Detecting positive selection

During this practical we will look for evidence of past selective pressures on a candidate gene that could be linked with the adaptation of clownfishes.

In order to test for positive selection, the *codeml* executable available in the PAML4.9i or PAML4.8a package (!! do not download the PAML-X one!!) will be used (**here again copy the executable where you have your data file**), and different tests (see below for details) will be performed.

1 Input files

codeML requires three different input files

1. one containing the sequences in **sequential phylip** format
2. another with the species tree
3. a last one (named codeml.ctl) containing the parameters of the model

These three files must be in the same folder with a copy of the programe. For practical reasons, create a different folder for each model, and put a copy of each of the three input files in it.

codeML is **very picky** concerning the input file. A *sequential* format is required and the sequence names must have **at least** two spaces before the start of the protein coding sequence.

For some of the branch models, you need to specify one or several branches expected to exhibit different selective pressures. To do so, a "#1" must be added in the tree file after the parenthesis representing the branches of interest. You need to do that in a **text editor** and find the ")" corresponding to the node of interest.

The file codeml.ctl contains all the settings necessary to run codeml. Just copy codeml.ctl from the paml software and paste it in the folders you created for your different models. This file is a text file and you need to edit it using a text editor. The first two lines must be changed since they contain the name of your sequences and tree files. Then the codon model based on the HKY85 model uses different frequencies for each codon (set the F3X4 option in codeml. It should be the default option, but check). Most of the other lines do not need to be changed. The *model* and *NSsites* options need to be set to specify the model you want to use. *model* must be set to 0 in case you want to use a *site model*. For a branch model, this must be changed to 2. If *NSsites* is set to 0, only one ω value will be calculated. If *NSsites* is 1, two ω values (purifying or neutral selection) will be estimated. In order to allow some sites to be under positive selection, *NSsites* must be set to 2 or 3. For some advanced models, other parameters must be changed but for a first contact with PAML, they can be kept to their default value.

2 Tests for positive selection

Today, we will use the two positive selection tests. The different models must be compared using a *likelihood ratio test* (LRT).

1. The first test compares models M1a and M2a. M1a is a site model without any site under positive selection. Codons are attributed either to the first class with ω fixed to one or to the second class with an ω estimated but smaller than one. M2a, an other site model, allows some sites to be under positive selection. In this model, sites are attributed to three different classes ($0 < \omega < 1$; $\omega = 1$; $1 < \omega$).
2. The second test compares the model A1 with and without ω fixed to 1. This model is a branch-site model and allows some sites to have an ω greater than one in some branches. Sites are attributed to four distinct classes. In the first class, ω is identical for the background branches (those not supposed to exhibit positive selection) and the foreground branch (the one of interest where positive selection is expected to have occurred). This ω is estimated and is smaller than one. In the second class again, ω is equal in foreground and background branches but is equal to one. The third class contains sites under purifying selection in the background branches but under positive selection (ω estimated and greater than one) in the foreground branch. In the last class, sites are under neutral selection in background branches and under positive selection in the foregroundbranch.

3 Options in *codeml*

The options to be set in the *codeml.ctl* file for the three models are the following:

Name	model	NSsites
M1a	0	1
M2a	0	2
A1	2	2

4 Running codeml

1. Download the data on the rhodopsin gene (Clownfish_rh1.fasta in the `data` folder in Switchdrive) sets and transform the fasta format into phylip. Use `R` or `SeaView` to do that as last week.

   Essential: open the phylip file you just created in a text editor and check that each sequence name is followed by 2 spaces.

2. reconstruct a phylogenetic tree for this data using PhyML as we did in the first practical.

3. root the tree using `FigTree` or `SeaView` with the species *Chromis cyanea*. Save the rooted tree for the next analyses.

4. Add, manually using a text editor, a "#1" at the ancestral node of the clownfish. We will test if positive selection occurs along the branch leading to the clownfish.

5. Create different codeml parameter files (i.e. codeml.ctl) in different folders for

   (a) the two site models M1a (null) and M2a (alternative)

   (b) the two branch-site models based on model A1 (null with $\omega = 1$, alternative with ω estimated)

   See previous section to know how to change the model and NSsites options in codeml.ctl to represent these three models.

6. For model A1, the branch suspected to exhibit particular past selective pressures is the one leading to the clownfish lineage (root the tree with *Chromis cyaneae* and annotate the branch leading to the clownfishes).

7. Once the analysis finished, results are mainly present in file *mlc*. Explore it. Search which proportions of codons has been attributed to the different ω classes and look at the estimations of ω.

8. Estimate the branch lengths with PhyML on the RH1 alignment using the HKY85 model

   ```
   ./phyml -i clownfish_rh1.phy -q -m HKY85 -a 1000 -v 0 -o lr \
       --inputtree clownfish_rh1_GTR_G_I.tree --run_id HKY85
   ```

   (I assume here that you saved the initial tree for the rh1 gene into the file names `clownfish_rh1_GTR_G_I.tree`)

   Are the branch lengths estimated by PhyML and codeml the same? Why?

9. For M2a and A1 models, look at the localisation of the positively selected sites on the protein. Do the same sites belong to the same ω classes with both models?

10. Once all your models are done, compare them through LRT. Beware of the degrees of freedom you must use. Which one is significantly better? What are biologically the differences between the two tests? What can you conclude about the selective pressures undergone by your gene? Why are you not allowed to compare models M2a and A1?

11. Repeat the same analyses on the following genes (again in the data folder on Switchdrive), which were found to be under positive selection at the onset of the clownfish radiation (Marcionetti et al. 2019):

    - clownfish_sic9a6.phy
    - clownfish_snai2.phy
    - clownfish_tbx2.phy