# BEAST估算物种分歧时间的总结 #

> - 提取orthomcl获得的结果
> - 序列比对
> - BEAST计算分歧时间
> - FigureTree or DensiTree进行结果的可视化

BEAST2下载地址

Beast2
Bayesian evolutionary analysis by sampling trees

> BEAST 2 is a cross-platform program for Bayesian phylogenetic analysis of molecular sequences. It estimates rooted, time-measured phylogenies using strict or relaxed molecular clock models. It can be used as a method of reconstructing phylogenies but is also a framework for testing evolutionary hypotheses without conditioning on a single tree topology. BEAST 2 uses Markov chain Monte Carlo (MCMC) to average over tree space, so that each tree is weighted proportional to its posterior probability. BEAST 2 includes a graphical user-interface for setting up standard analyses and a suit of programs for analysing the results.

提取orthomcl获得的结果

获得的每个fasta文件中，包含每个物种的单拷贝基因各一个

多序列比对（MSA）

###1. Guidance调用MAFFT对密码子进行（coden）进行比对，设置--seqCutoff 0.9 --colCutoff 0.93

```
perl guidance.pl --proc_num 10 --seqFile your-fasta  --msaProgram MAFFT --seqType codon --seqCutoff 0.9 --colCutoff 0.93 --outDir your-output-dir
```

###2. Gblock过滤gap

```
/data/Users/zhibin1/program/Gblocks_0.91b/Gblocks your-Guidance-result -t=C
```

###3. fasta2nexus（BEAST2的BEAUtil插件需nexus文件作为输入）

guidance->nexus的python代码：

```
#!/python
# -*- coding:utf-8 -*-
import os
import shutil
import re
import sys
import time
from Bio import SeqIO
from Bio.Alphabet import IUPAC

def guidance(fasta):
        reg = re.search(r'^(\w+)',fasta)
        outdir =str(os.getcwd()) + '/' + reg.groups()[0]
        aln = reg.groups()[0] + '_aln.fasta'
        gb = reg.groups()[0] + '_aln.fasta-gb'
        htm = reg.groups()[0] + '_aln.fasta-gb.htm'
        final = reg.groups()[0] + '_final.fasta'
        nexus = reg.groups()[0] + '_final.nex'
        ######perform guidance.pl######
        os.system('perl /data/program/guidance.v2.02/www/Guidance/guidance.pl --proc_num 10 \
        --seqFile %s  --msaProgram MAFFT --seqType codon --seqCutoff 0.9 --colCutoff 0.93 --outDir %s' \
        % (fasta,outdir))
        os.chdir(outdir)
        if not os.path.exists("MSA.MAFFT.Without_low_SP_Col.With_Names"):
                return 0
        if not os.path.getsize('Seqs.Orig_DNA.fas.FIXED.Removed_Seq'):
                print 'No sequence removed!!!'
                shutil.move("MSA.MAFFT.Without_low_SP_Col.With_Names","../%s" % aln)
        os.chdir("..") #返回上层目录
        shutil.rmtree(outdir)
        ##########Gblock###################
        if os.path.exists(aln):
                os.system('/data/Users/zhibin1/program/Gblocks_0.91b/Gblocks %s -t=C' % aln)
                open(final,"w").write(open(gb,"r").read().replace(" ","")) ##将fasta-gb中的空格去掉
                ######fasta2nexus######
                SeqIO.write(SeqIO.parse(open(final),"fasta",IUPAC.unambiguous_dna),open(nexus,"w"),"nexus")
                os.remove(gb)
                os.remove(htm)
        os.remove(fasta)
```

```
#######extract fasta for MSA #######
dic = {record.id:record.seq for record in SeqIO.parse('goodCDS.fasta','fasta')}

with open("single-copy.txt","r") as fh:
        terms = 0;
        for line in fh:
                info = line[:-1].split("\t")[:-1]
                i = 0
                for name in info:
                        i += 1
                        if(i == 1):
                                tit = name.replace(":",".fasta")
                                out = open(tit,"w")
                        elif re.search(r'At\|',name):
                                continue
                        else:
                                out.write('>' + name.split("|")[0] + '\n' + str(dic[name]) + '\n')
                out.close()
                guidance(tit) ##call guidance function
                terms += 1
                if terms >=100:
                        break
```

这里只选择100个nexus文件进行后续的BEAST2的分析 ##BEAST2

- ☑ BEAUtil：BEAST2参数配置，生成xml文件
- ☑ BEAST2：主程序，进行MCMC的迭代计算
- ☑ Tracer：分析BEAST2产生的结果
- ☑ LogCombiner：合并多个树文件（本例为100个树文件）为一个
- ☑ TreeAnnotator：所有树文件的概括
- ☑ FigTree + DensiTree：树的可视化

1. BEAUtil 配置xml文件（windows，鼠标）

（1）import alignment （partition model） file -> import alignment (导入生成的nexus alignment文件)

（2）Tip Dates model（跳过，但birth-death分析时会用到）

（3）Site Model （设置核苷酸替代模型） Gamma Category Count设置为4 选择HKY模型（认为转换和颠换的概率不同）

> In the HKY model, the rate of transitions A ↔ G and C ↔ T is allowed to be different from the rate of transversions A \↔ C, G ↔ T. Furthermore, the frequency of each base can be either "Estimated", "Empirical" or "All Equal". When we set the frequencies to "Estimated", the frequency of each base will be co-estimated as a parameter during the BEAST run

（4）clock model (分子钟模型) 选择 Relexed Clock Log Normal 模型 number of discrete rates ： -1 Clock.rate: 1

(5) Priors model （先验概率模型，参数设置非常繁琐） Tree.t -> Calibrated Yule Model (birth-only model) birthRateY -> Gamma(伽马分布，Alpha=0.001,Beta=1000) gammaShape.s -> Exponential（指数分布，mean=1） kappa.s -> Log Normal (对数正态分布，M=1，S=1.25，default) ucldMean.c -> Uniform(均匀分布，default) ucldStdev.c -> Gamma(伽马分布，default)

物种分歧节点的先验信息(根据science文献)： a. 水稻与（小麦族+短柄草+大麦）的分歧时间大约为75Mya b. 大麦和小麦族的分歧时间大约为45Mya c. 小麦族分化的根节点约为6.5Mya 因此，设置上述三个节点处的先验信息： 节点a:Osa-Tra -> Log Normal(M=4.36,S=0.08,对应68-91MYA) 节点b:Bdi-Tra -> Log Normal(M=3.8,S=0.08对应37.9-52.7MYA) 节点c:Bdi-Tra -> Log Normal(M=1.8,S=0.08对应5.17-6.9MYA) a,b,c的monophyletic均勾选

（6）MCMC Chian Length -> 1000000(马尔科夫链长度，越长使得ESS值越大，进而使有效群体数目增多，增加度量的准确性) trace.log 中 log Every-> 1000 (Chian Length\trace.log=1000即可) file name -> wheat.log.txt treelog.t 中 log Every-> 1000 （与trace.log 相等）

（7）saving as xml （作为demo文件）

2. BEAST

使用beagle能够优化BEAST的运行速度，添加beagle库至环境变量bashrc中：

```
LD_LIBRARY_PATH=/data/program/beast/beagle-lib-master/lib:$LD_LIBRARY_PATH
```

nexus -> xml -> BEAST的python脚本：

```
#!/python
#coding:utf-8
##re.compile中的正则可根据情况替换
import glob
import re
import os
from Bio import SeqIO
files = glob.glob("group_*_final.nex")
for nex in files:
        dic ={seq.id:seq.seq for seq in SeqIO.parse(nex,"nexus")}
        pref =nex.replace(".nex","")
        xml_file = pref + ".xml"
        out = open(xml_file,"w")
        ###############nexus2xml###########################
        with open("demo.xml") as fh:
                for line in fh:
                        if re.search(r'<sequence',line):
                                name=re.search("taxon=\"(\w+)\"",line).groups()[0]##获得括号中匹配的内容 作为name
                                regex1 = re.compile("value=\"(\w+)\"")##获得regex
                                out.write(regex1.sub("value=\"%s\"" % str(dic[name]),line)) ##用新序列去替换旧序列
                        elif re.search(r'demo',line):
                                regex2 = re.compile('demo')
                                out.write(regex2.sub(pref,line))
                        else:
                                out.write(line)
        out.close()
        ####################BEAST2###########################
        #os.system("source ~/.bashrc")
        os.system("/data/Users/zhibin1/program/jdk1.8.0_66/bin/java -jar /data/program/beast/lib/beast.jar -beagle_GPU -beagle_SSE %s" % xml_file)
        break
```

###3. Tracer (略过) ###4. LogCombiner.exe（生成多个树文件合并成的树文件） File Type: Tree files 导入100个tree文件，每个文件选择burin值为0.1（过滤前%10的不准确的树） Output file：-> tree.combined.txt

###5.TreeAnnotator（汇总LogCombiner.exe产生的树，形成一致树，95%的置信区间） burnin percentage -> 10% Posterior probability limit -> 0 target tree type -> Maximum clade crediblility tree Node heights -> Mean heights Input tree file -> tree.combined.txt Output tree file -> tree.summary.txt

###6. FigTree + DensiTree可视化 □