Introduction to calculating dN/dS ratios with codeml Version 2

Frank Aylward

Abstract

This is short tutorial on one way to calculate dN/dS ratios between pairs of protein-coding nucleic acid sequences using codeml in the PAML package.

Code is intended for use on an Ubuntu 16.04 LTS OS, but it may work on other Unix or Unix-like systems.

The programs used in this tutorial are:

codeml in the PAML package. On a Ubuntu 16.04 LTS system it should be able to install this tool with "sudo apt install paml". http://abacus.gene.ucl.ac.uk/software/paml.html

PAL2NAL. This is essentially a PERL script that you will want to have handy, either by putting it in the folder that you are working in or by putting it somewhere that is in your PATH. http://www.bork.embl.de/pal2nal/

clustal-omaga. You should be able to install this with "sudo apt install clustalo" This is a nice amino acid and nucleic acid alignement program. For purposes here your choice of aligner is not critical, so if you prefer MAFFT or Muscle or something else you can continue using those. http://www.clustal.org/omega/

Citation: Frank Aylward Introduction to calculating dN/dS ratios with codeml. **protocols.io** dx.doi.org/10.17504/protocols.io.qhwdt7e
Published: 28 May 2018

Protocol

Get the files organized

Step 1.

When calculating dnds ratios, file organization and consistent formatting are key. This is because it is necessary to create amino acid alignments of proteins first, and then convert them to nucleic acid. For this to happen we need to have both amino acid and nucleic acid sequences in separate files, and the proteins and genes in those file need to have the exact same unique identifiers in their FASTA

headers.

First we can download some pre-compiled data from GitHub:

git clone https://github.com/faylward/dnds

If you navigate into the dnds/ folder you should see four files: One amino acid FASTA file (.fna), one nucleic acid FASTA file (.faa), one codeml control file (.ctl), and one python script used for parsing the final codeml output (.py).

Create an amino acid alignment

Step 2.

First we want to align the amino acid sequences using clustal omega. The command here is simple enough if we use default parameters:

clustalo -i cluster_1.faa -o cluster_1.aln.faa

The .aln.faa file should have the amino acid alignment we need.

ANNOTATIONS

Carolina Mg 02 Jun 2018

Hi Dr. Frank! :)

I already solved the “installation problem”. The fact is that I was not typing perl before pal2nal, and so I was not calling Perl!

I should had write: perl pal2nal.pl cluster_1.aln.faa cluster_1.fna -output paml -nogap > cluster_1.pal2nal
Instead: pal2nal.pl cluster_1.aln.faa cluster_1.fna -output paml -nogap > cluster_1.pal2nal

Great things are learnt when you get an error :)

Convert aa alignment to na alignment

Step 3.

Now we can use pal2nal to get a codon-based nucleic acid alignment. This is critical since we need to be sure the nucleic acid alignment is alined codon-by-codon we we know when substitutions result in a synonymous or nonsynonymous amino change. If we had simply performed an alignment on the nucleic acid sequence, we could not be sure that every single codon was lined up for this kind of calculation.

pal2nal.pl cluster_1.aln.faa cluster_1.fna -output paml -nogap > cluster_1.pal2nal

Here we input the aligned amino acid sequences and the raw nucleic acid sequences. The flags '-output paml' indicates that we want the output format to be in paml format (for simplicity in subsequent steps). The '-nogap' flag indicates that we want to remove gaps and inframe stop codons, since those are not used in subsequent steps. Just type 'pal2nal.pl' for a full description of all of the options.

Run codeml

Step 4.

To run codeml all we need to do is type 'codeml' in the same folder that the codeml.ctl file is in. All of the options are in the codeml.ctl file. There are lots of different options here that are described in the PAML manual. The key ones here are:

seqfile = cluster_2.pal2nal [this tells the program where to find the codon-aligned nucleic acid sequences]

outfile = codeml.txt [this tells the program where we want the output]

runmode = -2 [this indicates we want to perform pairwise comparisons]

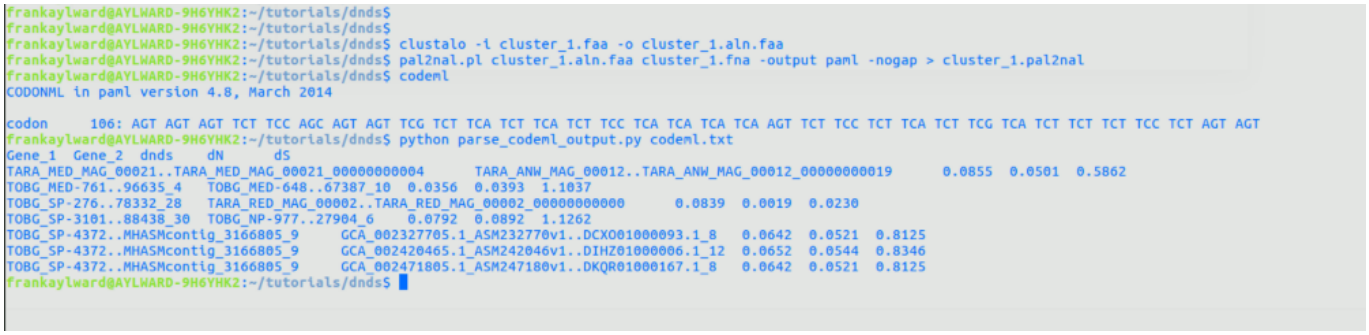
Some codeml applications require a phylogenetic tree to be provided in .nwk format, but since we are doing pairwise comparisons here it is not required.

When we run codeml, the calculation may take around a minute to finish, and a large number of files will be created. The file codeml.txt is what we want though. We can parse results with the parse_codeml_output.py python script:

python parse_codeml_output.py codeml.txt

Note that not all gene-pairs will be printed out. This is because the script filters out all pairs for which dS was < 0.01 or > 2. Values < 0.01 indicate that we may not get a reliable estimate of dN/dS, since the sequences are so similar. dS values > 2 indicate that the sequences are quite divergent and multiple substitutions have likely occured at most sites, so dN/dS estimates will again be compromised.

The overall workflow should provide results that look something like this:



The real power comes from estimating dN/dS from multiple genes across several genomes, and looking for overall trends. But that will be a tutorial for another day :)

ANNOTATIONS

Carolina Mg 02 Jun 2018

Done!

It seems that we are dealing with purifying selection!