Documentation for CLUMPAK, version 1.1

January 18, 2014

Note: CLUMPAK is currently in its beta version and we are improving the user interface. We will appreciate any feedback and bug reports

CLUMPAK aids users in automating the process of analyzing the results of genotype clustering programs such as STRUCTURE. CLUMPAK separates groups of runs representing distinct solutions, and identifies an optimal cluster label alignment across different values of *K*, simplifying the comparison of clustering results across *K*. In addition, CLUMPAK implements a method for the identification of a preferred choice of *K*, and enables easy visualization for comparing solutions obtained by different programs, models, or subsets of data.

Contents

1 Introduction

1.1 Overview

Clustering individuals into populations, based on multi-locus genotypes, has become a critical step in population genetics studies. Many different programs have been developed in order to face the challenge of dividing individuals into a predefined number of populations, *K*. The most widely used of these programs is STRUCTURE (Pritchard *et al.* 2000; Falush *et al.* 2003; Falush *et al.* 2007; Hubisz *et al.* 2009). We refer to these methods as STRUCTURE-like (Weiss and Long 2009). The result of a single cluster analysis is typically given as a matrix, $[Q]_{ik}$, describing the membership coefficients of each individual *i* in each of the *K* clusters. These coefficients can be interpreted as membership probabilities, or as the fraction of the genome originating from each cluster.

Many of the STRUCTURE-like programs are stochastic, and have the propensity of producing different outcomes for replicate runs, even when the same choice of model and parameters is used. For this reason, users often conduct multiple runs for the same model and parameters. Distinct solutions can be the result of multimodality in the solution space, or the result of label switching between clusters. In addition, since the user needs to define the number of clusters, many times a range of *K* values is used, each with multiple independent runs. Thus, the user is faced with the challenge of summing up and comparing hundreds, and sometimes thousands, of runs, within and across *K* values.

CLUMPAK - Clustering Markov Packager Across *K* - was developed in order to aid users in analyzing the results of STRUCTURE-like programs. The software offers a few alternative modes of action, with the main one offering a full pipeline for the summation and graphical illustration of the results obtained by STRUCTURE-like programs. Additional features allow comparison of programs or models, and selecting the preferred value of *K* according to the methods of Evanno *et al.* (2005) or Pritchard et al. (2000). The input to CLUMPAK are the *Q*-matrices obtained by STRUCTURE or by other STRUCTURE-like programs, properly formatted to match one of the three input formats supported by CLUMPAK.

1.2 Aims

CLUMPAK was designed to aid users in four main objectives: (1) Separate distinct solutions obtained from STRUCTURE-like programs. (2) Compare and align solutions obtained for different *K* values. (3) Compare results obtained using different models/ programs. (4) Indicate the preferred value of *K* according to Evanno *et al.* (2005) and Pritchard et al. (2000).

2 Getting started

2.1 Availability & web-server

The online version of CLUMPAK is available at http://clumpak.tau.ac.il. Detailed instructions on how to use the server are provided online. Users can also consult this manual, as using the server is much like using CLUMPAK's command line version.

2.2 Installation & requirements

Linux: the current version was tested on CentOS 5.2, using Perl 5.8.8. To install CLUMPAK on a local machine follow these steps:

(1) Download the current version from http://clumpak.tau.ac.il/download.html and unzip it on your machine. You can delete Mac_OSX_files.zip.

(2) Make sure that you have Perl 5.8.8 (or a later one) installed.

(3) Make sure that the following Perl modules are installed: Getopt::Long; File::Slurp; File::Path; List::MoreUtils; PDF::API2; PDF::Table; File::Basename; List::Permutor; GD::Graph::lines; GD::Graph::Data; Getopt::Std; List::Util; File::Slurp; Scalar::Util; Statistics::Distributions; Archive::Extract; Data::PowerSet; Array::Utils.

(4) CLUMPAK relies on having executables for DISTRUCT (Rosenberg 2004), CLUMPP (Jakobsson and Rosenberg 2007), and MCL (Enright *et al.* 2002; Van Dongen 2008). Linux executables are included in the downloadable zip file. However, users should verify that these executables are working properly on their system. If not, CLUMPP and DISTRUCT executables can be obtained from: http://www.stanford.edu/group/rosenberglab/clumpp.html
http://www.stanford.edu/group/rosenberglab/distruct.html

MCL source codes are included in CLUMPAK's zip, or can be obtained from http://micans.org/mcl/. Start by compiling MCL's source codes on your own system, since the problems are most likely related to MCL compilation and not to DISTRUCT or CLUMPP.

Mac: the current version was tested on MAC OSX 10.9.3. To install CLUMPAK on your local Mac, follow these steps:

(1) Download the current version from http://clumpak.tau.ac.il/download.html and unzip it on your machine.

(2) Make sure that you have Perl 5.8.8 (or a later one) installed.

(3) Make sure that the following Perl modules are installed: Getopt::Long; File::Slurp; File::Path; List::MoreUtils; PDF::API2; PDF::Table; File::Basename; List::Permutor; GD::Graph::lines; GD::Graph::Data; Getopt::Std; List::Util; File::Slurp; Scalar::Util; Statistics::Distributions; Archive::Extract; Data::PowerSet; Array::Utils.

(4) Install Ghostscript for Mac. There is no official distribution, but try version 9.15 from this link: http://pages.uoregon.edu/koch/

(5) Replace the MCL folder under CLUMPAK's source folder with the MCL folder located in the Mac_OSX_files.zip.

(6) Replace CLUMPP's executable file (in CLUMPP's subfolder, under CLUMPAK's source folder) with CLUMPP's executable located in the Mac_OSX_files.zip.

(7) Replace DISTRUCT's executable file (in DISTRUCT's subfolder, under CLUMPAK's source folder) with DISTRUCT's executable from located in the Mac_OSX_files.zip.

3 Input files and optional parameters

The input to all of CLUMPAK's features includes the result files as obtained through the STRUCTURE (or a STRUCTURE-like) program. Owing to graphical limitations of the DISTRUCT software, we currently support **up to 5,000 individuals in a dataset**. CLUMPAK supports three formats for result files: (1) STRUCTURE format. This is the full result file as obtained through STRUCTURE (see structure_output.txt in examples.zip). We recommend using POP_DATE = 1 when running STRUCTURE, as CLUMPAK can use population IDs for graphical purposes.
(2) A shortened/truncated STRUCTURE format (see truncated_structure_output.txt in examples.zip). This format, too, is referred to as STRUCTURE when running CLUMPAK.
(3) Simple *Q*-matrix (see simple_Q_output.txt in examples.zip). This format is referred to as ADMIXTURE. Since this format is lacking population IDs, we recommend providing an additional *populations_file* with population IDs or names, see below.

In case a different STRUCTURE-like program was used, some modifications might be required in order to match formats 2 or 3. In case ADMIXTURE (Alexander *et al.* 2009; Alexander and Lange 2011) was used, the *Q*-file output of ADMIXTURE can be used with no further modifications, as it matches format 3 (ADMIXTURE format).

Here is an explanation on the two *Q*-matrices formats (2 & 3 above): if NUM_INDS is the number of individuals in your data, and *K* is the number of predefined populations (i.e. clusters), then the simple *Q*-matrix (format 3) should contain NUM_INDS rows with *K* columns

per row. Each row shows the membership coefficients for one individual. See *Simple_Q.txt* for an example with NUM_INDS = 10 and *K* = 4. A truncated STRUCTURE format (format 2) is different from the simple *Q*-matrix format as it contains four or five additional columns before the membership coefficients (depending if population IDs are included). CLUMPAK can use population ID for graphical purposes. The other columns are ignored.

If simple *Q*-matrices are provided (i.e., ADMIXTURE format), an additional file can be uploaded, which contains the populations IDs or names for the individuals in the data – a *populations_file* (see *toy_data_populations_file.txt* in *examples.zip*). This file should have the same number of lines as the *Q*-matrix file (i.e. NUM_INDS rows), where each line contains an integer that codes for the population ID, or a population name. The order of individuals in result files (i.e. *Q*-matrices) and the *populations_file* should match. Population IDs (or names) are used for graphical purposes, and uploading this file is recommended if samples were obtained from multiple populations.

How to zip your result files: Result files can be zipped together regardless of their *K* value, or zipped separately for each *K*, and then zipped together to one final zip. Zip files can contain sub-folders or sub-zips, but CLUMPAK expects text files and zip files only. Other files will generate an error. If you are using Linux, you can use the command 'zip' to zip files. If you are using Windows, you can use WinRAR with the 'zip' option to zip files. For Mac users, zip files compressed on Mac may contain hidden files and folders – so we advice zipping files in the following manner:

1) put all your results in one folder 'results_folder'
2) Go to the command line terminal and type
   > zip -r results.zip results -x "*.DS_Store"
3) Type the following command to delete any hidden files
   > zip -d results.zip __MACOSX/*

3.1 'Advanced options' files

For the main pipeline, 'DISTRUCT for many *K*'s', and 'Compare' features, there are a number of other optional input files:

(1) *labels_file* which contains text labels for population IDs (see *toy_data_labels.txt* in *examples.zip*). This option is supported only for formats 2 & 3 above, which contain population IDs (i.e., STRUCTURE was used with POP_DATA=1). If provided, the order of populations in the produced figure will reflect their order in the *labels_file*, and the

labels will be used below the figure. In case it is not provided, population codes will be extracted from the results files.

(2) *colors_file* which contains colors to be used in the produced figures (see *colors_file.txt* in *examples.zip*). Colors recognized by CLUMPAK are those that are recognized by DISTRUCT, please consult the DISTRUCT manual for a full list of colors. Colors should be numbered and ordered, such that the number of colors is equal to - or larger than - the largest *K* value in the result files.

(3) *drawparams_file* which confronts the format of DISTRUCT's *drawparams* file (see *drawparams.txt* in *examples.zip*). Please consult the DISTRUCT manual for additional details.

3.2 Additional parameters

3.2.1 Input format

For each of CLUMPAK's features, the default format is STRUCTURE. Alternative formats can be specified in the command line. For the 'Main pipeline' and the 'DISTRUCT for many *K*'s' features, input format can be set to ADMIXTURE in the following manner:
--inputtype admixture

Under the 'Compare' feature, the format can be changed for each of the two data sets:
--firstinputtype admixture AND/OR *--secondinputtype admixture*

Under the 'Best K' feature, the alternative format is a *log_prob_file.txt* (see description in section 4.4):
--inputtype lnprobbyk

3.2.2 MCL threshold

The MCL algorithm is used by CLUMPAK both in the 'Main pipeline' and in the 'Compare' feature, to identify modes within single *K* values. The MCL algorithm can produce clusters with varying levels of granularity. In CLUMPAK, granularity is controlled via a threshold for the inclusion of edges in the graph of replicate runs. Under the default settings, CLUMPAK automatically sets this threshold in accordance to graph properties, separately for each *K* value. However, users can set the threshold to a fixed value, thus affecting clustering resolution:

--mclthreshold <THRESHOLD>

3.2.3 CLUMPP options within single *K* values

CLUMPAK makes use of CLUMPP for aligning single runs within *K* values, both in the 'Main pipeline' and in the 'Compare' features. In turn, CLUMPP uses one of three algorithms for attempting to find the optimal alignment of R replicate runs within a single *K* value. These algorithms are termed *FullSearch*, *Greedy* and *LargeKGreedy* (see CLUMPP's manual for a full description). Under default settings, CLUMPAK uses the *LargeKGreedy* algorithm with 2,000 random input sequences. Users can change the choice of algorithm by setting M to 1 (*FullSearch*), 2 (*Greedy*), or 3 (*LargeKGreedy*):
--clumppsearchmethod <M>

If M=2 or M=3 are chosen, the user should specify the number of input orders (REPEATS) to be tested (REPEATS_NUM):
--clumpprepeats <REPEATS_NUM>

clumpprepeats should normally be in the range of 100-5,000, and using larger values is not recommended, as it may lead to computation time longer than that of the *FullSearch* algorithm. Please refer to CLUMPP's manual for further details on these algorithms and estimated running times – which are determined by the number of individuals in the data, the number of runs, and the range of *K* values used.

By default, CLUMPAK uses a rule of thumb to check the expected running time under the settings provided by the user, rejecting runs which are expected to be too long. However, command-line users have two ways to change this check-up:

1) By changing the value of the threshold which is used for this check-up. The value is specified in the file 'CLUMPAK_CONSTS_and_Functions.pm', located in CLUMPAK's execution folder. The following line contains the value and can be modified to push this threshold up or down:
*use constant MAX_D => 10**13;*

(Comment: 10**13 means 10^{13})

2) By setting a check-up flag to 0. The flag is specified in the file 'CLUMPAK_CONSTS_and_Functions.pm'. If set to 0, CLUMPAK will not pose any threshold, both on the process of aligning single runs within *K* values, and on the process of for aligning the modes obtained for different *K* values (see below). The following line should be modified. Change from:
use constant COMPUTE_TIME_CHECK => 1;

to

use constant COMPUTE_TIME_CHECK => 0;

NOTE: CLUMPP parameters can drastically change the running time of CLUMPAK. Table 1 presents the running times that were observed on our machines for the toy data provided online (#individuals=399, *K*=2-6, #runs per *K*=20), for different choices of algorithm for the alignment of runs within *K* values. For further details on the algorithms and the REPEATS parameters, please refer to CLUMPP's manual.

ALGORITHM	**REPEATS**	**TIME**
Greedy	500	153 minutes
Greedy	1,000	206 minutes
Greedy	2,000	640 minutes
Greedy	5,000	1600 minutes
Greedy	8,000	2435 minutes
Greedy	10,000	3040 minutes
LargeKGreedy	500	2.666 minutes
LargeKGreedy	1,000	3.25 minutes
LargeKGreedy	***2,000***	***5.5 minutes***
LargeKGreedy	5,000	12 minutes
LargeKGreedy	8,000	18.5 minutes
LargeKGreedy	10,000	24 minutes

Table 1*:* Running times observed for the toy data provided on the web server (#individuals=399, *K*=2-6, #runs per *K*=20), for different choices of algorithm for aligning single runs within *K* values. The default CLUMPAK setting in marked in bold.

3.2.4 CLUMPP options between *K* values

CLUMPAK incorporates a novel procedure for aligning the modes obtained for different *K* values. This procedure is relevant to the 'Main pipeline', 'Compare', and 'Distruct for many *K*'s' features. In our implementation, for small *K* (up to *K*=8) we consider all possible permutations to identify this optimal alignment, while for larger *K* values we use a greedy procedure. However, command-line users have two ways to interfere with these choices:

1) Users can change the threshold (*K*=8) in the file 'CLUMPAK_CONSTS_and_Functions.pm' in CLUMPAK's execution folder, by modifying the line:

   use constant MAX_K => 8;

   The value can be set to any other integer value.

2) The file 'CLUMPAK_CONSTS_and_Functions.pm' contains a flag, COMPUTE_TIME_CHECK, which, if set to 0, will prevent CLUMPAK from posing any threshold, both on the process of aligning single runs within *K* values, and on the process of for aligning the modes obtained for different *K* values. The following line should be changed. Change from:

   use constant COMPUTE_TIME_CHECK => 1;
   to
   use constant COMPUTE_TIME_CHECK => 0;

4 Usage options

CLUMPAK offers four modes of action – the main pipeline, 'DISTRUCT for many *K*'s', 'Compare', and 'Best *K*'.

4.1 Main pipeline

The main pipeline of CLUMPAK aims at helping the users through the entire process of summing and presenting the results of STRUCTURE-like programs. The input for the main pipeline is a set of STRUCTURE runs, or *Q*-matrices obtained from STRUCUTRE-like programs, produced for the same data set for a range of *K* values. For example, the input might be made of 10 runs for each *K* value, with *K* ranging between 2 to 10. Owing to graphical limitations of the DISTRUCT software, we currently support **up to 5,000 individuals in a dataset**.

CLUMPAK can use population codes to order individuals according to their source population. If you are using STRUCTURE, we recommend using POP_DATE = 1. If populations codes are missing from your results (i.e. using STRUCTURE with POP_DATA = 0), CLUMPAK will assume that all the individuals were sampled from one source population. If your input is in the form of simple *Q*-matrices, you are encouraged to upload an additional file - a *populations_file* - which is used to identify the population code or name for each individual (see **section 3**). If this file is

missing, CLUMPAK will assume that all the individuals were sampled from one source population.

Optional input files are the *populations_file*, *labels_file*, *colors_file* and *drawparams_file* (see **section 3**).

The basic command-line for the main pipeline is as follows:
> perl CLUMPAK.pl --id <INTEGER> --dir <CLUMPAK_OUTPUT_DIR> --file <results.zip>

Note: The value of the integer provided as *id* is of no significance.

Additional options and parameters (see section 3 for details):
--labels labels_file.txt
--colors colors_file.txt
--drawparams drawparams.txt
--mclthreshold <THRESHOLD>
--clumppsearchmethod <M>
--clumppgreedyoption 1
--clumpprepeats <REPEATS >
--inputtype admixture
--podtopop populations_file

4.2 DISTRUCT for many K's

The 'Distruct for many *K*'s' feature aims at helping users align single results obtained for different K values. These single results might be individual runs or averages obtained for multiple independent runs. The required input is a zip file which contains a single result for each *K* value. For example, if the *K* value range is between 1 to 10, then 10 result files should be zipped together. We currently support **up to 5,000 individuals in a dataset**, due to graphical constraints.

CLUMPAK can use population codes to order individuals according to their source population. If you are using STRUCTURE, we recommend using POP_DATE = 1. If populations codes are missing from your results (i.e. using STRUCTURE with POP_DATA = 0), CLUMPAK will assume that all the individuals were sampled from one source population. If your input is in the simple *Q*-matrix format (i.e. ADMIXTURE format), you are encouraged to upload an additional file - a *populations_file* - which is used to identify the population code or name for each individual (see **section 3**). If this file is missing, CLUMPAK will assume that all the individuals were sampled from one source population.

Optional input files are the *populations_file*, *labels_file*, *colors_file* and *drawparams_file* (see **section 3**).

The basic command-line for 'DISTRUCT for many K's' is as follows:
> perl distructForManyKs.pl *--id <INTEGER> --dir <CLUMPAK_OUTPUT_DIR> --file <results.zip>*

Note: The value of the integer provided as *id* is of no significance.

Additional options (see section 3 for details):
--labels labels_file.txt
--colors colors_file.txt
--drawparams drawparams.txt
--inputtype admixture
--podtopop populations_file

4.3 Compare

The 'Compare' feature aims at helping users determine whether the results obtained for a single K value (using the same set of individuals) with different programs, modeling assumptions, or different subsets of markers, are significantly different. The required inputs are two zip files, each containing the results obtained from one model/program. The two sets of results will be compared. As in the other features, if your input is in the simple *Q*-matrix format (i.e. ADMIXTURE format), you are encouraged to upload an additional file - a *populations_file* - which is used to identify the population code or name for each individual (see **section 3**).

Optional input files are the *populations_file*, *label_file*, *colors_file* and *drawparams_file* (see **section 3**).

The basic command-line for 'Compare' is as follows:
> perl CompareDifferentPrograms.pl --id <INTEGER> --dir <CLUMPAK_OUTPUT_DIR> --firstfile model1.zip --secondfile model2.zip

Note: The value of the integer provided as *id* is of no significance. s

Additional options (see section 3 for details):
--labels labels_file.txt
--colors colors_file.txt
--drawparams drawparams.txt
--mclthreshold <THRESHOLD>
--clumppsearchmethod <M>

--clumppgreedyoption 1
--clumpprepeats <REPEATS_NUM>
--firstinputtype admixture
--secondinputtype admixture
--podtopop populations_file

4.4 Best *K*

There are two formats supported for this calculation: one option is to zip full STRUCTURE result files, which were produced for the same data set and for a range of *K* values. For example, the input might be composed of 10 runs for each *K* value, with *K* ranging between 2 to 10. A second option, which allows users of other STRUCTURE-like programs to perform this calculation, is to provide CLUMAPK with a *log_probability_file*, a text file containing two columns – the first designates *K* values, and the second designates the respective log probabilities (or log likelihoods) of the data. Each single run is represented by a line in this file, and multiple rows for each *K* value are expected (see toy_data_log_prob_file.txt in examples.zip). Note that the original paper by Evanno *et al.* (2005) focuses exclusively on the software STRUCTURE. It is up to the user to decide if and when it is appropriate to apply this method to the results of a different STRUCTURE-like program.

The basic command-line for 'Best K' for the first format (i.e. STRUCTURE files are uploaded in a zip file) is as follows:
> perl BestKByEvanno.pl --i <INTEGER> --d <CLUMPAK_OUTPUT_DIR> --f results.zip

If you would like to use the second format – i.e. a table with log probabilities – the command line is as follows:
> perl BestKByEvanno.pl --i <INTEGER> --d <CLUMPAK_OUTPUT_DIR> --f log_prob_file.txt --inputtype lnprobbyk

Note: The value of the integer provided as *id* is of no significance.

5 CLUMPAK outputs

If you are using CLUMPAK online, you will be able to download a zip file containing the results of your run. This zip will include intermediate outputs, allowing you to explore other directions of analysis and graphical display. Results will be available on our server for a month before being deleted. The outputs for each feature of CLUMPAK are explained below.

5.1 Main pipeline

The main results of the 'Main pipeline' include the images produced for the modes detected for different *K* values, as well as information of the average log probability of the data for each detected mode, the average similarity scores of runs within each mode, and details on which runs were assigned to each mode. Users who run the program through the web server should receive an email with a link to the results page. The headers of the figures on the results page indicate the *K* value, number of runs assigned to that specific mode out of the total number of runs, the average log probability for runs in the mode, and the average similarities between all pairs of runs within the mode. In addition, the results page includes links to two files which can be downloaded:

1) A pdf file that summarizes the main results, and includes figures and the number of runs assigned to each mode.
2) A zip file that contains all the results, including figures, information of the detected modes, and intermediate results. This zip is similar to the results folder obtained when running the program in the command line. Here is a description of the contents of this folder/zip:
    a. Figures for all the detected modes.
    b. A log file, *output.log*.
    c. A file providing addition information of detected modes, *detectedModesSummery.log*, which specifies the average log probability of the data for each detected mode and the average similarity scores of runs within each mode.
    d. For each *K*, the zip file contains a separate folder. Each *K* folder contains three or more folders: *CLUMPP.files*, *MCL.files*, and an additional folder for each mode detected for that *K*. The *CLUMPP.files* folder contains the input, output, and the parameter files used for running CLUMPP. The file *ClumppIndFile.output* in this folder contains the runs following the alignment within the mode. In addition, this folder contains a file *FilesToIndex*, which provides an index (starting from 0) for each of the original STRUCTURE/ADMIXTURE outputs provided for that *K* value. The *MCL.files* folder contains the inputs used for MCL, and the output, *MCL.clusters*. The number of lines in *MCL.clusters* reflects the number of modes for that *K*. Each line contains the indices of input files included in a specific cluster, as well as the average similarity score for pairs of runs in the mode.

Comment [IM1]: It is not specified what does this folder contains

Comment [IM2]: Not clear

5.2 DISTRUCT for many K's

The main results of 'DISTRUCT for many K's' are the images produced for all the result files uploaded. Users who run the program through the web server should receive an email with a link to the results page. The headers of the figures indicate the names of the result file presented. In addition, the results page includes links to two files which can be downloaded:

1) A pdf file which summarizes the main results, and includes figures and the corresponding file names.
2) A zip file which contains all the results, including figures and intermediate results. This zip is similar to the results folder obtained when running the program in the command line. Here is a description of the contents of this folder/zip:
    a. Figures for all the input files.
    b. A log file, *output.log*.
    c. A *input.files* folder, which contains the original input files, truncated input files, and input files following the alignment procedure between the input files. The aligned filed end with '.*ClumppPopFile*'.

5.3 Compare

The main results of 'Compare' include images produced for the detected modes for the two models that are compared, as well the similarities between all detected modes. Users who run the program through the web server receive an email with a link to the results page. The headers of the figures on the results page indicate the *K* value, number of runs assigned to that specific mode, and the corresponding model (out of the two). In addition, the results page includes links to two files which can be downloaded:

1) A pdf file which summarizes the main results.
2) A zip file which contains all the results, including figures and intermediate results. Here is a description of the contents of this folder/zip:
    a. Figures for the detected modes of the two models.
    b. A log file, *output.log*.
    c. A text file, *Comparison.Of.Models.txt*, which contains the similarity scores between the detected modes of the two models.
    d. A folder for each one of the two models, which contains CLUMPP's input and output files, and a file *FilesToIndex*, which provides an index (starting from 0) for each of the original STRUCTURE/ADMIXTURE outputs provided for that model. In addition, this folder contains the *MCL.files* folder, and an additional folder for each mode detected for that model.

5.4 Best *K*

The main results of the 'Best *K*' feature are two graphs: (1) DeltaK graph, which corresponds to the method of Evanno et al. (Evanno et al. 2005), and (2) and probability by *K* graph which plots the probability for each *K* according to Pritchard et al. (2000). Users who run the program through the web server should receive an email with a link to the results page. The results page includes a link to a zip file, which contains the graphs and the log file.

Comment [IM3]: It is quite odd that for the toy example these two methods produce very different results. Are you sure the prob-by-k graph is correct?

6 How to cite this program

The paper describing CLUMPAK has been submitted. Please email us to ask for a citation when it is required.

7 References

Alexander, D. H., J. Novembre and K. Lange, 2009 Fast model-based estimation of ancestry in unrelated individuals. Genome Res. **19:** 1655-1664.

Alexander, D. H., and K. Lange, 2011 Enhancements to the admixture algorithm for individual ancestry estimation. BMC Bioinformatics **12:** 246.

Enright, A. J., S. Van Dongen and C. A. Ouzounis, 2002 An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res. **30:** 1575-1584.

Evanno, G., S. Regnaut and J. Gould, 2005 Detecting the number of clusters of individuals using the software structure: A simulation study. Mol. Ecol. **14:** 2611-2620.

Falush, D., M. Stephens and J. K. Pritchard, 2003 Inference of population structure using multilocus genotype data: Linked loci and correlated allele frequencies. Genetics **164:** 1567-1587.

Falush, D., M. Stephens and J. K. Pritchard, 2007 Inference of population structure using multilocus genotype data: Dominant markers and null alleles. Mol. Ecol. Notes **7:** 574-578.

Hubisz, M. J., D. Falush, M. Stephens and J. K. Pritchard, 2009 Inferring weak population structure with the assistance of sample group information. Mol. Ecol. Resour. **9:** 1322-1332.

Jakobsson, M., and N. A. Rosenberg, 2007 Clumpp: A cluster matching and permutation program for dealing with label switching and multimodality in analysis of population structure. Bioinformatics **23:** 1801-1806.

Pritchard, J. K., M. Stephens and P. Donnelly, 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.

Van Dongen, S., 2008 Graph clustering via a discrete uncoupling process. SIAM J. Matrix Anal. Appl. **30:** 121-141.

Weiss, K. M., and J. C. Long, 2009 Non-darwinian estimation: My ancestors, my genes' ancestors. Genome Res. **19:** 703-710.