HIVtree Manual

Anna Nagel and Bruce Rannala

January 10, 2023

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.

Compile HIVtree and install R packages

Navigate into the HIVtree 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. To install the R packages necessary for analyze multiple genes, type

```
Rscript installRpackages.R
```

This installs the R packages GoFKernel, getopt, and kdensity.

Quick Start

Single gene

This section provides step by step instructions to run HIVtree with example datasets. Navigate to the examples directory,

```
cd examples
```

To run the program on the example dataset, type

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

This will estimate two latent integration times on a phylogeny with 11 sequences. 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 on the latency times and node ages as well as the substitution model parameters. The section single gene in detail describes how to interpret the output and the files used in this analysis.

Multiple genes

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, kdensity, and getopt must also be installed. If you have not installed these packages, return to the section "Compile HIVtree and install R packages". Once this can been accomplished, navigate to the next example.

```
cd combineGenes
```

Run HIVtree on the example datasets using the script provided.

```
./runExample.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. Then, the script 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. The section multiple genes in detail describes the files used in this analysis and how to interpret the output.

Single gene example in detail

This describes in detail the analysis performed in the first example. It follows the workflow in Figure 1 to infer two latent times for a single gene.

Preparing the sequence data file

The sequence data should be aligned using the users method of choice. The user must add the relative sample dates to the end of all of the sequence names in the sequence data file. In this example, this is "p1_mcmc.fa". For example, "MG823170.1_7262" was sampled 3205 days after "MG822922.1_4057". The time unit (e.g. days, weeks) does not matter as long it is consistent and the priors and specified with the time scale in consideration. The sequence file should have the number of sequences and the sequence length on the first line, followed by the alignment. The beginning of this sequence file, p1_mcmc.fa, is shown.

```
11 621
>MG822918.1_3599
atgggtagcaagtgttcaaaaagtaaggcgggtggatggcctgctataagggaaagaatgcaacaagctgagccagcagcagaaggggtggga⌋
↪ gcagcatc
tcgagacctagaaagacatggagcgatcacaagtagcaatgtaaataatgctgcttgtacctggctagaagcacaagcacaagaggaggaaga⌋
↪ ggtgggtt
ttccagtcagacctcaggtacctttaaggccaatgacttacaagggagcttttgatcttagcttctttttaaaagaaaaggggggactggaag⌋
↪ ggctaatt
tggtctcagaaaagacaagacatccttgatttatgggtctacaacacacaaggctacttccctgattggcataactacacaccagggccaggg⌋
↪ gtcagata
tccactgaccttcggatggtgcttcaagctagtaccagtggatccagataaggtggaagaggccactgagggagaaaacaacagcttgctaca⌋
↪ ccctatgt
gccagcatggaatggatgacccagagaaagaagtgctagtgtggaagtttgacagccgcctcgcacaccatcacatggcaagagagaaacatc⌋
↪ cggagttt
tacaaggactgctga
>MG822922.1_4057
atgggtagcaagtgttcaaaaagtaaggtgggtggatggcctgctataagggaaagaatgcaacaagctgagacagcagcagaaggggtggga⌋
↪ gcagcatc
...
```

Preparing the latent sequence file

A file with the names of the latent sequences must be created. In this example, the latent sequence file is "p1_seqsL". There are two latent sequences, "MG822922.1_4057" and "MG823170.1_7262" for a single gene. These are listed in the latent sequence file, whose contents are shown below. Each sequence should be on its own line.

```
MG822922.1_4057
MG823170.1_7262
```

Inferring a rooted phylogeny

The user should infer a rooted phylogeny using a method of their choice and remove the branch lengths. The tree file has the number of tips in the tree followed by the number of trees in the file as the first line. The second line is the tree in newick format without branch lengths. This is the file p1_mcmc.txt in the example and the contents are shown below. The gray arrow indicate textwrapping without line breaks in the original file.

```
11 1
(((((((MG823170.1_7262,MG822924.1_3286),(MG823012.1_2728,(MG822918.1_3599,MG822922.1_4057))),
↪  MG822944.1_1677),MG822946.1_2077),(MG822943.1_1272,MG822947.1_1391)),MG822955.1_626),MG82
↪  2952.1_128);
```

Preparing the control file

The control files specifies the models, priors, and parameters of the MCMC. The contents of p1.ctl are shown below.

```
seed = 1

seqfile = p1_mcmc.fa
treefile = p1_mcmc.txt
mcmcfile = mcmc.txt
outfile = out.txt

seqtype = 0 * 0 is nucleotide data
usedata = 1 * usedata = 1 produces the posterior distribution, usedate = 0 produces the prior
↪  distribution

ndata = 1
clock = 1 * Strict clock, required for HIVtree
TipDate = 1 1000

RootAge = G(8,60) * Root age prior
model = 4 * Substitution model, 0:JC69, 1:K80, 2:F81, 3:F84, 4:HKY85
alpha = 1.5 * Gamma model of rate variation, turned off if 0, otherwise it will be on
ncatG = 5 * Number of rate categories in discrete gamma for the +Gamma model of rate variation

cleandata = 0  * remove sites with ambiguity data (1:yes, 0:no)?

alpha_gamma = 4 8 * Gamma prior on alpha parameter for +G model of rate variation
rgene_gamma = 2 200 * Gamma prior on the substitution rate, note this must be choosen in the
↪  time transformed time units
kappa_gamma = 8 1 * Gamma prior on kappa parameter in DNA substitution models with kappa

print = 1 * Determines amount of output of program
burnin = 5000 * Length of the burnin
sampfreq = 2 * Sample every other iteration
nsample = 70000 * Total number of samples after the burnin

latentFile = p1_seqsL * File containing the names of the latent sequences
latentBound = 10.911 * Oldest possible age of latent sequence in the transformed time units
↪  used in HIVtree
```

`seqfile` specifies the file containing the sequence data. In this example, the file is "p1_mcmc.fa". `treefile` specifies the file containing the rooted phylogeny, which is called "p1_mcmc.txt" in this example. `latentFile`

specifies the file which lists the latent sequences. In this example, this is "p1_seqsL" The `burnin`, `samplefreq`, and `nsamples` specify the number of iterations of the MCMC (see MCMCtreeDOC.pdf for more detail). The `RootAge` specifies the prior on the root age. `latentBound` specifies the oldest possible integration time of any of the latent sequences in the analysis. How to specify `RootAge` and `latentBound` are both described in more detail in the section "Control File for HIVtree". `rgene_gamma` is the prior on the substitution rate. `alpha_gamma` is the prior on the alpha parameter for models with gamma distribution among site rate variation. `kappa_gamma` is the prior on kappa for DNA substitution models with different transition and transversion rates. The choice of priors should reflect the user's prior belief in the parameter value. The user may run HIVtree multiple times to see how the priors impact the infered latency times. The mutation rate prior should reflect the timescale units. Since `tipDate = 1 1000` and the dates are in days, `rgene_gamma = 2 200` specifies a prior with mean 0.01 substitutions per thousand days or 10^{-5} substitutions per day. If the times in the example were changed to be specifed in years, the prior would need to be updated. If we want the mean and standard deviation to remain the same, we have the following set of equations.

$$\frac{\alpha_{old}}{\beta_{old}}\text{days} = \frac{\alpha_{new}}{\beta_{new}}\text{years} \times \frac{365 \text{ days}}{\text{year}} \tag{1}$$

$$\sqrt{\frac{\alpha_{old}}{\beta_{old}^2}\text{days}} = \sqrt{\frac{\alpha_{new}}{\beta_{new}^2}\text{years}} \times \frac{365 \text{ days}}{\text{year}} \tag{2}$$

Solving these equations, we get $\alpha_{old} = \alpha_{new}$ and $\beta_{old} = \beta_{new} \times 365$.

After creating a control file and preparing the other input files in the appropriate format, HIVtree is run.

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

Understanding the output

The nodes of the tree are assigned numbers, which are printed near the top of the output to screen. This section of the output is shown below. For example, the latent sequence "MG823170.1_7262" is node 1. The times printed to the right are the starting times for the MCMC. For the nodes that are not latent, the time displayed is the node age. For the latent nodes, the times are the initial latent integration time. Node 12 is the root and has the root age prior listed in the fossil column. The times displayed are in the transformed time units used by HIVtree. In these units, the last sample is taken at time zero, and larger values are farther into the past. The time is scaled by the tipDate units. Since `tipDate = 1 1000` is specified in the control file and the last sample was taken at time 7262, the time 1.20988 for node 1 means that the integration time is 6052.12 days (7262 - 1.20988 $\times$ 1000). For node 2, $7262 - 3.976 \times 1000 = 3286$, which is the sample time. Since node 2 is not latent, the node age rather than the integration time is displayed.

```
Species tree
ns = 11  nnode = 21
father   node  name                 time     sons          fossil
18       1  MG823170.1_7262        1.20988
18       2  MG822924.1_3286        3.97600
19       3  MG823012.1_2728        4.53400
20       4  MG822918.1_3599        3.66300
20       5  MG822922.1_4057        3.32058
16       6  MG822944.1_1677        5.58500
15       7  MG822946.1_2077        5.18500
21       8  MG822943.1_1272        5.99000
21       9  MG822947.1_1391        5.87100
13      10  MG822955.1_626         6.63600
12      11  MG822952.1_128         7.13400
0       12                         7.19157  (13 11)  G ( 8.0000, 60.0000 )
12      13                         6.84887  (14 10)
13      14                         6.80286  (15 21)
14      15                         6.35558  (16  7)
```

```
15     16                          5.60562  (17  6)
16     17                          5.02859  (18 19)
17     18                          4.90459  ( 1  2)
17     19                          4.56750  ( 3 20)
19     20                          4.05376  ( 4  5)
14     21                          6.14101  ( 8  9)
```

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 in the same time scale as the time as the end of the sequence name. The values in the columns match the order of the line above the table, " Posterior means (95% Equal-tail CI) (95% HPD CI) HPD-CI-width". HPD stands for highest posterior density and CI stands for credible interval. The time parameters also have the mean and 95% HPD CI are also shown in the original time scale (as was used for dates at the end of the sequence names) after Jnode.

```
Posterior means (95% Equal-tail CI) (95% HPD CI) HPD-CI-width

t_n12         7.2929 ( 7.2059,  7.4107) ( 7.1980,  7.3978)  0.1998  (Jnode 20) time: -30.878 (-135.754, 64.000)
t_n13         7.0598 ( 6.8090,  7.3063) ( 6.8105,  7.3074)  0.4969  (Jnode 19) time: 202.227 (-45.409, 451.455)
t_n14         6.8793 ( 6.5837,  7.1733) ( 6.5850,  7.1743)  0.5893  (Jnode 18) time: 382.676 (87.657, 676.959)
t_n15         6.2727 ( 5.9161,  6.6866) ( 5.8947,  6.6572)  0.7625  (Jnode 17) time: 989.288 (604.771, 1367.269)
t_n16         6.0201 ( 5.7275,  6.4080) ( 5.7078,  6.3730)  0.6652  (Jnode 16) time: 1241.897 (888.970, 1554.176)
t_n17         5.7366 ( 5.3198,  6.1723) ( 5.2991,  6.1494)  0.8503  (Jnode 15) time: 1525.428 (1112.579, 1962.904)
t_n18         5.1042 ( 4.6056,  5.6240) ( 4.5981,  5.6160)  1.0179  (Jnode 14) time: 2157.777 (1646.015, 2663.890)
t_n19         5.4115 ( 4.9578,  5.8929) ( 4.9357,  5.8681)  0.9324  (Jnode 13) time: 1850.473 (1393.889, 2326.315)
t_n20         5.0243 ( 4.4879,  5.5884) ( 4.4688,  5.5668)  1.0980  (Jnode 12) time: 2237.700 (1695.224, 2793.223)
t_n21         6.6983 ( 6.3973,  7.0122) ( 6.3887,  7.0028)  0.6141  (Jnode 11) time: 563.686 (259.175, 873.304)
t_n1          4.4406 ( 3.4527,  5.1826) ( 3.5727,  5.2459)  1.6732  (Jnode 10) time: 2821.378 (2016.088, 3689.289)
t_n5          3.6452 ( 3.2242,  4.3715) ( 3.2050,  4.2447)  1.0397  (Jnode  9) time: 3616.800 (3017.291, 4056.961)
mu            0.0130 ( 0.0092,  0.0176) ( 0.0089,  0.0172)  0.0083
kappa         8.2379 ( 5.2636, 12.2745) ( 4.9335, 11.7697)  6.8362
alpha         0.1147 ( 0.0394,  0.2342) ( 0.0287,  0.2136)  0.1849
lnL       -1356.4405 (-1362.1410, -1352.4760) (-1361.5000, -1352.1310)  9.3690
```

The FigTree file is in nexus format and is also reports the results the transformed time units. The FigTree file can be viewed graphically with the FigTree program avaliable at http://tree.bio.ed.ac.uk/software/figtree/

```
#NEXUS
BEGIN TREES;

UTREE 1 = ((((((MG823170.1_7262 [95%HPD=3.57271, 5.24591}]: 0.663007, MG822924.1_3286:
 ↪  1.128122) [&95%HPD={4.59811, 5.61598}]: 0.632368, (MG823012.1_2728: 0.877767,
 ↪  (MG822918.1_3599: 1.361555,
MG822922.1_4057 [&95%HPD={3.20504, 4.24471}]: 1.379493) [&95%HPD={4.46878, 5.56678}]:
 ↪  0.387212) [&95%HPD={4.93568, 5.86811}]: 0.324723) [&95%HPD={5.2991, 6.14942}]: 0.283395,
 ↪  MG822944.1_1677: 0.434884)
[&95%HPD={5.70782, 6.37303}]: 0.252735, MG822946.1_2077: 1.087620) [&95%HPD={5.89473,
 ↪  6.65723}]: 0.606817, (MG822943.1_1272: 0.708536, MG822947.1_1391: 0.827536)
 ↪  [&95%HPD={6.3887, 7.00282}]: 0.180901) [&
95%HPD={6.58504, 7.17434}]: 0.180153, MG822955.1_626: 0.423590) [&95%HPD={6.81055, 7.30741}]:
 ↪  0.233309, MG822952.1_128: 0.158898) [&95%HPD={7.198, 7.39775}];

END;

[Note for FigTree: Under Time Scale, set Offset = 7262.0, Scale factor = -1000.0
Untick Scale Bar, & tick Tip Labels, Node Bars, Scale Axis, Reverse Axis, Show Grid.]
```

Checking convergence

MCMCs need to run long enough to converge to the posterior distribution. While there is no way to be certain an MCMC has converged, one check is to run the MCMC multiple times with different seeds and compare the results. If the results are very similar, the MCMC has likely converged. If the results are not similar, the MCMC should be run longer. Try increasing the `burnin` and `nsamples` or sampling less frequently by increasing `sampfreq`.

Multiple gene example 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.

Example script

The contents of the example script, runExample.sh are shown below.

```
#!/bin/bash
# Data is originally from Abrahams et al. 2019

mkdir ENV2_Prior ENV2 ENV3_Prior ENV3

echo Running HIVtree. The MCMCs may take a few minutes to run.
cd ENV2_Prior
../../../HIVtree ../ENV2_Prior.ctl &> output &

cd ../ENV3_Prior
../../../HIVtree ../ENV3_Prior.ctl &> output &

cd ../ENV2
../../../HIVtree ../ENV2.ctl &> output &

cd ../ENV3
../../../HIVtree ../ENV3.ctl &> output &

cd ../

wait;

echo Finished running MCMCs.

echo Parsing MCMCs and preparing files to combine estimates.
../.././parseMCMC.sh sequences.csv

echo
echo Combining estimates across regions.
# Example of how to run the output files

echo Running combineEstimates.R with C1C2_W19_QVOA_3921.txt
Rscript ../../combineEstimates.R -m C1C2_W19_QVOA_3921.txt -s 0 -b 3.921 -t 1000 -l 3921 -g 2

echo
echo Running combineEstimates.R with C1C2_W14_QVOA_3921.txt
Rscript ../../combineEstimates.R -m C1C2_W14_QVOA_3921.txt -s 0 -b 3.921 -t 1000 -l 3921 -g 2
wait;
```

Running HIVtree

The script first makes directories for the analysis of each gene under the prior and with data and runs HIVtree. Then, the script moves into each of the new directories and runs HIVtree for both genes under the prior and with

data. The output of HIVtree is redirected to a file called output, which is inside of the directory for each analysis. While this section of the script is running, the following messages will be displayed:

```
Running HIVtree. The MCMCs may take a few minutes to run.
Finished running MCMCs.
```

Parsing the output of HIVtree

Next the script parseMCMC.sh is run with a mapping file, called "sequences.csv" in this example. The mapping file specifies which sequences from different genes originated from the same proviral sequence. The mapping file is specified as a command line argument, as shown below.

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

While this is running, the following will be printed to screen.

```
Parsing MCMCs and preparing files to combine estimates.
The tipDate time unit is 1000.00.
The time of the last sample is 3921.00.
Processing row 2 in the csv file.
Processing row 3 in the csv file.
```

The tipDate time unit and the time of the last sample are needed for the next step. The script will generate the MCMC summary files needed to run combineEstimate.R.

Running combineEstimate.R

The last section of the example script will infer the posterior probability of the latent integration time for both proviruses. This happens in the lines

```
Rscript ../../combineEstimates.R -m C1C2_W19_QVOA_3921.txt -s 0 -b 3.921 -t 1000 -l 3921 -g 2
```

and

```
Rscript ../../combineEstimates.R -m C1C2_W14_QVOA_3921.txt -s 0 -b 3.921 -t 1000 -l 3921 -g 2
```

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. The following will be displayed on the screen.

```
Combining estimates across regions.
Running combineEstimates.R with C1C2_W19_QVOA_3921.txt
Loading required package: KernSmooth
KernSmooth 2.23 loaded
Copyright M. P. Wand 1997-2009
[1] "Mean: 1694.86931603405 , 95% Credible Interval: 1282.31221520857 - 2273.8536282376"

Running combineEstimates.R with C1C2_W14_QVOA_3921.txt
Loading required package: KernSmooth
KernSmooth 2.23 loaded
Copyright M. P. Wand 1997-2009
[1] "Mean: 2342.91723382267 , 95% Credible Interval: 1750.72106461723 - 3111.70124802179"
```

The RScript will print out the mean and 95% credible set of the posterior distribution for the latent integration time, shown above.

Files and arguments for multiple gene analysis

Files for 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 for HIVtree must be redirected to a file called "output", as is shown in the example script.

Files for parseMCMC.sh

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 this example, this file is "sequences.csv". 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.

MCMC summary file The script 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. The beginning of "C1C2_W14_QVOA_3921.txt"is shown.

```
t_n5,t_n3,t_n5,t_n3
0.7025169,1.6940531,0.6690711,0.6451491
0.7787734,1.6940531,0.4383413,0.6451491
1.1294270,1.9301337,1.4273967,0.8991026
0.9051177,1.9301337,1.4273967,1.1498608
...
```

Arguments for combineEstimate.R

There are 6 arguments needed by the R script, which are described in the inputs and outputs section. All 6 arguments are required.

Graphing the results 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.

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.

Input and Output

HIVtree

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

Output	Keyword in control file	Description
MCMC file	mcmcfile	This contains the results of the MCMC. It can be viewed in programs such as tracer.
Tree file	NA	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	outfile	This has summary information about the data and MCMC.
Summary printed to screen	NA	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

Input	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. This is given as a command line argument.

Output	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. The names of the output files will match the name of the first sequence in each row of the csv file. If there is missing data, the first non-missing sequence name will be used.

combineEstimates.R

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

Output	Description
Summary printed to screen	The mean and the 95% credible set (with equal tail probabilities) for single latency time

Required arguments for combineEstimates.R

flag	description	Explanation
-m, --mcmcSummary	MCMC summary file	file created by parseMCMC.sh
-s, --sampleTime	sample time of the latent sequence	This is the sample time of the latent provirus. It must be in the same time units as are used in HIVtree
-b, --latentBound	latentBound	This should have the same value as the `latentBound` keyword used in HIVtree
-t, --timeUnit	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.
-l, --lastSample	last sample date	This is the last sample date in the whole phylogeny. This assumes that the last sample date is the same for all of the phylogenies. Since the sequences names must have the relative dates at the end of their names, this number will be the same as the youngest sample time (largest number) at the end of the sequence names in the fasta file.
-g, --genes	number of genomic regions	This is the number of genomic regions (used for error checking). This should be half the number of columns in the MCMC summary file. The maximum number of genomic regions allowed is 10. 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.

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.

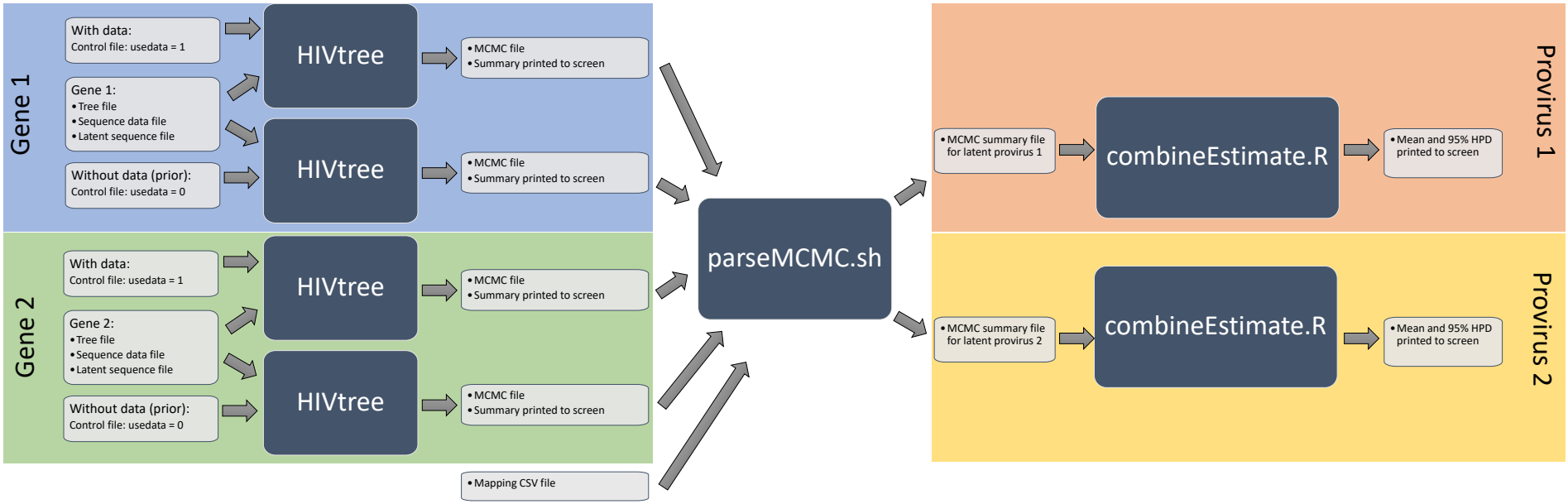

Figure 2: Workflow for estimating the latent integration time of two latent proviruses, each of which have sequences from two genes. Each additional gene adds a box on the left. Each additional provirus adds a box on the right. The dark boxes show programs or scripts and the light boxes show inputs and outputs.