MOLECULAR ECOLOGY
RESOURCES

Molecular Ecology Resources (2015) doi: 10.1111/1755-0998.12387

CLUMPAK: a program for identifying clustering modes and packaging population structure inferences across *K*

NAAMA M. KOPELMAN,* JONATHAN MAYZEL,* MATTIAS JAKOBSSON,† NOAH A. ROSENBERG‡ and ITAY MAYROSE*

**Department of Molecular Biology and Ecology of Plants, Tel Aviv University, Ramat Aviv 69978, Israel, †Department of Evolutionary Biology and SciLife Lab, Uppsala University, Uppsala 75236, Sweden, ‡Department of Biology, Stanford University, Stanford, CA 94305, USA*

Abstract

The identification of the genetic structure of populations from multilocus genotype data has become a central component of modern population-genetic data analysis. Application of model-based clustering programs often entails a number of steps, in which the user considers different modelling assumptions, compares results across different predetermined values of the number of assumed clusters (a parameter typically denoted *K*), examines multiple independent runs for each fixed value of *K*, and distinguishes among runs belonging to substantially distinct clustering solutions. Here, we present CLUMPAK (Cluster Markov Packager Across *K*), a method that automates the postprocessing of results of model-based population structure analyses. For analysing multiple independent runs at a single *K* value, CLUMPAK identifies sets of highly similar runs, separating distinct groups of runs that represent distinct modes in the space of possible solutions. This procedure, which generates a consensus solution for each distinct mode, is performed by the use of a Markov clustering algorithm that relies on a similarity matrix between replicate runs, as computed by the software CLUMPP. Next, CLUMPAK identifies an optimal alignment of inferred clusters across different values of *K*, extending a similar approach implemented for a fixed *K* in CLUMPP and simplifying the comparison of clustering results across different *K* values. CLUMPAK incorporates additional features, such as implementations of methods for choosing *K* and comparing solutions obtained by different programs, models, or data subsets. CLUMPAK, available at http://clumpak.tau.ac.il, simplifies the use of model-based analyses of population structure in population genetics and molecular ecology.

Keywords: admixture, ancestry, clustering, population structure

Received 30 August 2014; revision received 19 January 2015; accepted 28 January 2015

Introduction

Model-based identification of population structure from individual multilocus genotypes is of great importance for the study of genetic relationships among individuals and populations. In diverse organisms, population structure inference provides a basis for interpreting patterns of extant genetic variation, serving as a foundation for the study of evolutionary history (Rosenberg *et al.* 2002; Parker *et al.* 2004; Whitfield *et al.* 2006; Driscoll *et al.* 2007; Zhao *et al.* 2013). In association mapping of complex traits, inference of population structure can assist in avoiding the false-positive genotype–phenotype associations that structure can generate (Pritchard & Donnelly 2001; Thornsberry *et al.* 2001; Hoggart *et al.* 2003; Breseghello & Sorrells 2006). In conservation genetics and molecular ecology, population structure inferences are regularly used to investigate such topics as population boundaries, population connectivity and gene flow, aiding the understanding and management of natural populations (Manel *et al.* 2005; Gompert & Buerkle 2013).

A particularly important family of methods for investigating population structure examines multilocus genotypes of individuals using model-based cluster analysis (Pritchard *et al.* 2000; Dawson & Belkhir 2001; Corander *et al.* 2003, 2004, 2008a,b; Falush *et al.* 2003, 2007; Guillot *et al.* 2005; Tang *et al.* 2005; François *et al.* 2006; Pella & Masuda 2006; Chen *et al.* 2007; Gao *et al.* 2007; Huelsenbeck & Andolfatto 2007; Guillot 2008; Alexander *et al.* 2009; Durand *et al.* 2009; Hubisz *et al.* 2009; Shringarpure & Xing 2009; Alexander & Lange 2011; Huelsenbeck *et al.* 2011; Frichot *et al.* 2014). These methods, the most widely

Correspondence: N. M. Kopelman, Fax: +972-3-6409380; E-mail: naamakop@post.tau.ac.il and Itay Mayrose, Fax: +972-3-6409380; E-mail: itaymay@post.tau.ac.il

used of which is STRUCTURE (Pritchard *et al.* 2000; Falush *et al.* 2003, 2007; Hubisz *et al.* 2009), classify individuals into a prespecified number of populations, disregarding predefined labels for the sampled individuals and therefore performing 'unsupervised' clustering. In typical uses, STRUCTURE, like many other clustering programs, views each individual as a distinctive mixture of a set of inferred statistical populations or clusters, each characterized by distinct allele frequencies. Coefficients of membership of each individual in the various clusters, summing to 1 across clusters, are estimated in an iterative procedure. The various 'STRUCTURE-like' (Weiss & Long 2009) programs, which consider both mixed-membership models that view individuals as mixtures of the clusters, and assignment models in which membership coefficients represent probabilities of membership in the clusters, approach the same conceptual problem with a variety of choices of modelling assumptions and various alternative computational strategies. They enable rich and complex data analyses, allowing the user to modify a wide variety of parameters and to examine different models – for instance, supervised models, in which the predefined population labels are used to assist the inference process. For most programs, users have considerable choice in model assumptions, and they are advised to test different assumptions as well as a range of predefined values of the number of clusters, *K*.

STRUCTURE-like algorithms typically involve stochastic simulations, and therefore can produce different outcomes in replicate analyses. For this reason, it is important to perform several independent assessments of the same data set using the same modelling assumptions and free parameters (Rosenberg *et al.* 2001a; Gilbert *et al.* 2012). Differences among solutions tend to arise from two phenomena (Jakobsson & Rosenberg 2007). The first is 'label switching', caused by the arbitrary way in which inferred clusters are labelled, resulting in unmatched labels between replicate runs even when the same membership coefficients are produced. The second type of difference is 'genuine multimodality', in which data analyses result in two or more truly distinct solutions or modes. To distinguish label switching from genuine multimodality, Jakobsson & Rosenberg (2007) developed CLUMPP (Cluster Matching and Permutation Program), which computes a pairwise similarity score between pairs of runs with the same *K*, identifying an optimal alignment of the replicate runs that eliminates label switching so that genuine multimodality can be detected. A matrix of similarity scores produced by CLUMPP for pairs of aligned runs can be used to identify groups of runs that produce nearly identical solutions and can be said to fall into the same mode. For example, Wang *et al.* (2007) and Jakobsson *et al.* (2008) identified as modes all sets of replicates for which the pairwise similarity score for each pair of runs exceeded a specific threshold. This approach amounts to identifying fully connected subgraphs of a graph, or cliques, where vertices represent runs and edges indicate occurrences of pairwise similarity scores above the threshold. Once distinct modes are identified, users can choose a single run from each inferred mode, or they can alternatively average runs in each mode by a second application of CLUMPP only on those runs within the mode. A drawback of this approach is that determination of modes is often highly dependent on the exact threshold chosen. Moreover, cliques found by this approach can have a high percentage of overlap, and therefore might not necessarily represent truly distinct solutions.

In addition to eliminating label switching and identifying distinct modes for a single *K*, users often compare clustering modes for a range of *K* values. As CLUMPP aligns only those runs with a fixed value of *K*, however, automated alignment of runs across multiple *K* values has not been possible. This challenge is particularly noticeable in cases in which simultaneous examination of multiple modes at multiple *K* values is of interest, such as when the clustering pattern in the most frequently occurring mode for a given *K* does not provide a refinement of the corresponding solution for a smaller choice of *K* (Wang *et al.* 2007). Additional tasks, including comparisons of results obtained for multiple model choices, subsets of data, or different programs altogether, have also required that computations be performed external to CLUMPP and other cluster postprocessing software such as DISTRUCT (Rosenberg 2004) and STRUCTURE HARVESTER (Earl & Vonholdt 2012). From the end-user perspective, performing a thorough STRUCTURE-like analysis can be a tedious task.

To overcome these difficulties in identification of modes, alignment of runs across *K* values, and automation of additional postprocessing steps, we have developed CLUMPAK, the Cluster Markov Packager Across *K*. CLUMPAK clusters replicate runs for the same *K* via a Markov clustering algorithm that improves upon the commonly used fixed threshold approach. CLUMPAK summarizes the modes identified, eliminating the additional step of 're-CLUMPPing' runs within modes to obtain consensus results for each mode. A newly developed extension further enables cluster matching across different values of *K* and simplifies the examination and presentation of results across a range of *K* values. CLUMPAK additionally allows users to easily compare results obtained under different models, programs or subsets of the data. Here, we describe the methods underlying CLUMPAK, and we use simulations to demonstrate the ability of CLUMPAK to accurately cluster individual runs. We further illustrate the use of CLUMPAK on an example human population-genetic data set.

Materials and methods

Generating clusters of runs at a single value of K

CLUMPAK aids users in summarizing the entire set of runs produced for their data set using one or more clustering programs. Users normally run their analysis using a range of K values, performing multiple replicates for each K. CLUMPAK obtains similarity matrices for pairs of replicate runs using the SSC′ similarity score of Jakobsson & Rosenberg (2007) as implemented in CLUMPP, employing as a default the *LargeK-Greedy* algorithm of CLUMPP with 2000 random input sequences (both of these choices can be changed by the user). From among the replicates, CLUMPAK then identifies different modes within a single value of K – genuinely different solutions — if they are present.

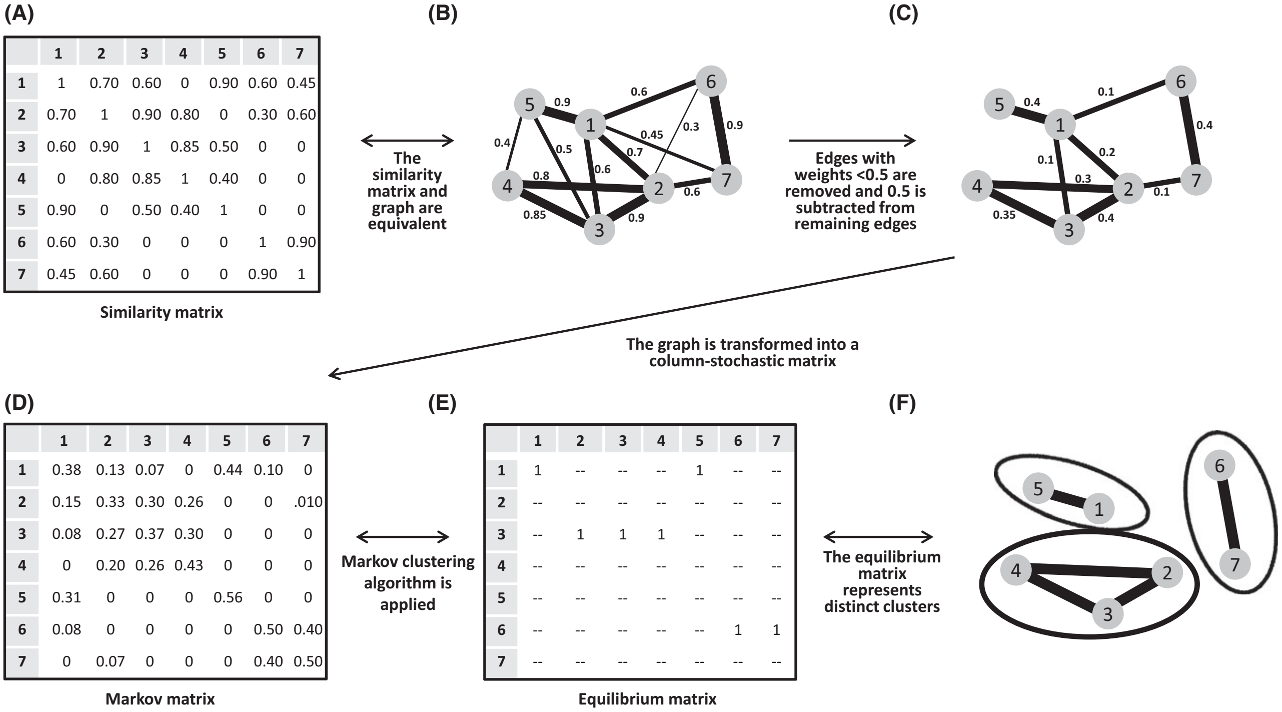
First, CLUMPAK treats the similarity matrix as a fully connected, undirected, weighted graph, $G = (V,E)$, where vertices in set V represent independent runs, and the edge weight connecting vertices u and v is the similarity score SSC′(u, v). Second, given the weighted graph, CLUMPAK uses the Markov clustering (MCL) algorithm (Van Dongen 2000, 2008) to partition the graph into distinct modes. This algorithm is a general method for identifying clusters, and it has previously been adapted for diverse biological problems, including orthology assignments (Enright *et al.* 2002), detection of operational taxonomic units (Ratnasingham & Hebert 2013), and identification of co-occurring associations among microbes (Faust *et al.* 2012). The similarity matrix obtained for a single K, as represented by the graph G, is transformed into a column-stochastic matrix or Markov matrix, where entry (i, j) represents a probability of ‘transition’ from vertex j to vertex i. The MCL algorithm involves alternating between matrix ‘expansion’ and matrix ‘inflation’, where expansion refers to taking the power of a stochastic matrix (i.e. normal matrix squaring), and inflation refers to taking the Hadamard power (Horn & Johnson 1991) of the matrix (taking powers entrywise) with coefficient r and rescaling the matrix so that columns sum to 1. These alternating steps of matrix expansion and matrix inflation are aimed at simulating random walks on a graph: expansion steps correspond to computing probabilities associated with paths of higher length, and inflation boosts the probabilities of intracluster walks, accentuating the similarity of related runs and the dissimilarity of divergent runs. The single parameter r, which determines the ‘granularity’ of the clusters obtained, controls the outcome of the algorithm, producing finer granularity, and more clusters, with larger r. Figure 1 illustrates the steps involved in the clustering process.

To obtain a suitable level of granularity with the MCL algorithm, a useful preprocessing step is to use a threshold for the inclusion of edges in the graph (S. van Dongen, personal communication). This threshold can be set by the user or determined dynamically given the properties of the graph analysed (the default setting in CLUMPAK). Specifically, for a given data set and for each value in the range of thresholds explored, the mean node degree, as well as the fraction of singleton nodes, representing runs that are not connected to any other runs, is examined (accomplished using the *vary-threshold* option of MCL). The dynamic threshold that is used by CLUMPAK is the largest threshold for which the fraction of singletons is smaller than 0.1, and the mean node degree across all nodes in the graph is at least 50% of the total number of vertices. Once a threshold has been set, edges whose weights are smaller than the threshold are removed, and the weights of the remaining edges are shifted downward by the value of the threshold (accomplished using the *tf* option of MCL). Other than using a threshold, the default parameters of MCL are used by CLUMPAK (equivalent to setting the parameter r to 2 without using preinflation). Notably, MCL can cluster two vertices together even if the input graph does not contain an edge connecting them; thus, the choice of the threshold is less influential than in the clique approach.

Once clusters are identified within replicate runs with the same K value, CLUMPAK utilizes CLUMPP (Jakobsson & Rosenberg 2007) for obtaining an average run representing each cluster and DISTRUCT (Rosenberg 2004) for graphical display of the results. CLUMPAK reports to the user the number of replicate runs clustered in each mode detected, the mean posterior probability of runs clustered in each mode, and the mean CLUMPP scores between all pairs of runs within a mode. Notably, the largest mode (the one containing the largest number of replicate runs) might not have the highest mean posterior probability; users are advised to consider both attributes when interpreting their results.

Aligning clustering results across different K values

CLUMPAK aligns cluster labels across replicate runs, clusters of runs, and different K values. CLUMPAK relies on a previously developed algorithm for alignments at fixed K values (Jakobsson & Rosenberg 2007), whereas the label-matching algorithm for different K values is a novel extension and proceeds as follows. We refer to the matrix of membership coefficients $[Q]_{C\times K}$ as the Q-matrix, with C rows corresponding to individuals and K columns corresponding to clusters, and the value in the cth row and kth column representing the membership coefficient for individual c in cluster k as inferred in a single run (or the mean of a set of runs). Each matrix consists of non-negative entries, and the sum of the entries in any


(A)
(B)
(C)
5 0.9 1 0.6 6 0.4 0.5 0.45 0.3 0.9 0.7 0.6 4 0.8 2 0.6 7 0.85 3 0.9
The similarity matrix and graph are equivalent
Edges with weights <0.5 are removed and 0.5 is subtracted from remaining edges
5 0.4 1 0.1 6 0.4 0.1 0.2 4 0.3 2 0.1 7 0.35 3 0.4
The graph is transformed into a column-stochastic matrix
(D)
(E)
(F)
Markov clustering algorithm is applied
The equilibrium matrix represents distinct clusters
6 5 1 7 4 2 3

	1	2	3	4	5	6	7
1	1	0.70	0.60	0	0.90	0.60	0.45
2	0.70	1	0.90	0.80	0	0.30	0.60
3	0.60	0.90	1	0.85	0.50	0	0
4	0	0.80	0.85	1	0.40	0	0
5	0.90	0	0.50	0.40	1	0	0
6	0.60	0.30	0	0	0	1	0.90
7	0.45	0.60	0	0	0	0.90	1

Similarity matrix

	1	2	3	4	5	6	7
1	0.38	0.13	0.07	0	0.44	0.10	0
2	0.15	0.33	0.30	0.26	0	0	.010
3	0.08	0.27	0.37	0.30	0	0	0
4	0	0.20	0.26	0.43	0	0	0
5	0.31	0	0	0	0.56	0	0
6	0.08	0	0	0	0	0.50	0.40
7	0	0.07	0	0	0	0.40	0.50

Markov matrix

	1	2	3	4	5	6	7
1	1	--	--	--	1	--	--
2	--	--	--	--	--	--	--
3	--	1	1	1	--	--	--
4	--	--	--	--	--	--	--
5	--	--	--	--	--	--	--
6	--	--	--	--	--	1	1
7	--	--	--	--	--	--	--

Equilibrium matrix

Fig. 1 Schematic of the Markov clustering algorithm for a similarity matrix of 7 independent STRUCTURE runs. (A) Similarity matrix, with the entry in a cell representing the similarity of the runs represented in the associated row and column. (B) A graph equivalent to the similarity matrix, with nodes representing runs and edges representing similarity scores for distinct pairs of runs (loops connecting nodes to themselves are not shown). (C) A graph following the removal of edges whose weights are smaller than a fixed threshold, 0.5. The weights of the remaining edges have been shifted by −0.5. (D) A column-stochastic matrix for the 7 runs. (E) The equilibrium matrix obtained by iterating expansion and inflation steps in the Markov clustering algorithm. (F) Clusters of runs identified by the Markov clustering algorithm.

row is 1. Following Jakobsson & Rosenberg (2007), the similarity score between replicate runs i and j (i.e. two Q-matrices of the same dimension) is defined as

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

where $\|\cdot\|_F$ denotes the Frobenius matrix norm. The normalization constant $\sqrt{2C}$ constrains G' to lie between 0 and 1 (see eq. 6 in Jakobsson & Rosenberg 2007).

Consider now a pair of Q-matrices produced for two consecutive K values, $[Q_\mathrm{i}]_{C\times K}$ and $[Q_\mathrm{j}]_{C\times(K+1)}$. To account for the different sizes of Q_i and Q_j, a column of zeros is added to Q_i, as a final $(K + 1)$ column, representing a cluster that does not exist for K clusters, but does exist for $K + 1$. This addition produces a $C \times (K + 1)$ matrix $\hat{Q}$. We then compute the similarity score

$$G''(Q_i, Q_j) = \frac{\|\hat{Q}_i - Q_j\|_F}{\sqrt{2C}}. \tag{2}$$

Using G'' to measure similarity, the optimal alignment of matrices Q_i and Q_j is defined as the permutation of columns of Q_j that maximizes G'' over all $(K + 1)!$ possible permutations. We sequentially align clustering solutions starting from an initial value K_0, aligning a solution with $K_0 + 1$ clusters to the K_0 solution, the $K_0 + 2$ solution to the $K_0 + 1$ solution, and so on. This process examines alignments of the various modes at $K_0 + 1$ clusters to the corresponding modes at K_0 clusters.

Comparing results across different models

CLUMPAK enables a comparison of clustering solutions across different programs (e.g. comparing output from STRUCTURE and ADMIXTURE), different model choice (e.g. comparing output for a no-admixture model and an admixture model), or different subsets of genetic markers across the same data set (e.g. comparing output for half of the markers to the other half), each with its own sets of runs. Hereafter, all three of these scenarios are termed 'different models'. CLUMPAK first analyses and summarizes the results of each model separately, but it aligns cluster labels across the two models, thus enabling easy visual inspection of the results. Additionally, CLUMPAK provides users with CLUMPP similarity scores between the modes identified for the two models, and for comparison, CLUMPP similarity scores within each mode.

Simulation study

We used simulations to assess the accuracy of CLUMPAK in differentiating runs into distinct clustering solutions. We simulated STRUCTURE-like outputs directly, for three *a priori* distinct populations, each with a sample of 25 individuals. *Q*-matrices were generated for three predefined clusters ($K = 3$). A single simulation contained 40 *Q*-matrices, corresponding to 40 individual STRUCTURE-like runs.

We explored two aspects of variation across runs. First, we allowed mean membership coefficients for each *a priori* population to vary across replicate runs, representing stochastic variation across individual STRUCTURE-like runs. Second, within each run and each population, we varied individual membership coefficients around the population mean for that run. The standard deviation (SD) parameters for these two sources of variation are termed $\mathrm{SD_{RUNS}}$ and $\mathrm{SD_{INDIVS}}$, respectively (described in more detail below). By increasing these standard deviations, we obtained *Q*-matrices that are increasingly more challenging for accurate clustering (see Supplementary Note 1 for the relation of the similarity scores between runs to $\mathrm{SD_{RUNS}}$ and $\mathrm{SD_{INDIVS}}$).

The simulations assumed that each *a priori* population is characterized by its own 'seed' membership coefficients, $q_k^{(i)}$, denoting the mean genomic proportion within *a priori* population i originating from cluster k ($k = 1, 2, 3$). Thus, for example $(q_1^{(1)}, q_2^{(1)}, q_3^{(1)}) = (0.5, 0.4, 0.1)$, corresponds to a scenario in which the mean membership coefficients for individuals belonging to population 1 are 0.5, 0.4 and 0.1 for clusters 1, 2 and 3, respectively. We chose the seed vectors on the basis of empirical examples, as described in Appendix S1 (Supporting Information).

We simulated data in a two-step manner as follows. First, for each run ($m = 1, 2, \ldots, 40$), a set of mean membership coefficients for each *a priori* population was sampled around the predefined population seed vector. For *a priori* population i, we sampled around the seed vector using a Dirichlet distribution $\mathrm{Dir}(\alpha^{(i)} \cdot q_1^{(i)}, \alpha^{(i)} \cdot q_2^{(i)}, \alpha^{(i)} \cdot q_3^{(i)})$, where $\alpha^{(\mathrm{i})}$ is a concentration parameter adjusted so that the standard deviation of the first component of the Dirichlet distribution was equal to a predefined value of $\mathrm{SD_{RUNS}}$ (see Appendix S2, Supporting Information). Let $(\hat{q}_1^{(i)}, \hat{q}_2^{(i)}, \hat{q}_3^{(i)})$ denote the mean membership coefficients for population i that were sampled for a certain run (for clarity, we omit the notation of the run index). This vector is the output of the first simulation step.

In the second step, we sampled individual membership coefficients around the run-specific $(\hat{q}_1^{(i)}, \hat{q}_2^{(i)}, \hat{q}_3^{(i)})$ vector of population-mean membership coefficients for each of the 25 individuals belonging to *a priori* population i. To this end, we sampled around $\hat{q}_k^{(i)}$ using $\mathrm{Dir}(\beta^{(i)} \cdot \hat{q}_1^{(i)}, \beta^{(i)} \cdot \hat{q}_2^{(i)}, \beta^{(i)} \cdot \hat{q}_3^{(i)})$, where $\beta^{(\mathrm{i})}$ is a concentration parameter adjusted so that the standard deviation of the Dirichlet distribution was equal to a predefined value of $\mathrm{SD_{INDIVS}}$ (Appendix S2, Supporting Information). In summary, obtaining a single *Q*-matrix requires sampling once around $(q_1^{(i)}, q_2^{(i)}, q_3^{(i)})$ for each *a priori* population $i = 1, 2, 3$, and then for each i, sampling 25 sets of membership coefficients around $(\hat{q}_1^{(i)}, \hat{q}_2^{(i)}, \hat{q}_3^{(i)})$. A single simulation iterates this process 40 times, obtaining a simulated set of 40 STRUCTURE-like runs on 75 individuals. These runs were then provided as input to CLUMPAK.

The procedure outlined above represents a scenario in which all 40 STRUCTURE runs are drawn from a single set of seed vectors, and we thus expect these runs to be clustered together, obtaining a single mode. We are also interested, however, in examining the performance of CLUMPAK under genuine bimodality, when STRUCTURE-like runs can be divided into two truly distinct solutions. The procedures for generating simulated data under bimodality were similar to the unimodal case, with two exceptions. First, instead of having a single seed membership vector per population we now have two: one representing a major mode and the other a minor mode. Second, among the 40 simulated runs, we choose a fraction of runs, f, to represent the minor mode, placing fraction $1\text{-}f$ into the major mode. We used several values of f, simulating data for $f = 0$, 0.125, 0.25, 0.375. The fraction 0.125, for example, corresponds to the case in which the minor mode has 5 runs and the major mode has 35 runs. The $f = 0$ case represents a unimodal simulation.

Simulations were performed for a range of values of $\mathrm{SD_{RUNS}}$ and $\mathrm{SD_{INDIVS}}$, 0.01 to 0.15, for both parameters. Values exceeding 0.15 represent very high variation, as evident from the similarity scores between simulated runs, compared to the similarities observed in the empirical data sets that we considered (Note S1, Supporting Information). For each combination of $\mathrm{SD_{RUNS}}$, $\mathrm{SD_{INDIVS}}$, and f, we performed 30 simulations. The seed membership coefficients were taken from empirical examples for which bimodality was evident (Appendix 1, Supporting Information). The seed vectors for the simulations in Figs 2 and 3 are based on mean membership coefficients in major and minor modes for the Mozabite, Bedouin, and Druze populations at $K = 3$, for the same set of runs presented in the Results section (Figs 4 and 5). Simulation sets based on other empirical examples (Supporting Information) yielded similar results.

We used the simulations both to assess accuracy of mode identification and to validate choices of settings in CLUMPAK. Each simulation was given to CLUMPAK as input under different program settings. First, we modified the

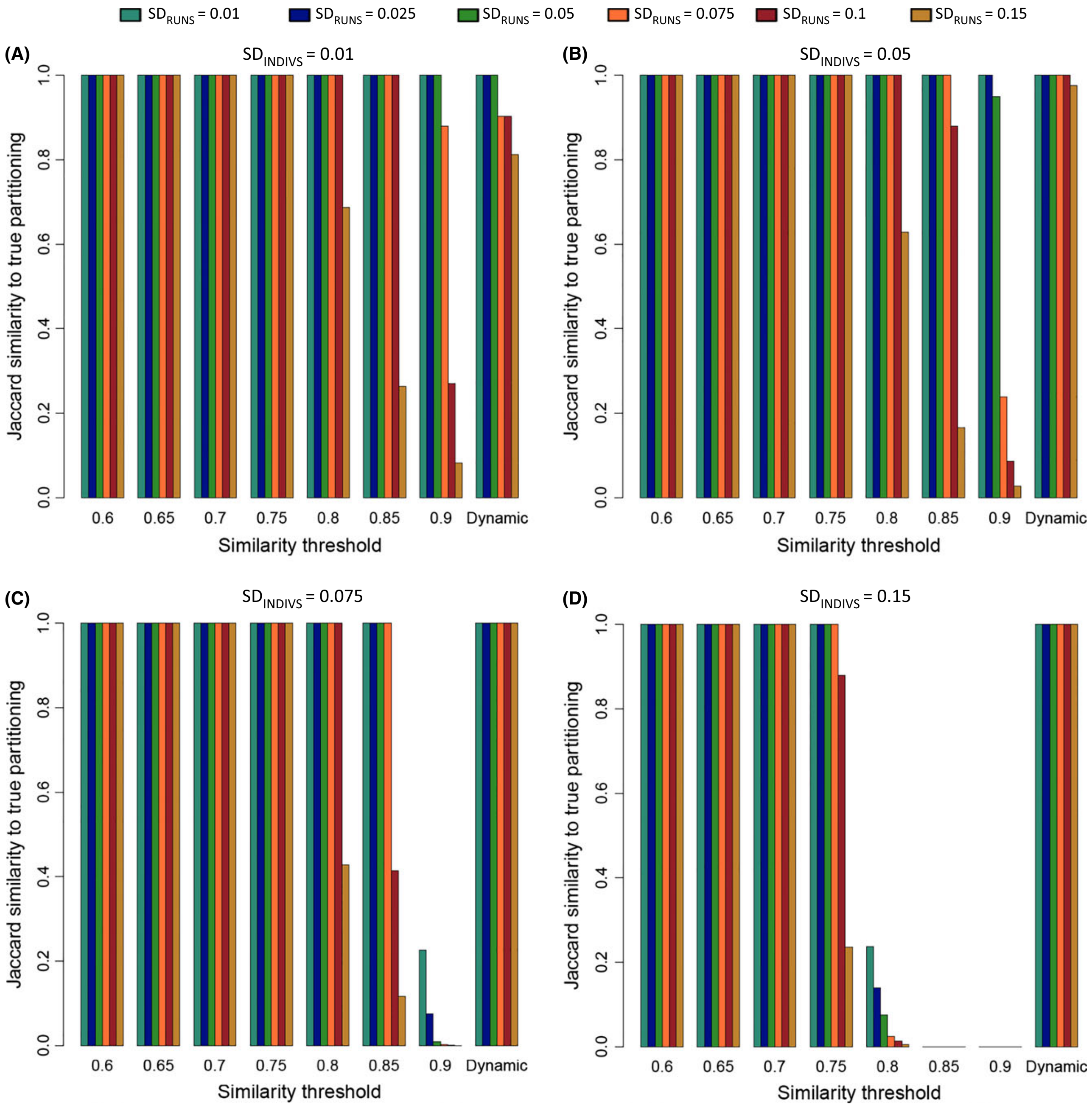


Fig. 2 Jaccard similarity scores between the clustering solutions obtained by CLUMPAK and the 'true' (simulated) partitioning in a unimodal case, as a function of $\mathrm{SD_{RUNS}}$, $\mathrm{SD_{INDIVS}}$ and either a fixed threshold value or a dynamic threshold. Simulations were carried out with one underlying mode. Different colours represent different values of $\mathrm{SD_{RUNS}}$ as given by the colour palette at the top of the figure. (A) $\mathrm{SD_{INDIVS}} = 0.01$. (B) $\mathrm{SD_{INDIVS}} = 0.05$. (C) $\mathrm{SD_{INDIVS}} = 0.075$. (D) $\mathrm{SD_{INDIVS}} = 0.15$.

similarity threshold, using either a fixed similarity threshold in the range of 0.6–0.9, reasonable in light of past choices (Wang *et al.* 2007; Jakobsson *et al.* 2008), or by the default approach in which the threshold was determined dynamically. We also examined values of the inflation parameter that differed from the CLUMPAK default of $r = 2$.

We used the Jaccard index (Jain & Dubes 1988; Kaufman & Rousseeuw 1990) to measure the accuracy of the clustering solution obtained by CLUMPAK in comparison to the 'real' (simulated) modes. This index varies from 0 to 1, with 1 being a perfect match between the inferred and simulated clustering of runs. Specifically, each simulation contains 40 runs;

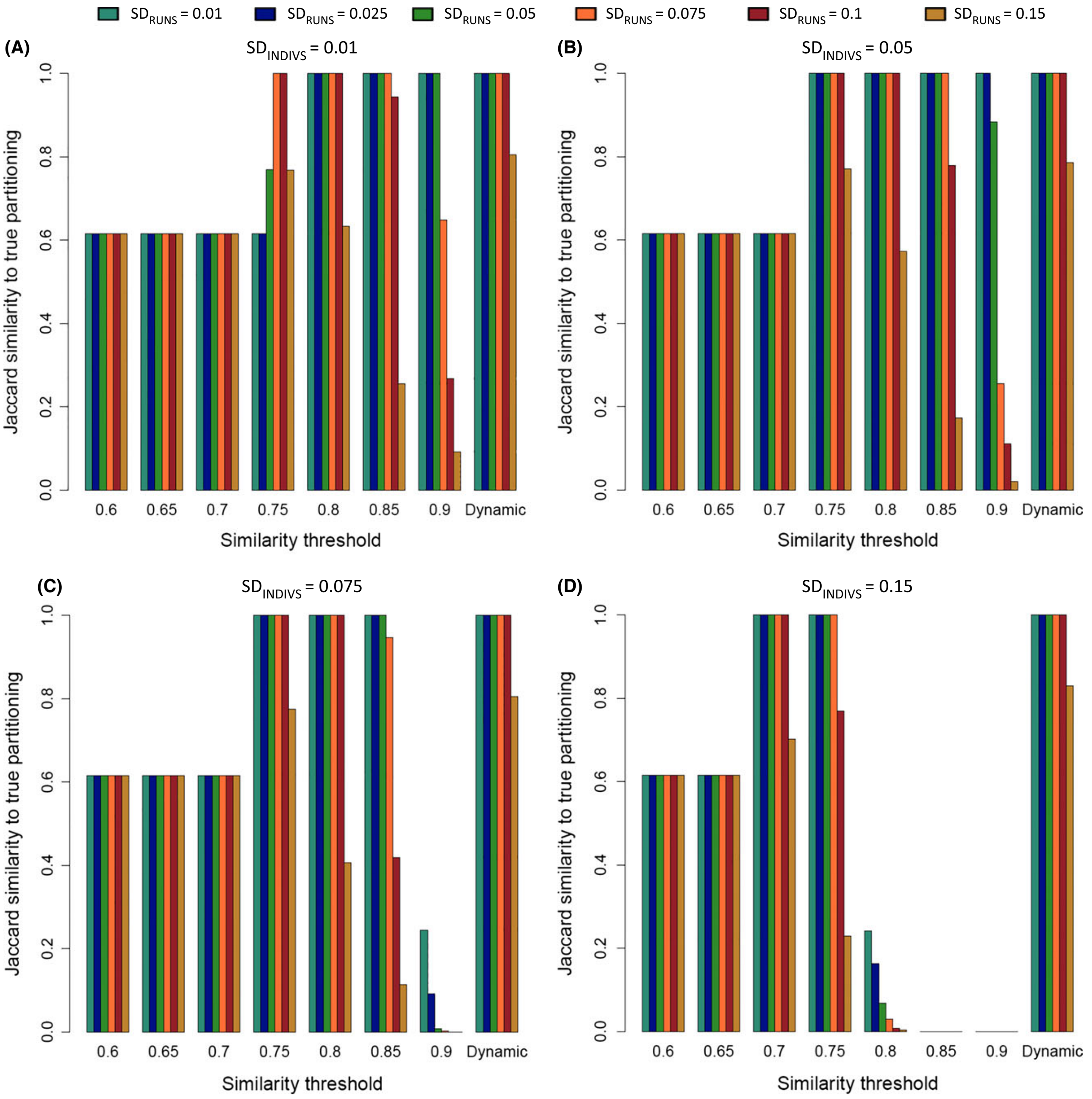


Fig. 3 Jaccard similarity scores between the clustering solution obtained by CLUMPAK and the 'true' (simulated) partitioning in a bimodal case, as a function of SD_{RUNS}, SD_{INDIVS} and either a fixed threshold value or a dynamic threshold. Simulations were carried out with a fraction $f = 0.25$ of the runs assigned to the minor mode. Different colours represent different values of SD_{RUNS} as given by the colour palette at the top of the figure. (A) $SD_{INDIVS} = 0.01$. (B) $SD_{INDIVS} = 0.05$. (C) $SD_{INDIVS} = 0.075$. (D) $SD_{INDIVS} = 0.15$.

thus, each has $(40 \times 39)/2 = 780$ pairs of runs. Each pair was tabulated in one of four categories: (i) N_{11}, the runs truly belong in the same mode and were placed in the same mode by CLUMPAK; (ii) N_{10}, the runs belong in the same mode, but were placed in different modes by CLUMPAK; (iii) N_{01}, the runs belong in different modes, but were placed in the same mode; and (iv) N_{00}, the runs belong in different modes and were placed in different modes. The Jaccard index is calculated according to the following formula:

$$J = \frac{N_{11}}{N_{11} + N_{10} + N_{01}} \qquad (3)$$

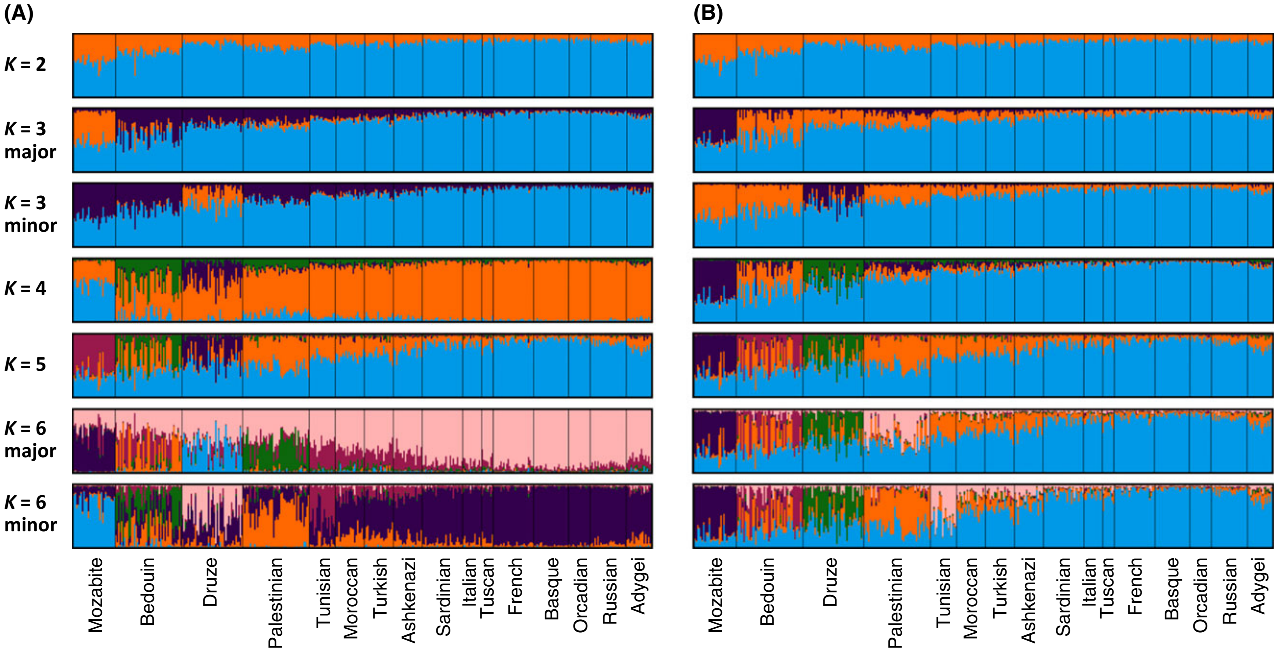


Fig. 4 Major and minor modes identified for 399 individuals from 16 populations, illustrating Clumpak's label matching across *K* values. For each *K* value, Structure was run 40 times. (A) Membership coefficients produced by Clumpak when label matching across *K* values was disabled (for illustration only). (B) Membership coefficients produced by Clumpak for the same set of runs as in (A), matching labels across *K* values. Clusters have been permuted to sequentially match the configuration for the lowest *K* in the range (*K* = 2).

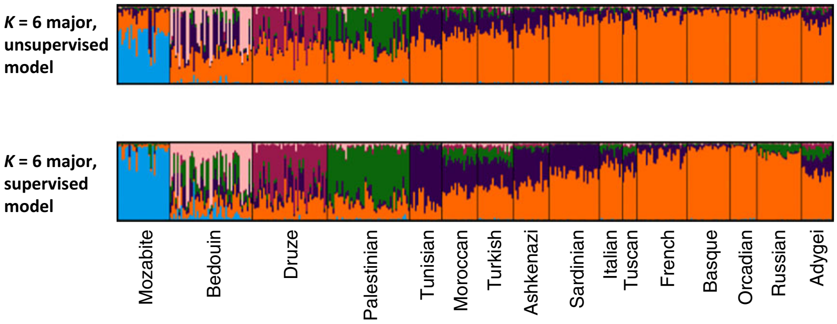


Fig. 5 Membership coefficients compared using two different Structure models with *K* = 6. The same set of 399 individuals from 16 populations was used for both panels. For each model, Structure was run 40 times, and Clumpak was used to identify distinct solutions among runs of the same model as well as to compare the different modes across the two models. For the runs without the locprior model (unsupervised), the runs are the same as in Fig. 4. Top: the major mode for the unsupervised admixture model without the locprior model. Bottom: the major mode for the locprior (supervised) admixture model.

Availability and requirements

Clumpak is available through a Web server, at http://clumpak.tau.ac.il, or can be downloaded and used on Linux, Unix, and Mac operating systems. With either approach, Clumpak includes calls to Clumpp, Distruct and MCL (all of which the user needs to install if running Clumpak locally). In the downloaded version, users can explore a range of values of optional parameters – especially those concerned with running Clumpp – that are more restricted in the Web server to limit exhaustive running times.

Using the Distruct program (Rosenberg 2004), and by allowing users to specify graphical parameters used by Distruct, Clumpak provides users with high-quality images produced for each *K* (and within each *K* for each mode of convergence); the modes are graphically aligned across *K*. For its main pipeline, Clumpak requires that users submit the results of clustering programs in Structure or Admixture format, allowing any range of *K* values and single or multiple runs for each K. CLUMPAK also implements the methods of Evanno *et al.* (2005) and Pritchard *et al.* (2000) for selecting a preferred value of K.

Results

Simulations

We used simulations to examine the accuracy of CLUMPAK in assigning independent STRUCTURE-like runs into separate modes. We first compared the default setting of CLUMPAK, under which a threshold for the inclusion of edges in the similarity graph (Fig. 1) is chosen dynamically, to a range of fixed values for the threshold.

Figure 2 presents the Jaccard similarity score between the partition obtained by CLUMPAK and the true partition for unimodal simulations, in which all replicate runs were sampled from the same mode. Under this setting, as the value of the fixed threshold increases, the accuracy of the clustering obtained using the MCL algorithm decreases. This pattern is expected, because higher thresholds lead to sparser graphs, resulting in the separation of replicate runs into distinct clusters, whereas in the correct configuration, all runs are clustered together to form a single mode. Comparing the four panels in Fig. 2, representing different values of SD_{INDIVS}, and comparing different values of SD_{RUNS} for fixed values of SD_{INDIVS}, the decline in accuracy with an increasing threshold is more noticeable for simulations at higher SD values. This pattern is also expected, as higher SD values indicate more variation between the runs, and hence sparser graphs at a fixed threshold.

Figure 3 presents the Jaccard score for the bimodal case, when two distinct modes were simulated, with 25% of the runs belonging to the minor mode (other fractions are examined in Supporting Information, Figs S1 and S2). Here, the accuracy of the fixed threshold exhibits a peaked pattern, with the most accurate clustering achieved at intermediate values of the threshold. Whereas low thresholds lead to erroneous clustering of all replicate runs into a single mode, thresholds that are too high break the similarity graph into too many components, identifying too many distinct modes. As the simulated SD values increase, the peak moves to the left, and thus, the optimal fixed threshold is lower.

The differing performance of the graph clustering procedure in the unimodal and bimodal cases in Figs 2 and 3 suggests that a single fixed threshold does not yield an optimal solution applicable in all scenarios, as the optimal threshold is influenced both by the number of modes among the replicate runs and by the level of variation across runs and individuals. These trade-offs are accommodated using the dynamic procedure implemented in CLUMPAK, which sets the threshold according to the characteristics of the input data. Indeed, as the level of variation increases in either of two ways (SD_{RUNS} and SD_{INDIVS}), the dynamic threshold is set to lower values, thus allowing more variability within modes inferred to be distinct (Note S1, Supporting Information). In both the unimodal and bimodal cases, the performance of the dynamic threshold is optimal or near the optimum for all simulation scenarios (rightmost bars in each panel of Figs 2 and 3). For example using the dynamic threshold, in the unimodal simulations, the mean Jaccard similarity score calculated across 30 simulations for each set of parameters is 1.0 or close to 1.0 in most settings, including the challenging setting of low SD_{INDIVS} values coupled with high values of SD_{RUNS}. The somewhat lower accuracy in such cases arises from the separation of the runs into multiple modes; the clustering is still adequate, as in nearly all simulations, there is a major cluster containing most replicate runs (>35 of 40). In the bimodal simulations, the accuracy of the inferred clusters obtained using the dynamic threshold is high for both low and high SD_{INDIVS} and SD_{RUNS} values and exceeds that of any single fixed value (Fig. 3).

In addition to the manner of choosing the threshold for inclusion in the similarity graph, the inflation parameter, r, of the MCL algorithm, potentially also influences the clustering of runs. We found that for the range of values examined (ranging from 1.1 to 12, representing the range of reasonable values for that parameter; S. van Dongen, personal communication), however, this parameter has little influence on the clustering results for low values of SD_{INDIVS} and SD_{RUNS}, whereas for higher values the optimal performance is obtained for $r = 2$, the CLUMPAK default (Fig. S3, Supporting Information). Taken together, for the clustering task of distinguishing between modes, we found that the threshold choice is more influential than the inflation parameter r, and that the dynamic threshold implemented in CLUMPAK produces high accuracy that is optimal or close to optimal.

Empirical example

To illustrate a data application of CLUMPAK, we have re-examined a data set of 678 autosomal microsatellites in 399 individuals from 16 European and Middle Eastern human populations, including 78 individuals of Jewish descent (Ashkenazi, Moroccan, Tunisian, and Turkish Jews). Using STRUCTURE, Kopelman *et al.* (2009) previously studied the membership coefficients of the 399 individuals for a range of K values. For each value of K, 40 individual STRUCTURE runs were conducted and assessed using CLUMPP (Jakobsson & Rosenberg 2007). The clique clustering approach was then used to detect distinct solutions.

To re-analyse this data set through CLUMPAK, we first generated new STRUCTURE replicates, running STRUCTURE 40 times for each value of K from 2 to 6, using the mixed-membership admixture model with a burn-in period of 10 000 iterations followed by 20 000 additional iterations for each run. The resulting runs were provided to CLUMPAK.

Figure 4 presents the output obtained from CLUMPAK for this collection of runs before and after label matching across different values of K. The main distinct solutions – the major modes for each K as identified by the Markov clustering algorithm – are similar to those presented by Kopelman *et al.* (2009) with of the clique approach. In addition to the major modes, CLUMPAK further identifies minor modes for $K = 3$ (13 runs), $K = 4$ (7 runs), $K = 5$ (4 runs), and $K = 6$ (10 runs). Figure 4 includes the minor modes for $K = 3$ and $K = 6$. Interestingly, the minor mode for $K = 6$ distinguishes the Tunisian Jewish population from other Jewish populations. This distinction, which was not present in the major modes reported by Kopelman *et al.* (2009), was in fact observed by Kopelman *et al.* when the population structure of Jewish individuals was examined separately. We might argue, however, that by simplifying the user experience of examining minor modes, CLUMPAK can uncover biologically interesting minor clustering solutions that might otherwise go unnoticed.

To demonstrate the comparison of models using CLUMPAK, we ran STRUCTURE a second time on the same data set of 399 individuals using the admixture model along with the locprior model (i.e. supervised clustering), which takes into consideration sampling locations of individuals, treating separate population identifiers as separate locations in prior information to assist clustering. This option is recommended for cases in which the inferred population structure is weak, or the populations examined are very closely related (Hubisz *et al.* 2009). We ran the locprior model with $K = 6$, comparing the results to those obtained without the use of the locprior model.

Figure 5 illustrates the comparison of 40 runs obtained for $K = 6$ without the locprior model to 40 runs with the locprior model. Visual comparison of the results obtained under the two models is simplified using the CLUMPAK alignment across the two models (as well as across multiple modes obtained under the same model). This comparison demonstrates the tendency of the supervised approach to intensify weak population structure within the *a priori* populations, leading to somewhat different inferences. The 'compare models' option of CLUMPAK further provides users with CLUMPP similarity scores between the modes that were identified by the two models. Indeed, the similarity score between the major modes obtained under the two models (0.79) is lower than the similarity score for the major and minor modes of the unsupervised runs (0.85), giving further indication that the results obtained under the two models are not identical.

Discussion

Analysis of population structure using model-based clustering methods is a complex task that requires careful handling (Pritchard *et al.* 2000; Rosenberg *et al.* 2001b; Evanno *et al.* 2005; Weiss & Long 2009). Due to the inherent stochasticity of STRUCTURE and other genetic clustering programs, independent runs of these programs often arrive at distinct solutions, and thus, understanding the replicability of the results and distinguishing distinct solutions are not always trivial tasks. We have developed CLUMPAK to simplify and enhance the user experience in evaluating, comparing, and displaying the results produced by these programs. CLUMPAK treats the problem of grouping STRUCTURE-like runs as a graph-based clustering problem, providing a useful method for separating distinct solutions within K values, averaging runs belonging to the same mode, and comparing runs across K values. It additionally supports the possibility of comparing different programs, models, and subsets of the data and identifying a preferred K value via the methods of Evanno *et al.* (2005) and Pritchard *et al.* (2000). Using simulations and an example data set of Kopelman *et al.* (2009), we found that CLUMPAK can identify major and minor solutions observed in replicate runs, potentially facilitating the assessment of biologically interesting clustering behaviour that is only visible in minor modes. We have also validated the dynamic procedure by which CLUMPAK identifies the optimal similarity threshold for each value of K, finding that this approach, set as the CLUMPAK default, is preferable to a fixed threshold approach.

Three other tools are currently available to assist users with postprocessing and presentation of the results from STRUCTURE-like programs – the aforementioned CLUMPP (Jakobsson & Rosenberg 2007), DISTRUCT (Rosenberg 2004), and STRUCTURE HARVESTER (Earl & Vonholdt 2012). As noted earlier, a typical postprocessing effort has involved use of two or all three of these programs. CLUMPAK combines many of the features of these existing tools, incorporating calls to CLUMPP and DISTRUCT, and providing users with a single program to accomplish primary aspects of postprocessing (Table 1). Most of the options provided by DISTRUCT and CLUMPP also appear in CLUMPAK, including the choice of algorithm in CLUMPP, and colour options, labelling, and order of populations in DISTRUCT. STRUCTURE HARVESTER is used for preparing CLUMPP input files and for determining the most suitable K according to the method of

Table 1 Programs that aid in the postprocessing of results obtained from Structure-like methods

Software	Platform	Input type/format	Description	References
Structure Harvester	Online & downloadable Python script	Zipped Structure result files	Determines a choice of *K* Produces input files for Clumpp	Earl & Vonholdt (2012)
Clumpp	Unix, Linux, Dos, Mac	Clumpp format*	Deals with label switching within a single *K* value Evaluates the similarity of replicate runs	Jakobsson & Rosenberg (2007)
Distruct	Unix, Linux, Dos, Mac	Distruct format*,†	Produces graphical display for a single replicate run	Rosenberg (2004)
Clumpak	Online & Unix, Linux, Mac	Multiple formats, enabling direct input from Structure and other Structure-like programs	Deals with label switching within a single *K* value Deals with label switching across multiple *K* values Clusters replicate runs into distinct modes Produces graphical displays for each mode within each *K* Determines a choice of *K*	This article

*Input files can be obtained from the output of Structure-like programs.
†One of the input files can be obtained from the output of Clumpp.

Evanno *et al.* (2005); the former functionality is subsumed in Clumpak, and the latter appears in Clumpak via the 'Best K' feature. Beyond its incorporation of features of these programs in a single convenient form, Clumpak is the first to automate the process of distinguishing between distinct solutions for a single *K* value and to perform cluster matching across different *K* values. Finally, Clumpak offers a natural visualization for comparing results obtained using different models, programs, or subsets of the data.

One view of 'genuine multimodality' in clustering studies, as opposed to label switching, is that it represents a failure of clustering programs to identify a single optimal mode, and that multimodality is simply an algorithmic artefact. From this idealistic standpoint, widely used algorithms such as Structure are problematic, and rather than facilitating the analysis of multimodality as a feature of clustering results, it would be preferable to improve algorithms to eliminate it. However, a pragmatic approach to available methods regards genuine Structure multimodality as a regular feature of the analysis that can be used to assist with biological interpretations (e.g. Rosenberg *et al.* 2001a; Wang *et al.* 2007; Kopelman *et al.* 2009). Especially for complex data sets, different modes for a given data set can reflect the existence of population groupings that are comparably supported (Pritchard *et al.* 2000), so that exploration of multiple modes contributes information to an analysis. For example, in examining how distinct solutions – both major and minor modes – change across a range of *K* values, Wang *et al.* (2007) observed multimodality at certain values of *K*, but found that solutions for larger *K* could be viewed as refinements of several of the modes observed at lower *K*. The distinct modes at smaller *K* each possessed some of the biological structure made apparent at larger *K* – structure portended in full by the collection of modes at small *K* – and they thus contributed to the understanding of major subdivisions in Native American population structure.

Finally, we note that although we have implemented Clumpak primarily for use specifically with Structure and Admixture, it is suitable for use with any program that generates cluster membership coefficients, provided the results produced by these programs are appropriately formatted for Clumpak. Notably, each of the model-based clustering programs requires that users make appropriate choices regarding the underlying model and optional running parameters (for example, running Structure long enough). Although Clumpak does not guide the users regarding these clustering choices, making sense of the results of such programs is rendered easier by the use of Clumpak. If the number of distinct clusters of runs inferred by Clumpak is too large, however, or if the similarity scores between replicate runs are low, then the Structure-like program might be uninformative or might have employed inappropriate settings; in these circumstances, we advise the user to inspect the original outputs and parameter choices for the program used to obtain the membership matrices.

Acknowledgements

We are grateful to Stijn van Dongen for helpful suggestions. We thank Ofer Chay for help with establishing the Web server, and Sohini Ramachandran and Aaron Behr for early testing and feedback. This work was supported by the Edmond J. Safra postdoctoral fellowship at Tel Aviv University (NMK), by the Israel Science Foundation grant 1265/12 (IM), and by NIH grant R01 HG005855 (NAR).

References

Alexander DH, Lange K (2011) Enhancements to the ADMIXTURE algorithm for individual ancestry estimation. *BMC Bioinformatics*, **12**, 246.

Alexander DH, Novembre J, Lange K (2009) Fast model-based estimation of ancestry in unrelated individuals. *Genome Research*, **19**, 1655–1664.

Breseghello F, Sorrells ME (2006) Association mapping of kernel size and milling quality in wheat (*Triticum aestivum* L.) cultivars. *Genetics*, **172**, 1165–1177.

Chen C, Durand E, Forbes F, François O (2007) Bayesian clustering algorithms ascertaining spatial population structure: a new computer program and a comparison study. *Molecular Ecology Notes*, **7**, 747–756.

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

Corander J, Waldmann P, Marttinen P, Sillanpaa MJ (2004) BAPS 2: enhanced possibilities for the analysis of genetic population structure. *Bioinformatics*, **20**, 2363–2369.

Corander J, Marttinen P, Siren J, Tang J (2008a) Enhanced Bayesian modelling in BAPS software for learning genetic structures of populations. *BMC Bioinformatics*, **9**, 539.

Corander J, Siren J, Arjas E (2008b) Bayesian spatial modeling of genetic population structure. *Computational Statistics*, **23**, 111–129.

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

Driscoll CA, Menotti-Raymond M, Roca AL *et al.* (2007) The Near Eastern origin of cat domestication. *Science*, **317**, 519–523.

Durand E, Jay F, Gaggiotti OE, François O (2009) Spatial inference of admixture proportions and secondary contact zones. *Molecular Biology and Evolution*, **26**, 1963–1973.

Earl DA, Vonholdt BM (2012) Structure Harvester: a website and program for visualizing Structure output and implementing the Evanno method. *Conservation Genetics Resources*, **4**, 359–361.

Enright AJ, Van Dongen S, Ouzounis CA (2002) An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, **30**, 1575–1584.

Evanno G, Regnaut S, Gould J (2005) Detecting the number of clusters of individuals using the software STRUCTURE: a simulation study. *Molecular Ecology*, **14**, 2611–2620.

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

Falush D, Stephens M, Pritchard JK (2007) Inference of population structure using multilocus genotype data: dominant markers and null alleles. *Molecular Ecology Notes*, **7**, 574–578.

Faust K, Sathirapongsasuti JF, Izard J *et al.* (2012) Microbial co-occurrence relationships in the human microbiome. *PLoS Computational Biology*, **8**, e1002606.

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

Frichot E, Mathieu F, Trouillon T, Bouchard G, François O (2014) Fast and efficient estimation of individual ancestry coefficients. *Genetics*, **196**, 973–983.

Gao H, Williamson S, Bustamante CD (2007) A Markov chain Monte Carlo approach for joint inference of population structure and inbreeding rates from multilocus genotype data. *Genetics*, **176**, 1635–1651.

Gilbert KJ, Andrew RL, Bock DG *et al.* (2012) Recommendations for utilizing and reporting population genetic analyses: the reproducibility of genetic clustering using the program STRUCTURE. *Molecular Ecology*, **21**, 4925–4930.

Gompert Z, Buerkle CA (2013) Analyses of genetic ancestry enable key insights for molecular ecology. *Molecular Ecology*, **22**, 5278–5294.

Guillot G (2008) Inference of structure in subdivided populations at low levels of genetic differentiation—the correlated allele frequencies model revisited. *Bioinformatics*, **24**, 2222–2228.

Guillot G, Estoup A, Mortier F, Cosson JF (2005) A spatial statistical model for landscape genetics. *Genetics*, **170**, 1261–1280.

Hoggart CJ, Parra EJ, Shriver MD *et al.* (2003) Control of confounding of genetic associations in stratified populations. *American Journal of Human Genetics*, **72**, 1492–1504.

Horn RA, Johnson CR (1991) *Topics in matrix analysis*. Cambridge University Press, Cambridge.

Hubisz MJ, Falush D, Stephens M, Pritchard JK (2009) Inferring weak population structure with the assistance of sample group information. *Molecular Ecology Resources*, **9**, 1322–1332.

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

Huelsenbeck JP, Andolfatto P, Huelsenbeck ET (2011) Structurama: Bayesian inference of population structure. *Evolutionary Bioinformatics*, **7**, 55–59.

Jain AK, Dubes RC (1988) *Algorithms for clustering data*. Prentice-Hall, Upper Saddle River, NJ.

Jakobsson M, Rosenberg NA (2007) CLUMPP: a cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure. *Bioinformatics*, **23**, 1801–1806.

Jakobsson M, Scholz SW, Scheet P, *et al.* (2008) Genotype, haplotype and copy-number variation in worldwide human populations. *Nature*, **451**, 998–1003.

Kaufman L, Rousseeuw PJ (1990) *Finding groups in data: an introduction to cluster analysis*. Wiley, New York, NY.

Kopelman NM, Stone L, Wang C *et al.* (2009) Genomic microsatellites identify shared Jewish ancestry intermediate between Middle Eastern and European populations. *BMC Genetics*, **10**, 80.

Manel S, Gaggiotti OE, Waples RS (2005) Assignment methods: matching biological questions with appropriate techniques. *Trends in Ecology & Evolution*, **20**, 136–142.

Parker HG, Kim LV, Sutter NB *et al.* (2004) Genetic structure of the purebred domestic dog. *Science*, **304**, 1160–1164.

Pella J, Masuda M (2006) The Gibbs and split-merge sampler for population mixture analysis from genetic data with incomplete baselines. *Canadian Journal of Fisheries and Aquatic Sciences*, **63**, 576–596.

Pritchard JK, Donnelly P (2001) Case-control studies of association in structured or admixed populations. *Theoretical Population Biology*, **60**, 227–237.

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

Ratnasingham S, Hebert PD (2013) A DNA-based registry for all animal species: the Barcode index number (BIN) system. *PLoS ONE*, **8**, e66213.

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

Rosenberg NA, Burke T, Elo K *et al.* (2001a) Empirical evaluation of genetic clustering methods using multilocus genotypes from 20 chicken breeds. *Genetics*, **159**, 699–713.

Rosenberg NA, Woolf E, Pritchard JK *et al.* (2001b) Distinctive genetic signatures in the Libyan Jews. *Proceedings of the National Academy of Sciences of the United States of America*, **98**, 858–863.

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

Shringarpure S, Xing EP (2009) mStruct: inference of population structure in light of both genetic admixing and allele mutations. *Genetics*, **182**, 575–593.

Tang H, Peng J, Wang P, Risch NJ (2005) Estimation of individual admixture: analytical and study design considerations. *Genetic Epidemiology*, **28**, 289–301.

Thornsberry JM, Goodman MM, Doebley J *et al.* (2001) Dwarf8 polymorphisms associate with variation in flowering time. *Nature Genetics*, **28**, 286–289.

Van Dongen S, 2000 *Graph clustering by flow simulation*. PhD thesis, University of Utrecht, Utrecht.

Van Dongen S (2008) Graph clustering via a discrete uncoupling process. *SIAM Journal on Matrix Analysis and Applications*, **30**, 121–141.

Wang S, Lewis CM, Jakobsson M *et al.* (2007) Genetic variation and population structure in Native Americans. *PLoS Genetics*, **3**, e185.

Weiss KM, Long JC (2009) Non-Darwinian estimation: my ancestors, my genes' ancestors. *Genome Research*, **19**, 703–710.

Whitfield CW, Behura SK, Berlocher SH *et al.* (2006) Thrice out of Africa: ancient and recent expansions of the honey bee, *Apis mellifera*. *Science*, **314**, 642–645.

Zhao S, Zheng P, Dong S *et al.* (2013) Whole-genome sequencing of giant pandas provides insights into demographic history and local adaptation. *Nature Genetics*, **45**, 67–71.

N.M.K. and I.M conceived and supervised the study, with advice from M.J. and N.A.R.. J.M. and N.M.K. coded and tested the software. N.M.K., I.M. and N.A.R. wrote the manuscript.

Data Accessibility

CLUMPAK is freely available for online use and for download at http://clumpak.tau.ac.il. The microsatellite data of Kopelman *et al.* (2009) used in the empirical example are available at https://rosenberglab.stanford.edu/datasets.html.

Supporting Information

Additional Supporting Information may be found in the online version of this article:

Note S1 The relation between similarity scores of replicate runs to values of SD_{RUNS} and SD_{INDIVS} and to the assigned dynamic threshold.

Table S1 Mean CLUMPP similarity scores for the unimodal simulation scenario at $K = 3$, under different choices of values of SD_{RUNS} and SD_{INDIVS}.

Table S2 Mean value of the dynamic threshold for the unimodal simulation scenario at $K = 3$, under different choices of values of SD_{RUNS} and SD_{INDIVS}.

Table S3 Mean value of the dynamic threshold for the bimodal simulation scenario at $K = 3$, under different choices of values of SD_{RUNS} and SD_{INDIVS}.

Fig. S1 Jaccard similarity scores between the clustering solution obtained by CLUMPAK and the 'true' (simulated) partitioning in a bimodal case, as a function of of SD_{RUNS}, SD_{INDIVS}, and either a fixed threshold value or a dynamic threshold.

Fig. S2 Jaccard similarity scores between the clustering solution obtained by CLUMPAK and the 'true' (simulated) partitioning in a bimodal case, as a function of SD_{RUNS}, SD_{INDIVS}, and either a fixed threshold value or a dynamic threshold.

Fig. S3 Jaccard similarity scores between the clustering solution obtained by CLUMPAK and the 'true' (simulated) partitioning in a bimodal case, as a function of SD_{RUNS}, SD_{INDIVS}, and the value of the MCL parameter r.

Appendix S1 Seed vectors used in the simulations.

Appendix S2 The procedure for setting the parameters of a Dirichlet distribution to obtain predetermined expectations and variances.