



## ProtTest: Selection of best-fit models of protein evolution version 2.4

Federico Abascal<sup>1,2</sup>, Rafael Zardoya<sup>2</sup> and David Posada<sup>1</sup>

1: Universidad de Vigo, Vigo, Spain

2: Museo Nacional de Ciencias Naturales, Madrid, Spain

[abascal@mncn.csic.es](mailto:abascal@mncn.csic.es), [dposada@uvigo.es](mailto:dposada@uvigo.es), [rafaz@mncn.csic.es](mailto:rafaz@mncn.csic.es)

- What can I use ProtTest for? – Introduction
- The program: using ProtTest
  - Download and installation
    - *Installing java*
  - Running ProtTest
    - *ProtTest through its graphical Interface (GUI)*
    - *ProtTest at the command-line*
    - *ProtTest at the web*
  - Optimization strategies
  - Alignment and tree formats
  - Restricting the set of candidate models
  - ProtTest's output. A guided example: the ribosomal L5 protein family
- Known bugs
- Program history
- BACKGROUND
  - Models of protein evolution
  - Statistics for model selection: Akaike Information Criterion and others
    - Phylogenetics and sample size
    - Akaike weights and the relative importance of parameters
      - *Parameter importance*
      - *Model-averaged parameter estimates*
- Credits and acknowledgements
- References

### What can I use ProtTest for? – Introduction

ProtTest is a bioinformatic tool for the selection of the most appropriate model of protein evolution (among the set of candidate models) for the data at hand. ProtTest makes this selection by finding the model with the smallest Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC) score. At the same time, ProtTest obtains model-averaged estimates of different parameters (Posada and Buckley 2004) and calculates the importance of each of these parameters. ProtTest differs from its nucleotide homolog Modeltest (Posada and Crandall 1998) in that it does not include likelihood ratio tests (many models implemented in ProtTest are not nested).

### The program: using ProtTest

ProtTest is written in java and takes advantage of the **PAL** library (Drummond and Strimmer 2001) for manipulating trees and alignments, and of the **Phyml** program (Guindon and Gascuel 2003) for the computation of likelihoods and the estimation of parameters. Given an alignment and a tree (provided by the user or calculated with the BIONJ algorithm (Gascuel 1997)), ProtTest currently computes the likelihood for each one of 112 candidate models of protein evolution: matrices WAG, Dayhoff, JTT, mtREV, MtMam, MtArt, VT, RtREV, CpREV, Blosum62, LG, DCMut, HIVw, and HIVb with the +I, +G, and +F parameters. Then, the fit of the models can be estimated using the AIC, AICc, and BIC.

### **Download and installation**

ProtTest works in Mac OSX, Windows, and Linux, and requires a version of the java runtime environment equal or posterior to 1.3 (read section “Installing java” if you don’t have it). ProtTest is available from <http://darwin.uvigo.es>. After registration, download the package and decompress it in any directory. Some examples are included.

#### Installing java

First of all, make sure that you have a Java Virtual Machine (JVM) properly installed in your system. To test your JVM

- 1) Go to <http://www.java.com/en/download/help/testvm.jsp>
- 2) Or in a terminal window, type “java -version”.

The JVM is also included in:

- Java Runtime Environment (JRE)
- Java 2 Platform Standard Edition (J2SE)

More information on obtaining the JVM in:

- <http://java.sun.com/>

To automatically download the JVM

- <http://java.sun.com/webapps/getjava/BrowserRedirect>

### **Running ProtTest**

There are three different ways to run ProtTest: using the graphical interface (recommended), using the command-line version, and through the web server (<http://darwin.uvigo.es>).

#### ProtTest through its graphical Interface (GUI)

Just double-click the jar file (**ProtTest.jar**).

Note: Some Linux environments need to be configured to respond to jar double-clicks. If you are not able to set up Linux to do so, you can launch ProtTest by running the runXProtTest script.

A window like the one in **figure 1** will appear in the screen. Now follow these steps:

1. Input an alignment in phylip (recommended) or nexus-sequential format (more about accepted formats in the section “Alignment and tree formats” below).
2. Optionally you can input a tree topology in newick format; if not, a BIONJ tree will be calculated.
3. Select the strategy: fast (fixed tree; optimization of model parameters and branch lengths; this is the strategy used by Modeltest) or slow (optimization of model, branches and topology of the tree) (see the “Optimization strategies” section below).
4. Optionally, you might want to restrict the set of candidate models. For that you should click the “models” button. Read first the section “Restricting the set of candidate models”.
5. And click the “Start” button!



Figure 1: **ProtTest** main window.

At this moment ProtTest starts computing the parameters for the different models. **Be aware that computation can take some time, maybe hours or even days.** You can watch the progress in a new window (the **console**, see **figure 2**). If there's some error related with the format of the alignment or the installation of the program, a warning will appear in the console.

When likelihood computations are finished, you'll be warned and prompted to select a statistical framework (AIC, AICc or BIC) for determining which of the candidate models best fits your data. If you select AICc or BIC you will be prompted to specify a criterion to estimate the size of the sample. Additionally, you can ask ProtTest to display a comparison of different selection scenarios (AIC, AICc and BIC with three different criteria for sample size) by clicking the "overall comparison" button.

If you want ProtTest to display the tree corresponding to the best model, select between an ASCII representation and the tree in Newick format. Under the **fast strategy** this tree becomes the topology of the initial tree (BIONJ or provided by the user) with branch lengths optimized under the best-fit model under the current selection criterion. Under the **slow strategy** this tree is the best ML tree (both topology and branch lengths will be optimized) under the best-fit model under the current selection criterion

To learn about the **output** of the program (and its interpretation) go to the section “ProtTest’s output. A guided example”.

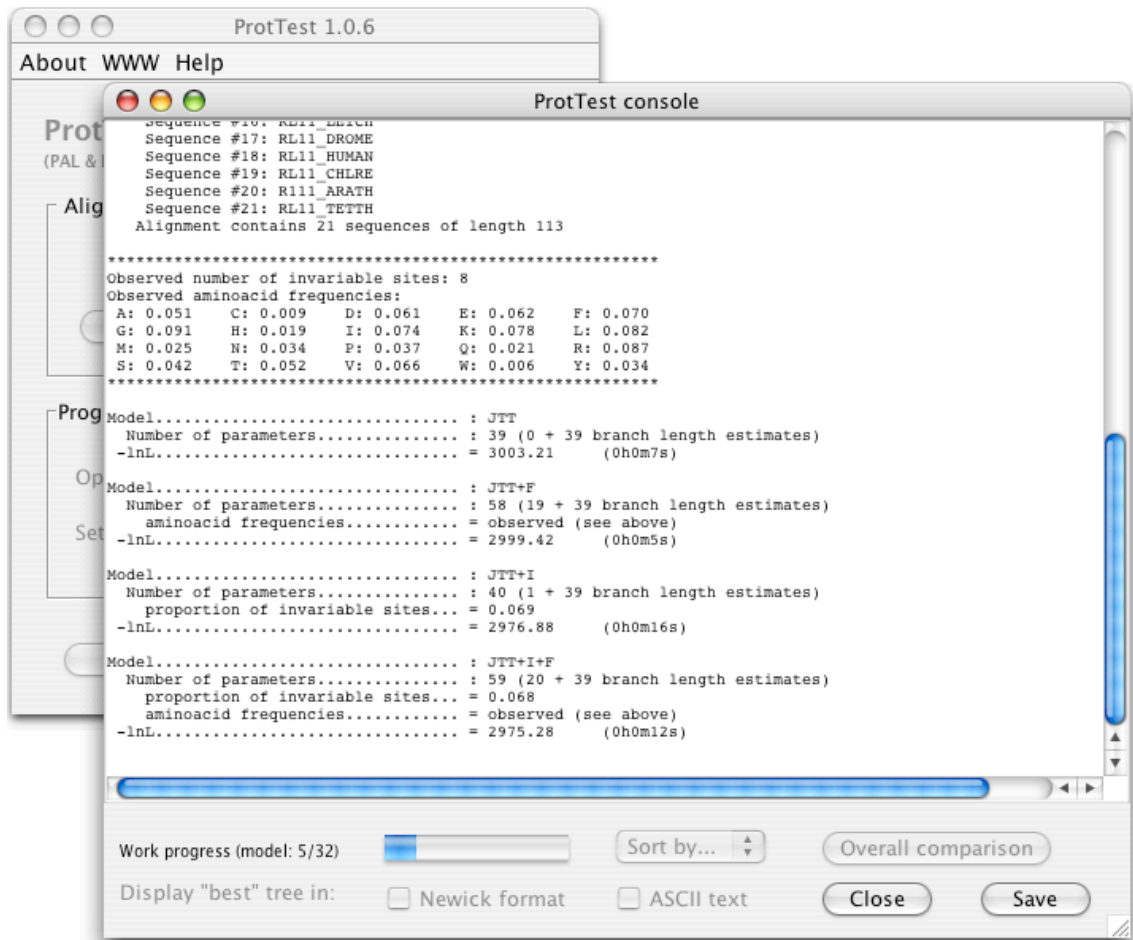


Figure 2: ProtTest’s console window

### ProtTest at the command-line

To run Prottest at the command line, open a shell window (terminal), change to the directory where ProtTest is installed and run the script runProtTest (**In Windows**, use the script runProtTest.bat instead), specifying the following **options**:

```

-i:      alignment file (required)
-t:      tree file      (optional) [default: NJ tree]
-o:      output file   (optional) [default: STDOUT]
-sort:   A/B/C/D       (optional) [default: A]
         A: AIC
         B: BIC
         C: AICc
         D: LnL
-all:   T/F. If true a 7-framework comparison table is displayed
         [default: true]
-S:     optimization strategy mode: [default: 0]
         0: Fast (optimize branch lengths & model)
         1: Slow (optimize branch lengths, model & topology)
-sample: sample size for AICc and BIC corrections [default: 2]
         0: Shannon-entropy Sum
         1: Average (0-1)Shannon-entropy x NxL
         2: Total number of characters (alignment length)

```

```

    3: Number of variable characters
    4: Alignment length x num taxa (NxL)
    5: Specified by the user
  -size: number specifying sample size, only for "-sample 5"
  -t1:   T/F. If true, display best-model's newick tree
         [default: false]
  -t2:   T/F. If true, display best-model's ASCII tree
         [default: false]
  -verbose: T/F (true or false) [default: true]
  -[model]: T/F (true or false) [default: all models are set to true]
            model= -JTT,-LG,-DCMut,-MtREV,-MtMam,-MtArt,-Dayhoff,-WAG,-
                  RtREV,-CpREV,-Blosum62,-VT,-HIVb,-HIVw,
  -[addon]: T/F (true or false)
            [default: all model addons are set to true]
            addon=-+F,-+I,-+I+G,-+G

```

The only required option is “-i”, and it must be followed by the name of the file that contains the alignment. Other options have their own values set by default (which you may want to change).

If you want to specify a tree topology (its branch lengths are of no importance, they will be optimized) use the “-t” option followed by the tree file name (this option doesn’t matter when “-S 1” is also specified).

Set the “-sort” option to indicate what statistic should be used for model selection (AIC, AICc, BIC or LnL).

If AICc or BIC (“-sort C” and “-sort B”, respectively) are selected, you might want to change the default criterion for sample size interpretation. To accomplish this, set the “-sample” option to one of the values shown above (0-5).

The optimization strategy (see the section “Optimization strategies” below) is specified by the -S option followed by a 0 or 1 (fast or slow).

The “-all” option is set to true by default. Since the command-line version of ProtTest is not interactive and you cannot *play* with the different frameworks once the likelihoods are calculated, the -all option is useful for having a table in which you can see at one sight how the best model selection is affected under seven different scenarios (AIC, AICc and BIC with three different criteria for sample size –corresponding to 0, 1 and 2 in the “-sample” option-).

If -t1 or -t2 options are set to “true”, the tree corresponding to the best model will be displayed in the output (in newick format or ASCII representation, respectively). The order in which you specify the options doesn’t matter.

Example:

```

runProtTest -i alignment_file.phylip -S 0 -sort C -sample 1 -o
results.txt -t1 T -MtArt F -MtREV F -MtMam F -+F F

```

By running ProtTest with these options we will find the model that fits best the alignment contained in “alignment\_file.phylip” according to the AICc criterion and taking as sample size the number of sequences multiplied by the length of the alignment and multiplied by a correction factor (the averaged Shannon’s entropy normalized to a 0-1 scale) (see more details in the “Phylogenetics and sample size” section). Results will be stored in a file named “results.txt”. The “-t1 T” is telling

ProtTest to return the tree optimized for the best fit model in newick format. This command line excludes MtArt, MtMam and MtREV models from the analysis, as well as the add-on +F.

To learn about the **output** of the program (and its interpretation) go to the section “ProtTest’s output. A guided example”.

### New options

It is now possible to restrict the set of candidate models in the command-line version of ProtTest. For example, if we would like to discard the “mitochondrial models”, we should type:

```
-MtArt F -MtREV F -MtMam F
```

In addition, if we would like to discard +I+G models as well as +F models, just type:

```
--I+G F --F F
```

To discard different models, proceed in a similar way.

### ProtTest at the web

ProtTest analysis can also be executed at its web-site: <http://darwin.uvigo.es/>. Functionality of the web version of ProtTest is similar to the graphical one, but the ability of restricting the set of candidate models, and the ability to select interactively different model selection criteria is not provided, as in the command-line version.

Enter the web page and just input an alignment (and optionally a tree), select the statistical criterion for model selection, and the other parameters. Your job will be sent to a queue and you’ll be notified by e-mail when the analysis is finished.

### Optimization strategies

Ideally, one should optimize the tree topology, its branch lengths and the model parameters (for each model) to assure maximum likelihood is achieved. This complete optimization strategy can be performed by ProtTest when the “slow” option is selected in the main window. However, model selection seems to be quite robust to topology as long as this is a reasonable representation of the true phylogeny (Posada and Crandall 2001). Therefore a faster strategy (and the one implemented in the program Modeltest (Posada and Crandall 1998)) is to estimate a “good” tree and make all likelihood calculations for all models in this fixed tree. This strategy is named “fast” in ProtTest. Because it only optimizes branch lengths and model parameters it has the advantage of being much faster.

### Alignment and tree formats

ProtTest is able to read (through the PAL library) the following alignment formats: **phylip** (interleaved or sequential) and **nexus** (sequential). The phylip format is recommended since the nexus reader has some bugs. For reading trees, the **newick** format is supported. You can find examples of these formats in the “formats-examples” directory.

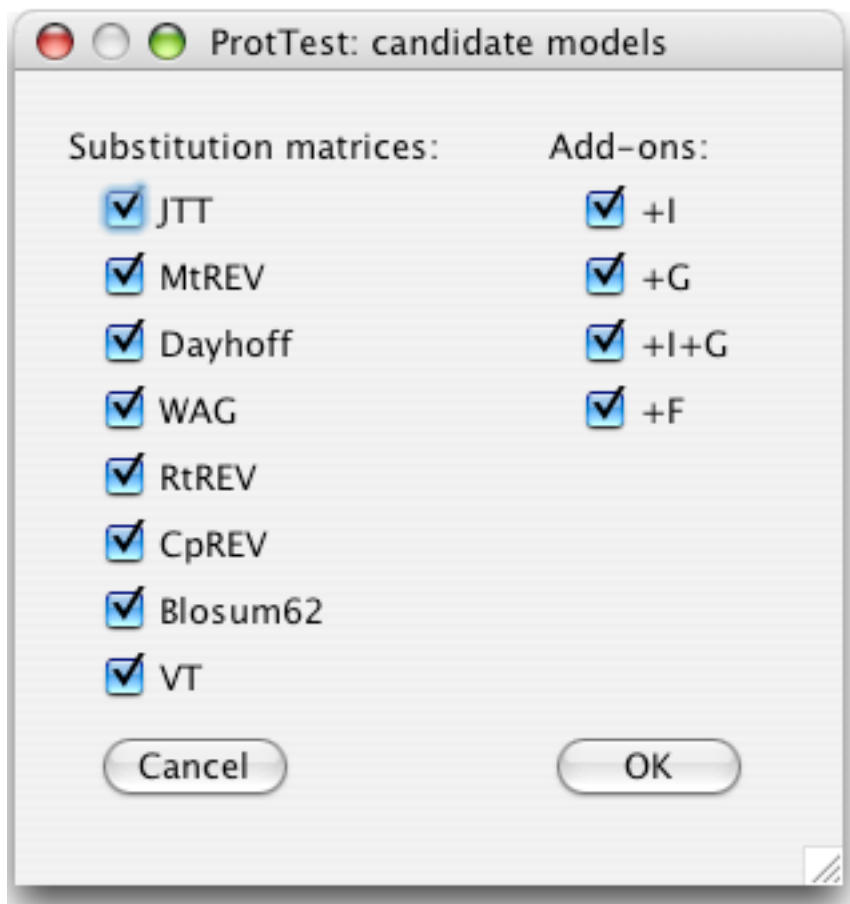
If your data is in a different format you can convert it to one of the accepted formats using programs such as MacClade (Maddison and Maddison 1992) or ReadSeq (Gilbert 2001). The second can be easily used through its web version: <http://bimas.dcrn.nih.gov/molbio/readseq/>.

You can get more information about alignment and tree formats at: <http://workshop.molecularevolution.org/resources/fileformats/>

[http://workshop.molecularevolution.org/resources/fileformats/tree\\_formats.php](http://workshop.molecularevolution.org/resources/fileformats/tree_formats.php)

### Restricting the set of candidate models

This functionality is only available through ProtTest's graphical interface. If the set of models you are interested in is a subset of those offered by ProtTest (e.g. you want to select the best model for using a program that doesn't support the Dayhoff matrix) you can restrict the set of candidate models by clicking the "models" button. Then select which empirical matrices (WAG, Dayhoff, JTT, MtREV, VT, RtREV, CpREV and Blosum62) and *improvements* (+I, +G, +F, +I+G) should be included in the analysis (see figure 3).



**Figure 3.** Restricting the set of candidate models.

### ProtTest's output. A guided example: the ribosomal L5 protein family (using ProtTest version 1.2.2)

In this section we'll explain the output of the program and its possible interpretations through a guided example: the case of the ribosomal L5 (C-terminal domain) protein family, which you can find in the `examples/Ribosomal_L5_PF00673` directory. We will use the graphical version of ProtTest.

First, we double-click the ProtTest.jar file. If we have the proper java version a window will appear. Then, we enter the alignment (we can use the file included in the examples folder at `examples/Ribosomal_L5_PF00673/alignment` file). We leave the other options as they are and press the “Start” button. A new window will appear (we will refer to this window as the “console” window, to distinguish it from the main one). In the console we can watch the progress of the analysis and check if everything is working properly. We will see a header reporting some information about the alignment and below the results for each model as they are being optimized. Something like:

```
The header [bla, bla]

Model..... : JTT
  Number of parameters..... : 39 (0 + 39 branch length estimates)
  -lnL..... = 2980.51      (0h0m15s)

Model..... : JTT+F
  Number of parameters..... : 58 (19 + 39 branch length estimates)
  aminoacid frequencies..... = observed (see above)
  -lnL..... = 2977.23      (0h0m6s)

Model..... : JTT+I
  Number of parameters..... : 40 (1 + 39 branch length estimates)
  proportion of invariable sites... = 0.069
  -lnL..... = 2955.16      (0h0m8s)
```

When likelihoods and model parameters are estimated for all models, we will be prompted to select a statistical framework for the selection of the best model. To accomplish this we should select one of the options that can be found at the bottom of the console. Even if raw **likelihoods** are not adequate for model selection we start by selecting this option to illustrate some concepts. We'll get something like this:

```
*****
Maximum Likelihood (-lnL) framework
*****
Best model according to -lnL: WAG+I+G+F
*****
Model          deltaAIC      AIC          -lnL*         AICw
-----
RtREV+I+G+F    0.96           5908.73      -2894.36      0.31
RtREV+G+F      2.25           5910.02      -2896.01      0.16
WAG+I+G+F      13.48          5921.25      -2900.63      0.00
WAG+G+F        16.46          5924.23      -2903.12      0.00
WAG+I+G        0.00           5907.77      -2912.89      0.49
WAG+G          5.29           5913.06      -2916.53      0.04
Blosum62+I+G  9.84           5917.61      -2917.81      0.00
CpREV+I+G+F   51.66          5959.43      -2919.71      0.00
(rest of lines omitted)
```

We can see which model has the highest likelihood: the RtREV+I+G+F. Some information related with the AIC framework is also displayed, but forget it by now.

Now we select the **AIC** framework and this is what we obtain:

```
*****
Akaike Information Chriterion (AIC) framework
*****
Best model according to AIC: WAG+I+G
*****
Model          deltaAIC*      AIC          AICw          -lnL
-----
```



WAG+I+G	0.00	5907.77	0.49	-2912.89
RtREV+I+G+F	0.96	5908.73	0.31	-2894.36
RtREV+G+F	2.25	5910.02	0.16	-2896.01
WAG+G	5.29	5913.06	0.04	-2916.53
Blosum62+I+G	9.84	5917.61	0.00	-2917.81
WAG+I+G+F	13.48	5921.25	0.00	-2900.63
Blosum62+G	14.07	5921.84	0.00	-2920.92
WAG+G+F	16.46	5924.23	0.00	-2903.12
(some lines omitted)				
MtREV+G	402.75	6310.52	0.00	-3115.26
MtREV+I	584.89	6492.66	0.00	-3206.33
MtREV	616.99	6524.76	0.00	-3223.38

-----  
 \*: models sorted according to this column  
 -----

```

*****
Relative importance of parameters
*****
alpha      (+G):      0.20
p-inv      (+I):      0.00
alpha+p-inv (+I+G):  0.80
freqs      (+F):      0.47
*****
Model-averaged estimate of parameters
*****
alpha (+G):      1.67
p-inv (+I):      0.07
alpha (+I+G):   2.69
p-inv (+I+G):   0.05
  
```

The best model according to the AIC criterion is the WAG+I+G model, and the probability that it is the best AIC model is 0.49 (its Akaike weight). The second-best model is RtREV+I+G+F and since the AIC difference is 0.96 (with a 0.31 weight), this model is also a good candidate. RtREV+I+G+F has the highest likelihood, but it has more parameters (19 from the +F, and two from “+I+G”, plus the ones for the branch lengths) and consequently is penalized by the AIC, what situates WAG+I+G as the best-fit model. The worst models are the ones built upon the MtREV matrix.

Below the table of models, we can see the **relative importance of parameters**. There we find that including +I+G is very important (the two best models use this distribution). +G seems to be also important, but +I alone seems to describe poorly the evolution of these proteins. In this example, adding +F has some importance (0.47) because the model RtREV+I+G+F is the second best.

Below we see a **model-averaged estimate of parameters**. In this example the averaged alpha shape of the models +I+G has a value of 2.69. Why? The alpha of the models +I+G (mainly the top-ranking: WAG+I+G and RtREV+I+G+F) is averaged using the weight of those models. Note that both the importance and the averaged estimate of alpha are separated for models +G and models +I+G, given the interdependence of +I and +G parameters. The same stands for the models +I.

Let’s try now the **AICc** framework to see if the AIC model selection is affected by a not enough large sample size. If we select AICc at the bottom of the console a new window will appear, prompting us to select a criterion to determine the sample size. We start selecting, for example, the “Alignment length”. As a result, the following will be displayed in the console:

```

*****
Second-order AIC (AICc) framework
Sample size: Total number of characters (alignment length)
              = 113.00
*****
Best model according to second-order AIC: WAG+G
*****
Model          deltaAICc*   AICc       AICcw      -lnL
-----
WAG+I+G        0.00         5956.28    0.76      -2912.89
WAG+G          2.34         5958.61    0.24      -2916.53
Blosum62+I+G  9.84         5966.12    0.01      -2917.81
Blosum62+G    11.12        5967.39    0.00      -2920.92
CpREV+I+G     19.31        5975.58    0.00      -2922.54
...

```

Umm, now the RtREV+I+G+F and RtREV+G+F models disappear from the top of the ranking. The AICc tells us that the sample size isn't large enough for supporting the addition of the +F extra parameters in those models. Using this framework we could say that WAG+I+G is the best-fit model, and that the WAG+G model is also an interesting candidate.

What if the alignment length is an underestimation of the size of the sample? If we try other sample size criteria, we'll see that as sample size increases, the support given by AICc to more complex models is higher (the larger the sample size the most similar the behaviour of AICc compared to AIC is).

What about the BIC framework? We may have a better perspective of the scenario if we click the "overall comparison" button in the console, what results in a table like the following, where the ranking of models, the importance of parameters, and other statistics are compared under seven frameworks:

```

-----
Table: Weights(Ranking) of the different models under the different frameworks
-----
model          AIC          AICc-1       AICc-2       AICc-3       BIC-1       BIC-2       BIC-3
WAG+I+G        0.49(1)      0.76(1)      0.86(1)      0.86(1)      0.78(1)      0.74(1)      0.57(1)
RtREV+I+G+F    0.31(2)      0.00(19)     0.00(11)     0.04(3)      0.00(12)     0.00(12)     0.00(18)
RtREV+G+F      0.16(3)      0.00(16)     0.00(10)     0.02(4)      0.00(11)     0.00(11)     0.00(16)
WAG+G          0.04(4)      0.24(2)      0.13(2)      0.07(2)      0.22(2)      0.25(2)      0.42(2)
Blosum62+I+G  0.00(5)      0.01(3)      0.01(3)      0.01(5)      0.01(3)      0.01(3)      0.00(4)
WAG+I+G+F      0.00(6)      0.00(23)     0.00(13)     0.00(7)      0.00(16)     0.00(17)     0.00(23)
Blosum62+G    0.00(7)      0.00(4)      0.00(4)      0.00(6)      0.00(4)      0.00(4)      0.01(3)
(some lines omitted)
-----
Relative importance of
parameters     AIC          AICc-1       AICc-2       AICc-3       BIC-1       BIC-2       BIC-3
+G             0.20         0.24         0.13         0.09         0.22         0.25         0.43
+I             0.00         0.00         0.00         0.00         0.00         0.00         0.00
+I+G          0.80         0.76         0.87         0.91         0.78         0.75         0.57
+F            0.47         0.00         0.00         0.06         0.00         0.00         0.00
-----
Model-averaged estimate of
parameters     AIC          AICc-1       AICc-2       AICc-3       BIC-1       BIC-2       BIC-3
alpha (+G)    3.02         2.01         2.01         2.05         2.01         2.01         2.02
p-inv (+I)    0.00         0.07         0.06         0.00         0.07         0.07         0.07
alpha (+I+G)  2.68         3.10         3.10         3.10         3.10         3.10         3.09
p-inv (+I+G)  0.03         0.05         0.05         0.05         0.05         0.05         0.05
-----
AIC      : Akaike Information Criterion framework.
AICc-x   : Second-Order Akaike framework.
BIC-x    : Bayesian Information Criterion framework.
AICc/BIC-1: sample size as: number of sites in the alignment (113.0)
AICc/BIC-2: sample size as: Sum of position's Shannon Entropy over the whole alignment (169.2)
AICc/BIC-3: sample size as: align. length x num sequences x averaged Sh. Entropy (822.2)
-----

```

We can end to some conclusions: the empirical WAG matrix is clearly the one that fits best the family of L5 proteins. However, modifying the matrices with the observed amino acid frequencies (applying +F) allows RtREV models to better-fit the data compared to WAG. Since the use of these observed frequencies add extra parameters to the model, AIC, AICc and BIC interpretate this better-fit of RtREV models as an over-fit, and penalizes them consequently. Including a gamma distribution to account for different rates of change at different positions is always of some importance (ranging from 0.09 in AICc-3 to 0.43 in BIC-3). Including an invariable sites distribution alone is not. But both +G and +I together do it better.

## Known bugs

Many users have reported errors when running ProtTest under **Windows**. Such errors are related to filenames and file-paths conflicts and can be usually circumvented by placing the ProtTest input files (the alignment and, optionally, the tree) in a lower directory, such as "C:\".

## Program history

**Version 2.4 (September 2009):** Bug fixed in the reading of the proportion of invariable sites.

**Version 2.2 (August 2009):** Some new options added to the command-line version of the program. E.g. `-numcat`.

**Version 2.1 (June 2009):** Updated to a new release of Phylml.

**Version 2.0 (March 2009):** Major update. LG and DCmut models included. Updated to Phylml v3 version.

**Version 1.4 (July 2007):** HIVb and HIVw models added to ProtTest.

**Version 1.3 (January 2006):** Version-tracking renumbered according to the release of a new version.

**Version 1.2.16 (November 2005):** Minor aesthetic change: some information is printed to the console when ProtTest is launched in the console-mode.

**Version 1.2.12 (July 2005):** A bug in the "overall comparison" has been fixed (thanks to Marc Elliot).

**Version 1.2.10 (April 2005):** New model in ProtTest. MtArt is a replacement matrix for arthropod mitochondrial proteins. It has been estimated with Paml.

**Version 1.2.8 (February 2005):** bug corrected so ProtTest is now java 1.3 compatible.

**Version 1.2.6 (January 2005):** added the ability to specify the number of rate categories for the gamma distribution.

**Version 1.2.4 (January 2005):** MtMam matrix added to ProtTest and Phylml.

**Version 1.2.2 (December 2004):** some adjustments to the calculation of model-averaged parameters and their importance.

**Version 1.2.0 (November 2004):** models based on VT, CpREV, RtREV and Blosum62 added to Phylml and ProtTest.

**Version 1.0.4 (October 2004):** updated to new version of Phylml (2.4.1). ProtTest now checks for taxa name duplicates (this caused problems with Phylml).

**Version 1.0.2 (October 2004):** A problem with spaces in the path has been corrected. Some improvements in the console interface.

**Version 1.0 (October 2004):** First release of ProtTest! Version-tracking renumbered.

## BACKGROUND

### Models of protein evolution

Basically a model of protein evolution indicates the probability of change from a given amino acid to another over a period of time, given some rate of change. Although mechanistic models exist (Thorne and Goldman 2003), models of protein evolution are preferentially based on empirical matrices for computational and data-complexity reasons. These matrices are constructed based on large datasets consisting of many diverse protein families. The resulting matrices state the relative rates of replacement from one amino acid to another. The most common matrices, which are the ones included in ProtTest, are the **Dayhoff** (Dayhoff et al. 1978), **JTT** (Jones et al. 1992), **WAG** (Whelan and Goldman 2001), **mtREV** (Adachi and Hasegawa 1996), **MtMam** (Cao et al. 1998), **VT** (Muller and Vingron 2000), **CpREV** (Adachi et al. 2000), **RtREV** (Dimmic et al. 2002), **MtArt** (Abascal et al. 2007), **HIVb/HIVw** (Nickle et al. 2007), **LG** (Le and Gascuel 2008), and **Blosum62** (Henikoff and Henikoff 1992) matrices.

Conservation of protein function and structure imposes constraints on which positions can change and which cannot. This evolutionary information can be inferred from a multiple alignment but cannot be specified in a substitution matrix such as the empirical ones described below. Fortunately, there are some ways we can model these constraints: we can consider that a fraction of the amino acids are invariable (commonly indicated with a “+I” code in the name of the model) (Reeves 1992), we can consider some different categories of change (low, medium, high rate, etc), and assign each site a probability to belong to each of these categories (usually indicated by a “+G” code) (Yang 1993), or we can include both in the model (+I+G). Also, we can use as equilibrium amino acid frequencies those observed in the alignment at hand (indicated as “+F”) (Cao et al. 1994).

### Statistics for model selection: Akaike Information Criterion and others

For a more detailed background on model selection, the user is referred to (Posada and Crandall 2001) and Posada and Buckley (in press). Burnham and Anderson (2003) provide a very good description of the AIC framework and its use for model averaging (which they call multimodel inference).

The fit of a model of protein evolution ( $M$ ) to a given data set ( $D$ ), given a tree ( $T$ ) and branch lengths ( $B$ ) is measured by the likelihood function ( $L$ ):

$$L = P(D|M, T, B)$$

One could think that the best model is the one which results in the maximum likelihood, but this is not necessarily true: the more parameters the model includes, the higher its advantage in fitting better the data, but also the higher the variance for the parameter estimates. So how many parameters should the best model include?

One way to answer this question is by using the Akaike Information Criterion (AIC) (Akaike, 1973):

$$AIC = -2LnL + 2K$$

(LnL: log-likelihood; K: number of parameters). The model with lowest AIC is expected to be the closest model to the true model among the set of candidate models. Since AIC is on a relative scale, it is useful to present also the **AIC differences** (or deltaAIC). For the  $i$ th model, the AIC difference is:

$$\Delta_i = AIC_i - \min AIC$$

where  $\min AIC$  is the smallest AIC among all candidate models.

The AIC might not be accurate when the size of the sample is small compared to the number of parameters. For these cases, it is recommended to use a **second-order AIC** or corrected AIC (**AICc** in ProtTest; (Sugiura 1978)), which includes a penalty for cases where the sample size is small:

$$AICc = AIC + \frac{2K(K+1)}{n-K-1}$$

where  $n$  is the size of the sample (see below). If  $n$  is large with respect to  $K$ , the second term is negligible, and AICc behaves similar to AIC. The corrected AIC is recommended when the relation  $n/K$  is small (for example  $n/K < 40$ , being  $K$  the number of parameters of the most complex model among the set of candidate models).

ProtTest also calculates the **Bayesian Information Criterion (BIC; (Schwarz 1978))**, which is another measure of model fit. The BIC is considered a good approximation of the (very computationally demanding) Bayesian methods, and is formulated as:

$$BIC = -2LnL + K \log(n)$$

### Phylogenetics and sample size

What is the sample size of a protein alignment is very unclear. ProtTest offers different criteria for sample size determination:

- Alignment length (default).
- Number of variable sites.
- Shannon entropy summed over all alignment positions.

Shannon Entropy: (Shannon 1948). It's a way to measure the disorder or entropy.

$$ShEn = \sum_i P_i \log_2(P_i)$$

In our case if a position is completely conserved it takes the value of 0. If completely disordered (frequency of every aminoacid equals 1/20) it takes the value of 4.32.

- Number of sequences × length of the alignment × normalized Shannon's entropy

The normalized Shannon's entropy is calculated by summing the entropies over all positions, dividing this quantity by the number of positions, and dividing the resulting quantity by the maximum possible entropy (4.32).

- Number of sequences  $\times$  length of the alignment.
- User's provided size.

### **Akaike weights and the relative importance of parameters**

The AIC (or AICc, or BIC) differences can be used for calculating the Akaike weights:

$$w_i = \frac{\exp(-1/2\Delta_i)}{\sum_{r=1}^R \exp(-1/2\Delta_r)}$$

these weights can be interpreted as the probability that a model is the best AIC model.

#### Parameter importance

By summing the weights of the models that include a given parameter, for example the gamma distribution, we get the relative importance of such parameter:

$$w_+(\varphi_\alpha) = \sum_{i=1}^R w_i I_{\varphi_\alpha}(M_i),$$

where

$$I_\alpha(M_i) = \begin{cases} 1 & \text{if } \alpha \text{ is in model } M_i \\ 0 & \text{otherwise} \end{cases}$$

#### Model-averaged parameter estimates

We can also obtain an averaged estimation of any parameter by summing the different estimates for the models that contain such parameter after multiplying them by the Akaike weight of the corresponding model. For example, the model-averaged estimate of alpha ( $\varphi_\alpha$ ) for  $R$  candidate models would be:

$$\hat{\varphi}_\alpha = \frac{\sum_{i=1}^R w_i I_{\varphi_\alpha}(M_i) \varphi_\alpha}{w_+(\varphi_\alpha)}$$

where

$$w_+(\varphi_\alpha) = \sum_{i=1}^R w_i I_{\varphi_\alpha}(M_i),$$

and

$$I_{\varphi_\alpha}(M_i) = \begin{cases} 1 & \text{if } \varphi_\alpha \text{ is in model } M_i \\ 0 & \text{otherwise.} \end{cases},$$

## Credits and acknowledgements

ProtTest takes advantage of the **PAL library** (Drummond and Strimmer 2001) for manipulating alignments and trees. The core of the computation is carried out by the **Phyml** program (slightly modified to accomplish some requirements and to include additional models, (Guindon and Gascuel 2003)), which calculates the likelihoods and optimizes the parameters. Phyml is also used for calculating BioNJ trees. The code of ProtTest takes also benefit from other resources found at the WWW, as indicated in the source java code.

Very special thanks to Stephane Guindon (Phyml) and Matthew Goode (PAL) for being so helpful and patient.

**This work was financially supported from a grant for research in bioinformatics from the Fundación BBVA.**

Given that ProtTest uses intensively Phyml and PAL, we encourage users to cite these programs as well when using ProtTest:

- **[ProtTest]** Abascal F, Zardoya R, Posada D. 2005. ProtTest: Selection of best-fit models of protein evolution. *Bioinformatics* **21**:2104-2105.
- **[Phyml]** Guindon S, Gascuel O. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol.* **52**: 696-704.
- **[PAL]** Drummond A, Strimmer K. 2001. PAL: An object-oriented programming library for molecular evolution and phylogenetics. *Bioinformatics* **17**: 662-663.

## References

- Abascal, F., Posada, D., and Zardoya, R. 2007. MtArt: a new model of amino acid replacement for Arthropoda. *Mol Biol Evol* **24**: 1-5.
- Adachi, J., and Hasegawa, M. 1996. Model of amino acid substitution in proteins encoded by mitochondrial DNA. *J. Mol. Evol.* **42**: 459-468.
- Adachi, J., Waddell, P.J., Martin, W., and Hasegawa, M. 2000. Plastid genome phylogeny and a model of amino acid substitution for proteins encoded by chloroplast DNA. *J Mol Evol* **50**: 348-358.
- Cao, Y., Adachi, J., Janke, A., Paabo, S., and Hasegawa, M. 1994. Phylogenetic relationships among eutherian orders estimated from inferred sequences of mitochondrial proteins: instability of a tree based on a single gene. *Journal of Molecular Evolution* **39**: 519-527.
- Cao, Y., Janke, A., Waddell, P.J., Westerman, M., Takenaka, O., Murata, S., Okada, N., Paabo, S., and Hasegawa, M. 1998. Conflict among individual mitochondrial proteins in resolving the phylogeny of eutherian orders. *J Mol Evol* **47**: 307-322.
- Dayhoff, M.O., Schwartz, R.M., and Orcutt, B.C. 1978. A model of evolutionary change in proteins. In *Atlas of Protein Sequence and Structure*. (ed. M.O. Dayhoff), pp. 345-352. National Biomedical Research Foundation, Washington, DC.
- Dimmic, M.W., Rest, J.S., Mindell, D.P., and Goldstein, R.A. 2002. rtREV: an amino acid substitution matrix for inference of retrovirus and reverse transcriptase phylogeny. *J Mol Evol* **55**: 65-73.
- Drummond, A., and Strimmer, K. 2001. PAL: an object-oriented programming library for molecular evolution and phylogenetics. *Bioinformatics* **17**: 662-663.
- Gascuel, O. 1997. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol Biol Evol* **14**: 685-695.
- Gilbert, D. 2001.
- Guindon, S., and Gascuel, O. 2003. A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol* **52**: 696-704.
- Henikoff, S., and Henikoff, J.G. 1992. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A* **89**: 10915-10919.
- Jones, D.T., Taylor, W.R., and Thornton, J.M. 1992. The rapid generation of mutation data matrices from protein sequences. *Comp. Appl. Biosci.* **8**: 275-282.
- Le, S.Q., and Gascuel, O. 2008. An improved general amino acid replacement matrix. *Mol Biol Evol* **25**: 1307-1320.
- Maddison, W.P., and Maddison, D.R. 1992. *MacClade: analysis of phylogeny and character evolution*. Sinauer Associates Inc., Sunderland, Massachusetts, USA, pp. 398.
- Muller, T., and Vingron, M. 2000. Modeling amino acid replacement. *J Comput Biol* **7**: 761-776.
- Nickle, D.C., Heath, L., Jensen, M.A., Gilbert, P.B., Mullins, J.I., and Kosakovsky Pond, S.L. 2007. HIV-specific probabilistic models of protein evolution. *PLoS ONE* **2**: e503.
- Posada, D., and Buckley, T.R. 2004. Model Selection and Model Averaging in Phylogenetics: Advantages of AIC and Bayesian approaches over Likelihood Ratio Tests. *Systematic Biology* **53**: 793-808.
- Posada, D., and Crandall, K.A. 1998. MODELTEST: testing the model of DNA substitution. *Bioinformatics* **14**: 817-818.



- Posada, D., and Crandall, K.A. 2001. Selecting the best-fit model of nucleotide substitution. *Syst Biol* **50**: 580-601.
- Reeves, J.H. 1992. Heterogeneity in the substitution process of amino acid sites of proteins coded for by mitochondrial DNA. *J Mol Evol* **35**: 17-31.
- Schwarz, G. 1978. Estimating the dimension of a model. *Ann. Statist.* **6**: 461-464.
- Shannon, C.E. 1948. A mathematical theory of communication. *Bell System Technical Journal* **27**: 379-423 and 623-656.
- Sugiura, N. 1978. Further analysis of the data by Akaike's information criterion and the finite correction. *Comm. Statist. A-Theory. Meth* **7**: 13-26.
- Thorne, J.L., and Goldman, N. 2003. Probabilistic models for the study of protein evolution. In *Handbook of Statistical Genetics*. (ed. M.B. D.J. Balding, and C. Cannings), pp. 209-226. John Wiley & Sons, Ltd., Chichester, England.
- Whelan, S., and Goldman, N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Mol Biol Evol* **18**: 691-699.
- Yang, Z. 1993. Maximum-likelihood estimation of phylogeny from DNA sequences when substitution rates differ over sites. *Mol Biol Evol* **10**: 1396-1401.