生物信息学

第七章 分子进化与系统发育分析（2）

同义与非同义的核苷酸替代

- 同义替代：编码区的DNA序列，核苷酸的改变不改变编码的氨基酸的组成
- 非同义替代：核苷酸改变，从而改变编码氨基酸的组成
- 计算方法：
  - 进化通径法
  - Kimura两参数法
  - 采用密码子替代模型的最大似然法

Ka/Ks：计算及含义

- ***Ka***：每个非同义位点的非同义替代数目
- ***Ks***：每个同义位点的同义替代数目
- 一般计算公式：考虑序列上所有可能的同义位点（S）和非同义位点（N），通过双序列比对发现存在的同义位点（S_d）和非同义位点（N_d），存在：

$$Ka/Ks = \frac{\frac{N_d}{N}}{\frac{S_d}{S}}$$

Ka/Ks：计算及含义

- Ka/Ks ~ 1: 中性进化
- Ka/Ks << 1: 阴性选择，净化选择
- Ka/Ks >> 1: 阳性选择，适应性进化
- 多数基因为中性进化，约1%的基因受到阳性选择->决定物种形成、新功能的产生
- PAML, MEGA等工具：计算Ka/Ks及统计显著性

进化通径法：Nei-Gojobori

- 首先需要考虑：潜在的同义（S）和非同义位点数（N）
- 基本假设：所有核苷酸的替代率相等
- 用f_i表示某一个密码子第i位的核苷酸上发生同义替代的比例；(i=1,2,3)
- 所有密码子潜在的同义和非同义替代的位点数定义如下：$s=\sum_{i=1}^{3} f_i$，n=3-s

潜在的同义和非同义位点数的估计

- 例如对于Phe, 密码子TTT, 第三位T变成C时为同义替代，变成A/G为非同义替代
- 因此：
- s=0+0+1/3
- n=3-1/3=8/3
- 终止密码子忽略不计；如Cys的TGT, s=0.5

	T	C	A	G
T	TTT Phe (F) TTC " TTA Leu (L) TTG "	TCT Ser (S) TCC " TCA " TCG "	TAT Tyr (Y) TAC TAA **Ter** TAG **Ter**	TGT Cys (C) TGC TGA **Ter** TGG Trp (W)
C	CTT Leu (L) CTC " CTA " CTG "	CCT Pro (P) CCC " CCA " CCG "	CAT His (H) CAC " CAA Gln (Q) CAG "	CGT Arg (R) CGC " CGA " CGG "
A	ATT Ile (I) ATC " ATA " **ATG** Met (M)	ACT Thr (T) ACC " ACA " ACG "	AAT Asn (N) AAC " AAA Lys (K) AAG "	AGT Ser (S) AGC " AGA Arg (R) AGG "
G	GTT Val (V) GTC " GTA " GTG "	GCT Ala (A) GCC " GCA " GCG "	GAT Asp (D) GAC " GAA Glu (E) GAG "	GGT Gly (G) GGC " GGA " GGG "

整个序列的同义与非同义估计

- $S = \sum_{j=1}^{C} S_j$ 和N=3C-S; Sj为第j位密码子的s值，C为所有密码子的总数

- S+N=3C：所比较的核苷酸的总数

S_d与N_d的计算：进化通径

- 当一对密码子仅存在一个差异时，可以立即判断是同义还是非同义，进化通径只有一种可能；例如对于GTT(Val)和GTA(Val),s_d=1,n_d=0;而对于ATT(I)和ATG(M)，s_d=0,n_d=1
- 一对密码子存在两个差异时：两种进化通径(简约法，即最少需要)。例如：比较TTT(Phe)和GTA(Val):
  - (1) TTT(Phe)<->GTT(Val)<->GTA(Val)
  - (2) TTT(Phe)<->TTA(Leu)<->GTA(Val)
- s_d=1/2=0.5,n_d=3/2=1.5
- 同样，终止密码子不予考虑

S_d与N_d的计算：进化通径 (2)

- 一对密码子存在三个差异时：六种进化通径。例如：比较TTG(Leu)和AGA(Arg):
  - (1) TTG(Leu)<->ATG(Met)<->AGG(Arg)<->AGA(Arg)
  - (2) TTG(Leu)<->ATG(Met)<->ATA(Ile)<->AGA(Arg)
  - (3) TTG(Leu)<->TGG(Trp)<->AGG(Arg)<->AGA(Arg)
  - *(4) TTG(Leu)<->TGG(Trp)<->TGA(Ter)<->AGA(Arg)*
  - (5) TTG(Leu)<->TTA(Leu)<->ATA(Ile)<->AGA(Arg)
  - *(6) TTG(Leu)<->TTA(Leu)<->TGA(Ter)<->AGA(Arg)*
- 通径4,6忽略，通径(1),(2),(3),(5)同义替代数目1,0,1,1;非同义替代2,3,2,2，因此s_d=3/4,n_d=9/4

Ka/Ks的计算

- 统计显著性的检验：Fisher’s Exact Test!

Nei-Gojobori的改进版本

- Nei-Gojobori的原始版本假设四种核苷酸之间的替代是随机的
- 实际情况中：转换变化率应该比颠换变化率高，并且第三位上发生转换变化常常是同义的。因此这种情况下，估算的S将比Nei-Gojobori所估计的数值大

$$Ka/Ks = \frac{\frac{N_d}{N}}{\frac{S_d}{S}}$$

S增大，则Ka/Ks值将增大

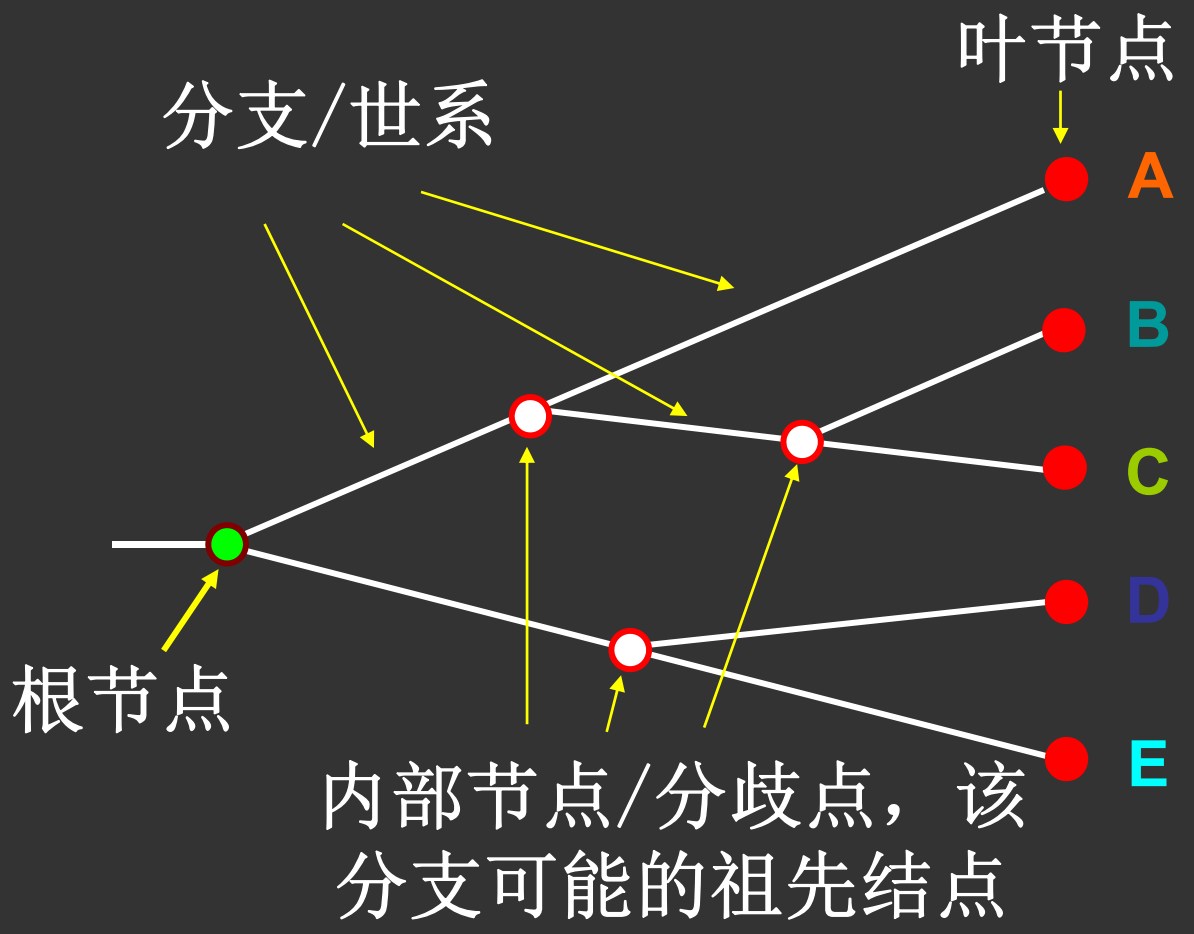
Nei-Gojobori的改进版本 (2)

- 转换/颠换比：R=α/(2β)
- 因此：$$\frac{\alpha}{\alpha+2\beta}=\frac{R}{1+2R}$$
- 无转换/颠换偏倚时，R=0.5
- 对于TTT(Phe), 假设R=0.8
  - Nei-Gojobori：s=0+0+1/3=1/3
  - Nei-Gojobori升级版：s=0+0+(0.8)/(1+0.8)=0.44
- R的估计：R=P/Q，通过对比较的两条DNA序列进行估计得到

系统发育树的构建

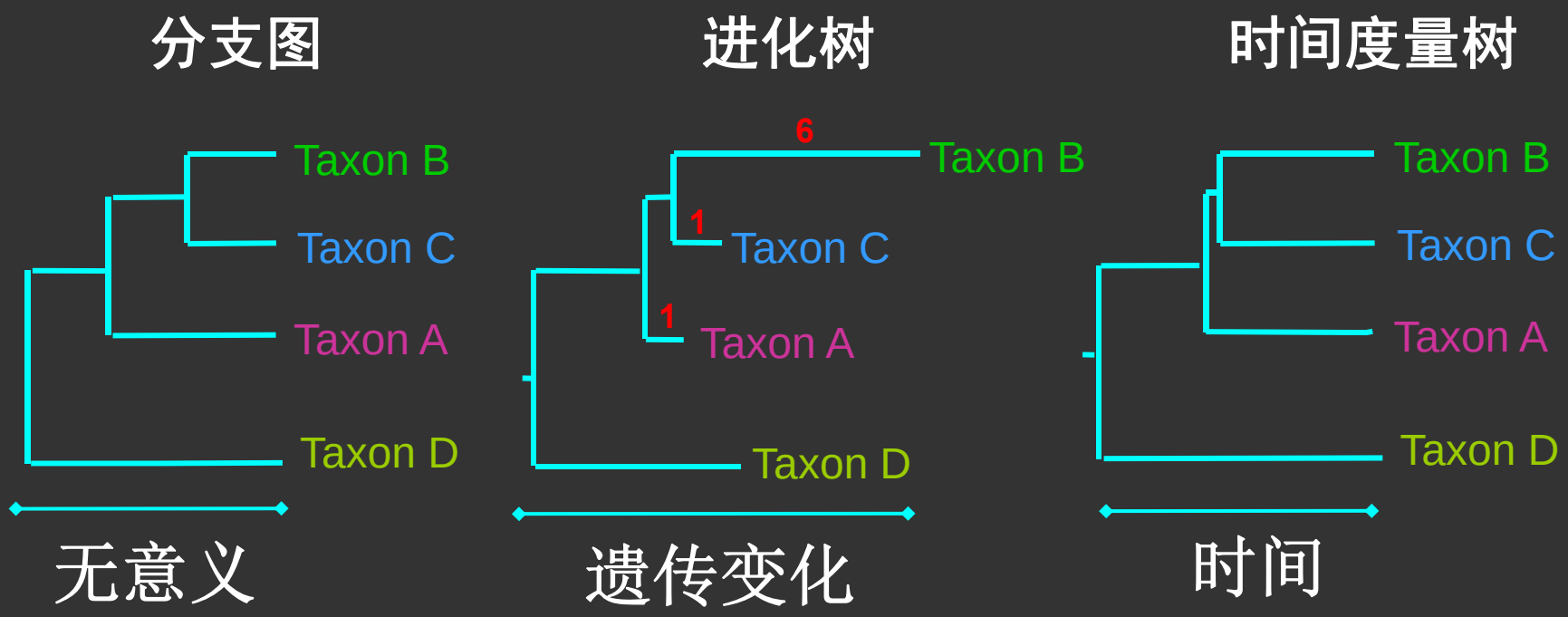
- 系统发育树：分子进化树/分子进化分析
- 通过进化树的构建，分析分子之间的起源关系，预测分子的功能
- 建树方法：
  - 最大简约法（Maximum Parsimony）
  - 距离法（Distance-based methods）
  - 最大似然性法（Maximum Likelihood）
  - 贝叶斯方法（Bayesian method）

系统发育树: 术语



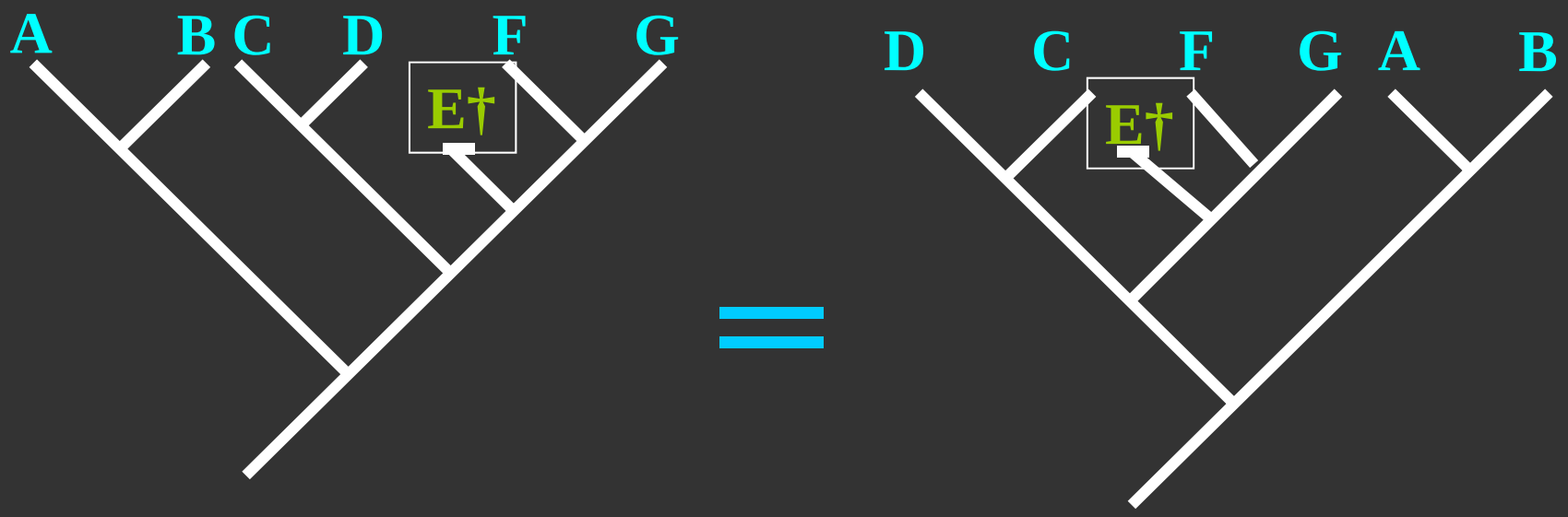
代表最终分类，可以是物种，群体，或者蛋白质、DNA、RNA分子等

系统发育树：三种类型

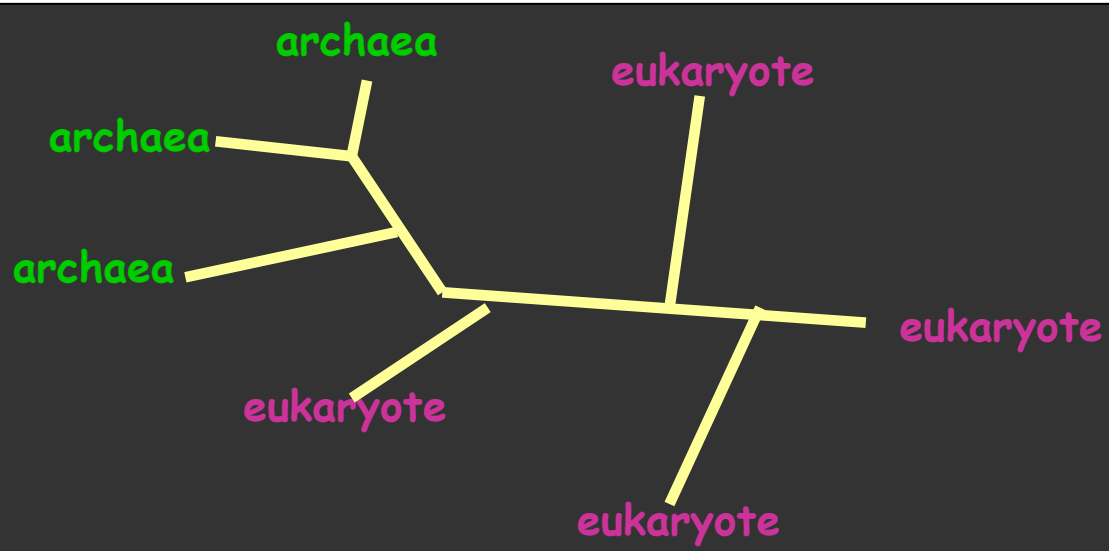


以上三种类型的系统发育树表示相同的分支状况，相同的进化关系

树只代表分支的拓扑结构



无根树，有根树，外围支



无根树

通过外围支
来确定树根

外围支

有根树

根

无根树和有根树：潜在的数目

#Taxa	无根树	有根树
3	1	3
4	3	15
5	15	105
6	105	945
7	945	10,395
…		
30	~$3.58X10^{36}$	~$2.04X10^{38}$

Taxa增多，计算量急剧增加，因此，目前算法都为优化算法，不能保证最优解

系统发育树重建分析步骤

多序列比对（自动比对，手工校正）

选择建树方法以及替代模型

建立进化树

进化树评估

系统发育树重建的基本方法

- 最大简约法 (maximum parsimony, MP)
- 距离法 (distance)
- 最大似然法 (maximum likelihood, ML)
- 贝叶斯方法 (Bayesian method)

最大简约法 (MP)

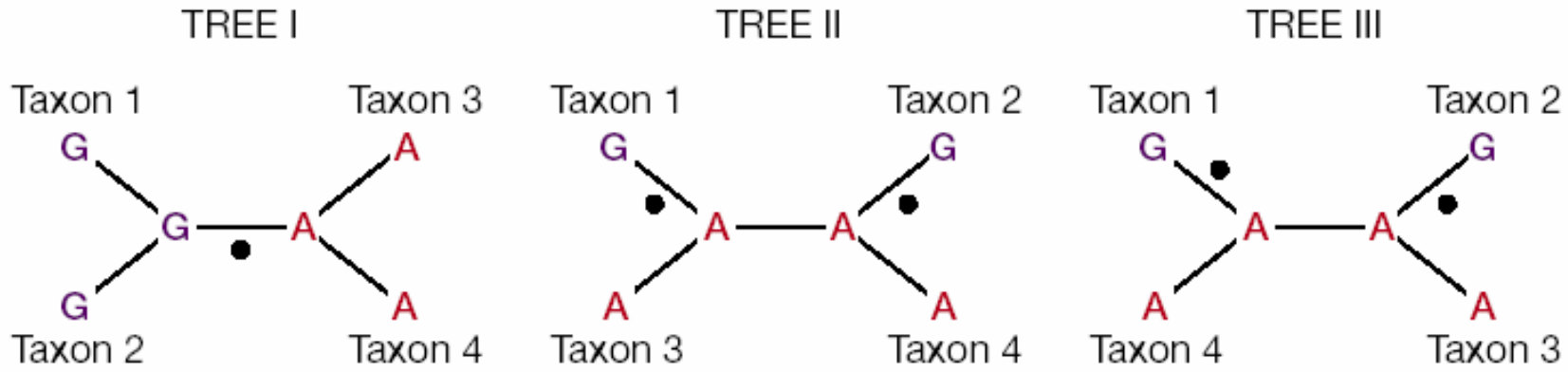
- 理论基础为奥卡姆剃刀（Ockham）原则：计算所需替代数最小的那个拓扑结构，作为最优树
- 在分析的序列位点上没有回复突变或平行突变，且被检验的序列位点数很大的时候，最大简约法能够推导获得一个很好的进化树
- 优点：不需要在处理核苷酸或者氨基酸替代的时候引入假设（替代模型）
- 缺点：分析序列上存在较多的回复突变或平行突变，而被检验的序列位点数又比较少的时候，可能会给出一个不合理的或者错误的进化树推导结果

信息位点 (Informative sites)

- 至少在2个taxa中保守
- 至少存在2个不同碱基/氨基酸
- 每个不同碱基/氨基酸至少出现2次

Taxa	Sequence position (sites) and character								
	1	2	3	4	5	6	7	8	9
1	A	A	G	A	G	T	G	C	A
2	A	G	C	C	G	T	G	C	G
3	A	G	A	T	A	T	C	C	A
4	A	G	A	G	A	T	C	C	G

Adapted from Li and Graur 1991.



Total tree length: 1 | 2 | 3 (2 plus 2 other character arrangements in trees II and III)

● is a substitution

上例

- Position 5, 7, 9为信息位点
- 基于position 5的三个MP树:
  - Tree 1长度1，Tree 2 & 3长度2
- Tree 1更为简约：总长：4
- Tree 2长5；Tree 3长6
- 计算结果：MP tree的最优结果为tree 1

距离法

- 又称距离矩阵法，首先通过各个物种之间的比较，根据一定的假设（进化距离模型）推导得出分类群之间的进化距离，构建一个进化距离矩阵。进化树的构建则是基于这个矩阵中的进化距离关系

简单的距离矩阵

A. Sequences

sequence A	ACGCGTTGGGCGATGGCAAC
sequence B	ACGCGTTGGGCGACGGTAAT
sequence C	ACGCATTGAATGATGATAAT
sequence D	ACACATTGAGTGATAATAAT

B. Distances between sequences, the number of steps required to change one sequence into the other.

n_{AB}	3
n_{AC}	7
n_{AD}	8
n_{BC}	6
n_{BD}	7
n_{CD}	3

C. Distance table

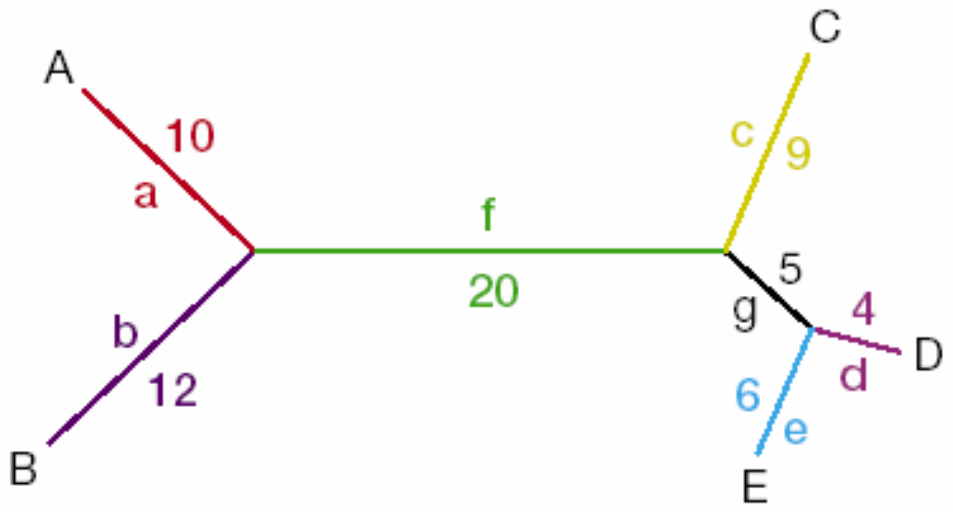
	A	B	C	D
A	–	3	7	8
B	–	–	6	7
C	–	–	–	3
D	–	–	–	–

通过距离矩阵建树的方法

- 由进化距离构建进化树的方法有很多，常见有：
  - **Fitch-Margoliash Method (FM法): 对短支长非常有效**
  - **Neighbor-Joining Method (NJ法/邻接法):求最短支长，最通用的距离方法**
  - **Unweighted Pair Group Method (UPGMA法)**

Fitch-Margoliash方法 (FM法)

	A	B	C	D	E
A	—	22	39	39	41
B	—	—	41	41	43
C	—	—	—	18	20
D	—	—	—	—	10
E	—	—	—	—	—

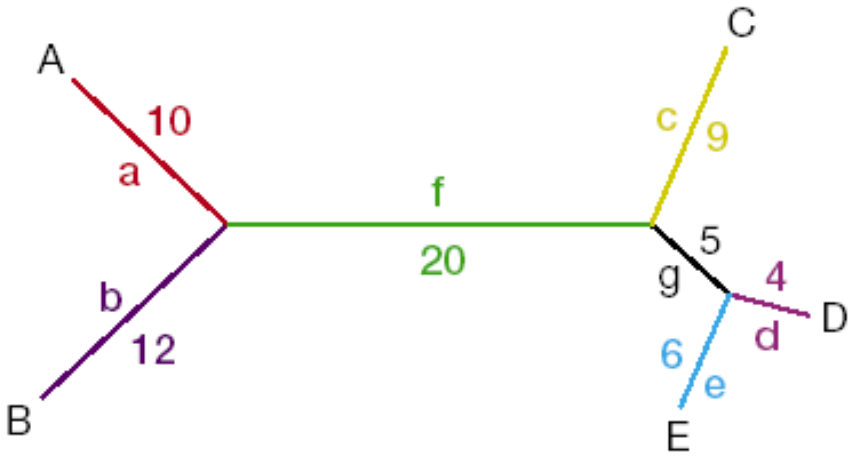


D和E最接近！

	A	B	C	D	E
A	—	22	39	39	41
B	—	—	41	41	43
C	—	—	—	18	20
D	—	—	—	—	10
E	—	—	—	—	—

分成三组：D, E, 以及ABC

	D	E	ave. ABC
D	—	10	32.7
E	—	—	34.7
average ABC	—	—	—



DE距离=d+e　(1)

D到ABC间的平均距离=d+m　(2)

E到ABC间的平均距离=e+m　(3)

(2)-(3)+(1)

d=4,e=6

	D	E	ave. ABC
D	—	10	32.7
E	—	—	34.7
average ABC	—	—	—

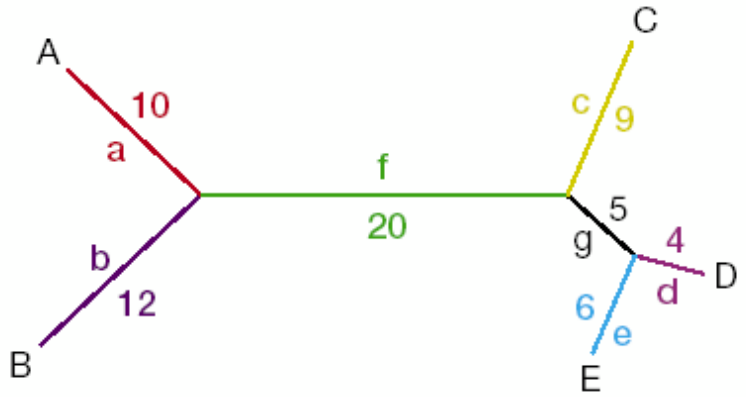
C最接近DE！

	A	B	C	(DE)
A	—	22	39	40
B	—	—	41	42
C	—	—	—	19
(DE)	—	—	—	—

分成三组：C, DE, 以及AB

	DE	C	Ave. AB
DE	—	19	41
C	—	—	40
Ave. AB	—	—	—

	DE	C	Ave. AB
DE	—	19	41
C	—	—	40
Ave. AB	—	—	—



c+g+(e+d)/2=19 (1)

c+f+(a+b)/2=40 (2)

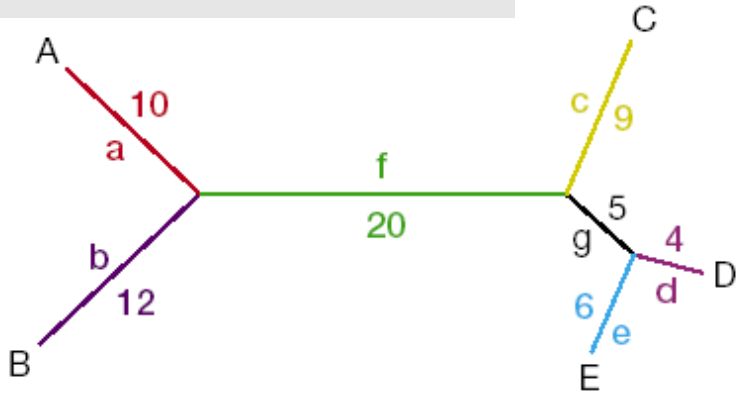
(e+d)/2+(a+b)/2+f+g=41 (2)

(1)+(2)-(3)

得：c=9

	A	B	C	D	E
A	—	22	39	39	41
B	—	—	41	41	43
C	—	—	—	18	20
D	—	—	—	—	10
E	—	—	—	—	—

	DE	C	Ave. AB
DE	—	19	41
C	—	—	40
Ave. AB	—	—	—

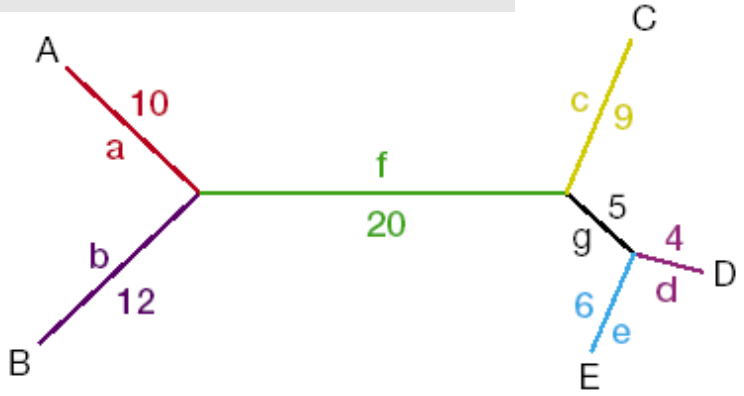


c+g+(e+d)/2=19

(e+d)/2=5,c=9,则g=5

	A	B	C	D	E
A	—	22	39	39	41
B	—	—	41	41	43
C	—	—	—	18	20
D	—	—	—	—	10
E	—	—	—	—	—

	DE	C	Ave. AB
DE	—	19	41
C	—	—	40
Ave. AB	—	—	—

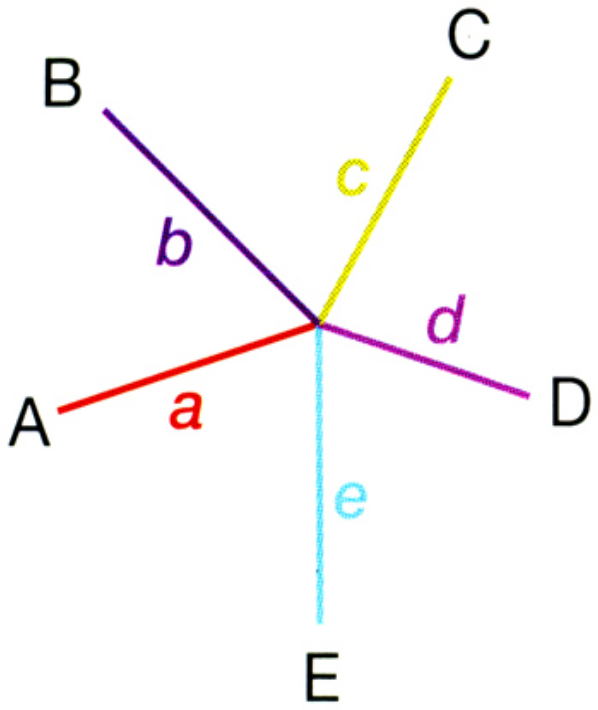


由：(a+b)/2+f+g+(d+e)/2=41 得：f=20

由：a+f+c=39 得：a=10，则b=12

NJ/邻接法

- 与FM方法非常类似
- 保证总的支长最短



- 总支长：a+b+c+d+e=314/4=78.5

找到距离最近的两个点

- 任意两个节点选为相邻序列的总支长计算公式:

$$S_{mn} = [(\Sigma\, d_{im} + d_{\text{in}})/2(N-2)] + d_{mn}/2 + \Sigma\, d_{ij}/N - 2$$

- 计算S_{AB}, S_{BC}, S_{CD}, S_{DE}…等数值

$$S_{\text{AB}} = 67.7,\ S_{\text{BC}} = 81,\ S_{\text{CD}} = 76,\ \text{and}\ S_{\text{DE}} = 70,$$

- 该例中， S_{AB}最小

计算A, B的分支长度

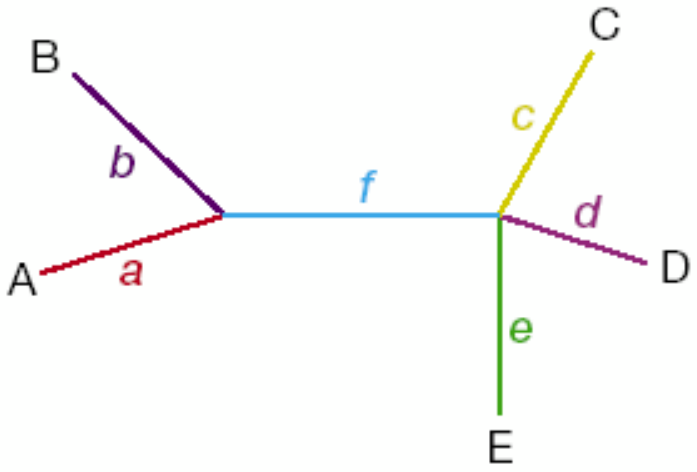
$$a = [d_{AB}+(d_{AC}+d_{AD}+d_{AE})/3-(d_{BC}+d_{BD}+d_{DE})/3]/2 =$$

$$(22+39.7-41.70)/2=10,$$

$$b = [d_{AB}+(d_{BC}+d_{BD}+d_{BE})/3-(d_{AC}+d_{AD}+d_{AE})/3]/2 =$$

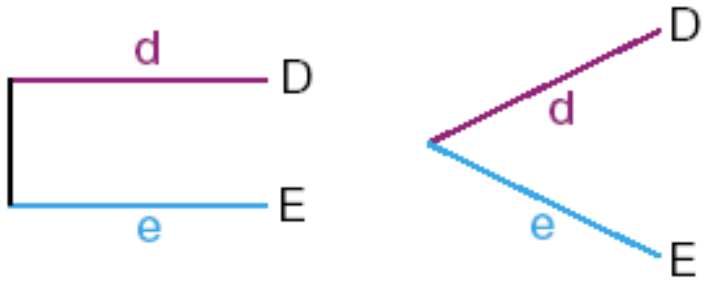
$$(22+41.7-39.7)/2=12.$$

把A、B看成一个新的复合序列，构建一个新的距离表，重复以上过程



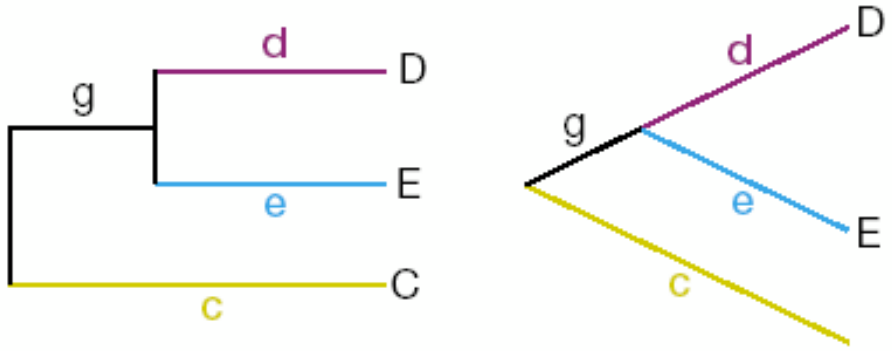
UPGMA法

	A	B	C	D	E
A	—	22	39	39	41
B	—	—	41	41	43
C	—	—	—	18	20
D	—	—	—	—	10
E	—	—	—	—	—



d=e=10/2=5

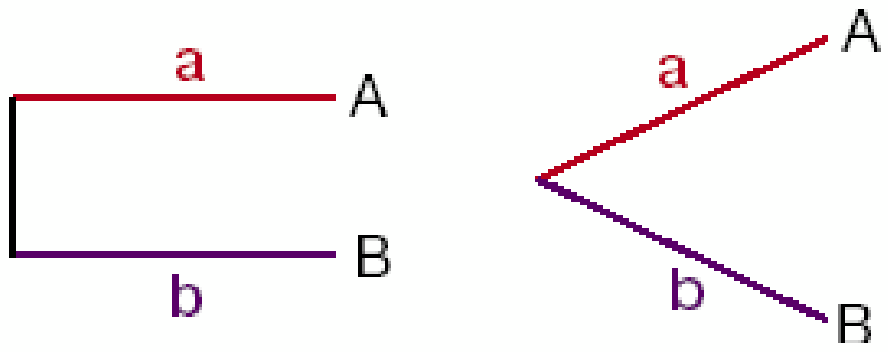
	A	B	C	(DE)
A	—	22	39	40
B	—	—	41	42
C	—	—	—	19
(DE)	—	—	—	—



c=19/2=9.5

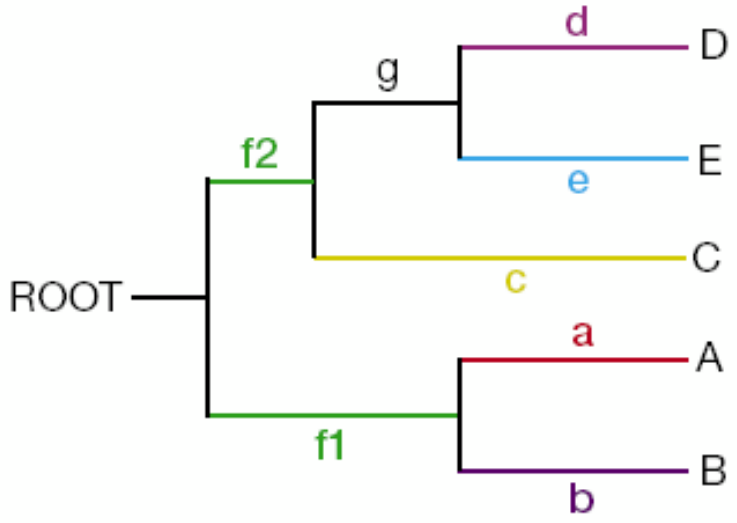
g=c-d=9.5-5=4.5

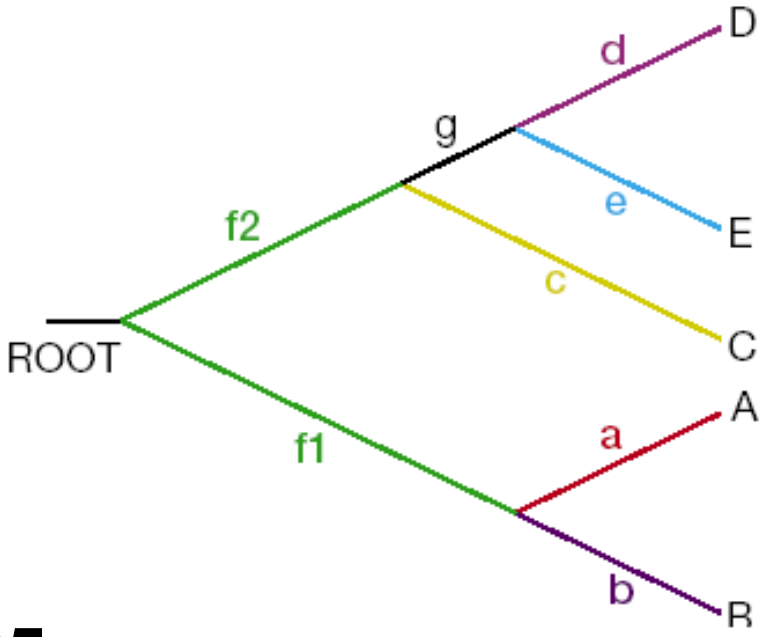
	A	B	(CDE)
A	-	22	39.5
B	-	-	41.5
(CDE)	-	-	-



a=b=22/2=11

	(AB)	(CDE)
(AB)	-	40.5
(CDE)	-	-



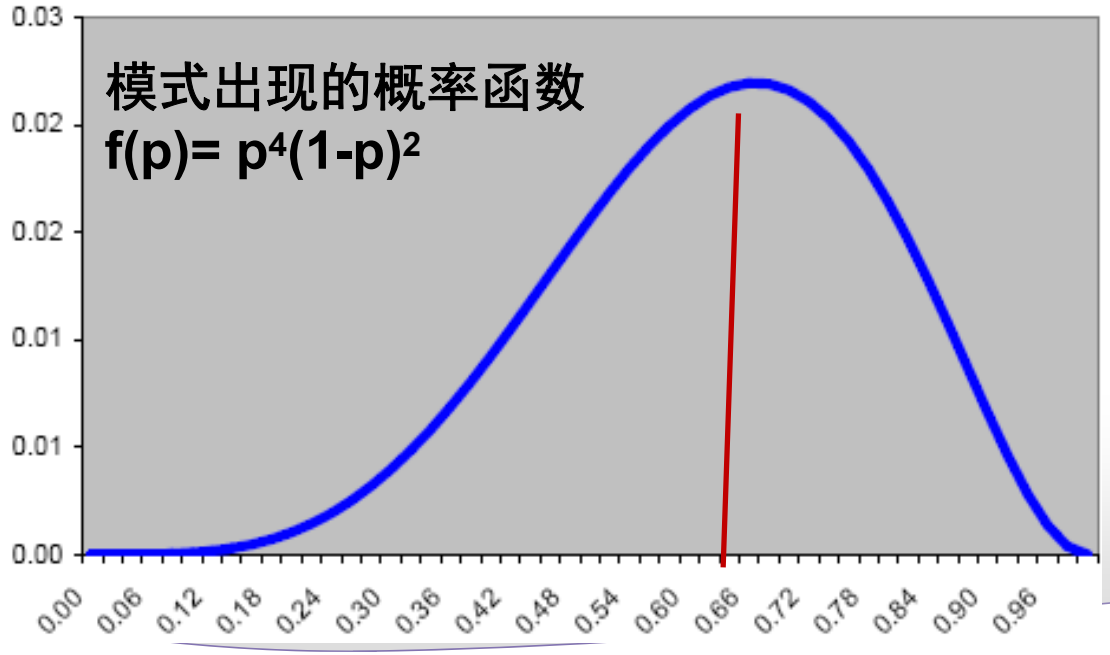


f1+a=f2+c=40.5/2=20.25

f1=9.25 , f2=11.75

基于似然性（likelihood）的推断

- 硬币两个面，正面 (H)，背面 (T)
  - 六次投掷后：HHHHTT
- 正面出现的概率*p*，背面出现的概率1-*p*
- 当*p*=0.67时，概率函数达到最大值
- 因此，正面出现的概率可能是0.67



最大似然法 (ML)

- 考虑一个进化模型*M*，例如Jukes-Cantor
  - 已知根节点上每个位点的先验分布
  - 所有的位点在进化中是独立且等同 (independently and identically, i.i.d.)
  - *p(x→y | t)* 即在分支*t*上*x*被替代成*y*的概率
- 在Jukes-Cantor模型中，替代速率3α= γ

α是参数，实际计算中可令αt=1

$$p(x \to y|t) = \frac{1}{4}(1 - e^{-4\alpha t}) \text{ (if } x \neq y)$$

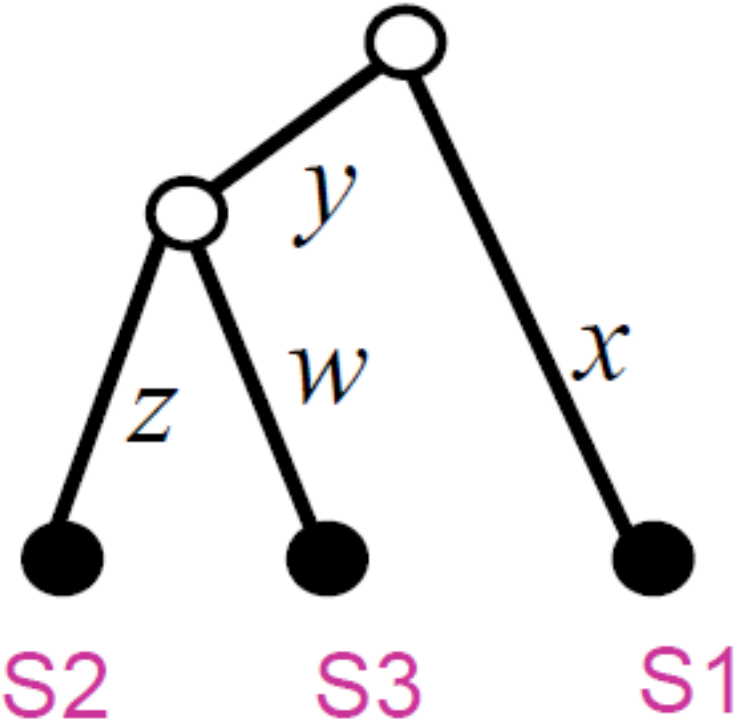
$$p(x \to y|t) = \frac{1}{4}(1 + 3e^{-4\alpha t}) \text{ (if } x = y)$$

最大似然法 (ML)

- 寻找树*H*包含*k*个叶节点，从而最大化条件概率
  - *L*=Pr[Data | *H*, *M*]
  - *L*即为在模型*M*下的似然性 (likelihood)
- 由于位点进化独立且等同，*L*等于比对结果中第*i*列的似然性L_i 的乘积

$$L=\prod_{1\le i\le n} L_i$$

- 其中L_i=Pr[Data$^{(i)}$ | *H*, *M*]

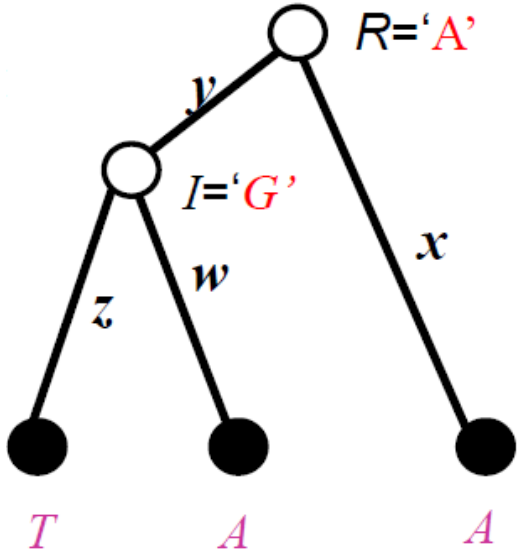


似然性的计算

- 假设第*i*列的位点，三条序列的字符为*T*, *A*, *A*, 对于其中一个可能的树*H*:
  - 根节点*R*有四种可能：

    A, *G*, *C*, *T*
  - 内部节点*I* 也有四种可能：

    A, *G*, *C*, *T*



- 对于每一对特定状态，例如: *R*=*A*, *I*=*G*, 进化概率为

$$p[R=A]\,p(A\rightarrow A|\,x)\,p(A\rightarrow G|\,y)\,p(G\rightarrow T|\,z)\,p(G\rightarrow A|\,w)$$

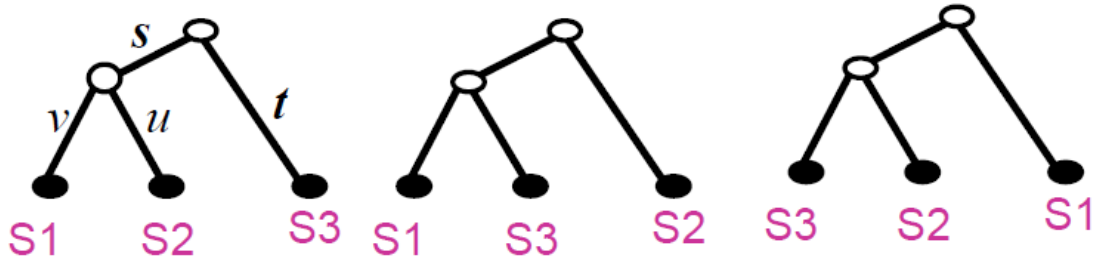
- 考虑所有的概率，则似然性L_i为

$$\Pr[\text{Data}^{(i)}\,|\,H,M]=\sum_{S\in\Delta}\sum_{Q\in\Delta}p[R=S]p(S\rightarrow A\,|\,t)p(S\rightarrow Q\,|\,s)p(Q\rightarrow T\,|\,v)p(Q\rightarrow A\,|\,u)$$

where $\Delta=\{A,\ G,\ C,\ T\}$.

例如，给定三条序列
S1: TGG S2: AGG S3: AGC

需要考虑三种拓扑结构



对于每一个树（例如最左边的树），基于Jukes-Cantor模型计算似然性Pr[Data | H, M] = $L_1 \times L_2 \times L_3$

$$L_1 = \sum_{I\in\Delta}\sum_{Q\in\Delta} p[R=I]p(I\to A\,|\,t)p(I\to Q\,|\,s)p(Q\to T\,|\,v)p(Q\to A\,|\,u)$$

$$L_2 = \sum_{I\in\Delta}\sum_{Q\in\Delta} p[R=I]p(I\to G\,|\,t)p(I\to Q\,|\,s)p(Q\to G\,|\,v)p(Q\to G\,|\,u)$$

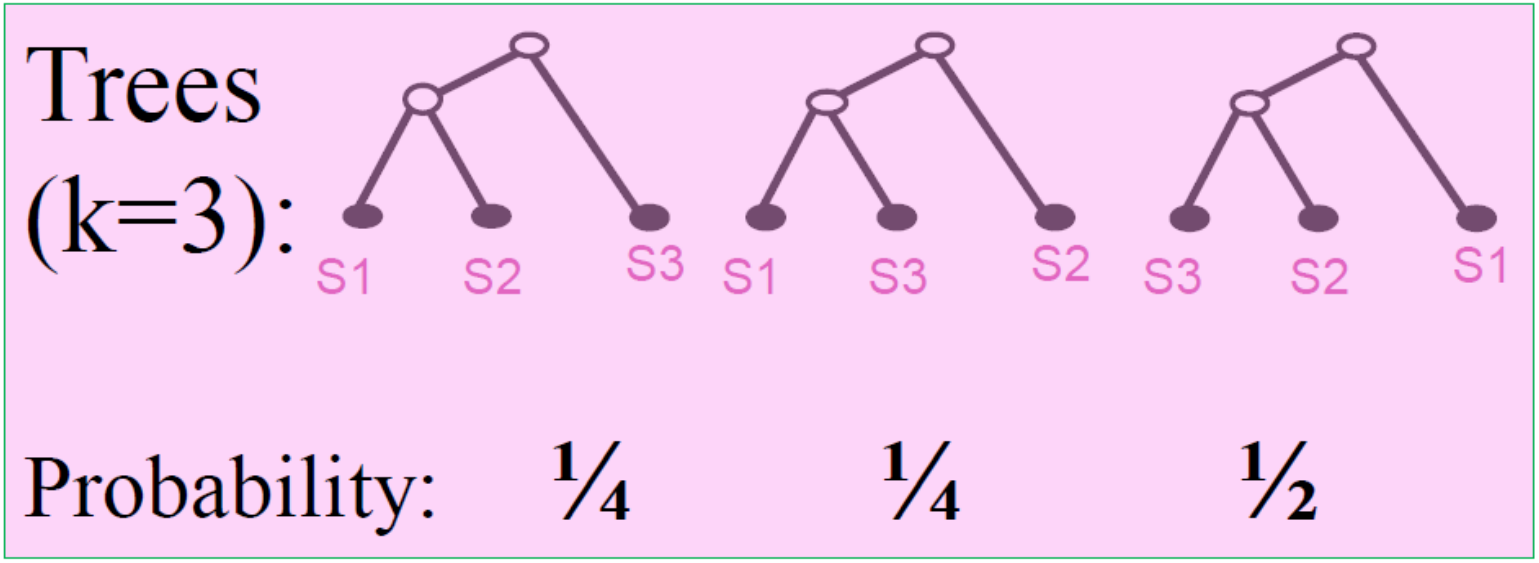
$$L_3 = \sum_{I\in\Delta}\sum_{Q\in\Delta} p[R=I]p(I\to C\,|\,t)p(I\to Q\,|\,s)p(Q\to G\,|\,v)p(Q\to G\,|\,u)$$

最大似然法 (ML)

- 对于每个树，我们需要确定4个分支的长度
  - 最大化似然性Pr[Data |H, M] , 该函数包含16 + 16 + 16 项，每一项是5个概率的乘积
- 最大似然法非常耗费时间
- NP-hard问题：太多树需要考虑

贝叶斯方法 (Bayesian Method)

- 考虑我们有一个比对A (data)，包含k条序列S_1, S_2, … S_k
- 假设我们知道所有树的概率分布，即先验概率分布(prior probability distribution)，需要独立于数据本身，例如：



贝叶斯方法 (Bayesian Method)

- 对于数据A，利用贝叶斯理论对给定树T计算概率

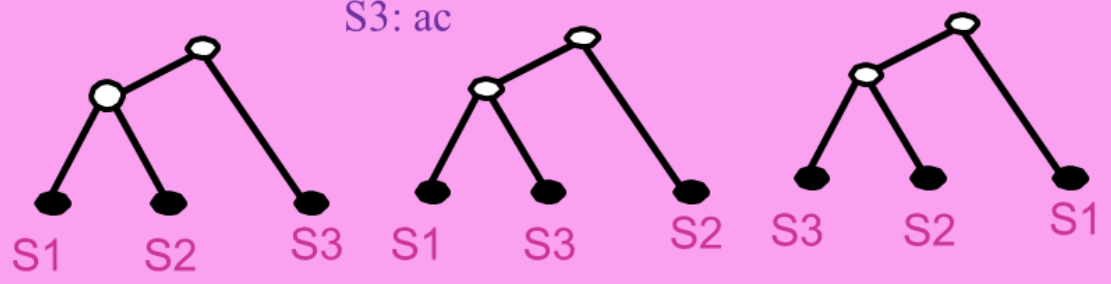
$$\Pr[T \mid A] = \frac{\Pr[T, A]}{\Pr[A]} = \frac{\Pr[A \mid T] \times \Pr[T]}{\Pr[A]}$$

- Pr[T|Data]是根据给定数据所观测到该树的概率，称为树的后验概率 (**posterior** probability)

比对数据:

S1: ga
S2: ga
S3: ac

树:



先验概率:	1/4	1/4	1/2
后验概率:	0.37	0.31	0.32

贝叶斯方法 (Bayesian Method)

- 如何确定先验概率分布？
  - Markov Chain Monte Carlo (MCMC)
  - 数据采样，建立先验的概率分布
- 如果先验概率是均匀分布，则贝叶斯方法等同于最大似然性方法
- 计算时间比最大似然性方法更久

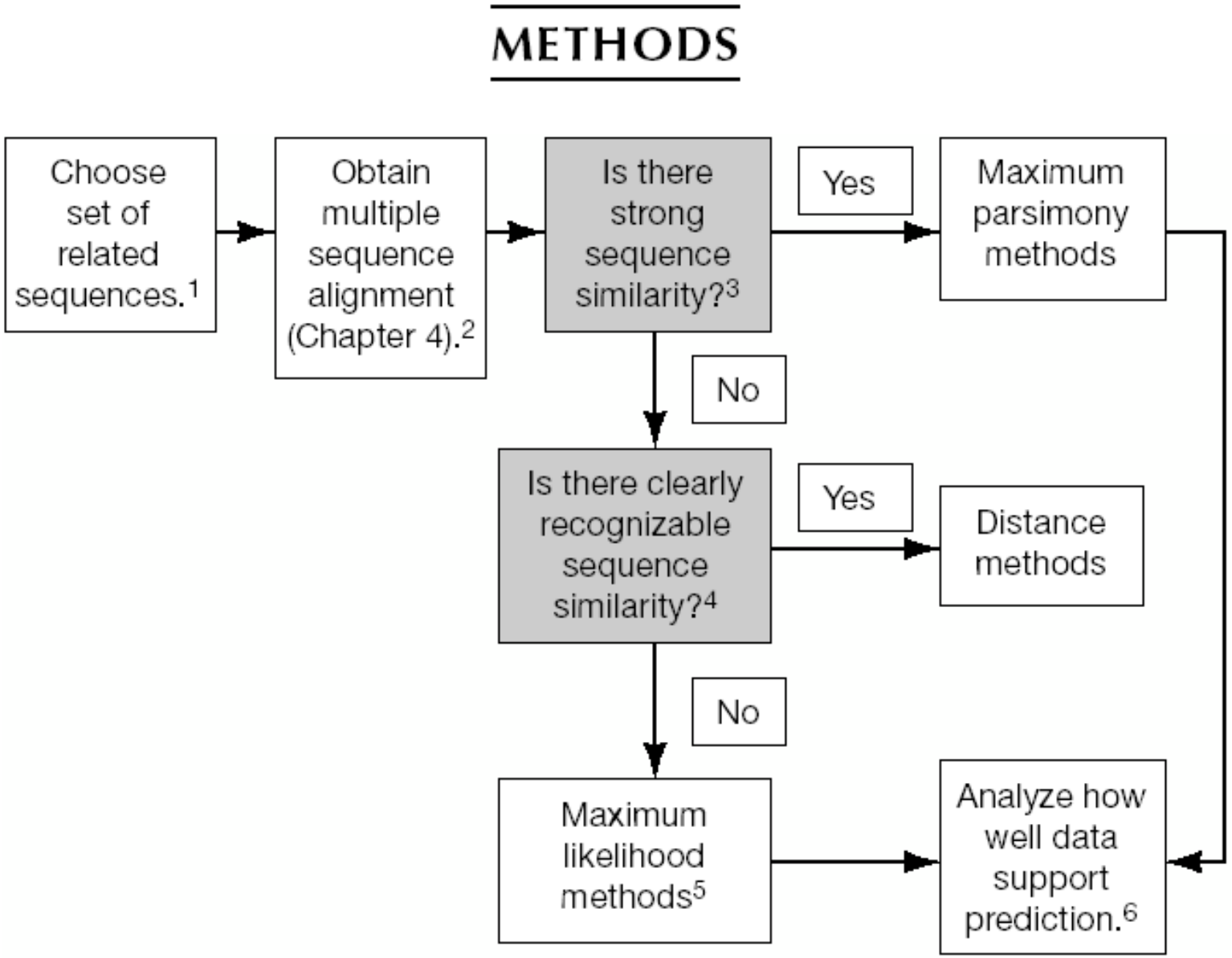
建树方法总结

快 ← **慢**

Neighbor-Joining UPGMA	Parsimony Method	Maximum Likelihood	Bayesian Method

不够准确 → **准确**

构建进化树的一般原则


METHODS
Choose set of related sequences.[1]
Obtain multiple sequence alignment (Chapter 4).[2]
Is there strong sequence similarity?[3]
Yes
Maximum parsimony methods
No
Is there clearly recognizable sequence similarity?[4]
Yes
Distance methods
No
Maximum likelihood methods[5]
Analyze how well data support prediction.[6]

构建进化树的一般原则 (2)

- 可靠的待分析数据
- 准确的多序列比对
- 选择合适的建树方法：
  - 序列相似程度高，NJ, MP首选
  - 序列相似程度较低，ML, 贝叶斯首选
  - 序列相似程度太低，无意义
- 一般采用两种及以上方法构建进化树，无显著区别可接受

选择外围支 (Outgroup)

- 选择一个或多个已知与分析序列关系较远的序列作为外围支
- 外围支可以辅助定位树根
- 外围支序列必须与剩余序列关系较近，但外围支序列与其他序列间的差异必须比其他序列之间的差异更显著

自展法

- 进化树的可靠性分析:自展法 (Bootstrap Method)
- 从排列的多序列中随机有放回的抽取某一列，构成相同长度的新的排列序列
- 重复上面的过程，得到多组新的序列
- 对这些新的序列进行建树，再观察这些树与原始树是否有差异，以此评价建树的可靠性

原始排列

Alpha AACAAC

Beta AACCCC

Gamma ACCAAC

Delta CCACCA

Epsilon CCAAAC

Bootstrap1

Alpha ACAAAC

Beta ACCCCC

Gamma ACAAAC

Delta CACCCA

Epsilon CAAAAC

Bootstrap2

Alpha AAAACC

Beta AACCCC

Gamma CCAACC

Delta CCCCAA

Epsilon CCAACC

Bootstrap3

Alpha ACAAAC

Beta ACCCCC

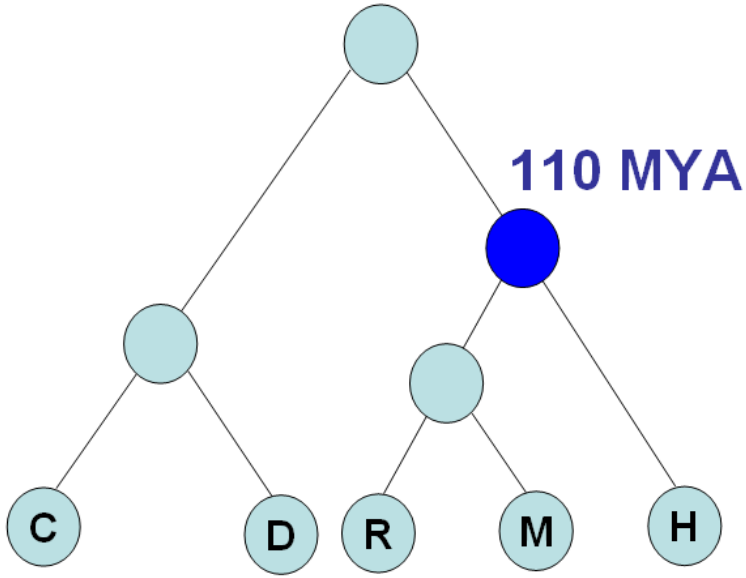
Gamma CCAAAC

Delta CACCCA

Epsilon CAAAAC

分子钟与线性树

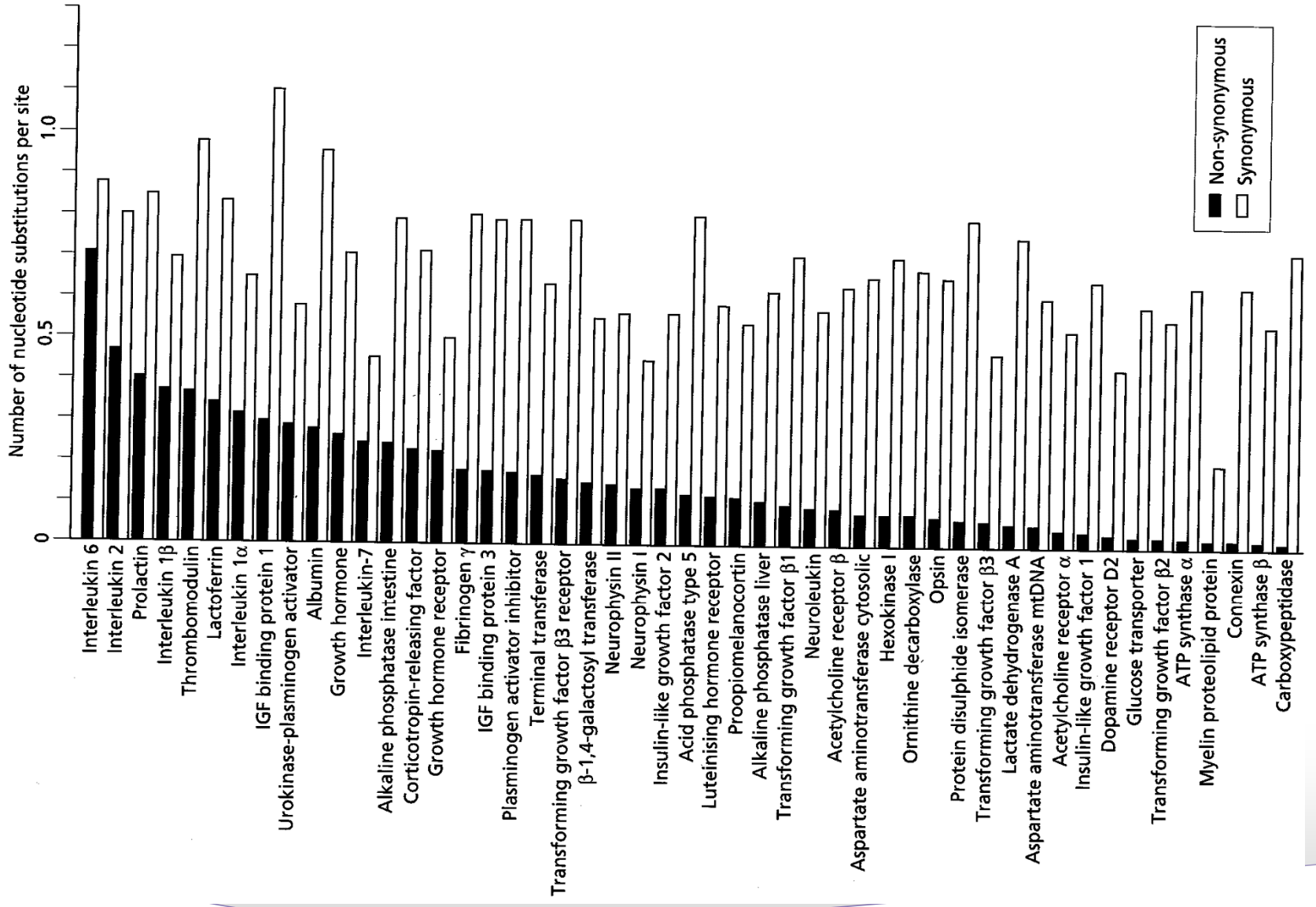
- 物种分化时间的推断：最理想应该是化石证据
- 由于化石证据的不足，可以采用分子数据推测物种的分化时间
- 给定一个进化树，已知：
  - 分支长度
  - 其中一个分歧点的分化时间
- 推测所有分歧点的分化时
- 间：突变的速率恒定！



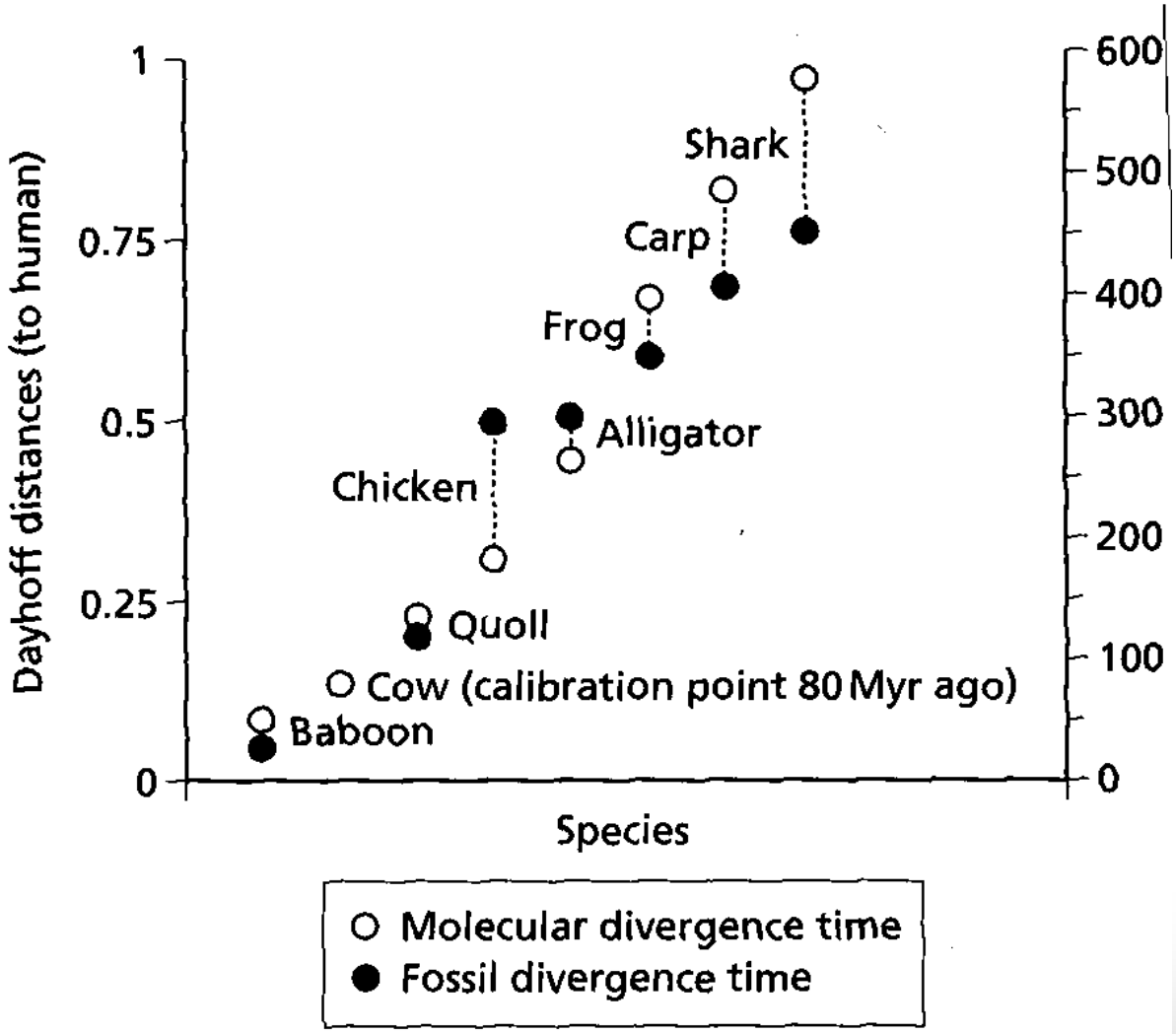
实际数据中

- 同义替代与非同义替代的速率不同
- 不同的基因/蛋白质，其进化的速率不同
- 然而，对于特定的基因，具有一定的、恒定的进化速率

基因同义替代与非同义替代的速率



速率恒定的证据：血色素


Dayhoff distances (to human)
1
0.75
0.5
0.25
0
600
500
400
300
200
100
0
Shark
Carp
Frog
Alligator
Chicken
Quoll
Cow (calibration point 80 Myr ago)
Baboon
Species
Molecular divergence time
Fossil divergence time

分子钟假设

- 序列之间的遗传差异的数量是自分化以来的时间的函数
- 分子变化的速率相当稳定，可以用来预测分化的时间

分子钟: 进化时间的估计

- 遗传距离d的计算：
  - 氨基酸序列：p-距离，d-距离
  - DNA序列：Jukes-Cantor距离，Kimura距离
- 物种分歧点：使用考古数据确定共有祖先，确定分化时间T
- 计算分子的分化/进化的速率：r=d/2T
- 对新的序列，计算分化时间：
  - $T_{new}=d_{new}/2r$

物种分化时间：化石证据

- 灵长目-啮齿动物： ~80 Myr ago
- 哺乳动物-鸟类： ~310 Myr ago
- 哺乳动物-两栖类： ~350 Myr ago
- 四肢动物-硬骨鱼： ~430 Myr ago
- 脊椎动物-果蝇 (昆虫)： ~830 Myr ago

- *Nature Genetics 31, 205 - 209 (2002)*