Crann Manual.

Crann: A program for detecting adaptive evolution in protein-coding DNA sequences.

Crann was written by Chris Creevey while working in the lab of James McInerney at the National University of Ireland between 1999 and 2002.

AIM: To detect adaptive evolution in protein-coding genes.

The program Crann has been developed in order to provide fast heuristic methods of detecting adaptive evolution in protein-coding genes. It is important that the user understands the advantages and limitations of these methods. It is also important for the user to know that the software is designed to perform a number of different tasks, however the interpretation of the results is left entirely to the user.

Disclaimer:

While we try to ensure that the software is free of bugs, this cannot be guaranteed. The software is provided as-is, with no guarantee that it will do anything, that it is suitable for any purpose whatsoever and that it will be of any use to anybody. We cannot be held responsible for any errors and we cannot be held responsible for the user being misled by any results they obtain when using the software.

Background.

Adaptive evolution can generally be described as the retention of new mutations that can somehow confer some kind of selective advantage on the individual that possesses this change.

The above statement is very broad and it seems that it is a little difficult to describe what is meant by adaptive evolution. Any mutation might be an advantageous one and occur in a protein-coding region, or in a control region, or anywhere in the genome really. It is likely that what is important is the interaction between this change and the environment in which the change has occurred. Take as a hypothetical example, an amino acid change from a leucine to a phenylalanine in a highly-expressed protein. We know that phenylalanine is an expensive (in terms of the number of high-energy bonds) amino acid to produce. This might mean that in a highly-expressed gene the organism might need to produce many thousands of phenylalanine molecules more than its wild-type ancestor. This additional metabolic burden might seem like a disadvantage, but if the mutant protein is now better-suited to carrying out its activity (whatever that might be), then the cost of producing more phenylalanines may be offset by the benefit of the new function [ref].

A mutation that changes the sequence of a promoter might result in a change that is beneficial to the possessor. We might, therefore, describe this situation as adaptive evolution. Again, it describes a situation where an advantage accrues as a result of a mutation.

We know that synonymous codon usage is frequently influenced by selective pressures for optimal translational speed and accuracy (Sharp, Stenico et al. 1993), as well as transcriptional and replicational efficiency (McInerney 1998). We might therefore describe any mutations that change the character state of a codon to that of a 'preferred' codon as an advantageous or adaptive event.

Crann was designed to implement algorithms for detecting a certain kind of adaptive evolution in protein-coding genes.

It might be useful to be a little more specific and speak about adaptive evolution in the sense that most people speak about it. We might describe a truly adaptive event as one that has the ability to overcome random genetic drift and become 'fixed' in a population. Random mutations tend to remove adaptive events. We tend to think that mutations occur with a relentless, constant frequency. Therefore, an adaptive event is probably more accurately described as one that can overcome mutation and random genetic drift.

A short digression into population genetics is now necessary. A small population is usually characterized by frequent extinctions of some lineages and radiations of other lineages – a phenomenon we think of as random genetic drift. A larger population is a more stable entity, with smaller amounts of random genetic drift. In order for an advantageous mutation to become fixed in a population, it is necessary for it to overcome random genetic drift. Darwin summarized his thoughts on natural selection by saying that "Natural selection is daily and hourly scrutinising, throughout the world, every variation, even the slightest; rejecting that which is bad, preserving and adding up all that is good".

We can digress a little further to talk about a specific example of positive selection. The case of Charles Darwin's Galapagos finches is a nice exemplar of positive selection. Darwin noticed that the diversity of finches on the Galapagos Islands exceeded the diversity of the finches on mainland Ecuador. On the Galapagos Islands there is a particularly high diversity of beak shapes and sizes, with each finch appearing to be perfectly suited to the environment in which it finds itself. On careful consideration, it was decided that this diversity was likely to have been created by natural selection. This seemed to be a bit of a conundrum, given that the island populations were presumably more recent than the mainland populations. The answer appears to be that when morphological changes that affected beak size and shape appeared, it was usually advantageous to retain these changes. This allowed the colonization of a greater diversity of niches.

Our analogy appears to hold true for the analysis of proteins. On occasion, it has been noticed that the rate of amino acid change in some proteins appears to be much higher than expected. Perhaps the most well-known of these incidences relates to the lysozyme proteins in the fore-gut fermenting primates, the colubines (Messier and Stewart 1997).

Methods For Detecting Adaptive Evolution In Protein-Coding Genes.

There are a number of ways in which positive selection can be detected in protein-coding genes. Three of these methods are implemented in Crann.

Historically, the first method of detecting adaptive evolution involved a simple pair-wise comparison of the rate of non-synonymous substitution per non-synonymous site (dn) to the rate of synonymous substitution per synonymous site (ds) (Li, Wu et al. 1985). The basic premise is that the rate of synonymous substitution is effectively equivalent to the mutation rate (or at least the rate at which neutral mutations would become fixed in a population). Therefore, an observation that the non-synonymous substitution rate was higher

than the synonymous substitution rate was taken as an indication that positive selection had occurred since the two sequences last shared a common ancestor.

This method suffers from a few limitations. First of all, even if some pair-wise comparisons indicate that adaptive evolution has occurred, it is often difficult to say with certainty when exactly the adaptive evolution events occurred. From this perspective, simple distance calculations lack precision in pinpointing adaptive events. There is also a problem with the fact that synonymous changes can occur quite quickly and as a result synonymously-variable sites can become saturated for change. This describes a situation where multiple changes at a single site make it difficult to accurately estimate the rate of synonymous substitution. In fairness, this is a limitation of all methods that attempt to analyze adaptive evolution in protein-coding genes.

This distance-based approach was later modified to incorporate a phylogeny (Messier and Stewart 1997). Basically, a phylogenetic tree is constructed that unites a group of sequences. The hypothetical ancestral sequences at each internal node of the phylogenetic tree are then reconstructed. Finally, all pair-wise dn and ds distances are calculated between all sequences, real and hypothetical. This has the desirable effect of allowing greater discriminating power when trying to pinpoint the timing of the adaptive evolutionary event.

A more rigorous approach to estimating dn:ds ratios has been developed more recently (Yang 1998). This method uses maximum likelihood estimation (Felsenstein 1981) of parameters on a phylogenetic tree. Basically, a model of sequence evolution (a phylogenetic tree and a substitution process) is optimized so that it best describes the data (the alignment of protein-coding DNA sequences). Model parameters that can be optimized include the dn:ds ratios for the dataset as a whole or for specific branches or sites in the alignment. This method is quite rigorous and appears to produce good results (Yang 2000; Yang, Nielsen et al. 2000).

An alternative approach to estimating dn and ds ratios has recently been suggested (Creevey and McInerney 2002). This method uses synonymous (or silent) substitutions as an estimate of the neutral substitution pattern and looks at the ways in which non-synonymous (or replacement) substitutions deviates from this pattern. The next section describes this method.

Creevey-McInerney Method:

MacDonald and Kreitman described a method of detecting adaptive evolution in closely-related species based on the neutral mutation hypothesis. This method can be described as a relative rate ratio test. This method seeks to compare the ratio between within-species polymorphism and between-species fixation for silent sites to the same set of ratios for replacement sites. If the replacement sites are evolving in the same way as the silent sites, these two ratios should not be significantly different. If a different selective pressure has been acting on replacement sites compared with silent sites then a significant difference in the two ratios would be observed.

This test provided the starting point for the Creevey-McInerney method. While MacDonald and Kreitman concentrated on the boundary between species and used multiple sequences from within a single species for their test, we have focused on phylogenetic trees. Therefore, although there are similarities between the two methods, the differences are substantial and significant. These trees can be derived from multiple sequences within a single species or can be obtained from multiple species, it doesn't matter.

Background:

The neutral theory states that much of molecular variation is due to the interaction of drift and mutation. Genetic drift has an effect in two ways, firstly as a dispersive force that removes genetic variation from populations (the rate of which is inversely proportional to the size of the population), and secondly in its affect on the probability of survival of new mutations. In fact the retention rate of beneficial mutations is nearly independent of the population size (Gillespie 1998). The dispersive effect of drift is countered by mutation, which puts variation back into populations and these two forces reach an equilibrium that can account for much of the variation seen in genetic data (Gillespie 1998). A fundamental prediction of the neutral theory (Kimura 1983) is that the neutral mutation rate determines both the rate of interspecific divergence and influences the amount of intraspecific polymorphism (Templeton 1996).

Researchers (McDonald and Kreitman 1991; Templeton 1996), have used this prediction to test if in some cases the rate of interspecific and intraspecific divergence differs. McDonald and Kreitman (1991) used the following argument. Assuming there is no recombination, consider a set of alleles from more than one species that are

connected by a phylogenetic tree. This phylogeny can be divided into two parts, between-species (interspecific) and within-species (intraspecific). A mutation located on a between species branch appears in all descendent individuals and is considered a fixed difference between the species, while a mutation on a within species branch will be a polymorphism and be a difference within the population of that species (McDonald and Kreitman 1991). If all mutations in an alignment are considered, they can be classified as between or within species (fixed or polymorphic) and also whether the mutation caused the amino acid to change (replacement), or not (silent). If protein evolution occurs through a neutral process, the ratio of replacement to silent fixed substitutions should be the same as the ratio of replacement to silent polymorphisms (McDonald and Kreitman 1991). However, a substitution that becomes fixed through selection (being beneficial making its retention rate independent of population size) will remain a polymorphism for less time than a substitution that becomes fixed through random drift (its retention rate being inversely proportional to the population size) (McDonald and Kreitman 1991; Gillespie 1998). This means that an adaptive (beneficial) substitution will remain polymorphic in a population for less time than a neutral substitution. When examining a dataset where adaptive evolution has occurred we will observe more fixed replacement substitutions than expected from the neutral hypothesis and because of this, the ratio of replacement to silent fixed substitutions will differ from the ratio of replacement to silent polymorphisms (McDonald and Kreitman 1991). These ratios can be compared using standard statistical tests like the G-test or Fishers exact test (Sokal and Rohlf 1981).

McDonald and Kreitman (1991) analysed the Adh locus in *Drosophila* using this method. They used sequences from 30 individuals from three species of *Drosophila*, twelve from *D. melanogaster*, twelve from *D. yakuba*, and six from *D. simulans*. A consensus sequence was constructed from the 30 sequences, and all nucleotide differences from the consensus were examined. If any nucleotide difference was contained only in one species, and was in every individual of that species, it was considered a fixed substitution, otherwise it was considered a polymorphism. The nucleotide difference was also categorised according to whether it was a replacement or silent change. They found 7:2 as the ratio of fixed to polymorphic replacement substitutions and 17:42 as the ratio of fixed to polymorphic silent substitutions. When examined for independence these ratios were found to be significantly different (with a p-value of

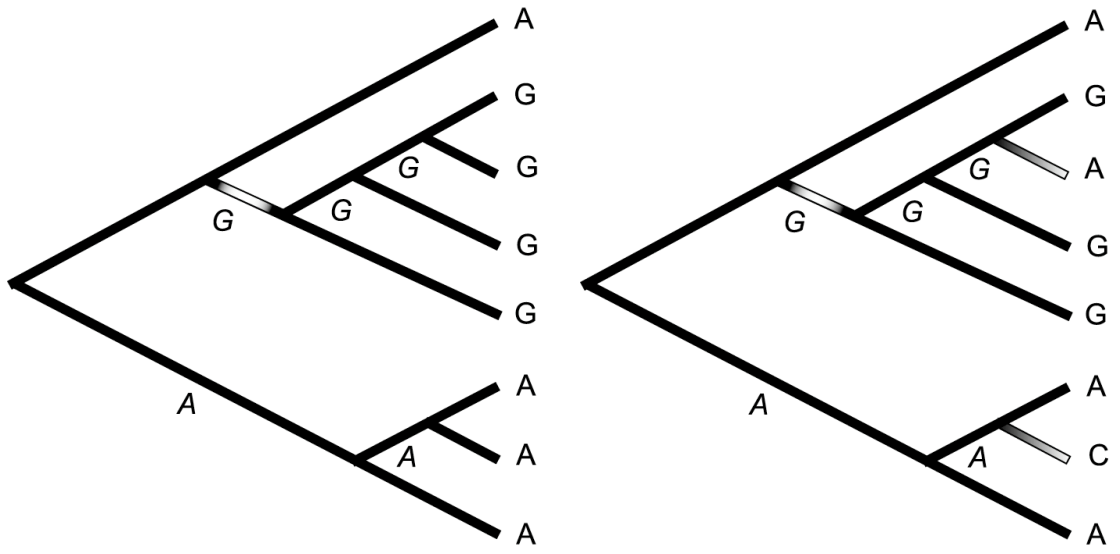
0.006). This evidence was compelling enough for them to sugest that the method was valid and accurate.
There was immediate criticism of their approach (Graur and Li 1991; Whittam and Nei 1991). It was thought that the method would underestimate the between species diversity, and that the classification of polymorphic sites would be sensitive to the number of sequences sampled. However, McDonald and Kreitman (1991) pointed out that the underestimation of the species diversity and the problems associated with the classification of polymorphic sites would effect the neutral replacement substitutions and neutral silent substitutions equally and therefore not affect the validity of the test (McDonald and Kreitman 1991). The method of classifying whether a subsitution was replacement or silent was also criticised (Whittam and Nei 1991), but it was pointed out that any method of classifying the substitutions into fixed or polymorphic substitutions could be used and would still be a valid test of the neutral theory, as long as the same classification system was used for both replacement and silent substitutions (McDonald and Kreitman 1991).

Maynard Smith (Smith 1970), was the first to propose the use of this predicted relationship between interspecific divergence and intraspecific polymorphism to test the neutral hypothesis. Other researchers (Kellogg and Appels 1995; Templeton 1996) have since adapted the test to analyse their own data. In fact since there is no limit to the number of classifications that can be made, studies have been made which subdivide the types of mutations into further categories, enhancing the test (Templeton 1996). Templeton (1996) divided the substitutions into external (tip) *vs.* internal (interior) mutations according to whether the mutations appeared only in some individuals or in whole populations.

Relative Rate ratio test (Creevey and McInerney 2002):

Our method is a relative rate ratio analysis and proceeds as follows. For any dataset, a phylogenetic tree is constructed and assumed to be correct. This tree is rooted by reference to an appropriate out-group. This effectively converts the data to polarised character types. Hypothetical ancestral sequences are reconstructed at each internal branch through a method that uses the principal of maximum parsimony (Hennig 1966) applied at the codon level. However since ambiguities are uninformative, we construct for each dataset substitution matrices based on the types of changes we observe in the dataset at 0, 2 and 4-fold degenerate positions. For any ambiguity between two (or more) codons we can then calculate how often we

would expect to see each codon, based on the summed changes we observe in the candidates at each 0, 2 and 4-fold degenerate site present in the codons. This method only works reliably when there is no ambiguity at the root, however maximum parsimony cannot guarantee this, so in this case the ancestral codon is assigned the same as the ancestor of the outgroup. This ensures that there are no ambiguities in the ancestral reconstruction. However in principle any method that accurately reconstructs ancestral character states may be used.



(a) (b)

Figure:(a) An Invariable substitution is defined as a substitution that occurs on an internal branch and is subsequently retained in all the descendant alleles. The hypothesised ancestral states at each internal branch are shown in italic. The white internal branch represents where a substitution occurred (A to G) and subsequently remained invariable within that clade.
(b) A substitution that occurs at an internal branch but subsequently changes elsewhere in the clade defined by that internal branch is variable. Variable substitutions may also be one that occurs in a single. Here the white branches represent where variable substitutions occurred. The hypothesised ancestral states at each internal branch are shown in italic.

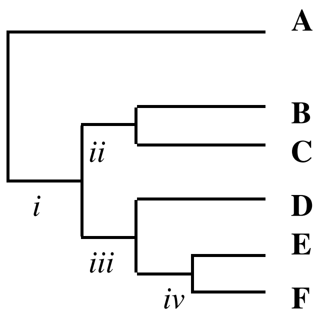
Using our reconstructed phylogeny, we identify all substitutions that occur across the tree and determine whether they result in a non-synonymous or synonymous codon change. We then consider each internal branch of the tree and count the changes in the descendent clade described by that internal branch. This results in four values representing those different types of substitutions that have occurred from that internal branch to the tips. The four types of substitutions are classified as replacement-invariable (RI, those replacement substitutions where the new character-state is preserved in all subsequent lineages), replacement-variable (RV, where the replacement substitution is not preserved in all lineages and has changed at least once more in a subsequent lineage), silent-invariable (SI, silent changes that were not observed to have changed again) and silent-variable (SV, silent changes that were observed to have changed again in a subsequent lineage). Using the example in the following box, the following steps are taken in calculating the values.

Step 1: Count changes that have occurred within taxa E and F and ancestor iv
Result iv: RI = 1, RV = 1, SI = 1, SV = 1

Step 2: Count changes that have occurred with taxa D, E and F and ancestor iii +result iv.
Result iii: RI = 1, RV = 2, SI = 2, SV = 1

Step 3: Count the changes that have occurred within taxa B and E and ancestor ii.
Result ii: RI = 0, RV = 3, SI = 0, SV = 2

Step 4: Count changes that have occurred within taxa B, C, D, E and F and in ancestor i + result iii + result ii.
Result i: RI = 1, RV = 5, SI = 2, SV = 3.



A	A T G	C C T	C G T	T T T	T G C	G A G
i	- - -	- - -	- - -	- - -	- - -	- - -
ii	- - -	- - -	- - -	- - -	- - -	- - -
B	G - -	- T C	- - -	- - -	- - -	- - -
C	- - -	- - -	G - -	- - -	- - T	- - -
iii	- - -	- - -	- - -	- - C	- - -	- - T
D	- - -	- - -	- - -	- - C	- - -	- - T
iv	- - -	- - C	- A -	- - C	- - -	- - T
E	- - -	- - C	- A -	- - C	- - G	- - -
F	- - -	- - C	- A -	- - C	- - -	- - T

Node	***Repl. Invar***[a]	***Repl. Var.***[b]	***Silent Invar***[c]	***Silent Var.***[d]
iv	1	1	1	1
iii	1	2	2	1
ii	0	3	0	2
i	1	5	2	3

Box: Substitutions counted across a phylogenetic tree. For each internal branch, we have a count of the number and type of changes that have occurred within the clade defined by that internal branch. The Table illustrates an example data set (A to F) and the reconstructed ancestors (i to iv). [a]Replacement invariable substitutions. [b]Replacement variable substitutions. [c]Silent invariable substitutions. [d]Silent variable substitutions.

Using a G-test (or Fisher's exact test (Sokal and Rohlf 1981), when numbers are small) we can compare the ratio of RI substitutions to RV substitutions, with the expectation that this ratio is the same as the ratio of SI to SV substitutions (McDonald and Kreitman 1991). The ratio of SI to SV is the expected value under the neutral model, since they represent neutrally evolving sites (Kimura 1983), and the ratio of RI to RV substitutions would be the same as SI to SV under neutrality (McDonald and Kreitman 1991). In the event of a significant difference in these ratios, it is possible to analyse the result to find out whether there are high numbers of RI substitutions or RV substitutions. The former is indicative of directional selection and the latter is indicative of non-directional selection. Positive selection favours any substitutions that confer an advantage. During an episode of positive directional selection, these advantageous substitutions will occur and subsequently remain invariable in all descendant lineages, at a rate significantly higher than expected from the neutral model (McDonald and Kreitman 1991). Under non-directional selection the advantageous substitutions will become variable at a rate significantly higher than expected from the neutral model.

Neutral substitution rate test

Neutral mutations acting on any random genetic sequence would result in about 3 times more replacement substitutions than silent substitutions. However due to factors like base composition bias, the transition to transverion ratio and codon usage for any particular dataset, the ratio of replacement to silent changes can differ. We can calculate the expected ratio of replacement to silent changes of any dataset, by basing them on the total number of replacement or silent sites. The total number of replacement or silent sites in each sequence within each clade is calculated as per the method of Li (Li 1993).This ratio is then used to calculate the expected number of replacement and silent substitutions within each clade. We consider whether the number of replacement substitutions or silent substitutions observed were significantly greater than expected from the neutral model. Using a G-test (or Fisher's exact test (Sokal and Rohlf 1981)), we can determine if any type of substitution occurred more often than expected from neutrality. With this information if there is no significant result using the first test, then we can determine if negative selection was acting or whether the substitutions were appearing at a rate expected under neutrality.

Installing and running Crann

Crann is available to download at the website of the Bioinformatics and Pharmacogenomics laboratory at NUI Maynooth, Ireland.
http://bioinf.may.ie/crann/

At the moment there are binary executables for four different Operating systems:
Apple Macintosh OS 8.x and 9.x
Apple Macintosh OS X (10.1 and 10.2)
Redhat Linux
Microsoft Windows PC.

Downloading:
When you click on the link for whatever operating system you are using, the appropriate version of Crann is saved to your computer's hard-drive. This is compressed to increase the download speed. In Microsoft Windows, and both Apple operating systems this file should be automatically uncompressed leaving you with the program ready to run. However, if this does not occur in Microsoft windows use the program 'winzip' (http://www.winzip.com) to uncompress Crann, or in either Apple Macintosh operating system use 'stuffit-expander' (http://www.aladdinsys.com).
In Redhat linux, issue the command "*gunzip crann_redhat.gz*" to uncompress the executable.

Note:
The versions of Crann for Redhat linux and Mac OS X (10.1 and 10.2) are command line programs. To run crann in either of these operating systems it is necessary to use a terminal window or shell. It may also be necessary in these operating systems to change the permissions to allow crann to be run for the first time. This can be achieved by typing the command "*chmod a+x crann*".

Running Crann:
In Microsoft windows and Mac Os 8.x and 9.x double clicking on the icon associated with Crann will run the program. In these operating systems, the actual Crann program **must** be located in the same directory as the input files, an alias or shortcut to Crann will not suffice.
In Mac OS X and Redhat linux, the Crann program may either be located in the same directory as the input files, or somewhere on your path (like *~/bin/* or */usr/local/bin*). If you do not know which directories are on your path, ask your systems administrator. To run

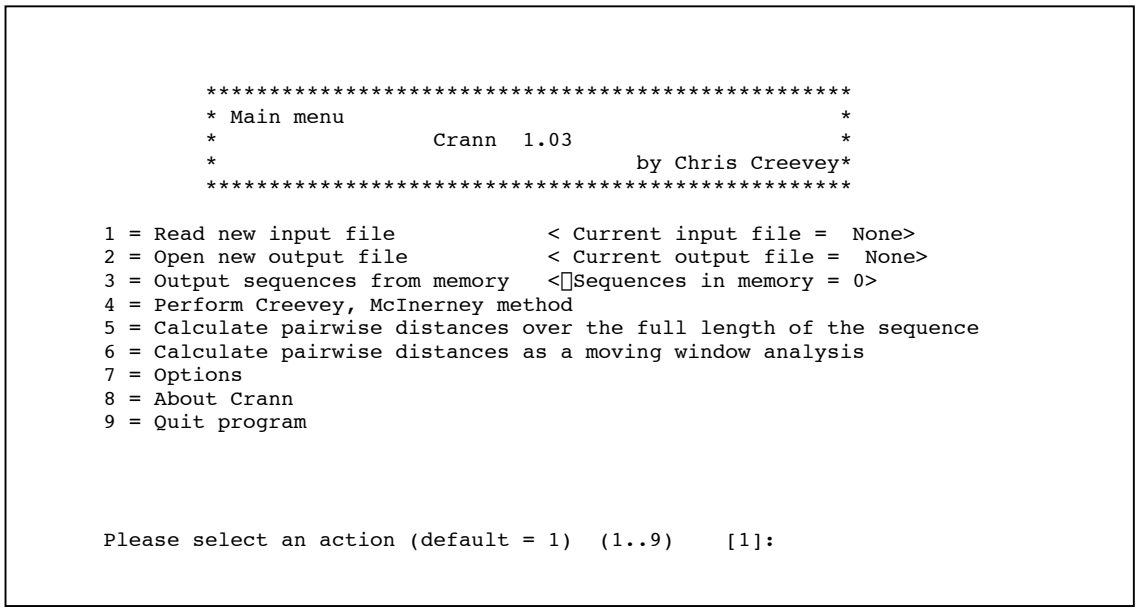
Crann in these operating systems, type the command "*./crann*" or "*crann*".

Input files.
Crann requires as input aligned homoligous sequences in fasta format. These sequences must all be contained in a single file. See the section on option 1 of the main menu for more details.
It is also possible to input a phylogeny for Crann to use in its execution. In this case the tree must be in phylip format without any branch lengths or distances included. See the section on option 7 of the general options menu for more details.

Output files.
Crann produces several output files, only one of which the user can specify the name. If Crann is run twice in the same directory, it will completely overwrite the results of the first analysis. Unless this is the desired effect, we advise users to use separate directories for each dataset being analysed. For more details on the output files see the section detailing the contents of each.

The main menu.

Upon starting Crann the user is presented with the main menu. This menu specifies the various functions that can be carried out using Crann. As each version is produced, its version number is displayed here at the top. Crann is a command driven program so when prompted to select any options in this or any other menu, it is necessary to enter the number associated with the option using the keyboard.



In advance of any analysis, it is necessary to specify the input and output files. To this end Crann will not carry out any analyses until the user has firstly specified the input file, and secondly the output file. Any problems with the files will be reported to the screen.

Main menu, option 1: Read new input file.

When the user selects to read a new input file, the program asks for a file name. The program will not be able to find the file if it is not contained in the same directory from which the program was called. Immediately the following is carried out on the alignment:

1. A check to see if the file is in FASTA format
2. A check to see if all the sequences are of the same length (and hence aligned)
3. A search of each sequence for stop codons based on the genetic code selected (see the general options menu). The default genetic code is the universal genetic code.

4. A search of each sequence for any non-standard characters. (A non standard character is anything other than ‘A’, ‘a’, ‘T’, ‘t’, ‘C’, ‘c’, ‘G’, ‘g’, ‘U’, ‘u’, or ‘-‘).
5. A summaryof the sequences contained in the input file is printed to screen:

```
//*****************************************************//
                  Input data summary

 Number of sequences in memory = 13
 genetic code = Universal genetic code

 Number of Non-standard characters = 0


 Stop codons were found in the following sequences

3 in 'Dog'
At positions:'168' '185' '278'
3 in 'Human'
At positions:'168' '185' '278'
3 in 'Cat'
At positions:'168' '185' '278'
3 in 'Mouse'
At positions:'168' '185' '278'
3 in 'Fugu'
At positions:'168' '185' '278'
1 in 'gorilla'
At positions:'191'
1 in 'cow'
At positions:'177'
1 in 'chimp'
At positions:'177'

 Stop codons will be excluded using pairwise deletion


//*****************************************************//
Press return to continue
```

This report details:

i) The number of sequences read into memory
ii) The selected genetic code
iii) The number of non-standard characters, and their positions in each sequence.
iv) The number of stop codons and their positions in each sequence (except if there is one in the final position, which is ignored).

The user must press the ‘return’ (or ‘□’ or ‘enter’) key to continue. The program then returns to the main menu, however now the name of the input file is specified alongside option 1. The number of sequences in memory is specified alongside option 3.

Main menu, option 2: Open new output file:

After selecting to open a new output file, the user is prompted to enter the name of the file and press return. The program will not be able to find the file if it is not contained in the same directory from which the program was called. Any errors opening the output file will be reported at this point. The program then returns to the main menu, however now the name of the input file is specified alongside option 2.

```
            ************************************************
            * Main menu                                    *
            *                 Crann  1.03                  *
            *                                 by Chris Creevey*
            ************************************************

      1 = Read new input file            < Current input file =  ancestor.out>
      2 = Open new output file           < Current output file =  outfile.txt>
      3 = Output sequences from memory   <# sequences in memory = 25>
      4 = Perform Creevey, McInerney method
      5 = Calculate pairwise distances over the full length of the sequence
      6 = Calculate pairwise distances as a moving window analysis
      7 = Options
      8 = About Crann
      9 = Quit program



     Please select an action (default = 4)  (1..9)    [4]:
```

Main menu, option 3: Output sequences from memory.

Choosing option 3, prints the sequences from memory to the specified output file (option 2). When this option is selected the user is prompted to choose the format in which the sequences are to be written.

```
Please select the output file format:
       1 = Codon numbers
       2 = Amino acid format
       3 = DNA sequences in tab delimited format (in columns)
       4 = Return to the main menu

        please select 1, 2, 3 or 4:  (1..4)    [4]:
```

Main menu, option 4: Perform Creevey, McInerney Method.

This option performs an analysis of the alignment as described by Creevey & Mcinerney (2002). When this option is selected, Crann carries out the following procedures:

1. All pairwise Dn and Ds distances are calculated for the dataset. By default, the method described by Li (1993) is used to calculate the distances, however the method described by Li, Wu

and Luo (1985) may also be used (see the general options menu). These distances are written to the following files; the Dn distances to 'Dn.dis', the Ds distances to 'Ds.dis' and the value of Dn/Ds to 'DnDs.dis'. All the Dn/Ds results are examined and a summary is written to the main output file (option 2, main menu). This summary details the number of Dn/Ds results that were greater, less than and equal to 1.

2. A hypothesis of relationships of all the sequences (a phylogeny) is created in memory. By default, this is created using the Neighbor Joining method using the Dn distances. The user may instead choose to use Ds or Dn/Ds distances or supply their own phylogeny instead (see general options menu), but this must be done prior to performing this analysis.

3. The user is requested to choose the outgroup(s) in the dataset. It is imperative to include outgroup sequence(s) as this will allow the phylogeny to be rooted so that directionality can be inferred and ancestral sequences can be reconstructed. At the prompt the user must enter the number of each sequence that is part of the outgoup, pressing 'return' after each one. As an outgroup is chosen, it is removed from the list of sequences displayed to choose from. When all the outgroups have been chosen, the user must enter the number zero (0) to indicate that they have finished their selection.

```
In order to give the tree directionality, please define those sequences which form the
outgroup

Sequence numbers, and names as follows
No: 1 Name:  Human
No: 2 Name:  Mouse
No: 3 Name:  chimp
No: 4 Name:  gorilla
No: 5 Name:  dog
No: 6 Name:  cat
No: 7 Name:  cow
No: 8 Name:  horse
No: 9 Name:  sheep
No: 10 Name: goat
No: 11 Name: fox
No: 12 Name: fugu

Please select the number of a sequence belonging to the outgroup, and press return
ter 0 when finished
 (0..25)    [0]:
```

4. Crann then performs a check on the chosen outgroup to verify their validity. If there is any problem with the selection the user is notified and asked to re-choose the outgroups or use the first (or last) sequence in memory as the outgroup.

5. Using the phylogeny, all the ancestral sequences are reconstructed at each internal node. A Maximum parsiomony method implemented at the codon level is used to carry this out. Ambiguities are solved using distance matrices constructed from the dataset for the 0, 2 and 4-fold degenerate sites. The ancestral sequences reconstructed and the sequences from the input file are all written to the file ‘ancestor.out’. The ancestral sequences are named according to the label attached to that node on the tree (see the section on output files, specifically ‘resulttree.ph’).

6. The relative rate ratio test and the neutral substitutions test as described in Creevey and McInerney (2002) are carried out on the data in the phylogeny. The results of the analyses are written to the main output file for the relative rate ratio test (option 2 in the main menu), and also in the phylip formatted file ‘Resulttree.ph’. The results of the neutral substitution test are written to the file 'substitutions.out' (see the section on this file for more details).

7. At this point Crann also carries out an analysis based on the method described by Messier and Stewart (19xx). This method calculates the Dn and Ds values of each pairwise comparison of

every reconstructed ancestral sequence to each of their direct descendents. The results of this are written to the file 'yadf.out' (Yet Another Distance File). The internal branch (node) numbers listed in this file relate to the label given to each internal node of the phylogeny (See the section on the output files, 'yadf.out' and 'Resulttree.ph' for more details).

On completion, Crann returns to the main menu.

Main menu, option 5: Calculate the pairwise distances over the full length of the sequence.

This option calculates the value of Dn and Ds for every possible pairwise comparison between all the sequences contained in the input file (option 1, main menu). A single Dn or Ds value is calculated for every pairwise comparison, representing the mean value of Dn or Ds across the entire length of the sequence. The algorithm described by Li (1993) is used by default to calculate these values but the algorithm described by Li, wu and Luo (1985) may also be used. The results of Dn are written to the file 'Dn.dis', Ds results are written to 'Ds.dis', and the value of Dn/Ds is written to the file 'DnDs.dis' (See the section on these files for more details).

Main menu, option 6: Calculate pairwise distances as a moving window analysis.

This option carries out a moving window analysis of Dn and Ds along the length of the sequences. The algorithm described by Li (1993) is used by default but the algorithm used by Li, Wu and Luo (1985) may also be used. When this option is selected the user is asked if they would like to use all the sequences in the calculation or select those sequences not to be included

```
Would you like to compare all sequences in memory,(1) (default)
 Or would you like to select those sequences not to be included (2) ?

Please select 1 or 2 (1..2)    [1]:
```

Next the user is requested to specify the size of window to be used in the analysis.

```
Please select the window size for analysis (in codons)
( or Press return for 10% of the total length) (1..365)    [36]:
```

This is the size of the section of the sequences that will be used to calculate the pair-wise distances (the size is specified in codons). It may be necessary to try different sizes of windows because if the window is too small, it may not be possible to calculate the Dn and Ds values and 'NaN' values will appear in results (The word 'NaN' represents where a calculation was not possible). Next the user is asked to specify how far the window is to be shifted down the sequence for each calculation.

```
Please select the shift size (in codons)
( or Press return for 50% of window size ) (1..36)    [18]:
```

This may be of any size from 1 codon to that specified as the window size. This ensures that every part of the sequence is analysed. Finally the user is requested to choose either the algorithm described by Li (1993) or by Li, Wu and Luo (1985) to calculate the distances.

```
Which method would you like to use?
        1 = 1985 method
        2 = 1993 method

        Please choose either 1 or 2 (1..2)    [2]:
```

The default method is the 1993 algorithm. Crann then calculates the Dn and Ds vlaues as a moving window analysis and writes them to the main output file (see the section on this file for more details). Once complete, Crann returns to the main menu.

Main menu, option 7: General options.

Selection of this option displays the general options menu. This is where some of the defaults in Crann may be changed to suit a particular analysis. Crann may carry out analyses without anything being changed here and the defaults will be used.

```
               ************************************************
               * General Options Menu                         *
               *                 Crann  1.03                  *
               *                                 by Chris Creevey*
               ************************************************

        1 = Genetic Code is Universal genetic code
        2 = Deletion of Non-Standard Characters = Pairwise
        3 = Deletion of Stop Codons  = Pairwise
        4 = Analyse all sequences in memory? = Yes
        5 = Analyse whole sequence length? = Yes
        6 = Which Li method to use? = 1993
        7 = Input Phylogentic tree? = No
        8 = Build a neighbour joining tree with which distances? = Dn

        0 = Return to main menu (default)


Enter a number to change that option, or press return to continue
 (0..9)    [0]:
```

General options 1: Genetic Code.

Selecting general option 1 allows the user to change the genetic code used for all the calculations carried out by Crann (including the relative rate ratio test and the Dn and Ds calculations). Upon selecting option 1 the user is asked to choose the genetic code being used in the input file (option 1, main menu).

```
Please specify which genetic code the input file contains:

                1 = Universal (default)
                2 = Vertebrate standard mitochondrial
                3 = Yeast mitochondrial
                4 = Mycoplasma/Spiroplasma/Mold/Protozoan/Coelenterate
                5 = Invertebrate mitochondrial
                6 = Ciliate
                7 = Echinoderm mitochondrial
                8 = Euplotid
                9 = Bacterial (same as universal)
                10= Alternative Yeast Nuclear
                11= Ascidian mitochondrial
                12= Flatworm mitochondrial
                13= Blepharisma Nuclear


        please select one of 1 to 13 from above:  (1..13)    [1]:
```

Once the user has selected the genetic code being used, Crann re-examines the sequences using the selected genetic code and displays a summary. This is the same type of summary as was calculated when the input file was first specified (option 1, main menu), and details the number and position of stop codons and non-standard characters.
Crann then returns to the general options menu.

General options 2: Deletion of non-standard characters

This option is only valid for the calculation of Dn and Ds values. It determines whether the position of a non-standard character is ignored in the entire dataset (even if a standard character is present in that position on another sequence) or if a non-standard character is to be ignored only when part of a pairwise comparison.

```
In these cases would you prefer to:
A = Use pairwise deletion
B = Use complete deletion

Please select from option A or B:
```

By default, pairwise deletion is initially chosen.

General options 3: Deletion of stop codons.

This option only applies to the calculation of Dn and Ds values. Crann will carry out an analysis when there are stop codons present in the sequences. It is up to the user to decide if such an analysis makes biological sense. As with non-standard characters the position of a stop codon may be excluded in every sequence during an analysis (complete deletion) or only if the stop codon is part of a pairwise comparison (pairwise deletion). By default, pairwise deletion of stop codons is the initial state.

General options 4: Analyse all sequences in memory?

This option allows the user to specify certain sequences that are not to be used during any of the analyses (relative rate ratio tests and Dn and Ds calculations). After choosing this option the user is asked to make sure that they want to remove some sequences.

```
Would you like to compare all sequences in memory,(1) (default)
 Or would you like to select those sequences not to be included (2) ?

Please select 1 or 2 (1..2)    [1]:
```

If the user confirms that they wish to select sequences not to be included (2) a list of all the sequences in memory are displayed to the screen. The user must enter the numbers of the sequences not to be included, pressing return after each one. When the selection of the sequences is complete, entering zero will return the user to the general options menu.

General options 5: analyse whole sequence length?

This option allows the user to specify a certain region of the alignment to analyse. The rest of the sequences length is ignored during the analyses. When selected the user is asked to specify the start codon position of the analysis. The default is the start of the alignment.

```
Please select the desired start postition for analysis of the sequences (in codons)
        (Press return for the begining of the sequences) (0..365)    [0]:
```

Next the user is asked to specify the end codon position for analysis. The default is the last codon in the alignment.

```
Please select the desired end postition for analysis of the sequences (in codons)
        (Press return for the end of the sequences) (0..365)    [365]:
```

The user is then returned to the general options menu.

General options 6: Which Li method to use?

This option specifies the algorithm used to calculate the values of Dn and Ds for the relative rate ratio test. The choices are either the Li (1993) method or the Li wu and Luo (1985) method. By default the Li's (1993) method is the initial setting. The user is then returned to the general options menu.

```
Which method would you like to use?
1 = 1985 method
2 = 1993 method

Please choose either 1 or 2 (1..2)    [2]:
```

General options 7: Input phylogenetic tree?

This option lets the user specify the phylogenetic tree to be used during an analysis. By default Crann will create a tree using the neighbor joining algorithm from the Dn distances calculated. However if the user has created a phylogeny for the dataset (option 1, main menu) this may be used as the phylogeny in any further analyses. Upon selecting this option the user is asked how they would like to create a phylogeny.

```
      Phylogenetic tree:
              Would you like to:
              1 = read in a tree file in nested parentheses format?
              2 = create a tree using the neighbour joining algorithm?

      Please choose 1 or 2    Please choose 1 or 2:
```

If the user chooses to read in a tree file in nested parentheses (phylip) format (1) Crann asks for the name of the file. This file must be contained in the same directory as the fasta formatted input file, and must not contain any labels or branch lengths. The labels of each taxa on the tree must be unique and exactly the same as the taxa labels in the fasta formatted input file. It is possible to use a truncated version of the names on the tree, but the first part of the names must be the same as those in the input file. Once the tree has been read into memory the user is returned to the general options menu.

General options 8: Build a neighbor joining tree with which distances?

This option is only displayed if the user has not read a phylogenetic tree into memory (option 7, general options menu). It allows the user to choose which distance matrix to use to create the neighbor joining tree.

```
      Which distances would you like to use in the algorithm
      1 = Dn values
      2 = Ds values
      3 = Rate of evolution (Dn/Ds)

      Please choose either 1, 2 or 3
(1..3)    [1]:
```

By default Dn distances are those choosen initally, however the user also has the choice of using Ds or Dn/Ds distances (As found in the files 'Dn.dis', 'Ds.dis' and 'DnDs.dis'). The user is then returned to the general options menu.

General options 0: Return to main menu.

This option is always the default when the user is at the general options page. Choosing this returns the user to the main menu.

Main Menu, option 8: About Crann.

Choosing this option prints a splash page to the screen with the details of the version and compilation date of the version of Crann being used. Author contact information is also provided.

Main menu, option 9: Quit program.

Choosing this option will cause Crann to terminate properly. This is the only way a user should exit from Crann, otherwise information may not be written to some output files and memory on the computer may not be deallocated properly.

Output files:

The results in each of the output files described here are those calculated from the lysozyme dataset as described in Creevey and McInerney (2002). The following steps were taken:

When Crann was started, Option 1 of the Main menu was selected and at the prompt the name of the file 'lyso2.seq' was entered. This contains the fasta formatted lysozyme sequences that were to be examined, i.e.:

```
>common marmoset CJU76923   447
atgaaggttctcattattctggggctggtcctcctttctgtcatggtccagggcaaggtctttgaaaggtgtgagttggccagaactctgaaaa
ggtttggactggatggctacaggggaatcagcctagcaaactggatgtgtttggccaaatgggagagtgattataacacacgtgctaca
aactacaatcctggagaccaaagcactgattatgggatatttcagatcaatagccactattggtgtaacaatggcagaaccccaggagc
agttaatgcctgtcatatatcctgcaatgctttgctgcaagatgacatcactgaagctgtggcctgtgcaaagagggttgtccgcgatccac
aaggcattagggcatgggtggcatggaaagctcattgtcaaaacagagatgtcagtcagtatgttcaaggttgtggagtataa
>cotton-top tamarin SOU76922  447
atgaaggttctcattcttctggggcttgtcctcctttctgtcatggtacagggcaaggtcttcgaaaggtgtgagttggccagaactctgaaaa
gactt…….etc
```

Option 2 of the main menu was then selected and the output file name was specified, i.e. 'output.txt'.

Next option 4 was selected to run the relative rate ratio test as described in Creevey and McInerney (2002).
Since we did not change anything in the general options menu (option 7, main menu), by default Crann calculated the values of Dn, Ds, and Dn/Ds for each pairwise comparison and used the Dn distances to build a phylogeny for the dataset.
Upon being asked to choose the outgroup sequences the new-world monkeys were identified (Common marmoset, Cotton-top tamarin and the Squirrel monkey, sequence numbers 1, 2 and 3 respectively).
Crann returned to the main menu and option 9 was selected to quit.

The following output files were produced:

Main output file (outputfile.txt)

This name of this file is specified by the user (option 2, main menu) and contains a summary of all the results calculated.

Results of the relative rate ratio test (option 4, main menu)

Upon completion of the relative rate ratio test the following are contained in the main output file:

At the top of the file is the following:

```
-------------------------------------------------------

Results from input file: lyso2.seq


 Totals for Dn/Ds:
        Greater than 1:         198
        Less than 1:            78
        Equal to 1:             0
```

This is a summary of all the Dn/Ds calculations for each pair-wise comparison between all the sequences in the input file (in this case 'lyso2.seq'). There are no statistics attached to this calculation, however it does give an indication of the level of positive selection acting in the dataset (the number of Dn/Ds ratios greater than 1).

Next in the file is a summary of the phylogenetic relationships calculated using the neighbor joining method along with a confirmation of the which distances were used to construct it.

After that the substitution matrices calculated at zero, two and four-fold degenerate site are displayed. These were used to solve ambiguities in the reconstruction of the ancestral sequences.

```
Nucleotide substitution matrices

0-fold degenerate sites
From: U 0.993970 0.006030 0.000000 0.000000 0.000000
From: C 0.010394 0.949296 0.020704 0.019605 0.000000
From: A 0.002396 0.011041 0.931231 0.055332 0.000000
From: G 0.000000 0.009721 0.049971 0.940307 0.000000
From: X 0.000000 0.000000 0.000000 0.000000 0.000000
```

2-fold degenerate sites
From: U 0.956915 0.029772 0.009999 0.003314 0.000000
From: C 0.053512 0.914942 0.027208 0.004339 0.000000
From: A 0.009704 0.023689 0.958329 0.008277 0.000000
From: G 0.015770 0.000000 0.003390 0.980840 0.000000
From: X 0.000000 0.000000 0.000000 0.000000 0.000000

4-fold degenerate sites
From: U 0.954963 0.030140 0.006756 0.008141 0.000000
From: C 0.021606 0.966705 0.002479 0.009209 0.000000
From: A 0.003305 0.003390 0.968133 0.025172 0.000000
From: G 0.007380 0.014279 0.049414 0.928927 0.000000
From: X 0.000000 0.000000 0.000000 0.000000 0.000000

Finally, the results from the relative rate ratio test as described in Creevey and McInerney (2002) are shown for each internal branch:

Results of Creevey, McInerney Test for each internal branch:

no	RI	RV	SI	SV			
branch 0	0	3	0	0	G = 0.000000	Gtest:1.000000 > pvalue > 0.990000	Fishers: p = 1.000000
branch 1	0	3	0	1	G = 0.000000	Gtest:1.000000 > pvalue > 0.990000	Fishers: p = 1.000000
branch 2	1	3	0	1	G = 0.263371	Gtest:0.900000 > pvalue > 0.500000	Fishers: p = 0.800000
branch 3	2	4	0	2	G = 0.976987	Gtest:0.500000 > pvalue > 0.200000	Fishers: p = 0.535714
branch 4	0	0	0	3	G = 0.000000	Gtest:1.000000 > pvalue > 0.990000	Fishers: p = 1.000000
branch 5	5	4	1	5	G = 2.175876	Gtest:0.200000 > pvalue > 0.100000	Fishers: p = 0.167832
branch 6	3	0	1	0	G = 0.000000	Gtest:1.000000 > pvalue > 0.990000	Fishers: p = 1.000000
branch 7	1	0	0	0	G = 0.000000	Gtest:1.000000 > pvalue > 0.990000	Fishers: p = 1.000000
branch 8	1	1	0	3	G = 1.435777	Gtest:0.500000 > pvalue > 0.200000	Fishers: p = 0.400000
branch 9	1	0	1	0	G = 0.000000	Gtest:1.000000 > pvalue > 0.990000	Fishers: p = 1.000000
branch 10	3	1	1	3	G = 1.762520	Gtest:0.200000 > pvalue > 0.100000	Fishers: p = 0.242857
branch 11	6	1	2	5	G = 4.507650	(Gtest:0.050000 > pvalue > 0.025000)	Fishers: p = 0.051282 *
branch 12	16	3	4	5	G = 4.213014	Gtest:0.050000 > pvalue > 0.025000	Fishers: p = 0.044127 *
branch 13	24	7	5	10	G = 8.048488	Gtest:0.005000 > pvalue > 0.000000	Fishers: incalculable *
branch 14	0	0	0	0	G = NaN	Gtest:1.000000 > pvalue > 0.990000	Fishers: p = 1.000000
branch 15	1	1	0	3	G = 1.435777	Gtest:0.500000 > pvalue > 0.200000	Fishers: p = 0.400000
branch 16	3	1	0	3	G = 4.127700	(Gtest:0.050000 > pvalue > 0.025000)	Fishers: p = 0.114286 *
branch 17	3	5	0	5	G = 2.903980	Gtest:0.100000 > pvalue > 0.050000	Fishers: p = 0.195804
branch 18	11	9	2	5	G = 1.383584	Gtest:0.500000 > pvalue > 0.200000	Fishers: p = 0.223671
branch 19	43	20	11	18	G = 7.352645	Gtest:0.005000 > pvalue > 0.000000	Fishers: incalculable *
branch 20	1	5	1	4	G = 0.016080	Gtest:0.900000 > pvalue > 0.500000	Fishers: p = 0.727273
branch 21	1	5	1	9	G = 0.116076	Gtest:0.900000 > pvalue > 0.500000	Fishers: p = 0.625000

This specifies the number of Replacement Invariable (RI), Replacement Variable (RV), Silent Invariable (SI), and Silent Variable (SV) substitutions within each clade defined by each internal branch in the phylogeny. The branch numbers refer to the label given to each internal branch of the phylogeny as displayed in the file 'result_tree.ph'.

The results of a G test for independence are displayed next, firstly with the actual G value calculated and then an approximation of is associated p value. The G test for independence is similar to the □-square test, and uses the □-square distribution to calculate the p-values. However as with the □-square test, if any of the values being tested is less than 5 its results are statistically unreliable. This is where Fishers Exact Test is used. This can be used to test for independence where some of the values are less than 5. For this purpose, along side the G-test analysis for every internal branch there is also the results of a fishers exact test analysis. The problem with the fishers exact test is that it becomes very computationally intensive when testing large values, and in these cases the word 'incalculable' appears where otherwise a p-value result would appear.

So which results should you use?
If the fishers exact test produces a p-value (not the word 'incalculable'), this should be taken as the 'true' p-value. The result from the G-test should only be used when the fishers exact test fails to produce a result.

This means that in some cases the G-test may flag a result as being significant, while the fishers exact test may say that the p-value fails to reach the critical level (<0.05). In these cases Crann places parentheses around the G-test result to signify that its results are unreliable due to the size of the values being tested. An example of this can be seen above at internal branches 11 and 16.

The user needs to look out for an occurrence of this as Crann will mark ANY significant results with an astrix (*) or a percentage sign (%) to the right of any line containing them. This is to allow the user to quickly spot a significancant result immediately. The astrix (*) represents where there are far more Replacement Invariable (RI) substitutions than would be expected from neutrality (and is evidence of positive directional selection). The percentage sign (%) represents where there are far more Replacement variable (RV) substitutions than would be expected from neutrality (and is evidence of positive non-directional selection).

The file 'result_tree.ph' contains this information in graphical form by placing labels at the internal nodes that showed a significant result. This tree can be viewed in a phylogeny-viewing program such as "Treeview".

Moving window analysis results:

If the user chooses to perform a moving window analysis using Dn and Ds calculations (option6, main menu) something like the following is written to the file:

Accumulative Ds values

0-36	3.975320
18-54	2.507325
36-72	2.257134
54-90	2.341436
72-108	5.277568
90-126	7.179453
108-144	4.232939
126-162	2.684290
144-180	3.816538
162-198	5.352417
180-216	5.698335
198-234	7.224224
216-252	9.417825
234-270	7.555713
252-288	5.423223
270-306	6.388596
288-324	5.308683
306-342	3.023236
324-360	5.850939

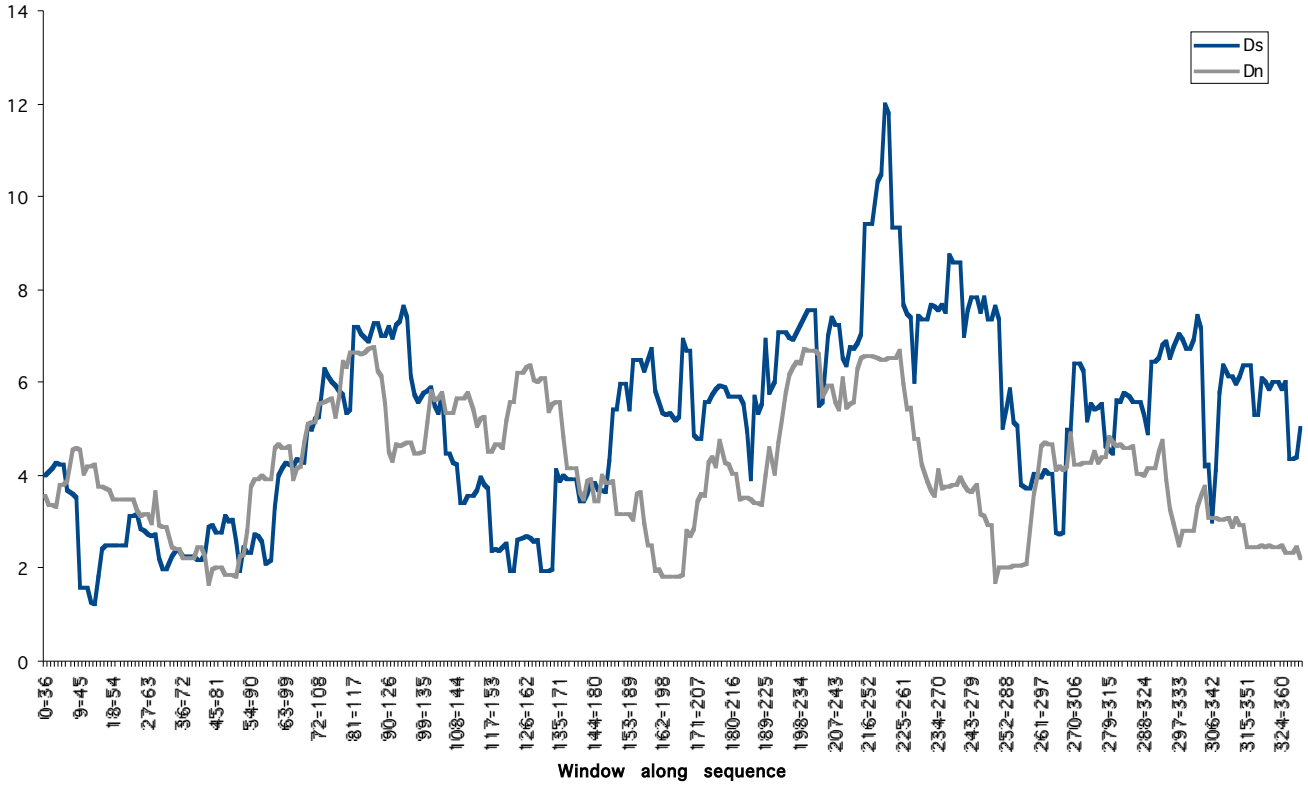
Accumulative Dn values

0-36	3.552402
18-54	3.479945
36-72	2.225918
54-90	3.749186
72-108	5.527779
90-126	4.520141
108-144	5.674173
126-162	6.344442
144-180	3.433806
162-198	1.814455
180-216	4.022491
198-234	6.392410
216-252	6.563881
234-270	4.114182
252-288	2.027652
270-306	4.227049
288-324	3.990405
306-342	3.085654
324-360	2.481808

The first column shows the size of the window being examined (in codon positions) and along side, the value of Dn or Ds calculated. This information can be plotted using something like Microsoft Exel to produce a graph of the Dn and Ds values as they vary across the

sequence length. The Dn or Ds values produced are the sum of every possible pairwise comparison between every sequence selected for that particular window. This can give an broad indication as to the rate of evolution at different parts of the sequence.

The value of the accumulated variance of Dn and Ds for each window is also printed to the file. This may be used to plot standard error bars on the graph for each position. An example of a moving window plot calculated from the lysozyme dataset is shown:



Result_tree.ph

This file contains two trees in phylip format. The following are the two trees produced from the lysozyme dataset:

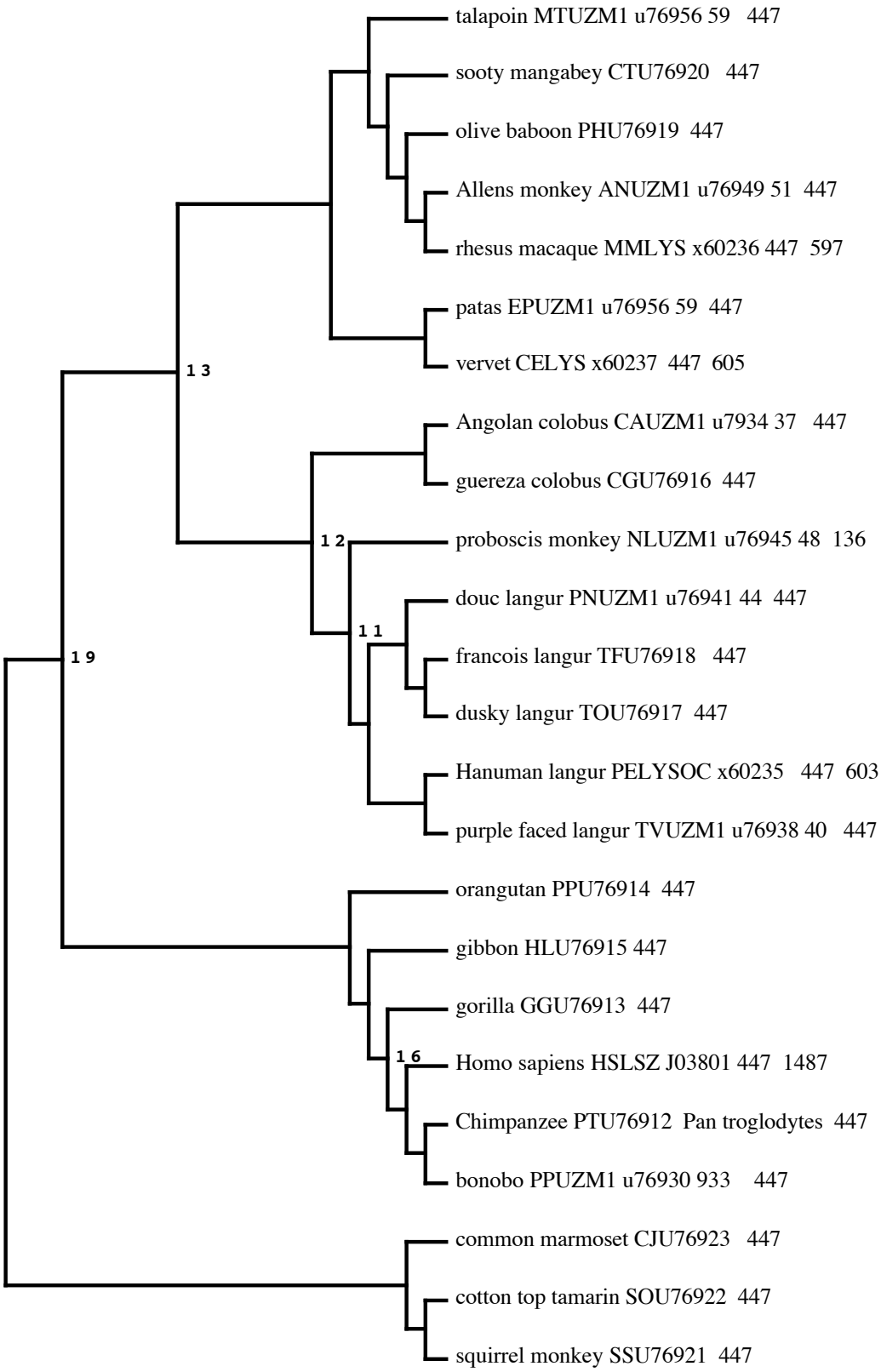
[Results from input file: lyso2.seq Creevey, McInerney method: internal nodes labeled indicate where the ratios differed significantly]

(((((talapoin MTUZM1 u76956_59 447,(sooty mangabey CTU76920 447,(olive baboon PHU76919 447,(Allens monkey ANUZM1 u76949_51 447,rhesus macaque MMLYS x60236 447 _597_)))),(patas EPUZM1 u76956_59 447,vervet CELYS x60237 447 _605_)),((Angolan colobus CAUZM1 u7934_37 447,guereza colobus CGU76916 447),(proboscis monkey NLUZM1 u76945_48 136,((douc langur PNUZM1 u76941_44 447 ,(francois langur TFU76918 447,dusky langur TOU76917 447)),(Hanuman langur PELYSOC x60235 447 _603_,purple_faced langur TVUZM1 u76938_40 447)))11)12)13 ,(orangutan PPU76914 447,(gibbon HLU76915 447,(gorilla GGU76913 447,(Homo sapiens HSLSZ J03801 447 _1487_,(Chimpanzee PTU76912 _Pan troglodytes_ 447,bonobo PPUZM1 u76930_933 447)))16)))19 ,(common marmoset CJU76923 447,(cotton_top tamarin SOU76922 447,squirrel monkey SSU76921 447)));

[This tree has every node labeled just for reference, there are no results in the tree]

(((((talapoin MTUZM1 u76956_59 447,(sooty mangabey CTU76920 447,(olive baboon PHU76919 447,(Allens monkey ANUZM1 u76949_51 447,rhesus macaque MMLYS x60236 447 _597_)0)1)2)3 ,(patas EPUZM1 u76956_59 447,vervet CELYS x60237 447 _605_)4)5 ,((Angolan colobus CAUZM1 u7934_37 447,guereza colobus CGU76916 447)6 ,(proboscis monkey NLUZM1 u76945_48 136,((douc langur PNUZM1 u76941_44 447 ,(francois langur TFU76918 447,dusky langur TOU76917 447)7)8 ,(Hanuman langur PELYSOC x60235 447 _603_,purple_faced langur TVUZM1 u76938_40 447)9)10)11)12)13 ,(orangutan PPU76914 447,(gibbon HLU76915 447,(gorilla GGU76913 447,(Homo sapiens HSLSZ J03801 447 _1487_,(Chimpanzee PTU76912 _Pan troglodytes_ 447,bonobo PPUZM1 u76930_933 447)14)15)16)17)18)19 ,(common marmoset CJU76923 447,(cotton_top tamarin SOU76922 447,squirrel monkey SSU76921 447)20)21);

This is how the trees look in phylip format, but when viewed using a phylogeny viewing program (treeview) the first tree look like this:


talapoin MTUZM1 u76956 59 447
sooty mangabey CTU76920 447
olive baboon PHU76919 447
Allens monkey ANUZM1 u76949 51 447
rhesus macaque MMLYS x60236 447 597
patas EPUZM1 u76956 59 447
vervet CELYS x60237 447 605
Angolan colobus CAUZM1 u7934 37 447
guereza colobus CGU76916 447
proboscis monkey NLUZM1 u76945 48 136
douc langur PNUZM1 u76941 44 447
francois langur TFU76918 447
dusky langur TOU76917 447
Hanuman langur PELYSOC x60235 447 603
purple faced langur TVUZM1 u76938 40 447
orangutan PPU76914 447
gibbon HLU76915 447
gorilla GGU76913 447
Homo sapiens HSLSZ J03801 447 1487
Chimpanzee PTU76912 Pan troglodytes 447
bonobo PPUZM1 u76930 933 447
common marmoset CJU76923 447
cotton top tamarin SOU76922 447
squirrel monkey SSU76921 447
13
12
11
19
16

This is the first tree in the file 'Result_tree.ph' and refers to the results of the relative rate ratio test described by Creevey and McInerney (2002). Those internal branches whose ratios deviated from that expected under neutrality (and resulted in a significant result) are labelled on this tree. This tree is directly comparable to the table at the end of the main output file (described previously). There is no indication of whether the results at these internal branches were directional or non-directional positive seleciton, and significant results from G tests are displayed here, without regard as to whether or not they were statistically reliable (see the section on main output file).
When viewed using "treeview", the label is printed to the right of the internal branch to which it belongs.

The second tree in result_tree.ph is used for reference only. It displays the label assigned to each internal branch of the phylogeny. This may be used to determine the position of internal branches that did not produce a significant result, but are of biological interest. This also allows the user to find the position of results from other methods performed by Crann (see the section on 'substitutions.out' and 'yadf.out' for more details).

Substitutions.out

The relative rate test described by Creevey and McInerney (2002) assesses whether the number of replacement invariable (RI) or variable (RV) substitutions is greater than expected from neutrality. If there is no significant result at an internal branch this indicates that the number of these substitutions does not deviate from that expected from neutrality. It does not say anything about whether or not negative selection was acting. This is where the result from this file can be used.

This file holds the results of the neutral substitution test. An example of the output for this file is on the next page. These are the actual results obtained from analysing the lysozyme dataset. This method assesses whether there were more or less replacement or silent substitutions than expected under neutrality. This is basically the same as as Dn/Ds ratio, except that rather than comparing sequences it looks at the number of substitutions that occurred along each lineage.

The first column represents each internal branch of the phylogeny. Each internal branch defines a clade within which we assess the number of substitutions. The labels given to each internal branch are constant across all the methods Cran implements and their positions on the tree are shown in the file 'result_tree.ph'.

This method uses the number of RI and RV substitutions calculated as part of the relative rate ratio test to count the number of total replacement substitutions that occurred within each clade (defined by an internal branch). The total number of replacements is given by RI + RV. The total number of silent substitutions that occurred within each clade (defined by an internal branch) is given by summing the number of SI and SV substitutions calculated during the relative rate ratio test. These two values are the observed number of replacement and silent substitutions in the second column on the next page.

Crann then calculates the total number of replacement and silent sites within each clade by using the method described by Li (1993). The sequence of every taxa and recontructed ancestor within each clade is used to arrive at this value. The number of replacement and silent sites calculated within each clade is shown in the fourth column on the next page. The number of sites within each clade is not normalised for the number of taxa and ancestral sequences within each clade, so the total number of sites gets larger as the base of the tree is approached.

The calculated values of replacement and silent substitutions at each internal branch and associated pvalues

Node	Observed		Expected		Total sites		Changes per site		pvalue (Observed vs expected)		
	Replacement	Silent	Replacement	Silent	Replacement	Silent	Replacement	Silent			
0	3	0	2.33	0.67	906.5	260.5	0.003309	0	G = 0.576348	Gtest:0.5 > pvalue > 0.2	fishers: p = 0.640593
1	3	1	3.143	0.857	1482.5	404.5	0.002024	0.002472	G = 0.011070	Gtest:0.95 > pvalue > 0.9	fishers: p = 0.578712
2	4	1	3.948	1.052	2058.5	548.5	0.001943	0.001823	G = 0.001324	Gtest:0.99 > pvalue > 0.95	fishers: p = 0.763613
3	6	2	6.335	1.665	2634.5	692.5	0.002277	0.002888	G = 0.034661	Gtest:0.9 > pvalue > 0.5	fishers: p = 0.631975
4	0	3	2.328	0.672	907	262	0	0.01145	G = 3.803430	Gtest:0.1 > pvalue > 0.05	fishers: p = 0.136771
5	9	6	11.827	3.173	3829.5	1027.5	0.00235	0.005839	G = 1.196249	Gtest:0.5 > pvalue > 0.2	fishers: p = 0.235347
6	3	1	3.12	0.88	915	258	0.003279	0.003876	G = 0.007824	Gtest:0.95 > pvalue > 0.9	fishers: p = 0.577875
7	1	0	0.776	0.224	911.5	263.5	0.001097	0	G = 0.104036	Gtest:0.9 > pvalue > 0.5	fishers: p = 0.887872
8	2	3	3.922	1.078	1493.5	410.5	0.001339	0.007308	G = 1.363847	Gtest:0.5 > pvalue > 0.2	fishers: p = 0.278945
9	1	1	1.553	0.447	914	263	0.001094	0.003802	G = 0.238411	Gtest:0.9 > pvalue > 0.5	fishers: p = 0.615680
10	4	4	6.266	1.734	2698.5	746.5	0.001482	0.005358	G = 1.290326	Gtest:0.5 > pvalue > 0.2	fishers: p = 0.254265
11	7	7	10.999	3.001	3278.5	894.5	0.002135	0.007826	G = 2.396997	Gtest:0.2 > pvalue > 0.1	fishers: p = 0.118312
12	19	9	21.989	6.011	4483.5	1225.5	0.004238	0.007344	G = 0.788686	Gtest:0.5 > pvalue > 0.2	fishers: incalculable
13	31	15	36.205	9.795	8601	2327	0.003604	0.006446	G = 1.471646	Gtest:0.5 > pvalue > 0.2	fishers: incalculable
14	0	0	0	0	904.5	274.5	0	0	G = NaN	Gtest:1.0 > pvalue > 0.9	fishers: incalculable
15	2	3	3.818	1.182	1521	471	0.001315	0.006369	G = 1.206959	Gtest:0.5 > pvalue > 0.2	fishers: p = 0.302219
16	4	3	5.394	1.606	2099	625	0.001906	0.0048	G = 0.564902	Gtest:0.5 > pvalue > 0.2	fishers: p = 0.408530
17	8	5	10.081	2.919	2677	775	0.002988	0.006452	G = 0.740680	Gtest:0.5 > pvalue > 0.2	fishers: p = 0.322778
18	20	7	21.009	5.991	3254	928	0.006146	0.007543	G = 0.099105	Gtest:0.9 > pvalue > 0.5	fishers: incalculable
19	63	29	72.212	19.788	12145	3328	0.005187	0.008714	G = 2.351645	Gtest:0.2 > pvalue > 0.1	fishers: incalculable
20	6	5	8.558	2.442	916.5	261.5	0.006547	0.01912	G = 1.250867	Gtest:0.5 > pvalue > 0.2	fishers: p = 0.242243
21	6	10	12.39	3.61	1539.5	448.5	0.003897	0.022297	G = 5.141066	Gtest:0.025 > pvalue > 0.01	fishers: p = 0.026085 *

If the sequences within each clade were evolving neutrally, then the proportion of replacement to silent substitutions would be the same as the proportion of replacement to silent sites. To test this, the total number of substitutions observed within each clade (replacement and silent) are used to calculate the expected number of replacement to silent substitutions based on the number of each type of site. The results of this calculation are shown in the third column in the previous page.

Finally to assess whether the observed number of replacement or silent substitutions are significantly greater than expected from neutrality. Statistical tests for independence are carried out to compare the ratio of observed replacement to silent substitutions to the ratio of expected replacement to silent substitutions. The first test used is the G-test for independence, the results of which are displayed for each internal branch in the previous page. The G value is the value calculated and used to estimate the p value based on a χ-square distribution. The second test used is the Fishers exact test. As previously stated, if the Fishers exact test returns any result (i.e. does not have the word 'incalculable' instead of a p value) then this should be taken as the 'true' p value. Otherwise the G test result should be used.

Unlike the results of the relative rate ratio test, significant results ($p<0.05$) are not flagged by any markers, so it is up to the user to examine this file carefully.

A significant result at any internal branch may either be evidence of positive or negative selection, depending on whether the result is due to an increased number of observed replacement or silent substitutions respectively. A significant increase of replacement substitutions is evidence of positive selection, however there is no statistical method to assess whether this is directional or non-directional positive selection. This is a recurring problem with this type of analysis.

It is possible to observe negative selection (a significantly higher number of silent substitutions) within a clade, and for the relative rate ratio test to indicate that directional or non-directional positive selection is acting within the same clade. What does this mean? In our experience, a result like this indicates that the majority of the sites within the clade are under severe negative selection, however there are some sites under positive selection (Positive selection in a sea of negative selection). In this case traditional Dn/Ds analyses would

never identify that positive selection is acting because of the high number of sites restricted from changing.

If the relative rate ratio test is not significant far a branch, then the results in this file will identify whether the clade is evolving neutrally or whether it is evolving under negative selection. If both the relative rate ratio test AND the results in this file are NOT significant for any clade, this is evidence that the sequences within the clade are evolving neutrally.

The fifth column shows the number of changes per site within each clade, this is for those who are more familiar with dealing with Dn and Ds ratios. The number of replacement substitutions per site is equivalent to the value of Dn and the number of silent substitutions per site is equivalent to Ds. The p-values could then be thought of as a statistical assessment of whether the value of Dn is significantly greater than that of Ds or vice versa. It essentially amounts to the same as described earlier.

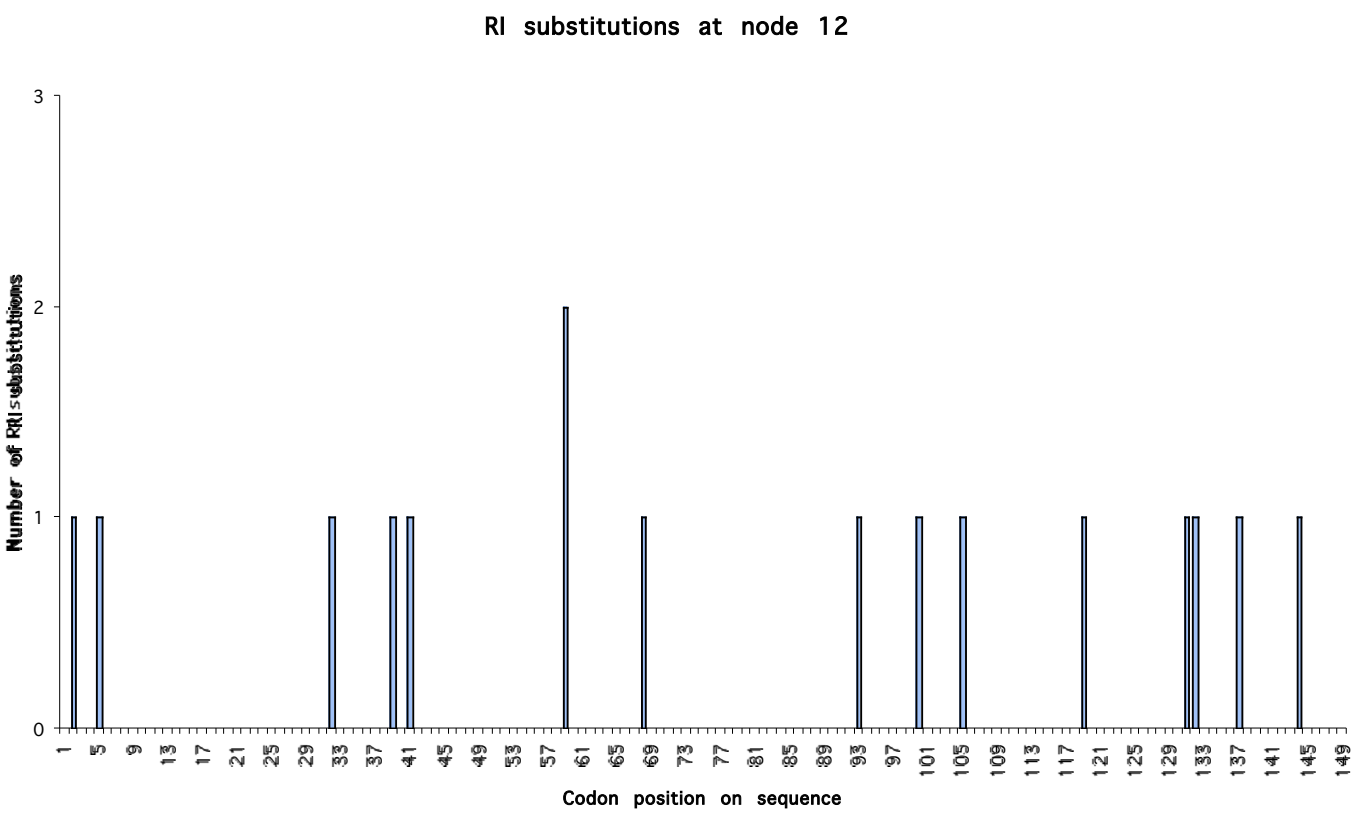
Seq_graph.out

This file reports the position of all the Replacement Invariable (RI) and variable (RV) substitutions and Silent Invariable (SI) and variable (SV) substitutions as calculated at each internal node of the tree by the relative rate ratio test.

An excerpt of the contents of the file from the lysozyme dataset follows:

```
Node 0  RI     RV     SI     SV     Node 1  RI    RV     SI     SV     Node 2  RI    RV     ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        0      0      0      0              0     0      0      0              0     0      ....
        ....   ....   ....   ....           ....  ....   ....   ....           ....  ....
```

The number of each type of substitution is displayed from the first site (in codons) to the last. If there was evidence of positive directional selection within any clade the RI substitutions would be the substitutions causing the result. Using the results as displayed in this file it is possible to plot where the RI substitutions occurred on the sequence and how many occurred. The file is in "tab-delimited" format which makes it possible to open in graph-building programs like Microsoft Exel. One of the node which showed a positive result was internal branch (node) number 12. The result at this internal branch is also supported by biochemical evidence (Activity at a lower ph than normal and resistance to cleavage by pepsin) and results by other researchers (Messier and Stewart (1997) and Yang (1998)). The RI substitutions ploted for just this internal branch looks like this:


RI substitutions at node 12
Number of RI substitutions
0
1
2
3
1
5
9
13
17
21
25
29
33
37
41
45
49
53
57
61
65
69
73
77
81
85
89
93
97
101
105
109
113
117
121
125
129
133
137
141
145
149
Codon position on sequence

YADF.out

The name of this file stands for **Y**et **A**nother **D**istance **F**ile and it contains the results of Crann's implementation of the method described by Messier and Stewart (1987). In this method the ancestral sequences reconstructed during the relative rate ratio test are used to perform Dn/Ds tests between each node and its children on the phylogeny. This is an attempt to clarify where in a phylogeny positive selection has occurred by identifying the internal branch on which it has occurred.

As can be seen on the next page, the Dn and Ds value are calculated for every internal branch of the tree. The value of Dn/Ds is then calculated (the words "NaN" and "Inf" represent where the calculation was not possible). The last two columns display for each internal branch the variance associated with each of the Dn and Ds calculations. This may be used to determine if Dn is statistically greater than Ds or vice-versa.

The positions of each of the internal branches on the phylogeny may be obtained by viewing the second tree contained in the file 'result_tree.ph'. This tree contains the labels associated with every internal branch of the tree.

Note:
It is possible to get a Dn or Ds result that is negative using Li's (1993 & 1985) methods. An example of this can be seen in the calculation of Ds between node 13 and node 12 on the next page. This is an artefact of the calculation caused by the occurrence of a much larger number of transversions than transitions. In this case is it possible to consider the value a positive number for the calculation of Dn/Ds.

Which branch of the tree?	Dn	Ds	Dn/Ds	var Dn	var Ds
node0 to Allens monkey ANUZM1 u76949_51 447	0.003534	0.014063	0.251265	0.003571	0.000203
node0 to rhesus macaque MMLYS x60236 447 _597_	0.003521	0	Inf	0.003559	0
node1 to node0	0	0	NaN	0	0
node1 to olive baboon PHU76919 447	0	0.014063	0	0	0.000203
node2 to node1	0	0	NaN	0	0
node2 to sooty mangabey CTU76920 447	0	0	NaN	0	0
node3 to node2	0.002695	0	Inf	0.002067	0
node3 to talapoin MTUZM1 u76956_59 447	0.00272	0.006419	0.423666	0.000007	0.003691
node4 to patas EPUZM1 u76956_59 447	0	0.012881	0	0	0.007481
node4 to vervet CELYS x60237 447 _605_	0	0.006364	0	0	0.003565
node5 to node3	0.003521	0	Inf	0.003559	0
node5 to node4	0	0	NaN	0	0
node6 to Angolan colobus CAUZM1 u7934_37 447	0	0	NaN	0	0
node6 to guereza colobus CGU76916 447	0	0	NaN	0	0
node7 to francois langur TFU76918 447	0	0	NaN	0	0
node7 to dusky langur TOU76917 447	0	0	NaN	0	0
node8 to node7	0.003496	0	Inf	0.003533	0
node8 to douc langur PNUZM1 u76941_44 447	0.003484	0.019802	0.175959	0.003521	0.01039
node9 to Hanuman langur PELYSOC x60235 447 _603_	0	0	NaN	0	0
node9 to purple_faced langur TVUZM1 u76938_40 447	0	0	NaN	0	0
node10 to node8	0	0	NaN	0	0
node10 to node9	0.003484	0.013867	0.251264	0.003521	0.000197
node11 to node10	0.003496	0	Inf	0.003533	0
node11 to proboscis monkey NLUZM1 u76945_48 136	0	0.020224	0	0	0.00345
node12 to node6	0.009746	0.006547	1.488478	0.007175	0.003712
node12 to node11	0.009753	0.006549	1.489284	0.007175	0.003671
node13 to node5	0.009801	0.006417	1.527321	0.009359	0.003092
node13 to node12	0.030842	-0.00008	-387.49576	0.023729	0.000001
node14 to Chimpanzee PTU76912 _Pan troglodytes_ 447	0	0	NaN	0	0

Ancestors.out

This file contains in fasta format all the ancestral sequences reconstructed during the relative rate ratio analysis along with all the taxa from the input file (option 1, main menu). The number associated with each internal branch is given as the name of its reconstructed ancestral sequence. The positions of the internal branch numbers can be viewed in the second tree in 'result_tree.ph'.

Dn.dis, Ds.dis & DnDs.dis

These files are produced during the calculation of the relative rate ratio test (option 4, main menu) or if the user chooses to calculate pair-wise distances along the entire length of the sequence (option 5, main menu). They contain the results of every pair-wise comparison between all the sequences in the input file (option 1, main menu). The file 'Dn.dis' contains the all the Dn distances, 'Ds.dis' contains all the Ds distances and 'DnDs.dis' contains all the results of every Dn/Ds calculation.

The results in these files are in "lower triangular format" and are in the correct format to be imported directly into the phylip program 'neighbor'.

Reference List.

Creevey, C. and J. O. McInerney (2002). An algorithm for detecting directional and non-directional positive selection, neutrality and negative selection in protein coding DNA sequences. Gene **300**: 43-51.

Felsenstein, J. (1981). Evolutionary trees from DNA sequences: a maximum likelihood approach. J Mol Evol **17**: 368-76.

Gillespie, J. H. (1998). Population genetics: a concise guide. London, The John Hopkins University Press.

Graur, D. and W. H. Li (1991). Neutral Mutation Hypothesis Test. Nature **354**: 114-115.

Hennig, W. (1966). Phylogenetic systematics. Urbana, University of Illinois Press.

Kellogg, E. A. and R. Appels (1995). Intraspecific and Interspecific Variation in 5s Rna Genes Are Decoupled in Diploid Wheat Relatives. Genetics **140**: 325-343.

Kimura, M. (1983). The neutral theory of miolecular evolution. Cambridge, Cambridge University Press.

Kimura, M. (1983). The neutral theory of molecular evolution. Evolution of genes and proteins. R. Koehn. Sunderland, Sinauer associates inc.

Li, W.-H., C.-I. Wu, et al. (1985). A new method for estimating synonymous and nonsynonymous rates of nucleotide substitution and considering the relative likelihood of nucleotide and codon changes. Mol. Biol. Evol. **2**: 150-174.

Li, W. H. (1993). Unbiased Estimation of the Rates of Synonymous and Nonsynonymous Substitution. Journal of Molecular Evolution **36**: 96-99.

McDonald, J. H. and M. Kreitman (1991). Adaptive Protein Evolution at the Adh Locus in Drosophila. Nature **351**: 652-654.

McDonald, J. H. and M. Kreitman (1991). Neutral Mutation Hypothesis Test - Reply. Nature **354**: 116-116.

McInerney, J. O. (1998). Replicational and transcriptional selection on codon usage in Borrelia burgdorferi. Proc Natl Acad Sci U S A **95**: 10698-703.

Messier, W. and C. B. Stewart (1997). Episodic adaptive evolution of primate lysozymes. Nature **385**: 151-4.

Sharp, P. M., M. Stenico, et al. (1993). Codon usage: mutational bias, translational selection or both? Biochem. Soc. Trans. **21**: 835-841.

Smith, J. M. (1970). Population size, polymorphism, and the rate of non-Darwinian evolution. American Naturalist **104**: 231-236.

Sokal, R. R. and F. J. Rohlf (1981). Biometry. San Fransisco, Freeman.
Templeton, A. R. (1996). Contingency tests of neutrality using intra/interspecific gene trees: The rejection of neutrality for the evolution of the mitochondrial cytochrome oxidase II gene in the hominoid primates. Genetics **144**: 1263-1270.
Whittam, T. S. and M. Nei (1991). Neutral Mutation Hypothesis Test. Nature **354**: 115-116.
Yang, Z. (1998). Likelihood ratio tests for detecting positive selection and application to primate lysozyme evolution. Mol Biol Evol **15**: 568-73.
Yang, Z. (2000). Maximum likelihood estimation on large phylogenies and analysis of adaptive evolution in human influenza virus A. J Mol Evol **51**: 423-32.
Yang, Z., R. Nielsen, et al. (2000). Codon-substitution models for heterogeneous selection pressure at amino acid sites. Genetics **155**: 431-49.