Molecular Ecology Notes (2006) **6**, 980–983 doi: 10.1111/j.1471-8286.2006.01527.x

PROGRAM NOTE

FASTRUCT: model-based clustering made faster

CHIBIAO CHEN,* FLORENCE FORBES* and OLIVIER FRANÇOIS†
**INRIA Equipe MISTIS, 655 Avenue de l'Europe, 38334 St Ismier France, †TIMC-TIMB, Dept Math Biology, Faculty of Medicine, F38706 La Tronche, France*

Abstract

Bayesian model-based clustering programs have gained increased popularity in studies of population structure since the publication of the software STRUCTURE. These programs are generally acknowledged as performing well, but their running-time may be prohibitive. FASTRUCT is a non-Bayesian implementation of the classical model with no-admixture uncorrelated allele frequencies. This new program relies on the expectation–maximization principle, and produces assignment rivalling other model-based clustering programs. In addition, it can be manyfold faster than Bayesian implementations. The software consists of a command-line engine, which is suitable for batch analysis of data, and a graphical interface, which is convenient for exploring data.

Keywords: assignment, clustering, EM algorithm, graphical user interface, population genetic structure

Received 20 May 2006; revision accepted 10 July 2006

In recent years, assignment methods combined with Bayesian clustering have been the subject of increasing interest to biologists studying population genetic structure, mainly because these approaches do not assume predefined subpopulations. These methods attempt to group samples into clusters of random mating individuals so that the Hardy–Weinberg (HW) and linkage disequilibria are minimized across the data set. They also provide a sound basis to analyse and interpret multilocus genotype data (Pritchard *et al*. 2000; Dawson & Belkhir 2001; Corander *et al*. 2003). Nevertheless, although these methods are robust and generally perform well to various sources of data, they share the drawback of being implemented through Markov chain Monte Carlo (MCMC) algorithms. MCMC implementations require setting many parameters, and generate slow-converging computer programs with no known reliable stopping rules. Usually, Bayesian methods are contrasted to distance-based clustering methods. Distance-based methods are simpler and often very fast, but they are heuristic, and generally lack the theoretical supports that make model-based methods so powerful. These less-elaborated methods are often considered more suited to exploratory analysis than to fine statistical inference.

Correspondence: Chibiao Chen, Fax: (+ 33) 04 76 61 52 52; E-mail: chen.chibiao@gmail.com

In this note, we propose a new algorithm (FASTRUCT) that cumulates two benefits: it is model-based and it is fast. This goal is achieved by solving the inference problem proposed by Pritchard *et al*. (2000) using the expectation-maximization (EM) algorithm (Dempster *et al*. 1977) instead of MCMC. The EM algorithm is coupled with a robust phylogenetic method — neighbour-joining (Saitou & Nei 1987) — to generate fast solutions. As do STRUCTURE and BAPS, FASTRUCT allows individuals to be of mixed ancestry, and proportionally assigns an individual genotype to several populations of origin. These programs provide similar final results due to the fact that they are based on a common model.

FASTRUCT has been implemented using the C++ programming language. It contains a command-line engine and a graphical user interface (GUI) shell. The command-line engine is mainly designed for expert users who demand simplicity and flexibility and users who need to batch-analyse a large amount of data. It accepts data files in the STRUCTURE format, and produces output results in both textual and graphical formats. The textual format stores the estimated assignment probabilities and allele frequencies in a plain ASCII file. There are two types of graphical outputs. The first shows the log-likelihood history of a run, which can be used for convergence diagnosis; the second displays the estimated assignment probabilities in a bar chart as does the DISTRUCT program (Rosenberg 2004). The command-line engine can be used to generate artificial genotype data as

well; it features a simulation module of the Dirichlet allele frequencies. When invoked without any options, the command-line engine shows its typical usage with some explanatory notes. Mandatory options are choice of simulation or analysis, input data, number of individuals in the sample, ploidy, number of loci, number of clusters, and number of iterations of the EM algorithm. Other options can be found in the reference manual of the program.

The GUI shell can help newbies to familiarize themselves with the software, and it is generally a convenient way to use the program. It provides facilities for creating and managing projects. A project is a coherent unit which groups the input data, the algorithmic parameter settings, and the output results altogether. By interacting with the GUI shell, users can check their data, specify the parameter settings, run the EM algorithm, and visualize the results without mastering the usage of the command-line engine.

To validate the EM algorithm, we performed several analyses using simulated and real data. The most interesting results came from a simulation study using the data sets created by Latch *et al*. (2006) who kindly provided the data. These data were originally designed to compare the relative performance of Bayesian clustering programs. Each data set contained 500 individuals genotyped at 10 loci. Each data set was sampled from the five-island model and differentiated at one of 10 F_{ST} levels (F_{ST} = 0.01–0.10). Five replicates at each level of F_{ST} were generated. Latch *et al*. (2006) reported that STRUCTURE and BAPS performed very well at low levels of population differentiation, and were able to identify subpopulations at F_{ST} around 0.03. The individual assignment data for FASTRUCT are reported in Table 1 (F_{ST} = 0.02–0.05). At F_{ST} = 0.01, FASTRUCT failed to detect structure. For data sets with F_{ST} = 0.06–0.10, FASTRUCT produced perfect assignment to the five subpopulations. Comparing results in Table 1 with those reported in (Latch *et al*. 2006), one can conclude that FASTRUCT provided similar or better assignment than STRUCTURE and BAPS for these particular data sets. One run took about 16 s on a laptop PC with a 1.73 GHz CPU and 512 MB of RAM (1000 iterations). Using a faster computer, Latch *et al*. (2006) allocated about 3 h for each STRUCTURE run (30 s for each BAPS run).

Table 1 Individual assignment data for FASTRUCT, averages obtained from simulated data sets at F_{ST} = 0.02–0.05 (data from Latch *et al*. (2006))

F_{ST}	Average proportion of genome belonging to correct subpopulation	Average % misassigned
0.02	0.6484	32.96
0.03	0.9001	8.68
0.04	0.9662	3.00
0.05	0.9903	0.96

We therefore conducted some additional experiments to see how the increase of the number of loci and the number of individuals influence the processing time. In our experiments, we started with 10 loci, and gradually increased this number to 500 by a step of 10. Similarly, the number of individuals was increased from 100 to 5000 by a step of 100. We assumed three subpopulations, and we ran the program for 1000 iterations, which warranted the convergence of each run. When considering processing time as a function of the number of loci, we fixed the number of individuals to 100. Similarly, when considering processing time as a function of number of individuals, we fixed the number of loci to 10. The processing time measured in seconds increased linearly as approximately 0.15 × number of loci, and it remained less than 60 s for 400 loci. The processing time increased nonlinearly with the number of individuals, but it remained less than 90 s for 2500 individuals in the sample.

To give a short example of real data analysis, we applied the program to a subset of American populations extracted from the CEPH human diversity panel data (Rosenberg *et al*. 2002). This subset of data contained 108 individuals genotyped at 377 autosomal microsatellite loci. The samples were from five populations: Karitiana, Surui, Colombian, Maya, and Pima which were almost correctly retrieved by STRUCTURE in (Rosenberg *et al*. 2002). Running FASTRUCT for 1000 iterations took about 88 s, and led to assignment results that exhibited less than 3% differences with those obtained from STRUCTURE.

In summary, FASTRUCT produces results that compete with those obtained from STRUCTURE and BAPS but this can be done manyfold faster. However the claim here is not that FASTRUCT generally provides better results than other programs do. For instance, the run with maximal likelihood chosen from 10 STRUCTURE runs may outperform FASTRUCT in a standard analysis (likelihoods are directly comparable). However, there are many reasons we believe that FASTRUCT will be useful. First, at a preliminary stage before running STRUCTURE, it can provide a reference likelihood value that can be compared with those obtained from STRUCTURE in further analyses. Second, data sets are becoming increasingly large due to the explosion of genomic projects. Such data sets may be used to identify subsets of loci with specific signatures. For example Bayesian clustering methods can help investigate the subsets of loci that cluster the sample in different ways that does the full data set. This provides a mean to identify subsets of outlier loci (Beaumont & Nichols 1996) that are potentially subject to natural selection, particular migration patterns, or other departures from the HW and linkage equilibria. Due to exploding combinatorics, this could not be achieved by MCMC. On the one hand, the genomic revolution feeds Bayesian algorithms with more and more data. On the other hand, it puts an additional load on MCMC programs, and limits their applicability. The view presented here is that MCMC methods will probably

reach a limit that further material computer improvements would not overcome, and faster methods will then become useful in preliminary analyses.

The program, sample project files, and user's manual for Microsoft Windows OS are available free of charge at http://www-timc.imag.fr/Olivier.Francois/.

Acknowledgements

This work was supported by grants from INRIA and IMAG-ALPB. We are grateful to Emily K. Latch for providing us with the benchmark data. We thank Kevin Livingstone for helpful comments.

References

Beaumont MA, Nichols RA (1996) Evaluating loci for use in the genetic analysis of population structure. *Proceedings of the Royal Society of London. Series B, Biological Sciences*, **263**, 1619–1626.

Corander J, Walmann P, Sillanpaa MJ (2003) Bayesian analysis of genetic differentiation between populations. *Genetics*, **163**, 367–374.

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

Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, **39**, 1–38.

Latch EK, Dharmarajan G, Glaubitz JC, Rhodes OE Jr (2006) Relative performance of Bayesian clustering software for inferring population substructure and individual assignment at low levels of population differentiation. *Conservation Genetics*, **7**, 295–302.

McLachlan GJ, Peel D (2000) *Finite Mixture Models*. John Wiley & Sons. New York.

Pritchard JK, Stephens M, Donnelly P (2000) Inference of population structure using multilocus genotype data. *Genetics*, **155**, 945–959.

Rosenberg NA (2004) DISTRUCT: a program for the graphical display of population structure. *Molecular Ecology Notes*, **4**, 137–138.

Rosenberg NA, Pritchard JK, Weber JL *et al.* (2002) Genetic structure of human populations. *Science*, **298**, 2381–2385.

Saitou N, Nei M (1987) The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, **4**, 406–425.

Ward JH (1963) Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, **58**, 236–244.

Appendix

Here we give a brief summary of the model and describe the EM equations. Suppose that N diploid individuals are genotyped at L loci. The data (genotypes) are written as $x = (x_\ell^{(i,1)}, x_\ell^{(i,2)})$, for $i = 1, \ldots, N$, and $\ell = 1, \ldots, L$. The aim of clustering algorithms is to assign each individual i to one of K populations. We denote z_i a variable that indicates the origin of individual i. z plays the role of the missing data in the EM scenario. We denote the unknown allele frequencies as $p_{k\ell j}$, $k = 1, \ldots, K$, $j = 1, \ldots, J_{\ell'}$ where J_ℓ is the number of distinct alleles observed at locus ℓ. Given the origin of each individual, the genotypes are assumed to be generated by drawing alleles independently from the appropriate population frequencies,

$$P(x_i|z_i = k,p) = \prod_{\ell=1}^{L}\prod_{a=1}^{2} p_{k\ell x_\ell^{(i,a)},} \qquad p = (p_{k\ell j}). \tag{eqn 1}$$

We denote $\pi_k = P(z_i = k)$. The entire set of parameters can be described as $\psi = (\pi_{k'}, p_{k\ell j})$. To assign individual i to a cluster, we compute the posterior assignment probabilities $\tau_{ik} = P(z_i = k \mid x_i)$. Hereafter, we write $\psi^{(q)}$ for an estimate of ψ at iteration q of the EM algorithm. We will also consider $x_\ell^{(i,a)}$ as a binary vector of size J_ℓ with $x_\ell^{(i,a)}(j) = 1$ and all the remaining components being 0 if $x_\ell^{(i,a)} = j$. The $\tau_{ik}^{(q)}$ can be described as

$$\tau_{ik}^{(q)} = \frac{\prod_{\ell=1}^{L}\prod_{j=1}^{J_\ell} p_{k\ell j}^{(q)\sum_{a=1}^{2} x_\ell^{(i,a)}(j)} \times \pi_k^{(q)}}{\sum_{k=1}^{K}\prod_{\ell=1}^{L}\prod_{j=1}^{J_\ell} p_{k\ell j}^{(q)\sum_{a=1}^{2} x_\ell^{(i,a)}(j)} \times \pi_k^{(q)}}. \tag{eqn 2}$$

Then the update formulae for the parameters can be derived using the standard method of the EM algorithm

$$\pi_k^{(q+1)} = \frac{\sum_{i=1}^{N} \tau_{ik}^{(q)}}{N}, \tag{eqn 3}$$

and

$$p_{k\ell j}^{(q+1)} = \frac{\sum_{i=1}^{N} \tau_{ik}^{(q)} \sum_{a=1}^{2} x_\ell^{(i,a)}(j)}{2 \times \sum_{i=1}^{N} \tau_{ik}^{(q)}}. \tag{eqn 4}$$

When applying the EM algorithm to data, we need to provide values for $\tau_{ik}^{(0)}$. There are mainly two types of initialization methods: random initialization methods and clustering-based initialization methods (McLachlan & Peel 2000). The random initialization methods assign individuals into clusters randomly, while the clustering-based initialization methods assign individuals into clusters according to some distance criteria. Our initialization method is based on the hierarchical clustering method of Ward (1963). The Ward method is equivalent to the neighbour-joining phylogenetic reconstruction algorithm. To escape from this initial clustering, the membership of a small portion (25%) of individuals is randomly changed. This second step of initialization is necessary to prevent settling down on a bad local maximum. Then the EM algorithm gets an opportunity to explore the parameter space and it may converge to a better maximum. Generally, the clustering-based initialization method provides a better final result for the EM algorithm than random initialization does, and it also contributes to the convergence speed.