Feedback — Model Selection

Help Center

You submitted this quiz on **Fri 2 Aug 2013 10:46 PM PDT**. You got a score of **14.00** out of **14.00**.

Overview

In this exercise you are going to investigate features of HIV-1 evolution. You will do this by analyzing a large set of env-genes from HIV-1, subtype B. specifically, the DNA sequences analyzed here correspond to a region surrounding the hypervariable V3 region of the gp120 protein.

Like other retroviruses, particles of HIV are made up of 2 copies of a single-stranded RNA genome packaged inside a protein core, or capsid. The core particle also contains viral proteins that are essential for the early steps of the virus life cycle, such as reverse transcription and integration. A lipid envelope, derived from the infected cell, surrounds the core particle. Embedded in this envelope are the surface glycoproteins of HIV: gp120 and gp41. The gp120 protein is crucial for binding of the virus particle to target cells, while gp41 is important for the subsequent fusion event. It is the specific affinity of gp120 for the CD4 protein that targets HIV to those cells of the immune system that express CD4 on their surface (*e.g.*, T-helper lymphocytes, monocytes, and macrophages).

The role gp120 plays in infection and the fact that it is situated on the surface of the HIV particle, means it is an obvious target for the immune response. That means that there may be a considerable selective pressure on gp120 for creating immune-escape mutants, where amino acids in the gp120 epitopes have been substituted. In this exercise you will construct a maximum likelihood tree that we will subsequently use to investigate whether you can detect such a selective pressure on parts of gp120, again using maximum likelihood methods.

One major goal with the exercise is to introduce you to statistically based methods for assessing the strength of evidence for a set of alternative hypotheses about some biological system of interest. The model selection method we will use is AIC (Akaike Information Criterion), based on which you will compute model probabilities. A second goal is to make you aware that phylogenetic analysis is not only about constructing trees, but that it is also a useful framework for analyzing biological questions more generally.

Specifically, you will

1. perform a multiple alignment of gp120 DNA sequences taking protein-level information into account (using `revtrans`).
2. select a suitable nucleotide substitution model (using `PAUP` and `modeltest`)

3. construct a phylogenetic tree (using `PAUP`).
4. try to detect positively selected sites in gp120 (using `PAML`).

Recipe for computing AIC values and model probabilities:

Later in today's exercise you will be asked to compute AIC values and model probabilities. Return to this section and follow the instructions when you need to do so.

1. Fit a set of models to your data, note the maximized log likelihoods (lnL) and the number of free parameters (K) for each model in the investigated set. The models you fit should represent a plausible and comprehensive set of hypotheses about your data.

2. Compute AIC for each of the models: **`AIC = -2 x lnL + 2K`**.
   For example: a model with `lnL = -2010` and `K = 5` will have `AIC = -2 x -2010 + 2 x 5 = 4030`.

3. Identify the model with the smallest AIC (this is the best model in the set). We will call the AIC for this model **"AIC_{min}"**.

4. Compute the "ΔAIC" values for each model: **$\Delta AIC = AIC - AIC_{min}$**
   For each model subtract the minimum AIC value. The best model will have a ΔAIC of zero. The rest of the models will have positive ΔAICs.

5. For each model compute the following quantity: **`numerator = exp(-0.5 x ΔAIC)`**
   For example, a model with `ΔAIC=4.2` will have `numerator = exp(-0.5 x 4.2) = exp(-2.1) = 0.1225`. Also compute the **sum** of the numerator values for all models.

6. Finally, the model probabilities for each model are found as: **`P(model) = numerator / sum`**
   For example, if `sum = 3.75` and a model has `numerator = 1.3`, then it has `P(model) = 1.3 / 3.75 = 0.35`

You may want to keep track of the computations by constructing a table along the following lines:

Model	K	lnL	AIC	Delta	Weight
TVM+I+G	9	-3553.50	7125.00	0.00	0.46
GTR+I+G	10	-3553.18	7126.36	1.36	0.23
TVM+G	8	-3555.33	7126.65	1.65	0.20
GTR+G	9	-3555.01	7128.02	3.02	0.10
K81uf+I+G	7	-3560.25	7134.51	9.51	0.00
TIM+I+G	8	-3559.52	7135.05	10.05	0.00

A good starting point for reading more about model selection and multimodel inference is this book:

- Model Based Inference in the Life Sciences: A Primer on Evidence, David R. Anderson.

Note that model probabilities can also be computed using Bayesian methods.

Getting started, description of data

- **Install modeltest program:**

```
wget http://wiki.bio.dtu.dk/~agpe/course_material/coursera/modeltest
```

```
chmod 755 modeltest
```

```
sudo mv modeltest /usr/bin/
```

  These commands fetch the modeltest program from the URL where I have placed it, sets file permissions so the program is executable by everyone, and finally moves the program to the proper directory (using administrator rights).

- **Create working directory, copy files:**

```
cd ~student
```

```
mkdir  modelselect
```

```
cd modelselect
```

```
cp ~/data/gp120.fasta ./gp120.fasta
```

```
cp ~/data/modelblock3.gorm ./modelblock3.gorm
```

```
cp ~/data/codeml.ctl ./codeml.ctl
```

- **Have a look at the DNA data file:**

```
nedit gp120.fasta &
```

The file contains several DNA sequences from HIV-1, subtype B. The sequences are approximately 500 bp long, and correspond to a region surrounding the hypervariable V3 region in the gene encoding gp120. Close the `nedit` window when you've had a look.

Question 1

Analysis of viral data set: alignment of coding DNA.

DNA sequences are a lot less informative than protein sequences and for this reason it is always preferable to align coding DNA in translated form. The simple fact that proteins are built from 20 amino acids while DNA only contains four different bases, means that the 'signal-to-noise

ratio' in protein sequence alignments is much better than in alignments of DNA. Besides this information-theoretical advantage, protein alignments also benefit from the information that is implicit in empirical substitution matrices such as BLOSUM-62. Taken together with the generally higher rate of synonymous mutations over non-synonymous ones, this means that the phylogenetic signal disappears much more rapidly from DNA sequences than from the encoded proteins. It is therefore preferable to align coding DNA at the amino acid level.

However, in the context of molecular evolution, DNA alignments retain a lot of useful information regarding silent mutations. Especially the **ratio** between silent and non-silent substitutions is informative. We would therefore like to construct a multiple alignment at the DNA level, but using information at the protein level, and the RevTrans server does exactly that.

RevTrans takes as input an unaligned set of DNA sequences, automatically translates them to the equivalent amino acid sequences, constructs a multiple alignment of the protein sequences, and finally uses the protein alignment as a template for constructing a multiple DNA alignment that is in accordance with the protein alignment. This also means that gaps are always inserted in groups of three so reading frames are kept in order. That is important if you want to analyze selection, as we will in this exercise.

- **Construct RevTrans alignment:**

  Open RevTrans server page: http://www.cbs.dtu.dk/services/RevTrans/

  On the RevTrans page: Choose the file `gp120.fasta` as input (or copy and paste the sequence into the sequence window)

  Click the "Submit query" button

  When the alignment is done you may have to click link named "here" to go to results page

  Download DNA alignment, by right-clicking the link for "Download result", and choosing "Save link as..." (save file under the name `gp120align.fasta` and make sure to save the file in the directory `modelselect`).

- **Convert alignment to NEXUS format:**

  Convert resulting alignment to NEXUS format using the EBI ReadSeq server

  Save NEXUS file in the `modelselect` directory under the name `gp120.nexus` (right-click the "Download" button, and "Save link as...")

- **Convert alignment to PHYLIP format:**

  Also convert alignment to Phylip4 format (not Phylip3.2 format). Make sure you save the file under the name "gp120.phy" in the

`modelselect` directory.

Now, open the file in nedit:

```
nedit gp120.phy &
```

Alter the first line of the alignment file so it looks like this:

```
 39    510 I
```

(i.e., add a space and the letter "`I`" to the end of the first line).

Important: Save the file and exit `nedit`.

Getting data files into formats that can be understood by various programs is something that the average bioinformatician spends a LOT of time doing... Here, we added an upper-case "i" so the PAML programs (which we will use later) will understand that the file is in "interleaved" format.

Selection of substitution model using PAUP and modeltest

As part of the present analysis we are going to build a phylogenetic tree based on the DNA alignment constructed above. We will construct the tree using maximum likelihood, but to do that we first have to decide which substitution model we want to use. Specifically, we are interested in using the model that best describes our data without having more parameters than strictly necessary (thus avoiding overfitting). We will investigate this issue by fitting a set of 56 different models to our data and then selecting one with a reasonable balance between model complexity and data fit.

- **Start PAUP, load data set:**

```
paup gp120.nexus
```

- **Fit 56 models, record negative log-likelihoods in scorefile:**

```
execute modelblock3.gorm
```

This causes PAUP to execute the commands present in the file `modelblock3.gorm`: first a neighbor joining tree is constructed using the Jukes and Cantor model. Then the tree is fixed and used as the basis for fitting a set of 56 different models to the data. For each model, the estimated model parameters and the negative log-likelihood are written to a file named "model.scores". In addition to varying sets of substitution rate parameters, some of these models also include extra parameters that take into account the presence of different rates between sites. This is done in two ways: (1) by fitting a gamma distribution of rates, and (2) by allowing for a proportion of constant ("invariable") sites.

Wait until PAUP is done fitting all 56 models (this will take a little while depending on your computer - you can tell PAUP has finished since the prompt "**paup>** " will re-appear). Now, quit the program by typing:

```
quit
```

(Don't worry if the program makes a remark about unsaved trees - in this part of the exercise we are not interested in the tree, we just want to know how well the 56 different models fit our data set, i.e., we only want their log likelihood values).

- **Inspect result, manually check model probabilities for three models:**

```
nedit model.scores &
```

For each model this file lists the negative log-likelihood and all estimated model parameters (excluding branch lengths). The results for each model takes up two lines in the file - a header line, and a line with parameter values. The first parameter for each model is the tree, which is the same in all the investigated cases here, and is simply called "Tree 1". The second value given for each model is its negative log likelihood. The remaining columns will give values for any additional free parameters in the file (e.g., nucleotide frequencies, gamma shape parameter, etc). This file is difficult to read because it is meant to be parsed by a computer so don't despair!

- **Manually compute model probabilities for three substitution models:**

  Use AIC-based model probabilities to investigate which of the following three substitution models are best at describing how the sequences have evolved:

  - Jukes and Cantor with fraction of invariant sites (JC+I)
  - Jukes and Cantor with gamma-distributed rates over sites (JC+G)

- Jukes and Cantor with invariant sites and gamma-distributed rates (JC+I+G)

Before you can do the computation you need to know the log likelihood and the number of parameters for each model. First, locate and note the log likelihood values for the JC+I, JC+G, and JC+I+G models in the `model.scores` file and note the values. As mentioned above, each model takes up two lines in the file, JC+I is model number 2, JC+G is number 3, and JC+I+G is number 4 in the file, so their outputs are found on lines 3+4, 5+6, and 7+8).

Make sure to get the signs right: the values reported in the file are *[Math Processing Error]* values, so you will need to reverse the sign to get the *[Math Processing Error]* (the *[Math Processing Error]* values you write down should be negative). Use the following values for the number of free parameters (K) for the three models:

- JC+I: K=2
- JC+G: K=2
- JC+I+G: K=3

Now, use the recipe above to compute AIC values and model probabilities.

- **Question:** For the JC+I model, report: AIC, *[Math Processing Error]*AIC, *[Math Processing Error]* (model probability). Enter the three values in that order, separated by spaces, using at least 4 significant digits.

You entered:

```
7496.65343804 33.03760028 0.000000044
```

Your Answer		Score	Explanation
7496.65343804	✔	0.33	
33.03760028	✔	0.33	
0.000000044	✔	0.33	
Total		1.00 / 1.00	

Question 2

For the JC+G model, report: AIC, *[Math Processing Error]*AIC, *[Math Processing Error]* (model probability). Enter the three values in that order, separated by spaces, using at least 4 significant digits.

You entered:

7463.61583776 0 0.65972

Your Answer		Score	Explanation
7463.61583776	✔	0.33	
0	✔	0.33	
0.65972	✔	0.33	
Total		1.00 / 1.00	

Question 3

For the JC+I+G model, report: AIC, *[Math Processing Error]*AIC, *[Math Processing Error]* (model probability). Enter the three values in that order, separated by spaces, using at least 4 significant digits.

You entered:

7464.93992392 1.32408616 0.34028

Your Answer		Score	Explanation
7464.93992392	✔	0.33	
1.32408616	✔	0.33	
0.34028	✔	0.33	
Total		1.00 / 1.00	

Question 4

Based on the model probabilities: wich model has more support?

Your Answer		Score	Explanation
◉ JC+G	✔	1.00	
○ JC+I			
○ JC+I+G			
Total		1.00 / 1.00	

Question 5

- **Use `modeltest` program to select best model:**

What you just did manually for JC+I, JC+G and JC+I+G, the program `modeltest` does automatically for the full set of 56 fitted models. Specifically, it uses the list of negative log likelihoods in the score-file to compute AIC and model probabilities, and uses this to select the model that best fits the sequence data.

```
modeltest < model.scores > modeltest.results
```

This command executes the modeltest program using model.scores as the input file and modeltest.result as the output file.

- **Inspect result:**

```
nedit modeltest.results &
```

This file contains the output of the modeltest program. The first part of the file shows model selection results based on the so-called likelihood ratio test method. Scroll past that until you find the section containing the AIC-based results.

- **Question:** What model was selected by `modeltest` based on the AIC values?

Your Answer		Score	Explanation
◉ TVM+I+G	✔	1.00	
○ TVM+G			
○ GTR+I+G			
○ JC+I+G			
○ K81uf+I+G			
○ GTR+G			
Total		1.00 / 1.00	

Question 6

Construction of phylogenetic tree using PAUP

In the model.results file, you should now scroll down to the "PAUP* Commands Block" section, right beneath the section giving the details of the selected model.

In this section, find the line giving PAUP commands that will implement the selected model. The command is enclosed between "BEGIN PAUP" and "END;" and should look something like this:

```
"Lset Base=(0.4064, ..."
```

You will need to copy this command to a PAUP session in the next step.

- **Start PAUP:**

  ```
  paup
  ```

  Above you used modeltest to select the most suitable substitution model for the present data set. You will now use this model to construct a maximum likelihood tree. You will again use PAUP for this purpose.

- **Load alignment:**

  ```
  execute gp120.nexus
  ```

- **Set tree-building criterion to maximum likelihood:**

  ```
  set criterion=likelihood
  ```

- **Set model parameters to winning estimates:**

  Above you located a set of lines in the file `modeltest.results` giving a PAUP command that sets the model parameters to the estimates that were found using the winning model. Copy and paste this `lset` command (without the BEGIN and END parts) into the window where PAUP is running.

  > *PASTE LSET COMMAND FROM MODELTEST RUN HERE*

- **Find best tree using selected model:**

  Still in the PAUP-window, enter the following command:

  ```
  hsearch swap=tbr start=nj rseed=973
  ```

  This command causes PAUP to perform a heuristic search for the best maximum likelihood tree. Once an initial tree has been constructed, the heuristic search proceeds by rearrangements of the "tree bisection and reconnection" type (TBR). We are using the model selected by modeltest, AND the parameter estimates found by modeltest on that model. You could also have chosen to simply estimate all the model parameters as part of this step (i.e., at the same time as finding the best tree), but fixing them improves speed tremendously. Findind the best tree should take a few minutes.

- **Save best tree to file:**

  ```
  savetrees format=newick brlens=yes file=gp120tree.phy from=1 to=1
  ```

- **Quit program:**

  ```
  quit
  ```

- **Question:**

  What is the negative log likelihood of the tree you just found? (Report the positive number that is output by PAUP, using at least 4 significant digits).

You entered:

3528.9390

Your Answer		Score	Explanation
3528.9390	✔	1.00	
Total		1.00 / 1.00	

Question 7

- **Have a look at the tree:**

  You have now produced an unrooted tree of the HIV sequences and saved it in the file `gp120tree.phy`. Note that in this exercise we will not be interested in the tree as such - our focus is instead on finding positive selection on a subset of codon positions and the tree is just something we need in order to be able to fit the different codon models to the data. If you want to see the tree, you can do so with the following command:

  ```
  figtree gp120tree.phy &
  ```

  There is no meaningful root placed in this tree, so you may want to choose the unrooted view (the third icon in the Layout section of the figtree window). Close the figtree window when you have had a look

Detection of positively selected sites in gp120

There is much more to phylogenetic analyses than merely reconstructing trees. One interesting result of probabilistic methods, is that the parameters of a model will have their values determined as part of the optimization procedure. This means that once such a model has been

fitted to the data, it is possible to investigate these estimated parameter values to learn features about the evolutionary history of the sequences under investigation. In the present example we will focus on investigating whether we can find positively selected sites in our data set, defined as sites where the *[Math Processing Error]* ratio is larger than 1. We do that by using a codon substitution model where the *[Math Processing Error]* ratio is one of its parameters.

A further strength of the probabilistic approach is that you get a probabilistic measure of how well any model fits the data. This means you can use a stringent approach to determine which model fits the data best. In this framework one uses likelihoods (probabilities of data given model) to determine which model fits the data best. As you saw above, it is for instance possible to compute AIC values and model probabilities from the likelihood values of fitted models, Since each model essentially corresponds to a hypothesis about the evolutionary history of the data, we can thus use a stringent statistical approach to decide which hypothesis best describes our data.

In outline, you will now use the following steps to investigate whether there is any evidence for positively selected codons in your data set:

1. Fit model M1, which assumes there are two classes of codons in the sequence: some with *[Math Processing Error]*, some with *[Math Processing Error]*.
2. Fit model M2, which assumes 3 distinct classes of codons: two with *[Math Processing Error]* ratios as for M1, and one extra class with *[Math Processing Error]*.
3. Assess the strength of evidence for the two models using AIC-based model probabilities
4. If M2 is better: identify the positively selected codons

- **Inspect the parameter file:**

```
nedit codeml.ctl &
```

The file "codeml.ctl" contains several settings that are relevant for running the program **`codeml`**:

```
seqfile =  gp120.phy:  name of alignment file
treefile =  gp120tree.phy: name of tree file
seqtype = 1: tells the program that our data consists of coding DNA.
NSsites = 1 2 : tells the program to analyze models M1 and M2.
cleandata = 1: tells the program to ignore positions with gaps.
```

The settings entered by us will cause codeml to analyze two hypotheses about *[Math Processing Error]* ratios. M1 says there are two classes of codons with different *[Math Processing Error]* ratios in the sequence: one class with *[Math Processing Error]* (codons under purifying or negative selection), and one class with *[Math Processing Error]* (no selection - neutrally evolving sites). M2 says there are 3 distinct *[Math Processing Error]* ratios for different sites in the sequence: one class with *[Math Processing Error]*, one class with *[Math Processing Error]* (these are the same type of classes as for M1), and one class with *[Math Processing Error]* (corresponding to sites under positive selection). The value of the *[Math Processing Error]* ratios (for those classes that have *[Math Processing Error]* or *[Math Processing Error]*), the fraction of sites belonging to each class, and the position of sites belonging to each class, are unknown at first and will be determined during the analysis.

- **Start the analysis**

```
codeml
```

This will start the codeml program using the settings in the file codeml.ctl. Depending on your computer, this will take some minutes to finish. (You may be able to see how the optimization procedure results in progressively better fits: the likelihood increases, meaning that negative log-likelihood decreases, as the fit improves).

- **Inspect result file**

Wait for the run to finish, and then look at the result file:

```
nedit selection.results &
```

This file contains a wealth of information concerning your analysis. The top part of the file gives an overview of your sequences, codon usage and nucleotide frequencies. You can ignore this information for now, and move on to the interesting part, namely the model likelihoods and parameter values:

- **Find likelihood, and number of free parameters for model M1:**

```
Search ==> Find... ==> enter "Model 1" and click Find
```

You are now in the region of the result file where the model likelihoods and parameter estimates are noted. Now, locate a line that looks a

bit like the one shown below:

```
lnL(ntime: 72  np: 74):  -4242.470345     +0.000000
```

Identify the number of "free parameters", *[Math Processing Error]*, used in model M1: This is indicated by "np", and is 74 in the example shown above (most of these parameters are branch lengths in the tree; specifically, the number of branch length parameters is indicated by "ntime", and is 72 in this example). Also note the log-likelihood of the fitted model. This is the number right after the parenthesis, and is -4242.470345 in the example here.

- **Question:**

  What are the values of *[Math Processing Error]* and *[Math Processing Error]* for model M1? (Enter the answers separated by a space, using at least 4 significant digits for the *[Math Processing Error]*)

You entered:

73 -3340.794980

Your Answer		Score	Explanation
73	✔	0.50	
-3340.794980	✔	0.50	
Total		1.00 / 1.00	

Question 8

- **Find *[Math Processing Error]* ratios and codon class proportions for model M1:**

  Scroll down a few lines until you get to a small table similar to this:

```
dN/dS for site classes (K=2)
p:   0.75111  0.24889
w:   0.06583  1.00000
```

This gives a summary of the *[Math Processing Error]* ratios that were found in the data set. The line starting *[Math Processing Error]*: lists the two *[Math Processing Error]* ratios that were found (in this case 0.06583 and 1.00000 - the last one was pre-specified by us as part of the model and was therefore not a free parameter). The line starting *[Math Processing Error]*: gives the proportion of codon sites belonging to each of the *[Math Processing Error]* ratio classes (in the example above approximately 75% belong to the first class , while 25% of all sites belong to the class having *[Math Processing Error]*).

- **Question:** What are the *[Math Processing Error]* value (*[Math Processing Error]*) and proportion (*[Math Processing Error]*) of sites for both classes? Enter the values separated by spaces, in the following order: *[Math Processing Error]*, *[Math Processing Error]*, *[Math Processing Error]*, *[Math Processing Error]*

You entered:

0.55706 0.08444 0.44294 1.00000

Your Answer		Score	Explanation
0.55706	✔	0.25	
0.08444	✔	0.25	
0.44294	✔	0.25	
1.00000	✔	0.25	
Total		1.00 / 1.00	

Question 9

- **Find likelihood, and K for model M2:**

  Scroll past the M1 output until you get to the results for model M2.

- **Question:**

  What are the values of *[Math Processing Error]* and *[Math Processing Error]* for model M2? (Enter the answers separated by a space, using at least 4 significant digits for the *[Math Processing Error]*)

You entered:

75 -3324.531002

Your Answer		Score	Explanation
75	✔	0.50	
-3324.531002	✔	0.50	
Total		1.00 / 1.00	

Question 10

- **Find *[Math Processing Error]* ratios and codon class proportions for model M2:**

  Now, scroll down a few lines until you get to a small table similar to the one you examined for M1 before. For this model there are 3 separate classes of codons.

- **Question:** What are the *[Math Processing Error]* value (*[Math Processing Error]*) and proportion (*[Math Processing Error]*) of sites for all three classes? Enter the values separated by spaces, in the following order:

  [Math Processing Error], *[Math Processing Error]*, *[Math Processing Error]*, *[Math Processing Error]*, *[Math Processing Error]*, *[Math Processing Error]*

You entered:

0.51488 0.08490 0.37597 1.00000 0.10916 3.03865

Your Answer		Score	Explanation
0.51488	✔	0.17	
0.08490	✔	0.17	
0.37597	✔	0.17	
1.00000	✔	0.17	
0.10916	✔	0.17	
3.03865	✔	0.17	
Total		1.00 / 1.00	

Question 11

- **Assess strength of evidence for models M1 and M2:**

  M2 will always have a better (higher) log-likelihood than model M1 because M2 has more free parameters, and M1 is nested within M2.

You should now use the recipe given above to compute AIC values and model probabilities for M1 and M2.

- **Question:**

  For the M1 model, report: AIC, *[Math Processing Error]*AIC, *[Math Processing Error]* (model probability). Enter the three values in that order, separated by spaces, using at least 4 significant digits.

You entered:

6827.58996 28.527956 .0000006386

Your Answer		Score	Explanation
6827.58996	✔	0.33	
28.527956	✔	0.33	
.0000006386	✔	0.33	
Total		1.00 / 1.00	

Question 12

For the M2 model, report: AIC, *[Math Processing Error]*AIC, *[Math Processing Error]* (model probability). Enter the three values in that order, separated by spaces, using at least 4 significant digits.

You entered:

6799.062004 0 0.99999936

Your Answer		Score	Explanation
6799.062004	✔	0.33	
0	✔	0.33	
0.99999936	✔	0.33	
Total		1.00 / 1.00	

Question 13

Is M2 better than M1?

Your Answer		Score	Explanation
Yes	✔	1.00	
No			
Total		1.00 / 1.00	

Question Explanation

If M2 has more support (larger model probability) than M1, then we have statistical evidence that selection has been acting on these HIV sequences. This selective pressure most likely originates from the immune system.

Question 14

- **Examine list of positively selected sites**

  If your M2 is clearly better than M1 (we firmly believe it should be if you did things according to our instructions...), then you have evidence for the existence of positively selected sites in the gp120 gene. That's not bad for a few hours of work! Now, scroll down to the end of the result file and locate a list similar to this one (note: This is the "Bayes Empirical Bayes" table, not the "Naive Empirical Bayes" table. It is not important what the distinction is in this context):

```
Bayes Empirical Bayes (BEB) analysis
Positively selected sites

             Prob(w>1)     mean w

    25 A 0.959*        3.133 +- 0.769
    27 P 0.906         2.990 +- 0.877
    56 K 0.987*        3.197 +- 0.687
    59 V 0.915         3.032 +- 0.873
    78 R 0.637         2.351 +- 1.129
    88 K 0.573         2.148 +- 1.077
    95 V 0.925         3.046 +- 0.843
    ...
```

  This gives you a list of which residues (if any) that were found to belong to the positively selected *[Math Processing Error]*-class. Also listed is the probability that the site really is in the codon class where *[Math Processing Error]*, and a weighted average of the *[Math Processing Error]* at the site. Using only DNA sequences you have now identified likely epitopes on the gp120 protein.

  Question:

  List all sites having more than 95% probability of belonging to the positively selected class (enter just the site numbers, in numerical order, separated by spaces).

You entered:

14 16 84 113 119

Your Answer		Score	Explanation
14	✔	0.20	
16	✔	0.20	
84	✔	0.20	
113	✔	0.20	
119	✔	0.20	
Total		1.00 / 1.00	