用 RAxML 构建极大似然进化树

RAxML 是用极大似然法建立进化树的软件之一，可以处理超大规模的序列数据，包括上千至上万个物种，几百至上万个已经比对好的碱基序列。作者是德国慕尼黑大学的 A. Stamatak 博士。

RAxML 有若干版本（有的版本支持在多个 CPU 上运行），本文以最常用的单机版 raxmlHPC 为例。

1 下载和安装

RAxML 可以在 Linux, MacOS, DOS 下运行，下载网址为
http://icwww.epfl.ch/~stamatak/index-Dateien/Page443.htm
也可以使用 www.phylo.com 的超级计算机运行。
对于 Linux 和 Mac 用户
下载 RAxML-7.0.4.tar.gz 用 gcc 编译即可
make –f Makefile.gcc
Windows 用户可以下载编译好的 exe 文件，而无需安装。

2 数据的输入

RAxML 的数据位 PHYLIP 格式，但是其名字可以增加至 256 个字符。“RAxML 对 PHYLIP 文件中的 tabs，inset 不敏感”。输入的树的格式为 Newick

RAxML 的查错功能

1 序列的名称有重复，即不同的碱基却拥有一致的名称。
2 序列的内容重复，即两条不同名称的序列，碱基完全一致。
3 某个位点完全由序列完全由未知符号组成，如氨基酸序列完全由 X,?,*,-组成，DNA 序列完全由 N,O,X,?,-组成。
4 序列完全由未知符号组成，如氨基酸序列完全由 X,?,*,-组成，DNA 序列完全由 N,O,X,?,-组成。
5 序列名称中禁用的字符 如包括空格、制表符、换行符、:,(),[]等

3 RAxMLHPC 下的选项

-s sequenceFileName 要处理的 phy 文件
-n outputFileName 输出的文件
-m substitutionModel 模型设定
方括号中的为可选项：
[-a weightFileName] 设定每个位点的权重，必须在同一文件夹中给出相应位点的权重

[-b bootstrapRandomNumberSeed]　设定 bootstrap 起始随机数
[-c numberOfCategories] 设定位点变化率的等级
[-d] -d 完全随机的搜索进化树，而不是从 maximum parsimony tree 开始。在 100 至 200 个分类单元间，该选项可能会生成拓扑结构完全不同的局部最大似然树。
[-e likelihoodEpsilon]默认值为 0.1
[-E excludeFileName] 排除的位点文件名
[-f a|b|c|d|e|g|h|i|j|m|n|o|p|s|t|w|x]　f 算法
-f a rapid Bootstrap
-f b draw the bipartitions using a bunch of topologies
-f c checks if RAxML can read the alignment.
-f d rapid hill-climbing algorithm
-f e optimize the model parameters
-f g compute the per–site log Likelihoods for one ore more trees passed via -z.
-f h compute a log likelihood test (SH-test [21]) between a best tree passed via -t and a bunch of other trees passed via -z.
-f i performs a really thorough standard bootstrap
……

[-g groupingFileName]　预先分组的名称
[-h] program options
[-i initialRearrangementSetting] speccify an innitial rearrangement setting for the ininital phase of the search algorithm.
[-j]
[-k] optimize branchlength and model parameters on bootstrapped trees
[-l sequenceSimilarityThreshold]　Specify a threshold for sequence similarity clustering.
[-L sequenceSimilarityThreshold]
[-M]　模型设定
-m GTRCAT: GTR approximation
-m GTRMIX: Search a good topology under GTRCAT
-m GTRGAMMA: General Time Reversible model of nucleotide subistution with the gamma model of rate heterogeneity.
-m GTRCAT_GAMMA: Inference of the tree with site-specific evolutionary rates. 4 discrete GAMMA rates,
-m GTRGAMMAI: Same as GTRGAMMA, but with estimate of proportion of invariable sites
-m GTRMIXI: Same as GTRMIX, but with estimate of proportion of invariable sites.
-m GTRCAT GAMMAI: Same as GTRCAT_GAMMA, but with estimate of proportion of invariable sites.

-n outputFileName 输出文件名
-o outgroupName(s) 设定外类群 如果有两个以上外类群，两者之间不能用空格，而应该用英文的","
DNA, gen1=1-500
DNA, gen2=501-1000

[-p parsimonyRandomSeed]
[-P proteinModel]
[-q multipleModelFileName]
-q multiple modelfile name

如将以下信息拷贝到另存为文件 genenames
DNA, rbcLa = 1-526
DNA, matK = 527-1472
调用方法 -q genenames
-m GTRGAMMA

[-r binaryConstraintTree]
-s sequenceFileName 待分析的 phy 文件
[-t userStartingTree] 用户指定的进化树拓扑结构
[-T numberOfThreads]
[-u multiBootstrapSearches] Specify the number of multiple BS searches per replicate to obtain better ML trees for each replicate.
[-v] 版本信息
[-w workingDirectory] 将文件写入的工作目录
[-x rapidBootstrapRandomNumberSeed] invoke rapidBootstrap
[-y] -y 只输出简约树拓扑结构，之后推出，该树也可以用于 GARLI 等软件
[-z multipleTreesFile]
[-#|-N numberOfRuns]

生成的文件
RAxML log.exampleRun: 运行时间、似然值/ number of checkpoint file
RAxML result.exampleRun: 树文件
RAxML info.exampleRun: -m GTRGAMMA or -m GTRMIX contains information about the model and algorithm used
RAxML parsimonyTree.exampleRun: -t.
RAxML randomTree.exampleRun: -d.
RAxML checkpoint.exampleRun.checkpointNumber: -j
RAxML bootstrap.exampleRun: -# and -b or -x
RAxML bipartitions.exampleRun: -f b
RAxML reducedList.exampleRun: -l or -L
RAxML bipartitionFrequencies.exampleRun: -t , -z , -f m
RAxML perSiteLLs.exampleRun: -f g
RAxML bestTree.exampleRun: -x 12345 -f a
RAxML distances.exampleRun: -f x

4 分析实例

若当前已经有比对好的序列，名为 test1.phy 文件

raxmlHPC -x 12345 -p 12345 -# 100 -m GTRGAMMA out1 -s test1.phy -f d -q gennames -n TEST
将以上语句粘贴到记事本中，另存为 test1.bat 文件，保存到 raxmlHPC.exe 相同的文件夹，双击 test1.bat 即可运行。 运行结束后，程序将自动关闭。

-x 用快速方法进行 Bootstrap
-p 设定随机数
-# Bootstrap100 次
-m GTRGAMMA 模型
-o out1 将 out1 序列设置为外类群
-s test1.phy 输入的 phy 文件为 ex_al.phy
-n TEST 输出的各结果文件中包含 TEST
-q gennames 设定的基因的各位点分割位置
-f d rapid hill-climbing algorithm

5 详实的例子

1 生成一系列随机化的 MP 树
raxmlHPC -y -s ex_al -m GTRCAT -n ST0
raxmlHPC -y -s ex_al -m GTRCAT -n ST1
raxmlHPC -y -s ex_al -m GTRCAT -n ST2
raxmlHPC -y -s ex_al -m GTRCAT -n ST3
raxmlHPC -y -s ex_al -m GTRCAT -n ST4
2 infer the ML trees for those starting trees using a fixed setting -i 10
raxmlHPC -f d -i 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST0 -n FI0
raxmlHPC -f d -i 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST1 -n FI1
raxmlHPC -f d -i 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST2 -n FI2
raxmlHPC -f d -i 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST3 -n FI3
raxmlHPC -f d -i 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST4 -n FI4

3 using the automatically determined setting on the same starting trees:
raxmlHPC -f d -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST0 -n AI0
raxmlHPC -f d -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST0 -n AI1
raxmlHPC -f d -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST0 -n AI2
raxmlHPC -f d -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST0 -n AI3
raxmlHPC -f d -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST0 -n AI4

6 正确的碱基变化等级

Getting the Number of Categories right
大样本量的时候建议采用 GTRMIX 搜寻极大似然树
用 GTRCAT 进行相应的 Bootstrap
因此，需要设定几种-c 值，查看那种给出最大的 gamma 的似然值。
raxmlHPC -f d -c 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST0 -n C10_0
raxmlHPC -f d -c 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST1 -n C10_1
raxmlHPC -f d -c 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST2 -n C10_2
raxmlHPC -f d -c 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST3 -n C10_3
raxmlHPC -f d -c 10 -m GTRMIX -s ex_al -t RAxML_parsimonyTree.ST4 -n C10_4

7 寻找已知最优的极大似然树

The Best-Known Likelihood tree (BKL)
RAxML 从逐步随机添加的最大简约树开始，搜寻极大似然树。在最大简约树建好之后，部分树将进行简约树重排，从而找到更为可靠的简约树。为什么不以 NJ 树开始，而是以 MP 树开始呢？ 这是因为在不同的搜索中，MP 树的拓扑结构可能是不同的，而不同的初始拓扑结构，会使用户有更大的可能发现极大似然树。

注意
specifying -m GTRCAT in combination with -# is not a good idea, because you will probably want to compare the trees inferred under GTRCAT based on their likelihood values and will have to compute the likelihood of the final trees under GTRGAMMA anyway.

8 Bootstrapping 的设置

举例：
raxmlHPC -f d -m GTRCAT -s ex_al -# 100 -b 12345 -n MultipleBootstrap
raxmlHPC-MPI -f d -m GTRCAT -s ex_al -# 100 -b 12345 -n MultipleBootstrap

按照 A. Stamatak 博士的 The RAxML 7.0.4 Manual
中国科学院植物研究所　张金龙编
Jinlongzhang01@gmail.com
zhangjl@ibcas.ac.cn
2010 年 2 月 2 日