PAML软件的一些简单的具体的使用操作

1. 首先用Clustal X进行序列比对：要保证：保证核苷酸序列是三的倍数，没有终止密码子，核苷酸序列的第一位是密码子的第一位。假设序列名为cox1.fas

2. 使用DAMBE软件进行转换成PML格式。软件使用截图： 打开要换换的文件，然后“file” “save and convert sequence format”，在保存类型中选择“Yang’s PAML”。那么此时的序列名为“cox1.PML”

3 这样就可以得到文件“\*.PML”，然后就直接把后缀改成“\*.nuc”。那么此时的序列名为“cox1.nuc” 这样就完成了文件格式的转换。

4 打开PAML软件的文件夹，找到文件名是“bin”的文件夹，打开之后，找到程序“codeml.exe”,把该程序复制到D盘的根目录下。（这一步并不是必要的，只是要把用到的几个程序放在同一个目录下）

5 在你使用ClustalX进行序列比对的时候，会生成一棵进化树，适用treeview软件可以打开，你需要的是把文件的后缀名改称“\*.trees”。即树的文件名是“cox1.trees”，这就完成了树的格式的转换。

6 然后再PAML4的文件夹中找到一个后缀是“\*.ctl”的文件，把文件名改成“cox1.ctl”，复制到和“codeml.exe”相同的地方。

7 要对codeml.ctl文件中的各个选项的值进行修改，具体内容如下：

8 seqfile = cox1.nuc 按你自己的文件名进行修改，就可以了，

9 treefile = cox1.trees

outfile = mlc \* main result file name ，

noisy = 9 \* 0,1,2,3,9: how much rubbish on the screen ，

verbose = 0 \* 0: concise; 1: detailed, 2: too much

runmode = 0

seqtype = 1 \* 1:codons; 2:AAs; 3:codons-->AAs

CodonFreq = 2 \* 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table

 clock = 0

aaDist = 0 \* 0:equal, +:geometric; -:linear, 1-6:G1974,Miyata,c,p,v,a

aaRatefile = wag.dat \* only used for aa seqs with model=empirical(\_F)  \* dayhoff.dat, jones.dat, wag.dat, mtmam.dat, or your own

model = 0，这是使用的最简单的模型， \* models for codons:  \* 0:one, 1:b, 2:2 or more dN/dS ratios for branches

\* models for AAs or codon-translated AAs:

 \* 0:poisson, 1:proportional, 2:Empirical, 3:Empirical+F 28 \* 6:FromCodon, 7:AAClasses, 8:REVaa\_0, 9:REVaa(nr=189)

NSsites = 0 3 1 2 7 8 ，依次选取了6个模型。也可以选其中的两个，但必须是0和3，1和2，7和8。相互配对

 icode = 4 \* 0:universal code; 1:mammalian mt; 2-10:see below如果是核基因的话就选0。

fix\_kappa = 0

kappa = 5

fix\_omega = 0

 omega = 0.2

getSE = 0

RateAncestor = 0

Small\_Diff = .5e-6

cleandata = 1 \* remove sites with ambiguity data (1:yes, 0:no)?

method = 0 \* 0: simultaneous; 1: one branch at a time

10 最后保证在同一个文件夹内同时具有：三个文件“codeml.exe”,“cox1.nuc”“cox1.trees”，这时候你双击codeml.exe，就可以运行程序。

如果不能正确运行的话，你可以通过运行cmd，在dos情况下，运行codeml.exe，这样会有错误提示，知道你错在哪里了。

常见命令解释： 1. Baseml.ctl的命令说明：

2. noisy用来控制输出结果的多少，如果模型适用的运算比较多的话，noisy的值可以选择的比较大，verbose可以控制结果文件中结果的多少。

3. runmode = 0 表明在树的结构文件中估算树的拓扑结构。这个选项是我们通常情况下选择的，基本上可以满足我们的需要。

4. Runmode = 1 or 2 表明通过星状-分解算法来进行启发式搜索树。Runmode = 2 这种算法是从星状树开始搜索，而runmode = 1 则表明软件读取多歧树是从树的结构文件中，并且同过比较去估计最佳二歧树。 5. runmode = 3 表明是逐步增加的。

6. runmode = 4 通过简约法来搜索具有NNI perturbation的起始树。

7. runmode = 5 表明从树的结构文件中来读取NNI perturbation with 起始树。

8. Model 0，1，…，8 分别代表以下模型：JC69, K80, F81, F84, HKY85, T92, TN93, REV( also know as GTR), and UNREST。

9. Mgene 用于和序列数据文件中的option G进行联合，用于多个基因和多个位点的联合分析。如果不使用option G的话，则选择0。

10. ndata 用于指定文件中的分隔的数据集的数目。它的变化被用于模拟，你可以使用evolver来产生200个复制数据集，这是设置ndata = 200，然后用baseml进行分析。

11. clock 用于指定谱系之间速率恒定或变化的模型。Clock = 0，意味着整棵树中，不同分支之间不存在clock现象；Clock = 1，意味着global clock，所有的分支具有相同的进化速率；clock = 2 意味着local clock，所有分支之间的进化速率被分成几个部分；clock = 3意味着多个基因或多重分隔数据，允许分支的进化速率以不同的方式变化。；；

12. Codeml.ctl的使用说明：

13. CodonFreq 用于平衡密码子替换模型中的密码子使用频率。Codonfreq = 0 说明每种密码子的使用频率是相同的；codonfreq = 1 说明是从平均核苷酸频率中计算出来的；codonfreq = 2说明是从三个密码子位置的平均核苷酸频率得来的；codonfreq = 3则使用了三个参数。Codonfreq = 0，1，2和3 所代表的模型中使用的参数的数目分别为：0，3，9，和60。

14. aadist用于指定氨基酸距离是否是相同的（= 0），还是属于Grantham’s matrix（= 1）。

15. runmode = -2 执行ML方法来推测蛋白序列两两之间的dn和ds。

16. model 用于估计各个分支之间的w值。 Model = 0，表明所有的谱系具有一个w比率（one w ratio）；model = 1，每一支具有一个速率（free-ratio）；model = 2 表明速率的任意数字。

17. NSsites 主要是用于指定模型允许dn/ds（w）在不同的位点之间变化。NSsites = m 表明对应于 model = m。变化的 ncatG被用来指定在一些特定的模型下的w分布的类型的数目。NcatG的值被用于执行一下分析：paper are 3 for M3 (discrete), 5 for M4 (freq), 10 for the continuous distributions (M5 gamma, M6: 2gamma, M7: beta, M8:beta&w, M9:beta&gamma, M10: beta&gamma+1, M11:beta&normal>1, and M12:0&2normal>1, M13:3normal>0). This means M8 will have 11 site classes (10 from the beta distribution plus 1 additional class) 。通过NSsites可以同时执行多个模型，例如：NSsites = 0 1 2 3 7 8，的意思就是同时执行M0，M1，M2a，M3，M7，和M8。作者建议：使用M1a和M2a来重建LRT，使用M7和M8来重建LRT，使用M2a和M8来鉴别受到正选择的位点。

18. icode 用来更改所选序列的遗传密码子，以期得到更加准确的结果。

19. RateAncestor = 1 表明你想重建原始序列，如果 RateAncestor = 0 说明你将避免这个计算。不过使用效果并不明显，还需要进一步研究如何使用。 PAML使用中最重要的就是模型的选择： PAML中所有的模型都在baseml和codeml这两个程序中使用。这两个程序是最大似然程序，它们使用数值优化算法来最大化对数似然值。这些模型最大的用途就是适用likelihood ratio test（似然比率检验）来检测有趣的生物学假设。这些模型是在Baseml中使用的，软件中常用的数学模型有：JC69（Jukes and Cantor 1969），K80（Kimura 1980），F81（Felsenstein 1981），F84（Felsenstein 1984），HKY85（Hasegawa 1984，1985），Tamura（1992），Tamura and Nei（1993），and REV，also know as GTR for general-time-reversible（Yang 1994）。 模型的一般遵循以下假设： 1. 在不同的谱系中替换是独立发生的。 2. 在不同的位点中替换也是独立发生的。 3. 替换的过程我们通过时间均匀马尔科夫过程（time-homogeneous Markov process）。

常用的两种检测方法：

1. Maximum likelihood estimates（MLEs）：观测到的数据X的概率（probability），当做为一个未知参数θ的函数的时候，就叫做似然函数（likelihood function）：L (θ：∣X) = f（θ∣X）。根据似然规则（likelihood principle），似然函数包括数据中关于参数θ所有的信息。参数θ的最佳点估计（optimal point estimate）可以通过最大化似然L的θ值或l（θ；X）的似然对数进行估计。并且，似然曲线可以为未确定的点估计提供信息。

 2. Likelihood ratio tests （LRTs）：假设一个简单模型或无效模型（simpler or null model）有一个参数 p0，更通用的模型或可选择的模型（general or alternative model）有一个参数p1，两个模型的最佳似然值分别为l0和l1。那么对数似然值差异（log likelihood difference）的两倍是：2△l = 2（l1 - l0），如果无效模型（null model）成立的话，那么对数似然值差异的二倍将与自由度是d.f. = p1- p0的卡方分布具有渐进关系（asymptotically）。因此，对数似然值差异的二倍的检验统计可以通过比较卡方分布来检验无效模型（null model）是否拒绝备择模型（alternative model）。 所谓Likelihood ratio test（似然比率检测）是用来检验两个模型的。

离散伽玛模型（discrete-gamma model）允许不同位点具有不同的变化速率。 Baseml中有核苷酸替换模型，Codeml中有不同位点替换速率变化的模型。

1. 作者在Codeml中进行比较的两个模型比较有：M1a（Nearly Neutral）和M2a（Positive Selection）；M7（beta）和M8（beta&ω）。

2. 作者认为M3对于正选择的LRT检测并不是十分适合，并不推荐适用M3模型。 使用似然比率检测可以验证正选择（Testing positive selection using the likelihood ratio test）。作者推荐使用二到三种LRT来验证正选择。第一个检测是比较M1a和M2a，

第二个检测是比较M7和M8。 Gamma分布中形状参数所表示的含义： 1. α＞ 1，大多数位点的替换速率在1附近，但有少数位点具有比较高或比较低的替换速率。曲线形状为 bell-shaped 2. α→∞，表明所有的位点具有一个相同速率。 3. α ≤ 1，表明大部分位点的替换速率比较低，或接近于不变，可是有一些位点具有比较高的替换速率。曲线形状为 L-shape。 PAML的一个重要功能就是检测基因是否受到正选择，即适应性选择。但是现在用于估计适应性选择的方法，忽略了氨基酸的化学性质，这样得出的结果是不准确的，作者表示，直接通过dn和ds的比较来确定受到什么样的选择压力，是不准确的。 PAML中的无效模型（null model）是指不允许任何位点的ω值大于1。PAML中null model是不允许w值大于1，如果null model成立，则w小于1，基因受到负选择；如果null model不成立，则w大于1，基因受到正选择。研究表明，通过比较两个点模型，而得到的结果尤其可靠。 Ancestral reconstruction 为探索数据提供了一个直观的方法，他被用于大量的数据分析，例如，评估不同谱系中的选择压力。但是由于这种方法的简单和直观，会产生很多错误。大多数重建原始序列的工作都忽略了这样一个事实，即使用假数据（pseudo-data）代替真实观察到的数据（real observed data），并且仅仅使用处于最佳状态（optimal character states）的一些特征，而忽略未处于最佳状态（suboptimal states）的一些特征，从而产生一些系统上的偏差（systematic biases）。如果数据中的ds区域饱和的话，那么会导致我们低估ds，从而使dn/ds的值偏高，即 ω的值偏高。进化距离估计中有关序列间隔的处理进化距离估计中，排列时的间隔导致了某些复杂的问题。同时，由于实验上的原因，也可能出现丧失信息的位点。在距离估计中，一般忽略这些位点，可用两种不同的方法来进行处理。一种方法是从数据分析中删除这些位点，称为完全删除（complete deletion）。一般来说，这种方法较好，因为DNA或氨基酸序列的不同区段往往具有不同的演变规律。然而在所研究的序列中，间隔不大或者是随机分布的，则可计算每个配对序列间的距离，并只忽略两个配对序列间的那些间隔。这种方法称为成对删除（pairwise deletion）。这个过程在PAML中可以通过cleandata = 0 或1。使用PAML进行数据分析的时候，所选用的序列越多，则得到的结果越可靠。一般应该大于17条序列，此外影响LRT检验结果好坏的因素还有：序列长度（sequence length）、序列分异度（sequence divergence）和正选择的强度(the strength of positive selection )(Anisimova, Bielawski et al. 2001; Opazo, Palma et al. 2005)。 还有使用PAML进行序列分析的时候必须检验序列是否发生过重排，可以使用的软件由PLATO 2.0，HYPHY等软件(Pond, Frost et al. 2005; Kosakovsky Pond, Posada et al. 2006; Petersen, Bollback et al. 2007)。如果是检测序列受到的选择压力的话，那么序列大于50codons即可，但是这些序列对于系统进化分析可能得到的结果不可靠(Pie 2006)。 物种的基因只有在受到正选择作用才能不断的适应环境的变化，所以正选择在物种进化中起到了非常重要的作用(Vallender and Lahn 2004)。 PAML中常用的模型有以下几个： 1. M0，所有的谱系具有相同的ω0值。 2. M3，discrete，它的位点具有三个离散类（discrete classes），并且具有不同的ω0。 3. M1a，nearly neutral model，允许两个位点分类，0＜ ω0 ＜1 或ω＞1。 4. M2a，selection，具有一个额外的位点分类ω＞1。 5. M7，beta，ω＜1，具有额外10个位点分类。 6. M8，ω＞1，具有额外11个位点分类。其中可以用于检验正选择的模型是M2a，M3和M8，得到了结果之后，我们可以再使用NEB（naïve empirical）和BEB（bayes empirical ）进行验证，通过后验概率，一般应该大于0.95。 PAML软件的模型中需要考虑的问题是： 1. 遗传密码子的结构。 2. 转换和颠换的比值。 3. 密码子不同位置的速率。 PAML操作中存在的问题： 1. 在codeml中，runmode是不是通常选择0，如果选择-2的话，就是使用ML法进行计算蛋白序列两两之间的dn和ds？那核苷酸序列可以吗？ 2. model，通常下，我选的为0，意思是所有的谱系具有一个相同的ω值，这个在分析的时候怎么选择？如何检测各个谱系之间的ω的差异？ 3. 对于NSsites的话，如果要进行两个模型之间比较，同时选择两个模型是否可以？ 4. 例如，我要比较M0和M3，那么在NSsites = 0 3。这样选择，在结果中计算的是这两个不同模型下的结果，还是进行的两个模型的比较？如果进行比较，是否需要分别计算出各个模型的lnL值，在使用chi程序进行两两比较？ 5. 其余还有什么关键性的参数？ 6. 还有运算量大不大？一般的512M内存，3.0G的计算机能运行完吗？ 7. 这个提示是什么意思？model should be 0 in the batch run?. 如何重建基因家族中的原始序列（Ancestral sequence）？ PAML软件中的序列处理： 1. 如果序列之间的分异度比较明显的话，那么需要4-5条序列；如果有10条序列的话，效果会比较好；如果序列数大于20，那么得到的结果就会比较可靠。当然这也和序列之间的分异度有比较密切的联系。 2. 最佳序列分异取决于序列的数目，如果树比较大的话，则可以容忍更多的变异。一般情况下，如果总共的ds的距离大于0.5，那么就认为这种方法是合理的。 3. 一般情况下，软件可以鉴别出一到两个位点受到较强的选择压力，但是有可能一些位点受到了选择压力，但比较弱，这是LRT方法会告诉你，这个位点存在选择压力，但是对于鉴别出，比较麻烦。 4. 一般情况下，使用简单模型和复杂模型得出的结果应该是一致的。所以，M0作为比较简单的模型，使用它得出的枝长、K以及w值应该和其他复杂模型得出的结果是一致的。 5. 如果比对的序列是高度相似或高度分异的话，我们应该执行程序两遍。 6. 如果使用NSsites执行多个模型，ncatG的参数也需要重新设置。 7. 在计算过程中，序列的饱和性并不是一个主要的问题，一般较高的序列分异会带来更多的问题，不同的序列之中会具有不同的密码子使用偏好性和核苷酸组成。 8. 计算出来的w值是不能为负值的。 9. 现在已经找到，lnL的值在rst文件中，但是每个位点的w值，在什么地方呢？即使找到了相关的参数，那么接下来再如何分析呢？ 10. 在进行序列分析的时候，要把编码序列末端的终止密码子去掉，以防止出现误差。 PAML中用到的进化树的格式： A tree structure file is used when runmode = 0 or 1. The file name is specified in the appropriate control file. The tree topology is typically specified using the parenthesis notation (括号标序法), although it is possible to use a branch representation, as described below. Parenthesis notation. The first is the familiar parenthesis representation, used in most phylogenetic software. The species can be represented using either their names or their indexes corresponding to the order of their occurrences in the sequence data file. If species names are used, they have to match exactly those in the sequence data file (including spaces or strange characters). Branch lengths are allowed. The following is a possible tree structure file for a data set of four species (human, chimpanzee, gorilla, and orangutan, occurring in this order in data file). The first tree is a star tree, while the next four trees are the same. If the tree has branch lengths, baseml and codeml allow you to use the branch lengths in the tree as starting values for maximum likelihood iteration. Whether you should use rooted or unrooted trees depends on the model, for example, on whether a molecular clock is assumed. Without the clock (clock = 0), unrooted trees should be used, such as ((1,2),3,4) or (1,2,(3,4)). With the clock or local-clock models, the trees should be rooted and these two trees are different and both are different from (((1,2),3),4). In PAML, a rooted tree has a bifurcation at the root, while an unrooted tree has a trifurcation or multifurcation at the root. Tree files produced by PAUP and MacClade. PAML programs have only limited compatibility with the tree file generated by PAUP or MacClade. First the “[& U]” notation for specifying an unrooted tree is ignored. For the tree to be accepted as an unrooted tree by PAML, you have to manually modify the tree file so that there is a trifurcation at the root, for example, by changing “(((1,2),3),4)” into ”((1,2),3,4)”. Second, the “Translate” keyword is ignored by PAML as well, and it is assumed that the ordering of the sequences in the tree file is exactly the same as the ordering of the sequences in the sequence data file. Branch or node labels. Some models implemented in baseml and codeml allow several groups of branched on the tree, which are assigned different parameters of interest. For example, in the local clock models (clock = 2 or 3) in baseml or codeml, you can have, say, 3 branch rate groups, with low, medium, and high rates respectively. Also the branch-specific codon models (model = 2 or 3 for codonml) allow different branch groups to have different ωs, leading to so called “two-ratios” and “three-ratios” models. All those models require branches or nodes in the tree to be labeled. Branch labels are specified in the same way as branch lengths except that the symbol “#” is used rather than “:”. The branch labels are consecutive integers starting from 0, which is the default and does not have to be specified. For example, the following tree is from the tree file examples/lysozyme/lysozyme.trees, with a branch label for fitting models of different ω ratio for branched. Anisimova, M., J. P. Bielawski, et al. (2001). "Accuracy and Power of the Likelihood Ratio Test in Detecting Adaptive Molecular Evolution." Mol Biol Evol 18(8): 1585-1592. Kosakovsky Pond, S. L., D. Posada, et al. (2006). "GARD: a genetic algorithm for recombination detection." Bioinformatics 22(24): 3096-3098. Opazo, J. C., R. E. Palma, et al. (2005). "Adaptive Evolution of the Insulin Gene in Caviomorph Rodents." Mol Biol Evol 22(5): 1290-1298. Petersen, L., J. P. Bollback, et al. (2007). "Genes under positive selection in Escherichia coli." Genome Res. 17(9): 1336-1343. Pie, M. R. (2006). "The Influence of Phylogenetic Uncertainty on the Detection of Positive Darwinian Selection." Mol Biol Evol 23(12): 2274-2278. Pond, S. L. K., S. D. W. Frost, et al. (2005). "HyPhy: hypothesis testing using phylogenies." Bioinformatics 21(5): 676-679. Vallender, E. J. and B. T. Lahn (2004). "Positive selection on the human genome." Hum. Mol. Genet. 13(suppl\_2): R245-254.