BIOINFORMATICS ORIGINAL PAPER Vol. 23 no. 14 2007, pages 1801–1806
doi:10.1093/bioinformatics/btm233

Genetics and population analysis

CLUMPP: a cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure

Mattias Jakobsson* and Noah A. Rosenberg

Center for Computational Medicine and Biology, Department of Human Genetics, University of Michigan, Ann Arbor, MI, USA

Received on December 28, 2006; revised on March 14, 2007; accepted on April 25, 2007

Advance Access publication May 7, 2007

Associate Editor: Keith Crandall

ABSTRACT

Motivation: Clustering of individuals into populations on the basis of multilocus genotypes is informative in a variety of settings. In population-genetic clustering algorithms, such as *BAPS*, *STRUCTURE* and *TESS*, individual multilocus genotypes are partitioned over a set of clusters, often using unsupervised approaches that involve stochastic simulation. As a result, replicate cluster analyses of the same data may produce several distinct solutions for estimated cluster membership coefficients, even though the same initial conditions were used. Major differences among clustering solutions have two main sources: (1) 'label switching' of clusters across replicates, caused by the arbitrary way in which clusters in an unsupervised analysis are labeled, and (2) 'genuine multimodality,' truly distinct solutions across replicates.

Results: To facilitate the interpretation of population-genetic clustering results, we describe three algorithms for aligning multiple replicate analyses of the same data set. We have implemented these algorithms in the computer program *CLUMPP* (CLUster Matching and Permutation Program). We illustrate the use of *CLUMPP* by aligning the cluster membership coefficients from 100 replicate cluster analyses of 600 chickens from 20 different breeds.

Availability: *CLUMPP* is freely available at http://rosenberglab.bioinformatics.med.umich.edu/clumpp.html

Contact: mjakob@umich.edu

1 INTRODUCTION

A variety of population-genetic applications—such as association mapping, molecular ecological studies and studies of human evolution—make use of the clustering of individual multilocus genotypes into populations. Many clustering algorithms have now been developed for employing population-genetic data to assign individuals—and fractions of individuals—to clusters (Anderson and Thompson, 2002; Chen *et al*., 2006, 2007; Corander and Marttinen, 2006; Corander *et al*., 2003, 2004; Dawson and Belkhir, 2001; Falush *et al*., 2003; François *et al*., 2006; Huelsenbeck and Andolfatto, 2007; Pella and Masuda, 2006; Pritchard *et al*., 2000). The result of a single cluster analysis is typically possible to represent as a matrix, where each individual is given a 'membership coefficient' for each cluster—interpreted as a probability of membership, or as a fraction of the genome with membership in the cluster, depending on the setting—with membership coefficients summing to 1 across K clusters. The number of clusters is predefined by the user for some methods, and for others it is inferred.

Because clustering algorithms may incorporate stochastic simulation as part of the inference, independent analyses of the same data may result in several distinct outcomes, even though the same initial conditions were used. The main differences across replicates are of two types: 'label switching' and 'genuine multimodality'. 'Label switching' refers to a scenario in which different replicates obtain the same membership coefficient estimates, except with a different permutation of the cluster labels (Jasra *et al*., 2005; Stephens, 2000). In unsupervised cluster analyses, because the meaning of each cluster label is not known in advance, a clustering algorithm may be equally likely to reach any of $K!$ permutations of the same collection of estimated membership coefficients.

It is also possible for replicate cluster analyses to arrive at truly distinct solutions that are not equivalent up to permutation. Unlike 'label switching', which is simply a nuisance that makes interpretation of the clustering results more difficult, this 'genuine multimodality' may result from real biological factors that cause multiple parts of the space of possible membership coefficients to provide similarly appropriate explanations for the data.

Regardless of the source of differences in clustering outcomes, some method is needed for handling the results from replicate analyses. We develop three algorithms for searching for optimal alignments of R replicate cluster analyses of the same data, and we have implemented these algorithms in the computer program *CLUMPP*. Our program takes as input the estimated cluster membership coefficient matrices of multiple runs of a clustering program, for any number of clusters. It outputs these same matrices, permuted so that all replicates have as close a match as possible. Thus, *CLUMPP* strips away the 'label switching' heterogeneity so that the 'genuine multimodality' can be detected and quantified. The input file for

*To whom correspondence should be addressed.

CLUMPP is a file similar to the output from *STRUCTURE* (Falush *et al*., 2003; Pritchard *et al*., 2000), and the output from *CLUMPP* can be used directly as input by the cluster visualization program *DISTRUCT* (Rosenberg, 2004).

2 ALGORITHMS

We refer to the $C\times K$ matrix of membership coefficients for a single cluster analysis as the *Q-matrix*, with the C rows corresponding to individuals (or populations) and the K columns corresponding to clusters. *CLUMPP* attempts to maximize a measure of similarity of the Q-matrices of R creplicates over all $(K!)^{R-1}$ possible alignments of the replicates.

Consider a pair of Q-matrices, Q_i and Q_j for runs i and j, where the value in the cth row and kth column of Q_i is the membership coefficient for individual c in cluster k as inferred in run i. Each matrix consists of non-negative entries, and the sum of the entries in any row is 1. We define the pairwise similarity of matrices Q_i and Q_j as follows:

$$G(Q_i, Q_j) = 1 - \frac{\|Q_i - Q_j\|_F}{\sqrt{\|Q_i - W\|_F \|Q_j - W\|_F}}. \quad (1)$$

In this equation, W is a $C\times K$ matrix with all elements equal to $1/K$ and $\|\cdot\|_F$ is the Frobenius matrix norm (Golub and Van Loan, 1996).

$$\|A\|_F = \sqrt{\sum_{c=1}^{C}\sum_{k=1}^{K} a_{ck}^2}, \quad (2)$$

where C and K respectively denote the numbers of rows and columns of A, and a_{ck} is the value in row c and column k.

Using G to measure similarity, the optimal alignment of matrices Q_i and Q_j is defined as the permutation of the columns of Q_j that maximizes the similarity G over all permutations P in the set S_K of permutations of K clusters. The maximum value, or

$$\mathrm{SSC}(Q_i, Q_j) = \max_{P\in S_K} G(Q_i, P(Q_j)), \quad (3)$$

is the quantity named by Nordborg *et al*. (2005) the 'symmetric similarity coefficient' (SSC) of the pair of runs [see also Rosenberg *et al*., (2002) for an earlier statistic]. The SSC for two runs is bounded above by 1—which it equals if the Q-matrices are identical up to a permutation of the clusters—and it decreases as the similarity of the Q-matrices decreases. The SSC statistic is generally expected to be positive if non-trivial clustering patterns are present in Q_i and Q_j, although it is possible for it to be negative.

For a collection of R replicates, the average pairwise similarity is defined as

$$H(Q_1, Q_2, \ldots, Q_R) = \frac{2}{R(R-1)}\sum_{i=1}^{R-1}\sum_{j>i}^{R} G(Q_i, Q_j). \quad (4)$$

To find the optimal alignment of R replicates, we search for the vector of permutations that maximizes this average pairwise similarity:

$$\mathrm{SSC}_R(Q_1, Q_2, \ldots, Q_R) = \max_{(P_1,P_2,\ldots,P_R)\in S_K^R} H(P_1(Q_1), P_2(Q_2), \ldots, P_R(Q_R)). \quad (5)$$

Without loss of generality, we take P_1 to be the identity permutation I, so that the clusters of runs 2,...,R are permuted to align with the clusters of run 1. As S_K contains $K!$ permutations, with P_1 set to equal I, the maximum in Equation (5) is taken over $(K!)^{R-1}$ vectors.

We make use of three algorithms for attempting to find the optimal alignment of R replicates. In decreasing order of the extent of the search, and in increasing order of computational speed, these algorithms are termed *FullSearch*, *Greedy* and *LargeKGreedy*. These algorithms supersede earlier methods that we described in Nordborg *et al*. (2005).

Note that our approach can proceed analogously using alternative functions to measure similarity in place of G. Although it is undefined when one of the two matrices equals W, G is designed to have large negative values when one of the two runs reflects substantial population structure and the other has relatively little structure (i.e. little difference from W). We can define a second similarity function G', which is guaranteed to lie in [0,1]:

$$G'(Q_i, Q_j) = 1 - \frac{\|Q_i - Q_j\|_F}{\sqrt{2C}}. \quad (6)$$

The normalization constant $\sqrt{2C}$, which guarantees that G' lies in [0,1], arises from the definition of the Frobenius norm:

$$\|A - B\|_F = \sqrt{\sum_{c=1}^{C}\left(\sum_{k=1}^{K} a_{ck}^2 + b_{ck}^2 - 2a_{ck}b_{ck}\right)}.$$

If A and B have non-negative entries and row sums of 1, then $-2a_{ck}b_{ck}\leq 0$ and $\sum_{k=1}^{K} a_{ck}^2 = (\sum_{k=1}^{K} a_{ck})^2 - 2\sum_{k=1}^{K-1}\sum_{\ell>k}^{K} a_{ck}a_{c\ell} = 1 - 2\sum_{k=1}^{K-1}\sum_{\ell>k}^{K} a_{ck}a_{c\ell} \leq 1$. Similarly, $\sum_{k=1}^{K} b_{ck}^2 \leq 1$. It then follows that $\|A - B\|_F \leq \sqrt{2C}$.

The quantities SSC′, H' and SSC'_R can then be defined by replacing G in Equations (3–5) with G'. We proceed to describe our algorithms using the G statistic to measure similarity; to instead use G', the approach is analogous with G', SSC′, H' and SSC'_R in place of G, SSC, H, and SSC_R.

FullSearch

The *FullSearch* algorithm computes H for each of the $(K!)^{R-1}$ alignments of the K clusters in R replicates. Considering all possible vectors of permutations (I, P_2, P_3,..., P_R), the algorithm computes $H(I(Q_1),P_2(Q_2),P_3(Q_3),\ldots,P_R(Q_R))$ and returns the vector of permutations that maximizes SSC_R. As the number of possible vectors grows quickly with K and R, however, even for moderate values of K and R, it is unrealistic to test every alignment. The *FullSearch* algorithm runs in time proportional to $T_{FullSearch} = (K!)^{R-1}[R(R-1)/2]KC$: the number of permutation vectors is $(K!)^{R-1}$, the number of computations of G in each evaluation of Equation (4) is $R(R-1)/2$, and the time required for each computation of G is proportional to KC. Although it proceeds slowly, unlike our other algorithms, the *FullSearch* algorithm is guaranteed to find the optimal alignment of clusters across multiple runs.

Greedy

The *Greedy* algorithm employed by *CLUMPP* proceeds as follows:

(1) Choose one run, Q_1.

(2) Choose a second run, Q_2, and fix the permutation, P_2, that maximizes $G(P_1(Q_1),P(Q_2))$ over all possible permutations P (where P_1 is the identity permutation).

(3) Continue sequentially with each remaining run, Q_x, where $x=3,\ldots,R$, and fix the permutation, P_x of Q_x, that maximizes the average similarity with the previously fixed $x-1$ runs, or

$$J(P_1(Q_1), P_2(Q_2), \ldots, P_{x-1}(Q_{x-1}), P(Q_x)) = \frac{1}{x-1}\sum_{\ell=1}^{x-1} G(P_\ell(Q_\ell), P(Q_x)), \quad (7)$$

over all permutations P.

This algorithm runs in time proportional to $T_{Greedy}=(K!)[R(R-1)/2]KC$. The number of permutations tested for each run from 2 to R is $K!$. For run r (r ranging from 2 to R), the number of computations of G performed for each permutation is $r-1$. Thus, considering all runs from 2 to R, the total number of computations of G performed is $(K!)[R(R-1)]/2$. Each computation of G runs in time proportional to KC.

Because the order in which the runs are considered can affect the result, several different sequences of runs should be tested. *CLUMPP* offers three options for testing different sequences: test all possible sequences of runs, test a pre-defined number of random sequences and test a specific set of user-defined sequences.

LargeKGreedy

When $K \gtrsim 15$, the number of permutations, $K!$, is very large, and it may not be possible to calculate G for all permutations of a particular pair of Q-matrices. Instead of testing every permutation as in the *Greedy* algorithm, the *LargeKGreedy* algorithm proceeds as follows:

(1) Choose one run, Q_1.

(2a) Choose a second run, Q_2. Compute G for all pairs of columns, one from Q_1 and one from Q_2. This computation is simply the value of G for two columns—hence no permutations of Q_2 are computed, unlike in step 2 for the *Greedy* algorithm.

(2b) Pick the pair of columns Q_{1,y_1} and Q_{2,z_1} with highest G-value and fix these columns (Q_{1,y_1} refers to column y_1 of matrix Q_1). Then pick the pairs of columns Q_{1,y_2} and Q_{2,z_2} with the next highest G-value, ignoring all G-values of pairs containing either of the previously chosen columns Q_{1,y_1} and Q_{2,z_1}. Repeat this procedure until K pairs of columns, one each from Q_1 and Q_2, have been picked, and fix the permutation of Q_2 that matches up these pairs of columns, or $P_2(Q_2)$.

(3a) Continue sequentially with each remaining run, Q_x, where $x = 3,\ldots, R$. For each y and z from 1 to K, compute the average similarity,

$$J(P_{1,y}(Q_1), \ldots, P_{x-1,y}(Q_{x-1}), Q_{x,z})) = \frac{1}{x-1}\sum_{\ell=1}^{x-1} G(P_{\ell,y}(Q_\ell), Q_{x,z})), \quad (8)$$

where $P_{\ell,y}$ denotes column y of the permuted matrix $P_\ell(Q_\ell)$. This quantity is the similarity of column z of Q_x to column y of each of the previously fixed permutations, averaged across all runs previously considered. No permutations of Q_x are computed, unlike in Step 3 for the *Greedy* algorithm.

(3b) Pick the pair of columns y_1 of $P_1(Q_1)$, $P_2(Q_2),\ldots,P_{x-1}(Q_{x-1})$ and z_1 of Q_x with highest average similarity in Equation (8). Then pick the columns y_2 and z_2 with the next highest similarity in Equation (8), ignoring similarity scores of pairs containing either of the previously chosen columns y_1 and z_1. Repeat this procedure until K pairs of columns, one for the matrices $P_1(Q_1),P_2(Q_2),\ldots,P_{x-1}(Q_{x-1})$ and one for Q_x, have been picked. Fix the permutation of Q_x that matches up these pairs of columns, or $P_x(Q_x)$.

A candidate for the vector of permutations of the R runs that maximizes H across all possible vectors has now been constructed. This algorithm runs in time proportional to $T_{LargeKGreedy}=[R(R-1)/2]K^2C$. The number of pairs of columns, one from the run under consideration and one from the previously fixed runs, is K^2. For run r (r ranging from 2 to R), the number of computations of G performed for each pair of columns is $r-1$. Considering all runs from 2 to R, the total number of computations of G performed is $K^2[R(R-1)]/2$. Since G is computed only for columns rather than for whole matrices, the time of computation of G is proportional only to C rather than to KC, as in the other algorithms.

As is true for the *Greedy* algorithm, the order in which the runs are considered can affect the result. For the *LargeKGreedy* algorithm *CLUMPP* offers the same three options for selecting the input sequence of runs as it provides for the *Greedy* algorithm.

To get an idea of which algorithm to use, we have found it useful to compute the quantity $D=TN$ for each algorithm, where T is a quantity proportional to the time required by an algorithm (as described above) and N is the number of input sequences to be tested (for *FullSearch*, $N=1$). If $D \lesssim 10^{13}$ for *FullSearch*, then this algorithm is fast enough and is preferred; otherwise the *Greedy* algorithm can be used. If $D \gtrsim 10^{13}$ for the *Greedy* algorithm, then this algorithm is probably also too slow. In that case, the *LargeKGreedy* algorithm should be used, as it can handle $K>20$, $R>100$ and $C>1000$ in reasonable time. We recommend testing at least 100 input sequences for the *Greedy* and *LargeKGreedy* algorithms—and preferably many more—to find the approximately highest SSC_R value.

Example

Rosenberg *et al.* (2001) analyzed a data set of 27 microsatellites genotyped in 600 chickens from 20 breeds.

Using *STRUCTURE* (Pritchard *et al.*, 2000), they inferred membership coefficients of the 600 chickens in 19 clusters. By repeating the cluster analysis 100 times, they assessed the robustness of the clusters, and Figure 1A displays the cluster membership estimates from 25 of the 100 replicates (randomly chosen). From Figure 1A, the 'label-switching' problem is apparent. Focusing on Runs 15 and 16, we could quickly 'match up' the clusters (i.e. the colored segments). However, because of some genuine multimodality, it would be not only tedious, but also difficult, to visually align the clusters for *all* replicates in an optimal manner. *CLUMPP* is designed to address exactly this type of situation.

Using *CLUMPP* and the *LargeKGreedy* algorithm on the 100 replicates (10 sets of 100 random input sequences of runs—for each set this takes ~6 min on a 2.4 GHz processor), we find the highest value of H to be in the range of 0.51–0.54 (and the highest value of H' to be between 0.69 and 0.70). If we allow the algorithm to run longer (30 000 random input sequences), the highest H equals 0.5546. The highest-scoring input sequences tend to produce quite similar alignments of the replicates. Excluding the input sequence that produced the highest H-value, the next 10 highest-scoring input sequences all produced H-values of at least 0.5441, and had on average 2.0% differences compared to the input sequence that led to the highest H. In other words, considering the permuted position of a randomly chosen cluster (among 19) in a randomly chosen run (among 100) based on the output of *CLUMPP* using a randomly chosen input sequence (among the 10 next

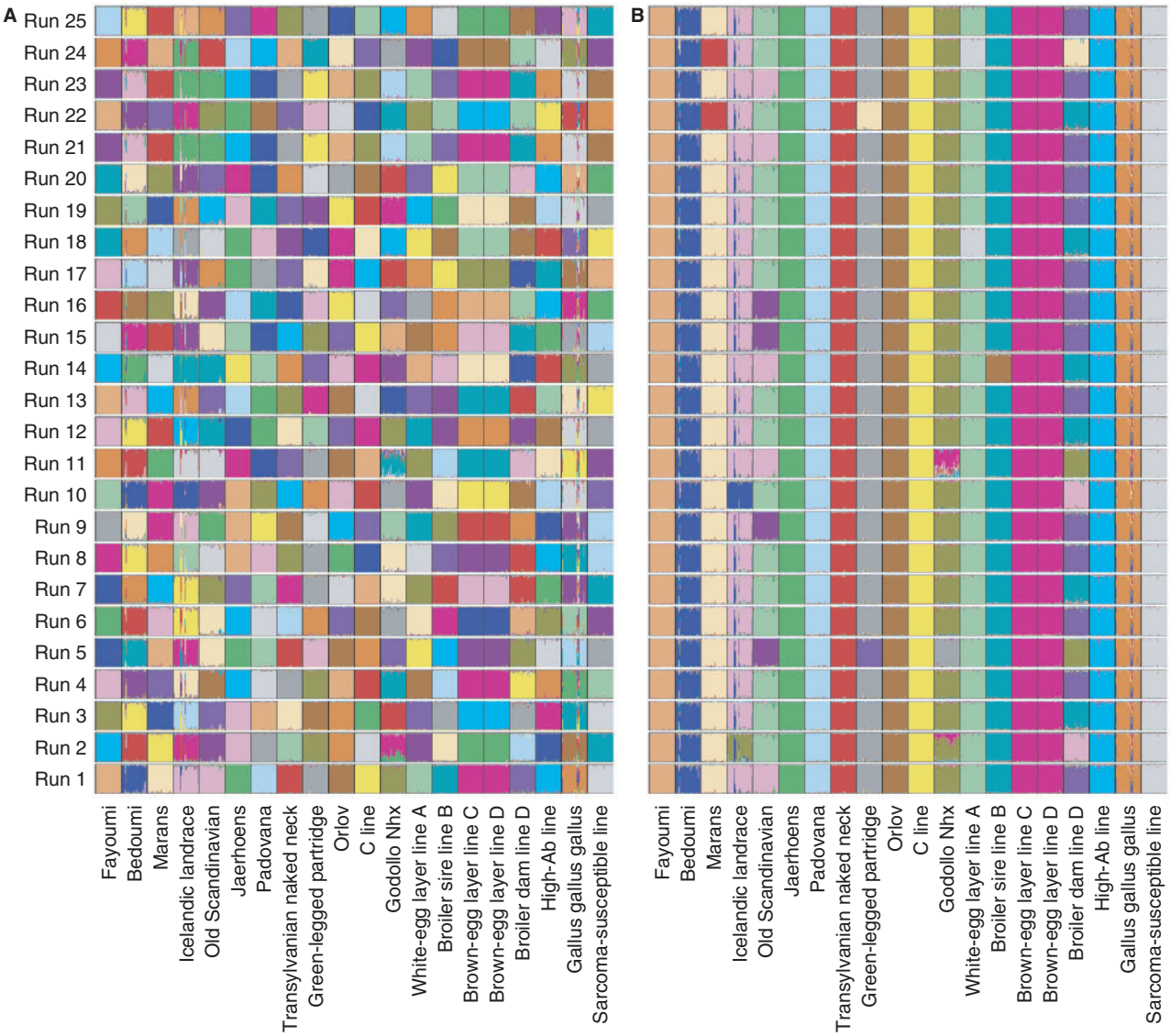

Fig. 1. The membership coefficients (Q-matrices) for 600 chickens, pre-*CLUMPP* and post-*CLUMPP*. The matrices are visualized using *DISTRUCT*. (**A**) Membership coefficients from the first 25 runs of *STRUCTURE* from a total of 100 runs performed by Rosenberg *et al.* (2001). Each individual is represented by a vertical line partitioned into 19 colored segments corresponding to its membership coefficients in 19 clusters. Each color represents a different cluster, and black lines separate the individuals of different breeds. (**B**) The same membership coefficients as in A, permuted using *CLUMPP* so that cluster labels match across runs.

highest-scoring sequences, after the one with the highest H-value), the cluster had a 98.0% chance of being aligned in the same way that it was aligned when using the highest-scoring of all input sequences.

The permuted membership coefficients of the 25 runs in Figure 1A for the input sequence among the 30 000 that leads to the highest H are shown in Figure 1B. Comparing the permuted membership coefficients for this input sequence to the clustering behavior in Table 2 of Rosenberg *et al.* (2001), we recapture the same patterns of clustering. For example, Brown-egg layer lines C and D always share a cluster, and this cluster sometimes (in 8 of 100 runs) includes Godollo Nhx (Run 11 in Fig. 1).

When we consider the *LargeKGreedy* algorithm with 10 000 fixed input sequences (chosen randomly among the 30 000 described above), for each of the 10 000 sequences, the alignment of the replicates obtained by *CLUMPP* is identical regardless of which of two statistics—H or H'—is used. The same input sequence that produces the highest H-value (H=0.5508) produces the highest H'-value (H'=0.7099). Excluding the input sequence that produced the highest values of H and H', the next 10 highest-scoring input sequences—the same sequences for both statistics—all produced H-values of at least 0.5392 and H'-values of at least 0.7022, and had on average 7.0% differences compared to the input sequence that led to the highest H and H'. These results suggest that although matrices can be constructed so that the two statistics can lead to different alignments, in practice, their properties are extremely similar. The larger number of differences for the highest-scoring input sequences from among 10 000 sequences compared to the highest-scoring among the 30 000 sequences described above highlights the importance of employing a large number of input sequences whenever possible.

3 DISCUSSION

Interpreting the results of multiple population genetic cluster analyses is not always straightforward (Corander *et al.*, 2004; Evanno *et al.*, 2005; Pritchard *et al.*, 2000; Rosenberg *et al.*, 2001). One could argue that the single replicate with the highest likelihood (according to the criterion of the clustering program) is the optimal clustering solution, and that solutions with lower likelihood can be discarded. However, for complex data sets that produce several distinct solutions with similar likelihoods, this strategy may not be satisfactory, as it disregards both uncertainties of the analysis and important population structure information that may only be visible in different modes. From a Bayesian perspective, a possible approach would be to weight different solutions by their likelihoods. However, this approach can be difficult to apply with individual membership coefficients in the case of genuine multimodality. A third approach is to summarize data on the outcomes of many replicates, such as in Table 2 of Rosenberg *et al.* (2001). Regardless of the choice of procedure for employing multiple cluster analyses to make biological interpretations from population-genetic data, *CLUMPP* can provide a useful way of assessing the similarity of the outcomes of individual runs.

We note that our focus here has been on the setting in which the clustering algorithm explores only one mode during the iterative process that constitutes a single replicate analysis, but may explore different modes in different replicates. As was recognized by Pritchard *et al.* (2000), this setting typically applies to the *STRUCTURE* approach, for which the results of individual replicates are reasonably straightforward to summarize and interpret, but for which difficulties may arise in interpreting differences across replicates. It is possible to envision an alternative scenario in which multiple important modes (potentially including multiple permutations of each of the genuinely distinct modes) are explored during the course of a single analysis. In this situation, replicate analyses would produce similar results, but because the meaning of a cluster label could vary across intermediate steps in the analysis, it would be challenging to summarize the outcome of any single replicate. Although our analysis has used the language of multiple replicate analyses rather than that of multiple intermediate steps of a single analysis, it is worth noting that the algorithms we have proposed for the matching of clusters can also be applied with each 'replicate' described above corresponding to a specific state of the membership coefficient estimates encountered during the course of a single analysis. Thus, in addition to their uses in interpreting multiple replicate analyses, our algorithms can augment existing approaches to the analysis of individual replicates (e.g. Dawson and Belkhir, 2001; Huelsenbeck and Andolfatto, 2007) and can contribute new approaches for the problem of summarizing the intermediate steps of a single run.

ACKNOWLEDGEMENTS

We thank O. François and two anonymous reviewers for helpful comments on a previous version of the manuscript. This work was supported by NIH grant HL084729, by grants from the Burroughs Wellcome Fund and the Alfred P. Sloan Foundation, and by a postdoctoral fellowship to M.J. from the University of Michigan Center for Genetics in Health and Medicine.

Conflict of Interest: none declared.

REFERENCES

Anderson,E.C. and Thompson,E.A. (2002) A model-based method for identifying species hybrids using multilocus genetic data. *Genetics*, **160**, 1217–1229.

Chen,C. *et al.* (2006) FASTRUCT: model-based clustering made faster. *Mol. Ecol. Notes*, **6**, 980–983.

Chen,C. (2007) Bayesian clustering algorithms ascertaining spatial population structure: a new computer program and a comparison study. *Mol. Ecol. Notes*, doi: 10.1111/j.1471-8286.2007.01769.x.

Corander,J. and Marttinen,P. (2006) Bayesian identification of admixture events using multilocus molecular markers. *Mol. Ecol.*, **15**, 2833–2843.

Corander,J. *et al.* (2003) Bayesian analysis of genetic differentiation between populations. *Genetics*, **163**, 367–374.

Corander,J. *et al.* (2004) BAPS 2: enhanced possibilities for the analysis of genetic population structure. *Bioinformatics*, **20**, 2363–2369.

Dawson,K.J. and Belkhir,K. (2001) A Bayesian approach to the identification of panmictic populations and the assignment of individuals. *Genet. Res.*, **78**, 59–77.

Evanno,G. *et al.* (2005) Detecting the number of clusters of individuals using the software STRUCTURE: a simulation study. *Mol. Ecol.*, **14**, 2611–2620.

Falush,D. *et al.* (2003) Inference of population structure using multilocus genotype data: linked loci and correlated allele frequencies. *Genetics*, **164**, 1567–1587.

François,O. *et al.* (2006) Bayesian clustering using hidden Markov random fields in spatial population genetics. *Genetics*, **174**, 805–816.

Golub,G.H. and Van Loan,C.F. (1996) *Matrix Computations*. 3rd edn. Johns Hopkins University Press, Baltimore.

Huelsenbeck,J.P. and Andolfatto,P. (2007) Inference of population structure under a Dirichlet process model. *Genetics*, **175**, 1787–1802.

Jasra,A. *et al.* (2005) Markov chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Stat. Sci.*, **20**, 50–67.

Nordborg,M. *et al* (2005) The pattern of polymorphism in *Arabidopsis thaliana*. *PLoS Biol.*, **3**, 1289–1299.

Pella,J. and Masuda,M. (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. *et al.* (2000) Inference of population structure using multilocus genotype data. *Genetics*, **155**, 945–959.

Rosenberg,N.A. (2004) DISTRUCT: a program for the graphical display of population structure. *Mol. Ecol. Notes*, **4**, 137–138.

Rosenberg,N.A. *et al.* (2001) Empirical evaluation of genetic clustering methods using multilocus genotypes from 20 chicken breeds. *Genetics*, **159**, 699–713.

Rosenberg,N.A. *et al.* (2002) Genetic structure of human populations. *Science*, **298**, 2381–2385.

Stephens,M. (2000) Dealing with label switching in mixture models. *J. R. Stat. Soc. B*, **62**, 795–809.