From: **Evosite3D** noreply+feedproxy@google.com
Subject: Evosite3D
Date: 4 October 2021 at 03:48
To: birkeland.siri@gmail.com

Evosite3D

Identifying positive selection in genomic sequences

Posted: 30 May 2013 09:23 AM PDT

In this post, I will make a short tutorial on one of my favourite programs, CodeML, which is definitely not the easiest to use.

Theoretical principles:

The selective pressure in protein coding genes can be detected within the framework of comparative genomics. The selective pressure is assumed to be defined by the **ratio (ω) dN/dS**. dS represents the synonymous rate (keeping the amino acid) and dN the non-synonymous rate (changing the amino acid). In the absence of evolutionary pressure, the synonymous rate and the non-synonymous rate are equal, so the dN/dS ratio is equal to 1. Under purifying selection, natural selection prevents the replacement of amino acids, so the dN will be lower than the dS, and dN/dS < 1. And under positive selection, the replacement rate of amino acid is favoured by selection, and dN/dS > 1.

CodeML and substitutions models:

CodeML is a program from the package **PAML**, based on Maximum Likelihood, and developed in the **lab of Ziheng Yang**, University College London.
It estimates various parameters (Ts/Tv, dN/dS, branch length) on the codon (nucleotide) alignment, based on a predefined topology (phylogenetic tree).

Different codon models exist in CodeML. The model 0 estimates a unique dN/dS ratio for the whole alignment. Not really interesting, except to define a null hypothesis to test against. The branch models estimate different dN/dS among lineages (ie **ASPM, a gene expressed in the brain of primates**). The site models estimate different dN/dS among sites (ie in the **antigen-binding groove of the MHC**). The **branch-site models** estimate different dN/dS among sites and among branches. It can detect episodic evolution in protein sequences, as in the **interactions between chains in the avian MHC**. In my opinion, this is the most powerful application and this is the one used in the **Sectome database** (to which I contributed during my PhD).

First, we have to define the branch where we think that position could have occurred. We will call this branch the "foreground branch" and all other branches in the tree will be the "background" branches. The background branches share the same distribution of ω = dN/dS value among sites, whereas different values can apply to the foreground branch.

To compute the likelihood value, two models are computed: a null model, in which the foreground branch may have different proportions of sites under neutral selection to the background (i.e. relaxed purifying selection), and an alternative model, in which the foreground branch may have a proportion of sites under positive selection.
As the alternative model is the general case, it is easier to present it first.

Four categories of sites are assumed in the branch-site model:

Sites with identical dN/dS in both foreground and background branches:
K0 : Proportion of sites that are under purifying selection (ω0 < 1) on both foreground and background branches.
K1 : Proportion of sites that are under neutral evolution (ω1 = 1) on both foreground and background branches.

Sites with different dN/dS between foreground and background branches:
K2a: Proportion of sites that are under positive selection (ω2 ≥ 1) on the foreground branch and under purifying selection (ω0 < 1) on background branches.
K2b: Proportion of sites that are under positive selection (ω2 ≥ 1) on the foreground branch and under neutral evolution (ω1 = 1) on background branches.

For each category, we get the proportion of sites and the associated dN/dS values.

In the null model, the dN/dS (ω2) is fixed to 1:

Sites with identical dN/dS in both foreground and background branches:
K0 : Sites that are under purifying selection (ω0 < 1) on both foreground and background branches.
K1 : Sites that are under neutral evolution (ω1 = 1) on both foreground and background branches.

Sites with different dN/dS between foreground and background branches:
K2a: Sites that are under neutral evolution (ω2 = 1) on the foreground branch and under purifying selection (ω0 < 1) on background branches.
K2b: Sites that are under neutral evolution (ω2 = 1) on the foreground branch and under neutral evolution (ω1 = 1) on background branches.

For each model, we get the log likelihood value (lnL1 for the alternative and lnL0 for the null models), from which we compute the Likelihood Ratio Test (LRT).
The 2×(lnL1-lnL0) follows a χ^2 curve with degree of freedom of 1, so we can get a p-value for this LRT.

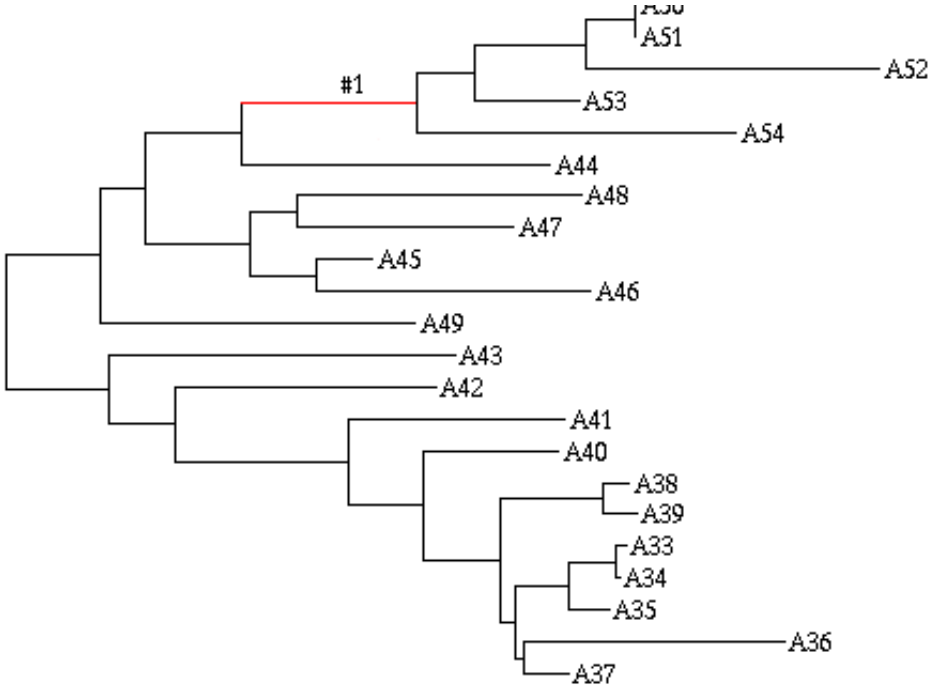
Let's go in details.

File Preparation:

We need four files to run CodeML:

1) The multiple nucleotide (CDS) alignment, in PHYLIP format. CodeML will strictly remove any position that contains at least one gap or an unknown "N" nucleotide: **TF105351.Eut.3.phy**

2) The phylogenetic tree in newick format, with the branch of interest specified by "#1" (You can view it with **NJplot** or **FigTree**): **TF105351.Eut.3.53876.tree**



3) A command file where all parameters to run CodeML under the alternative model are specified: **TF105351.Eut.3.53876.ctl**

4) A command file where all parameters to run CodeML under the null model are specified: **TF105351.Eut.3.53876.fixed.ctl**

Execute CodeML

Run command file (alternative model):

We estimate the Ts/Tv ratio (fix_kappa = 0) and the dN/dS (fix_omega = 0). The branch-site model is specified by setting the model parameter to 2 (different dN/dS for branches) and the NSsites value to 2 (which allows 3 categories for sites: purifying, neutral and positive selection).

```
  seqfile = TF105351.Eut.3.phy           * sequence data file name
 treefile = TF105351.Eut.3.53876.tree    * tree structure file name
  outfile = TF105351.Eut.3.53876.mlc     * main result file name


    noisy = 9   * 0,1,2,3,9: how much rubbish on the screen
  verbose = 1   * 1: detailed output, 0: concise output
  runmode = 0   * 0: user tree;  1: semi-automatic;  2: automatic
                * 3: StepwiseAddition; (4,5):PerturbationNNI; -2: pairwise

  seqtype = 1   * 1:codons; 2:AAs; 3:codons-->AAs
CodonFreq = 2   * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table
    clock = 0   * 0: no clock, unrooted tree, 1: clock, rooted tree
```

```
      aaDist = 0   * 0:equal, +:geometric; -:linear, {1-5:G1974,Miyata,c,p,v}
       model = 2   * models for codons:
                   * 0:one, 1:b, 2:2 or more dN/dS ratios for branches
     NSsites = 2   * 0:one w; 1:NearlyNeutral; 2:PositiveSelection; 3:discrete;
                   * 4:freqs; 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;10:3normal
       icode = 0   * 0:standard genetic code; 1:mammalian mt; 2-10:see below
       Mgene = 0   * 0:rates, 1:separate; 2:pi, 3:kappa, 4:all
   fix_kappa = 0   * 1: kappa fixed, 0: kappa to be estimated
       kappa = 2   * initial or fixed kappa
   fix_omega = 0   * 1: omega or omega_1 fixed, 0: estimate
       omega = 1   * initial or fixed omega, for codons or codon-based AAs
       getSE = 0       * 0: don't want them, 1: want S.E.s of estimates
RateAncestor = 0       * (0,1,2): rates (alpha>0) or ancestral states (1 or 2)
  Small_Diff = .45e-6  * Default value.
   cleandata = 1       * remove sites with ambiguity data (1:yes, 0:no)?
 fix_blength = 0       * 0: ignore, -1: random, 1: initial, 2: fixed
```

Run command file (null model):

The command file for the null model is the same as for the alternative model, except for two parameters (in red):
1) The name of the output file (outfile) is different.
2) The dN/dS ratio is fixed to 1 (fix_omega = 1).

```
     seqfile = TF105351.Eut.3.phy              * sequence data file name
    treefile = TF105351.Eut.3.53876.tree       * tree structure file name
     outfile = TF105351.Eut.3.53876.fixed.mlc  * main result file name

       noisy = 9   * 0,1,2,3,9: how much rubbish on the screen
     verbose = 1   * 1: detailed output, 0: concise output
     runmode = 0   * 0: user tree;  1: semi-automatic;  2: automatic
                 * 3: StepwiseAddition; (4,5):PerturbationNNI; -2: pairwise

     seqtype = 1   * 1:codons; 2:AAs; 3:codons-->AAs
   CodonFreq = 2   * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table
       clock = 0   * 0: no clock, unrooted tree, 1: clock, rooted tree
      aaDist = 0   * 0:equal, +:geometric; -:linear, {1-5:G1974,Miyata,c,p,v}
       model = 2   * models for codons:
                   * 0:one, 1:b, 2:2 or more dN/dS ratios for branches
     NSsites = 2   * 0:one w; 1:NearlyNeutral; 2:PositiveSelection; 3:discrete;
                   * 4:freqs; 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;10:3normal
       icode = 0   * 0:standard genetic code; 1:mammalian mt; 2-10:see below
       Mgene = 0   * 0:rates, 1:separate; 2:pi, 3:kappa, 4:all
   fix_kappa = 0   * 1: kappa fixed, 0: kappa to be estimated
       kappa = 2   * initial or fixed kappa
   fix_omega = 1   * 1: omega or omega_1 fixed, 0: estimate
       omega = 1   * initial or fixed omega, for codons or codon-based AAs
```

```
      getSE = 0       * 0: don't want them, 1: want S.E.s of estimates
RateAncestor = 0      * (0,1,2): rates (alpha>0) or ancestral states (1 or 2)
  Small_Diff = .45e-6 * Default value.
   cleandata = 1      * remove sites with ambiguity data (1:yes, 0:no)?
 fix_blength = 0      * 0: ignore, -1: random, 1: initial, 2: fixed
```

Launch CodeML:
In Unix (Linux, MacOSX), this will look like:

```
codeml ./TF105351.Eut.3.53876.ctl
codeml ./TF105351.Eut.3.53876.fixed.ctl
```

Analyse results:

1) Assign significance of the detection of positive selection on the selected branch:

Two output files are produced:
TF105351.Eut.3.53876.mlc (alternative model) and TF105351.Eut.3.53876.fixed.mlc (null model).

We retieve the likelihood values lnL1 and lnL0 from TF105351.Eut.3.53876.mlc and TF105351.Eut.3.53876.fixed.mlc files, respectively:

> lnL(ntime: 41 np: 46): -4707.210163 +0.000000 (lnL1)
> lnL(ntime: 41 np: 45): -4710.222252 +0.000000 (lnL0)

We can construct the LRT:
-->
$\Delta LRT = 2\times(lnL1 - lnL0) = 2\times(-4707.210163 - (-4710.222252)) = 6.024178$

The degree of freedom is 1 (np1 - np0 = 46 - 45).
p-value = 0.014104 (under χ^2) => significant.

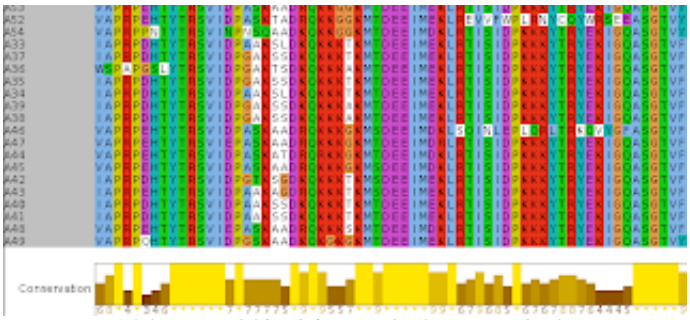
A significant result with the branch-site codon model means that positive selection affected a subset of sites during a specific evolutionary time (also called **episodic model of protein evolution**).

2) If significant, we can retrieve sites under positive selection:

In the TF105351.Eut.3.53876.mlc, we can retrieve sites under positive selection using the **Bayes Empirical Bayes (BEB) method**:

```
Positive sites for foreground lineages Prob(w>1):
    36 K 0.971*
   159 C 0.993**
```

Amino acids K and C refer to the first sequence in the alignment.
Position 36 has a high probability (97.1%) of being under positive selection. Position 159 has a very high probability (99.3%) of being under positive selection.



Position 36 shifted from a lysine to a glycine.

In future posts, I will speak about various potential problems (or not) and limits of the inference of positive selection.

RAS
