HIVtree Manual

Anna Nagel and Bruce Rannala

November 11, 2022

HIVtree is a Bayesian phylogenetic inference program that estimates HIV latent integration times and node ages on a fixed phylogenetic tree. This program was originally modified from PAML version 4.9. The program requires latent sequences and serially sampled sequence data from non-latent sequences with known sample dates for all sequences.

Quick Start

HIVtree

This section provides step by step instructions to run HIVtree with example datasets. To start, clone the program from github by typing the following into the command line:

```
git clone https://github.com/nage0178/HIVtree.git
```

Navigate into the cloned directory.

```
cd HIVtree
```

HIVtree must be compiled. This required a C compiler and the make system be installed on the computer. To compile, type

```
make
```

This compiles HIVtree into an executable. Now navigate to the examples directory,

```
cd examples
```

To run the program on the example dataset, type

```
./../HIVtree p1.ctl
```

The program will take a few minutes to run. The progress on the MCMC will be printed to screen. When the MCMC finishes, summary statistics will be printed to screen.

Combining inferences across the genome (using parseMCMC.sh and combineEstimates.R)

To combine inferences across multiple genes or genomic regions, the user must run the program on a unix computer using the bash shell and R must first be installed. The R packages "GoFKernel" and "kdensity" must also be installed. Once this can been accomplished, navigate to the next example.

```
cd combineGenes
```

Run HIVtree on the example datasets using the script provided.

```
./runExampleMCMC.sh
```

This runs HIVtree for two genes both under the prior and with data. Running HIVtree works in the same way as in the previous example. This will take a few minutes to run. Next, run the following command.

```
./runExample.sh
```

This first parses the output of the MCMCs with a script parseMCMC.sh and creates files to be used in the R script combineEstimates.R. Then it estimates a mean and 95% credible interval for the latent integration time of the example latent provirus, using two genomic regions.

Inputs and Output

HIVtree

Inputs	Description
Control file	This provides all of the priors and specifications of the MCMC. See the control file section for more detail.
Tree file	This contains a rooted tree in newick format without branch lengths. This is specified with the keyword treefile in the control file.
Sequence data file	This file provides the DNA sequence data in PHYLIP format. This is provided in the control file with the keyword seqfile. The relative sample dates for the sequences must be at the end of each sequence name.
Latent sequence file	This provides the names of the sequences which are latent. Each sequence name should be on its own line. This is provided in the control file with the keyword latentFile.

Outputs	Description
MCMC file	This contains the results of the MCMC. It can be viewed in programs such as tracer. This is specifed with the keyword mcmcfile in the control file.
Tree file	This contains a rooted tree in newick format with branch lengths. It can be viewed with programs such as FigTree. This is printed to FigTree.tre
Summary file	This has summary information about the data and MCMC. This is specifed by the keyword outfile in the control file
Summary printed to screen	Summary information about the input data, the priors, and the progress of the MCMC is printed to screen. Summaries of the results are printed once the MCMC has finished.

parseMCMC.sh

Inputs	Description
MCMC files	Output MCMC files from HIVtree for each gene run under the prior and with data.
HIVtree summaries printed to screen	For each gene, the information printed to the screen during HIVtree should be redirected to a file named output.
Mapping file	This is csv file which specifies which sequences from different genes came from the same provirus.

Outputs	Description
MCMC summary file(s)	The file contains the MCMC samples of the latent integration times for all the genes from a singular latent provirus. There will be multiple files if the mapping file specified sets of sequences from multiple proviruses.

combineEstimates.R

Inputs	Description
MCMC summary file	The file generated by parseMCMC.sh for a single provirus

Outputs	Description
Summary to printed to screen	The mean and the 95% highest posterior density interval

Command line argument number and description	Explanation
1: MCMC summary file	file created by parseMCMC.sh
2: sample time of the latent sequence	This must be in the same time units as HIVtree
3: latentBound	This is the sample as the latentBound arguement used in HIVtree
4: time unit	This should match the second argument of TipDate used in the control file for HIVtree. If "tipDate = 1 1000", this argument should be 1000.
5: last sample date	Last sample date in the whole phylogeny. This assumes that the last sample date is the same for all of the phylogenies. This will be the same as the highest number at the end of the sequence names in the fasta file.
6: number of genomic regions	This should be half the number of columns in the MCMC summary file and is used for error checking. This cannot be greater than 10. Without any missing data, this will be equal to the number of genomic regions. If there is missing data for some genes for a provirus, the number of genomic regions will be greater than the number genes used for analysis. The analysis will still work correctly with missing data.

Control File for HIVtree

The control file is very similar to that of mcmctree, a program in PAML. Here, only new or changed options to the control file will be detailed. We refer readers to the mcmctree manual for a full description.

clock: 1 must be used for the clock model, which specifies a strict clock. Other clock models are available in PAML but not in HIVtree.

latentFile: This is a the name of text file that provides the names of all of the latent sequences. The sequence names should match the names in the alignment and tree file with one name per line.

latentBound: This provides a hard upper bound on all of the latent integration times in the analysis. This is specified in backward time in the time units specified by the TipDate option. For example, consider the options "latentBound = 3", "tipDate = 1 1000", and time specified in days in the sequence names. This means no latent ages can be more than 3000 days older than the time of the last sample.

RootAge: This specifies the prior on the root age. There are two options, either a shifted gamma prior, $G(\alpha, \beta)$, or a uniform prior, $\mathrm{U}(a, b)$. The gamma distribution is shifted by adding the first sample time to the distribution. This ensures there is no density after sequences are sampled. The uniform prior has hard bound, so there is no density outside of the range between a and b. The parameters for both the uniform and gamma distributions must also be chosen with the time unit transformation going backward in time. For example, with option "tipDate = 1 1000" and the dates for the sequences specified in days, U(3,4) would be a uniform root age prior between 3000 and 4000 days prior to the *last* sample time. G(1, 1) would be a gamma prior with mean 1000 days prior to the *first* sample time with variance 1000 days. Note that the user specified prior will not match the induced prior when running without data (option usedata = 0) because of the constraints imposed by the tip ages and rank order of the node. The user should run without data to see what the induced prior will be.

Example 1 in detail

Preparing the dataset and running HIVtree

This describes in detail the analysis performed in the first example. It follows the workflow in Figure 1. There are two latent sequences, "MG822922.1_4057" and "MG823170.1_7262" for a single gene. First, the user must add the relative sample dates to the end of all of the sequence names. For example, "MG823170.1_7262" was sampled 3205 days after "MG823170.1_7262". The time unit (e.g. days, weeks) does not matter as long it is consistent and the priors and specified one the same scale. Next, the user should infer a rooted phylogeny using a method of their choice and remove the branch lengths. After creating a control file and preparing the other input files in the appropriate format, HIVtree is run.

```
./../HIVtree p1.ctl
```

Understanding the output

At the end of the output to screen, there is a summary of the posterior distribution of node ages and latent integration times displayed both in the transformed time units and calendar time (the same time scale as the time as the end of the sequence names). The nodes of the tree are assigned numbers, which are printed near the top of the output to screen. These match the number in the summary output. For example, a node 12 has a time t_n12.

The MCMC file and the FigTree file both report the results the transformed time units. This is scaled by the option "tipDate" and increasing into the past. For example, with option "tipDate = 1 1000" and sample times specified in days, a node age or latent integration date of 1 means the node is 1000 days older than the last sample time.

Example 2 in detail

This describes in detail the analysis performed in the second example. It follows the workflow in Figure 2. There are sequences from two latent proviruses, W14 and W19. These were split into two regions, C1C2 and C2C3 (both part of ENV and sometimes labeled ENV2 and ENV3 in the example files). Additionally, there are non-latent sequences for both of these regions with known sample dates.

Step 1. Run HIVtree

The same file preparations must be made as in the first example for each gene. HIVtree should be run for each gene with and without data. To run without data, usedata=0 should be included in the control file. To run with data, usedata=1. The control file should contain the line"mcmcfile = mcmc.txt", which gives sets the name of the output file of the MCMC. Note that HIVtree will overwrite existing files, so each analysis should be run in a separate directory. The output should be redirected to a file called output, as is shown below and in the runExampleMCMC.sh script used in the quick start.

```
./HIVtree control.ctl &> output &
```

Step 2. Parse the output of HIVtree

Mapping file

To combine the results from different regions of the genome from a single latent provirus, the user must create a mapping file that specifies which latent sequences are from the same latent provirus.

In the example dataset, the latent sequences are named "C1C2_W14_QVOA_3921", "C1C2_W19_QVOA_3921", "C2C3_W14_QVOA_3921", and "C2C3_W19_QVOA_3921" in the fasta files. For the analysis under the prior, "prior" was added to the latent sequence names. This is optional. This was done in the example so that sequence names are unique in the csv file for illustration purposes. The output of the MCMCs are in directories named "ENV2", "ENV3", ""ENV2_Prior", and "ENV2_Prior". The mapping file is shown below.

```
ENV2,ENV3,ENV2_Prior,ENV3_Prior
C1C2_W14_QVOA_3921,C2C3_W14_QVOA_3921,C1C2_W14_QVOA_prior_3921,C2C3_W14_QVOA_prior_3921
C1C2_W19_QVOA_3921,C2C3_W19_QVOA_3921,C2C3_W19_QVOA_prior_3921,C2C3_W19_QVOA_prior_3921
```

The first row includes the directory names for all of the MCMC runs. The runs with the data should be first in the csv file and then the runs without data (priors). The genes should be in the same order for the priors and posteriors. The following rows give the name of the latent sequence in each of the MCMC runs. An improperly formatted csv file will result in the script not running correctly. If the script does not appear to give the correct output, check the to make sure the input file is in a csv file format with the same number of commas on each row and names that match the names in the fasta files. This program allows for missing data. The entry in the csv file should be blank if there is no sequence for a particular gene for a given latent provirus.

Running parseMCMC.sh

The mapping is given as an input to parseMCMC.sh, which is run as follows.

```
./../../parseMCMC.sh sequences.csv
```

This will generate the MCMC summary files needed to run the analysis in R in step 3. There will be one less file generated than there are rows in the sequences.csv file. Each file is a csv file where each column is the estimate of a single latent integration time from a MCMC. If there are n different genomic regions, the first n columns will be the posterior distributions for the latent time and the $n+1$ to $2n$ columns will be the prior distributions. This will be in the same order as the input csv file. The names of the output files will match the name of the first sequence in each row of the csv file. For example, the above csv would have files names "C1C2_W14_QVOA_3921.txt" and "C1C2_W19_QVOA_3921.txt". If there is missing data, the first non-missing sequence name will be used.

Step 3. Running combineEstimate.R

The will infer the posterior probability of the latent integration times for a single provirus. The prior distributions are divided out for all genes, resulting in a prior for the latency time that is uniform between the lower and upper integration bounds. There are 6 arguments needed by the R script, which are described in the inputs and outputs section. All 6 arguments are required. This analysis can be run on the example dataset as follows.

```
Rscript ../../combineEstimates.R C1C2_W14_QVOA_3921.txt 0 3.695 1000 3650 2
```

The RScript will print out the mean and 95% credible set of the posterior distribution for the latent integration time. If the user wishes the view the posterior distribution, the plot command in the Rscript can be uncommented. Note that in some rare cases, the numerical integration of the kernel density estimate can fail in R. One reason this may occur is if the posterior densities for the genes analyzed do not overlap.

Figure 1: Workflow for estimating the latent integration time for a single gene. The dark boxes show programs or scripts and the light boxes show inputs and outputs.

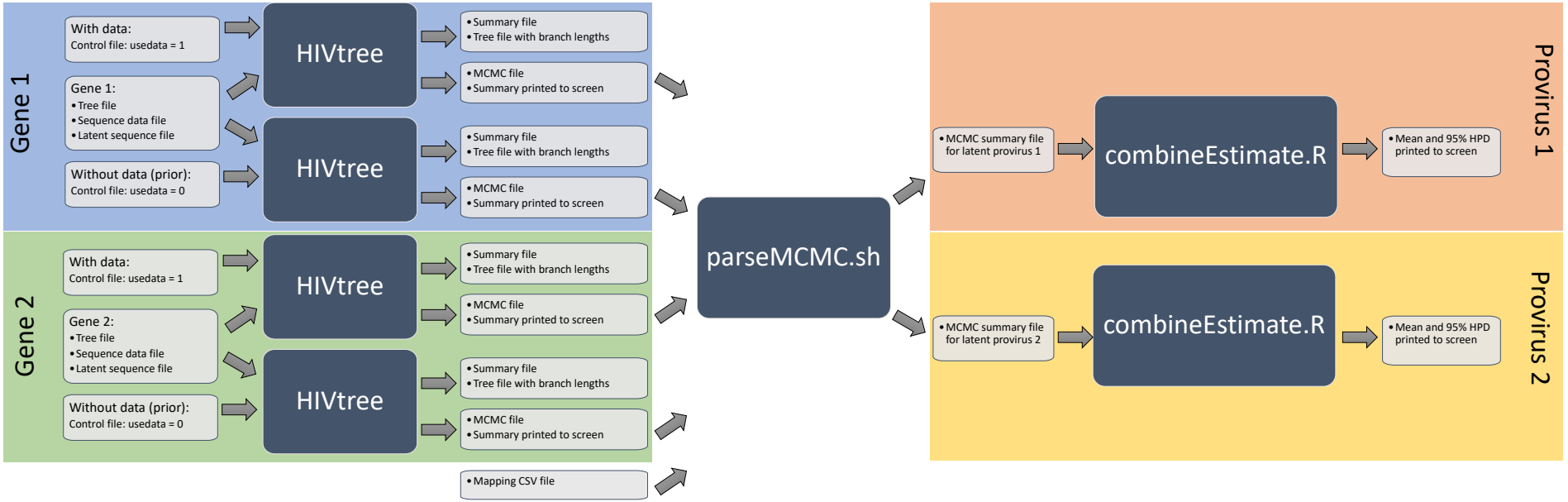

Figure 2: Workflow for estimating the latent integration time of two latent proviruses, each of which have sequences from two genes. With more genes, the blue block would be repeated for each additional gene. With more proviruses, the orange block would be repeated for each additional provirus. The dark boxes show programs or scripts and the light boxes show inputs and outputs.