primeGSR

Örjan Åkerborg, Bengt Sennblad, Lars Arvestad, Jens Lagergren

October 22, 2008

1 Introduction

This is a provisional manual for our program `primeGSR`. The current version of `primeGSR` is a prototype implementation accompanying our manuscript [3]. Our aim is to provide a properly versioned release of the program, including more efficient code and binaries for multiple platforms, in the near future.

`primeGSR` is a program for reconstructing gene trees from sequence data given a known species tree. It integrates probabilistic models for gene duplication and loss, relaxed molecular clock for substitution rate variation over the tree, and sequence evolution. Thanks to new algorithms, that uses a discretization of divergence times, it can reconstruct gene trees in time comparable to standard substitution model-based reconstruction methods. A detailed description of the models and algorithms is given in [3]. While we are intending to design integrated methods for species tree reconstruction, primeGSR is aimed at gene tree reconstruction alone. It is a comparative genomics tool for gene family analysis in a multiple species context.

2 Installation

While we provide the specific binary executable that was used in [3], the main release is the source code, which must be compiled on the computer before it can be used. It is designed for use in a Unix- or Linux-like system; for MacOSX, the included Unix-base (Darwin and X11) will work, for MS Windows various emulators exists, e.g., Cygwin (`http://www.cygwin.com`). `primeGSR` uses some functions from the LAPACK package (`http://www.netlib.org/lapack`, which need to be installed. Various Unix system-specific LAPACK packages exists as rpms. For MacOSX and MS Windows packages can be downloaded, e.g., using MacPorts (`http://www.macports.org`) or Cygwin, respectively. For LAPACK installation instructions, please refer to the documentation for the respective packages. See further the `README` file included in the code package.

For installation of primeGSR, we here assume that you work with a terminal based compiler:

1. Download the tarred and gzipped file containing the source code (primeGSR_XX.tgz, where XX is some version number) from `http://prime.sbc.su.se/primeGSR` to your chosen directory on your computer.

2. Extract the tar archive:
   `tar -xvzf primeGSR_XX.tgz`

3. You may need to do some changes in the `Makefile`. A few system-specific commands are given in the `Makefile`, but for other systems additional changes may be needed. For example, if the LAPACK library is installed in a non-standard location on the computer, you need to tell in the `Makefile` where they are installed.

4. Enter the `primeGSR` directory and compile the program by typing:
   ```
   cd PrimeGSR_XX
   make
   ```

5. If the compilation worked, you may wish to move or copy the file `primeGSR` (or sometimes `primeGSR.exe`) to a convenient location (e.g., `$HOME/bin`, i.e., a directory called `bin` in your home directory) and also ascertain that this location is in your path (check this by doing `echo $PATH`). How you change your path may depend on the shell used, e.g., in Bash you add `export PATH=$PATH:$HOME/bin` to your `.bashrc` file in your home directory.

3 Usage

The program takes as input:

1. A data file containing aligned sequences in Fasta format. The set of names associated with the sequences will be the labels of the leaves in the gene tree.

2. A tree file with a species tree with divergence times in Newick- or PRIME format (actually optional, but often you want to do this). In Newick format *ultrametric edge times* are given, prefixed by a colon ':', after each node in the species tree. In PRIME format, edge times can be given using 'tags' of the format `[&&PRIME ET=0.1]`, after each node in the tree. Alternatively, divergence times can be given directly using the tag `NT`. Thus, the following three strings describe the same species tree and divergence times,
   ```
   ((A:0.1,B:0.1):0.2,C:0.3)
   ((A[&&PRIME ET=0.1],B[&&PRIME ET=0.1])[&&PRIME ET=0.2],C[&&PRIME ET=0.3])
   ((A[&&PRIME NT=0],B[&&PRIME NT=0])[&&PRIME NT=0.1],C[&&PRIME ET=0])[&&PRIME NT=0.3]
   ```
   are equivalent. The PRIME format is rather versatile, and a fuller description is given in the `primetv` manual (see `prime.sbc.su.se/primetv`).

3. A gene-species map-file with a two column table (delimited by spaces) associating sequence (or equivalently gene tree leaf) names to species tree leaf names. A simple example, the gene-species map:
   ```
   a1  A
   a2  A
   b   B
   c   C
   ```
   associates the two genes `a1` and `a2` to the species `A`, gene `b` to species `B` and gene `c` to species `C`.

`primeGSR` is a Unix command line program, so you need to run it from a terminal window on your computer. The general syntax for starting an analysis is,

```
primeGSR [<options>] <datafile> [<gs>]
```

where the parameters are as follows,

- <datafile> is the name of the file containing the sequence data in aligned Fasta format.
- <gs> is a gene-species map file, as described above.

The options allows the user to customize certain program settings, but can be left out altogether when default settings are used. The options availabel are listed in Table 1, and are divided into several subgroups.

- General options control the settings of the MCMC-chain itself, e.g., the number of iterations to be run (`-i`), at what interval samples should be saved to the outfile (`-t`) and how often we should write progress to the terminal (`-w`). The `-q` option reduces the output to the terminal. Other options are mainly for debugging, e.g., the `-s` option allows exact repetition of an analysis by making the random number generator behave in the same way.
- Substitution model options. A few standard substitution models are implemented, and there is also a crude way allowing the user to define an own substitution model, giving, all in on line, the data type, the stationary state frequencies (`Pi`), and the instantaneous transition matrix (`R`, given as a upper triangular matrix, written line by line as a vector). This makes it possible to estimate the substitution matrix *a priori* using e.g., ModelTest or PAUP. The `-Sn` and `-Sa` options allows rate variation across sites to be modeled using a discrete Gamma distribution [2].
- Edge rate model options.It is only possible to use the `iid` model with `primeGSR`. This models edge rates as independent and identically distributed random variables according to a choice of four different underlying distribution can be used (`-Ed`) Gamma, LogNormal, Inverse Gaussian and Uniform. The mean and the variance of the distribution can be fixed to user-defined values (`-Ef`), but are otherwise estimated during analysis.
- Species tree options. By default a single edged species tree is used, but most often a user defined tree should be given (`-Hi`)
- Gene tree options. A user-defined starting tree can be given (`-Gi`); this can also be fixed during analysis (`-Gg`).
- Lastly, start values (`-Bp`) or fixed values (`-Bf`) for the duplication-loss model (an extended birth-death model) can be set.

4 Output

The output of primeGSR can be said to comprise four parts.

1. First the string used to start the program are listed. The random number generator seed used for the analysis is also given. This is useful for later reference or for debugging.
2. Second, a summary of the models and settings used is given in running text, indented hierarchically.

Table 1: List of options for `primeGSR`

Option	Required value	Explanation
	General options	
-u, -h		A help text.
-o	<filename>	output file
-i	<int>	number of iterations
-t	<int>	thinning
-w	<int>	Output to cerr <int> times less often than to cout
-s	<int>	Seed for pseudo-random number generator.
-q		Do not output diagnostics to stderr.
-d		Debug info.
-l		Output likelihood. No MCMC.
	-S<option> Options related to Substitution model	
-Sm	'JC69'/'UniformAA'/ 'JTT'/'UniformCodon'	The substitution model to use, (JC69 is the default). Must fit data type
-Su	'DNA'/'AminoAcid'/'Codon' Pi=float1 float2 ... floatn R=float1 float2 ...float(n*(n-1)/2)	The user-defined substitution model to use.The size of pi and R must fit data type (DNA: n=4, AminoAcid: n=20, Codon: n = 62), respectively. If both -Su and -Sm is given (don't do this!), only the last given is used
-Sn	<float>	nCats, the number of discrete rate categories (default 1)
-Sa	<float>	alpha, the shape parameter of the Gamma distribution for site rates (default: 1). Only meaningful when the number of rate categories > 1 (cf. -Sn)
	-E<option> Options relating to edge rate model	
-Ed	'Gamma'/'InvG'/ 'LogN'/'Uniform'	Density function to use for edge rates, default is Uniform
-Ef	<float> <float>	Fixed mean and variance of edge rate model
	-H<option> Options relating to the host tree	
-Hi	<treefile>	Use tree in file <treefile> as host tree
	-G<option> Options relating to the tree	
-Gg		Fix tree. No branchswapping performed
-Gi	<treefile>	Use tree in file <treefile> as start tree
	-B<option> Options relating to the birth death process	
-Bp	<float> <float>	Starting values BD parameters.
-Bf	<float> <float>	Fix BD parameters to these values.

3. The third part is a number of columns giving the MCMC output. First a header lists all parameter names and types associated to each column, then the values for all sampled iterations are given row by row.

4. Last some summary statistics are given such as acceptance ratio, best state, etc.

The output file can be fed directly to an analysis program (written in Perl), `mcmc_analysis`, which will summarize the results in text format if run with no options. The options are,

- `-t` present output in LaTeX format.
- `-p <parameter>` outputs the column corresponding to `<parameter>` in a form useful for plotting, e.g., with the plotutils program `graph` (`http://www.gnu.org/software/plotutils`). For example
  `mcmc_analysis -p L output.mcmc |graph -TX -`
  brings up an Xwindows plot of the parameter `L`. To find out parameter names, do `mcmc_analysis -p help output.mcmc`.
- `-coda` outputs the mcmc-output in a format that can be read by the R-package Coda [1].
- `-P` If several chains are given as input to `mcmc_analysis`, these will be treated as consecutive output from a long MCMC-chain. To treat them as independent chains use the `-P` option (affects how burnin is treated).

5 Feedback

We appreciate any feedback on the program that can help us improve the software. If you wish to report a bug, please send an email to `prime-bugs@csc.kth.se`. Try to describe the bug as detailed as possible and if you can, also send input and output files, this will simplify bug-fixing enormously.

References

[1] Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 1(1):7–11, March 2006.

[2] Z. Yang. Maximum likelihood phylogenetic estimation from dna sequences with variable rates over sites: Approximate methods. *J Mol Evol*, 39(3):306–314, 1994.

[3] Ö. Åkerborg, B. Sennblad, L. Arvestad, and J. Lagergren. Gene tree reconstruction come of age: a bayesian integrated model for genes, sequence, and rates. submitted.