

# *StepBrothers-1.0*

Erik Bloomquist

November 3, 2007

## 1 Installation

Please follow these steps on how to install *StepBrothers-1.0* .

1. Before you can use *StepBrothers-1.0* , you must obtain and install a Java virtual machine (JVM) on your computer. The JVM must support the Java 1.2 language specification. We currently use Sun's Java version 1.5. Free versions can be downloaded from [www.sun.com](http://www.sun.com).
2. Download `StepBrothers-1-0.tar.gz` from the website.
3. The next step is to unpack `StepBrothers-1-0.tar.gz`. To do this, create a new directory, say `StepBrothers`, and move `StepBrothers-1-0.tar.gz` into this directory. Then move into the directory you just created and type the following commands under the Linux bash shell.

```
[calypso]$ gunzip StepBrothers-1-0.tar.gz  
[calypso]$ tar -xvf StepBrothers-1-0.tar
```

4. Next, the environment variable `CLASSPATH` must be set such that it includes `StepBrothers-1.0.jar`. A command appropriate under the Linux bash shell is shown below. The command may either be issued from the command prompt or in the login script `.bashrc`.

```
[calypso]$ export CLASSPATH=$CLASSPATH:path_to_file/StepBrothers-1.0.jar
```

5. You should now be able to run the software. Please read the next few sections on how to specify the command file, how to specify the sequence file, and finally, how to run the software. To test that the software is running correctly, please see Section 8.

## 2 Sequence data

*StepBrothers-1.0* reads in Phylip 4.0 formatted sequence files. There is no maximum line length, so an entire sequence for one taxon may be entered on a single line. Sequences may also be wrapped by interleaving them. ***StepBrothers-1.0* assumes that the putative recombinant sequences are the LAST sequences in the Phylip data file.** Erroneous results will obtain if the putative recombinant are positioned elsewhere. Below is an example of a Phylip 4.0 formatted sequence file with putative recombinant sequences R and S positioned last.

```
9 80
A      NNNNNNNCGA GAGCGTCAGT ATTAAGNGGG GGAAAATTAG ATGCATGGGA
B      ATGGGTGCGA GAGCGTCAGT ATTAAGCGGG GGAGAATTAG ATAGATGGGA
C      ATGGGTGCGA GAGCGTCAAT ATTAAGAGGG GNAAAATTAG ATAAATGGGA
D      ATGGGTGCGA GAGCGTCAGT ATTAAGCGGG GGAAAATTGG ATGCATGGGA
F      ATGGGTGCGA GAGCGTCAGT ATTAAGCGGG GGAAAATTAG ATGCATGGGA
G      ATGGGTGCGA GAGCGTCAGT ATTAAGCGGG GGAAAATTAG ATGCTTGGGA
H      ATGGGTGCGA GAGCGTCAGT ATTAAGCGGG GGAAAATTAG ATGCTTGGGA
R      ATGGGTGCGA GAGCGTCAAT ATTAAGTGGG GGAAAATTAG ANGATTGGGA
S      ATGGGTGCGA GAGCGTCAGT ATTAAGCGGG GGAAAATTAG ATGCATGGGA

      GAGTCAAGTA CAACAGACA- --NNNAACAT
      GAGCCAAGTA ACAAATTCAN NNGCTACCAT
      GAGCCAAGCA AACAGTNCC- -----NNNAT
      GAGCCAAGCA ACAAGTTCAG CTGCTGCAGT
      GAGCCAAGCA ACAAATACAN NNNNNNCCAT
      GAGCCAGGCA TCAGGTGCAG CAGCAGCCAT
      GAGCCAAGTA ACAAATGCAA ATGCAGCCAT
      GAGCCAAGTG ACCAATACN- -----AACAT
      GAGCCAAGTA ---AATACAA ATGCAGTTAT
```

Allowable characters for the dataset include the standard for nucleotides, dashes, and all wild card characters. The wild card characters follow the IUPAC standard. **The program does not recognize uracil or U, you must use T.**

## 3 Command file

*StepBrothers-1.0* requires a text-based **command file** to run. This file specifies the parameters of the MCMC sampler. Efficient sampling depends on reasonable choices for these parameters.

In the **command file**, parameters are specified one per line in the form:

<parameter name>: <value>

. The only exception is the parameter `top_lambda`; its specification is given below. **The last line of the command file must contain the string END.** Below is a complete list of the parameters that may appear in the **command file** along with a default value and a brief description of each parameter's function. Two examples of command files can be found within the folders HIV and HBV in the *StepBrothers-1.0* release.

Parameter	Default Value	Description
input:	no default	The name of the data file in Phylip 4.0 format.
output:	no default	The name of the output file.
seed:	1	A integer to initialize the random number generator
mixing:	optional	This parameter specifies the file name of an <b>optional</b> mixing file for the tuning constants. Details for the mixing file are given in the next section.
length:	1100000	Total number of samples to generate
burnin:	100000	Number of initial samples to discard
subsample:	50	Frequency at which samples are saved to the <b>output file</b> after the burn-in
start_tree:	no default	Specifies the starting tree used during posterior simulation. Sequences within the tree should be labeled $0, 1, 2, \dots, R + P - 1$ , where $P$ is the number of parentals and $R$ are the number of recombinants. The recombinants are always use the numbers $P, \dots, M + P - 1$ . The location of the recombinants on the tree does not matter, but the topology of the parentals remains fixed throughout the alignment. Branch lengths must be specified at the beginning, but these parameters will be estimated during simulation. An example of a tree with 3 parentals and 2 recombinants is $(0:1,(1:0.5,(2:0.25,(3:0.125,4:0.125):0.125):0.25):0.5)$ .

num_recom:	no default	Represents the number of recombinants in the dataset. Remember, there must be at least two parents. <b>At the moment, the sampler can only handle 1 or 2 recombinants</b>
par_lambda:	1	Hyperprior parameter representing the prior mean number of substitution process change-points (See Equation (3) Minin et al. [2005]). In Minin et al. [2005], information is also given on how to specify this parameter.
tree_lambda:	0.1	Value representing $\lambda$ in Yule process specification. We recommend setting $\lambda$ small so that the yule process is uniform over the tree.
overlap_weight:	0	Value representing $w$ in the overlap prior. A value of 0 assumes that the locations of the breaks between sequences are independent.
top_lambda:	no default	Hyperprior parameter representing the prior mean number of topology break-points. Its specification depends on the number of recombinants present in the data. If there is only a single recombinant, you specify top_lambda as [0.5], where 0.5 is the value. If there are two recombinants, you specify top_lambda as [0.5,0.3] and so on. To determine an appropriate value of top_lambda, we recommend sampling from the prior distribution. In other words, feed into <i>StepBrothers</i> -1.0 an input file with all dashes “-”. Then, from the output file, calculate the mean number of topological change points for the recombinant sequences.
window_length:	10	Size of window around which existing change-points are randomly moved during update.
sigma_alpha:	0.75	Spread of $\alpha$ s when proposing new segments (see [Suchard et al., 2003])
sigma_mu:	0.75	Spread of $\mu$ s when proposing new segments (see [Suchard et al., 2003])
move_sd	0.4	Standard deviation of the normal distribution when adjusting branch lengths.

subst_hyper_mean:	no default	prior mean of $\log \kappa$
subst_hyper_variance:	no default	prior variance of $\log \kappa$
diver_hyper_mean:	no default	prior mean of $\log \mu$
diver_hyper_variance:	no default	prior variance of $\log \mu$
<hr/> <p>If you want to fix the four above parameters, you must provide values for all of them. If some of the values are missing the program will terminate. If ALL values for hyperparameters are missing they will be estimated simultaneously with other model parameters. (See Minin et al. [2005])</p> <hr/>		
top_breaks:	1	1 = Update topological breakpoint locations, 0 = do not update their locations
par_breaks:	1	1 = Update parameter breakpoint locations, 0 = do not update their locations

## 4 Mixing file

This section describes the mixing file. This file is not necessary to specify, but the values in this file can increase the efficiency of the sampler. Note that the bivariate jumps and the MoveBrothers jump are not used when the number of recombinants is only 1. **The last line of the mixing file must contain the string END.**

Parameter	Default Value	Description
AddOneTop:	0.07	The probability the sampler tries to add one topology break.
AddTwoTop:	0.03	The probability the sampler tries to add two topology breaks.
DeleteTop:	0.1	The probability the sampler tries to delete a topology break.
AddParam:	0.1	The probability the sampler tries to add a parameter break.
DeleteParam:	0.1	The probability the sampler tries to delete a parameter break.

BranchLength:	0.08	The probability the sampler tries to update branch lengths.
MoveBranch:	0.08	The probability the sampler tries to move a branch.
MoveBrothers:	0.05	The probability the sampler tries to move brother branches. Two branches are brothers if first they come from distinct recombinants, and second, the locations of the corresponding sequence space for the two branches overlap.
BivariateAddTop:	0.05	The probability the sampler tries to add multiple topological breaks. Not used if the number of recombinants is 1.
BivariateDeleteTop:	0.05	The probability the sampler tries to delete multiple topological breaks. Not used if the number of recombinants is 1.
		The remaining jumps, which includes updating the substitution parameters, updating the changepoint locations, and updating the hyperparameters, run with probability $1 - \sum$ (jumps above). With the default parameters, the remaining jumps occur with probability, 29%.

## 5 Running *StepBrothers-1.0*

To run the program, type in the Linux bash shell

```
[calypso]$ java mcp.app.StepBrothers <command file>
```

. here is an example

```
java mcp.app.StepBrothers bf103520.cmd
```

that tells *StepBrothers-1.0* to use the `bf103520.cmd` command file as input. After finishing the burnin, *StepBrothers-1.0* estimates the total runtime for the sampler. We usually find this estimate fairly accurate.

## 6 Interpreting the output

*StepBrothers*-1.0 saves each sample from the posterior distribution on a single line. Here is an example of an output with two recombinants.

```
125000 [L: 0, R: 50 Breaks: [0,1] P: Y IDs: [2,3], L:51, R: 100 Breaks: []
P: Y IDs: [],L: 101, R: 150 Breaks: [0] P: N IDs: [4,3], L:151 R:200 Breaks:
[0,1] P: N IDs:[5,6]] (((0:0.5,(2:0.25,3:0.25):0.25):0.25,4:0.75):0.25,
(1:0.5,(5:0.25,6:0.25):0.25):0.5) [(0.1,2),(0.2,5),(0.2,5),(0.2,5)]
```

The 125000 represents the state of the chain. The next portion represents the partition list. The first element says that the region from sequence position 0 to position 50 has its own set of substitution parameters, recombinant 0 corresponds to tip 2 on  $\tau$ , and recombinant 1 corresponds to tip 3. Note that the first position in the data is called position 0.

The next portion of the partition list says that the region from 51 to 100 in the sequence space corresponds to a different set of substitution parameters than the previous partition. Also note that Breaks: [] is empty, meaning recombinant 0 still corresponds to tip 2, and recombinant 1 corresponds to tip 3. The next element in the partition list says the substitution parameters remain the same (P: N), but recombinant 0 now corresponds to a new tip on the tree, tip 4. The final partition says both recombinant 0 and recombinant 1 correspond to new tips, 5 and 6, respectively.

The next portion after the partition list represents the tree topology in NEXUS format. The final portion of the output represents the substitution parameters. The first part (0.1,2) says that the sequence region from 0 to 50 has mutation rate,  $\mu$ , equal to 0.1, and transition-transversion rate,  $\kappa$ , equal to 2. The next portion (0.2,5) represents the sequence region from 51 to 100, and so on.

## 7 Summarizing the output

To help summarize the output, we provide five utility applications.

### 7.1 StepTopology

This application is run under the following command.

```
java mcp.tool.StepTopology <output file> <command file> <topology summary file>
    <key file> <recombinant>
```

This program finds the posterior probability of parentage for each site in the sequence space. The output file is the output from *StepBrothers-1.0*. The command file is the command file for *StepBrothers-1.0*. The topology summary file is the output file from StepTopology. The key file interprets the columns in the topology summary file, and the recombinant is an integer saying which recombinant to analyze. For example if recombinant = 0, the first recombinant will be analyzed, if recombinant = 1, the second recombinant will be analyzed, etc.

Here is an example key file and truncated topological summary file for a dataset using the start tree (0:1,(1:0.5,(2:0.25,3:0.25):0.25)) and one recombinant. We first begin with

```
[calypso]$ java mcp.app.StepBrothers example.cmd
```

, and then we type

```
[calypso]$ java mcp.tool.StepTopology example.out example.cmd example.top
example.key 0
```

to obtain the summary file. The key file, `example.key` output is

```
Column 1: 0
Column 2: 1
Column 3: 2
Column 4: 1 2
Column 5: 0 1 2
```

and a truncated version of the topological summary file, `example.top` is

```
0: 0.0 0.9 0.1 0.0 0.0
1: 0.0 0.8 0.2 0.0 0.0
2: 0.0 0.7 0.3 0.0 0.0
3: 0.0 0.4 0.6 0.0 0.0
4: 0.0 0.4 0.6 0.0 0.0
```

The 5 columns in `example.key` are the label headings for last five columns in `example.top`. The first row, says that sequence position 0 has a 90% posterior probability of a parentage from taxon 1. The final row says that sequence position 4 has a 60% posterior probability of parentage from taxon 2, and a 40% posterior probability of parentage from taxon 1.

## 7.2 StepBreakpoints

This program is run under the command

```
java mcp.tool.StepBreakpoints <output file> <breakpoint file> <recombinant>
```

and it summarizes the posterior distribution of the locations of the topological breakpoints for the recombinant entered. The first line of output gives the number of samples for which a breakpoint is located between sequence position 0 and sequence position 1. The second line gives the number of samples a breakpoint occurred between 1 and 2, and so on.

## 7.3 StepDate

This program is run under the command

```
java mcp.tool.StepDate <output file> <date file> <parentals> <recombinant>
```

and it gives information on the height from the parental tree for every site in the sequence. The first column gives the 97.5% percentile, the second column gives the median, the third column gives the 2.5% percentile, and the final column gives the sample mean. More information on what these heights mean can be found in Bloomquist et al. [2007].

## 7.4 StepProfile

This program is run under the command

```
java mcp.tool.StepProfile <output file> <profile file>
```

and it gives a summary of  $\mu$  and  $\kappa$  for all sites in the sequence. The first column is the 2.5% quantile of  $\mu$  from the posterior distribution; the second column is the 50% quantile of  $\mu$ ; the third column is the 97.5% quantile of  $\mu$ ; and the fourth column is the posterior expectation of  $\mu$ . The next four columns give the same summary statistics on  $\kappa$  rather than  $\mu$ .

## 7.5 StepOverlap

This program is run under the command

```
java mcp.tool.StepOverlap <over file>
```

, where the over file has the following form

```

output: bf103520.out
overlap: bf103520.overlap
numrecom: 2
parents: 4
r1: (0,0,1500)
r2: (1,0,1500)
END

```

. The output command specifies the name of the output file from *StepBrothers*-1.0 ; the overlap command specifies where you want StepOverlap to dump its output; numrecom gives the number of recombinants; and parents is simply the number of parental sequences. The command r1 has three components: the first says which recombinant you want to analyze, the second says which region of the sequence space you want to start at, the final command specifies the last location of the sequence space you want to look at. The command r2 has the same form.

The output from StepOverlap is a large matrix where every cell is the probability of overlap between a site in the recombinant specified in r1 and the recombinant specified in r2. Note, the same recombinant can be specified for both r1 and r2. Each column consists of the sequence positions in r2, while each row consist of the sequence positions in r1. More details on the meaning of the values in the matrix can be found in Bloomquist et al. [2007].

We also provide an R-script, overlapmatrix.Rscript, that transforms the matrix into a heat map. To run the R-script, simply open the file with a text editor and set the five values in the top of the script. Then save the file, and run it from R; a postscript file will appear.

## 8 Testing

Two examples are provided, an HIV example and a HBV example. If running correctly, the first line of screen output on the HIV example should read,

```

Burnin 1% complete
((0:0.784447,(((3:0.163278,6:0.163278):0.228369,8:0.391647):0.093662,7:0.485309
):0.299138):0.215553,(((4:0.388262,5:0.388262):0.185748,(1:0.389374,9:0.389374)
:0.184637):0.333451,2:0.907462):0.092538)
[L: 0 R: 21 Breaks: [0,1] P: Y IDs: [8,7], L: 22 R: 109 Breaks: [] P: Y IDs: [8
,7], L: 110 R: 163 Breaks: [] P: Y IDs: [8,7], L: 164 R: 209 Breaks: [] P: Y ID
s: [8,7], L: 210 R: 216 Breaks: [] P: Y IDs: [8,7], L: 217 R: 245 Breaks: [1] I
Ds: [8,5], L: 246 R: 490 Breaks: [0] IDs: [4,5], L: 491 R: 793 Breaks: [] P: Y
IDs: [4,5], L: 794 R: 1458 Breaks: [0] IDs: [9,5], L: 1459 R: 1496 Breaks: [0]

```

IDs: [6,5]]

and for the HBV example,

Burnin 1% complete

```
(((((0:0.155926,1:0.155926):0.030377,((2:0.009247,16:0.009247):0.158087,10:0.167334):0.018969):0.110163,(13:0.167874,11:0.167874):0.128592):0.646268,(6:0.404454,7:0.404454):0.538280):0.057267,((((9:0.005508,14:0.005508):0.169423,(5:0.162928,15:0.162928):0.012003):0.050167,(4:0.193696,3:0.193696):0.031402):0.007045,8:0.232143):0.073121,12:0.305264):0.694736)
```

```
[L: 0 R: 7 Breaks: [0,1] P: Y IDs: [9,15], L: 8 R: 116 Breaks: [1] IDs: [9,8], L: 117 R: 477 Breaks: [] P: Y IDs: [9,8], L: 478 R: 571 Breaks: [] P: Y IDs: [9,8], L: 572 R: 641 Breaks: [] P: Y IDs: [9,8], L: 642 R: 686 Breaks: [] P: Y IDs: [9,8], L: 687 R: 892 Breaks: [] P: Y IDs: [9,8], L: 893 R: 1371 Breaks: [] P: Y IDs: [9,8], L: 1372 R: 1381 Breaks: [1] IDs: [9,10], L: 1382 R: 1564 Breaks: [0] IDs: [11,10], L: 1565 R: 1794 Breaks: [] P: Y IDs: [11,10], L: 1795 R: 2167 Breaks: [1] IDs: [11,13], L: 2168 R: 2208 Breaks: [] P: Y IDs: [11,13], L: 2209 R: 2231 Breaks: [0] IDs: [14,13], L: 2232 R: 2357 Breaks: [] P: Y IDs: [14,13], L: 2358 R: 2783 Breaks: [] P: Y IDs: [14,13], L: 2784 R: 2832 Breaks: [] P: Y IDs: [14,13], L: 2833 R: 2874 Breaks: [1] IDs: [14,16], L: 2875 R: 2877 Breaks: [] P: Y IDs: [14,16], L: 2878 R: 2914 Breaks: [1] IDs: [14,12], L: 2915 R: 3014 Breaks: [] P: Y IDs: [14,12], L: 3015 R: 3171 Breaks: [] P: Y IDs: [14,12], L: 3172 R: 3223 Breaks: [] P: Y IDs: [14,12]]
```

## 9 Colt

The Colt library for scientific computing is included in `StepBrothers-1.0.jar`. More information on Colt, and its corresponding packages can be found at <http://dsd.lbl.gov/~hoschek/colt/>. The Colt library is distributed under the following license.

Packages `cern.colt*`, `cern.jet*`, `cern.clhep`

Copyright (c) 1999 CERN - European Organization for Nuclear Research.

Permission to use, copy, modify, distribute and sell this software and its documentation for any purpose is hereby granted without fee, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. CERN makes no representations about the suitability of this software for any purpose. It is provided "as is" without expressed or implied warranty.

Packages hep.aida.\*

Written by Pavel Binko, Dino Ferrero Merlino, Wolfgang Hoschek, Tony Johnson, Andreas Pfeiffer, and others. Check the FreeHEP home page for more info. Permission to use and/or redistribute this work is granted under the terms of the LGPL License, with the exception that any usage related to military applications is expressly forbidden. The software and documentation made available under the terms of this license are provided with no warranty.

## References

- E.W. Bloomquist, K.S. Dorman, and M.A. Suchard. The (evil) stepbrothers: inferring partially shared ancestries among recombinant viral sequences. *Biostatistics*, Submitted, 2007.
- V.N. Minin, K.S. Dorman, F. Fang, and M.A. Suchard. Dual multiple change-point model leads to more accurate recombination detection. *Bioinformatics*, 21(13):3034–3042, 2005.
- M.A. Suchard, R.E. Weiss, K.S. Dorman, and J.S. Sinsheimer. Inferring spatial phylogenetic variation along nucleotide sequences: A multiple changepoint model. *Journal of the American Statistical Association*, 98(462):427–437, 2003.