WebStruct and VisualStruct: web interfaces and visualization for *Structure* software implemented in a cluster environment

Jayashree B[1], Rajgopal S[2], Hoisington D[3], Prasanth VP[4], and Chandra S[5]

[1]Bioinformatics Unit, ICRISAT, Patancheru 502 324, India
[2]Advanced Technology Centre, TCS, Madhapur, Hyderabad 500081,India
[3]GTL-Biotechnology, ICRISAT, Patancheru 502 324, India
[4]Advanta India Limited, Secunderabad 500 003, India
[5] Department of Primary Industries, Tatura 3616, Vic, Australia

Abstract

Structure, is a widely used software tool to investigate population genetic structure with multi-locus genotyping data. The software uses an iterative algorithm to group individuals into "K" clusters, representing possibly K genetically distinct sub-populations. The serial implementation of this programme is processor-intensive even with small datasets. We describe an implementation of the program within a parallel framework. Speedup was achieved by running different replicates and values of K on each node of the cluster. A web-based user-oriented GUI has been implemented in PHP, through which the user can specify input parameters for the programme. The number of processors to be used can be specified in the background command. A web-based visualization tool "Visualstruct", written in PHP (HTML and Java script embedded), allows for the graphical display of population clusters output from Structure, where each individual may be visualized as a line segment with K colors defining its possible genomic composition with respect to the K genetic sub-populations. The advantage over available programs is in the increased number of individuals that can be visualized. The analyses of real datasets indicate a speedup of up to four, when comparing the speed of execution on clusters of eight processors with the speed of execution on one desktop. The software package is freely available to interested users upon request.

1 Introduction

The programme *Structure* [1] assigns individuals of unknown origin to populations based on their likelihood of occurrence in a population computed from estimated allele frequencies using an MCMC (Markov chain Monte Carlo) algorithm. This program is available in the public domain (http://pritch.bsd.uchicago.edu/software.html) and has been widely used in problems that require identification of population structure, detecting migrants [2, 3] and population admixtures [4]. The MCMC algorithm starts with an initial configuration of parameter values and iteratively updates a subset of the parameters to new values conditional on the current values of the other parameters and the data. In a single iteration all parameters are updated once. After sufficient number of iterations (n) the algorithm converges to the posterior distribution of all of the parameters for that dataset. The algorithm has now been extended to apply to dominant markers as well [5]. The program is computationally intensive and takes considerable time depending upon the number of iterations involved and the size of the dataset. We describe the usage of the structure program on compute clusters that considerably reduce analysis time. We also describe the availability of user interfaces that will help a user to use the application on a cluster without having to know or deal with the configuration of the cluster. Also available is a visualization tool “VisualStruct” that allows

doi:10.2390/biecoll-jib-2008-89

visualization of the results as individual line segments partitioned into colored components representing the populations to which the individual belongs. The software can be accessed[1] through web interfaces at http://hpc.icrisat.cgiar.org/webstructure/login.php.

2 Implementation of software

The software package uses the freely available *Structure* [1] programme. *Structure* has been implemented to run on clusters using the MPICH implementation of MPI (message passing interface). MPI being one of the more popular standards for writing parallel programs efficiently manages message buffers and has more than one freely available quality implementation. The MPICH wrapper has been written in the C programming language. Since the *Structure* algorithm itself is linear and does not lend itself to parallelization, speedup was achieved by running different replicates and values of K on each node of the cluster, where K represents distinct genetic sub-populations generated by the programme. Users can access the program through a web-based graphical user interface implemented using PHP. The entire application comprises over a dozen screens, which allow the user to input parameters, upload files, submit jobs and visualize the results. A web-based visualizer, VisualStruct, implemented in PHP and Javascript, provides a graphical display of the output. The output of this programme is similar to that of the distruct[6] software. Other software dependencies include the apache web server.

3 Results and Discussion

The resulting software carries out the same functionality as *Structure* with the difference that it can be implemented to work on any Beowulf cluster or SMPs (symmetric multiprocessors). At ICRISAT the software is being used on a Paracel four node cluster of 64-bit dual AMD Opteron processors.

Table 1: Comparison of overall runtime to process various number of individuals, loci, testing populations, iterations and burn-in period and MCMC running Structure on a desktop as opposed to a cluster of three or four 64-bit dual AMD opteron nodes of the Paracel high performance linux cluster.

No. of individuals	No. of loci	Testing populations	No. of iterations	Burn-in period	MCMC	Software	No. of CPUs	Overall runtime
48	100	2-5	2	100,000	100,000	*Structure*	1	4.5 hrs
48	100	2-5	2	100,000	100,000	WebStruct	8 (4 dual AMD Opterons)	1 hr
192	45	2-5	1	1,000,000	1,000,000	*Structure*	1	28 hrs
192	45	2-5	1	1,000,000	1,000,000	WebStruct	6 (3 dual AMD Opterons)	12.5 hrs
3000	35	2-5	1	1,000,000	1,000,000	*Structure*	1	18 days
3000	35	2-5	1	1,000,000	1,000,000	WebStruct	8 (4 dual AMD Opterons)	6 days 5hours

[1] username: Demo; password: demowebstruct

Table 1 indicates the reductions in run time obtained by using the programme on clusters given the same parameter settings. The input and output files for each of the programmes available in the package are outlined:

1. The first software module accepts data from the user in the form of excel sheets containing allele size data and converts them into (i) allele position data (ii) allele size data format compatible to *Structure* input.
2. The WebStructure module provides the web interface to *Structure* analysis (Figure 1). Through these pages the user can upload his dataset, provide input file information and data format, and set parameters that include values for the run length, ancestry model, allele frequency model and other advanced features such as computing the probability of the data. The user interfaces also allow setting of the number of simulations to run, saving parameter settings to files and retrieval of output files.

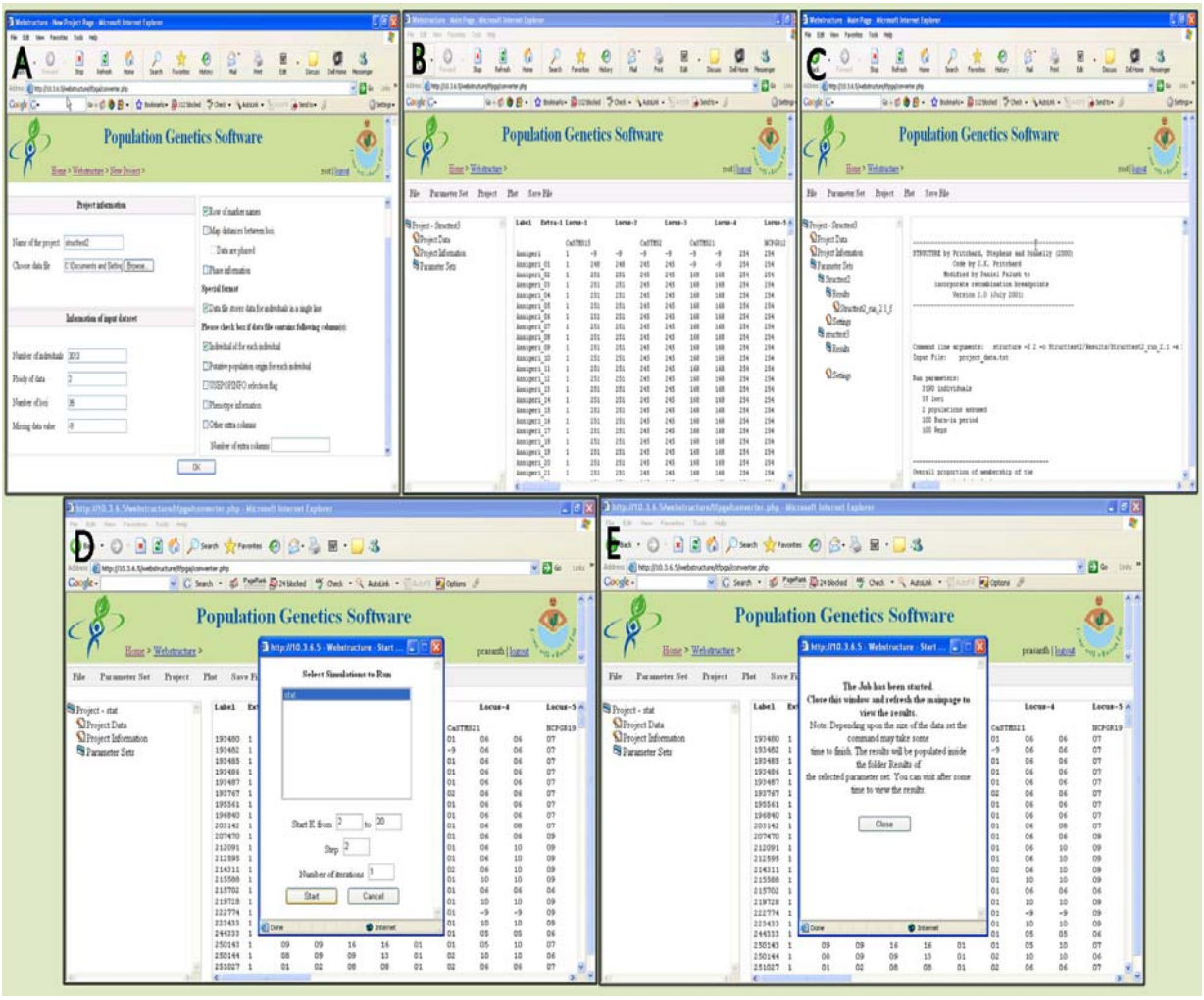

Figure 1: WebStructure interfaces (A) entering input file information (B) setting Structure parameters (C) invoking the Structure program (D) setting number of simulations and (E) executing a job.

3. The VisualStruct module provides visualization of *Structure* output and accepts three data formats – (i) the structure output file (ii) individual Q matrices (iii) population Q matrix where Q is a matrix of membership coefficients.

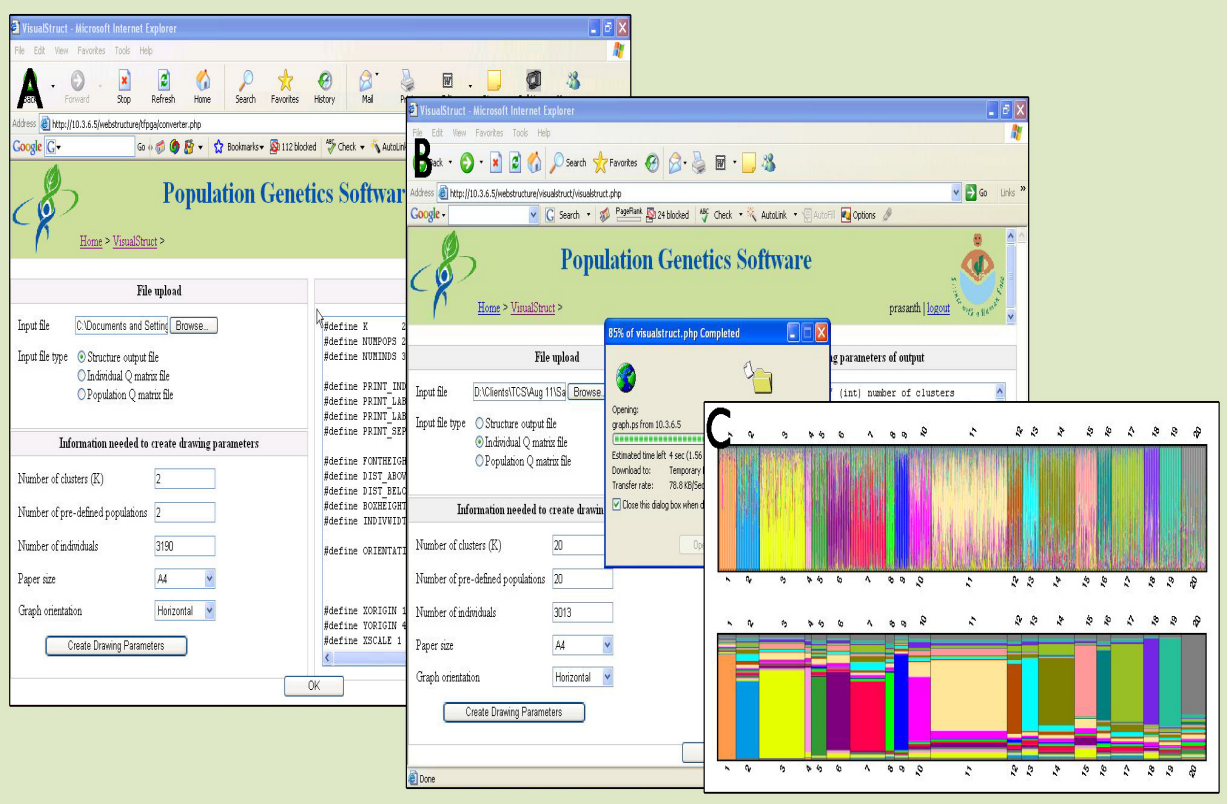

Figure 2: VisualStruct interfaces (A&B) providing input file information and setting drawing parameters, (C) a typical output image.

Distruct is the companion software available with *Structure* that helps display *Structure* results [6]. A convenient way to plot the results is to show each individual as a line. This line is partitioned into K colored segments, which represents the individual's estimated membership coefficients in the K clusters. To allow for ongoing changes in the *structure* code and for added flexibility, the *Structure* output file itself is not used as an input for the Distruct program. Instead, the program takes a file with the population Q-matrix (required), and a separate file with individual Q-matrix (optional), both printed in the format of the *Structure* output. This makes the input file preparation process tedious. Further, if the individuals do not have prior population information, then the preparation of these input files becomes even more difficult. The VisualStruct program available in this package automates the file conversion process to generate suitable input files from structure output for visualization; the resulting images are similar to the Distruct software (Figure 2). The improvement in this program relative to Distruct is that it also handles situations where prior population information is not available. This is achieved by assigning the population membership to each individual based on the maximum probability that it was derived from that particular population. Further, Visualstruct does not have any limitation on the number of individuals displayed, unlike Distruct. The parallelization approach used with the Structure software here has been quite simple, and providing access through user interfaces allows anyone skilled in using interfaces execute and manage results; without knowledge of the cluster environment he is interfacing with. The software can be part of a workflow in a distributed computing repository.

In conclusion, we have developed a software package that can implement the population genetics analysis software *Structure* within a compute cluster environment. Speedup is achieved by running different replicates and values of K on each node of the cluster. User interfaces have been written to the software to facilitate use by those who do not know how

the cluster functions. The package includes format conversion software as well as visualization software that are an enhancement over available programs. We believe that this tool would be of much interest to the plant and animal genetics and breeding communities as well as other users of *Structure* software.

Acknowledgement: The authors gratefully acknowledge financial support through the Generation Challenge Program for the high performance computing facility and software tools development. Thanks are also due to Dr. Jonathan Pritchard of the University of Chicago and author of the Structure algorithm for constructive comments.

References

[1] J.K. Pritchard, M. Stephens and P. Donelly. Inference of population structure using multilocus genotype data. Genetics, 155: 945-959, 2000.

[2] N.A. Rosenberg, J.K. Pritchard, J.L. Weber, H.M. Cann, K.K. Kidd, L.A. Zhivotovsky and M.W. Feldman. Genetic structure of human populations. Science, 298: 2381-2385, 2002.

[3] D. Falush, T. Wirth, B. Linz, J.K. Pritchard, M. Stephens, M. Kidd, M.J. Blaser, D.Y. Graham, S. Vacher, G.I. Perez-Perez, Y. Yamaoka, F. Mégraud, K. Otto, U. Reichard, E. Katzowitsch, X. Wang, M. Achtman and S. Suerbaum. Traces of human migrations in *Helicobacter pylori* populations. Science, 299: 1582-1585, 2003.

[4] R. Lecis, M. Pierpaoli, Z.S. Biro, L. Szemethy, B. Ragni, F. Vercillo and E. Randi. Bayesian analyses of admixture in wild and domestic cats using linked microsatellite loci. Molecular Ecology, 15: 119-131, 2006.

[5] D. Falush, M. Stephens, J.K. Pritchard. Inference of population structure using multilocus genotype data: dominant markers and null alleles. Molecular Ecology Notes, 7:574-578, 2007.

[6] N.A. Rosenberg. DISTRUCT: a program for the graphical display of population structure. Molecular Ecology Notes. 4: 137-138, 2004.