m_{k'})/2$. Not surprisingly, parameterization via node-specific parameters induces
847 implicit regularization by substantially constraining the feasible set of graph's edge weights. In
848 some cases, this has the desirable property of imposing an additional degree of similarity among
849 edge weights, but it often restricts the model's capacity to capture a richer set of structure
850 present in the data, (e.g. Petkova et al., 2016, supplementary figure 2). To be concrete, Supp.
851 Fig. 15 displays two different fits of FEEMS based on edge parameterization (Supp. Fig. 15A)
852 and node parameterization (Supp. Fig. 15B), run on a previously published dataset of human
853 genetic variation from Africa (see Peter et al. (2018) for details on the description of the dataset).
854 Running FEEMS with a node-based parameterization is straightforward in our framework—all
855 we have to do is to reparameterize the edge weights by the average effective migration and
856 solve the corresponding optimization problem (9) with respect to \mathbf{m} . It is evident from the
857 results that FEEMS with edge parameterization exhibits subtle correlations that exist between
858 the annotated demes in the figure whereas node parameterization fails to recover them. We also
859 compare the model fit of FEEMS to the observed genetic covariance (Supp. Fig. 16) and find
860 that edge-based parameterization provides a better fit to the African dataset. Supp. Fig. 17
861 further demonstrates that in the coalescent simulations with anisotropic migration, the node
862 parameterization is unable to recover the ground truth of the underlying migration rates even
863 when the nodes are fully observed.

864 Smooth penalty with ℓ_1 norm

865 FEEMS's primary optimization objective (see equation (9)) is:

$$\underset{\mathbf{l} \leq \mathbf{w} \leq \mathbf{u}}{\text{Minimize}} \ell(\mathbf{w}, \sigma^2; \mathbf{C}\hat{\Sigma}\mathbf{C}^\top) + \phi_{\lambda,\alpha}(\mathbf{w}),$$

866 where the spatial smoothness penalty is given by $\phi_{\lambda,\alpha}(\mathbf{w}) = \frac{\lambda}{2} \|\Delta(\mathbf{w} + \alpha \log(\mathbf{w}))\|_2^2$. It is widely
867 known that ℓ_1 -based method leads to better local adaptive fitting and structural recovery than
868 ℓ_2 -based methods (Wang et al., 2016), but at the cost of handling non-smooth objective functions
869 that are often computationally more challenging and demanding. In a spatial genetic dataset,

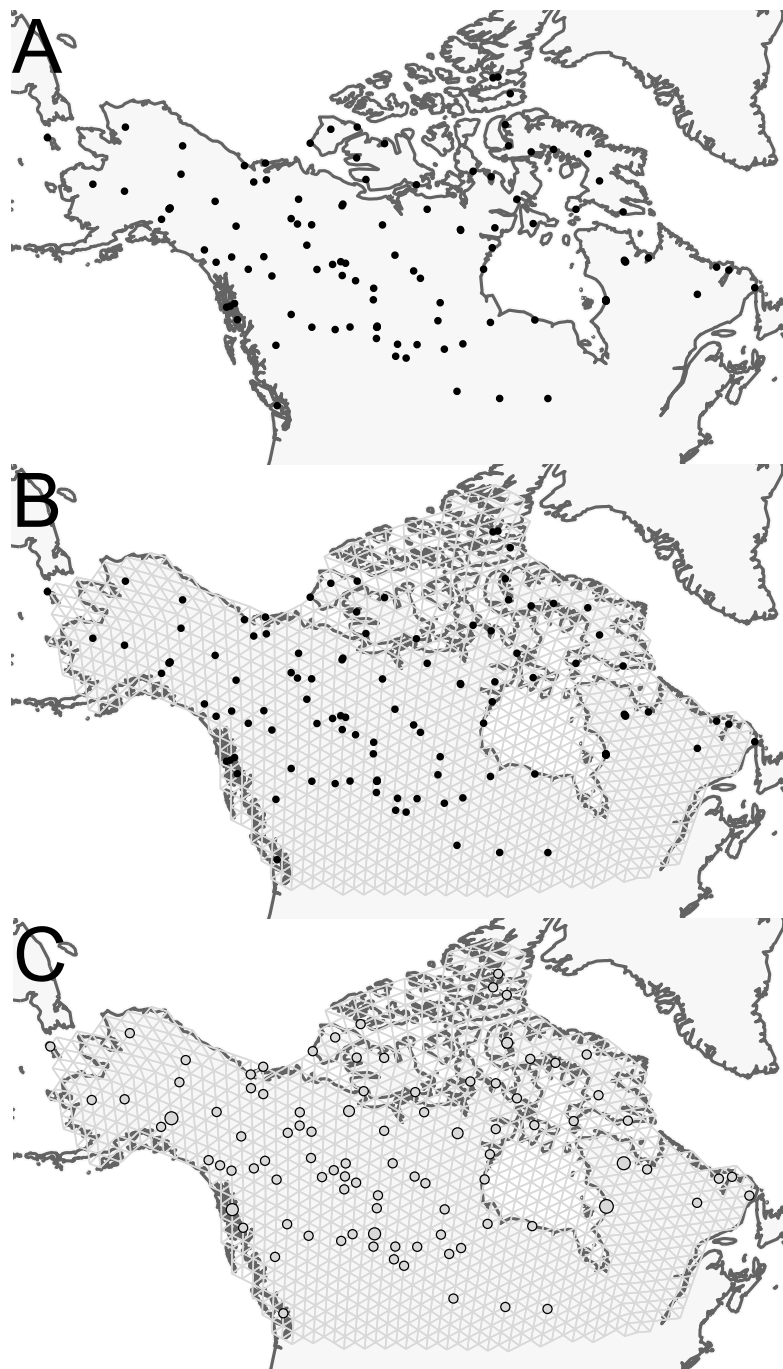
870 one major challenge is to deal with the relatively sparse sampling design where there are many
871 unobserved nodes on the graph. In this challenging statistical setting, our finding is that an
872 ℓ_2 -based method enables more accurate and reliable estimation of the geographic features.

873 Specifically, writing $\phi_{\lambda,\alpha}^{\ell_1}(\mathbf{w}) = \lambda\|\mathbf{\Delta}(\mathbf{w} + \alpha \log(\mathbf{w}))\|_1$, we considered the alternate following
874 composite objective function:

$$\ell(\mathbf{w}, \sigma^2; \mathbf{C}\widehat{\Sigma}\mathbf{C}^\top) + \phi_{\lambda,\alpha}^{\ell_1}(\mathbf{w}). \quad (20)$$

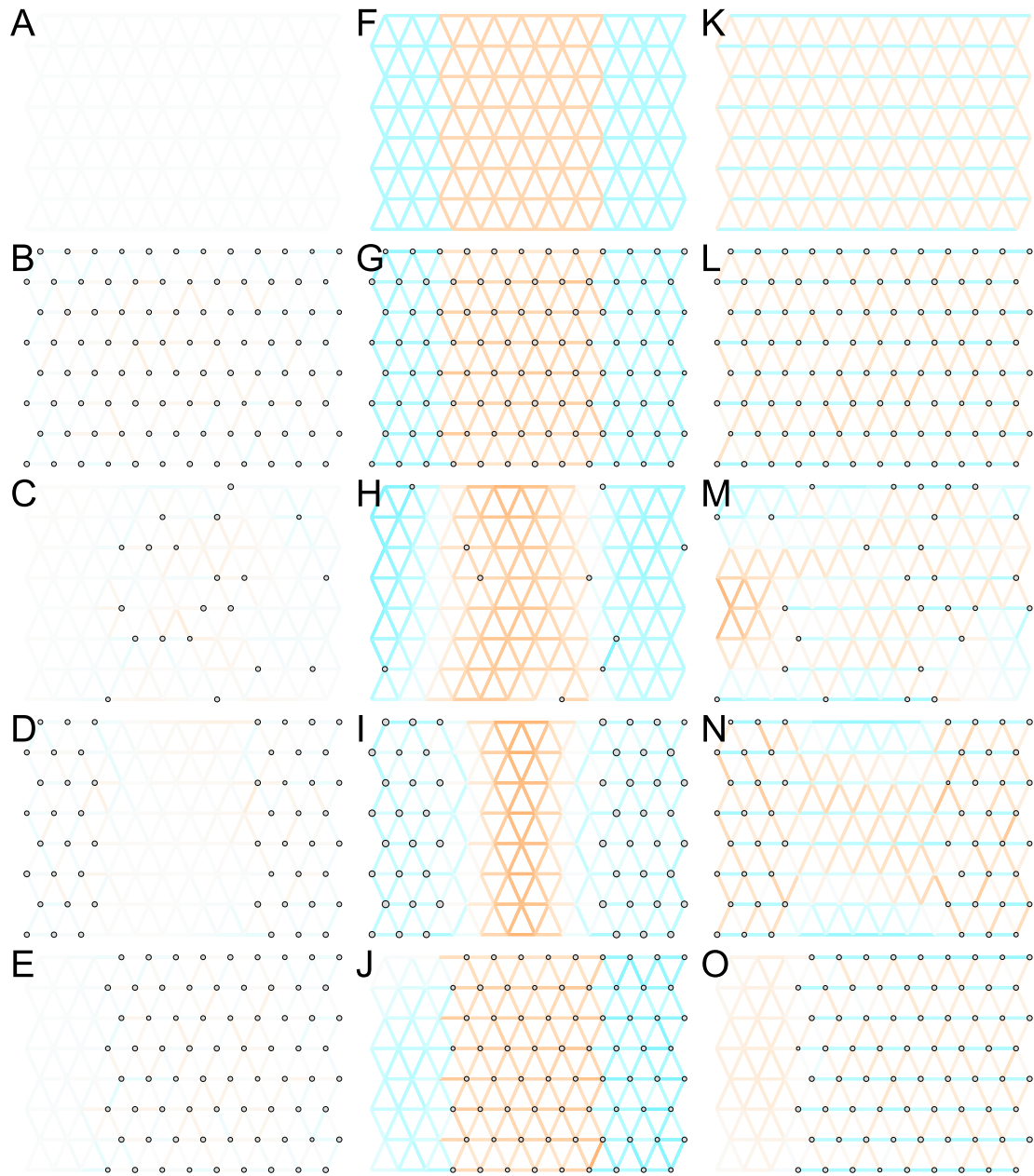
875 To solve (20), we apply linearized alternating direction method of multipliers (ADMM) (Boyd
876 et al., 2011), a variant of the standard ADMM algorithm, that iteratively optimizes the aug-
877 mented Lagrangian over the primal and dual variables. The derivation of the algorithm is a
878 standard calculation so we omit the detailed description of the algorithm. As opposed to the
879 common belief about the effectiveness of the ℓ_1 norm for structural recovery, the recovered graph
880 of FEEMS using ℓ_1 -based smooth penalty shows less accurate reconstruction of the migration
881 patterns, particularly when the sampling design has many locations with missing data on the
882 graph (Supp. Fig. 18A, Supp. Fig. 19H). It appears that the ℓ_1 -based penalty function is not
883 capable of accurately estimating edge weights at regions with little data, partially due to its local
884 adaptation, in contrast to the ℓ_2 -based method that considers regularization more globally. This
885 suggests that in order to use the ℓ_1 penalty $\phi_{\lambda,\alpha}^{\ell_1}(\mathbf{w})$ in the presence of many missing nodes, one
886 needs an additional regularization term that promotes global smoothness of the graph's edge
887 weights, e.g., a combination of $\phi_{\lambda,\alpha}^{\ell_1}(\mathbf{w})$ and $\phi_{\lambda,\alpha}(\mathbf{w})$ (same spirit as elastic net (Zou and Hastie,
888 2005)), or $\phi_{\lambda,\alpha}^{\ell_1}(\mathbf{w})$ on top of node-based parameterization (Supp. Fig. 18B).

889 Supplementary Figures

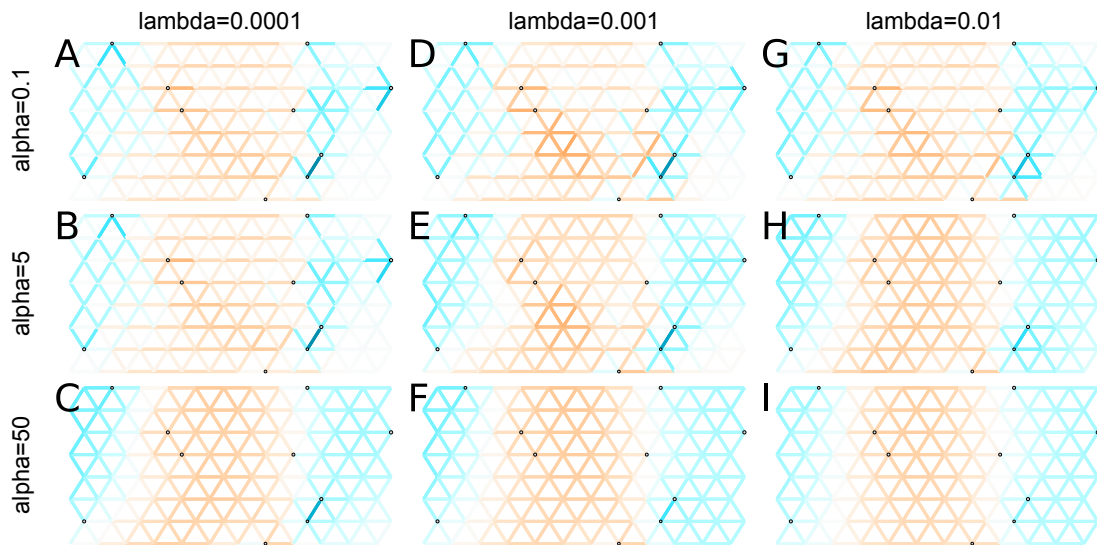


Supplementary Figure 1: Visualization of grid construction and node assignment:

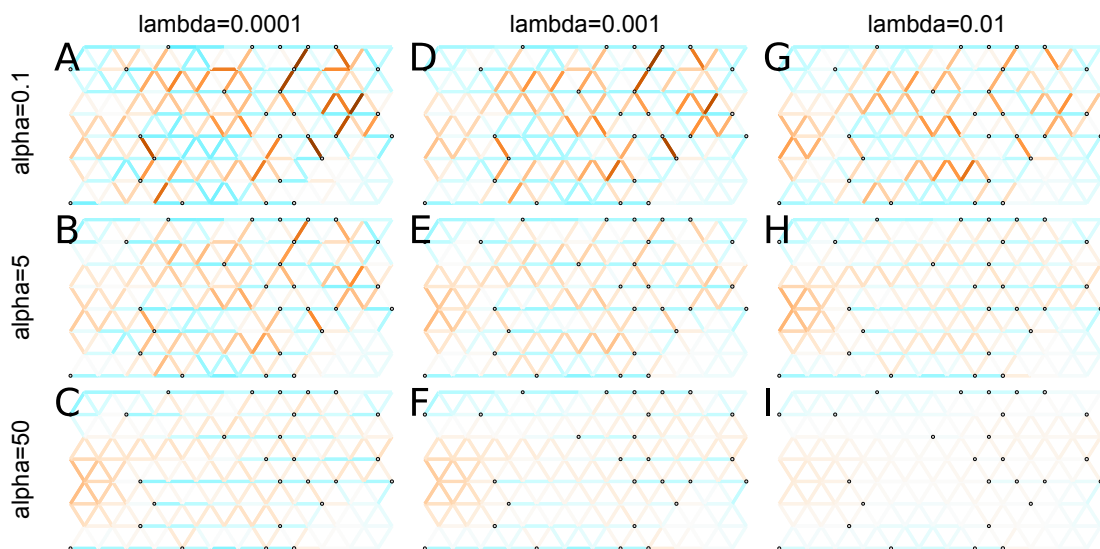
(A) Map of sample coordinates (black points) from a dataset of gray wolves from North America. The input to FEEMS are latitude and longitude coordinates as well as genotype data for each sample. (B) Map of sample coordinates with an example dense spatial grid. The nodes of the grid represent sub-populations and the edges represent local gene-flow between adjacent sub-populations. (C) Individuals are assigned to nearby nodes (sub-populations) and summary statistics (e.g., allele frequencies) are computed for each observed location.



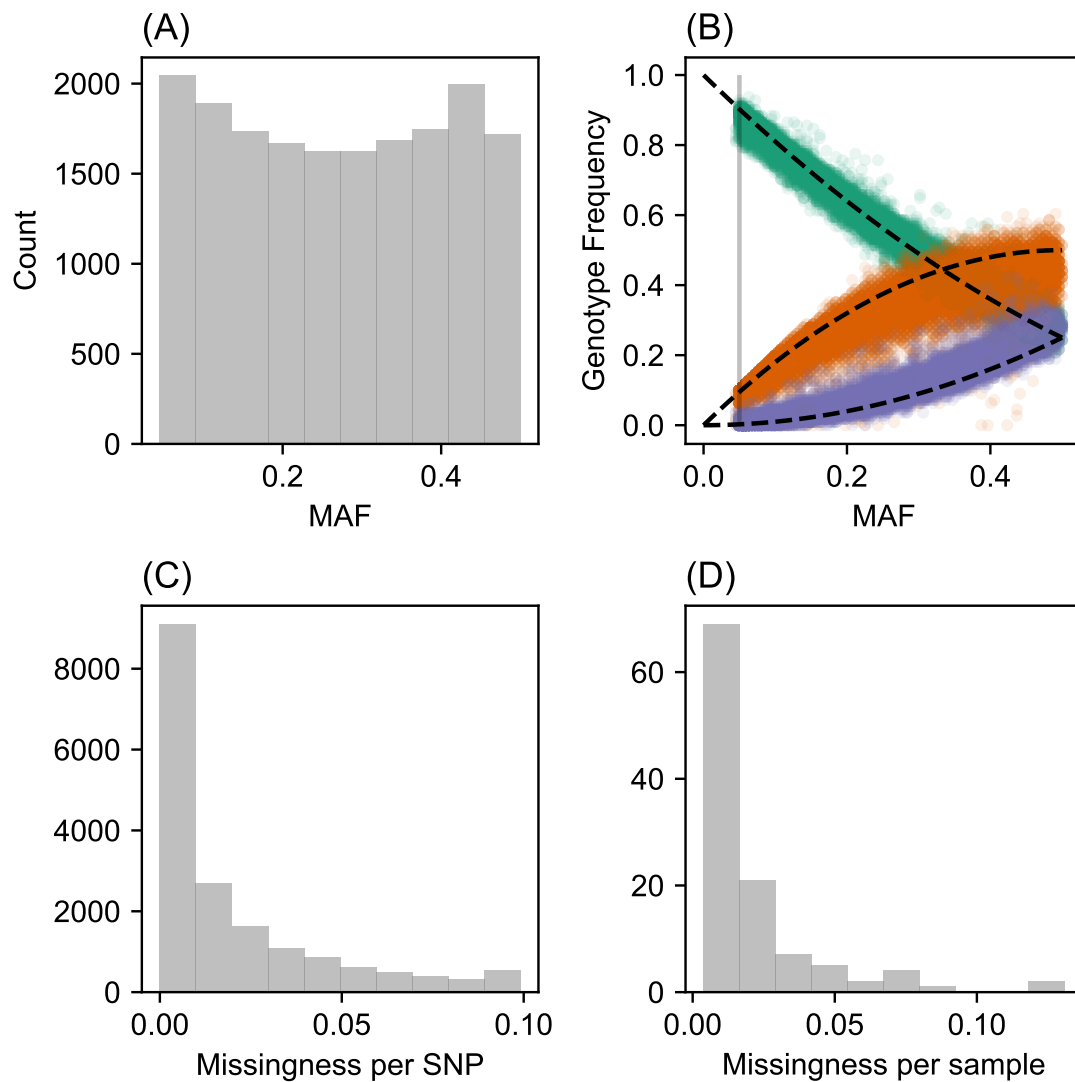
Supplementary Figure 2: Application of FEEMS to an extended set of coalescent simulations: We display an extended set of coalescent simulations with multiple migration scenarios and sampling designs. The sample sizes across the grid are represented by the size of the grey dots at each node. The migration rates are obtained by solving FEEMS objective function (9) where the regularization parameters are specified at $\lambda = 10^{-2}, \alpha = 30$ (I), $\lambda = 10^{-4}, \alpha = 30$ (N), and $\lambda = 10^{-3}, \alpha = 30$ for the rest. (A, F, K) display the ground truth of the underlying migration rates. (B, G, L) Shows simulations where there is no missing data on the graph. (C, H, M) Shows simulations with sparse observations and nodes missing at random. (D, I, N) Shows simulations of biased sampling where there are no samples from the center of the simulated habitat. (E, J, O) Shows simulations of biased sampling where there are only samples on the right side of the habitat.



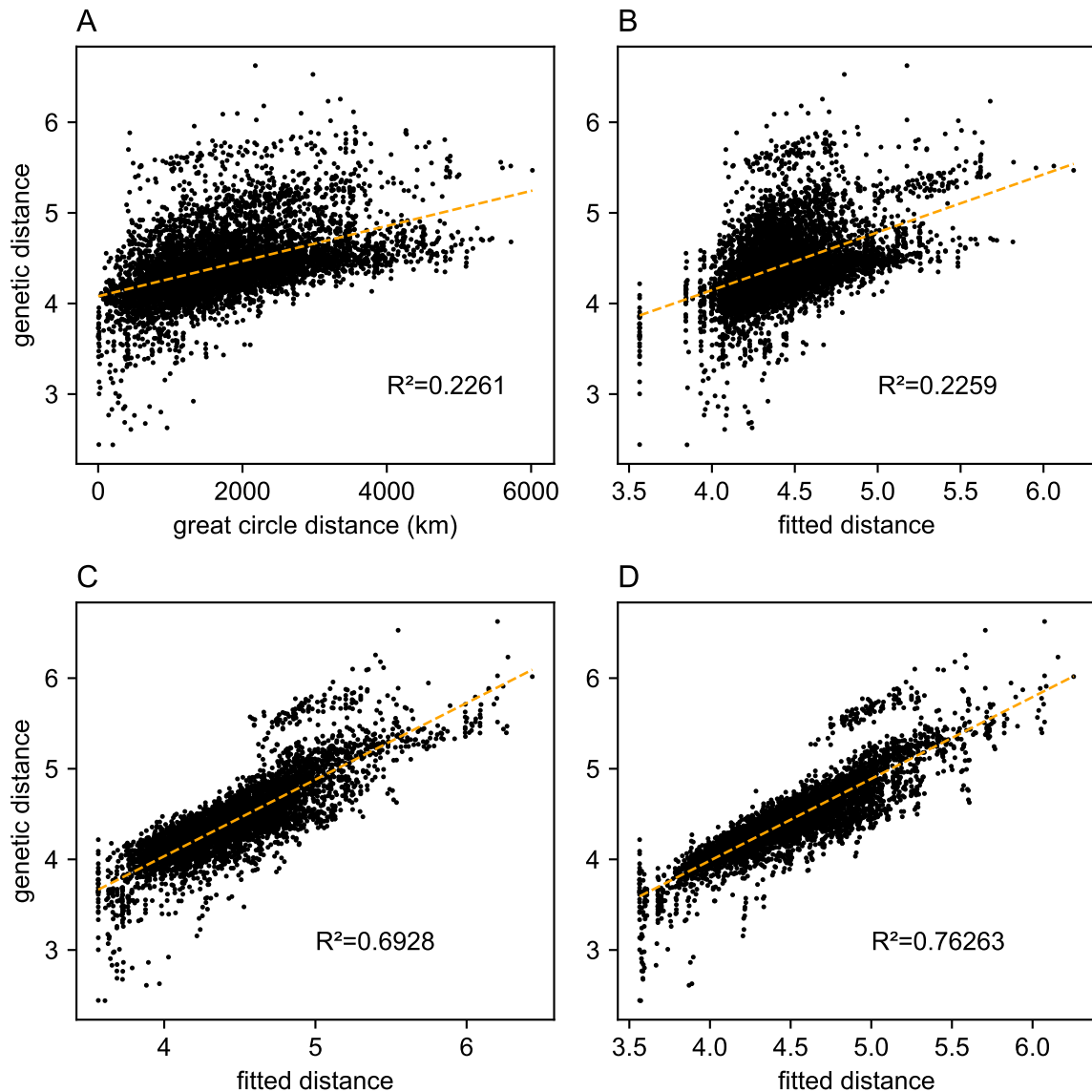
Supplementary Figure 3: Application of FEEMS to a heterogeneous migration scenario with a “missing at random” sampling design: We run FEEMS on coalescent simulation with a non-homogeneous process while varying hyperparameters λ (rows) and α (columns). We randomly sample individuals for 20% of nodes. When λ grows, the fitted graph becomes overall smoother, whereas α effectively controls the degree of similarity among low migration rates.



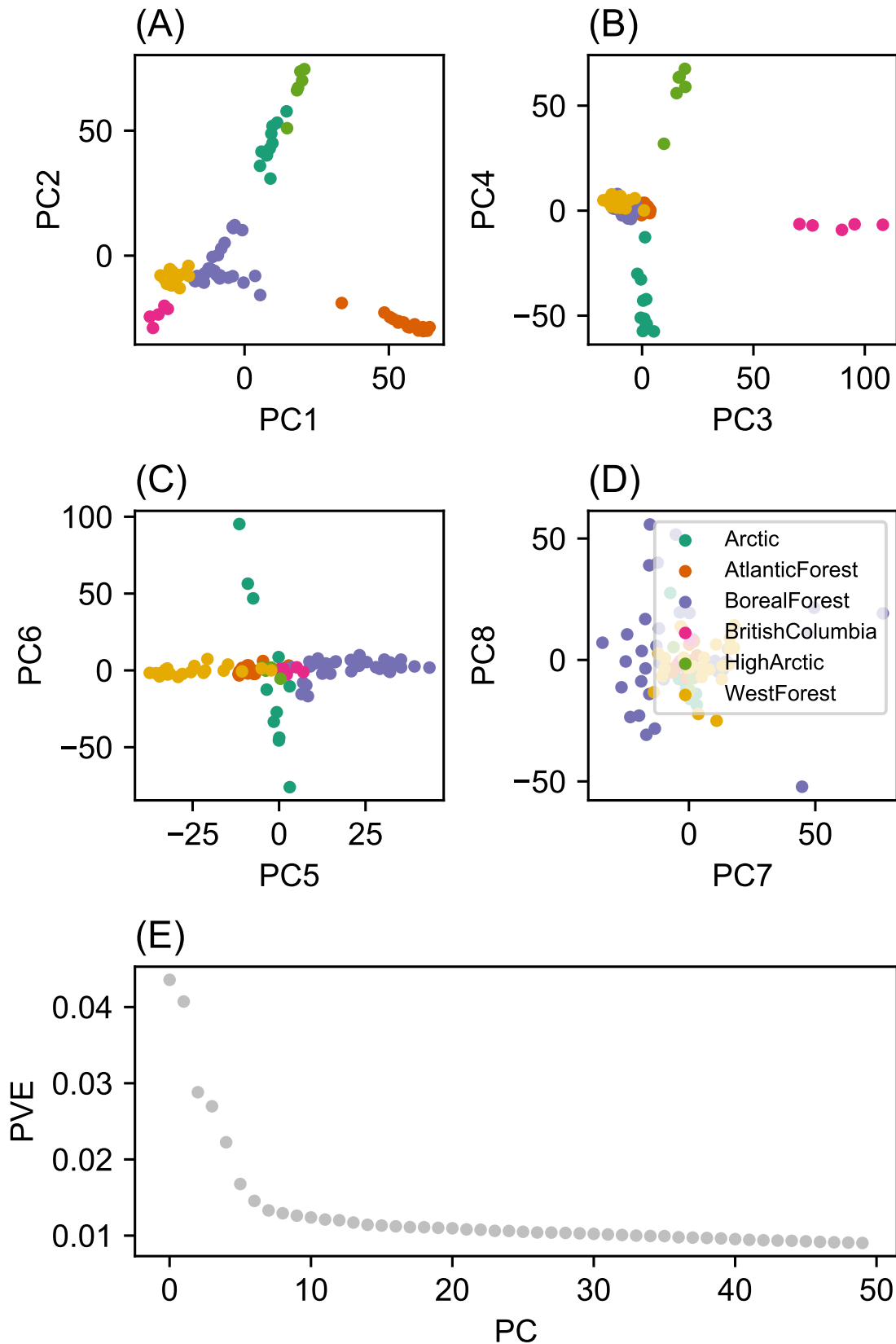
Supplementary Figure 4: Application of FEEMS to an anisotropic migration scenario with a “missing at random” sampling design: We run FEEMS on coalescent simulation with an anisotropic process while varying hyperparameters λ (rows) and α (columns). We randomly sample individuals for 20% of nodes. When λ grows, the fitted graph becomes overall smoother, whereas α effectively controls the degree of similarity among low migration rates.



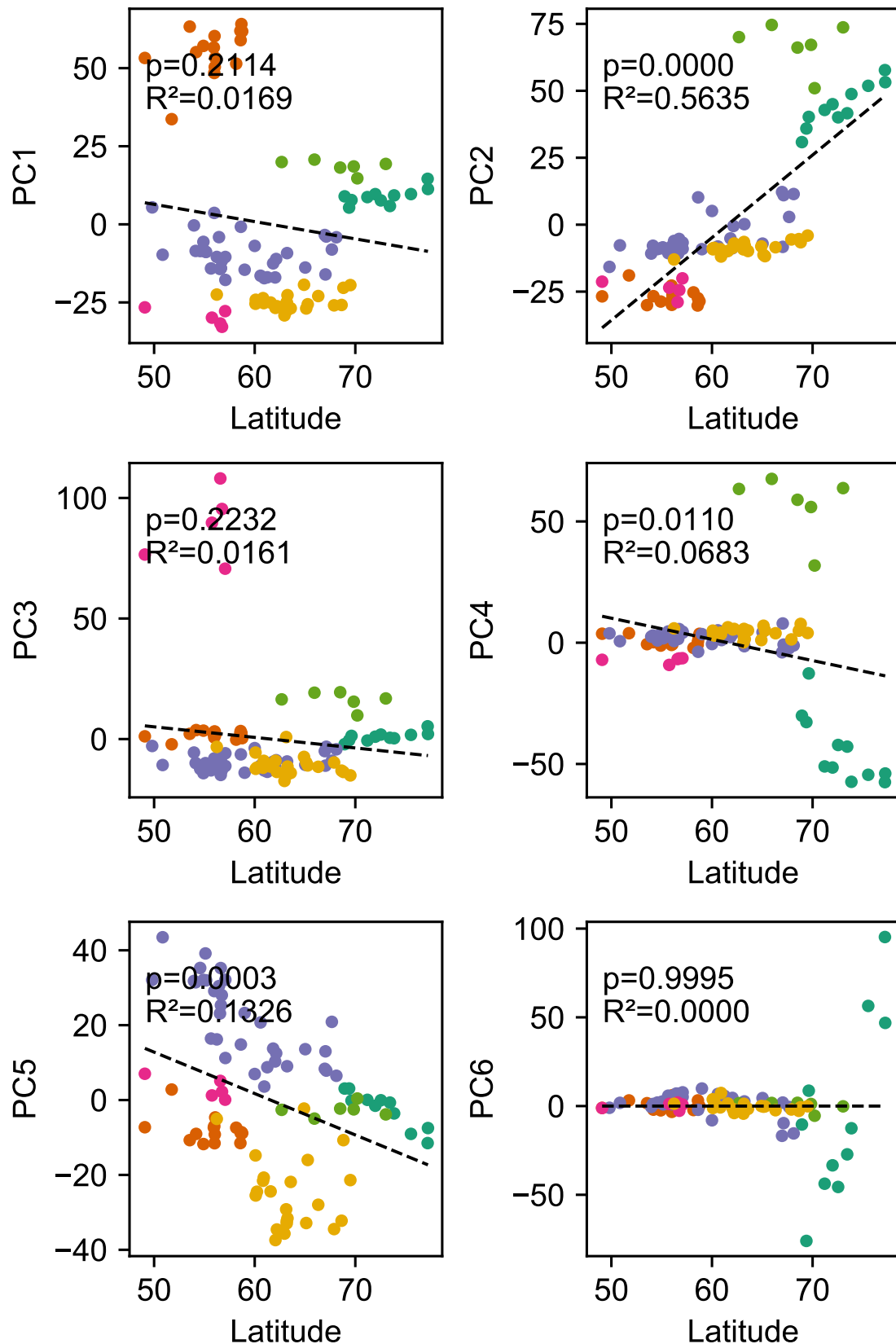
Supplementary Figure 5: SNP and individual quality control: (A) Displays a visualization of the sample site frequency spectrum. Specifically, we display a histogram of minor allele frequencies across all SNPs. We see a relatively uniform histogram which reflects the ascertainment of common SNPs on the array that was designed to genotype gray wolf samples. (B) Visualization of allele frequencies plotted against genotype frequencies. Each point represents a different SNP and the colors represent the 3 possible genotype values. The black dashed lines display the expectation as predicted from a simple binomial sampling model i.e. Hardy-Weinberg equilibrium. (C) Displays a histogram of the missingness fraction per SNP. We observe the missingness tends to be relatively low for each SNP. (D) Displays a histogram of the missingness fraction per sample. Generally, the missingness tends to be low for each sample.



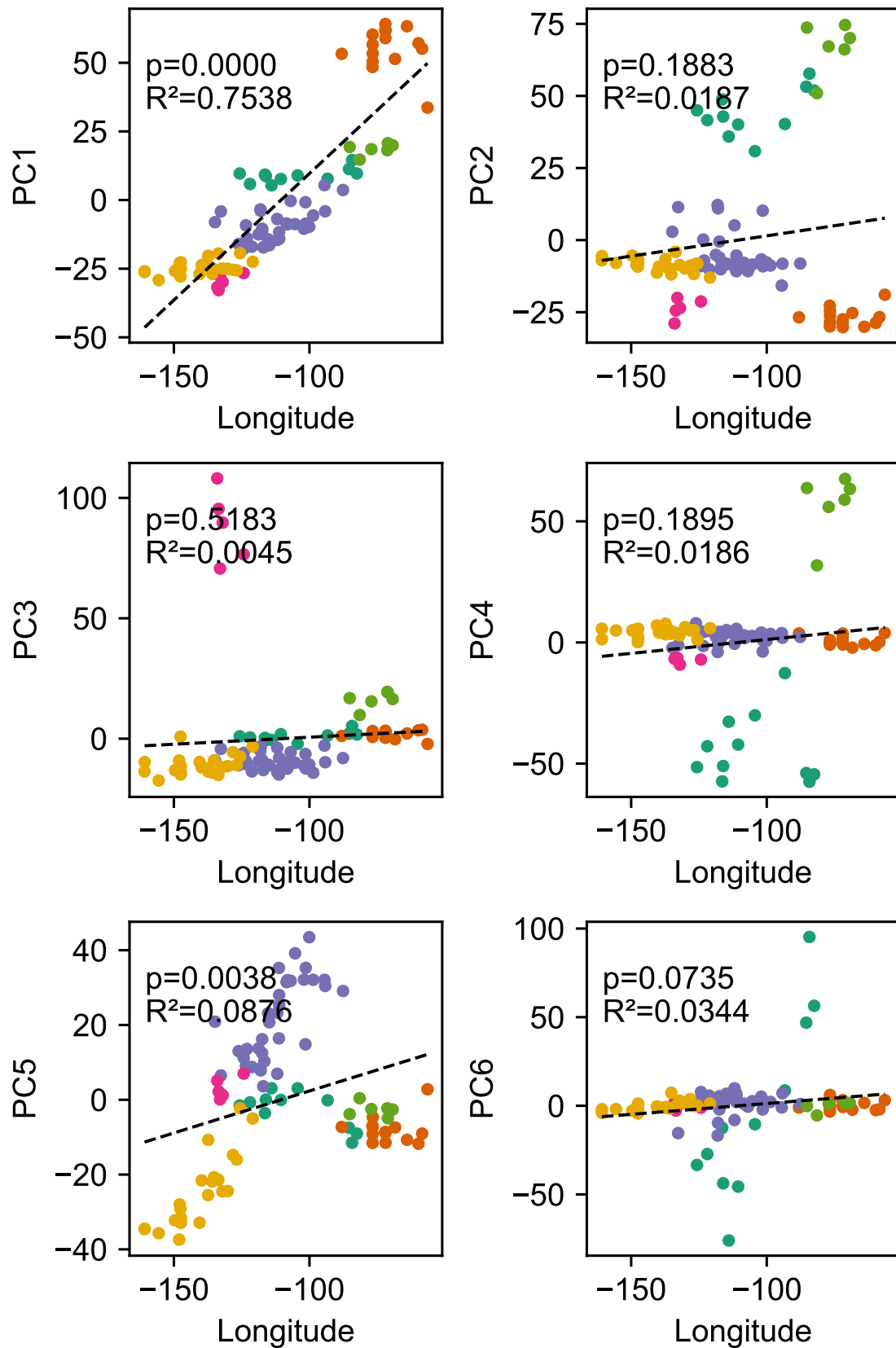
Supplementary Figure 6: Comparing predictions of observed genetic distances: We display different predictions of observed genetic distances using geographic distance or the fitted genetic distance output by FEEMS. (A) The x-axis displays the geographic distance between two individuals, as measured by the great circle distance (haversine distance). The y-axis displays the squared Euclidean distance between two individuals averaged over all SNPs. (B-D) The x-axis displays the fitted genetic distance as predicted by the FEEMS model and y-axis displays the squared Euclidean distance between two individuals averaged over all SNPs. For (B-D) we display the fit of λ getting subsequently smaller ($10, 10^{-3}, 10^{-5}$) and as expected the fit becomes better because we tolerate more complex surfaces and we are not evaluating the fit on out-of-sample data.



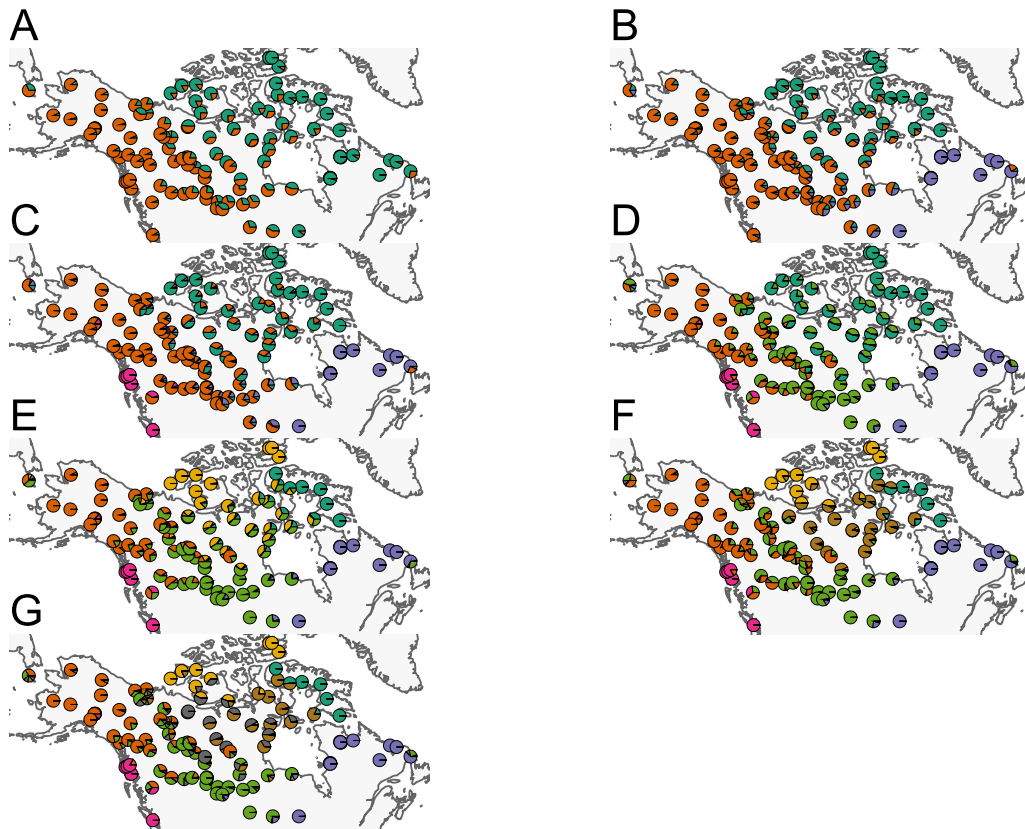
Supplementary Figure 7: Summary of top axes of genotypic variation: We display a visual summary of Principal Components Analysis (PCA) applied to the normalized genotype matrix from the North American gray wolf dataset. (A-D) Displays PC bi-plots of the top seven PCs plotted against each other. The colors represent predefined ecotypes defined in (Schweizer et al., 2016). We can see that the top PCs delineate these predefined ecotypes. (E) Shows a “scree” plot with the proportion of variance explained for each of the top 50 PCs. As expected by genetic data (Patterson et al., 2006), the eigen-values of the genotype matrix tend to be spread over many PCs.



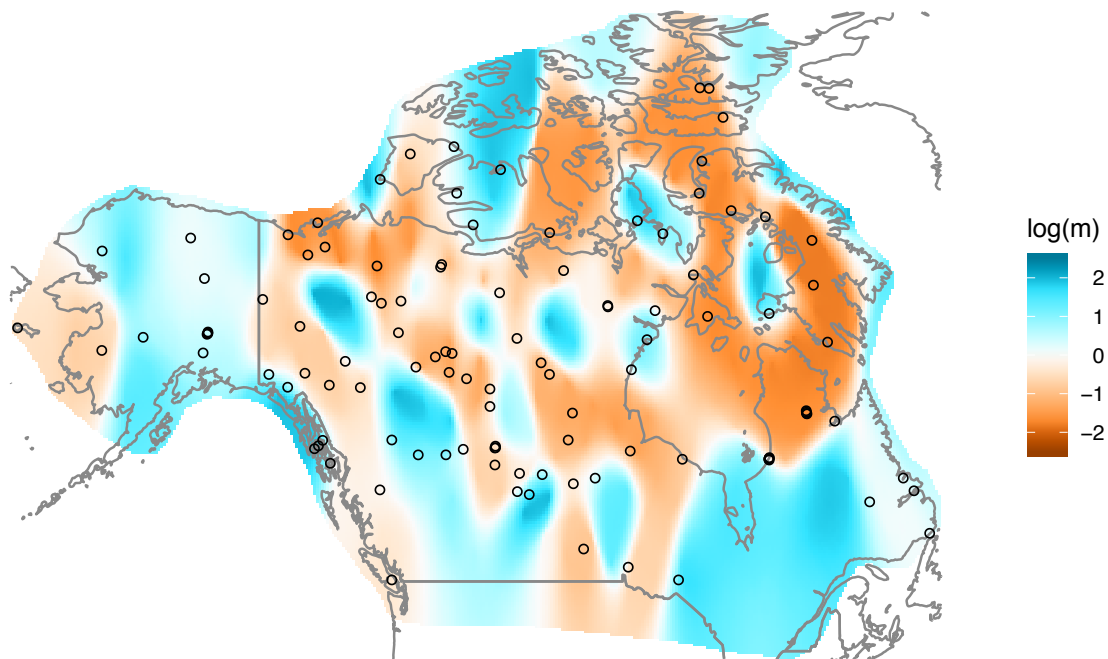
Supplementary Figure 8: Relationship between top axes of genetic variation and latitude: In each sub-panel we plot the PC value against latitude for each sample in gray the wolf dataset. We see many of the top PCs are significantly correlated with latitude as tested by linear regression.



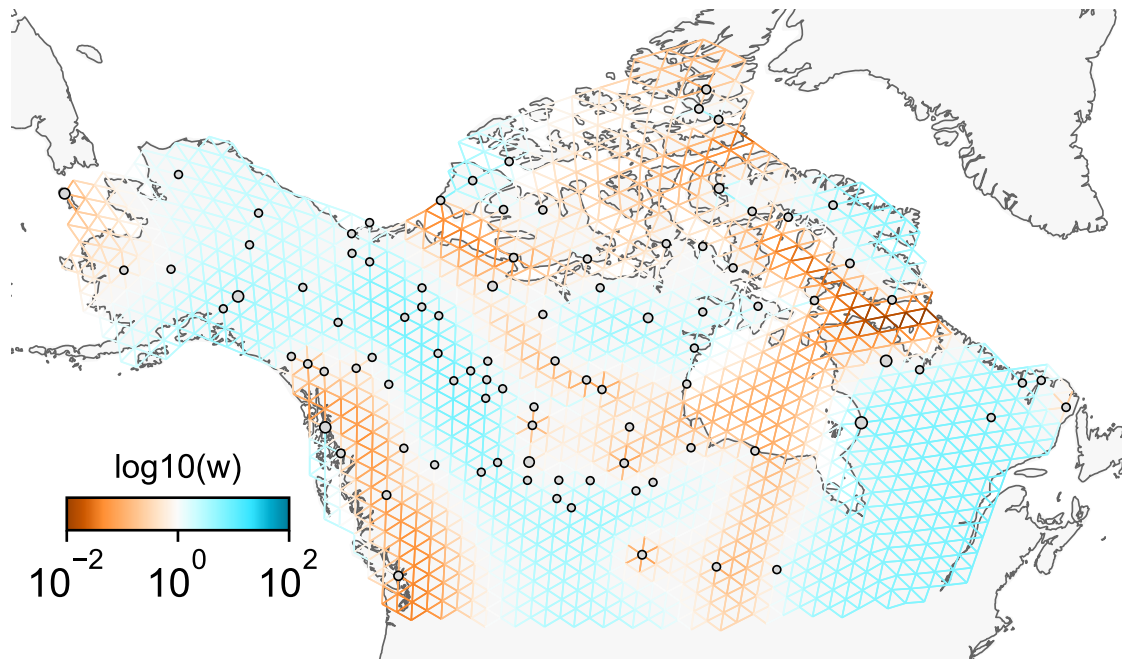
Supplementary Figure 9: Relationship between top axes of genetic variation and longitude: In each sub-panel we plot the PC value against longitude for each sample in the gray wolf dataset. We see many of the top PCs are significantly correlated with longitude as tested by linear regression.



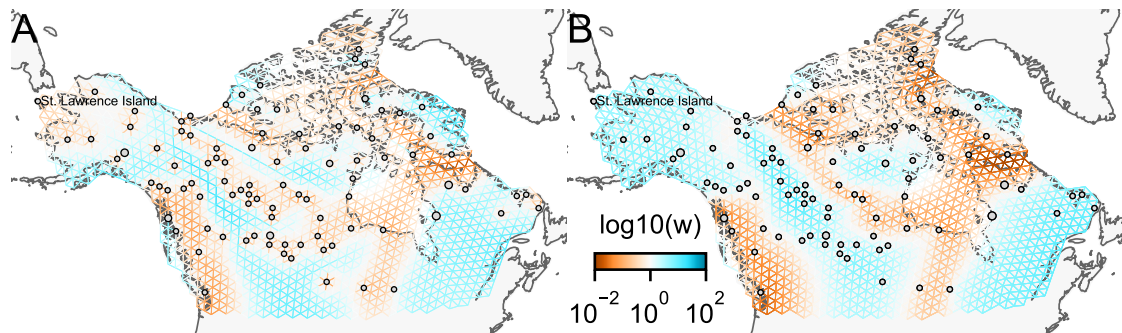
Supplementary Figure 10: Summary of ADMIXTURE results: (A-G) Visualization of ADMIXTURE results for $K = 2$ to $K = 8$. We display admixture fractions for each sample as colored slices of the pie chart on the map. For each K we ran 5 replicate runs of ADMIXTURE and in this visualization we display the solution that achieves the highest likelihood amongst the replicates. The ADMIXTURE results qualitatively reveal a spatial signal in the data as admixture fractions tend to be spatially clustered.



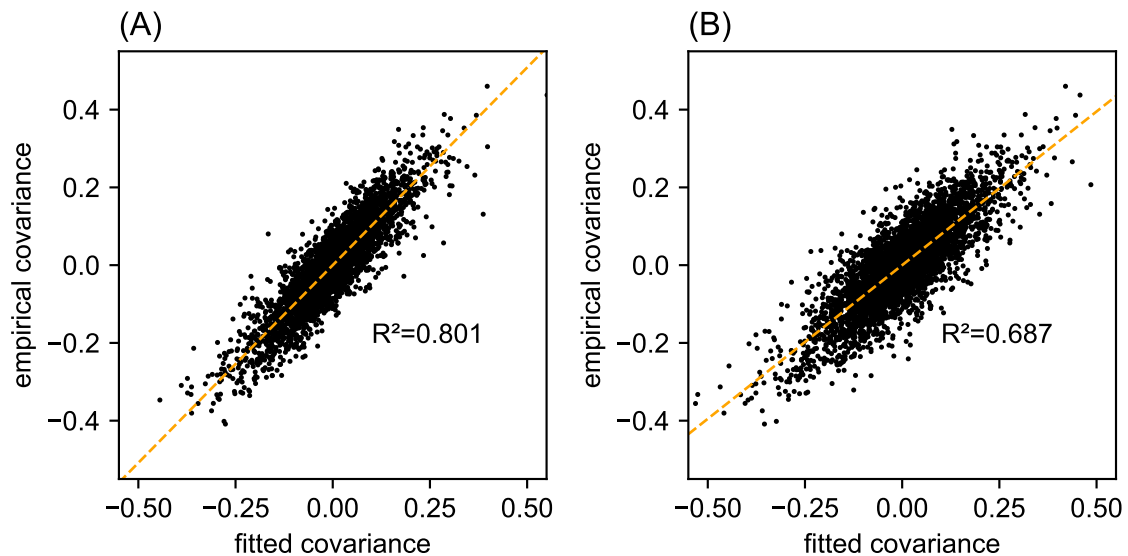
Supplementary Figure 11: Application of EEMS to the North American gray wolf dataset: We display a visualization of EEMS applied to the North American gray wolf dataset. The more orange colors represent lower than average effective migration on the log-scale and the more blue colors represent higher than average effective migration on the log-scale. The results of EEMS are qualitatively similar to FEEMS.



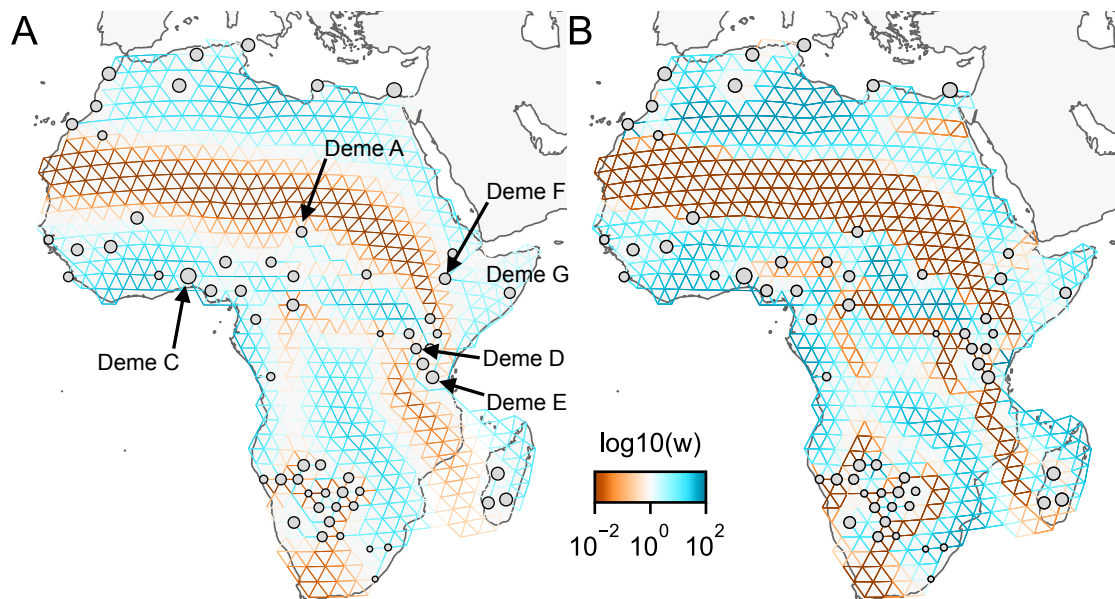
Supplementary Figure 12: Application of FEEMS on the North American gray wolf dataset with an exact likelihood model: We display the fit of FEEMS based in the formulation (19) to the North American gray wolf dataset. This fit corresponds to a setting of tuning parameters at $\lambda = 10^{-3}$, $\alpha = 50$. Additionally we set the lower bound of the edge weights to $l = 0.01$, to ensure that the diagonal elements of \mathbf{L} does not become too small—this has an implicit effect on L_{kk}^\dagger , preventing it from blowing up at unobserved nodes. The more orange colors represent lower than average effective migration on the log-scale and the more blue colors represent higher than average effective migration on the log-scale. Visually the result is comparable to that of FEEMS fit (Figure 4) based in the formulation (9).



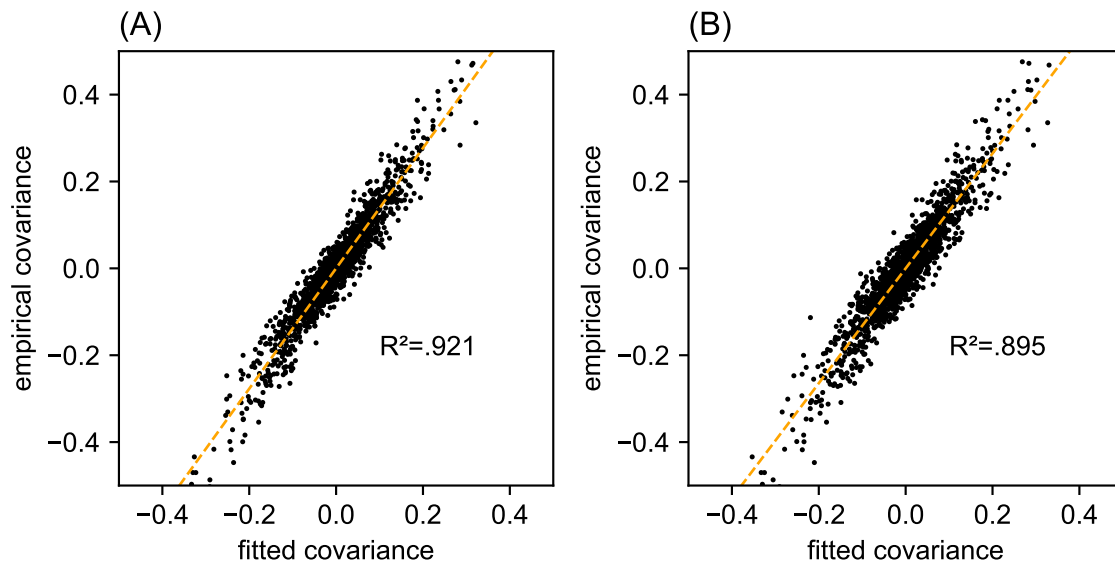
Supplementary Figure 13: Application of FEEMS on the North American gray wolf dataset with joint estimation of the residual variance and graph's edge weights: We show visualizations of fits of FEEMS to the North American gray wolf dataset when the residual variance and edge weights of the graph are jointly estimated. Both fits correspond to a setting of tuning parameters at $\lambda = 10^{-3}$, $\alpha = 50$. (A) Displays the estimated effective migration surfaces where every deme shares a single residual parameter σ . The result is similar to the procedure that prefixes σ from the homogeneous isolation by distance model (Figure 4), except the high migration edge forming long path in (A) which disappears with higher values of α . (B) Displays the estimated effective migration surfaces where each node has its own residual parameter σ_k for all nodes k . These node specific residual parameters allow more flexible graphs, but at the cost of over-fitting to the data. In particular, without adding smooth regularization term on the residual variances, it fails to recover some geographic features like St. Lawrence Island.



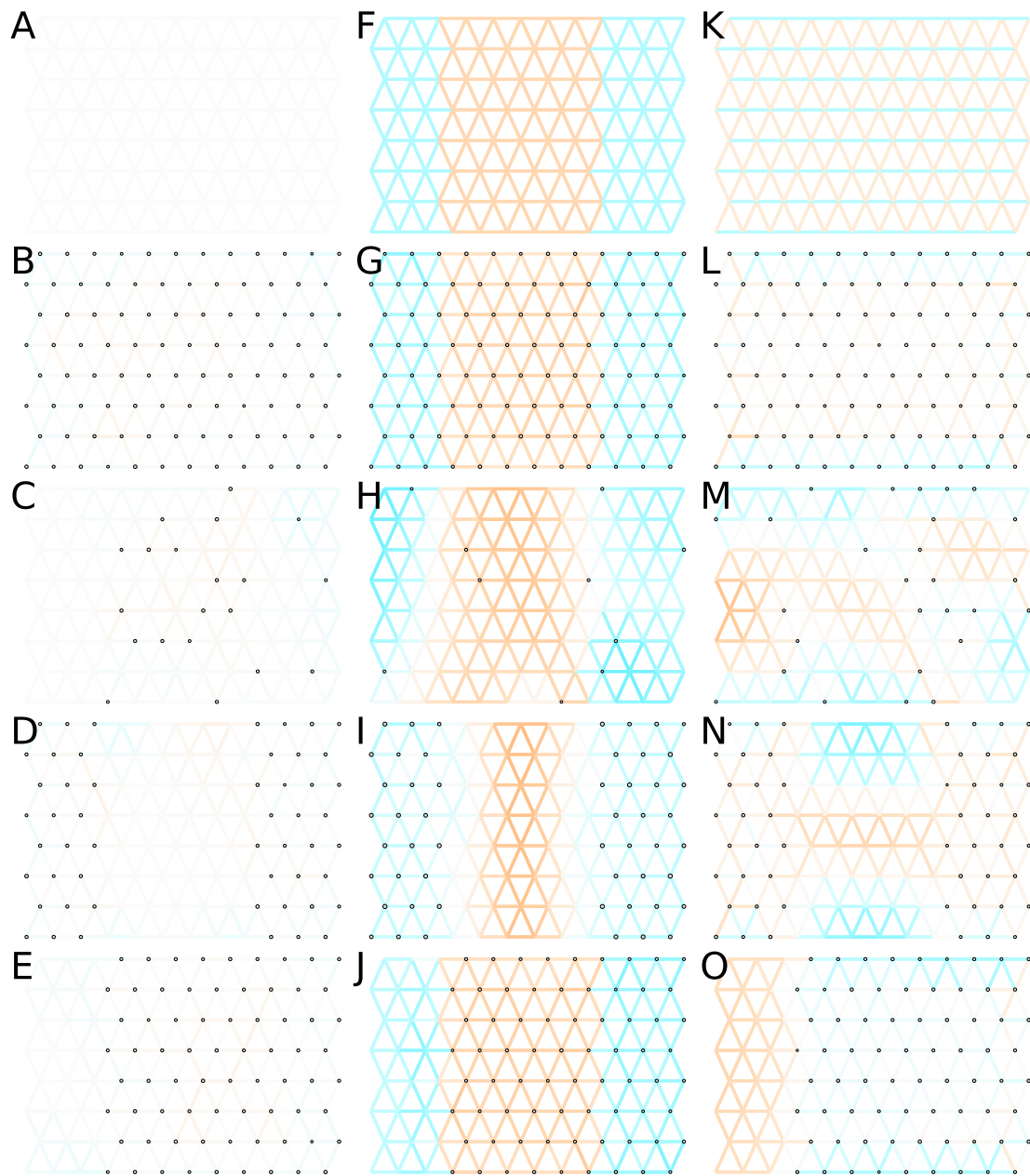
Supplementary Figure 14: Relationship between fitted and empirical covariance on the North American gray wolf dataset: We display scatter plots of empirical genetic covariances versus fitted covariances from FEEMS fits on the gray wolf dataset. (A) Corresponds to the result shown in Figure 4. (B) Corresponds to the result shown in Supp. Fig. 13B. The x-axis represents the transformed fitted covariance matrix, i.e. $CA\hat{L}^\dagger A^\top C^\top + \hat{\sigma}^2 C \text{diag}(n^{-1}) C^\top$ (see equation (6)). The y-axis represents the transformed sample covariance matrix, i.e. $C\hat{\Sigma}C^\top$. The simple linear regression fit is shown in orange dashed lines and R^2 is given.



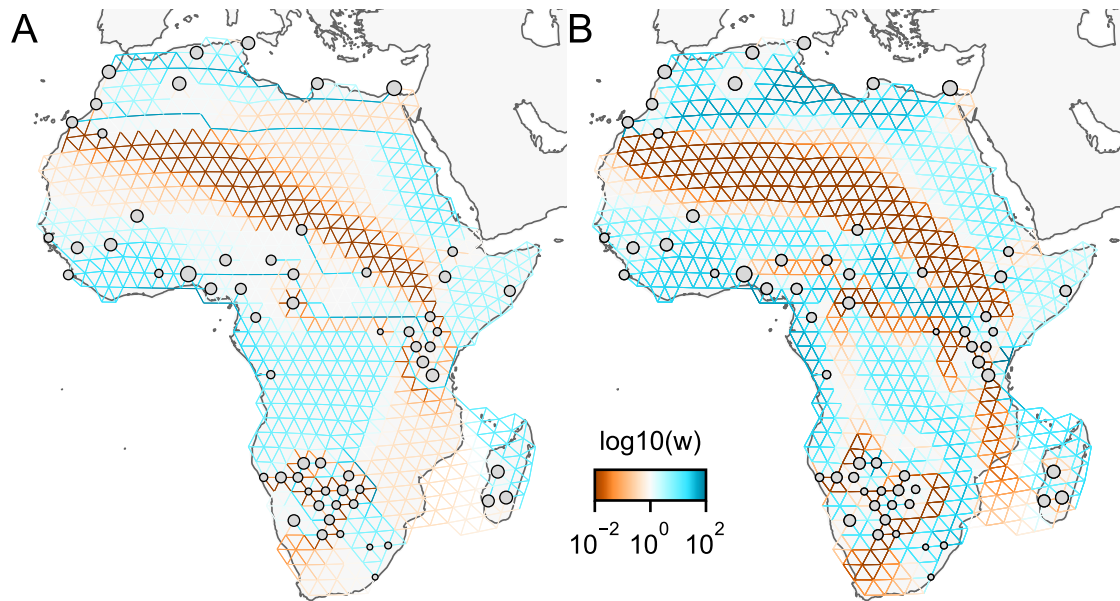
Supplementary Figure 15: Application of FEEMS to a dataset of human genetic variation from Africa with different parameterization: We display visualizations of FEEMS to a dataset of human genetic variation from Africa with different parameterization of the graph's edge weights. See (Peter et al., 2018) for the description of the dataset. (A) Displays the recovered graph under the edge parameterization. (B) Displays the recovered graph under the node parameterization. Both parameterization have their own regularization parameters λ and α , but these parameters are not on the same scale. We set $\lambda = 2 \cdot 10^{-4}$, $\alpha = 10$ for the node parameterization which is seen to yield similar results to those in (Peter et al., 2018). For the edge parameterization, we keep the same λ value while we set $\alpha = 60$ so that the resulting graph reveals similar geographic structure to the node parameterization. We also set the lower bound $l = 0.01$. From the plots, it is worth noting two important distinctions: (1) We see the migration surfaces shown in (B) recover sharper edge features while the migration surfaces in (A) are overall smoother. This is attributed to the fact that node parameterization has its own additional regularization effect on the edge weights, and in order to achieve similar degree of regularization strength for the edge parameterization, it needs a higher regularization parameters, which results in more blurring edges than the node parameterization. (2) When measuring correlation of the estimated allele frequencies among nodes, we find that Deme B is the node with the second highest correlation to Deme A, whereas Deme C (and nearby demes) is not as much correlated to Deme A compared to Deme B. Panel (A) reflects this feature by exhibiting a corridor between Deme A and Deme B and reduced gene-flow beneath that corridor. This reduced gene-flow disappears in (B), even if the regularization parameters are varied over a range of values. Additionally, Deme D is most highly correlated to Deme E, F, and G, and this is implicated by a long-range corridor connecting those demes appearing in Panel (A) while not shown in (B). These results point a conclusion that the form of the node parameterization is perhaps too strong and in this case it limits model's ability to capture desirable geographic features that are subtle to detect.



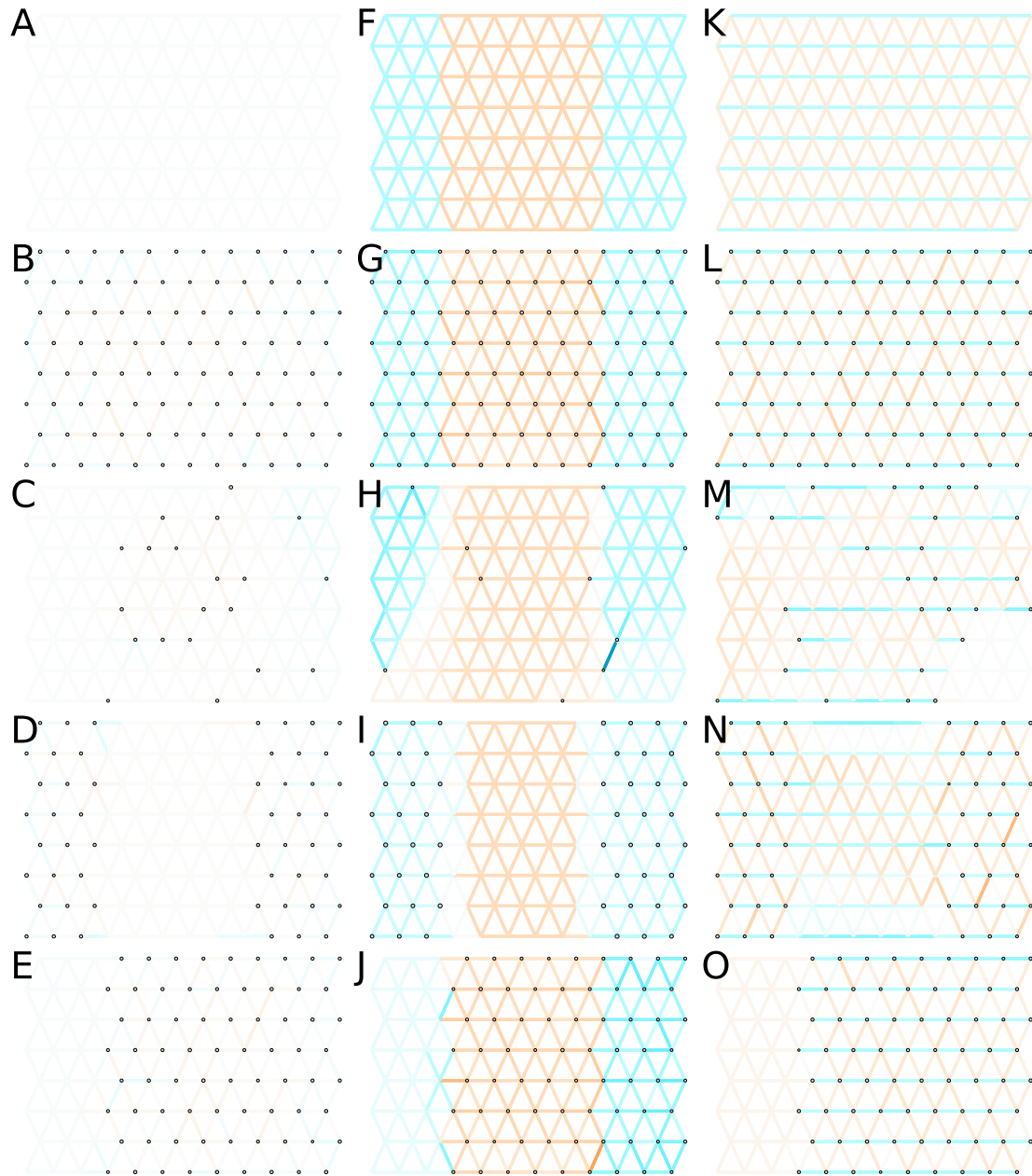
Supplementary Figure 16: Relationship between fitted and empirical covariance on a dataset of human genetic variation from Africa: We display scatter plots of empirical genetic covariance versus fitted covariance from FEEMS fits on the African dataset. (A) Corresponds to the result shown in Supp. Fig. 15A. (B) Corresponds to the result shown in Supp. Fig. 15B. The x-axis represents the transformed fitted covariance matrix, i.e. $CAL^\dagger A^\top C^\top + \hat{\sigma}^2 C \text{diag}(n^{-1}) C^\top$ (see equation (6)). The y-axis represents the transformed sample covariance matrix, i.e. $C\hat{\Sigma}C^\top$. The simple linear regression fit is shown in orange dashed lines and R^2 is given.



Supplementary Figure 17: Application of FEEMS based on node parameterization to an extended set of coalescent simulations: We display an extended set of coalescent simulations with the same migration scenarios and sampling designs as Supp. Fig. 2. The sample sizes across the grid are represented by the size of the grey dots at each node. The migration rates are obtained by solving the FEEMS objective function (9) with node parameterization where the regularization parameters are specified at $\lambda = 10^{-3}$, $\alpha = 50$. (A, F, K) display the ground truth of the underlying migration rates. (B, G, L) Shows simulations where there is no missing data on the graph. (C, H, M) Shows simulations with sparse observations and nodes missing at random. (D, I, N) Shows simulations of biased sampling where there are no samples from the center of the simulated habitat. (E, J, O) Shows simulations of biased sampling where there are only samples on the right side of the habitat.



Supplementary Figure 18: Application of ℓ_1 -norm-based FEEMS to a dataset of human genetic variation from Africa: We display visualizations of FEEMS to a dataset of human genetic variation from Africa with the ℓ_1 -based penalty function. See [Peter et al. \(2018\)](#) for the description of the dataset. (A) Displays the recovered graph under the edge parameterization with ℓ_1 norm based penalty where the regularization parameters are specified at $\lambda = 4 \cdot 10^{-2}, \alpha = 30$. (B) Displays the recovered graph under the node parameterization with ℓ_1 norm based penalty where the regularization parameters are specified at $\lambda = 4 \cdot 10^{-2}, \alpha = 1$. To minimize the objective (20), linearized ADMM is applied with 20,000 number of iterations. The lower bound is set to be $l = 0.01$ for both parameterizations. Note that due to the high degrees of missingness, the estimated effective migration surfaces using solely ℓ_1 -based penalty exhibit many likely artifacts (e.g., high migration edges forming long paths, seen in A) unless an additional penalty term is added to promote global smoothness of the edge weights such as a combination of ℓ_1 norm penalty function and node parameterization as shown in (B).



Supplementary Figure 19: Application of ℓ_1 -norm-based FEEMS to an extended set of coalescent simulations: We display an extended set of coalescent simulations with the same migration scenarios and sampling designs as Supp. Fig. 2. The sample sizes across the grid are represented by the size of the grey dots at each node. The migration rates are obtained by solving ℓ_1 norm based FEEMS objective (20) where the regularization parameters are specified at $\lambda = 10^{-1}, \alpha = 30$ (I), $\lambda = 10^{-3}, \alpha = 30$ (N), and $\lambda = 10^{-2}, \alpha = 30$ for the rest. (A, F, K) display the ground truth of the underlying migration rates. (B, G, L) Shows simulations where there is no missing data on the graph. (C, H, M) Shows simulations with sparse observations and nodes missing at random. (D, I, N) Shows simulations of biased sampling where there are no samples from the center of the simulated habitat. (E, J, O) Shows simulations of biased sampling where there are only samples on the right side of the habitat.