[RAxML][https://cme.h-its.org/exelixis/web/software/raxml/hands_on.html]使用极大似然法构建进化树软件，可用于进化树的分析，和分析比对生成进化树文件。(post analyses of sets of phylogenetic trees, analyses of alignments and, evolutionary placement of short reads)

软件安装:brew install raxml

- The CAT model of rate heterogeneity

CAT has been designed to accelrate the computations on large datasets with many taxa! It is not a good idea to use the CAT arppoximation of rate heterogeneity on datasets with less than 50 taxa. In general there will not be enough data per alignment column available to reliably estimate the per-site rate parameters.

The GTRCAT approximation is a computational work-around for the widely used General Time Reversible mode of nucleotide substitution under the Gamma model of rate heterogeneity. CAT servers teh analogous purpose, that is, to accommodate searches that incorpate rate heterogeneity.

The main idea behind GTRCAT is to allow for integration for rate heterogeneity into phylogenetic analyses at a significantly lower computational cost(about 4 times faster) and memory consumption(4 times lower).

1. 输入文件格式

- 比对输入文件格式：PHYLIP或FASTA。

[PHYLIP][http://scikit-bio.org/docs/0.2.3/generated/skbio.io.phylip.html]文件格式存储多重序列比对信息

```
      5    42
Turkey    AAGCTNGGGC ATTTCAGGGT GAGCCCGGGC AATACAGGGT AT
Salmo gairAAGCCTTGGC AGTGCAGGGT GAGCCGTGGC CGGGCACGGT AT
H. SapiensACCGGTTGGC CGTTCAGGGT ACAGGTTGGC CGTTCAGGGT AA
Chimp     AAACCCTTGC CGTTACGCTT AAACCGAGGC CGGGACACTC AT
Gorilla   AAACCCTTGC CGGTACGCTT AAACCATTGC CGGTACGCTT AA
```

PHYLIP格式为plain text格式，包含两部分，header部分描述了比对的规模，紧接着就是多重序列比对信息本身。STrict PHYLIP要求序列识别符包含精确的10字符长度(padded with spaces as necessary)，其他生信软件(RAxML)可放松该要求，从而可使用更长的序列ID.

header必须为第一行，且包含单行描述比对的规模。包含合适的空格及一个或多个空格分开的两个正整数n和m。第一个n指定序列数目，也就是多少个用于比对的序列；第二个m指定序列长度，也就是比对中的列数目。最小规模为1x1。

序列比对紧跟head而行，其中使用'-'或'.'来表示gap字符。

- 输入树格式，[Newick][http://evolution.genetics.washington.edu/phylip/newicktree.html]

```
(B:6.0,(A:5.0,C:3.0,E:4.0):5.0,D:11.0);
```

- 比对错误查询

RAxML在分析前将自动查询以下错误：

1. 多重比对中相同序列名称出现多次
2. 相同的序列出现多次
3. 某些列仅包含未知的字符,AA: X, ?, *, - ; DNA: N, O, X, ?, -
4. 某序列仅包含未知的字符

- RAxML参数选项

-s 输入需要处理的文件名, PHYLIP或FASTA格式

-n 输出文件名

-o 输出文件名

-m 指定置换模型名称

-a 指定每一列的权重，该权重文件必须为整数且和比对文件的列相同

```
5 1 1 2 1 1 1 1 1 1 1 2 1 1 3 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 4 1 1 1 4 1 1
```

-b 指定整数(随机种子)，开启bootstrapping，默认为OFF；可用于重复运行结果

-# 指定重复运行次数，和-b参数结合，调用多重bootstrap分析，默认为1，单次分析

-c 在rate heterogeneity设置为CAT时，指定为明确的率分类的数值，默认25

-E 指定包含希望排除的比对位置(列)的文件名

```
raxmlHPC -E excludeFile -s alg -m GTRCAT -q part -n TEST
```

那么包含排除列的比对文件将会被命名为alg.excludeFile，而包含排除列的部分文件将会命名为part.excludeFile

例如希望排除100-199和200-299位置，则exludeFile文件为：

```
100-199
```

```
200-299
```

-f 选择算法，a为rapid bootstrap analysis，用于在一个程序运行中搜索best-scroing ML tree

```
raxmlHPC -f a -p 12345 -s alg -x 12345 -# 100 -m GTRCAT -n TEST
```

-f d为默认参数，new rapid hill-climbing，RAxML默认算法整体上要快与原始搜索算法。最终生成的trees几乎和full search algorithm一样。

```
raxmlHPC -f d -m GTRCAT -p 12345 -s alg -n TEST
```

-k 指定bootstrapped trees输出包含branch lengths，需更长运行时间，默认OFF

-K 指定multi-stat substitution models用于RAxML：ORDERED, MK, GTR，默认为GTR模型。

-p 指定随机种子用于[parsimony inference][https://www.researchgate.net/post/Whats_the_difference_between_neighbor_joining_maximum_likelihood_maximum_parsimony_and_Bayesian_inference]，该选项允许重建输出结果，当未指定，程序将会报错退出。

-x 设定随机整数(随机种子)，开启开素bootstrapping

-m 指定使用的置换模型：Binary(morphological)，Nucleotide，Multi-State，Amino Acid；模型字符串最后加上X表示，使用maximum likelihood estiamte评估base frequencies。

-m CAT，表示使用CAT model of rate heterogeneity

-m CATI，表示经过CAT搜索的到tree后，在最后使用GAMMA加上proportion of invariable sites estimate评估最终的trees，而不是仅使用默认的纯GAMMA

-m GTRCAT: GTR approximation

-m GTRMIX: serach a good topology under GTRCAT

-m GTRGAMMA: general time reversible model of nucleotide substitution with the gamma model of rate heterogeneity

-m GTECAT_GAMMA: inference of the tree with site-specific evolutionary rates, 4 discrete GAMMA rates

-q 指定包含比对部分的模型的文件

例如：含有纯DNA比对共1000bp，位置1-500为gene1，位置501-1000为gene2，则该partition文件为：

```
DNA, gene1=1-500
```

```
DNA, gene2=501-1000
```

或者第一部分使用ML estimate of frequencies

```
DNAX, gene1=1-500
```

```
DNA, gene2=501-1000
```

若含有DNA和AA比对，DNA位置为1-500，AA位置为501-1000

```
DNA, gene1=1-500
```

```
WGA, gene2=501-1000
```

```
raxmlHPC -s alg -m GTRGAMMA -q part -p 12345 -n TEST
```

- RAxML输出文件

RAxML_info.exampleRun，包含所使用的模型和算法信息

RAxML_log.exampleRun，包含运行时间，相似性值等log信息

RAxML_result.exampleRun，包含当前运行的最终树结构文件

RAxML_parsimonyTree.exampleRun，包含随机parsimony starting tree，除非通过-t指定了starting tree

RAxML_bestTree.exampleRun，包含彻底ML分析后的最佳评分的ML tree

- 示例

1. DNA/AA数据

```
raxmlHPC -m GTRGAMMA -p 12345 -s dna.phy -# 20 -n T6
```

```
raxmlHPC -m PROTGAMMAWAG -p 12345 -s protein.phy -# 20 -n T7
```

2. bootstrapping

```
raxmlHPC -m GTRGAMMA -p 12345 -# 20 -s dna.phy -n T13
```

该命令根据不同的starting trees生成了20个ML trees,同时将最相似性的tree输出到RAxML_betsTree.T13中。

使用-x调用快速bootstapping算法

```
raxmlHPC -m GTRGAMMA -p 12345 -x 12345 -# 100 -s dna.phy -n T19
```

3. partitioned analyses

```
raxmlHPC -m GTRGAMMA -p 12345 -q simpleDNApartition.txt -s dna.phy -n T21
```

simpleDNApartition.txt文件：

```
DNA, p1=1-30
```

```
DNA, p2=31-60
```

p1和p2为部分的任意名称。

```
raxmlHPC -m GTRGAMMA -p 12345 -q dna_protein_partition.txt -s dna_protein.phy -n T24
```

dan_protein_partition.txt文件：

```
DNA, p1=1-50
```

```
WAG, p2=51-110
```

p1部分为DNA,使用GTR+GAMMA；p2位蛋白,使用WAG+GAMMA

[RAxML-NG][https://github.com/amkozlov/raxml-ng/wiki/Tutorial]

RAxML-NG取代标准的RAxML，为对应的ExaML的超级计算版本。但是仅支持最重要和最常用的RAxML选项。

参数选项：

--evaluate，评估树的相似性(with model+brlen optimizition)

--search，搜索ML tree

--bootstrap，bootstrapping

--all，all-in-one(ML搜索+bootstrapping)

--check，查看比对的正确性同时移除空的列或行

- 准备比对文件

查看MSA(多重比对文件)是否存潜在错误或重复名称等，在开始前执行该步骤非常重要，超过50%的失败由于该原因

```
raxml-ng --check --msa prim.phy --model GTR+G --prefix T1
```

```
Alignment can be successfully read by RAxML-NG.


Execution log saved to: /Data_analysis/RAxML-NG_test/T1.raxml.log
```

针对大的比对文件，推荐使用--parse取代--check：

```
raxml-ng --parse --msa prim.phy --model GTR+G --prefix T2
```

该命令(parse)除了检查MSA外，将执行两有用操作，压缩并存储MSA为二进制格式文件(RAxML Binary Alignment, RBA)：T2.raxml.rba；评估CUPs/线程最佳数目，和内存需要

```
36 NOTE: Binary MSA file created: T2.raxml.rba
37
38 * Estimated memory requirements               : 2 MB
39
40 * Recommended number of threads / MPI processes: 2
```

- 生成tree

使用GTR+GAMMA，采用默认参数生成tree，同时使用2个线程

```
raxml-ng --msa prim.phy --model GTR+G --prefix T3 --threads 2 --seed 2
```

该命令将使用10个随机和10个parsimony-based starting trees，执行20次搜索，并且挑选最佳分值输出

默认参数设置足以合理应对很多实际分析。假如计算资源允许，可增加starting tree数目来彻底分析搜索

```
raxml-ng --msa prim.phy --model GTR+G --prefix T4 --threads 2 --seed 2 --tree pars{25},rand{25}
```

一般而言，可使用--search1参数执行快速直接的single random starting tree

```
raxml-ng --search1 --msa prim.phy --model GTR+G --prefix T5 --threads 2 --seed 2
```

比较三次运行结果：

```
$grep "Final LogLikelihood:" T{3,4,5}.raxml.log
T3.raxml.log:Final LogLikelihood: -5708.925454
T4.raxml.log:Final LogLikelihood: -5708.925454
T5.raxml.log:Final LogLikelihood: -5708.940534
```

看起来很棒，由于明显的相似性表现(likelihood surface seems to have a clear peak)。T5相似性结果稍微差些，使用--rfdist计算topological Robinson-Foulds(RF)距离：

```
cat T{3,4}.raxml.mlTrees T5.raxml.bestTree > mltrees
```

```
raxml-ng --rfdist --tree mltrees --prefix RF
```

```
Reading input trees from file: mltrees
Loaded 71 trees with 12 taxa.

Average absolute RF distance in this tree set: 0.000000
Average relative RF distance in this tree set: 0.000000
Number of unique topologies in this tree set: 1
```

共生成71个tree，无差别。以下例子将输出差异trees：`raxml-ng --msa fusob.phy --model GTR+G --prefix T6 --seed 2 --threads 2`

```
$grep "ML tree search #" T6.raxml.log
[00:00:04] ML tree search #1, logLikelihood: -9974.666508
[00:00:08] ML tree search #2, logLikelihood: -9974.675213
[00:00:13] ML tree search #3, logLikelihood: -9974.664137
[00:00:18] ML tree search #4, logLikelihood: -9974.667097
[00:00:22] ML tree search #5, logLikelihood: -9974.664311
[00:00:27] ML tree search #6, logLikelihood: -9974.666968
[00:00:32] ML tree search #7, logLikelihood: -9974.663409
[00:00:37] ML tree search #8, logLikelihood: -9974.664385
```

查看是否存在不同topology的tree

`raxml-ng --rfdist --tree T6.raxml.mlTrees --prefix RF6`

```
Reading input trees from file: T6.raxml.mlTrees
Loaded 20 trees with 38 taxa.

Average absolute RF distance in this tree set: 2.694737
Average relative RF distance in this tree set: 0.038496
Number of unique topologies in this tree set: 2
```

在20个推导的trees中出现2个不同的topology。

因此，使用multiple starting trees很重要

- Bootstrapping

通过重新对比对列取样执行标准非参数bootstrap，同时针对每一个bootstrap replicate MSA推导tree

`raxml-ng --bootstrap --msa prim.phy --model GTR+G --prefix T7 --seed 2 --threads 2`