CAFE:
software for Computational Analysis of gene Family Evolution

Tijl De Bie, Nello Cristianini, Jeffery P. Demuth, and Matthew W. Hahn

January 13, 2006

This document describes how to download and use CAFE. The purpose of this software is to analyze changes in gene family size in a way that accounts for phylogenetic history and provides a statistical foundation for evolutionary inferences. The program uses a random birth and death process to model gene gain and loss across a user specified tree structure. The distribution of family sizes generated under the random model provides a basis for assessing the significance of the observed family size differences among taxa.

The necessary inputs for CAFE are:

1) a **data file** containing gene family sizes for the taxa included in the phylogenetic tree
2) a **Newick format phylogenetic tree**, including branch lengths
3) an initial value for the **birth and death parameter, λ**

From the inputs above, CAFE will compute:

1) the **maximum likelihood value of λ**
2) **ancestral states** for each node in the phylogenetic tree
3) **p-values** for each gene family describing the likelihood of the observed sizes given random gain and loss.
4) **average gene family expansion** along each branch in the tree
5) numbers of **gene families with expansions, contractions, or no change** along each branch in the tree

CAFE is a stand-alone Java program developed with portability and ease-of-use in mind. The general instructions below should be sufficient to employ CAFE on most systems, but we also include several notes that reflect our experience on Mac OSX and Windows. Throughout the text: *italicized type* begins platform specific notes; *underlined italics* are important notes for all platforms; `courier font` is used to indicate filenames, paths, and other user inputs (including data examples); and **bold type** indicates links to outside URLs and other text within the document.

CITING CAFE

The appropriate citation for use of CAFE in published research is:

> De Bie et al. 2006. CAFE: A computational tool for the study of gene family evolution. Bioinformatics x:xx-xx.

Original development of the statistical framework and algorithms implemented in CAFE are published in:

> Hahn et al. 2005. Estimating the tempo and mode of gene family evolution from comparative genomic data. Genome Research 15: 1153-60.

The first application of CAFE to complete genome analysis:

Demuth et al. 2006. Creation, extinction and evolution of mammalian gene families. *Nature* (in review)

SYSTEM REQUIREMENTS

CAFE requires Java Version 1.5 or greater.

Java 1.5 is the version included with J2SE 5.0. To verify which version of Java is installed and install any necessary updates visit **http://java.com**.

OSX users note:

OSX comes with Java 1.3 and 1.4.2 installed. Installing Java 1.5 (J2SE 5.0) will not replace either of the native OSX versions. In fact, Java 1.4.2 remains the default version of Java used by applications and applets even after the Java 1.5 update is installed. To change the default to Java 1.5, open the following utility: `/Applications/Utilities/Java/J2SE 5.0/Java Preferences`. Under the "Java Application Runtime Settings" change the version order by dragging `J2SE 5.0` to the top of the list then `Save` to close. Additional documentation related to configuring Java on OSX, can be found at the Apple Website (**http://developer.apple.com/java/**)

DOWNLOADING

The CAFE program consists of a single 40 KB file titled `cafe.jar` which can be downloaded from **http://www.bio.indiana.edu/~hahnlab/Programs/cafe.jar**. Once saved, no additional installation is required.

PC users note:

In some cases a different default extension (such as .zip) will be suggested in the download dialog box. To make CAFE executable, you must rename the file with the .jar extension (or otherwise associate the file with Java Virtual Machine).

The example data file used to generate the examples in this manual are also available for download at: (**http://www.bio.indiana.edu/~hahnlab/Programs/CAFE/example_data.tab**).

LAUNCHING

There are two ways to launch CAFE. The simplest way is to double click the `cafe.jar` file. Provided Java 1.5 is the default version, CAFE with launch with the default maximum memory allocation set by your version of Java or by your operating system (typically 64MB).

COMMAND LINE: The primary reason to run CAFE from the command line is to increase the maximum memory available to Java Virtual Machine. Another advantage to running CAFE from the command line is that the terminal window provides immediate access to the progress log of each analysis as it is computed by CAFE. A log of the analysis is also written to `logfile.txt` at the conclusion of each run even if CAFE is not launched from the command line.

PC command line:

```
java -jar -XmxNNNm \path_to_cafe\cafe.jar
```

OSX command line:

```
/path_to_java_1.5/java -jar -XmxNNNm /path_to_cafe/cafe.jar
```

The flag `-XmxNNNm` increases the maximum memory available to JVM by `NNN` MB. The default maximum memory allocation is 64 MB. Several factors influence the amount of memory necessary to run CAFE, including: which analyses are chosen, the number of taxa, the number of gene families, and the gene family sizes. If you attempt to run an analysis, but the program stops and the logfile.txt has the message "java.lang.OutOfMemoryError" at the end, you will need to increase the maximum allotment.

Benchmark memory requirements:

With 512MB allocated, CAFE could not complete the Likelihood Ratio Test on ~9,500 families in 5 taxa using 10,000 random samples. The analysis ran successfully with 2,048MB of memory allotted, but CAFE may not have required the full 2GB.

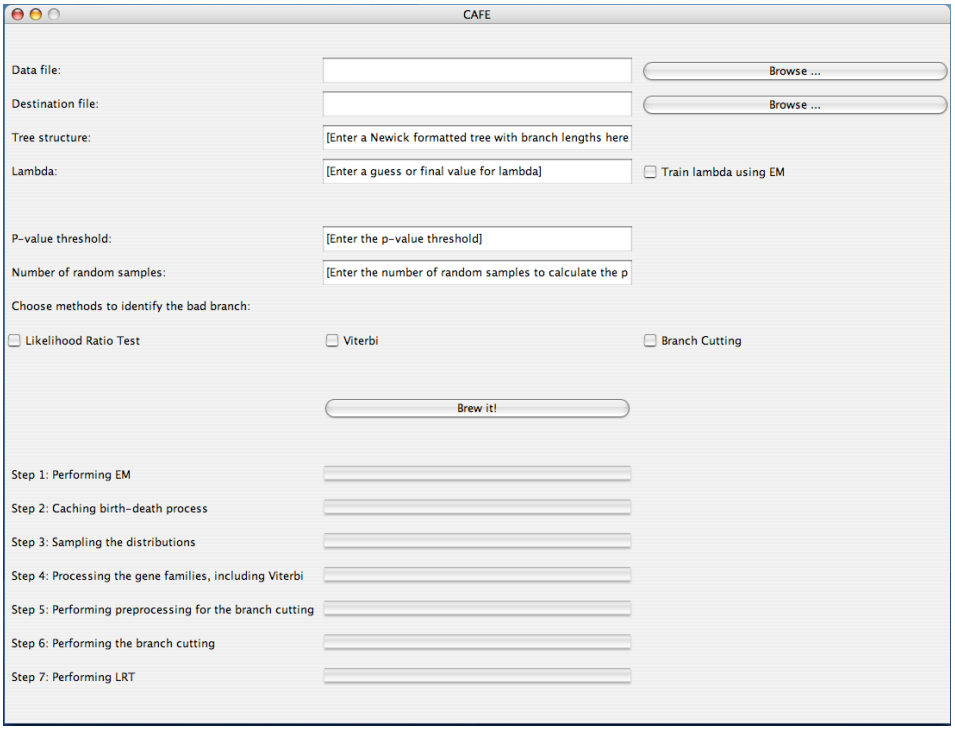
OSX users note:

When launching CAFE from the Unix Terminal in OSX you must specify the path to the `java` command in Java 1.5, otherwise, OSX defaults to Java 1.4 and you will get an error message. This happens irrespective of whether you set up Java 1.5 as the default under the Java Preferences Utility.

OSX Command Line Example (with the default Java 1.5 installation path):

```
/System/Library/Frameworks/JavaVM.framework/Versions/1.5.0/Home/
bin/java -jar -Xmx2048m /Users/username/Desktop/CafeFolder/cafe.jar
```

Successfully launching CAFE opens the interface below in a new window.



INPUTS

DATA FILE: Enter the path to the file containing gene family data. The data file format must be tab delimited with Unix (Plain Text) line endings. The first line contains the names of extant taxa separated by tabs (column headers). The taxon names must be spelled exactly as they are in the **tree structure**, but the order of taxa in the data file does not matter. Subsequent lines each correspond to a single gene family and contain data corresponding to each column header listed on line 1. If the data file contains column headers that do not appear in the tree structure, they are copied directly to the destination file without being considered. This is a useful property for annotating gene family data in the **destination file**

Example Data File:

```
FAMILY  Dog  Chimp  Human  Mouse  Rat  FAMILYDESC
ENSF00000002057  0   0   0   0   0   UNKNOWN
ENSF00000001251  11  12  14  12  4   RHO GUANINE NUCLEOTIDE
ENSF00000001658  8   11  21  8   5   EXOCYST COMPONENT
ENSF00000001751  4   7   25  10  7   AMBIGUOUS
ENSF00000001803  12  10  16  9   6   VANIN
ENSF00000001304  5   10  26  10  2   AMBIGUOUS
ENSF00000001340  12  7   7   15  12  GLUTAMATE RECEPTOR
ENSF00000002474  4   8   12  13  16  GABA A RECEPTOR ASSOC
ENSF00000001420  6   13  17  9   8   AMBIGUOUS
ENSF00000002563  7   9   18  12  6   SPROUTY HOMOLOG SPRY
```

The complete example data file used to generate example outputs is available for download at:
http://www.bio.indiana.edu/~hahnlab/Programs/CAFE/example_data.tab

Important! Because the random birth and death process assumes that each family has at least one gene at the root of the tree, CAFE will not give accurate results if gene families are included that are not present in the most recent common ancestor (MRCA) of all taxa included in the data file. For example, even if all the taxa in the data file had a family with 0 genes, CAFE will assign the MRCA a gene family size of 1, and include the family in estimation of the birth and death rate. This difficulty does not affect analyses containing families that go extinct subsequent to the root node.

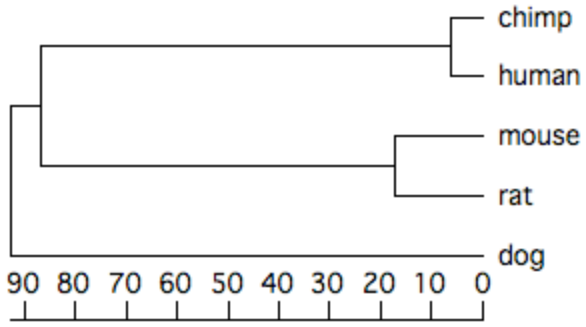
DESTINATION FILE: Enter the path and filename where CAFE will write the **main output**. If the file does not exist, CAFE will create it for you. If the file already exists, CAFE will overwrite the previous file.

TREE STRUCTURE: Enter the phylogenetic tree with branch lengths in Newick format. Branch lengths should be integer units of time and ultrametric (i.e. the sum of lengths from root to tip should be the same for all paths). For instructions on converting trees to Newick format visit: **http://evolution.genetics.washington.edu/phylip/newicktree.html**.

CAFE will allow commas or spaces to separate sister taxa. Taxon names should not contain spaces. Do not include a semicolon at the end of the line.

In Newick format, the tree diagram below is represented as:

```
(((chimpanzee:6,human:6):81,(mouse:17,rat:17):70):6,dog:93)
```



LAMBDA: Enter a value for λ, the birth and death rate parameter (CAFE assumes birth and death rates are equal). If **Train Lambda using EM** is selected, CAFE will use the user input as a seed value for the maximum likelihood estimation of λ. CAFE will not run without a user input for λ. Additionally, CAFE does not allow the product of λ and the longest branch in the tree structure to exceed one (i.e. $\lambda * t < 1$ must be true; where t is the longest branch). $\lambda = 0.002$ is a good starting value if there is no prior expectation for the true value.

TRAIN LAMBDA USING EM: If this box is selected, CAFE will compute the maximum likelihood estimate of λ for the distribution of family sizes observed in the data file. The expectation maximization (EM) algorithm employed by CAFE computes the likelihoods for the 11 values $1.2^i * \lambda$, where $i = -5,-4,\ldots,0,\ldots,4,5$. The value resulting in the largest likelihood becomes λ for all subsequent analyses.

Important! Always check the **logfile** to see which value of λ resulted in the highest likelihood during the EM procedure. If it was one of the most extreme values tested (i.e. $i = -5$ or 5), then the Train Lambda using EM procedure should be run again using the new estimate for λ as the user input. Repeat this process until the maximum likelihood estimate of λ is not one of the most extreme values tested. Because of the potential for the EM to select a sub-optimal λ, it is advisable to run only the Train Lambda using EM (no additional analyses selected) until the best value of λ is determined. This is particularly true for large data sets because CAFE may require a long computation time for analyses that may need to be re-run with a different λ.

P-VALUE THRESHOLD: For each family in the data file, CAFE computes a probability (p-value) of observing the data given random gain and loss of genes. All else being equal, families with more variance in size are expected to have lower p-values. The p-value threshold allows the user to specify the cutoff for subsequent analyses. Families with p-values larger than the designated threshold will not be included in identification of the most unlikely branch.

NUMBER OF RANDOM SAMPLES: To compute p-values, CAFE uses a Monte Carlo re-sampling procedure. Enter the number of samples CAFE should use to calculate p-values. The tradeoff is between precision and computation time; in most cases 1000 samples should provide reasonable balance.

CHOOSE METHODS TO IDENTIFY THE BAD BRANCH: For each family with a p-value below the user specified threshold, select which method(s) CAFE should use to identify the branch that is the most likely cause of deviation from the random model.

VITERBI: uses the so-called 'Viterbi' assignments to the ancestral nodes, and subsequently computes a p-value for the transition from parent to child node along each branch of the tree. Branches with low p-values represent unusually large changes, either contractions or expansions.

BRANCH CUTTING: calculates whether the overall p-value associated with a gene family increases if we cut one of the branches of the tree. By `cutting' a branch we mean removing the probabilistic coupling between the parent and child family sizes for that branch. A p-value is then computed for the gene family given the tree with one branch removed as a model (and this is done for each branch separately). If the p-value increases considerably after cutting a branch, this branch may be held responsible for the overall low p-value of the complete model.

LIKELIHOOD RATIO TEST: maximizes the likelihood of the gene family by estimating a separate value for the evolutionary rate parameter, λ, along the branch under investigation. The ratio of the likelihood model with two parameters to the likelihood with just a single parameter can be used to assess the need for an extra parameter along individual branches. High values therefore indicate branches along which there has been a larger-than-expected amount of evolutionary change.

Although the three methods are expected to yield similar results, they differ in nature, and comparison among methods can provide insights not apparent from analysis using any one individually. Additional details germain to each method are presented in Hahn et al. (2005).

BREW IT!: As soon as you select the "Brew it!" button, status bars to the right of Steps 1-7 should start to turn blue as each of the computations progresses. Steps that are not necessary for the user specified analyses should immediately turn completely blue. On large data sets some computationally intesive steps may appear not to move for extended periods. This is particularly true of Step 7: Performing the LRT.

To verify whether CAFE is indeed running, the user can either check the command window (if launched from the command line) or open a system-monitoring tool (e.g. Task Manager on PC or Activity Monitor on OSX) to see whether the Java Virtual Machine is still using a large percentage of CPU resources. When CAFE completes the analysis, a summary is written to `logfile.txt`. If CAFE stops unexpectedly, the logfile will contain an error message at the point where the analysis failed.

OUTPUTS

DESTINATION FILE: The primary output file is tab-delimited with a header line specifying the contents of each column. Subsequent lines specify results for individual gene families.

Example output (viewed in spreadsheet):

Tree in Newick format	P-value	Viterbi tree in Newick format	P-value for branch 1	P-value for branch 2	P-value after cutting branch 1	P-value after cutting branch 2	Likelihood Ratio for branch 1	Likelihood Ratio for branch 2	family	familydesc
(((0:6 0:6):81 (0:17 0:17):70):6 0:93)	0.6545	(((0 0 0) 1 (0 0 0)) 1 0)							ENSF00000002057	UNKNOWN
(((12:6 14:6):81 (12:17 4:17):70):6 11:93)	0.001	(((12 13 14) 11 (12 9 4)) 11 11)	0.7120831	0.4406644	0.001	5.00E-04	1	1	ENSF00000001251	RHO EXCHANGE FACTOR 10
(((11:6 21:6):81 (8:17 5:17):70):6 8:93)	0	(((11 15 21) 10 (8 7 5)) 10 8)	0.6997728	0.0912251	0	0	1	1.8866537	ENSF00000001658	EXOCYST
(((7:6 25:6):81 (10:17 7:17):70):6 4:93)	0	(((7 13 25) 8 (10 8 7)) 8 4)	0.6719996	0.0585492	0	0	1.7504777	2.1025074	ENSF00000001751	AMBIGUOUS
(((10:6 16:6):81 (9:17 6:17):70):6 12:93)	0.002	(((10 13 16) 11 (9 8 6)) 11 12)							ENSF00000001803	VANIN
(((10:6 26:6):81 (10:17 2:17):70):6 5:93)	0	(((10 15 26) 9 (10 7 2)) 9 5)	0.6864451	0.0320278	0	0	1	4.4688782	ENSF00000001304	AMBIGUOUS
(((7:6 7:6):81 (15:17 12:17):70):6 12:93)	0.503	(((7 7 7) 11 (15 13 12)) 11 12)							ENSF00000001340	GLUTAMATE RECEPTOR
(((8:6 12:6):81 (13:17 16:17):70):6 4:93)	0.005	(((8 10 12) 10 (13 14 16)) 10 4)							ENSF00000002474	GABA A RECEPTOR ASSOCIATED
(((13:6 17:6):81 (9:17 8:17):70):6 6:93)	0.026	(((13 14 17) 10 (9 9 8)) 10 6)							ENSF00000001420	AMBIGUOUS
(((9:6 18:6):81 (12:17 6:17):70):6 7:93)	0	(((9 13 18) 10 (12 9 6)) 10 7)	0.6997728	0.2423472	0	0	1	1.0526113	ENSF00000002563	SPROUTY HOMOLOG SPRY
(((12:6 16:6):81 (10:17 5:17):70):6 9:93)	0.005	(((12 14 16) 10 (10 8 5)) 10 9)							ENSF00000001515	TOUSLED KINASE
(((7:6 21:6):81 (10:17 7:17):70):6 7:93)	0	(((7 13 21) 10 (10 9 7)) 10 7)	0.6997728	0.2423472	0	0	1	1.1303973	ENSF00000001004	DSH HOMOLOG
(((11:6 17:6):81 (9:17 12:17):70):6 3:93)	0	(((11 13 17) 9 (9 10 12)) 9 3)	0.6864451	0.1097658	0.002	0	3.8935174	1.4682365	ENSF00000001656	TUMOR TCTP
(((11:6 12:6):81 (11:17 11:17):70):6 7:93)	0.737	(((11 11 12) 10 (11 11 11)) 10 7)							ENSF00000001765	SYNUCLEIN
(((7:6 13:6):81 (18:17 7:17):70):6 7:93)	0	(((7 10 13) 10 (18 12 7)) 10 7)	0.6997728	0.7859352	0	0	1	1	ENSF00000002390	OTOPETRIN
(((7:6 14:6):81 (11:17 18:17):70):6 2:93)	0	(((7 10 14) 9 (11 13 18)) 9 2)	0.6864451	0.5054834	0	0	8.9532699	1	ENSF00000001861	40S RIBOSOMAL S7
(((11:6 19:6):81 (11:17 5:17):70):6 6:93)	0	(((11 14 19) 9 (11 8 5)) 9 6)	0.6864451	0.0746819	0	0	1	1.6394468	ENSF00000001827	GEFDB SL REGULATOR
(((6:6 15:6):81 (14:17 7:17):70):6 10:93)	0	(((6 10 15) 10 (14 10 7)) 10 10)	0.6997728	0.7859352	0	0	1	1	ENSF00000001957	AMBIGUOUS

Column 1: The Newick tree structure specified by the user but with taxon names replaced by the number of genes indicated in the data file.

Column 2: p-values for each gene family

Column 3: Newick tree indicating family sizes from the data file as well as the CAFE assignment of sizes at ancestral nodes

The next series of columns report results of methods for identifying the bad branch. If all analyses were chosen, the first *n* columns report Viterbi results, the second *n* columns report branch cutting, and the third *n* columns report the likelihood ratio test results (*n* being the number of branches in the phylogenetic tree). Interpretation of the values in these columns is noted **above**. To match the branch numbers in the output file with actual branches in the tree structure, see the **logfile**.

The last columns are those that were present in the data file but did not have corresponding identifiers in the tree structure.

The complete output from the example-data are available for download at:
http://www.bio.indiana.edu/~hahnlab/Programs/CAFE/example_output.tab.

Important! No output is written to the destination file if CAFE fails to complete all of the requested analyses.

LOGFILE: The log of CAFE's progress is written at completion of the analyses or in the event that CAFE stops unexpectedly. This file contains a record of user inputs and a summary of the computations in each of the analysis steps. If CAFE was launched via double-clicking, `logfile.txt` is written to the same directory that contains the `cafe.jar` file. If CAFE was launched from the command line, `logfile.txt` is written to your systems default directory (on both PC and OSX the default is typically the home directory for the current user).

Important! CAFE automatically overwrites, or creates, `logfile.txt` at the beginning of each run (even though the file is empty until the end of all analyses). To save logfiles from previous analyses be sure to change the file name or move the prior logfile to a different directory.

RESULTS CONTAINED IN THE LOGFILE:

The first section reports **user inputs**. The branch numbering assigned by CAFE is presented directly below the user specified tree structure.

If Train Lambda using EM was chosen by the user, the next section, "Step 1", records the tested values of λ, their corresponding likelihoods, and the "optimal value for lambda" (see **caveats** above to determine whether this is actually the optimal λ).

The next several sections record progress on Steps 2-7 and are useful for determining where CAFE stopped in the event that an analysis fails to finish.

The last section of the logfile, "Global information per branch", reports the "average expansion size along each branch" (average contractions = negative expansions) and the number of gene families that changed size. The column order for both sets of results is: branch 1, 2, 3, … *n* (*n* is the total number of branches). CAFE calculates the average expansion size as the difference in the proportion of genes across all families that were gained or lost along each branch; thus, negative values indicate average contractions.

$$\text{Average expansion size} = \frac{\text{total genes gained along branch} - \text{total genes lost along branch}}{\text{total genes at ancestral node of branch}}$$

The final three rows indicate the numbers of gene families that underwent: expansion (row 1), no change (row 2), or contraction (row 3), along each branch.

For example:

BRANCH 1 BRANCH 8

```
-----------------------------------------------------------------
Global information per branch

Average expansion size along each branch:
-0.0033200532  1.2337317      -1.9614874     2.7151394      0.62284195     2.2709162      -1.5876493     -1.9143426

Number of expansions, number of no-changes, and number of contractions along each branch:
0     916   100   1275  706   1199  161   149
1502  356   153   161   459   187   175   147
4     234   1253  70    341   120   1170  1210
-----------------------------------------------------------------
```

USERPROFILE: A file `userprofile.txt` is created in the same directory as `logfile.txt` and contains a record of user inputs. This file is only a convenience so that users are not required to re-enter all of the inputs before each run.