

Compiling and Running IMA2p

By Arun Sethuraman and Jody Hey

Center for Computational Genetics and Genomics, Temple University

1. Introduction

This document details how to compile and execute IMA2p, a parallel version of IMA2, implemented under an OpenMPI-C++ framework.

IMA2p uses the same command set as IMA2, and users should refer to the documentation on using IMA2 and IM/IMa for further details.

Details of IMA2p's implementation can be found in Sethuraman and Hey (2015).

Citation:

Sethuraman, A, J Hey. 2015. IMA2p - Parallel MCMC and inference of ancient demography under the Isolation with Migration (IM) model. Molecular Ecology Resources DOI: 10.1111/1755-0998.12437

2. Compiling using autoconf

IMA2p has been written using the OpenMPI-C++ framework, and can be compiled using standard MPI flavors of the GNU compiler (including mpicc and mpicxx). OpenMPI must be installed. For details on installing OpenMPI, please see this [page](#). Also, once installed, please make sure that the “bin” directory of OpenMPI (which contains all the compilers, and executables) is added to your machine's path.

If you have already installed OpenMPI, to compile using *autoconf* (assuming you have some stable version of autoconf on your Unix machine – else see [here](#)), download the package, *unzip/untar* it, then *cd* into the folder. Once inside make sure that you have permissions to install using:

```
chmod +x *
```

Then, type:

```
./configure --with-mpi=yes --prefix=/path/to/install
```

This should create the necessary ‘make’ files to install the package.

If you aren't sure, the configuration script will determine if your machine contains OpenMPI definitions or not, and create ‘make’ files accordingly. To do this:

```
./configure --with-mpi=auto --prefix=/path/to/install
```

Alternately, IMA2p can also be compiled using a generic GNU C++ compiler for serial use (similar to the original IMA2 package). To do this, type:

```
./configure --with-mpi=no --prefix=/path/to/install
```

Note however, that if your machine does not have OpenMPI installed, or if the configuration script is unable to find it, it will compile a serial version of the program (similar to IMA2) with a regular GNU C++ compiler.

Once the ‘make’ files have been generated successfully, you should be able to install IMA2p by typing:

```
make
```

This should create an executable called “IMA2p” inside the *src* folder. Alternately, you can create a separate executable folder by typing:

```
make install
```

This should create a *bin* folder inside your installation path, which will contain the executable (“IMA2p”).

3. Compilation from source

If you wish to compile from source, then you need to either delete or comment out line #24 from *src/imamp.hpp* (which says `#include <config.h>`).

Thereon, to compile from source, *cd* into the *src* folder, and type:

```
mpicxx -DMPI_ENABLED -o IMA2p *.cpp
```

This will create an executable called *IMA2p* in the *src* folder. To run the program in serial mode (single processor), run the executable directly with an appropriate command line (for details see the [manual for IMA2](#)).

To run it in parallel use *mpirun* or *mpiexec*, for example:

```
mpirun -np <number of processors to use> IMA2p ... <IMA2 command line options>
```

An IMA2p run must invoke at least 1 MCMC chain per run. In IMA2 the *-hn* flag is used to indicated the total number of chains. In IMA2p the *-hn* flag is used to indicate the number of chains per processor. For example, to invoke Metropolis coupling using a total of 10 chains, distributed among 5 processors:

```
mpirun -np 5 IMA2p -hn 2 <other IMA2 command line options>
```

4. To install on a multi-core Windows machine (PC/Server), we will need to do this through a Linux emulator like Cygwin (<http://cygwin.com/install.html>). While installing/updating Cygwin, ensure that you select to install the GNU C/C++/Fortran libraries, OpenMPI libraries, autoconf (Devel), and git.

Once installed, open a Cygwin window, and use:

```
git clone https://github.com/arunsethuraman/ima2p
```

This will create a folder called *ima2p* on your machine. Then cd into it.

Change permissions using:

```
chmod +x *
```

Now you should be able to install using:

```
./configure --with-mpi=yes
```

Or try compiling from source as noted above.

You should then be able to execute using:

```
mpiexec -np <number of processors> ./IMa2p
```

5. Information in the Output files: Differences from IMa2:
 - a) Main output file: In addition to information on swapping rates between chains, the output file summarizes the swapping rate between processors.
 - b) *.ti file – if you have set genealogies to be saved into separate *.ti files, these will be created for each processor, but the sampled genealogies will be saved only on the head node. This file will be named *.ti
 - c) *.mcf.<processor number> files – if you have set the MCMC states to be saved on each chain, this information is saved in files with the *.mcf extension. Each chain on each processor saves a different state file. All these files are important, if and when you wish to restart your ‘M’ or ‘L’ mode run in parallel.
 - d) *.burntrend.out.<processor number> file – burn-in trend files are created at the end of burn-in runs. These contain update rates for genealogies, which are unique to each processor (since there are coupled chains running on each processor). However, the burn-trend is collated onto the head node, and can be seen in the *.burntrend.out.0 file.
6. Please report any issues with compilation/running IMa2p to Arun Sethuraman (arun@temple.edu).