Estimating the Number of Subpopulations (K) in Structured Populations

Robert Verity^{*.1} and Richard A. Nichols[†]

^{*}Medical Research Council Centre for Outbreak Analysis and Modelling, Imperial College London, London W2 1PG, United Kingdom, and [†]Queen Mary University of London, London E1 4NS, United Kingdom

ORCID ID: 0000-0002-4801-9312 (R.A.N.).

ABSTRACT A key quantity in the analysis of structured populations is the parameter K , which describes the number of subpopulations that make up the total population. Inference of K ideally proceeds via the *model evidence*, which is equivalent to the likelihood of the model. However, the evidence in favor of a particular value of K cannot usually be computed exactly, and instead programs such as Structure make use of heuristic estimators to approximate this quantity. We show—using simulated data sets small enough that the true evidence can be computed exactly—that these heuristics often fail to estimate the true evidence and that this can lead to incorrect conclusions about K . Our proposed solution is to use thermodynamic integration (TI) to estimate the model evidence. After outlining the TI methodology we demonstrate the effectiveness of this approach, using a range of simulated data sets. We find that TI can be used to obtain estimates of the model evidence that are more accurate and precise than those based on heuristics. Furthermore, estimates of K based on these values are found to be more reliable than those based on a suite of model comparison statistics. Finally, we test our solution in a reanalysis of a white-footed mouse data set. The TI methodology is implemented for models both with and without admixture in the software Maverick1.0.

KEYWORDS population structure; K ; model evidence; thermodynamic integration; model comparison

THE detection and characterization of population structure is one of the cornerstones of modern population genetics. Ever since Wright (1949) and his contemporaries (Malécot 1948) it has been recognized that genetic samples obtained from a large population may be better understood as a series of draws from multiple partially isolated subpopulations or demes. While traditional methods (such as those based on the fixation index, F_{ST}) assume that the allocation of individuals to demes is known *a priori*, many modern programs such as Structure (Pritchard *et al.* 2000; Falush *et al.* 2003a, 2007; Hubisz *et al.* 2009) take a different approach, attempting to infer the group allocation from the observed data. What makes this possible is the simple genetic mixture modeling framework used by these programs, together with the efficiency of Markov chain Monte Carlo (MCMC) methods for sampling from this broad class of models.

However, even within the flexible framework of Bayesian mixture models, the number of demes (denoted K) is difficult to ascertain. While the allocation of individuals to demes is a parameter *within* a particular model, the value of K is fixed for a given mixture model, and so the problem of estimating K involves a comparison *between* models. One of the most common ways of comparing between models in a Bayesian setting is through the model evidence, defined as the probability of the observed data under the model (equivalently the likelihood of the model). This quantity can be estimated for a range of K , and the model with the highest evidence value can then become the focus of our analysis. However, there are two potential issues with this approach. The first one is philosophical and revolves around the idea that there is a single true value of K that we can estimate from the data. In reality populations are rarely divided into discrete subpopulations, and so the idea of a single true value of K does not strictly apply. This does not mean that K cannot be a useful quantity, but it is better viewed as a flexible parameter that describes just one point on a continuously varying scale of population structure. This flexible interpretation of K has been advocated by a number of previous authors (Raj *et al.* 2014; Jombart and Collins 2015), including the authors of the Structure program (Pritchard *et al.* 2010).

Copyright © 2016 by the Genetics Society of America

doi: 10.1534/genetics.115.180992

Manuscript received July 21, 2015; accepted for publication June 4, 2016; published Early Online June 10, 2016.

Supplemental material is available online at www.genetics.org/lookup/suppl/doi:10.1534/genetics.115.180992/-/DC1.

¹Corresponding author: MRC Centre for Outbreak Analysis and Modelling, Imperial College London, London W2 1PG, United Kingdom. E-mail: r.verity@imperial.ac.uk

The second issue is purely statistical—computing the model evidence in complex, multidimensional models is not straightforward. For this reason it is common to resort to heuristic estimators of the true evidence. These heuristics tend to have some direct mathematical connection to the model evidence, but also make certain simplifying assumptions in their derivation. For example, in the original article on which Structure is based, Pritchard *et al.* (2000) comment on the difficulties in obtaining the model evidence directly and instead opt for an *ad hoc* procedure in which a heuristic (denoted L_K here) is used as an approximation of $-2 \times \log(\text{evidence})$. The derivation of this statistic rests on certain simplifying assumptions, and the authors are careful to emphasize that these assumptions are “dubious.”

Here we focus on the latter problem: reliable estimation of the model evidence. Rather than resorting to heuristics, what we want is a direct way of estimating the model evidence that is both accurate and straightforward to implement. As noted by Gelman and Meng (1998), such a method already exists and has been known in the physical sciences for some time. This method—referred to in the statistical literature as *thermodynamic integration* (TI)—uses the output of several closely related MCMC chains to obtain a direct estimate of the evidence. Crucially, this is not just another heuristic. Rather, it is a true statistical estimator that can be evaluated to an arbitrary degree of precision by simply increasing the number of MCMC iterations used in the calculation. The TI methodology was introduced into population genetics by Lartillot and Philippe (2006) and has since been applied to a range of problems in phylogenetics and coalescent theory, including comparing models of demographics (Baele *et al.* 2012), migration (Beerli and Palczewski 2010), relaxed molecular clocks (Lepage *et al.* 2007), and sequence evolution (Blanquart and Lartillot 2006).

In the remainder of this article we demonstrate the effectiveness of TI as a method for estimating K in simple genetic mixture models. For small data sets we find that the TI estimator is several orders of magnitude more accurate and precise than the L_K estimator for the same computational effort. We also explore the ability of different statistics to correctly estimate K for larger data sets, finding that TI outperforms Evanno’s ΔK (Evanno *et al.* 2005), the Akaike information criterion (AIC), the Bayesian information criterion (BIC), and the deviance information criterion (DIC). Finally we reanalyze data from an earlier study on the genetic structure of white-footed mouse populations in New York City (Munshi-South and Kharchenko 2010b). All of the methods described here are made available through the program MaverickK (www.bobverity.com/MaverickK).

Materials and Methods

Evidence and Bayes factors

In a Bayesian setting the problem of deciding between competing models can be addressed using Bayes’ rule. The

posterior probability of the model $\mathcal{M}$, given the observed data $\mathbf{x}$, can be written

$$\Pr(\mathcal{M}|\mathbf{x}) = \frac{\Pr(\mathbf{x}|\mathcal{M}) \Pr(\mathcal{M})}{\Pr(\mathbf{x})}. \quad (1)$$

The quantity $\Pr(\mathbf{x}|\mathcal{M})$ —the probability of the observed data $\mathbf{x}$ given just the model $\mathcal{M}$ —is defined as the model evidence.

The ratio of the evidence between competing models, known as the *Bayes factor*, can be used to measure the strength of evidence in favor of one model over another. Bayes factors can be used on their own, or they can be combined with priors on the different models to arrive at the posterior odds:

$$\underbrace{\frac{\Pr(\mathcal{M}_1|\mathbf{x})}{\Pr(\mathcal{M}_2|\mathbf{x})}}_{\text{posterior odds ratio}} = \underbrace{\frac{\Pr(\mathbf{x}|\mathcal{M}_1)}{\Pr(\mathbf{x}|\mathcal{M}_2)}}_{\text{Bayes factor}} \times \underbrace{\frac{\Pr(\mathcal{M}_1)}{\Pr(\mathcal{M}_2)}}_{\text{prior odds ratio}}. \quad (2)$$

A large Bayes factor in (2) provides evidence in favor of model $\mathcal{M}_1$ over model $\mathcal{M}_2$, whereas a small Bayes factor provides evidence in favor of model $\mathcal{M}_2$ over model $\mathcal{M}_1$. A useful scale for interpreting Bayes factors can be found in Kass and Raftery (1995); however, it is important to note that this scale is meaningful only if priors are chosen appropriately (see *Discussion*).

The problem of estimating the number of demes in a structured population can be understood in this light: If we let $\mathcal{M}_K$ denote a genetic mixture model in which K demes are assumed, then the problem of estimating K becomes one of comparing between different models. Ideally we want to solve this problem using the exact model evidence, $\Pr(\mathbf{x}|\mathcal{M}_K)$. Unfortunately, however, calculating the model evidence in complex, multidimensional models is not straightforward, as most of the time we cannot write down the probability of the data under the model without also conditioning on certain known parameters, denoted $\boldsymbol{\theta}$. Obtaining the evidence from the likelihood requires that we integrate over a prior on $\boldsymbol{\theta}$:

$$\Pr(\mathbf{x}|\mathcal{M}_K) = \int_{\boldsymbol{\theta}} \Pr(\mathbf{x}|\boldsymbol{\theta}, \mathcal{M}_K) \Pr(\boldsymbol{\theta}|\mathcal{M}_K) d\boldsymbol{\theta}. \quad (3)$$

It is this integration step that makes calculating the model evidence difficult in practice. In genetic mixture models $\boldsymbol{\theta}$ might represent the allele frequencies in all K demes, perhaps alongside some additional admixture parameters, making the integral in (3) extremely high dimensional (a 100-dimensional integral would not be uncommon). For this reason it makes practical sense to turn to numerical methods or heuristic approximations.

Estimating and approximating the evidence

Perhaps the simplest way of estimating the model evidence is through the harmonic mean estimator, $\hat{h}_K$ (Newton and Raftery 1994),

Figure 1 True and estimated values of the model evidence in log space and in linear space. Error bars give 95% confidence intervals around estimates.

$$\Pr(\mathbf{x}|\mathcal{M}_K) \approx \left[\frac{1}{t} \sum_{m=1}^t \frac{1}{\Pr(\mathbf{x}|\boldsymbol{\theta}_m, \mathcal{M}_K)} \right]^{-1} = \hat{h}_K, \quad (4)$$

where $\boldsymbol{\theta}_m$ for $m \in \{1, \dots, t\}$ denotes a series of draws from the posterior distribution of $\boldsymbol{\theta}$. Part of the appeal of this estimator is its simplicity—it is straightforward to calculate $\hat{h}_K$ from the output of a single MCMC run. As an example, the program Structurama (Huelsenbeck and Andolfatto 2007; Huelsenbeck *et al.* 2011), which contains within it a version of the basic Structure model, has an option for using $\hat{h}_K$ to estimate the model evidence (we note that this is not the primary purpose of Structurama, which also implements a Dirichlet process model). However, in spite of its intuitive appeal, the harmonic mean estimator has been widely criticized due to its instability; $\hat{h}_K$ has been found to be very sensitive to the choice of prior, often being dominated by the reciprocal of a few small values (Neal 1994; Raftery *et al.* 2006).

To avoid some of the problems inherent in the harmonic mean estimator, the approach taken by Pritchard *et al.* (2000) was to define the heuristic estimator L_K (our notation) as

$$-2 \log [\Pr(\mathbf{x}|\mathcal{M}_K)] \approx \hat{\mu} + \frac{\hat{\sigma}^2}{4} = L_K, \quad (5)$$

where $\hat{\mu}$ and $\hat{\sigma}^2$ are simple statistics that can be calculated from the posterior draws (see Supplemental Material, [File S1](#) for a more detailed derivation of this and other statistics). The key assumption that underpins this heuristic is that the posterior deviance is approximately normally distributed, which may or may not be true in practice. L_K is usually evaluated for a range of K , and the smallest L_K (corresponding to the largest evidence) is used as an indication of the most likely model. Alternatively, these values can be transformed out of log space to provide direct estimates of the evidence that, once normalized, can be used to approximate the full posterior distribution of K :

$$\Pr(\mathcal{M}_K|\mathbf{x}) \approx \frac{\exp(-(1/2)L_K)}{\sum_k \exp(-(1/2)L_k)}. \quad (6)$$

This procedure is rarely carried out in practice, despite being recommended in the Structure software documentation (Pritchard *et al.* 2010).

Table 1 Accuracy of estimation methods compared with the exact model evidence

K	−2 log(evidence)						Normalized evidence					
	MSD			MAD			MSD			MAD		
	$\hat{T}_K$	$\hat{h}_K$	L_K	$\hat{T}_K$	$\hat{h}_K$	L_K	$\hat{T}_K$	$\hat{h}_K$	L_K	$\hat{T}_K$	$\hat{h}_K$	L_K
1	0.00e + 00	0.00	28.72	0.00e + 00	0.00	28.72	7.03e-06	−6.50e-03	−2.76e-02	4.59e-05	6.50e-03	2.88e-02
2	−1.54e-03	2.52	42.65	6.99e-03	2.52	42.65	6.67e-07	7.76e-03	4.93e-02	2.37e-04	1.41e-02	6.49e-02
3	−1.95e-03	3.51	46.65	7.71e-03	3.51	46.65	−6.00e-05	3.19e-02	8.96e-02	5.19e-04	3.96e-02	1.37e-01
4	−1.96e-03	3.60	46.74	7.17e-03	3.60	46.74	−8.42e-08	4.07e-02	1.37e-01	6.25e-04	5.54e-02	2.27e-01
5	−1.61e-03	3.37	45.59	6.70e-03	3.37	45.59	−9.85e-06	3.23e-02	1.37e-02	6.09e-04	5.16e-02	1.27e-01
6	−1.39e-03	3.10	44.72	6.73e-03	3.10	44.72	1.59e-05	1.42e-02	−5.46e-02	6.62e-04	4.08e-02	8.33e-02
7	−1.47e-03	2.85	44.78	6.47e-03	2.85	44.78	−7.72e-06	−6.09e-03	−6.99e-02	6.67e-04	3.17e-02	7.97e-02
8	−1.18e-03	2.61	45.11	5.94e-03	2.61	45.11	2.01e-05	−2.56e-02	−6.74e-02	6.28e-04	3.17e-02	8.03e-02
9	−1.21e-03	2.43	45.53	5.99e-03	2.43	45.53	4.17e-05	−4.09e-02	−5.11e-02	6.22e-04	4.23e-02	7.94e-02
10	−1.44e-03	2.26	45.90	5.77e-03	2.26	45.90	−2.17e-05	−5.30e-02	−2.08e-02	5.79e-04	5.31e-02	9.44e-02
Mean	−1.38e-03	2.63	43.64	5.95e-03	2.63	43.64	−1.41e-06	−5.23e-04	−2.12e-04	5.19e-04	3.67e-02	1.00e-01

Shown are mean signed difference (MSD) and mean absolute difference (MAD) of various estimation methods compared with the exact value, obtained by brute force. Formulas for $\hat{T}_K$, $\hat{h}_K$, and L_K are given in Equations 9, 4, and 5, respectively. Values are shown in log space (columns 2–7) and linear space after exponentiating and normalizing to sum to 1 (columns 8–13). Values of K here denote the value used in the inference step, with each row being an average over 1000 simulations (a more detailed breakdown can be found in Table S1).

Thermodynamic integration

The TI estimator differs fundamentally from L_K in the sense that it is not a heuristic estimator—it makes no simplifying assumptions about the distribution of the likelihood. It also differs from $\hat{h}_K$ in that it is well behaved, having finite and quantifiable variance. The approach centers around the “power posterior” (Friel and Pettitt 2008), defined as follows:

$$P_\beta(\boldsymbol{\theta}|\mathbf{x}, \mathcal{M}_K) = \frac{\Pr(\mathbf{x}|\boldsymbol{\theta}, \mathcal{M}_K)^\beta \Pr(\boldsymbol{\theta}|\mathcal{M}_K)}{u(\mathbf{x}|\beta, \mathcal{M}_K)}. \quad (7)$$

This is nothing more than the ordinary posterior distribution of $\boldsymbol{\theta}$, but with the likelihood raised to the power β [the value $u(\mathbf{x}|\beta, \mathcal{M}_K)$ is a normalizing constant that ensures the distribution integrates to 1]. In the same way that we can design an MCMC algorithm to draw from the posterior distribution of $\boldsymbol{\theta}$, we can design a similar algorithm to draw from the power posterior distribution. Details of the MCMC steps are given in the *Appendix* for models both with and without admixture. The resulting draws from the power posterior are written $\boldsymbol{\theta}_m^\beta$, where the superscript β indicates the power used when generating the draws. The TI methodology then proceeds in two simple steps. First, we calculate the mean log-likelihood of the power posterior draws:

$$\hat{D}_\beta = \frac{1}{t} \sum_{m=1}^t \log[\Pr(\mathbf{x}|\boldsymbol{\theta}_m^\beta, \mathcal{M}_K)]. \quad (8)$$

[It is important to note that the notation $\boldsymbol{\theta}_m^\beta$ refers to values drawn from the power posterior with power β ; it does not indicate that the values of $\boldsymbol{\theta}$ (or these likelihoods) are raised to the power β]. This step is repeated for a range of values β_i for $i = \{1, \dots, r\}$ spanning the interval $[0, 1]$. Second, we calculate the area under the curve made by the values $\hat{D}_{\beta_i}$, using a suitable numerical integration scheme, such as the trapezoidal rule:

$$\hat{T}_K = \sum_{i=1}^{r-1} \frac{1}{2} (\hat{D}_{\beta_{i+1}} + \hat{D}_{\beta_i}) (\beta_{i+1} - \beta_i). \quad (9)$$

The value $\hat{T}_K$ is the TI estimator of the model evidence (see [File S1](#) for a more detailed derivation). It can be seen that $\hat{T}_K$ is straightforward to calculate, although it does require us to run multiple MCMC chains to obtain a single estimate of the evidence, making it computationally intensive. On the other hand, the method has greater precision than some alternatives that can be calculated faster. In our comparisons this trade-off was taken into account by using the same number of MCMC iterations for all methods.

Data availability

The authors state that all data necessary for confirming the conclusions presented in the article are represented fully within the article and are available upon request.

Results

Comparison against the exact model evidence

Our first objective was to measure the accuracy and precision of different estimators of the model evidence against the exact value, obtained by brute force (see *Appendix*). The difficulty in calculating the exact model evidence meant that this was possible only for very small simulated data sets of $n = 10$ diploid individuals at $L = 20$ loci, generated from the same without-admixture model implemented in the program Structure2.3.4. A total of 1000 simulated data sets were produced, with K ranging from 1 to 10 (100 simulations each) and with $\lambda_{ij} = 1.0$ for each of $J_i = 5$ alleles (see Table A1 for a list of parameters). Each data set was then analyzed using the program Maverick1.0. This program is written in C++ and was designed specifically to carry out TI for structured populations via the algorithms described in the *Appendix*. In

addition, Maverick1.0 implements certain features that lead to efficient and reliable exploration of the posterior, including solving the label switching problem via the method of Stephens (2000) (see [File S2](#) for further details of the main algorithm). The output of Maverick1.0 includes values of $\hat{h}_K$, L_K , and the TI estimator $\hat{T}_K$. Calculation of L_K was compared extensively against Structure2.3.4 to ensure agreement. For the TI estimator the number of “rungs” used (the value of r) was set to 50, while for $\hat{h}_K$ and L_K the analysis was repeated 50 times to obtain a global mean and standard error over replicates, thereby ensuring that the same computational effort was expended for all methods. A total of 10,000 samples were obtained from the posterior distribution in each MCMC analysis, with a burn-in of 1000 iterations.

Figure 1 shows the results of one such analysis, in which the true number of demes was $K = 2$.

It can be seen that both $\hat{h}_K$ and L_K are negatively biased in this example, leading to estimates of $-2 \times \log(\text{evidence})$ that are smaller than the true value. Any bias that is constant over K goes away after transforming to a linear scale and normalizing; however, $\hat{h}_K$ and particularly L_K still give poor estimates of the true posterior distribution.

The accuracy and precision of the different estimators was evaluated across all 1000 simulated data sets in the form of the mean signed difference (MSD) and the mean absolute difference (MAD). The MSD measures the average difference between the true and estimated values and hence can be considered a measure of bias, while the MAD measures the average *absolute* difference and hence is influenced by both the bias and the precision of the estimator (small values represent estimates that are both accurate and precise). Results are given in Table 1, broken down by the value of K used in the inference step (a more detailed breakdown can be found in [Table S1](#)).

It can be seen that the average MAD of the L_K estimator after normalizing is ~ 0.1 , while the MAD of the $\hat{T}_K$ estimator is 5.19×10^{-4} for the same computational effort. The harmonic mean estimator is intermediate between these values, differing from the true evidence by ~ 0.04 on average. Based on these results we would expect estimates of the posterior distribution of K made using $\hat{h}_K$ or L_K to be qualitatively different from the true posterior distribution.

Accuracy for larger data sets

Although the results in Table 1 are suggestive of a weakness in heuristic estimators, we are limited here to looking at small data sets in which the exact model evidence can be calculated by brute force. It is plausible based on these results that the bias in $\hat{h}_K$ and L_K could be amplified in small data sets due to a lack of information and would cease to be a problem if more data were available. Here we therefore use larger simulated data sets to address the question of whether the TI method produces improvements that would be of practical importance. Although we cannot calculate the true evidence by brute force here, the advantage of using simulated data sets is that we can generate observations from the exact model

Table 2 Percentage times K correctly identified

K	$\hat{T}_K$	$\hat{h}_K$	L_K	Δ_K	AIC	BIC	DIC _s	DIC _G
10 loci								
1	100.0	76.0	0.0	—	88.0	100.0	0.0	100.0
2	100.0	83.0	0.0	100.0	99.0	100.0	0.0	98.0
3	100.0	83.0	0.0	76.0	100.0	100.0	0.0	94.0
4	100.0	77.0	0.0	67.0	95.0	99.0	0.0	89.0
5	100.0	71.0	0.0	58.0	98.0	92.0	0.0	90.0
6	100.0	72.0	1.0	45.0	96.0	75.0	0.0	90.0
7	100.0	65.0	3.0	43.0	96.0	35.0	0.0	94.0
8	100.0	46.0	6.0	42.0	93.0	7.0	0.0	84.0
9	100.0	57.0	17.0	14.0	96.0	1.0	0.0	94.0
10	100.0	100.0	100.0	—	96.0	0.0	100.0	99.0
Mean	100.0	73.0	12.7	55.6	95.7	60.9	10.0	93.2
20 loci								
1	100.0	100.0	0.0	—	95.0	100.0	0.0	89.0
2	100.0	100.0	2.0	100.0	100.0	100.0	0.0	86.0
3	100.0	100.0	64.0	95.0	100.0	100.0	10.0	92.0
4	100.0	99.0	85.0	93.0	100.0	100.0	44.0	97.0
5	100.0	98.0	90.0	97.0	100.0	100.0	70.0	100.0
6	100.0	92.0	88.0	94.0	100.0	100.0	78.0	100.0
7	100.0	94.0	85.0	93.0	100.0	94.0	84.0	99.0
8	100.0	91.0	87.0	97.0	100.0	73.0	78.0	100.0
9	100.0	87.0	82.0	88.0	100.0	26.0	70.0	97.0
10	100.0	100.0	100.0	—	100.0	3.0	100.0	100.0
Mean	100.0	96.1	68.3	94.6	99.5	79.6	53.4	96.0
50 loci								
1	100.0	100.0	45.0	—	100.0	100.0	17.0	25.0
2	100.0	99.0	21.0	100.0	100.0	100.0	100.0	19.0
3	100.0	90.0	30.0	99.0	100.0	100.0	100.0	17.0
4	100.0	97.0	32.0	100.0	100.0	100.0	100.0	23.0
5	100.0	98.0	28.0	100.0	100.0	100.0	100.0	20.0
6	100.0	97.0	42.0	100.0	100.0	100.0	100.0	27.0
7	100.0	98.0	47.0	100.0	100.0	100.0	100.0	30.0
8	100.0	95.0	58.0	99.0	100.0	95.0	100.0	47.0
9	100.0	96.0	63.0	99.0	100.0	83.0	100.0	57.0
10	100.0	100.0	99.0	—	100.0	45.0	100.0	100.0
Mean	100.0	97.0	46.5	99.6	100.0	92.3	91.7	36.5

Shown is the percentage times K is correctly identified by each method, broken down by the value of K used when generating the data. Formulas for $\hat{T}_K$, $\hat{h}_K$, and L_K are given in Equations 9, 4, and 5, respectively, while formulas for AIC, BIC, DIC_s, and DIC_G are given in [File S1](#) Equations 40, 43, 46, and 47, respectively. The formula for Δ_K can be found in [Evanno et al. \(2005\)](#).

used in the inference step and for a *known* value of K . We can then measure the proportion of times that the true value of K is correctly identified. As well as comparing the estimators $\hat{T}_K$, $\hat{h}_K$, and L_K , in which the smallest value of the estimator indicates the most likely model, we also compared Evanno’s Δ_K ([Evanno et al. 2005](#)), in which the largest value indicates the point of maximum curvature of L_K , and the AIC, BIC, and DIC statistics, in which the smallest value indicates the best-fitting model. Values of the DIC were calculated using the method of Spiegelhalter et al. (2002) (DIC_s) as well as the method of Gelman et al. (2004) (DIC_G). To ensure that our results were not driven by a lack of information, larger data sets of $n = 200$ diploid individuals at $L = 10, 20$, and 50 loci were generated from the same without-admixture model used above. As before, 1000 simulated data sets were produced with K ranging from 1 to 10 (100 simulations each). Maverick1.0 was run under the without-admixture model

Figure 2 Estimates of the model evidence for $K = 1 : 20$ obtained using (A) the Structure estimator L_K , (B) the harmonic mean estimator $\hat{h}_K$, and (C) the TI estimator $\hat{T}_K$. For A and B, solid points give the mean over 21 replicates and error bars give 95% confidence intervals calculated from the variance over replicates. For C the TI estimation procedure results in a single point estimate of the evidence and an estimate of the 95% confidence interval without the need to average over replicates.

with 1000 burn-in iterations and 10,000 sampling iterations. For the TI estimator 50 runs were used, and for L_K and $\hat{h}_K$ the analysis was repeated 50 times.

Table 2 gives the proportion of times that the correct value of K was identified by each of the methods. It can be seen that the TI-based method of choosing K provided 100% reliable results across all simulated data sets. Estimates of K based on $\hat{h}_K$ were less reliable, although still reasonable when the number of loci was large, whereas estimates based on L_K were generally not reliable and particularly poor when the number of loci was small. This appears to be due to the well-documented tendency of L_K to continually increase with larger values of K (Pritchard *et al.* 2010), also giving the false impression that L_K is highly accurate when $K = 10$ in this example. Evanno's Δ_K mitigated this to some extent, but still did not provide consistently reliable results (note that Δ_K cannot be calculated on the smallest or largest K in any analysis as a consequence of how it is derived). Of the model comparison statistics the AIC was the most consistently reliable, providing accurate estimates across a range of simulations.

Returning to the question of whether the inaccuracy in $\hat{h}_K$ and L_K in Table 1 was driven by a lack of information, it can be seen from Table 2 that the quantity of data certainly plays a role. However, the fact that TI provides reliable estimates across the range of simulations indicates that there is sufficient signal in the data to detect the value of K even in relatively small data sets. Thus, the increased precision of the TI approach is of practical as well as theoretical importance.

Reanalysis of white-footed mouse data

Our main reason for focusing on simulated data sets above is for the purposes of comparing different statistical methods under very controlled circumstances. By simulating data from the exact model used in the inference step we can tease apart the issue of whether inaccuracies are due to statistical problems or simply a lack of model fit to the data (the latter being ruled out). However, ultimately our interest lies in real-world analyses of population structure. Here the parameter K has a less literal meaning and should be seen as a convenient way of summarizing the structure in the available data, rather than as an exact description of the number of demes.

To test Maverick1.0 in a realistic setting we reanalyzed data from a study by Munshi-South and Kharchenko (2010b), made available through the Dryad digital repository (Munshi-South and Kharchenko 2010a). The data consist of diploid genotypes at 18 putatively neutral microsatellite loci in 312 white-footed mice (*Peromyscus leucopus*), sampled from 15 distinct locations in and around New York City (see the original article for details). White-footed mice are known to be urban adaptors, and so the original study investigated the effects of urbanization and habitat fragmentation on the mouse population, concluding that there has been pervasive genetic differentiation and the emergence of strong population structure. The authors carried out a range of statistical tests, including but not limited to an analysis with Structure2.3 under the admixture model with correlated allele frequencies and with α inferred as part of the MCMC. They explored values of K from 1 to 20 (repeating each analysis 10 times), finding that the mean L_K peaked at $K = 16$ while Evanno's Δ_K had peaks at $K = 6$ and $K = 16$, although generally the distribution of this statistic was complex (see figure 2 in Munshi-South and Kharchenko 2010b).

We carried out a similar analysis in Maverick1.0, using TI to estimate the evidence for K as well as using $\hat{h}_K$ and L_K . We used the same admixture model as in the original study, in which α is inferred as part of the MCMC; however, the correlated allele frequencies model is not implemented in Maverick1.0 and so we assumed a model of independent allele frequencies. For this reason our results are not directly comparable with those of the original study, although our assumptions are broadly similar. We explored K from 1 to 20. When carrying out TI we used $r = 21$ runs, and for the other estimation methods we took the mean and standard error over 21 replicates. For each MCMC analysis we ran 10 chains, each with 10,000 burn-in iterations and 50,000 sampling iterations, before trimming and merging chains to obtain 500,000 sampling iterations (we found that this gave better results than running one long chain).

The results of this analysis are shown in Figure 2. It can be seen that L_K increases smoothly with K , in a trend similar to that found by Munshi-South and Kharchenko (2010b), the difference being that we find no peak at $K = 16$. The harmonic mean estimator increases rapidly until $K = 5$ but at

Figure 3 Posterior assignment of all 312 individuals into $K = 5$ clusters. Site names correspond to locations in and around New York City, and major landmasses are also given (the Black Rock Forest site is not within any of the five New York City boroughs). Further details of sampling sites can be found in Munshi-South and Kharchenko (2010b).

this point saturates and cannot distinguish between higher values of K . In contrast to both of these statistics, the TI estimator has a strong peak at $K = 5$ with narrow confidence intervals. Based on the arguments presented above we conclude that this is the most accurate curve for the model evidence, and so $K = 5$ has the strongest support under this model. The posterior allocation plot for $K = 5$ is shown in Figure 3 (plots for all values of K can be found in Figure S1). Comparing this with figure 3 in Munshi-South and Kharchenko (2010b), we see some striking similarities—for example, the strong population differentiation in the Hunters Island and Willow Lake (a.k.a. Flushing Meadows) samples and the greater uncertainty in samples from the Black Rock Forest location. However, we also group together several populations that were previously found to be distinct, including locations 3, 4, 5, and 7 (all from the Bronx) and locations 8, 9, 11, and 15 (all from central Queens). The fact that we found evidence for fewer distinct populations than the original study may be due to our use of an uncorrelated allele frequencies model, although the geographical proximity of these regions gives us some confidence that this clustering is biologically plausible. Moreover, the striking difference between Figure 2A and Figure 2C demonstrates that different estimation methods can lead to quantitatively different conclusions even conditional on the same underlying model.

Discussion

Model-based clustering methods have proved extremely useful within population genetics. The probabilistic allocation of individuals to demes employed by programs such as Structure has made it possible to tease apart population subdivision within a wide range of organisms, including humans (Rosenberg *et al.* 2002; Li *et al.* 2008; Tishkoff *et al.* 2009), human pathogens (Falush *et al.* 2003b), plants (Garris *et al.* 2005), and animals (Parker *et al.* 2004). However, these posterior assignments are always produced conditional on the known value of K . Choosing an appropriate value of K is statistically much more challenging than estimating population assignments, as it involves a comparison between models rather than simple parameter estimation within a given model. Thermodynamic integration offers a way to do this, providing estimates of the evidence for K that are both accurate and precise. Our reanalysis of the white-footed mouse

data demonstrates that this is of practical as well as theoretical importance, with the potential to lead to quantitatively different conclusions about the data.

The main disadvantage of TI is the computational cost. Multiple MCMC chains are needed, each drawn from a different version of the power posterior, to compute a single estimate of the model evidence. If the number of runs is too low, then the trapezoidal rule step in (9) will not capture the shape of the underlying curve that it is approximating, leading to bias in the estimator. We must also be careful to take account of autocorrelation in the samples. This is dealt with automatically in Maverick1.0 through the use of effective sample size (ESS) calculations (see File S1 for details), which result in estimates of the model evidence that are accurate even in the presence of autocorrelation. However, it is still the case that high levels of autocorrelation require us to obtain a large number of posterior draws, and so we cannot ignore autocorrelation completely. This is a particular problem for the admixture model with α free to vary, where the much higher dimensionality of the model (compared with the without-admixture case) tends to result in poor MCMC mixing.

For this reason, TI may be suitable only for small- to medium-sized data sets of the sort analyzed here, at least for the time being. The use of TI for large SNP data sets—for example, data from the Human Genome Diversity Project (HGP) analyzed by Li *et al.* (2008)—is therefore not practically possible at this stage without devoting significant computational resources to the problem. Good results will tend to be obtained when applied to data sets on the order of hundreds of individuals and tens to hundreds of loci, depending on the parameter set used. Fortunately, the accuracy of some heuristic estimators and traditional model comparison statistics appears to improve for larger data sets, and so it may be possible to sidestep this issue. It is also worth noting that, when genetic markers are sufficiently dense, loci can no longer be considered independent, alternative approaches such as chromosome painting may be more appropriate (Lawson *et al.* 2012).

An important consequence of working with the model evidence is that we must be careful in our choice of priors. In ordinary parameter estimation it is common practice to use relatively uninformative priors—the logic being that the model should be free to be driven by the data and not by our prior assumptions. However, when calculating the evidence (as in

Equation 3), the thinness of the prior has an effect that is not diminished by adding more data. This can result in models being unduly punished if the observed data are extremely unlikely *a priori*. For example, our use of independent Dirichlet priors on the allele frequencies in all populations can be considered a fairly thin prior, as no combination of allele frequencies is any more likely than any other *a priori*. This will tend to result in conservative estimates of K , as there is a large cost (in evidence terms) of adding more populations unless they can justify their existence by a commensurate increase in the likelihood. Alternative model formulations, such as the correlated allele frequencies model of Falush *et al.* (2003a), may therefore be better at detecting subtle signals of population subdivision. This model is likely to feature in later versions of MaverickK.

Finally, it is important to keep in mind that when thinking about population structure, we should not place too much emphasis on any single value of K . The simple models used by programs such as Structure and MaverickK are highly idealized cartoons of real life, and so we cannot expect the results of model-based inference to be a perfect reflection of true population structure (see discussion in Waples and Gaggiotti 2006). Thus, while TI can help ensure that our results are statistically valid conditional on a particular evolutionary model, it can do nothing to ensure that the evolutionary model is appropriate for the data. Similarly—in spite of the results in Table 2—we do not advocate using the model evidence (estimated by TI or any other method) as a way of choosing the single “best” value of K . The chief advantage of the evidence in this context is that it can be used to obtain the complete posterior distribution of K , which is far more informative than any single point estimate. For example, by averaging over the distribution of K , weighted by the evidence, we can obtain estimates of parameters of biological interest (such as the admixture parameter α) without conditioning on a single population structure. Although one value of K may be most likely *a posteriori*, in general a range of values will be plausible, and we should entertain all of these possibilities when drawing conclusions.

The MaverickK program and documentation can be downloaded from www.bobverity.com/MaverickK.

Acknowledgments

We are grateful to Jason Munshi-South and Katerina Kharchenko for making the data from their 2010 white-footed mouse analysis publicly available, to James Borrell for patiently testing early versions of the program, and to three anonymous reviewers whose suggestions substantially improved this article.

Literature Cited

- Baele, G., P. Lemey, T. Bedford, A. Rambaut, M. A. Suchard *et al.*, 2012 Improving the accuracy of demographic and molecular clock model comparison while accommodating phylogenetic uncertainty. *Mol. Biol. Evol.* 29: 2157–2167.
- Beerli, P., and M. Palczewski, 2010 Unified framework to evaluate panmixia and migration direction among multiple sampling locations. *Genetics* 185: 313–326.
- Blanquart, S., and N. Lartillot, 2006 A Bayesian compound stochastic process for modeling nonstationary and nonhomogeneous sequence evolution. *Mol. Biol. Evol.* 23: 2058–2071.
- Corander, J., P. Waldmann, and M. J. Sillanpää, 2003 Bayesian analysis of genetic differentiation between populations. *Genetics* 163: 367–374.
- Evanno, G., S. Regnaut, and J. Goudet, 2005 Detecting the number of clusters of individuals using the software structure: a simulation study. *Mol. Ecol.* 14: 2611–2620.
- Falush, D., M. Stephens, and J. K. Pritchard, 2003a Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics* 164: 1567–1587.
- Falush, D., T. Wirth, B. Linz, J. K. Pritchard, M. Stephens *et al.*, 2003b Traces of human migrations in helicobacter pylori populations. *Science* 299: 1582–1585.
- Falush, D., M. Stephens, and J. K. Pritchard, 2007 Inference of population structure using multilocus genotype data: dominant markers and null alleles. *Mol. Ecol. Notes* 7: 574–578.
- Friel, N., and A. N. Pettitt, 2008 Marginal likelihood estimation via power posteriors. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 70: 589–607.
- Garris, A. J., T. H. Tai, J. Coburn, S. Kresovich, and S. McCouch, 2005 Genetic structure and diversity in *Oryza sativa* L. *Genetics* 169: 1631–1638.
- Gelman, A., and X.-L. Meng, 1998 Simulating normalizing constants: from importance sampling to bridge sampling to path sampling. *Stat. Sci.* 13: 163–185.
- Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin, 2004 *Bayesian Data Analysis*, 2nd edition. Chapman & Hall/CRC, Boca Raton, Florida.
- Hubisz, M. J., D. Falush, M. Stephens, and J. K. Pritchard, 2009 Inferring weak population structure with the assistance of sample group information. *Mol. Ecol. Resour.* 9: 1322–1332.
- Huelsenbeck, J. P., and P. Andolfatto, 2007 Inference of population structure under a Dirichlet process model. *Genetics* 175: 1787–1802.
- Huelsenbeck, J. P., P. Andolfatto, and E. T. Huelsenbeck, 2011 Structurama: Bayesian inference of population structure. *Evol. Bioinform. Online* 7: 55.
- Jombart, T., and C. Collins, 2015 A tutorial for discriminant analysis of principal components (dapc) using adegenet 2.0.0. Available at: <http://adegenet.r-forge.r-project.org/files/tutorial-dapc.pdf>.
- Kass, R. E., and A. E. Raftery, 1995 Bayes factors. *J. Am. Stat. Assoc.* 90: 773–795.
- Lartillot, N., and H. Philippe, 2006 Computing Bayes factors using thermodynamic integration. *Syst. Biol.* 55: 195–207.
- Lawson, D. J., G. Hellenthal, S. Myers, and D. Falush, 2012 Inference of population structure using dense haplotype data. *PLoS Genet.* 8: e1002453.
- Lepage, T., D. Bryant, H. Philippe, and N. Lartillot, 2007 A general comparison of relaxed molecular clock models. *Mol. Biol. Evol.* 24: 2669–2680.
- Li, J. Z., D. M. Absher, H. Tang, A. M. Southwick, A. M. Casto *et al.*, 2008 Worldwide human relationships inferred from genome-wide patterns of variation. *Science* 319: 1100–1104.
- Malécot, G., 1948 *Mathématiques de l'Hérédité*. Masson & Cie, Paris.
- Munshi-South, J., and K. Kharchenko, 2010a Data from: Rapid, pervasive genetic differentiation of urban white-footed mouse (*Peromyscus leucopus*) populations in New York City. Dryad digital repository. Available at: <http://dx.doi.org/10.5061/dryad.1893.10.5061/dryad.1893>
- Munshi-South, J., and K. Kharchenko, 2010b Rapid, pervasive genetic differentiation of urban white-footed mouse (*Peromyscus*

- leucopus) populations in New York City. *Mol. Ecol.* 19: 4242–4254.
- Neal, R. M., 1994 Contribution to the discussion of “Approximate Bayesian inference with the weighted likelihood bootstrap” by Michael A. Newton and Adrian E. Raftery. *J. R. Stat. Soc. B* 56: 41–42.
- Newton, M. A., and A. E. Raftery, 1994 Approximate Bayesian inference with the weighted likelihood bootstrap. *J. R. Stat. Soc. B* 56: 3–48.
- Parker, H. G., L. V. Kim, N. B. Sutter, S. Carlson, T. D. Lorentzen *et al.*, 2004 Genetic structure of the purebred domestic dog. *Science* 304: 1160–1164.
- Pella, J., and M. Masuda, 2006 The Gibbs and split merge sampler for population mixture analysis from genetic data with incomplete baselines. *Can. J. Fish. Aquat. Sci.* 63: 576–596.
- Pritchard, J. K., M. Stephens, and P. Donnelly, 2000 Inference of population structure using multilocus genotype data. *Genetics* 155: 945–959.
- Pritchard, J. K., X. Wen, and D. Falush, 2010 Documentation for structure software: version 2.3. Available at: http://pritchardlab.stanford.edu/structure_software/release_versions/v2.3.4/structure_doc.pdf.
- Raftery, A. E., M. A. Newton, J. M. Satagopan, and P. N. Krivitsky, 2006 Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. *Bayes. Stat.* 8: 1–45.
- Raj, A., M. Stephens, and J. K. Pritchard, 2014 Faststructure: variational inference of population structure in large SNP data sets. *Genetics* 197: 573–589.
- Rannala, B., and J. L. Mountain, 1997 Detecting immigration by using multilocus genotypes. *Proc. Natl. Acad. Sci. USA* 94: 9197–9201.
- Rosenberg, N. A., J. K. Pritchard, J. L. Weber, H. M. Cann, K. K. Kidd *et al.*, 2002 Genetic structure of human populations. *Science* 298: 2381–2385.
- Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. Van Der Linde, 2002 Bayesian measures of model complexity and fit. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 64: 583–639.
- Stephens, M., 2000 Dealing with label switching in mixture models. *J. R. Stat. Soc. Ser. B Stat. Methodol.* 62: 795–809.
- Tishkoff, S. A., F. A. Reed, F. R. Friedlaender, C. Ehret, A. Ranciaro *et al.*, 2009 The genetic structure and history of Africans and African Americans. *Science* 324: 1035–1044.
- Waples, R. S., and O. Gaggiotti, 2006 Invited review: What is a population? An empirical evaluation of some genetic methods for identifying the number of gene pools and their degree of connectivity. *Mol. Ecol.* 15: 1419–1439.
- Wright, S., 1949 The genetical structure of populations. *Ann. Eugen.* 15: 323–354.

Communicating editor: N. A. Rosenberg

Appendix

MCMC Under the Without-Admixture Model

To carry out the TI estimation approach we need to be able to draw from the power posterior distribution. This is straightforward in the case of genetic mixtures and requires nothing more than a simple extension of existing MCMC algorithms. In the following we strive to bring our notation in line with previous studies wherever possible, but the complexities of certain likelihood functions also motivate us to define some new notation (see Table A1). It is worth noting, for example, that we will write individual genotypes in simple list form (as in Pritchard *et al.* 2000), using the notation $\mathbf{x}_{il}$ for the l th locus of the i th individual, but also in allelic partition form (as in Huelsenbeck and Andolfatto 2007), using the notation $\mathbf{s}_{il}$. For example, a diploid individual homozygous for the third allele at a particular locus can be written $\mathbf{x}_{il} = (3, 3)$ or equivalently $\mathbf{s}_{il} = \{0, 0, 2, 0, 0\}$, where there are five possible alleles to choose from in this example. Conditioning on the model $\mathcal{M}_K$ is also implicit throughout this section.

In the basic algorithm described by Pritchard *et al.* (2000) there are two free parameters to keep track of—the allocation of individuals to demes, denoted $\mathbf{z}$ here, and the allele frequencies in all K demes, denoted $\mathbf{p}$. Under the assumptions of Hardy–Weinberg and linkage equilibrium it is possible to write the probability of the observed data given the known values of these free parameters, $\Pr(\mathbf{x}|\mathbf{z}, \mathbf{p})$. Combining this likelihood with a Dirichlet($\lambda_{l1}, \dots, \lambda_{lL_l}$) prior on the allele frequencies at each locus, we can derive the conditional posterior distribution of the allele frequencies given the known group allocation, $\Pr(\mathbf{p}|\mathbf{x}, \mathbf{z})$. Alternatively, multiplying by an equal $1/K$ prior on the allocation of individuals to demes, we can derive the conditional posterior distribution of the group allocation given the known allele frequencies, $\Pr(\mathbf{z}|\mathbf{x}, \mathbf{p})$. Algorithm 1 of Pritchard *et al.* (2000) works by alternately sampling from each of these conditional distributions, resulting (after sufficient burn-in) in a series of draws from the full posterior distribution. More often than not we are interested in the posterior allocation, in which case the posterior allele frequencies can simply be ignored.

However, as stated in the original derivation of Rannala and Mountain (1997) and reiterated by later authors (Corander *et al.* 2003; Pella and Masuda 2006; Huelsenbeck and Andolfatto 2007), it is possible to integrate over the allele frequencies analytically, thereby greatly reducing the dimensionality of the problem. The new likelihood, conditional only on the group allocation, can be written

$$\Pr(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^K \prod_{l=1}^L \frac{\Gamma(\lambda_{l0})}{\Gamma(\lambda_{l0} + y_{kl0})} \prod_{j=1}^{J_l} \frac{\Gamma(\lambda_{lj} + y_{klj})}{\Gamma(\lambda_{lj})} \quad (\text{A1})$$

(see Table A1 for parameter definitions). This expression is extremely useful to us, as it means the likelihood can be calculated without having to take into account an explicit representation of the unknown allele frequencies—our uncertainty in the allele frequencies has already been integrated out of the problem.

Rather than using (A1) directly, later authors including Corander *et al.* (2003), Pella and Masuda (2006), and Huelsenbeck and Andolfatto (2007) used this analytical solution to define an efficient MCMC algorithm. Dividing the probability of the data $\mathbf{x}$ by the probability of the data with the i th observation removed, denoted $\mathbf{x}^{(-i)}$, we obtain the conditional probability of observation i given all others. Using the fact that $\mathbf{y}_{kl} = \mathbf{y}_{kl}^{(-i)} + \mathbf{s}_{il}$, we obtain

$$\Pr(\mathbf{x}_i | z_i = k, \mathbf{y}_k^{(-i)}) = \prod_{l=1}^L \frac{\Gamma(\lambda_{l0} + y_{kl0}^{(-i)})}{\Gamma(\lambda_{l0} + y_{kl0}^{(-i)} + s_{il0})} \prod_{j=1}^{J_l} \frac{\Gamma(\lambda_{lj} + y_{klj}^{(-i)} + s_{ilj})}{\Gamma(\lambda_{lj} + y_{klj}^{(-i)})}. \quad (\text{A2})$$

Computing (A2) for all k and normalizing, we obtain the conditional posterior probability that individual i belongs to deme k :

$$\Pr(z_i = k | \mathbf{x}_i, \mathbf{y}_k^{(-i)}) = \frac{(1/K) \Pr(\mathbf{x}_i | z_i = k, \mathbf{y}_k^{(-i)})}{\sum_{u=1}^K (1/K) \Pr(\mathbf{x}_i | z_i = u, \mathbf{y}_u^{(-i)})}. \quad (\text{A3})$$

By repeatedly drawing new group allocations for all individuals from (A3), we obtain a series of draws from the posterior distribution without ever needing to invoke the unknown allele frequencies. Thus, the two-step algorithm of Pritchard *et al.* (2000) can be reduced to the more efficient one-step algorithm of Corander *et al.* (2003).

The reason these results are pertinent to our problem is that we can make use of the same gains in efficiency when designing an MCMC algorithm for the purposes of TI. In fact, the only difference when carrying out TI is that the likelihood in (A1) should be raised to the power β , allowing us to draw from the power posterior. On making this change we find that the conditional

posterior distribution in (A2) should also be raised to the power β [this follows from the fact that (A2) can be derived as a ratio of two ordinary likelihoods]. Thus, we arrive at a new expression for the probability of individual i being assigned to group k :

$$P_\beta(z_i = k | \mathbf{x}_i, \mathbf{y}_k^{(-i)}) = \frac{(1/K) \Pr(\mathbf{x}_i | z_i = k, \mathbf{y}_k^{(-i)})^\beta}{\sum_{u=1}^K (1/K) \Pr(\mathbf{x}_i | z_i = u, \mathbf{y}_u^{(-i)})^\beta}. \quad (\text{A4})$$

By repeatedly sampling new group allocations for all individuals from (A4), we obtain a series of allocation vectors drawn from the power posterior (note that when $\beta = 0$, we are essentially drawing from the prior). The likelihood of each vector can then be computed using (A1), at which point we have everything we need to calculate $\hat{D}_\beta$ as in (8). Carrying out this entire procedure for a range of values β_i , we obtain a series of points $\hat{D}_{\beta_i}$ that can be used to calculate the TI estimator $\hat{T}_K$, as in (9). The complete TI algorithm for the model without admixture can be defined as follows:

Algorithm 1 (without admixture)

1. For r distinct values of β_i spanning the interval $[0, 1]$
 - a. Perform MCMC by repeatedly drawing from (A4) for all $i \in \{1, \dots, n\}$. This results (after discarding burn-in) in t draws from the power posterior group allocation.
 - b. Calculate the likelihood of each group allocation, using (A1).
 - c. Calculate $\hat{D}_{\beta_i}$ as the average log-likelihood, as in (8). If calculating the variance of the estimator, calculate $\hat{V}_{\beta_i}$ using the formula in [File S1](#), taking care to use an appropriate value of the ESS.
2. Use the values $\hat{D}_{\beta_i}$ to calculate $\hat{T}_K$ in a suitable numerical integration scheme, for example using the trapezoidal rule as in (9).

MCMC Under the Admixture Model

The model with admixture described by Pritchard *et al.* (2000) is slightly complicated by the fact that each gene copy is free to originate from a different deme. However, we can still apply the same basic logic described above to arrive at a simple one-step algorithm for sampling from the power posterior. First, we note that the probability of the data conditional on the known group allocation is identical in this model to the probability in the without-admixture model and is given by (A1). This is true because we make the same assumption that gene copies are drawn independently from demes, and we apply the same Dirichlet priors on allele frequencies, meaning the final likelihood does not change. The difference in the admixture model is that the group allocation takes place at the level of the gene copy, rather than at the level of the individual, and so the values z_{ila} are no longer restricted to being the same for all (l, a) . This is reflected in the $\mathbf{y}_k$ values used to keep track of the gene copies allocated to a particular deme, which are now free to contain only a partial contribution of the genome of each individual.

Following the same approach as for the without-admixture model, we can obtain the conditional probability of gene copy x_{ila} by dividing through the probability of the complete data by the probability of the data with this element removed [denoted $\mathbf{x}^{(-ila)}$]. Most of the terms in the resulting expression cancel out, leading to the following simple result:

$$\Pr(x_{ila} | z_{ila} = k, \mathbf{y}_k^{(-ila)}) = \left(\frac{\lambda_{lx_{ila}} + y_{klx_{ila}}^{(-ila)}}{\lambda_{l0} + y_{kl0}^{(-ila)}} \right). \quad (\text{A5})$$

As before, this likelihood should be combined with the prior probability of assignment to each deme. If the admixture proportions for individual i are given by the vector $\mathbf{q}_i$, then, under the assumptions of the model described by Pritchard *et al.* (2000), the number of gene copies in this individual that are allocated to each deme can be considered a multinomial draw from $\mathbf{q}_i$. Integrating over a Dirichlet($\alpha, \dots, \alpha$) prior on these frequencies, we obtain

$$\Pr(\mathbf{z} | \mathbf{v}, \alpha) = \prod_{i=1}^n \frac{\Gamma(K\alpha)}{\Gamma(K\alpha + v_{i0})} \prod_{k=1}^K \frac{\Gamma(\alpha + v_{ik})}{\Gamma(\alpha)}. \quad (\text{A6})$$

We can use this expression to write down the prior probability of gene copy a at locus l in individual i being allocated to deme k , conditional on the allocations of all other gene copies:

$$\Pr(z_{ila} = k \mid \mathbf{v}_i^{(-ila)}, \alpha) = \left(\frac{\alpha + v_{ik}^{(-ila)}}{K\alpha + v_{i0}^{(-ila)}} \right). \quad (\text{A7})$$

Bringing together the prior with the likelihood raised to the power β , we obtain the following expression for the power posterior probability of an individual gene copy being allocated to deme k :

$$P_\beta(z_{ila} = k \mid x_{ila}, \mathbf{y}_k^{(-ila)}, \mathbf{v}_i^{(-ila)}, \alpha) = \frac{\Pr(z_{ila} = k \mid \mathbf{v}_i^{(-ila)}, \alpha) \Pr(x_{ila} \mid z_{ila} = k, \mathbf{y}_k^{(-ila)})^\beta}{\sum_{u=1}^K \Pr(z_{ila} = u \mid \mathbf{v}_i^{(-ila)}, \alpha) \Pr(x_{ila} \mid z_{ila} = u, \mathbf{y}_u^{(-ila)})^\beta}. \quad (\text{A8})$$

By repeatedly sampling new allocations for all gene copies at all loci within all individuals (*i.e.*, all z_{ila}), we obtain a series of draws from the power posterior group allocation under the admixture model. Again, this algorithm is made more efficient by the fact that the unknown allele frequencies in all populations *and* the unknown admixture proportions in all individuals have been integrated out of the problem at an early stage.

A common extension to the basic admixture model is to leave α as a free parameter, updating it as part of the MCMC. This can be accommodated within the TI framework by using a simple Metropolis–Hastings step. If α' is a new value of α , drawn from some suitable proposal distribution $g(\alpha' \mid \alpha)$, then the acceptance probability under Metropolis–Hastings is given by

$$\Pr(\alpha \rightarrow \alpha') = \min \left(1, \frac{\Pr(\mathbf{z} \mid \mathbf{v}, \alpha') g(\alpha \mid \alpha')}{\Pr(\mathbf{z} \mid \mathbf{v}, \alpha) g(\alpha' \mid \alpha)} \right). \quad (\text{A9})$$

Note that the core probability that drives this expression is the *prior* probability of the allocation $\mathbf{z}$, which is given in (A6). The actual probability of the data—*i.e.*, the expression that is raised to the power β in the power posterior calculation—does not feature here. Thus, we can use the same Metropolis–Hastings step to update α irrespective of the value of β .

The complete TI algorithm for the model with admixture can be defined as follows:

Algorithm 2 (with admixture)

1. For r distinct values of β_i spanning the interval $[0, 1]$
 - a. Perform MCMC by repeatedly drawing from (A8) for all gene copies at all loci in all individuals (all a, l, i). If α is a free parameter, then update this value using a Metropolis–Hastings step, as in (A9). This results (after discarding burn-in) in t draws from the power posterior group allocation.
 - b. Calculate the likelihood of each group allocation, using (A1).
 - c. Calculate $\hat{D}_{\beta_i}$ as the average log-likelihood, as in (8). If calculating the variance of the estimator, calculate $\hat{V}_{\beta_i}$ using the formula in [File S1](#), taking care to use an appropriate value of the ESS.
2. Use all the values $\hat{D}_{\beta_i}$ to calculate $\hat{T}_K$ in a suitable numerical integration scheme, for example using the trapezoidal rule as in (9).

Finally, we note that the expressions derived in this section can be used to obtain the exact model evidence by brute force in restricted settings. For example, focusing on the model without admixture, we could sum over the likelihood of all possible group allocations to obtain the true model evidence,

$$\Pr(\mathbf{x}) = \sum_{\mathbf{z}} \Pr(\mathbf{x} \mid \mathbf{z}) \Pr(\mathbf{z}), \quad (\text{A10})$$

where $\Pr(\mathbf{x} \mid \mathbf{z})$ is given by (A1), and for this model $\Pr(\mathbf{z}) = 1/K^n$ for all group allocations. Although this is possible in theory, the sheer number of allocations that are required to sum over makes this method impractical in all but the simplest situations. Even if we exploit redundancies in the labeling of different allocations, we are still restricted to values of n and K not much > 10 . This method is therefore only really useful as a way of checking the accuracy of other estimation methods.

Table A1 Definitions of parameters used in this study

Parameter	Description
α	Dirichlet parameter on admixture proportions
β	Power used in power posterior calculation
$c^{(-i)}$	Evaluation of a parameter while excluding information for individual i (c could be any of the parameters listed above)
$c^{(-l/p)}$	Evaluation of a parameter while excluding information for gene copy a at locus l in individual i (c could be any of the parameters listed here)
J_l	No. of unique alleles observed at locus l
K	No. of populations
L	No. of loci
λ_{lj}	Dirichlet parameter for frequency of allele j at locus l [where $j \in (1, \dots, J_l)$]
λ_{l0}	Sum of the Dirichlet parameters for locus l [i.e., $\lambda_{l0} = \sum_{j=1}^{J_l} \lambda_{lj}$]
n	No. of individuals sampled
$\mathbf{p}$	Allele frequencies in all populations at all loci
$\mathbf{q}_i$	Admixture proportions in individual i
$\mathbf{s}_{il}$	Allelic partition of alleles in individual i at locus l . For example, the genotype $\mathbf{x}_{il} = (3, 3)$ can be written $\mathbf{s}_{il} = \{0, 0, 2, 0, 0\}$ in allelic partition form, where in this example $J_l = 5$
s_{ilj}	No. of copies of allele j at locus l in individual i [where $j \in (1, \dots, J_l)$]
s_{il0}	No. of copies of any allele at locus l in individual i [i.e., $s_{il0} = \sum_{j=1}^{J_l} s_{ilj}$]
$\Gamma(\cdot)$	Denotes the gamma function.
$\mathbf{v}$	Partition of gene copies to populations in all individuals
$\mathbf{v}_i$	Partition of gene copies to populations in individual i
v_{ik}	No. of gene copies in individual i assigned to population k
v_{i0}	No. of gene copies in individual i assigned to any population [i.e., $v_{i0} = \sum_{k=1}^K v_{ik}$]
$\mathbf{x}$	Genetic information for all individuals
$\mathbf{x}_i$	Genetic information for individual i
$\mathbf{x}_{il}$	Genetic information for individual i at locus l
x_{ila}	Allelic type of the a th gene copy in individual i at locus l [where $x_{ila} \in (1, \dots, J_l)$]
$\mathbf{y}_k$	Allelic partition at all loci of all gene copies assigned to population k
$\mathbf{y}_{kl}$	Allelic partition at locus l of all gene copies assigned to population k
y_{klj}	No. of copies of allele j at locus l assigned to population k
y_{kl0}	No. of copies of any allele at locus l assigned to population k [i.e., $y_{kl0} = \sum_{j=1}^{J_l} y_{klj}$]
$\mathbf{z}$	Assignment of all gene copies in all individuals
z_{ila}	Assignment of gene copy a at locus l in individual i to a population [where $z_{ila} \in (1, \dots, K)$]
z_i	Assignment of individual i to a population. When referring to z_i it is implied that z_{ila} is identical for all (l, a) , meaning all gene copies within this individual are assigned together

GENETICS

Supporting Information

www.genetics.org/lookup/suppl/doi:10.1534/genetics.115.180992/-/DC1

Estimating the Number of Subpopulations (K) in Structured Populations

Robert Verity and Richard A. Nichols

Figure S1. Posterior assignment plots for all 312 individuals from the Munshi-South and Kharchenko (2010b) data described in the main paper. Results are shown for K=2:20.

Table S1. Mean signed difference and mean absolute difference for all values of K explored in the first simulated data analysis. (.xlsx, 59 KB)

Available for download as a .xlsx file at:

<http://www.genetics.org/lookup/suppl/doi:10.1534/genetics.115.180992/-/DC1/TableS1.xlsx>

File S1: Derivation of Evidence Estimators and Model Comparison Statistics

Supplementary material for the paper *Estimating K in
Genetic Mixture Models*

Robert Verity, Richard A. Nichols

Contents

1	Problem Specification	2
2	Direct Estimators of the Model Evidence	2
2.1	The Harmonic Mean Estimator	2
2.2	Thermodynamic Integration	3
3	Approximations to the Model Evidence	5
3.1	The ‘Structure estimator’ L_K	5
3.2	The BIC	6
4	General Model Comparison Statistics	10
4.1	The AIC	10
4.2	The BIC (again)	11
4.3	The DIC	11

1 Problem Specification

The purpose of this supplementary text is to gather together in one place the mathematical derivation of the various estimators and model comparison statistics that are implemented in the main text. We present the rationale for each method in a manner that emphasises the common underpinnings that connect the different approaches.

From a notational point of view let us assume a data vector $\mathbf{x}$, a model $\mathcal{M}$, and a parameter vector $\boldsymbol{\theta}$ of length d (i.e. there are d free parameters under the model). The likelihood under the model can be written $\Pr(\mathbf{x} \mid \boldsymbol{\theta}, \mathcal{M})$, and the prior can be written $\Pr(\boldsymbol{\theta} \mid \mathcal{M})$. The model evidence can be obtained from the likelihood and the prior as follows:

$$\Pr(\mathbf{x} \mid \mathcal{M}) = \int_{\boldsymbol{\theta}} \Pr(\mathbf{x} \mid \boldsymbol{\theta}, \mathcal{M}) \Pr(\boldsymbol{\theta} \mid \mathcal{M}) d\boldsymbol{\theta} . \quad (1)$$

Some of the model comparison statistics described below can be considered direct estimators of, or approximations to, this quantity. Others are derived from entirely different perspectives.

2 Direct Estimators of the Model Evidence

2.1 The Harmonic Mean Estimator

The harmonic mean estimator can be derived from Bayes' rule, which when written in reverse reads

$$\frac{\Pr(\mathbf{x} \mid \boldsymbol{\theta}, \mathcal{M}) \Pr(\boldsymbol{\theta} \mid \mathcal{M})}{\Pr(\mathbf{x} \mid \mathcal{M})} = \Pr(\boldsymbol{\theta} \mid \mathbf{x}, \mathcal{M}) . \quad (2)$$

Dividing both sides of (2) by the likelihood $\Pr(\mathbf{x} \mid \boldsymbol{\theta}, \mathcal{M})$ and integrating over $\boldsymbol{\theta}$ we obtain

$$\frac{1}{\Pr(\mathbf{x} \mid \mathcal{M})} = \int_{\boldsymbol{\theta}} \frac{\Pr(\boldsymbol{\theta} \mid \mathbf{x}, \mathcal{M})}{\Pr(\mathbf{x} \mid \boldsymbol{\theta}, \mathcal{M})} d\boldsymbol{\theta} . \quad (3)$$

Replacing the integral on the RHS of (3) by its Monte Carlo estimator we obtain

$$\frac{1}{\Pr(\mathbf{x} \mid \mathcal{M})} \approx \frac{1}{t} \sum_{m=1}^t \frac{1}{\Pr(\mathbf{x} \mid \boldsymbol{\theta}_m, \mathcal{M})} , \quad (4)$$

where $\boldsymbol{\theta}_m$ for $m \in \{1, \dots, t\}$ represents a series of draws from the posterior distribution of $\boldsymbol{\theta}$. Taking the reciprocal of (4) we arrive at the harmonic

mean estimator, $\hat{h}$:

$$\hat{h} = \frac{1}{\frac{1}{t} \sum_{m=1}^t \frac{1}{\Pr(\mathbf{x} | \boldsymbol{\theta}_m, \mathcal{M})}} , \quad (5)$$

as defined by Newton and Raftery (1994). Although elegant in its simplicity, and technically unbiased, this estimator is highly unstable, often having infinite variance. See Raftery et al. (2006) for constructive suggestions on how to avoid some of the problems inherent in this estimator.

2.2 Thermodynamic Integration

Thermodynamic Integration (TI) provides direct estimates of the model evidence that are unbiased and have finite and quantifiable variance. The method exploits the ‘power posterior’ (Friel and Pettitt, 2008), defined as follows:

$$P_\beta(\boldsymbol{\theta} | \mathbf{x}, \mathcal{M}) = \frac{\Pr(\mathbf{x} | \boldsymbol{\theta}, \mathcal{M})^\beta \Pr(\boldsymbol{\theta} | \mathcal{M})}{u(\mathbf{x} | \beta, \mathcal{M})} . \quad (6)$$

where $u(\mathbf{x} | \beta)$ is the following normalising constant:

$$u(\mathbf{x} | \beta, \mathcal{M}) = \int_{\boldsymbol{\theta}} \Pr(\mathbf{x} | \boldsymbol{\theta}, \mathcal{M})^\beta \Pr(\boldsymbol{\theta} | \mathcal{M}) d\boldsymbol{\theta} , \quad (7)$$

(in subsequent expressions the conditioning on the model $\mathcal{M}$ will be suppressed). The crucial step in the TI method is the following derivation:

$$\begin{aligned} \frac{d}{d\beta} \log[u(\mathbf{x} | \beta)] &= \frac{1}{u(\mathbf{x} | \beta)} \frac{d}{d\beta} u(\mathbf{x} | \beta) , \\ &= \frac{1}{u(\mathbf{x} | \beta)} \int_{\boldsymbol{\theta}} \frac{d}{d\beta} \Pr(\mathbf{x} | \boldsymbol{\theta})^\beta \Pr(\boldsymbol{\theta}) d\boldsymbol{\theta} , \\ &= \frac{1}{u(\mathbf{x} | \beta)} \int_{\boldsymbol{\theta}} \Pr(\mathbf{x} | \boldsymbol{\theta})^\beta \log[\Pr(\mathbf{x} | \boldsymbol{\theta})] \Pr(\boldsymbol{\theta}) d\boldsymbol{\theta} , \\ &= \int_{\boldsymbol{\theta}} \log[\Pr(\mathbf{x} | \boldsymbol{\theta})] \frac{\Pr(\mathbf{x} | \boldsymbol{\theta})^\beta \Pr(\boldsymbol{\theta})}{u(\mathbf{x} | \beta)} d\boldsymbol{\theta} , \\ &= \int_{\boldsymbol{\theta}} \log[\Pr(\mathbf{x} | \boldsymbol{\theta})] P_\beta(\boldsymbol{\theta} | \mathbf{x}) d\boldsymbol{\theta} , \\ &= \mathbb{E}_{\boldsymbol{\theta} | \mathbf{x}, \beta} [\log[\Pr(\mathbf{x} | \boldsymbol{\theta})]] . \end{aligned} \quad (8)$$

In other words, the gradient of $\log[u(\mathbf{x} | \beta)]$ is equivalent to the expected log-likelihood of the parameter $\boldsymbol{\theta}$, where the expectation is taken over the power posterior. It is easy to demonstrate that the integral of (8) with respect to β over the range $[0, 1]$ is equal to the logarithm of the model evidence:

$$\begin{aligned} \int_0^1 \frac{d}{d\beta} \log[u(\mathbf{x} | \beta)] d\beta &= \log[u(\mathbf{x} | \beta=1)] - \log[u(\mathbf{x} | \beta=0)] , \\ &= \log[\text{Pr}(\mathbf{x})] , \end{aligned} \quad (9)$$

where we have made use of the fact that $u(\mathbf{x} | \beta=1)$ is equal to the model evidence, and $u(\mathbf{x} | \beta=0)$ is equal to the integral over the prior, which is 1 by definition. Therefore *if* we could calculate the expectation in (8) and *if* we could integrate this expectation over $\beta \in [0, 1]$ then we would arrive at the logarithm of the model evidence.

Thermodynamic integration is essentially a numerical approximation to this scheme. Let $\boldsymbol{\theta}_m^\beta$ for $m \in \{1, \dots, t\}$ represent a series of independent draws from the power posterior with power β . Then the expectation in (8) can be estimated using the Monte Carlo estimator $\widehat{D}_\beta$, defined as follows:

$$\widehat{D}_\beta = \frac{1}{t} \sum_{m=1}^t \log [\text{Pr}(\mathbf{x} | \boldsymbol{\theta}_m^\beta)] . \quad (10)$$

Turning to the integral in (9), we cannot carry out this integration step analytically, but we can approximate it using the values $\widehat{D}_\beta$ in a simple numerical integration technique, such as the trapezoidal rule. For example, if the values $\beta_i = (i-1)/(r-1)$ for $i = \{1, \dots, r\}$ represent a series of equally spaced powers spanning the interval $[0, 1]$, where r denotes the number of ‘rungs’ used ($r \geq 2$), then the integral in (9) can be approximated using

$$\begin{aligned} \widehat{T} &= \sum_{i=1}^{r-1} \frac{\frac{1}{2}(\widehat{D}_{\beta_{i+1}} + \widehat{D}_{\beta_i})}{r-1} , \\ &= \frac{1}{r-1} \left(\frac{1}{2} \widehat{D}_{\beta_1} + \frac{1}{2} \widehat{D}_{\beta_r} + \sum_{i=2}^{r-1} \widehat{D}_{\beta_i} \right) . \end{aligned} \quad (11)$$

There are two sources of error in this estimator. First, there is ordinary statistical error associated with replacing (8) by its Monte Carlo estimator, and second, there is discretisation error caused by replacing a continuous integral with a numerical approximation. Both of these sources of error are quantified in Lartillot and Philippe (2006). In the example above the sampling variance of the estimator can be calculated using

$$\text{Var}[\widehat{T}] = \frac{1}{(r-1)^2} \left(\frac{1}{4} \text{Var}[\widehat{D}_{\beta_1}] + \frac{1}{4} \text{Var}[\widehat{D}_{\beta_r}] + \sum_{i=2}^{r-1} \text{Var}[\widehat{D}_{\beta_i}] \right) , \quad (12)$$

where the sampling variance of individual elements is obtained by dividing the variance by the MCMC effective sample size (ESS):

$$\text{Var}[\hat{D}_\beta] \approx \frac{\hat{V}_\beta}{\text{ESS}}, \quad (13)$$

$$\hat{V}_\beta = \frac{1}{t-1} \sum_{m=1}^t \left(\log [\text{Pr}(\mathbf{x} | \boldsymbol{\theta}_m^\beta)] - \hat{D}_\beta \right)^2. \quad (14)$$

The ESS is a crucial quantity in this expression. MCMC draws are often highly auto-correlated, meaning the effective number of independent samples is often much smaller than the number of samples. In order to obtain reliable estimates of the variance in $\hat{T}$ we must take this autocorrelation into account. A standard method is to calculate the ESS directly from the autocorrelation of the MCMC chain:

$$\text{ESS} = \frac{t}{1 + 2 \sum_{l=1}^{\infty} \rho_l}, \quad (15)$$

where ρ_l gives the autocorrelation in the log-likelihood at l lags. This method is implemented within MAVERICK1.0, meaning autocorrelation is automatically taken into account in estimates of the model evidence.

The sampling variance of $\hat{D}_\beta$ – and therefore the sampling variance of the final estimator $\hat{T}$ – can be reduced by increasing the number of effectively independent draws from the posterior. This involves either increasing the raw number of draws, or improving the mixing of the MCMC to achieve greater independence. The discretisation error can also be kept within bounds. As noted by Friel et al. (2013), the curve traced by $\text{E}_{\boldsymbol{\theta}|\mathbf{x},\beta}[\log[\text{Pr}(\mathbf{x} | \boldsymbol{\theta})]]$ is always-increasing with β , meaning there are strict upper and lower bounds on the error that is possible given a series of known values distributed along this curve (and ignoring interactions between statistical error and discretisation error). These bounds can be quantified, leading to an estimate of the worst-case scenario discretisation error. In practice we find that these estimates are too pessimistic to be useful, and prefer to simply examine the shape of the estimated curve to ensure that sufficient rungs have been used to capture the curvature.

3 Approximations to the Model Evidence

3.1 The ‘Structure estimator’ L_K

The heuristic estimator L_K (our notation), described by Pritchard et al. (2000) and implemented in the program STRUCTURE, is closely connected to

the harmonic mean estimator, but designed to avoid some of the problems inherent in the original method. As with many model comparison statistics, it can be derived from the *Bayesian deviance*, defined as -2 times the log-likelihood:

$$D(\boldsymbol{\theta}) = -2 \log [\Pr(\mathbf{x} | \boldsymbol{\theta}, \mathcal{M}_K)] . \quad (16)$$

(The notation $\mathcal{M}_K$ has been used here to emphasise the connection with the value of K , although in general this method can be applied to any model). The Bayesian deviance is a function of $\boldsymbol{\theta}$, and can therefore be calculated for any chosen value of $\boldsymbol{\theta}$ – for example, it could be calculated from the posterior draws $\boldsymbol{\theta}_m$ described earlier. The derivation of L_K revolves around the assumption that the values $D(\boldsymbol{\theta}_m)$ are approximately normally distributed. Note that this is not the same as assuming that the log-likelihood is normally distributed – rather, it assumes that the log-likelihood of a series of posterior draws is approximately normally distributed. The mean and variance of this normal distribution can be estimated in the usual way:

$$\mathbb{E}[D(\boldsymbol{\theta}) | \mathbf{x}] \approx \frac{1}{t} \sum_{m=1}^t D(\boldsymbol{\theta}_m) = \hat{\mu} , \quad (17)$$

$$\text{Var}[D(\boldsymbol{\theta}) | \mathbf{x}] \approx \frac{1}{t-1} \sum_{m=1}^t (D(\boldsymbol{\theta}_m) - \hat{\mu})^2 = \hat{\sigma}^2 . \quad (18)$$

If the values $D(\boldsymbol{\theta}_m)$ are indeed drawn from a $\text{Normal}(\hat{\mu}, \hat{\sigma}^2)$ distribution then it follows from a simple transformation of (16) that the values $1/\Pr(\mathbf{x} | \boldsymbol{\theta}_m, \mathcal{M}_K)$ must be drawn from a $\text{Lognormal}(\hat{\mu}/2, \hat{\sigma}^2/4)$ distribution. The expectation of this distribution is given by $\exp\left\{\frac{\hat{\mu}}{2} + \frac{\hat{\sigma}^2}{8}\right\}$, and the RHS of equation (4) can be seen an estimator of this expected value. It follows that

$$\frac{1}{\Pr(\mathbf{x} | \mathcal{M}_K)} \approx \exp\left\{\frac{\hat{\mu}}{2} + \frac{\hat{\sigma}^2}{8}\right\} , \quad (19)$$

from which we obtain the heuristic estimator L_K :

$$-2 \log [\Pr(\mathbf{x} | \mathcal{M}_K)] \approx \hat{\mu} + \frac{\hat{\sigma}^2}{4} = L_K . \quad (20)$$

3.2 The BIC

The Bayesian Information Criterion (BIC) can be understood as a heuristic estimator of the model evidence, or as a general model choice criterion. In this section we deal with the former interpretation.

The derivation of the BIC as an estimator of the model evidence relies on several assumptions, and hence definitions of the BIC tend to vary slightly from text to text depending on which assumptions are used. The version presented here attempts to bridge the gap between these different definitions, and is based on the excellent explanation by Raftery (1995).

First, let us define the function $g(\boldsymbol{\theta})$ as the logarithm of the prior times the likelihood:

$$g(\boldsymbol{\theta}) = \log [\Pr(\mathbf{x} | \boldsymbol{\theta}, \mathcal{M}) \Pr(\boldsymbol{\theta} | \mathcal{M})] . \quad (21)$$

Note that the true model evidence is equal to the integral of $\exp \{g(\boldsymbol{\theta})\}$ over $\boldsymbol{\theta}$. The first step in the BIC derivation is to assume that $g(\boldsymbol{\theta})$ is well-approximated by a second order Taylor series expansion about the maximum value $\tilde{\boldsymbol{\theta}}$

$$g(\boldsymbol{\theta}) \approx g(\tilde{\boldsymbol{\theta}}) + (\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^T g'(\tilde{\boldsymbol{\theta}}) + \frac{1}{2}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^T g''(\tilde{\boldsymbol{\theta}})(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) , \quad (22)$$

where $g'(\boldsymbol{\theta})$ and $g''(\boldsymbol{\theta})$ represent the first and second derivatives of the original function. The $g'(\boldsymbol{\theta})$ term in (22) must equal zero (the gradient at the maximum value $\tilde{\boldsymbol{\theta}}$ is zero by definition). Exponentiating the remaining terms we find that

$$\Pr(\mathbf{x} | \boldsymbol{\theta}, \mathcal{M}) \Pr(\boldsymbol{\theta} | \mathcal{M}) \approx \exp \{g(\tilde{\boldsymbol{\theta}})\} \exp \left\{ -\frac{1}{2}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^T \tilde{H}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) \right\} , \quad (23)$$

where $\tilde{H} = -g''(\tilde{\boldsymbol{\theta}})$ is the hessian matrix for the function $g(\boldsymbol{\theta})$, evaluated at the point $\tilde{\boldsymbol{\theta}}$. The quantity on the RHS of (23) is proportional to the multivariate normal density. Hence, we can say that the first assumption of the BIC is that the posterior distribution is approximately normally distributed about the maximum. Integrating (23) over $\boldsymbol{\theta}$ we obtain

$$\begin{aligned} \Pr(\mathbf{x} | \mathcal{M}) &\approx \exp \{g(\tilde{\boldsymbol{\theta}})\} \int_{\boldsymbol{\theta}} \exp \left\{ -\frac{1}{2}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}})^T \tilde{H}(\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}) \right\} d\boldsymbol{\theta} , \\ &= \exp \{g(\tilde{\boldsymbol{\theta}})\} (2\pi)^{\frac{d}{2}} |\tilde{H}|^{-\frac{1}{2}} , \end{aligned} \quad (24)$$

where $|\tilde{H}|$ indicates the determinant of the matrix $\tilde{H}$ (recall that d denotes the number of free parameters in the model, hence we have a $d \times d$ Hessian matrix). Taking logarithms and making use of (21) we obtain

$$\begin{aligned} \log [\Pr(\mathbf{x} | \mathcal{M})] &\approx \log [\Pr(\mathbf{x} | \tilde{\boldsymbol{\theta}}, \mathcal{M})] + \log [\Pr(\tilde{\boldsymbol{\theta}} | \mathcal{M})] + \frac{d}{2} \log [2\pi] \\ &\quad - \frac{1}{2} \log [|\tilde{H}|] . \end{aligned} \quad (25)$$

Next, we assume that the maximum *a posteriori* value $\tilde{\boldsymbol{\theta}}$ is approximately equal to the maximum likelihood value $\hat{\boldsymbol{\theta}}$. Similarly, we assume that the

matrix $\tilde{H} = -g''(\tilde{\boldsymbol{\theta}})$ can be effectively replaced by $\hat{H} = -g''(\hat{\boldsymbol{\theta}})$. This assumption is valid when the prior is relatively flat around the maximum likelihood value, and/or when the sample size is large (and hence the likelihood is sharply peaked). This leads us to the expression

$$\begin{aligned} \log [\Pr(\mathbf{x} | \mathcal{M})] &\approx \log [\Pr(\mathbf{x} | \hat{\boldsymbol{\theta}}, \mathcal{M})] + \log [\Pr(\hat{\boldsymbol{\theta}} | \mathcal{M})] + \frac{d}{2} \log[2\pi] \\ &\quad - \frac{1}{2} \log [|\hat{H}|] . \end{aligned} \quad (26)$$

We can write the $(u, v)^{\text{th}}$ element of $\hat{H}$ as follows:

$$\begin{aligned} \hat{H}_{uv} &= - \left. \frac{\partial^2 g(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}_u \partial \boldsymbol{\theta}_v} \right|_{\hat{\boldsymbol{\theta}}} , \\ &= - \left. \frac{\partial^2 \log [\Pr(\mathbf{x} | \boldsymbol{\theta}, \mathcal{M}) \Pr(\boldsymbol{\theta} | \mathcal{M})]}{\partial \boldsymbol{\theta}_u \partial \boldsymbol{\theta}_v} \right|_{\hat{\boldsymbol{\theta}}} , \\ &= - \left. \frac{\partial^2 \log [\Pr(\mathbf{x} | \boldsymbol{\theta}, \mathcal{M})]}{\partial \boldsymbol{\theta}_u \partial \boldsymbol{\theta}_v} \right|_{\hat{\boldsymbol{\theta}}} - \left. \frac{\partial^2 \log [\Pr(\boldsymbol{\theta} | \mathcal{M})]}{\partial \boldsymbol{\theta}_u \partial \boldsymbol{\theta}_v} \right|_{\hat{\boldsymbol{\theta}}} . \end{aligned} \quad (27)$$

Once again, making use of our assumption that the prior $\Pr(\boldsymbol{\theta} | \mathcal{M})$ is relatively flat around the maximum likelihood value, we find that the second term in (27) can be dropped (ie. there is no curvature at the maximum). Focussing on the first term, we note that the log-likelihood can be decomposed as follows:

$$\log [\Pr(\mathbf{x} | \boldsymbol{\theta}, \mathcal{M})] = \frac{1}{n} \sum_{i=1}^n n \log [\Pr(\mathbf{x}_i | \boldsymbol{\theta}, \mathcal{M})] . \quad (28)$$

Assuming that the sample size is large, we can invoke the weak law of large numbers on the random variable

$Z = \frac{1}{n} \sum_{i=1}^n n \log [\Pr(\mathbf{x}_i | \boldsymbol{\theta}, \mathcal{M})]$ to obtain

$$\frac{1}{n} \sum_{i=1}^n n \log [\Pr(\mathbf{x}_i | \boldsymbol{\theta}, \mathcal{M})] \xrightarrow{P} \mathbb{E} [n \log [\Pr(\mathbf{x}_0 | \boldsymbol{\theta}, \mathcal{M})]] , \quad (29)$$

where $\Pr(\mathbf{x}_0 | \boldsymbol{\theta}, \mathcal{M})$ denotes the average probability of a single observation drawn from the model. Substituting back into (27) we obtain

$$\begin{aligned} \hat{H}_{uv} &= - \left. \frac{\partial^2 \mathbb{E} [n \log [\Pr(\mathbf{x}_0 | \boldsymbol{\theta}, \mathcal{M})]]}{\partial \boldsymbol{\theta}_u \partial \boldsymbol{\theta}_v} \right|_{\hat{\boldsymbol{\theta}}} , \\ &= -n \left. \frac{\partial^2 \mathbb{E} [\log [\Pr(\mathbf{x}_0 | \boldsymbol{\theta}, \mathcal{M})]]}{\partial \boldsymbol{\theta}_u \partial \boldsymbol{\theta}_v} \right|_{\hat{\boldsymbol{\theta}}} , \\ &= n I_{uv} , \end{aligned} \quad (30)$$

where I is the expected Fisher information matrix. Simply put; as the sample size increases, the average amount of information contained in our observed set of draws tends towards the theoretical expected amount of information predicted for any single draw from the model. Using this simplification we find that

$$|\hat{H}| = n^d |I| . \quad (31)$$

Finally, substituting this value back into (26) we obtain

$$\log [\Pr(\mathbf{x} | \mathcal{M})] \approx \log [\Pr(\mathbf{x} | \hat{\boldsymbol{\theta}}, \mathcal{M})] - \frac{d}{2} \log[n] - \frac{1}{2}C , \quad (32)$$

where

$$C = -2 \log [\Pr(\hat{\boldsymbol{\theta}} | \mathcal{M})] - d \log[2\pi] + \log [|I|] . \quad (33)$$

All the terms that make up C are constant, in the sense that they do not depend on the data. Thus, (33) contains all the elements of the BIC approximation that tend towards a constant value as $n \rightarrow \infty$. The BIC in its most general form includes this unknown constant, and is defined as -2 times (32):

$$\text{BIC} = -2 \log [\Pr(\mathbf{x} | \hat{\boldsymbol{\theta}}, \mathcal{M})] + d \log[n] + C . \quad (34)$$

In particular, this is the value of the BIC assuming a flat, uninformative prior on $\boldsymbol{\theta}$. In order to eliminate the constant C we must make a different assumption about the prior – we must assume that the prior is multivariate normal, with mean $\hat{\boldsymbol{\theta}}$ and covariance matrix I^{-1} . This is a particular form of the ‘unit information prior’, so called because the prior contains the same information as would, on average, a single observation drawn from the model (in this case drawn from a standard normal likelihood). Note that Raftery (1999) provides an argument in favour of the unit information prior as a very general and relatively uninformative prior. Under this prior we find that

$$\log [|I|] = 2 \log [\Pr(\hat{\boldsymbol{\theta}} | \mathcal{M})] + d \log[2\pi] , \quad (35)$$

and hence $C = 0$. This brings us to the more common definition of the BIC:

$$\text{BIC} = -2 \log [\Pr(\mathbf{x} | \hat{\boldsymbol{\theta}}, \mathcal{M})] + d \log[n] . \quad (36)$$

An alternative argument for why ignoring the quantity C might be reasonable in practice is that it tends to cancel out to some extent when comparing different models. The log-odds ratio between two competing models can be estimated using the difference in BIC values:

$$\log \left[\frac{\Pr(\mathbf{x} | \mathcal{M}_1)}{\Pr(\mathbf{x} | \mathcal{M}_2)} \right] \approx -\frac{1}{2} (\text{BIC}_1 - \text{BIC}_2) , \quad (37)$$

which in turn leads to an expression involving the difference in C values:

$$\begin{aligned} \text{BIC}_1 - \text{BIC}_2 = & -2 \left[\log [\text{Pr}(\mathbf{x} | \hat{\boldsymbol{\theta}}, \mathcal{M}_1)] - \log [\text{Pr}(\mathbf{x} | \hat{\boldsymbol{\theta}}, \mathcal{M}_2)] \right] \\ & + (d_1 - d_2) \log[n] + (C_1 - C_2) . \end{aligned} \quad (38)$$

As long as $C_1 - C_2$ is small we find that the difference in BIC values as defined by (34) is similar to the difference as defined by (36), and so the latter formula suffices. Thus it may be that the log-odds can be estimated with some accuracy, even though the log-evidence of each model cannot. Similarly, as long as C_i is small and/or similar across a range of models we find that the model evidence can be approximated using

$$\text{Pr}(\mathbf{x} | \mathcal{M}_i) \approx \frac{\exp\{-\frac{1}{2}\text{BIC}_i\}}{\sum_j \exp\{-\frac{1}{2}\text{BIC}_j\}} . \quad (39)$$

Although this argument can be used to justify the use of the BIC as an estimator of the model evidence, in the main text we take the more general interpretation of the BIC as a measure of model fit.

4 General Model Comparison Statistics

4.1 The AIC

The Akaike Information Criterion (AIC) differs from the statistics described above in that it does not aim to approximate the model evidence. Rather, it is derived with the aim of minimising the expected Kullback-Leibler divergence between a ‘true’ model and an approximating model. A complete derivation of the AIC is beyond the scope of this document, but can be found in Burnham and Anderson (2002, 2004). The formula for the AIC is as follows:

$$\text{AIC} = -2 \log [\text{Pr}(\mathbf{x} | \hat{\boldsymbol{\theta}}, \mathcal{M})] + 2d , \quad (40)$$

The model with the smallest AIC value is usually chosen, although models can also be weighted using Akaike weights, w_i . These are calculated using

$$w_i = \frac{\exp\{-\frac{1}{2}\Delta_i\}}{\sum_j \exp\{-\frac{1}{2}\Delta_j\}} , \quad (41)$$

where

$$\Delta_i = \text{AIC}_i - \text{AIC}_{\min} . \quad (42)$$

Notice the direct connection between Akaike weights and the BIC-based estimate of the model evidence defined in (39). This is particularly interesting considering the very different paradigms that are used to derive these statistics.

4.2 The BIC (again)

As well as being a heuristic estimator of the model evidence, the BIC can be seen in a more general light as a simple model choice criterion. This comes about due to its connection with the AIC. Recall the basic formula for the BIC:

$$\text{BIC} = -2 \log [\Pr(\mathbf{x} \mid \hat{\boldsymbol{\theta}}, \mathcal{M})] + d \log[n] . \quad (43)$$

Notice that the only difference between (43) and (40) is in the way that the log-likelihood is penalised by the number of free parameters; the BIC tends to punish complex models more harshly than the AIC (where complexity refers to the number of free parameters in the model). This is seen by some as an advantage of the BIC, as the AIC has often been found to overestimate the number of parameters needed (Raftery (1999) and references therein).

4.3 The DIC

The Deviance Information Criterion (DIC), proposed by Spiegelhalter et al. (2002), can be seen as an extension of earlier model fitting concepts to models in which the true number of independent parameters may be unknown – for example as a result of prior distributions that introduce a dependence between parameters. In place of a fixed number of parameters d we have an ‘effective number of parameters’ p_D , which can be estimated from MCMC output. The DIC is defined as

$$\text{DIC} = \bar{D} + p_D , \quad (44)$$

where $\bar{D}$ is the expected posterior deviance. In practice $\bar{D}$ can be estimated using $\hat{\mu}$, defined in section 3.1. There are two main ways of calculating p_D . The first, due to Spiegelhalter et al. (2002), is to use

$$p_D = \bar{D} - D(\boldsymbol{\theta}^*) , \quad (45)$$

where $\boldsymbol{\theta}^*$ is some point estimate of $\boldsymbol{\theta}$. Substituting the value $\bar{D} = D(\boldsymbol{\theta}^*) + p_D$ from (45) into (44) we obtain

$$\text{DIC}_S = D(\boldsymbol{\theta}^*) + 2p_D . \quad (46)$$

The connection to the AIC and BIC now becomes obvious, and in fact if $\boldsymbol{\theta}^* = \hat{\boldsymbol{\theta}}$ and $p_D = d$ then DIC_S becomes exactly equivalent to the AIC.

The second method of calculating p_D is due to Gelman et al. (2013), and estimates p_D using half the variance of the posterior deviance. This variance is equivalent to the value $\hat{\sigma}^2$ defined in section 3.1. Using $\hat{\mu}$ as an estimate

of $\bar{D}$ as before, we find that the Gelman et al. (2013) version of the DIC can be defined as follows:

$$\text{DIC}_G = \hat{\mu} + \frac{\hat{\sigma}^2}{2} . \quad (47)$$

Thus we find that the DIC is also extremely closely related the estimator L_K , differing only in how much the variance $\hat{\sigma}^2$ contributes to the final estimate (a point made in the appendix of Pritchard et al. (2000)).

References

- Kenneth P Burnham and David R Anderson. *Model selection and multimodel inference: a practical information-theoretic approach*. Springer, 2002.
- Kenneth P Burnham and David R Anderson. Multimodel inference understanding aic and bic in model selection. *Sociological methods & research*, 33(2):261–304, 2004.
- Nial Friel and Anthony N Pettitt. Marginal likelihood estimation via power posteriors. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(3):589–607, 2008.
- Nial Friel, Merrilee Hurn, and Jason Wyse. Improving power posterior estimation of statistical evidence. *Statistics and Computing*, pages 1–15, 2013.
- Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*. CRC press, 2013.
- Nicolas Lartillot and Hervé Philippe. Computing bayes factors using thermodynamic integration. *Systematic biology*, 55(2):195–207, 2006.
- Michael A Newton and Adrian E Raftery. Approximate bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 3–48, 1994.
- Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2): 945–959, 2000.
- Adrian E Raftery. Bayesian model selection in social research. *Sociological methodology*, 25:111–164, 1995.

- Adrian E Raftery. Bayes factors and bic. *Sociological Methods & Research*, 27(3):411–417, 1999.
- Adrian E Raftery, Michael A Newton, Jaya M Satagopan, and Pavel N Krivitsky. Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. 2006.
- David J Spiegelhalter, Nicola G Best, Bradley P Carlin, and Angelika Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4): 583–639, 2002.

Documentation for MAVERICK software: Version 1.0

Robert Verity
MRC centre for outbreak analysis and modelling
Imperial College London

and

Richard A. Nichols
Queen Mary University of London

May 19, 2016

Software available from
www.bobverity.com/Maverick

Contents

1	Introduction	2
2	Program execution	3
2.1	Running the program	3
2.2	The parameters file	4
2.3	The data file	5
3	Available models	6
3.1	Basic model	7
3.2	Admixture model	8
4	The core algorithm	9
4.1	The collapsed Gibbs sampler	9
4.2	Smoothed posterior estimates	10
4.3	Label switching	11
4.4	Thinning and autocorrelation	13
5	Estimating K	15
5.1	Thermodynamic integration	15
5.2	Alternative methods	16
6	Detailed inputs and outputs	17
6.1	List of input arguments/parameters	17
6.2	List of output files	23

1 Introduction

MAVERICK is a program for inferring population structure on the basis of genetic information. The mixture modelling framework assumed in MAVERICK is identical to that used in the program STRUCTURE (Pritchard et al., 2000; Falush et al., 2003, 2007; Hubisz et al., 2009), although the implementation is quite different. The chief strength of MAVERICK lies in its ability to estimate the number of demes (denoted K) in a reliable way using a technique known as *thermodynamic integration* (TI). While this technique is more computationally intensive than standard MCMC based methods, it has been found to provide estimates of K that are more accurate and precise than those

based on some heuristic estimators. MAVERICK also implements certain MCMC techniques that reduce the number of iterations needed to achieve convergence, and allows for some non-standard inputs and outputs.

This document gives details of MAVERICK v1.0, including the models that are available, the format of input and output files, and the parameters and arguments that can be passed to the program. It also describes the basic workings of the core algorithm and gives a sketch of how TI can be used to estimate the evidence for K . This document is intended for reference only. **Users are strongly encouraged to look at the accompanying tutorial file which contains worked examples showing how to use the program (available from www.bobverity.com/maverick).**

2 Program execution

MAVERICK v1.0 is written in C++, and separate versions exist for Windows and Mac OS X operating systems. From a work-flow point of view MAVERICK is very simple. First, the program reads in two text files – a parameters file, called `parameters.txt` by default, and a data file, called `data.txt` by default. The program then carries out the analysis specified by the user in the parameters file. Outputs are produced in the form of multiple `.csv` and `.txt` files. This section deals with program execution and the formats of the `parameters.txt` and `data.txt` files.

2.1 Running the program

MAVERICK v1.0 is distributed in the form of a simple executable. On a Mac, and almost all of the time on a PC, this executable should be run from command line (i.e. through Terminal on a Mac or from the console on a PC). After navigating to the correct location the program can be run using the command `./Maverick` on a Mac or simply `Maverick` on a PC. See the OS-specific tutorial files for further details, including a refresher on how to use the command line on both Mac and PC.

Options for choosing what sort of analysis to perform are specified in two places. The first place is the `parameters.txt` file. This is where majority of options will tend to be defined; for example the choice of model, the number of MCMC iterations and a list of desired outputs can all be specified here.

The second place is on the command line, as arguments to the program. Arguments can be passed to the program immediately after the execution command (see section 6.1 for a complete list of available arguments). **Every parameter present in the `parameters.txt` file can also be specified on the command line, and if parameters are specified in both locations then the command line takes precedence.** There are a couple of reasons for choosing this setup. First, there are some parameters that cannot be specified in the `parameters.txt` file – namely all those parameters that direct the program to the `parameters.txt` file! Second, having the option to define parameters on the command line allows the user to make slight modifications to an existing scheme without having to produce a new copy of the `parameters.txt` file every time.

2.2 The parameters file

The `parameters.txt` file contains a list of parameter names and values, separated by a single tab character. Parameters can be listed in any order, and all characters and white space that are not associated with a valid parameter name are ignored, making it easy to insert notes and comments throughout the file. Parameters ending in `_on` are Boolean parameters, and must be set to `1\t\true\T\True\TRUE` or `0\f>false\F\False\FALSE`. Other parameters are integers, decimal values, or in some rare cases lists of decimal values separated by a comma. Any parameters that are not specified either in the `parameters.txt` file or on the command line are given default values. An example of a correctly formatted `parameters.txt` file is given in Figure 1.

```

Analysis of human population data under admixture
model with K=3.

20 Jan 2015

##### Data properties
headerRow_on    t
ploidyCol_on    t
popCol_on       t
ploidy_2        t
missingData     -9

##### Model parameters
Kmin            3
Kmax            3
admix_on        t
fixAlpha_on     t
alpha           1.0
alphaPropSD     0.10

.
.
.

```

Figure 1: Example of a correctly formatted `parameters.txt` file (only a subset of possible parameters are shown here). Parts of the file that are interpreted by MAVERICK are circled in red, all other parts are ignored.

A complete list of all possible parameters and their default values is given in section 6.1.

2.3 The data file

The data format accepted by MAVERICK is very similar to the basic data format used by STRUCTURE. Information is arranged in a tab-delimited matrix, with rows corresponding to individuals and columns corresponding to loci. In some cases we may want the first row of the file to contain information that should be ignored by the program – for example header information for each locus. This can be accommodated by setting the `headerRow_on` parameter to true, in which case anything on the first line of the `data.txt` file is ignored. Moving on to columns, the first column must contain sample IDs. The second column can optionally be used to specify the population of origin or sampling location of each individual. This option is activated by setting the parameter `popCol_on` to true. It is important to note that **population of origin data is not used within any existing model in Maverick v1.0**. Rather, this option is included because some outputs can be produced at the population level (see Q-matrix information in section 3.1), and also to give some flexibility when reading in different data types. The next

column (either the second column if `popCol_on` is false or the third column if `popCol_on` is true) can be used to specify the ploidy of each individual, thereby allowing for mixed ploidy. This option is activated by setting the parameter `ploidyCol_on` to true, otherwise the ploidy of all individuals will be set by the integer parameter `ploidy`. If polyploid individuals are being used then a different row should be used for each gene copy (as in the standard STRUCTURE format). There are two possible formats that can be used: i) ploidy is defined in the *first* row of each individual, ii) ploidy is defined identically in *every* row of each individual (and analogously for the population-of-origin column). All remaining columns are for genetic information only, with each column corresponding to a different locus. Missing data can be represented using any numerical value or character string, and is defined using the parameter `missingData`. An example of a correctly formatted `data.txt` file is given in Figure 2.

sampleID	pop	ploidy	locus1	locus2	locus3	locus4	locus5
indiv1.1	1	2	3	2	5	4	4
indiv1.2	1		5	1	5	2	2
indiv2.1	1	2	5	4	5	2	5
indiv2.2	1		2	4	5	4	2
indiv3.1	1	2	1	2	5	1	2
indiv3.2	1		5	1	1	2	3
indiv4.1	2	2	1	2	3	4	5
indiv4.2	2		5	4	5	2	3
indiv5.1	2	2	3	3	5	2	5
indiv5.2	2		5	1	1	1	5
indiv6.1	2	2	4	1	5	5	2
indiv6.2	2		5	2	3	2	2

Figure 2: Example of a correctly formatted `data.txt` file that should be analysed with `headerRow_on` set to true, and both `popData_on` and `ploidyCol_on` set to true. Here ploidy is defined using the first formatting style, and population of origin using the second formatting style.

3 Available models

The models available in MAVERICK v1.0 are identical to a subset of models available in STRUCTURE v2.3, including models both with and without admixture. There are, however, some models available in STRUCTURE v2.3 that are not implemented in MAVERICK v1.0. For example, here we assume that allele frequencies are independent between demes, while STRUCTURE v2.3 has options for allowing allele frequencies to be correlated between demes.

Similarly, we do not make use of population-of-origin information, with all individuals having an equal chance of originating from any deme. Later versions of MAVERICK are likely to contain a richer model set.

3.1 Basic model

In the basic model used by both STRUCTURE and MAVERICK we start with K distinct subpopulations, or demes. Each of these demes is assumed to be in Hardy Weinberg equilibrium with its own set of allele frequencies. These allele frequencies are unknown, and are given a Dirichlet prior. The observed data are then modelled as a series of independent draws from the population as a whole, in which we first draw a random deme, with equal probability, and then a random genotype from the allele frequency distribution within this deme. All loci are assumed to be in linkage equilibrium. Mathematically we can write

$$\begin{aligned} x_{ila} &\sim \text{Categorical}(\mathbf{p}_{z_i l}) , \\ z_i &\sim \text{Categorical}(\frac{1}{K}) , \\ \mathbf{p}_{kl} &\sim \text{Dirichlet}(\lambda_{l1}, \dots, \lambda_{lJ_l}) , \end{aligned} \tag{1}$$

where x_{ila} is the genetic information for individual i at locus l and gene copy a , z_i is the group allocation of individual i , $\mathbf{p}_{kl}$ is the allele frequency distribution of deme k at locus l , and λ_{lj} is a prior parameter on the allele frequencies which defaults to 1.0 for all λ_{lj} (J_l denotes the number of alleles at locus l). This model is due to Pritchard et al. (2000).

The major advantage of this framework is that, despite its simplicity, it captures a fundamental aspect of structured populations – that our observed data may represent a mixture of several distinct genetic clusters, rather than one single cluster. By modelling these clusters explicitly and then attempting to infer the allocation of individuals to clusters (demes) probabilistically, we can tease apart the distinct genetic groups that are hidden within the aggregate data.

The most important output from the basic model described above is the so-called Q-matrix. This matrix contains the posterior allocation probability of each individual belonging to each cluster. In other words, the Q-matrix gives the posterior distribution of $\mathbf{z}$. As well as producing a Q-matrix at the individual level, it is possible to produce a Q-matrix at the population level

by making use of population-of-origin information (here the word ‘population’ refers to the sampling location of individuals, and should be kept conceptually distinct from the demes used in the model). The allocation probability at the population level is simply calculated as the average of the allocation probabilities of all individuals belonging to that population. This sort of aggregate view of the information can be useful when dealing with large samples, or when we have strong prior reason to believe that different sampling locations will represent different genetic clusters.

3.2 Admixture model

In some cases it may be too simplistic to assume that the entire genome of an individual belongs to the same genetic cluster. Instead we may find when looking across the genome that loci can be separated into clusters independently, or in blocks. This can come about due to admixture of several populations, resulting in a genome that is a mosaic of different underlying allele frequency distributions.

MAVERICK v1.0 allows for the same basic admixture model employed by STRUCTURE v2.3, which is an extension of the model described above to allow for the possibility that different gene copies within an individual may belong to different demes. Thus, rather than an entire individual being allocated to a deme using the variable z_i , we model each gene copy as being allocated to a deme using the variable z_{ila} . The *proportion* of individual i that belongs to each deme is given by the vector $\mathbf{q}_i$, and z_{ila} is simply modelled as a draw from these proportions. For example, if $\mathbf{q}_i$ gives an equal proportion to each deme then every gene copy within individual i has an equal chance of belonging to each deme, whereas if $\mathbf{q}_i$ gives an unequal proportion to each deme then the gene copies within an individual are likely to co-associate with one another. If $\mathbf{q}_i$ gives *all* of the proportion of an individual to just one deme then we arrive back at a model in which genetic information is clustered at the individual level. Where we lie on this spectrum is controlled by the parameter α , which enters into the model via a Dirichlet(α) prior on $\mathbf{q}_i$. When $\alpha=0$ clustering occurs at the individual level, and this model becomes formally equivalent to the basic without-admixture model described above, while when $\alpha \rightarrow \infty$ clustering occurs independently at the gene copy level. Anything between these values represents an intermediate level of admixture. Mathematically

we can write

$$\begin{aligned}
x_{ila} &\sim \text{Categorical}(\mathbf{p}_{z_{ila}l}) , \\
z_{ila} &\sim \text{Categorical}(\mathbf{q}_i) , \\
\mathbf{p}_{kl} &\sim \text{Dirichlet}(\lambda_{l1}, \dots, \lambda_{lJ_l}) , \\
\mathbf{q}_i &\sim \text{Dirichlet}(\alpha, \dots, \alpha) .
\end{aligned} \tag{2}$$

A small extension to this basic model is to assume that the admixture parameter α is unknown, in which case it can be given a $\text{Uniform}(0, 10)$ prior (this extension is activated by default under the admixture model in STRUCTURE v2.3). This can be switched on in MAVERICK by setting the parameter `fixAlpha_on` to false. Once again **we take absolutely no credit for this model**, which was originally described by Pritchard et al. (2000). Our use of this model in MAVERICK is simply to provide results that are directly comparable with those produced by STRUCTURE.

4 The core algorithm

Although the models used by MAVERICK v1.0 are identical to a subset of those used by STRUCTURE v2.3, the ways in which the MCMC algorithms work in the two programs is slightly different. This section describes key differences in the MAVERICK algorithm and explains the consequences of these differences in terms of program execution.

4.1 The collapsed Gibbs sampler

The method used to draw from the posterior distribution in the STRUCTURE software is known as Gibbs sampling. The basic logic behind this method is that complex likelihood surfaces can be explored by iteratively updating parameter values one at a time conditional on all other values. Thus, the allocation of individuals to demes (denoted Z) can be updated conditional on the known allele frequencies in those demes (denoted P), and vice versa. What differentiates Gibbs sampling from Metropolis-Hastings sampling (the other main MCMC sampling method) is that parameters are drawn from their *exact* conditional distributions, rather than proposing a move by drawing from some

arbitrary distribution and then accepting or rejecting the move as needed. In essence, a Gibbs sampler can be thought of as a Metropolis-Hastings sampler in which every move is accepted.

Although Gibbs sampling has been known about in mixture modelling applications for some time (Diebolt and Robert, 1994), the method only came to fruition in a genetic context through the work of Pritchard et al. (2000), which itself built on the work of Rannala and Mountain (1997). The conditional distributions derived by these authors form the basis of the Gibbs sampling method used in programs such as STRUCTURE. However, there is more than one way to design a Gibbs sampler from these expressions. As stated by Rannala and Mountain (1997), and reiterated by Huelsenbeck and Andolfatto (2007), one of the parameters in the update step – the allele frequencies in all demes, P – can be integrated out of the problem analytically, meaning it does not need to be updated as part of the algorithm. This leads to a more efficient algorithm in which only the group allocation Z is updated according to a new set of probabilities. This method of integrating over parameters prior to the update step is sometimes referred to as ‘collapsed’ Gibbs sampling, although really this is nothing more than an ordinary Gibbs sampler defined on a slightly different version of the likelihood. MAVERICK uses this collapsed Gibbs sampling method.

There are some advantages to using a collapsed Gibbs sampler where possible. The fact that the allele frequencies are integrated out of the likelihood leads to reduced autocorrelation between sequential iterations of the MCMC, as individual allocations are not all drawn conditional on the same set of allele frequencies. This in turn leads to better mixing of the MCMC, and reduces the number of samples that are required to obtain pseudo-independent draws from the posterior.

4.2 Smoothed posterior estimates

The objective in MCMC is to design an algorithm that converges on the posterior distribution of interest (see Sorensen and Gianola (2002) for a more formal definition). The output of an MCMC algorithm is a series of values that are (asymptotically) drawn from the posterior. Ordinarily these raw values are used to summarise the posterior distribution; for example by producing a histogram or kernel density plot. However, in many cases it is extremely wasteful to use just these raw values when we have access to much more detailed information. Recall that in Gibbs sampling we draw values at

each iteration from their conditional distribution given all other parameters. Thus, at each iteration we have access to both a single random realisation from this distribution *and* knowledge of the entire distribution itself. As shown by Gelfand and Smith (1990), estimating the posterior distribution using the average of these conditional distributions produces estimates that are *always* more accurate than estimates based on the realisations themselves. This argument is a version of the well-known Rao-Blackwellization theory in statistics.

This method is applied in MAVERICK to produce smoothed estimates of the posterior allocation of individuals. Rather than estimating allocation probabilities by taking the average number of iterations that each individual spends allocated to each deme, estimates are produced by averaging over the *probabilities* of being allocated to each deme (Figure 3).

Figure 3: Schematic showing how averaging over distributions rather than counting random draws from these distributions tends to produce more accurate estimates of the posterior allocation probability.

This results in greater accuracy for a given number of iterations, or phrased another way, fewer iterations are needed to obtain the same level of accuracy. The error in estimation is one of the additional outputs of MAVERICK. This provides the user with a measure of confidence in results that can be used to decide whether sufficient sampling iterations have been obtained, or whether more are needed.

4.3 Label switching

A persistent problem in mixture modelling applications is the non-identifiability of mixture components. Simply put, although we specify a model with K demes, there is nothing to distinguish any one deme from another in an

absolute sense. If many individuals are allocated to deme 1 then another genetically similar individual is also likely to belong to this deme, but if these same individuals were allocated to deme 2 then the probabilities are mirrored and the new individual is likely to belong to deme 2. We can imagine that the likelihood surface is made of up of a single unique unit that is reflected $K!$ (K factorial) times, corresponding to the $K!$ ways that the demes can be relabelled. This causes problems when performing inference under mixture models, as the different reflections tend to cancel each other out. Although an MCMC sampler might be very confident that a certain group of individuals belong in deme 1 at a particular iteration, if the sampler is run for long enough it will eventually move into a different reflection in the likelihood space and be equally confident that these same individuals belong in deme 2 – a process referred to as *label switching*. Technically speaking, run for long enough the MCMC will place all individuals into each deme an exactly equal number of times, meaning when we come to estimate allocation probabilities we have zero power to detect population structure, no matter how strong it is! In practice the problem is not usually this severe, as MCMC chains are rarely run for this long, but some label switching may still be present in shorter runs, leading to poor estimates of population structure.

One way around the label switching problem is to deliberately design an MCMC sampler to be poorly mixing. If a sampler tends to get ‘stuck’ in one part of the likelihood space then it is unlikely to traverse the reflections, and so label switching is unlikely (but not impossible) to occur. However, this method cannot guarantee that label switching has not occurred, and the deliberate choice of a poorly mixing MCMC is a sacrifice for other reasons.

To avoid this problem, MAVERICK implements an explicit solution to the label switching problem put forward by Stephens (2000). The method works by comparing each new iteration against a running estimate of allocation probabilities, and swapping labels round on-the-fly if label switching appears to have occurred. More formally, in each new iteration the Kullback-Leibler divergence (a measure of dissimilarity) between the current distribution and the running estimate of allocation probabilities is calculated under every possible permutation of the component labels, and the permutation with the lowest divergence is chosen. Although exploring every permutation of component labels in every iteration of the MCMC sounds like a computationally intensive task, this is implemented in an efficient way so that in the vast majority of cases only a handful of operations need to be performed (most of the time labels do not need to be swapped).

4.4 Thinning and autocorrelation

A well-designed MCMC will produce a series of draws from the posterior distribution of interest, however, these draws are not necessarily independent. If the likelihood space is difficult to traverse (as is often the case for models with a large number of parameters) the MCMC will tend to produce draws that are highly autocorrelated. This autocorrelation can be visualised by plotting the likelihood at each iteration of the MCMC, or even better by producing an autocorrelation plot as in Figure 4B.

Figure 4: **A)** Trace of the marginal log-likelihood, showing burn-in as well as sampling iterations (samples start from iteration 1). **B)** An autocorrelation plot of the same values, produced using the R function `acf()`.

It is important to recognise the situations where we need to be worried about autocorrelation, but it is just as important to know the situations where we do not. For example, it is common practice when looking for population structure to run the same MCMC several times, calculating the STRUCTURE estimator L_K on each repeat to obtain a mean and variance or standard error over repeats. Notice that in this case we do not need to worry about

autocorrelation influencing our estimate of the standard error of L_K . This empirical estimate of the standard error is simulation-based, and therefore any autocorrelation is taken into account. This is also true of the Q-matrix standard error in MAVERICK, which is calculated from several independent estimates of the Q-matrix.

On the other hand, the standard error of the TI estimator is calculated as follows:

$$\text{SE}(\hat{T}_K) = \sqrt{\frac{\text{Var}(\hat{T}_K)}{\text{ESS}}}, \quad (3)$$

where ESS is the MCMC *effective sample size*. The ESS gives the *effective* number of independent samples among a set of not-necessarily-independent values. In most standard statistical applications we have independent samples, and so the ESS is equivalent to the sample size n and we arrive back at the more familiar formula for the standard error. However, for our purposes we can expect some autocorrelation and so we must take this into account. The ESS can be obtained from the following formula:

$$\text{ESS} = \frac{n}{1 + 2 \sum_{l=1}^{\infty} \rho_l}, \quad (4)$$

where ρ_l gives the autocorrelation at lag l (i.e. the average correlation between values separated by l iterations). MAVERICK performs this calculation automatically, and therefore estimates of the TI standard error produced by the program are already corrected for autocorrelation. A very important consequence of this is that **we do not need to thin chains!** Ordinarily autocorrelation is removed by “thinning” samples such that the remaining values are spaced far enough apart to be approximately independent. MAVERICK v1.0 retains the ability to thin chains (partly for historical reasons), however, thinning is nothing more than a less-efficient version of the correction described above, and so as a rule thinning should be avoided.

Moving on to the ways in which we do need to worry about autocorrelation, there is an important caveat to all of the points above. When working with MCMC we need to make every effort to ensure that the complete posterior distribution has been explored; if the MCMC tends to get stuck in one region of the posterior, and if we don’t run the chain for long enough for it to escape, then we can get a misleading picture of the final posterior distribution. This interacts with the issue of autocorrelation because what constitutes “long

enough” will depend on how autocorrelated the chain is, with more samples needed for highly autocorrelated chains. The proper way to deal with this is to run multiple chains, each from a different starting position (starting positions are drawn at random in MAVERICK v1.0), and observe whether the chains converge on the same distribution. This can be done informally in MAVERICK v1.0 by plotting likelihood traces and Q-matrix estimates over many replicates of the main MCMC. Future versions of the program are likely to deal with this problem in a more formal way using convergence diagnostics.

5 Estimating K

The main strength of MAVERICK v1.0 lies in its ability to estimate K reliably using thermodynamic integration (TI). Although **we strongly advocate using this method of estimating K** , some other methods are also supported, partly for purposes of comparison. This section gives a short overview of the TI methodology and describes the other methods that are implemented in MAVERICK.

5.1 Thermodynamic integration

Thermodynamic integration is a method for performing integration in high-dimensional spaces where ordinary quadrature-based methods are not suitable. In this case the integration is over all free parameters of the mixture model, resulting in an estimate of the model evidence (the marginal likelihood of the model). The TI method makes use of several closely related MCMC chains, each drawn the *power posterior* with a different power, β_i , spanning the interval $[0, 1]$ (see Lartillot and Philippe (2006) for details). The log-likelihood of each of these chains is then calculated, and the mean log-likelihoods over all iterations are output from the program. When these mean log-likelihoods are plotted against β they form a continuous path (Figure 5).

Figure 5: Mean log-likelihood plotted against the power used in the MCMC chain. In this example a large number of samples was taken, meaning error bars around individual points can hardly be seen, and a large number of rungs was used, meaning the path is relatively smooth over most of the domain. The area of the shaded region is the TI estimate of the model evidence.

The total area between this path and zero gives an estimate of the model evidence. When performing TI it is important that the number of samples obtained from the posterior distribution is large enough to obtain accurate estimates of the mean log-likelihood. It is also important that the number of ‘rungs’ used (the number of powers β_i) is large enough to produce a smooth curve when plotted as in Figure 5.

The fact that each rung requires a separate MCMC chain makes TI computationally intensive compared with some other MCMC based estimation methods. Thus, for very large data sets, in which running even a single MCMC to the point of convergence may be a challenge, TI will most likely not be possible. However, estimates produced in this way have been found to be more accurate and precise than estimates based on some simple heuristics, and so this method is recommended wherever computationally feasible.

5.2 Alternative methods

Although the principle goal of MAVERICK v1.0 is to estimate K through TI, a number of simpler estimation methods are also implemented. Most of these methods only require a single MCMC chain in their computation, making them considerably faster than TI. However, this increase in speed tends to come at the cost of some accuracy.

The first of these is the statistic L_K used by the program STRUCTURE, which is derived as an approximation to (rather than a direct estimator of) the model evidence. The second is Evanno’s Δ_K statistic, which uses multiple

repeats of L_K in an attempt to find the point of maximum curvature in the distribution of L_K (Evanno et al., 2005). These two methods have been by far the most popular ways of estimating K in studies of population structure to date. The third statistic is the harmonic mean estimator $\hat{h}_K$ of Newton and Raftery (1994). This estimator has been found to be highly unstable in some applications (Neal, 1994; Raftery et al., 2006), and for this reason estimates based on $\hat{h}_K$ should be treated with extreme caution until further validation. Some more traditional model comparison estimates are also implemented, including the Akaike information criterion (AIC), the Bayesian information criterion (BIC) and the deviance information criterion (DIC) calculated using the methods of Spiegelhalter et al. (2002) (DIC_S) and Gelman et al. (2014) (DIC_G). Finally, MAVERICK contains an option for calculating the exact model evidence by brute force. This is extremely computationally intensive, and should only be attempted on very small data sets. Models both with and without admixture are supported, although only the model without admixture permits reasonable sample sizes (reasonable meaning around 10 individuals)!

6 Detailed inputs and outputs

Section 6.1 presents a list of all available inputs to MAVERICK v1.0, along with default values that are used if no other value is specified. Section 6.2 presents a list of all available outputs from MAVERICK v1.0, along with a description of the output file format.

6.1 List of input arguments/parameters

Inputs can be passed to MAVERICK as parameters in the `parameters.txt` file or as arguments on the command line (as described in section 2.1). Parameters and arguments are equivalent, except that arguments are prefixed with a ‘-’ character while parameters are not. For example, to set the ploidy level we could include the line `ploidy 2` in the `parameters.txt` file, where the name and the value are separated by a single tab character, or we could include the expression `-ploidy 2` on the command line after the execution command, where the name and the value are separated by a single space. All arguments that can be passed to MAVERICK v1.0 are listed in Table 1 (arguments are listed alphabetically).

Argument Name	Default Value	Description
<code>-admix_on</code>	<code>false</code>	Whether to use the admixture model.
<code>-alpha</code>	<code>1.0</code>	Value of admixture parameter α . This is the fixed value of α if <code>fixAlpha_on</code> is true, otherwise this is the initial value of α prior to updating in the MCMC. This parameter can either be a scalar value that applies to all K , or a comma-separated list of values with one entry per value of K explored.
<code>-alphaPropSD</code>	<code>0.1</code>	Standard deviation of the normal proposal distribution used to update α when <code>fixAlpha_on</code> is false. This parameter can either be a scalar value that applies to all K , or a comma-separated list of values with one entry per value of K explored.
<code>-data</code>	<code>data.txt</code>	Name of the input data file.
<code>-EMalgorithm_on</code>	<code>false</code>	Whether to calculate maximum likelihood parameter estimates using the EM algorithm. This option is required when producing model comparison statistics.
<code>-EMiterations</code>	<code>100</code>	Number of iterations in each run of the EM algorithm.
<code>-EMrepeats</code>	<code>10</code>	Number of times to repeat the EM algorithm, each time from a random starting location. Increasing this value increases the chance of arriving at the global maximum likelihood.
<code>-exhaustive_on</code>	<code>false</code>	Whether to calculate the exact model evidence using an exhaustive approach. Only possible for very small sample sizes.
<code>-fixAlpha_on</code>	<code>true</code>	Whether the admixture parameter α should be fixed or free to vary (only applies to the admixture model).
<code>-fixLabels_on</code>	<code>true</code>	Whether to solve label switching problem. This option is required to produce Q-matrix estimates.

Argument Name	Default Value	Description
<code>-headerRow_on</code>	<code>false</code>	Whether to ignore the first line when reading in data.
<code>-inputRoot</code>	(empty)	Sets the directory of all input files. Defined relative to <code>masterRoot</code> .
<code>-Kmax</code>	2	Upper value of K range to explore.
<code>-Kmin</code>	1	Lower value of K range to explore.
<code>-mainBurnin</code>	100	Number of iterations that will be discarded prior to sampling from the posterior distribution in the main MCMC. Burn-in iterations are not thinned.
<code>-mainRepeats</code>	1	Number of times to carry out the main MCMC algorithm. Values greater than 1 are needed to produce estimates of the standard error of the STRUCTURE estimator, the harmonic mean estimator, and the Q-matrix.
<code>-mainSamples</code>	1000	Number of samples (after thinning) to obtain from the posterior.
<code>-mainThinning</code>	1	Number of iterations needed to produce one draw from the posterior. Note that thinning is not usually necessary, as explained in section 4.4.
<code>-masterRoot</code>	(directory of executable)	Sets the global root directory. All other file paths are defined relative to this root.
<code>-missingData</code>	-9	Value to use as missing data. Character strings are allowed as well as numerical values.
<code>-outputComparisonStatistics</code>	<code>outputComparisonStatistics.csv</code>	Name of the output file containing standard model comparison statistics.
<code>-outputComparisonStatistics_on</code>	<code>false</code>	Whether to generate a file containing standard model comparison statistics.
<code>-outputEvanno</code>	<code>outputEvanno.csv</code>	Name of the output file containing Evanno's Δ_K statistic.
<code>-outputEvanno_on</code>	<code>false</code>	Whether to generate a file containing Evanno's Δ_K statistic. Values are not produced for the smallest or largest values of K explored, due to how this statistic is calculated.

Argument Name	Default Value	Description
<code>-outputEvidence</code>	<code>outputEvidence.csv</code>	Name of the output file containing estimates of the model evidence for each K .
<code>-outputEvidence_on</code>	<code>true</code>	Whether to generate a file containing estimates of the model evidence for each K .
<code>-outputEvidenceDetails</code>	<code>outputEvidenceDetails.csv</code>	Name of the output file containing further details of evidence calculations.
<code>-outputEvidenceDetails_on</code>	<code>false</code>	Whether to generate a file containing further details of evidence calculations, including the elements that make up the STRUCTURE estimator and the TI estimator.
<code>-outputEvidenceNormalised</code>	<code>outputEvidenceNormalised.csv</code>	Name of the output file containing normalised estimates of the model evidence.
<code>-outputEvidenceNormalised_on</code>	<code>true</code>	Whether to generate a file containing estimates of the model evidence that have been normalised to sum to 1 over all K .
<code>-outputLikelihood</code>	<code>outputLikelihood.csv</code>	Name of the output file containing likelihoods at each iteration of the main MCMC.
<code>-outputLikelihood_on</code>	<code>false</code>	Whether to generate a file containing likelihoods at each iteration of the main MCMC. Under the admixture model this file also gives values of α at each iteration.
<code>-outputLog</code>	<code>outputLog.txt</code>	Name of the output log file.
<code>-outputLog_on</code>	<code>true</code>	Whether to generate a log file (this is highly recommended).
<code>-outputMaxLike_admixFreqs</code>	<code>outputMaxLike_admixFreqs.csv</code>	Name of the output file containing maximum likelihood estimates of admixture proportions in all individuals.
<code>-outputMaxLike_admixFreqs_on</code>	<code>false</code>	Whether to generate a file containing maximum likelihood estimates of admixture proportions in all individuals.

Argument Name	Default Value	Description
<code>-outputMaxLike.alleleFreqs</code>	<code>outputMaxLike.alleleFreqs.csv</code>	Name of the output file containing maximum likelihood estimates of allele frequencies in all demes.
<code>-outputMaxLike.alleleFreqs_on</code>	<code>false</code>	Whether to generate a file containing maximum likelihood estimates of allele frequencies in all demes.
<code>-outputPosteriorGrouping</code>	<code>outputPosteriorGrouping.csv</code>	Name of the output file containing the complete posterior grouping at each iteration of the main MCMC.
<code>-outputPosteriorGrouping_on</code>	<code>false</code>	Whether to generate a file containing the complete posterior grouping at all iterations of the main MCMC.
<code>-outputQmatrix.gene</code>	<code>outputQmatrix.gene.csv</code>	Name of the Q-matrix output file at the gene copy level. The last four characters of the name (the <code>.csv</code> part) are automatically trimmed and replaced with <code>_K\$.csv</code> , where <code>\$</code> is the current value of <code>K</code> .
<code>-outputQmatrix.gene_on</code>	<code>false</code>	Whether to generate a Q-matrix output file at the gene copy level (only possible under the admixture model).
<code>-outputQmatrix.ind</code>	<code>outputQmatrix.ind.csv</code>	Name of the Q-matrix output file at the individual level. The last four characters of the name (the <code>.csv</code> part) are automatically trimmed and replaced with <code>_K\$.csv</code> , where <code>\$</code> is the current value of <code>K</code> .
<code>-outputQmatrix.ind_on</code>	<code>true</code>	Whether to generate a Q-matrix output file at the individual level.
<code>-outputQmatrix.pop</code>	<code>outputQmatrix.pop.csv</code>	Name of the Q-matrix output file at the population level. The last four characters of the name (the <code>.csv</code> part) are automatically trimmed and replaced with <code>_K\$.csv</code> , where <code>\$</code> is the current value of <code>K</code> .
<code>-outputQmatrix.pop_on</code>	<code>false</code>	Whether to generate a Q-matrix output file at the population level (only possible if <code>popCol_on</code> is true).

Argument Name	Default Value	Description
<code>-outputQmatrix_structureFormat_on</code>	<code>false</code>	When set to <code>true</code> additional Q-matrix files are produced in same format produced by STRUCTURE2.3.
<code>-outputQmatrixError_gene</code>	<code>outputQmatrixError_gene.csv</code>	Name of the standard error file for the Q-matrix at the gene copy level. The last four characters of the name (the <code>.csv</code> part) are automatically trimmed and replaced with <code>_K\$.csv</code> , where <code>\$</code> is the current value of <code>K</code> .
<code>-outputQmatrixError_gene_on</code>	<code>false</code>	Whether to generate a file containing standard error of the Q-matrix at the gene copy level (only possible under the admixture model).
<code>-outputQmatrixError_ind</code>	<code>outputQmatrixError_ind.csv</code>	Name of the standard error file for the Q-matrix at the individual level. The last four characters of the name (the <code>.csv</code> part) are automatically trimmed and replaced with <code>_K\$.csv</code> , where <code>\$</code> is the current value of <code>K</code> .
<code>-outputQmatrixError_ind_on</code>	<code>false</code>	Whether to generate a file containing standard error of the Q-matrix at the individual level.
<code>-outputQmatrixError_pop</code>	<code>outputQmatrixError_pop.csv</code>	Name of the standard error file for the Q-matrix at the population level. The last four characters of the name (the <code>.csv</code> part) are automatically trimmed and replaced with <code>_K\$.csv</code> , where <code>\$</code> is the current value of <code>K</code> .
<code>-outputQmatrixError_pop_on</code>	<code>false</code>	Whether to generate a file containing standard error of the Q-matrix at the population level (only possible if <code>popCol_on</code> is true).
<code>-outputRoot</code>	(empty)	Sets the directory of all output files. Defined relative to <code>masterRoot</code> .
<code>-parameters</code>	<code>parameters.txt</code>	Name of the input parameters file.
<code>-ploidy</code>	2	Ploidy of all individuals (data file should contain this many rows per individual). This parameter only applies if <code>ploidyCol_on</code> is false.

Argument Name	Default Value	Description
<code>-ploidyCol.on</code>	<code>false</code>	Whether the data file contains a column giving the ploidy of all individuals.
<code>-popCol.on</code>	<code>false</code>	Whether the data file contains a column giving the population of origin or sampling location of all individuals.
<code>-suppressWarning1.on</code>	<code>false</code>	Option to suppress warning that prevents user from running exhaustive approach on large data sets.
<code>-thermodynamic.on</code>	<code>true</code>	Whether to estimate the evidence for K via thermodynamic integration.
<code>-thermodynamicBurnin</code>	100	Number of iterations to discard prior to sampling from the power posterior when carrying out thermodynamic integration.
<code>-thermodynamicRungs</code>	21	Number of rungs (powers between 0 and 1) to use in thermodynamic integration procedure.
<code>-thermodynamicSamples</code>	1000	Number of samples (after thinning) to obtain from the power posterior when carrying out thermodynamic integration.
<code>-thermodynamicThinning</code>	1	Number of iterations needed to produce one draw from the power posterior. Note that thinning is not usually necessary, as explained in section 4.4.

Table 1: List of input arguments and default values.

6.2 List of output files

Quite a large number of outputs can be produced by MAVERICK v1.0. These outputs are turned on and off using the arguments above, and are written to the folder `masterRoot/outputRoot/`. In the case of `outputLikelihood.csv` and `outputPosteriorGrouping.csv` it is a good idea to turn off these outputs when they are not being used, as writing values to file in every iteration of the MCMC can take up space and cause the program to run slowly.

outputLog The log file contains a complete record of the analysis; including the paths of the input files, a complete list of parameter values used (and where they were specified), and some basic properties of the data. It is almost always a good idea to produce a log file.

outputLikelihood The likelihood file contains values of the joint and marginal likelihood at every iteration of the main MCMC, along with the value of the parameter α if the admixture model is being used. The main use of values in the likelihood file is for checking the behaviour of the MCMC (e.g. checking convergence and autocorrelation).

As described in section 4, the core MCMC algorithm works by updating the group allocation of all individuals (or all gene copies under the admixture model) with the unknown allele frequencies already integrated out of the problem. The column `loglike_marginal` gives this integrated likelihood in log space, and both the harmonic mean and the TI estimators are based on these values. However, for the purposes of computing the STRUCTURE estimator L_K we need access to a different expression – the likelihood conditional on the allele frequencies as well as the grouping. Therefore, when L_K is being calculated a set of allele frequencies is drawn from the conditional posterior (Dirichlet) distribution given the known grouping, and the joint likelihood of the grouping *and* the allele frequencies is given in the column `loglike_joint`.

A single likelihood file contains values for all K , all repetitions of the main MCMC, and all iterations within the main MCMC (burn-in iterations are indicated by values less than one in the ‘MCMCsample’ column). These values are produced for the main MCMC only – draws from the power posterior are not stored.

outputPosteriorGrouping The posterior grouping file is the most detailed form of output from MAVERICK v1.0, and is intended for those individuals that don’t feel comfortable with a program unless they can see exactly how it works! This file gives the group allocation of every individual (or every gene copy under the admixture model) for all iterations, all values of K , and all repeats of the main MCMC. When combined with the `outputLikelihood.csv` file this shows the complete workings of the main MCMC.

Producing this file is computationally intensive, and so this option should be turned off unless the output is explicitly needed.

outputQmatrix_ind and outputQmatrixError_ind The `outputQmatrix_ind.csv` files give the posterior allocation probability of each individual belonging to each deme. A different file is produced for each value of K explored, appended with the name `_K$.csv` where $\$$ is the current value of K . By default this file is produced in a simple comma separated format, with individuals in rows and demes in columns, however, the option `outputQmatrix_structureFormat_on` can be used to produce additional output in a format that is identical to that produced by STRUCTURE v2.3, meaning it can be used by the same downstream tools. The `outputQmatrixError_ind.csv` files give the statistical error associated with each Q-matrix estimate at the individual level. Values give standard errors; so, for example, a standard error of 0.01 means that our estimate of the posterior allocation probability could easily be out by 2% in either direction.

outputQmatrix_pop and outputQmatrixError_pop The `outputQmatrix_pop.csv` files give the posterior allocation probability at the population level. A different file is produced for each value of K explored, appended with the name `_K$.csv` where $\$$ is the current value of K . These values are obtained by averaging over the allocation probabilities of all individuals within a sampling location. As with individual level output, the argument `outputQmatrix_structureFormat_on` can be used to produce additional output in a format that is identical to that produced by STRUCTURE v2.3. The `outputQmatrixError_pop.csv` files give the statistical error associated with each Q-matrix estimate at the population level.

outputQmatrix_gene and outputQmatrixError_gene The `outputQmatrix_gene.csv` files give the posterior allocation probability at the level of individual gene copies, available under the admixture model only (see section 3.2). A different file is produced for each value of K explored, appended with the name `_K$.csv` where $\$$ is the current value of K . The `outputQmatrixError_gene.csv` files give the statistical error associated with these estimates. Both files are in long format, with every copy of every gene in every individual being in a different row, and every deme being in a different column.

outputEvidence The `outputEvidence.csv` file gives the raw value of the log-evidence produced by the various estimation methods being used. When

multiple repetitions of the main MCMC are used, each will result in a different estimate of the log-evidence. These are given individually, and are combined in the form of a grand mean and grand standard error. For the TI estimate there is no grand mean, as only a single repetition of the main MCMC is used (or more accurately, multiple chains are already combined to produce one estimate). This still results in a single point estimate and standard error.

Values that are not calculated are shown as **NA**. For example, if **exhaustive_on** is set to false then the exact evidence will be shown as **NA** for all values of K greater than 1 (for $K = 1$ the exact evidence is always calculated, as this calculation is quick and easy).

outputEvidenceNormalised The normalised evidence file contains transformed estimates of the model evidence that have been exponentiated and then normalised to sum to 1 over all K . In other words, these values give the posterior distribution of K as estimated by the various methods.

Transforming estimates in this way is non-trivial, as the error in the original estimates must be dealt with carefully. MAVERICK v1.0 uses a simulation-based method to get around this. For all values of K considered, a random value is simulated by drawing from a normal distribution with mean and standard deviation equal to the point estimate and standard error of the estimate in log space (this approach makes a standard assumption that the distribution of the test statistic is approximately normal). These values are then exponentiated and normalised to sum to 1. This process is repeated 10^6 times, resulting in 10^6 draws from the posterior distribution in which the correlations introduced by normalising have automatically been taken into account. The mean and 95% upper and lower limits (denoted UL and LL) of this distribution are then calculated for each K , resulting in the final values in the normalised evidence file.

It is important to note that **values in the outputEvidenceNormalised.csv file are only relevant for the range of K considered in the original model run**. Thus, for example, it is not correct to calculate normalised estimates for $K = \{1, 2, 3\}$ and compare these against normalised estimates for $K = \{4, 5, 6\}$. In order to make this comparison the raw values of both runs should be combined before normalising, so that the new posterior distribution is defined over $K = \{1, \dots, 6\}$.

outputEvidenceDetails The evidence details file gives further details of the individual elements that make up estimates of the model evidence, as obtained via the STRUCTURE method and via TI. The STRUCTURE heuristic estimator L_K is calculated from the mean and variance of the log-likelihood of the posterior draws. These values are given in this file for each repetition of the main MCMC. The TI estimator is calculated from a series of points, each of which pertains to a particular rung of the power posterior. Estimates for all rungs are given in this file. It is a good idea to plot the `TIpoint_mean` values, as in Figure 5, to ensure that the path is relatively smooth – if it is too jagged then the TI estimator will be biased.

outputComparisonStatistics The comparison statistics file gives values of the AIC, BIC and DIC – calculated by the method of Spiegelhalter et al. (2002) (DIC_S) and Gelman et al. (2014) (DIC_G) – for all values of K .

outputEvanno In addition to the model comparison statistics given above, MAVERICK v1.0 can produce values of Evanno’s Δ_K statistic (Evanno et al., 2005) calculated over multiple repetitions of the main MCMC loop. Δ_K is calculated from second-order differences between L_K values, meaning values are not produced for $K=1$ or $K=Kmax$.

outputMaxLike_alleleFreqs and outputMaxLike_admixFreqs As part of the AIC, BIC and DIC calculations MAVERICK v1.0 carries out maximum likelihood estimation of allele frequencies and admixture proportions. This is achieved via an expectation-maximisation (EM) algorithm, in which parameters are iteratively and alternately updated to their conditional maximum likelihood estimates given everything else. The algorithm is repeated multiple times from random starting conditions to mitigate the chance of converging on a local rather than global optimum. The `outputMaxLike_alleleFreqs.csv` and `outputMaxLike_admixFreqs.csv` files give these maximum likelihood values in simple matrix format, broken down by deme, locus and allele in the case of allele frequencies, and by individual and deme in the case of admixture proportions.

References

- Jean Diebolt and Christian P Robert. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 363–375, 1994.
- Guillaume Evanno, Sebastien Regnaut, and Jérôme Goudet. Detecting the number of clusters of individuals using the software structure: a simulation study. *Molecular ecology*, 14(8):2611–2620, 2005.
- Daniel Falush, Matthew Stephens, and Jonathan K Pritchard. Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, 164(4):1567–1587, 2003.
- Daniel Falush, Matthew Stephens, and Jonathan K Pritchard. Inference of population structure using multilocus genotype data: dominant markers and null alleles. *Molecular ecology notes*, 7(4):574–578, 2007.
- Alan E Gelfand and Adrian FM Smith. Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association*, 85(410):398–409, 1990.
- Andrew Gelman, John B Carlin, Hal S Stern, and Donald B Rubin. *Bayesian data analysis*, volume 2. Taylor & Francis, 2014.
- Melissa J Hubisz, Daniel Falush, Matthew Stephens, and Jonathan K Pritchard. Inferring weak population structure with the assistance of sample group information. *Molecular ecology resources*, 9(5):1322–1332, 2009.
- John P Huelsenbeck and Peter Andolfatto. Inference of population structure under a dirichlet process model. *Genetics*, 175(4):1787–1802, 2007.
- Nicolas Lartillot and Hervé Philippe. Computing bayes factors using thermodynamic integration. *Systematic biology*, 55(2):195–207, 2006.
- Radford M Neal. Contribution to the discussion of “Approximate Bayesian inference with the weighted likelihood bootstrap” by Michael A. Newton and Adrian E. Raftery. *Journal of the Royal Statistical Society, Series B*, 56(1):41–42, 1994.

- Michael A Newton and Adrian E Raftery. Approximate bayesian inference with the weighted likelihood bootstrap. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 3–48, 1994.
- Jonathan K Pritchard, Matthew Stephens, and Peter Donnelly. Inference of population structure using multilocus genotype data. *Genetics*, 155(2): 945–959, 2000.
- Adrian E Raftery, Michael A Newton, Jaya M Satagopan, and Pavel N Krivitsky. Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. 2006.
- Bruce Rannala and Joanna L Mountain. Detecting immigration by using multilocus genotypes. *Proceedings of the National Academy of Sciences*, 94 (17):9197–9201, 1997.
- Daniel Sorensen and Daniel Gianola. *Likelihood, Bayesian, and MCMC methods in quantitative genetics*. Springer Science & Business Media, 2002.
- David J Spiegelhalter, Nicola G Best, Bradley P Carlin, and Angelika Van Der Linde. Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(4):583–639, 2002.
- Matthew Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4): 795–809, 2000.