SNAP: workbench management tool for evolutionary population genetic analysis

Eric W. Price1,2 and Ignazio Carbone1,*

1Center for Integrated Fungal Research, Department of Plant Pathology and 2Department of Computer Sciences, North Carolina State University, Raleigh, NC 27695, USA.

Received on June 18, 2004; revised on August 5, 2004; accepted on August 24, 2004

Advance Access publication September 7, 2004

ABSTRACT

Summary: The reconstruction of population processes from DNA sequence variation requires the coordinated implementation of several coalescent-based methods, each bound by specific assumptions and limitations. In practice, the application of these coalescent-based methods for parameter estimation is difficult because they make strict assumptions that must be verified a priori and their parameter-rich nature makes the estimation of all model parameters very complex and computationally intensive. A further complication is their distribution as console applications that require the user to navigate through console menus or specify complex command-line arguments. To facilitate the implementation of these coalescent-based tools we developed SNAP Workbench, a Java program that manages and coordinates a series of programs. The workbench enhances population parameter estimation by ensuring that the assumptions and program limitations of each method are met and by providing a step-by-step methodology for examining population processes that integrates both summary-statistic methods and coalescent-based population genetic models.

Availability: SNAP Workbench is freely available at http://snap.cifr.ncsu.edu. The workbench and tools can be downloaded for Mac, Windows and Unix operating systems. Each package includes installation instructions, program documentation and a sample dataset.

Contact: ignazio_carbone@ncsu.edu

Supplementary information: A description of system requirements and installation instructions can be found at http://snap.cifr.ncsu.edu

INTRODUCTION

In recent years, rapid advances in DNA sequencing technology and population genetic theory have resulted in a plethora of new approaches for making inferences on population processes from DNA sequence variation. Among these are the tools for estimating population mutation and migration rates such as MIGRATE, (Beerli and Felsenstein, 1999, 2001), MDIV (Nielsen and Wakeley, 2001) and Genetree (Bahlo and Griffiths, 2000); recombination rates such as Recom58 (Griffiths and Marjoram, 1996), Infs and Fins (Fearnhead and Donnelly, 2001) and Recombine (Kuhner et al., 2000); migration and recombination such as LAMARC (Beerli and Felsenstein, 2001; Kuhner et al., 2000); and selection (Neuhauser and Krone, 1997). These approaches were built on Wright–Fisher population genetic models by incorporating probability-based coalescent methods, which take full advantage of the data and their inherent stochastic properties (Kingman, 1982a,b,c).

In practice, there are three main limitations for using these coalescent-based methods: (1) they make strict assumptions that must be verified a priori; (2) their parameter-rich nature makes the estimation of all model parameters very complex and computationally intensive; and (3) they are distributed as console applications written in C and require the user to navigate through console menus or specify complex command-line arguments. Although many tools and techniques are being developed for analyzing population-based DNA sequence variation, very few provide step-by-step methodologies for integrating multiple analysis methods into a readily accessible, user-friendly package.

The development of an integrated software environment would eliminate incompatibilities due to the strict data format requirements of different programs and allow data input and output to flow seamlessly between different analysis modules. This approach has been the goal of several emerging program suites, such as Mesquite (Maddison and Maddison, 2002, http://mesquiteproject.org) and EMBoss (Rice et al., 2000), which present the user with an arsenal of molecular tools and analysis methods. Two major limitations with these software programs are that they provide little or no guidance on how to perform a specific analysis and they require source code modification of existing C program modules before their inclusion.

*To whom correspondence should be addressed.
Fig. 1. Flowchart illustrating the sequence of steps and tools implemented in SNAP Workbench. The names of software tools are italicized; Hudson refers to the programs Seqtomatrix and Permtest developed by R. Hudson. The programs pars and consense are from the PHYLIP package (Felsenstein, 2004).

SYSTEMS AND METHODS

We have developed a workbench program that can manage and coordinate a suite of nucleotide analysis programs (SNAP). These include the alignment program ClustalW (Thompson et al., 1994); the phylogenetic analysis program PHYLIP (Felsenstein, 2004) and PAUP* 4.0 (Swofford, 1998); the non-parametric permutation analysis programs Seqtomatrix and Permtest (Hudson et al., 1992); the recombination detection programs RecMin (Myers and Griffiths, 2003) and RecPars (Hein, 1990); the coalescent-based programs Genetree, Recom58, MDIV and MIGRATE; and additional programs [SNAP Combine, Map, Clade and Matrix; Fig. 1 and Carbone et al. (2004)]. The workbench was designed to facilitate the incorporation of new tools as they become available, thereby serving as a bridge between theoretical and applied population genetic analysis. Although our workbench was designed to target population genetics, its solution to the problem of workflow integration is flexible and powerful enough to provide solutions in other fields as well. Our goal in developing the workbench was to:

1. eliminate the requirement for using command line,
2. integrate a wide array of approaches for analyzing population genetic data based on both traditional summary-statistic methods and the newer coalescent-based population genetic models,
3. ensure that the assumptions and program requirements of each method are not violated and
4. provide user interactive tutorials for teaching and training.

FRAMEWORK

The workbench was programmed in Java to preserve platform independence across multiple operating systems. The program modules integrated in the workbench are written
in C or Java and can be readily compiled on a variety of computing platforms. SNAP Workbench allows the user to customize the interface for available program modules without requiring computer programming or shell scripting skills. This is accomplished using the template design feature of the workbench. Templates allow the user to create and organize drop-down menus in the interface. Each menu option is further divided into submenus. Submenus define a set of programs or options within a single program that are executed sequentially to complete a particular analysis. For example, the submenu option for performing a 'Nonparametric test for population subdivision' under the 'Migration' menu requires the user to execute the programs SNAP Map, Seqtomatrix and Permutest sequentially. Because there is usually more than one way to perform these analyses, the workbench supports, via multithreading, simultaneous program executions to allow the user to explore different scenarios. Multiple files are displayed in a tabbed format for easy access and line wrapping has been disabled to facilitate viewing of long DNA sequences. Files may be edited and saved directly in the workbench using basic text-editing functions.

IMPLEMENTATION

SNAP Workbench has recently been used to examine recombination and migration in Cryphonectria hypovirus 1 (CHV-1) (Carbone et al., 2004). The multithreaded capability of our workbench was particularly important in this study because many independent coalescent runs were necessary to ensure convergence of the programs MIGRATE, MDIV, Recom58 and Genetree. A flowchart showing all the programs and analysis paths for inferring migration and recombination processes in CHV-1 is shown in Figure 1. The template design feature of the workbench allowed us to create menus consisting of a defined set of programs, assumptions and parameter settings for following a particular path in the flowchart (e.g. see paths 1 and 2 in Fig. 1). Currently, the workbench is designed to operate on a single machine. Future versions of SNAP Workbench will be able to use distributed parallel processing on Linux clusters and supercomputers for performing computationally intensive simulations and will integrate tutorials, providing comprehensive hands-on training, for the different analysis methods in the workbench.

ACKNOWLEDGEMENTS

The authors thank Doug Brown, Judy Jakobek and two anonymous reviewers for providing valuable comments.

REFERENCES


