아미노산 서열과 코돈 서열의 진화 모형

(아미노산 서열과 코돈 서열의 진화 모형(서태건 2023)논문의 요약 및 추가 설명입니다; ver.2025.01.24)

아미노산 치환 모형 **(Amino acid model)**

4×4 행렬로 DNA 모형을 설정하는 방식과 비슷하게 아미노산 치환 모형은 20×20 행렬로 아미노산 사이의 치환율을 정의한다. 행렬의 (i, j) 원소는 i번째 아미노산 a_i가 j번째 아미노산 a_j로 치환되는 치환율을 나타내며 다음과 같이 정의된다.

$$R_{a_i a_j} = s_{a_i a_j} \pi_{a_j}, \tag{2}$$

여기에서 $s_{a_i a_j}$ 값은 치환율을 지정하는 모수이고 대량의 아미노산 서열의 비교로 추정한 고정된 값이다.[5] π_{a_j} 는 아미노산 a_j의 빈도이다. DNA 모형과 마찬가지로(서태건 2022 참조) 아미노산 모형도 시간가역성(time reversibility)을 가정한다. 시간가역성은 $\pi_{a_i} R_{a_i a_j} = \pi_{a_j} R_{a_j a_i}$라는 성질을 가짐을 의미하며, 이는 $s_{a_i a_j} = s_{a_j a_i}$를 의미한다.[6]

$S_{a_i a_j}$: EC (Exchangeability Coefficient)

```
0.425093
0.276818 0.751878
0.395144 0.123954 5.076149
2.489084 0.534551 0.528768 0.062556
0.969894 2.807908 1.695752 0.523386 0.084808
1.038545 0.363970 0.541712 5.243870 0.003499 4.128591
2.066040 0.390192 1.437645 0.844926 0.569265 0.267959 0.348847
0.358858 2.426601 4.509238 0.927114 0.640543 4.813505 0.423881 0.311484
0.149830 0.126991 0.191503 0.010690 0.320627 0.072854 0.044265 0.008705 0.108882
0.395337 0.301848 0.068427 0.015076 0.594007 0.582457 0.069673 0.044261 0.366317 4.145067
0.536518 6.326067 2.145078 0.282959 0.013266 3.234294 1.807177 0.296636 0.697264 0.159069 0.137
1.124035 0.484133 0.371004 0.025548 0.893680 1.672569 0.173735 0.139538 0.442472 4.273607 6.312
0.253701 0.052722 0.089525 0.017416 1.105251 0.035855 0.018811 0.089586 0.682139 1.112727 2.592
1.177651 0.332533 0.161787 0.394456 0.075382 0.624294 0.419409 0.196961 0.508851 0.078281 0.249
4.727182 0.858151 4.008358 1.240275 2.784478 1.223828 0.611973 1.739990 0.990012 0.064105 0.182
2.139501 0.578987 2.000679 0.425860 1.143480 1.080136 0.604545 0.129836 0.584262 1.033739 0.302
0.180717 0.593607 0.045376 0.029890 0.670128 0.236199 0.077852 0.268491 0.597054 0.111660 0.619
0.218959 0.314440 0.612025 0.135107 1.165532 0.257336 0.120037 0.054679 5.306834 0.232523 0.299
2.547870 0.170887 0.083688 0.037967 1.959291 0.210332 0.245034 0.076701 0.119013 10.649107 1.70

0.079066 0.055941 0.041977 0.053052 0.012937 0.040767 0.071586 0.057337 0.022355 0.062157 0.099

 A   R   N   D   C   Q   E   G   H   I   L   K   M   F   P   S   T   W   Y   V
Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys Met Phe Pro Ser Thr Trp Tyr Val
```

그림 1. PAML프로그램에 포함된 lg.dat 파일. LG 아미노산 치환 모형의 $s_{a_i a_j}$ 값과 아미노산 빈도가 저장되어 있다. PAML 프로그램에는 다수의 *.dat 파일이 있어 위와 같은 포맷으로 아미노산 모형에 관한 정보를 담고 있다.

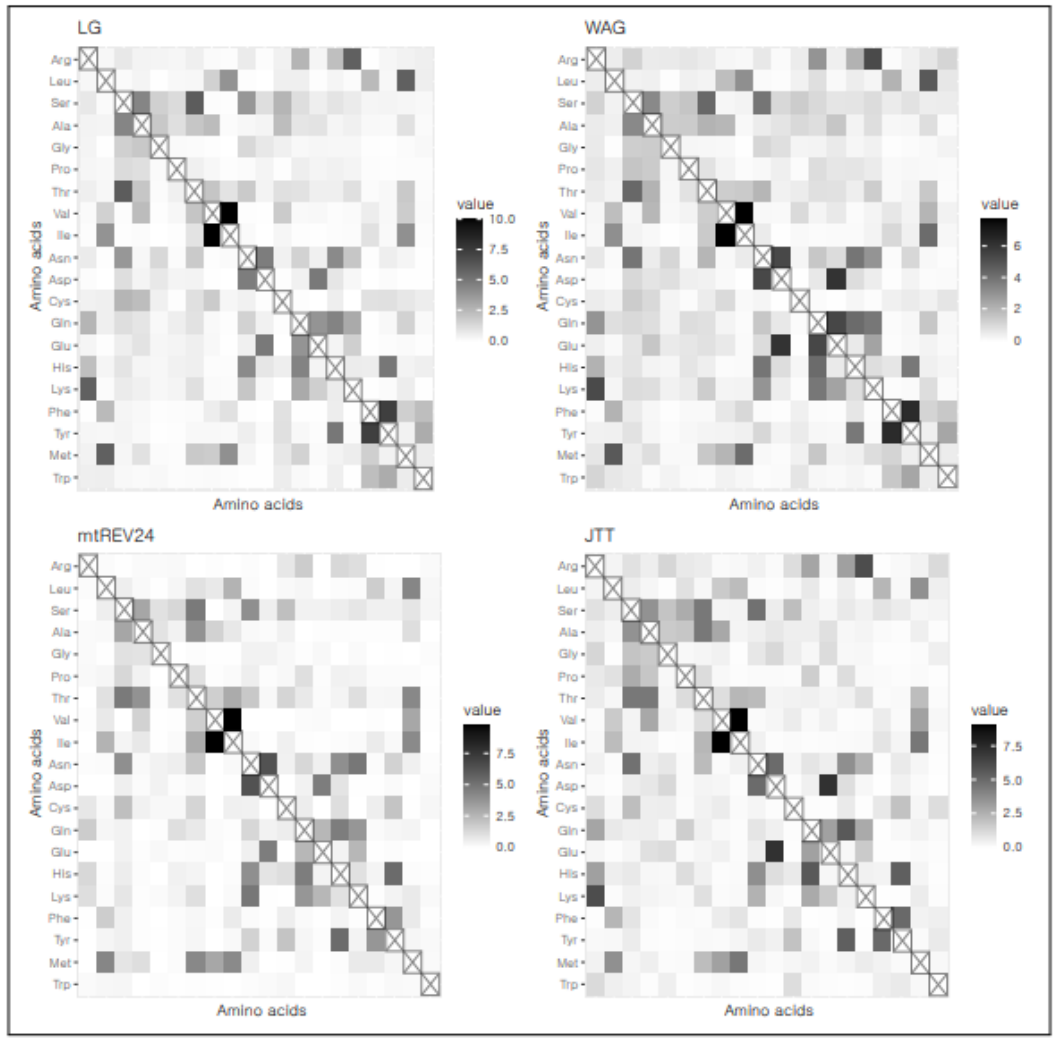


그림 2. 아미노산 모형의 $s_{a_i a_j}$ 값들. $s_{a_i a_j}$ 값이 크면 해당 아미노산 쌍은 치환이 빈번하게 일어남을 의미한다. 임의로 LG(Le and Gascuel 2008), WAG(Whelan and Goldman 2001), mtREV24(Adachi et al. 1996), JTT(Jones et al. 1992) 네 종류 아미노산 치환 모형을 선택했다. 가로축 아미노산 순서 (좌에서 우)는 세로축 아미노산 순서(위에서 아래)와 같다.

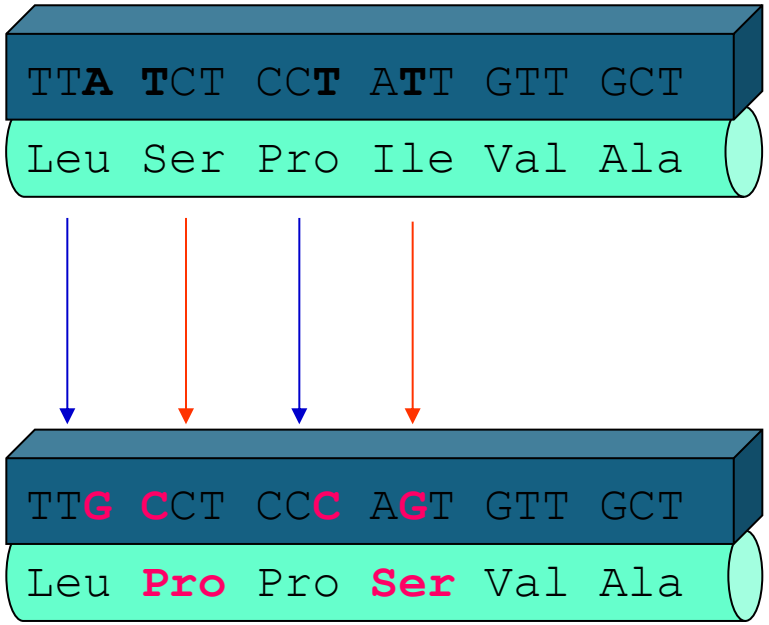
- 단백질 코딩 영역의 DNA 염기치환의 분류

- Synonymous 염기치환 (동의치환) → 아미노산이 변하지 않음

  ex. TTA (Leu) → TTG (Leu)

- Nonsynonymous 염기치환 (비동의 치환) → 아미노산이 변함

  ex. TCT(Ser) → CCT(Pro)



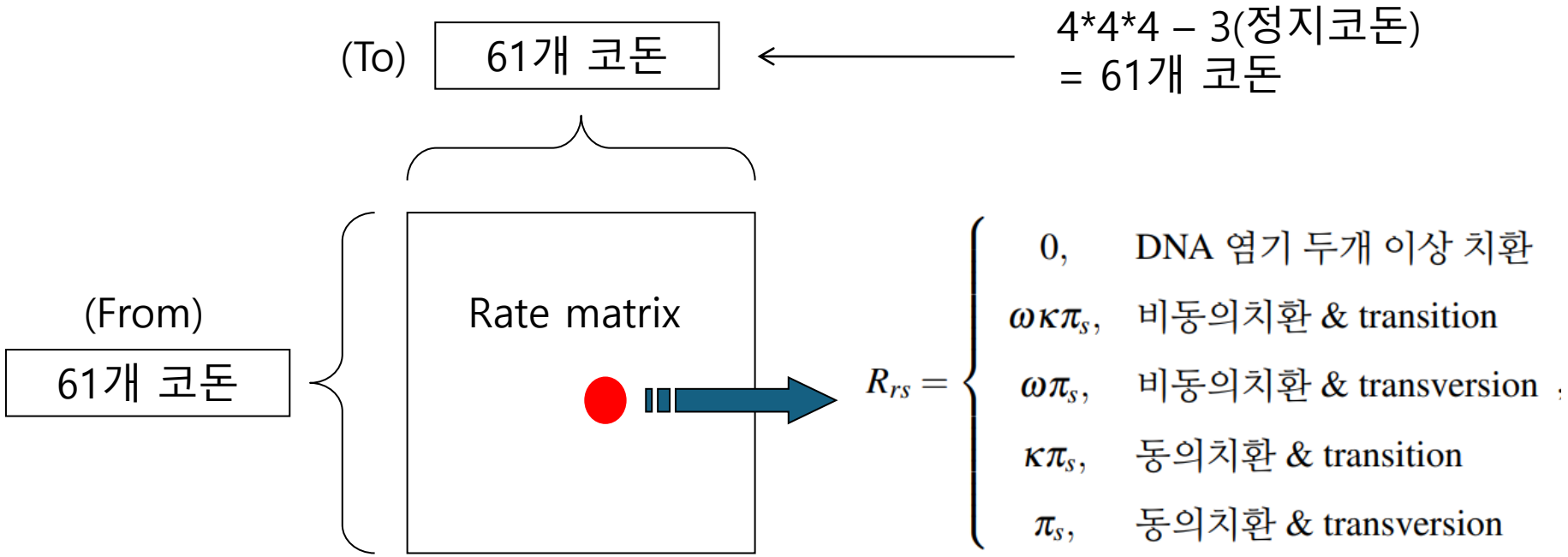
Synonymous substitution

Nonsynonymous substitution

- Synonymous, Nonsynonymous 치환을 구분함으로서 얻을수 있는 정보

- Synonymous 치환
  - 자연선택의 영향을 받지 않음
- Nonsynonymous 치환
  - 자연선택의 영향을 받음
  - 단백질 기능향상 → 치환속도 증가
  - 단백질 기능저하 → 치환속도 감소
- Nonsynonymous/synonymous 치환 속도의 비율 (= ω, dn/ds, Ka/Ks)
  - > 1 : **positive (diversifying) selection**
  - $= 1$: **neutral evolution**
  - < 1 : **purifying selection**

코돈치환 모형



(e.g., Goldman & Yang 1994)

코돈 치환 모형

코돈 모형은 3개의 정지코돈을 제외한 61개의 코돈 사이의 치환율을 규정한 모형이다.[9] 최초의 코돈 모형 (Goldman and Yang 1994; Muse and Gaut 1994)이 제시된 이래 여러 가지 버전의 코돈 모형이 소개되어 왔으나 여기서는 비교적 널리 사용되는 Goldman and Yang (1994)의 코돈 치환 모형 (이하 GY94모형)에 대해 간략히 설명하겠다. GY94 모형은 코돈 r로부터 코돈s로 매우 짧은 시간동안 일어나는 치환율을 다음과 같이 규정한다.

$$R_{rs} = \begin{cases} 0, & \text{DNA 염기 두개 이상 치환} \\ \omega\kappa\pi_s, & \text{비동의치환 \& transition} \\ \omega\pi_s, & \text{비동의치환 \& transversion} \\ \kappa\pi_s, & \text{동의치환 \& transition} \\ \pi_s, & \text{동의치환 \& transversion} \end{cases}, \tag{3}$$

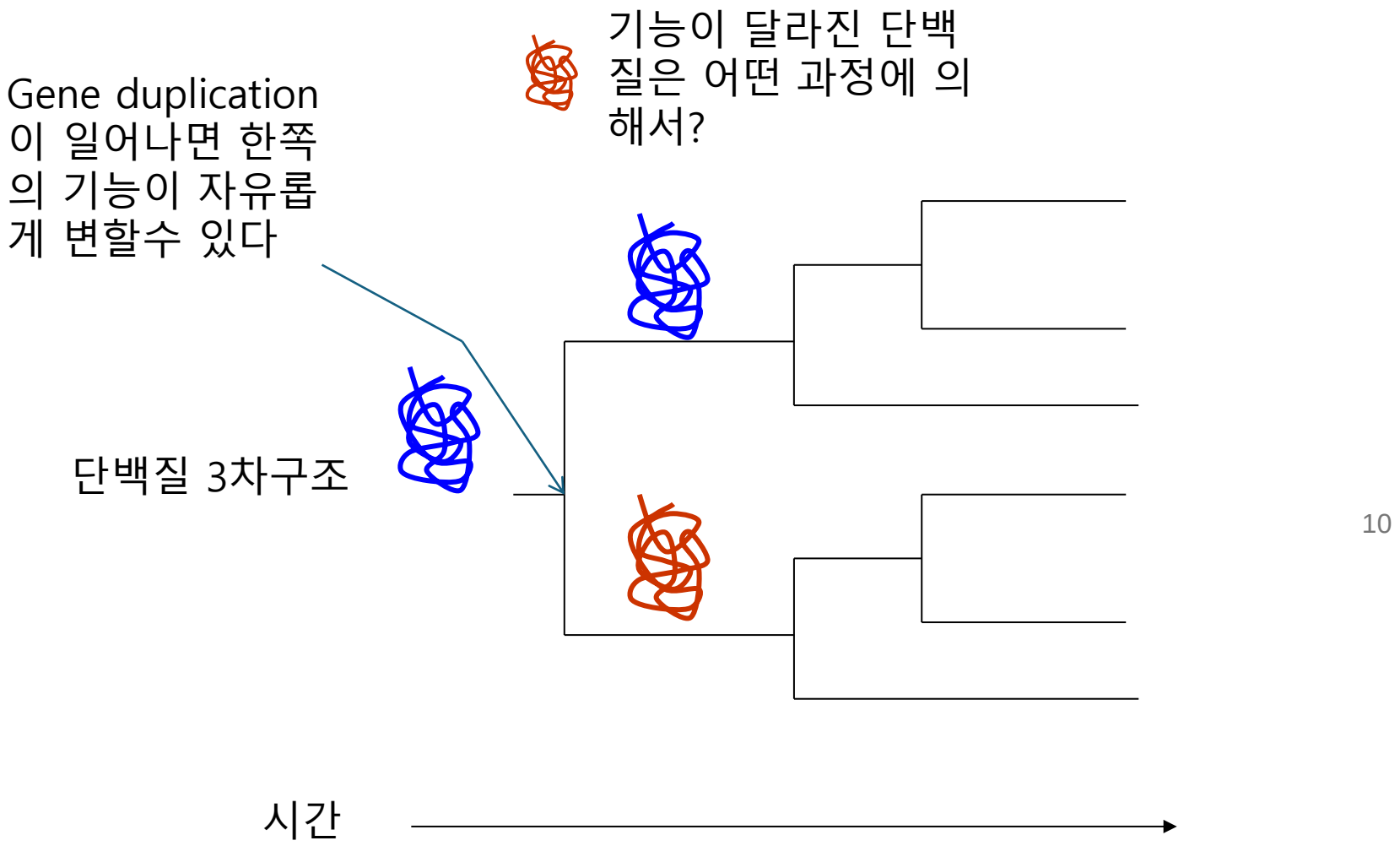
여기서 π_s는 코돈 s의 빈도를 의미한다. GY94 모형은 '코돈 치환은 DNA 염기치환이 축적되어 일어난다'고 가정한다. 그리고 아주 짧은 시간 동안에는 DNA 염기치환이 한번씩만 일어나고 두번 이상의 염기치환이 동시에 일어나지는 않는다고 가정한다. 한번의 염기치환이 일어날때, 염기치환이 transition타입이면 κ를 곱하여 transition과 transversion의 차이를 모형화 했다. 이는 DNA 모형에서 HKY모형(Hasegawa et al. 1985; 서태건 2022)과 유사한 설정이다. 동의치환은 아미노산을 변화시키지 않으므로 표현형에 영향

al. 1985; 서태건 2022)과 유사한 설정이다. 동의치환은 아미노산을 변화시키지 않으므로 표현형에 영향을 주지 않는 반면,[10] 비동의 치환은 직접적으로 표현형에 영향을 끼칠 수 있으므로 자연선택의 영향을 받는다. 만약 치환된 아미노산이 개체의 생존에 유리하다면 그 비동의 코돈치환은 동의치환에 비하여 빠르게 일어날 것이고, 생존에 불리하다면 그 비동의 코돈 치환은 일어나지 않거나(개체가 성체가 되기 전에 사망하여 유전자가 집단내에서 소멸함) 혹은 느리게 일어날 것이다. 이를 모형화 하여 비동의치환의 상대적인 발생율을 ω 모수로 표현하였다. 비동의치환이 자연선택이 적용되지 않아 중립적으로 진화하였을 경우, 양의 자연선택이 작용하였을 경우, 음의 자연선택이 작용하였을 경우 각각 $\omega = 1, \omega > 1, \omega < 1$ 이 된다. 따라서 데이터로부터 추정한 미지의 모수 ω의 크기로 해당 데이터에 자연선택이 작용하였는지 판단 할 수 있는 것이다.

$$
\begin{array}{c}
 & \begin{array}{cccccc} CGT & CGC & CGA & CGG & AGA & AGG \end{array} \\
\begin{array}{c} CGT \\ CGC \\ CGA \\ CGG \\ AGA \\ AGG \end{array} &
\begin{bmatrix}
- & \kappa\pi_{CGC} & \pi_{CGA} & \pi_{CGG} & 0 & 0 \\
\kappa\pi_{CGT} & - & \pi_{CGA} & \pi_{CGG} & 0 & 0 \\
\pi_{CGT} & \pi_{CGC} & - & \kappa\pi_{CGG} & \pi_{AGA} & 0 \\
\pi_{CGT} & \pi_{CGC} & \kappa\pi_{CGA} & - & 0 & \pi_{AGG} \\
0 & 0 & \pi_{CGA} & 0 & - & \kappa\pi_{AGG} \\
0 & 0 & 0 & \pi_{CGG} & \kappa\pi_{AGA} & -
\end{bmatrix}
\end{array}
$$

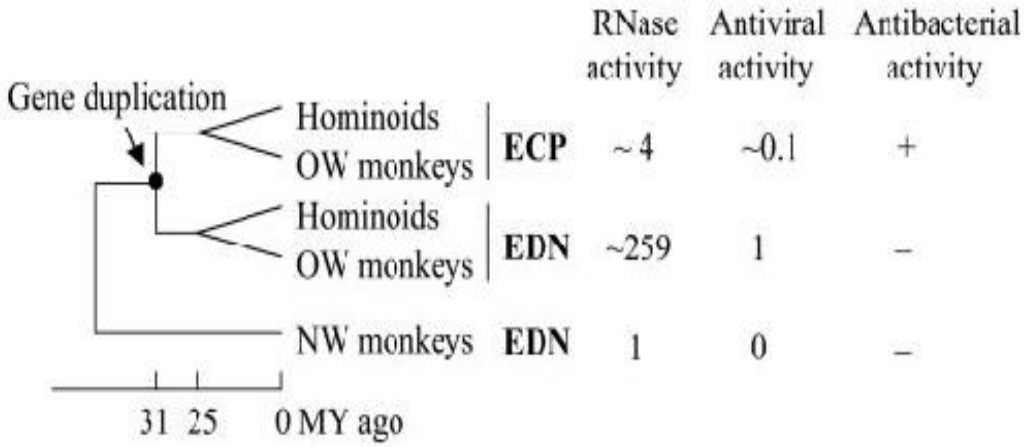
(아르기닌을 코딩하는 코돈끼리의 치환률 예시)

코돈치환 모델을 이용한 단백질의 분자진화 연구



- Synonymous/Nonsynonymous 염기치환을 이용한 연구의 예

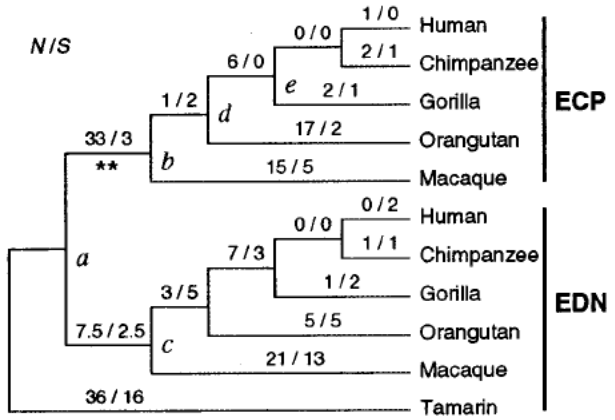
Zhang et al. (1998; PNAS 95 (7) 3708-3713)



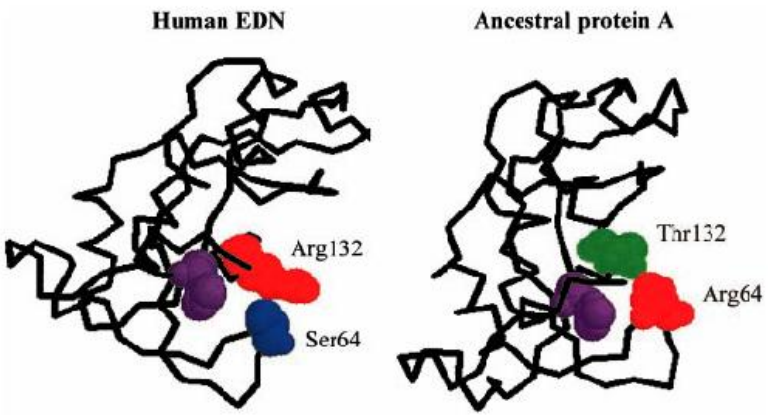
- NW (New World) monkeys : 중남미에 분포
- OW(Old World) monkeys : 아시아,아프리카에 분포

- EDN: eosinophil-derived neurotoxin
- ECP : eosinophil cationic protein
- EDN은 본래 antibacterial activity를 가지고 있지 않음. ECP는 가짐
- Hominoids/OW monkey는 EDN의 활성이 크다

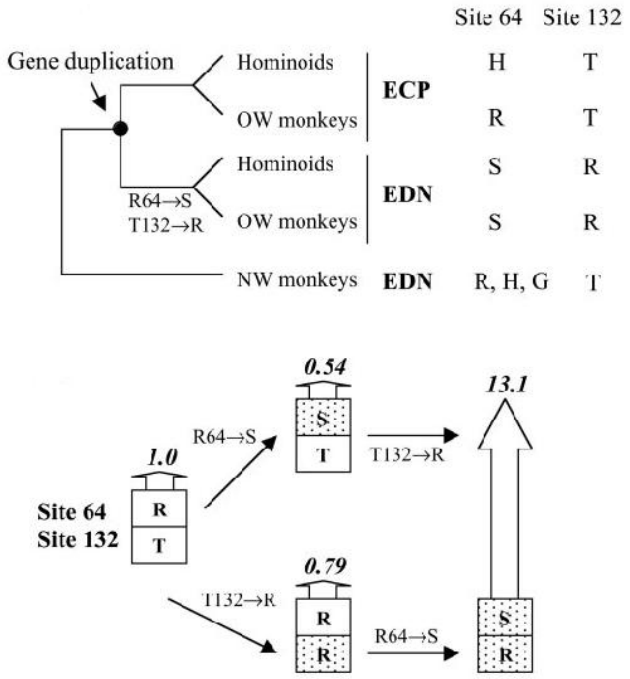
Zhang & Rosenberg (2002)



Gene duplication 직후에 Nonsynonymous 염기치환 (N)이 synonymous 염기치환(S)에 비하여 매우 왕성하게 일어났음을 알수 있다 -> 이 nonsynonymous 치환이 단백질의 기능과 밀접하게 관련이 있을것이라 추측됨



조상형의 단백질을 추정 → EDN의 기능에 있어서 64번째, 132번째 아미노산 치환이 중요

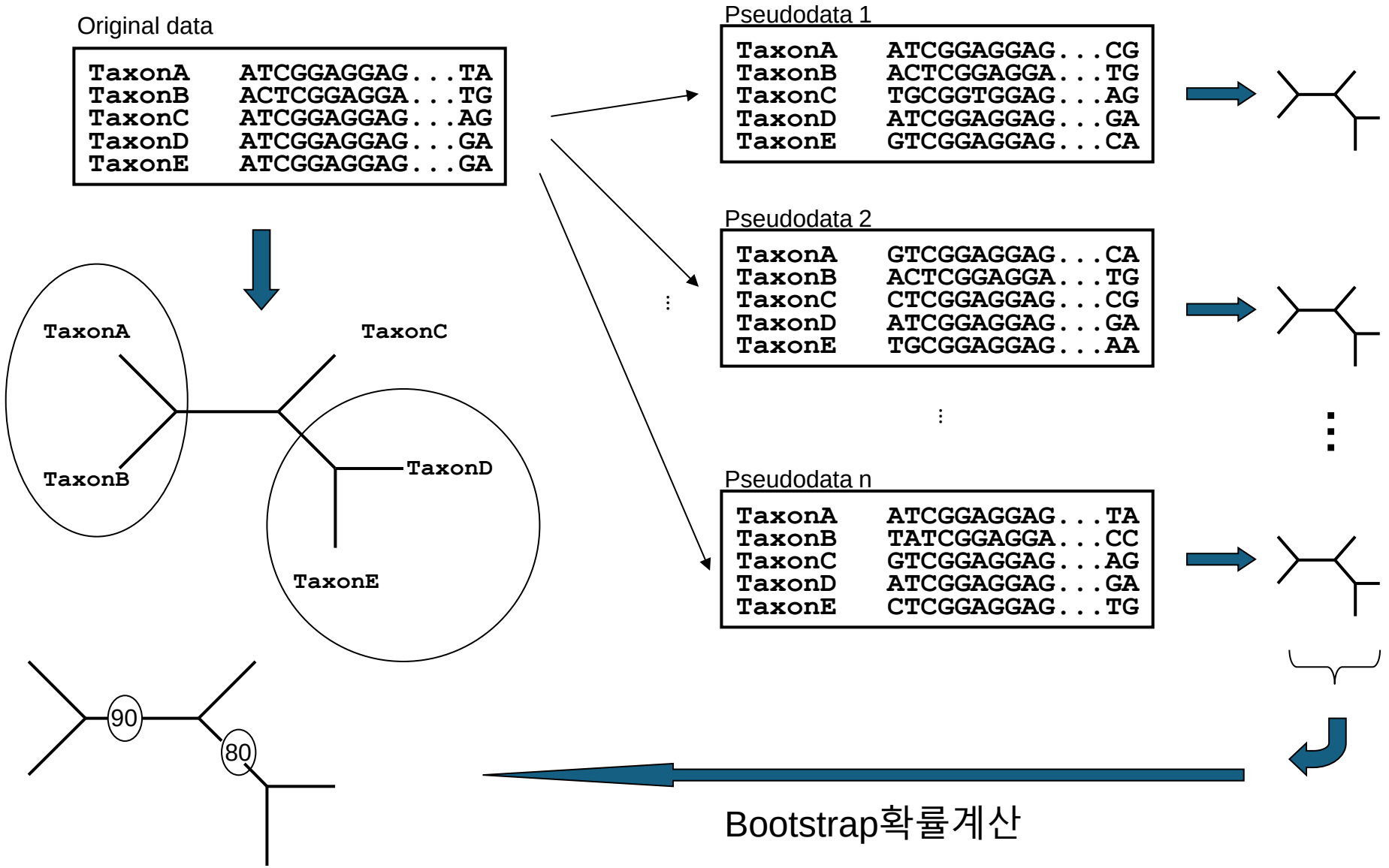


EDN의 기능향상과 관련하여 단백질을 실제로 합성하여 증명

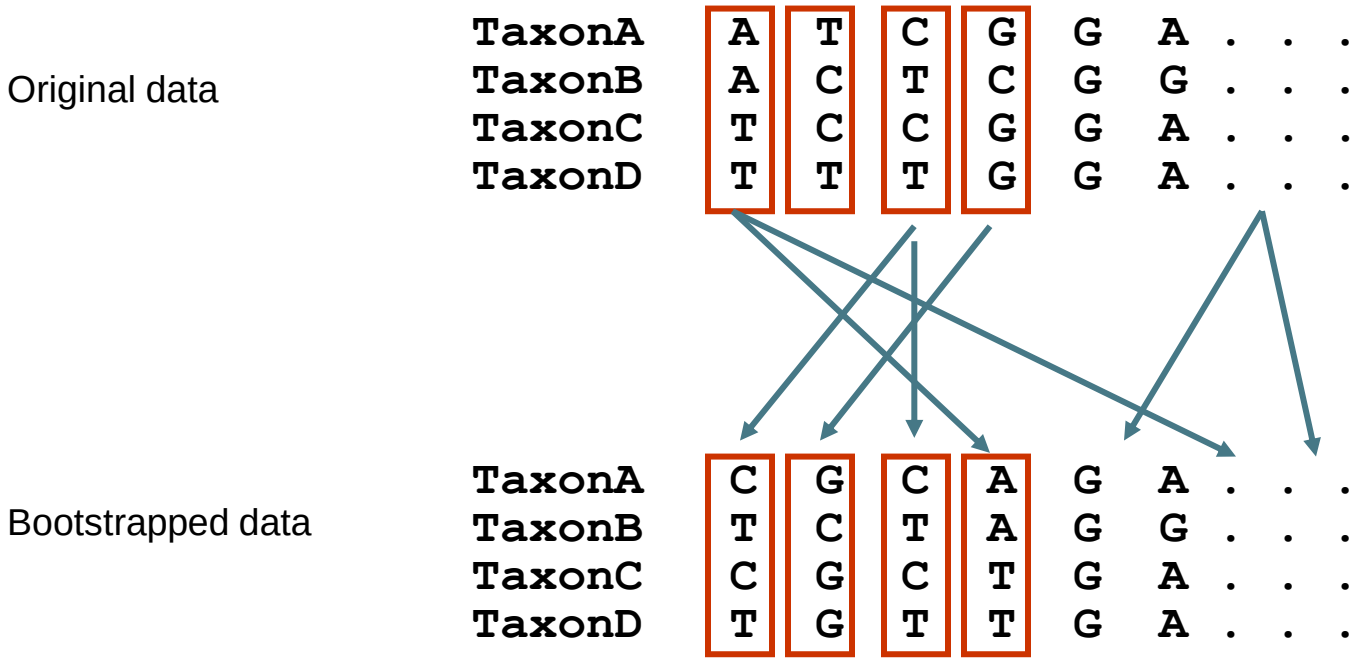
모형	장점	단점
(1) DNA 치환 모형	계산속도가 빠름	비현실적인 (암묵적인) 가정 • 정지코돈의 존재 가정 • 정지코돈의 치환 가정
(2) 아미노산 치환 모형	경험적으로 얻은 치환 정보 반영 동의치환 포화에 영향을 받지 않음	동의치환 정보 손실
(3) 코돈 치환 모형	자연선택 검출 가능	계산속도 느림

표 1. 세 그룹 모형의 장단점

Bootstrap method (계통수의 신뢰도 추정)

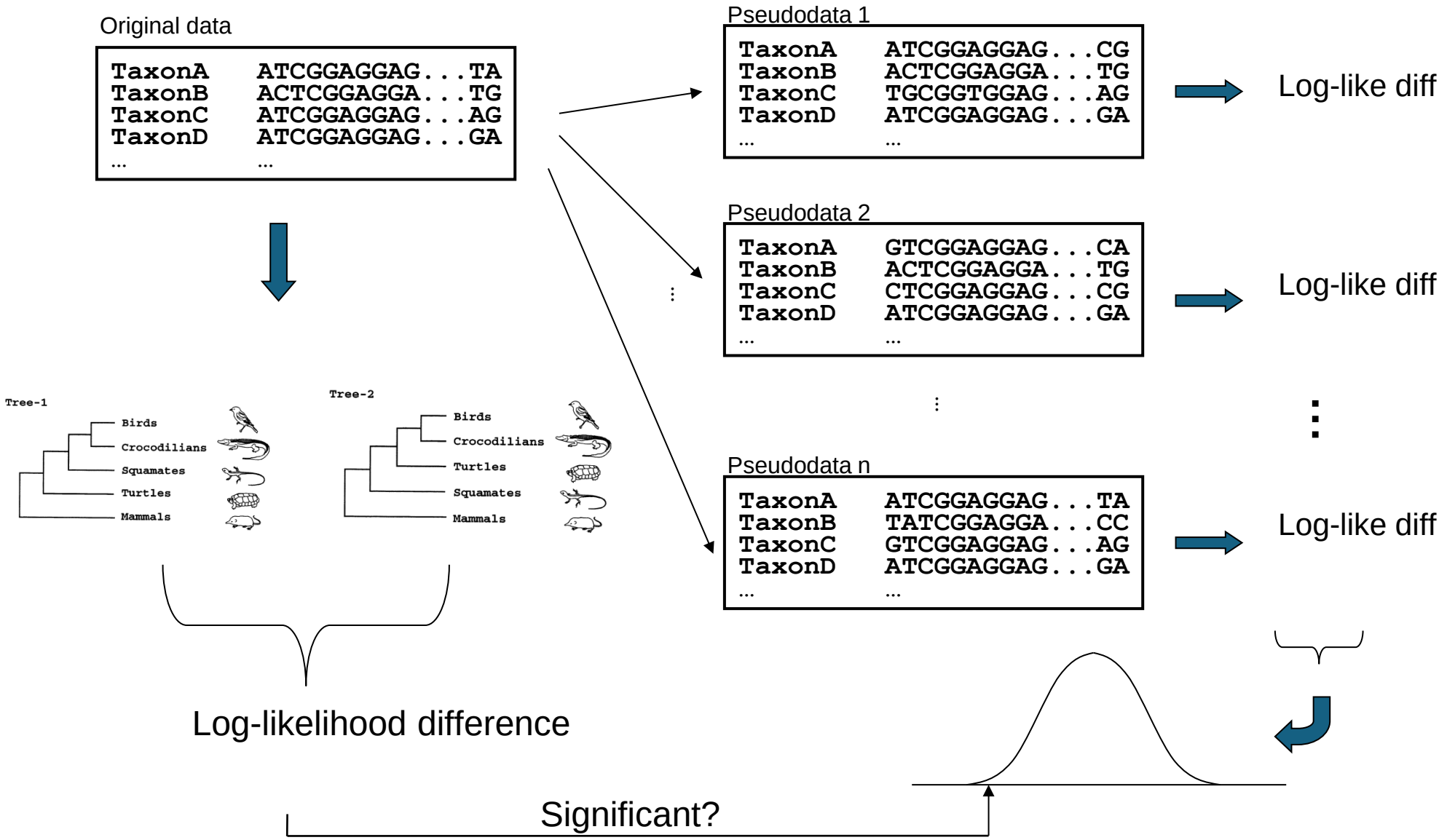


- Generation of pseudo-data



- *i.i.d.* assumption : ***i***ndependently and ***i***dentically ***d***istributed sample
- Each alignment column is resamplied

Bootstrap method (계통수의 신뢰도 추정)



실습

Mx 유전자는 myxovirus에 대해 저항성 (antiviral activity)을 발현하는 유전자이며, 선행연구(Hou et al. 2007)에서 행한 12종(조류 2종, 포유류 10종)의 분석 결과 닭에 이르는 진화과정중에 양의 자연선택이 작용했고 포유류의 진화과정중에는 음의 자연선택이 작용했음이 알려졌다. Álvarez-Carretero et al.(2023)

PAML (Phylogenetic Analysis by Maximum Likelihood; Yang 2007) 은 그 이름에서 유추할 수 있듯이 최대가능도(Maximum Likelihood;ML) 추정법을 이용하여 다양한 분자진화 분석을 할 수 있는 프로그램 패키지이다. 다양한 프로그램을 포함하고 있는데 그 중에서 코돈 모형과 아미노산 모형을 이용하여 분석할 수 있는 프로그램은 codeml이다.[12]

IQ-TREE를 이용한 **ML** 계통수의 추정

코돈 서열 데이터 분석을 위한 codeml 프로그램은 계통수 탐색 기능이 없어 사용자가 계통수를 지정해야 한다. 사전에 주어진 계통관계가 있으면 그것을 사용하면 되겠으나, 여기서는 계통관계가 미정인 일반적인 상황을 상정하여 계통수 추정 단계부터 시작한다.

먼저 DNA 모형을 이용하여 ML 계통수를 추정해보자. 이전 논문(서태건 2022)에서 설명한 IQ-TREE 프로그램[15]을 사용한다. 윈도우의 명령프롬프트를 열어 (찾기 → cmd.exe 입력 후 엔터) 프로그램을 실행할 폴더 (본 분석에서는 임의로 C:\temp로 설정) 로 이동한다. Álvarez-Carretero et al. 논문에서 사용한 Mx_aln.phy 파일과 IQ-TREE 프로그램 실행에 필요한 libiomp5md.dll, iqtree.exe 파일을 같은 폴더에 복사한다.

```
#nexus
begin sets;
    charset part1 = Mx_aln.phy: 1-1989\3 2-1989\3;
    charset part2 = Mx_aln.phy: 3-1989\3;
    charpartition mine = GTR+F+G:part1 , GTR+F+G:part2;
end;
```

그림 3. 파티션 지정 방법 예시. 별도의 파일 partition_info.txt에 파티션 정보를 저장한다.

파티션 설정 파일 partition_info.txt 을 작성한 후에 명령 프롬프트에서 다음과 같이 실행한다. 여기에

```
C:\temp>  iqtree -seed 1 -s Mx_aln.phy -spp partition_info.txt -bb
1000 -redo   ↵
```

그림 4. Mx 염기서열 데이터에 GTR+F+G DNA 모형 적용, ML 계통수를 추정하는 방법.

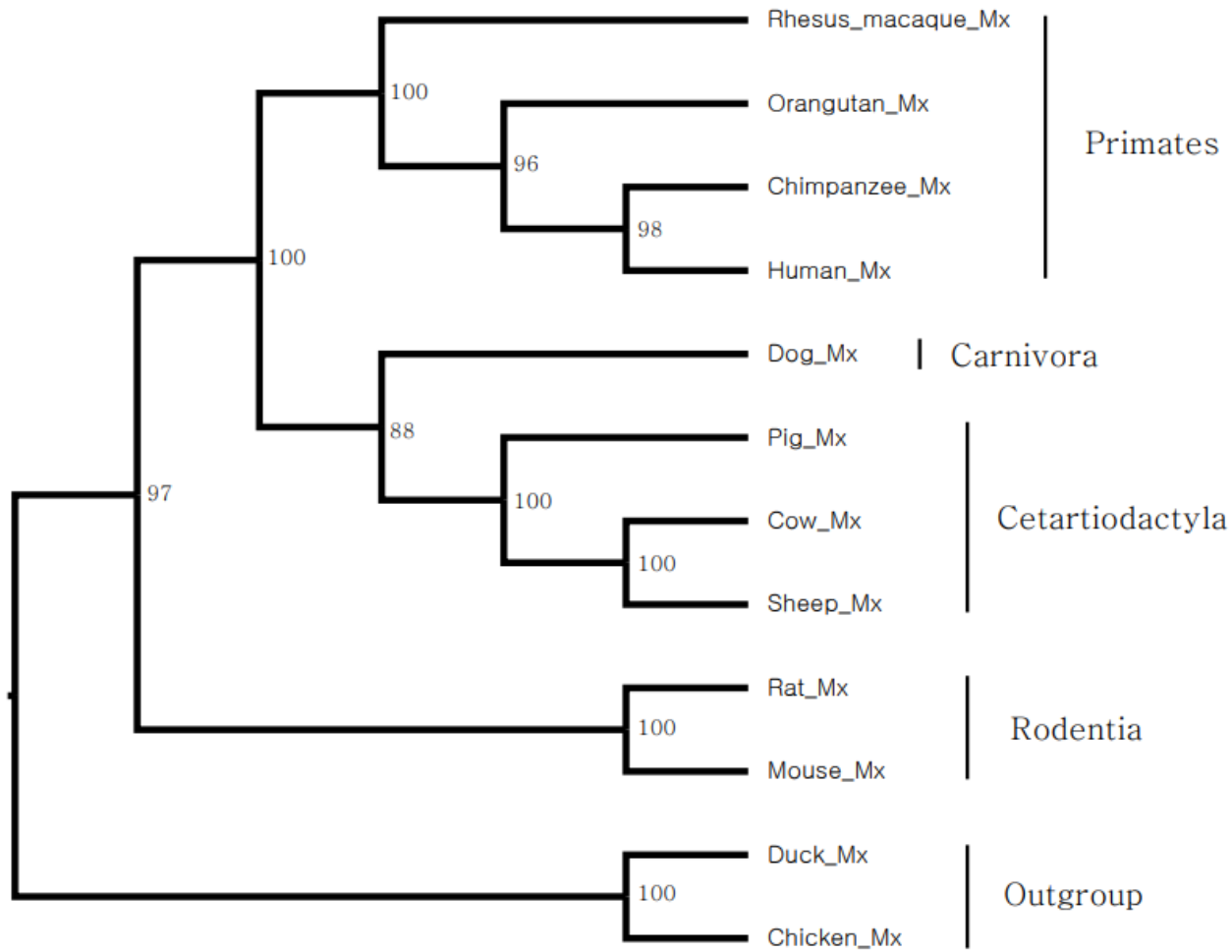


그림 5. partition_info.txt.treefile 파일에 저장된 계통수. 내부노드의 수치는 붓스트랩 확률이다. 계통수 가지의 길이는 무시하고 계통관계만 표현한 cladogram임에 주의하라.

```
  seqfile = Mx_aln.phy              * Path to the alignment file
 treefile = Mx_unrooted_trees.txt   * Path to the tree file
  outfile = out_M0.txt              * Path to the output file

    noisy = 3          * How much rubbish on the screen
  verbose = 1          * More or less detailed report

  seqtype = 1          * 1:codons; 2:AAs; 3:codons-->AAs
CodonFreq = 3          * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table
                       * 4:F1x4MG, 5:F3x4MG, 6:FMutSel0, 7:FMutSel

    icode = 0          * 0:universal code; 1:mammalian mt; 2-10:see below

    model = 0          * 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
                       * 6:FromCodon, 7:AAClasses, 8:REVaa_0, 9:REVaa(nr=189)

fix_omega = 0          * 1: omega or omega_1 fixed, 0: estimate
    omega = 0.5        * initial or fixed omega, for codons or codon-based AAs

fix_alpha = 0          * 0: estimate gamma shape parameter; 1: fix it at alpha
    alpha = 0.5        * initial or fixed alpha, 0:infinity (constant rate)
    ncatG = 5          * # of categories in dG of NSsites models

cleandata = 0          * remove sites with ambiguity data (1:yes, 0:no)?
   method = 0          * Optimization method 0: simultaneous; 1: one branch a time

* Genetic codes: 0:universal, 1:mammalian mt., 2:yeast mt., 3:mold mt.,
* 4: invertebrate mt., 5: ciliate nuclear, 6: echinoderm mt.,
* 7: euplotid mt., 8: alternative yeast nu. 9: ascidian mt.,
* 10: blepharisma nu.
* These codes correspond to transl_table 1 to 11 of GENEBANK.
```

그림 9. M0 모형 분석을 위한 codeml.ctl 파일.

```
12  3
(((((((Chimpanzee_Mx,Human_Mx),Orangutan_Mx),Rhesus_macaque_Mx),(((Sheep_Mx,Cow_M:
(((((((Chimpanzee_Mx,Human_Mx),Orangutan_Mx),Rhesus_macaque_Mx),((Sheep_Mx,Cow_M:
(((((((Chimpanzee_Mx,Human_Mx),Orangutan_Mx),Rhesus_macaque_Mx),(Mouse_Mx,Rat_Mx)
```

그림 10. 비교를 위한 세종류의 계통수. Newick 형식

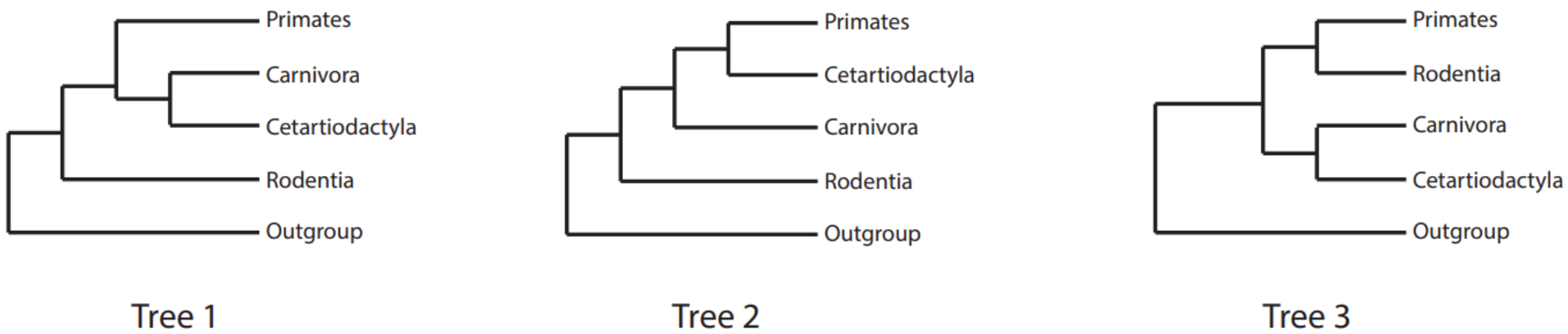


그림 8. 세가지 대표적인 포유류 유전자 계통관계. Tree1은 *Mx* 유전자를 DNA모형, 코돈모형으로 분석하여 얻은 ML 계통수이고 Tree2는 아미노산 모형으로 얻은 ML 계통수이다. Tree3은 Song et al. (2012)이 보고한 종 계통수와 합치하는 계통수이다.

```
C:\temp> codeml codeml.ctl
```

그림 11. codeml.ctl 파일에 모든 설정을 저장하고 위와 같이 입력하면 codeml 프로그램이 실행된다.

```
Tree comparisons (Kishino & Hasegawa 1989; Shimodaira & Hasegawa 1999)
Number of replicates: 10000

 tree            li        Dli      +- SE      pKH       pSH     pRELL

    1*  -12307.766      0.000      0.000   -1.000    -1.000     0.667
    2   -12310.797     -3.031      5.488    0.290     0.434     0.298
    3   -12317.040     -9.274      5.416    0.043     0.136     0.035

pKH: P value for KH normal test (Kishino & Hasegawa 1989)
pRELL: RELL bootstrap proportions (Kishino & Hasegawa 1989)
pSH: P value with multiple-comparison correction (MC in table 1 of Shimodaira & Hasegawa 1999)
(-1 for P values means N/A)
```

그림 12. M0 모형. 세종류 계통수에 대한 SH 테스트

Dli 열에 표시되어 있다. pKH열과 pRELL열의 결과에 대한 고찰은 생략하고 pSH열의 결과에 초점을 맞추자. pSH열은 SH 검정 통계량의 p값을 나타내며 Dli열의 값이 통계적으로 유의한지 판단할 수 있는 근거를 제공한다 (최적계통수 Tree1에 부여된 '-1.000'값은 의미 없으므로 무시해도 좋다). Tree 1과 Tree 2의 스코어 차이는 3.031 ($\approx -12307.766 - (-12310.797)$)이고 SH 검정의 p값은 0.434 (pSH 열)이므로 Tree 1과 Tree2의 차이는 그다지 크지 않다는 것을 알수 있다. 마찬가지로 Tree1과 Tree3의 스코어 차이도 유의하지 않다 (pSH=0.136). 즉, DNA 모형과 코돈 모형으로 추정한 ML 계통수는 Tree1이지만 다른 두 계통수 Tree2와 Tree3도 Tree1에 비하여 그다지 나쁘지 않은 계통수라는 것을 의미한다.

```
TREE #  1:  ((((((3, 4), 2), 1), (((8, 7), 6), 5)), (10, 9)), 12, 11);   MP s
lnL(ntime: 21  np: 24): -12307.766006      +0.000000
  13..14   14..15   15..16   16..17   17..18   18..3    18..4    17..2    16
 2.994459 0.325955 0.345201 0.053578 0.022700 0.015648 0.022608 0.026465 0.0

Note: Branch length is defined as number of nucleotide substitutions per cod

tree length =  8.230340

((((((3: 0.015648, 4: 0.022608): 0.022700, 2: 0.026465): 0.053578, 1: 0.0758

((((((Chimpanzee_Mx: 0.015648, Human_Mx: 0.022608): 0.022700, Orangutan_Mx: 

Detailed output identifying parameters

kappa (ts/tv) =  2.65054

omega (dN/dS) =  0.25425

alpha (gamma, K = 5) =  1.29688
rate:   0.15800  0.43595  0.76320  1.23429  2.40856
freq:   0.20000  0.20000  0.20000  0.20000  0.20000
dN & dS for each branch

branch          t       N       S    dN/dS      dN      dS   N*dN   S*dS

 13..14     2.994  1453.2   535.8   0.2542  0.5576  2.1930  810.2 1175.1
 14..15     0.326  1453.2   535.8   0.2542  0.0607  0.2387   88.2  127.9
 15..16     0.345  1453.2   535.8   0.2542  0.0643  0.2528   93.4  135.5
 16..17     0.054  1453.2   535.8   0.2542  0.0100  0.0392   14.5   21.0
 17..18     0.023  1453.2   535.8   0.2542  0.0042  0.0166    6.1    8.9
 18..3      0.016  1453.2   535.8   0.2542  0.0029  0.0115    4.2    6.1
 18..4      0.023  1453.2   535.8   0.2542  0.0042  0.0166    6.1    8.9
```

그림 13. M0 모형에 의해 추정된 모수.

ω 값이 모두 동일

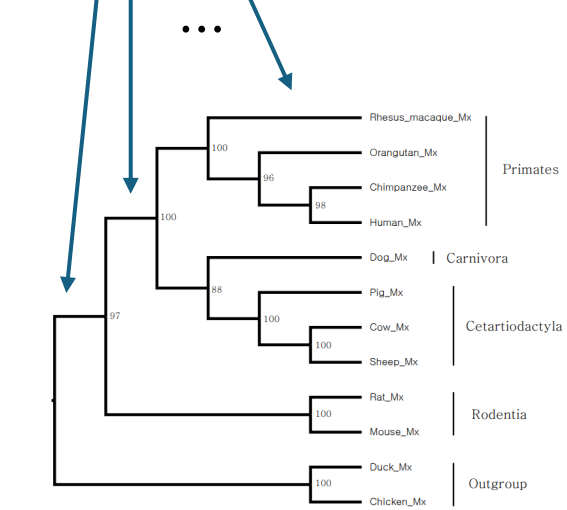


그림 5. partition_info.txt.treefile 파일에 저장된 계통수. 내부노드의 수치는 붓스트랩 확률이다. 계통수 가지의 길이는 무시하고 계통관계만 표현한 cladogram임에 주의하라.

Branch 마다 서로 다른 ω를 상정하고 추정해보자(Branch 모델)

Branch 모형으로 실행하기 위해 model=1 설정한다

```
      seqfile = Mx_aln.phy             * Path to the alignment file
     treefile = Mx_unrooted_trees.txt  * Path to the tree file
      outfile = out_M0.txt             * Path to the output file

        noisy = 3        * How much rubbish on the screen
      verbose = 1        * More or less detailed report

      seqtype = 1        * 1:codons; 2:AAs; 3:codons-->AAs
    CodonFreq = 3        * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table
                         * 4:F1x4MG, 5:F3x4MG, 6:FMutSel0, 7:FMutSel

        icode = 0        * 0:universal code; 1:mammalian mt; 2-10:see below

        model = 0        * 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
                         * 6:FromCodon, 7:AAClasses, 8:REVaa_0, 9:REVaa(nr=189)

    fix_omega = 0        * 1: omega or omega_1 fixed, 0: estimate
        omega = 0.5      * initial or fixed omega, for codons or codon-based AAs

    fix_alpha = 0        * 0: estimate gamma shape parameter; 1: fix it at alpha
        alpha = 0.5      * initial or fixed alpha, 0:infinity (constant rate)
        ncatG = 5        * # of categories in dG of NSsites models

    cleandata = 0        * remove sites with ambiguity data (1:yes, 0:no)?
       method = 0        * Optimization method 0: simultaneous; 1: one branch a time

* Genetic codes: 0:universal, 1:mammalian mt., 2:yeast mt., 3:mold mt.,
* 4: invertebrate mt., 5: ciliate nuclear, 6: echinoderm mt.,
* 7: euplotid mt., 8: alternative yeast nu. 9: ascidian mt.,
* 10: blepharisma nu.
* These codes correspond to transl_table 1 to 11 of GENEBANK.
```

그림 9. M0 모형 분석을 위한 codeml.ctl 파일.

계산시간 단축을 위해 "12 1" 로 변경

```
12  3
((((((Chimpanzee_Mx,Human_Mx),Orangutan_Mx),Rhesus_macaque_Mx),(((Sheep_Mx,Cow_M:
(((((((Chimpanzee_Mx,Human_Mx),Orangutan_Mx),Rhesus_macaque_Mx),((Sheep_Mx,Cow_M:
((((((Chimpanzee_Mx,Human_Mx),Orangutan_Mx),Rhesus_macaque_Mx),(Mouse_Mx,Rat_Mx)
```

그림 10. 비교를 위한 세종류의 계통수. Newick 형식

Branch 모형(model=1) 결과

```
TREE #  1:  ((((((3, 4), 2), 1), (((8, 7), 6), 5)), (10, 9)), 12, 11);   MP
lnL(ntime: 21  np: 43): -12525.800361      +0.000000
  13..14   14..15   15..16   16..17   17..18   18..3    18..4    17..2    16
 2.968843 0.223108 0.300390 0.048883 0.020762 0.013864 0.020561 0.023494 0.0

Note: Branch length is defined as number of nucleotide substitutions per cod

tree length =  6.900743

((((((3: 0.013864, 4: 0.020561): 0.020762, 2: 0.023494): 0.048883, 1: 0.0640

((((((Chimpanzee_Mx: 0.013864, Human_Mx: 0.020561): 0.020762, Orangutan_Mx:

Detailed output identifying parameters

kappa (ts/tv) =  2.37994

w (dN/dS) for branches:  0.10773 0.37561 0.16725 0.17723 0.12587 0.43194 0.1

dN & dS for each branch

branch          t       N       S    dN/dS      dN      dS   N*dN   S*dS

 13..14     2.969  1467.4   521.6   0.1077  0.3120  2.8962  457.8 1510.5
 14..15     0.223  1467.4   521.6   0.3756  0.0518  0.1379   76.0   71.9
 15..16     0.300  1467.4   521.6   0.1673  0.0434  0.2597   63.7  135.4
 16..17     0.049  1467.4   521.6   0.1772  0.0073  0.0415   10.8   21.6
 17..18     0.021  1467.4   521.6   0.1259  0.0025  0.0195    3.6   10.2
 18..3      0.014  1467.4   521.6   0.4319  0.0034  0.0080    5.0    4.1
 18..4      0.021  1467.4   521.6   0.1011  0.0021  0.0203    3.0   10.6
```

그림 14. Branch 모형에 의해 추정된 모수.

ω 값이 모두 다름

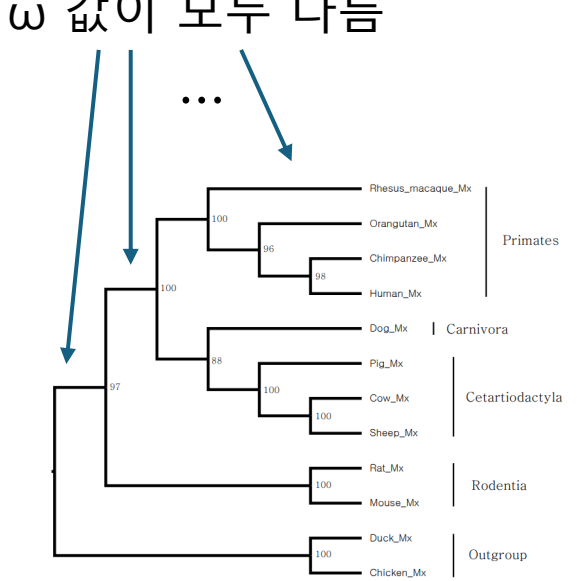


그림 5. partition_info.txt.treefile 파일에 저장된 계통수. 내부노드의 수치는 붓스트랩 확률이다. 계통수 가지의 길이는 무시하고 계통관계만 표현한 cladogram임에 주의하라.

다음으로 염기서열을 아미노산으로 번역한 후에 아미노산 모형으로 분석을 해보자.

```
      seqfile = Mx_aln.phy              * Path to the alignment file
     treefile = Mx_unrooted_trees.txt   * Path to the tree file
      outfile = out_AA_lg.txt             * Path to the output file

        noisy = 3          * How much rubbish on the screen
      verbose = 1          * More or less detailed report

      seqtype = 3          * 1:codons; 2:AAs; 3:codons-->AAs
    CodonFreq = 3          * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table
                           * 4:F1x4MG, 5:F3x4MG, 6:FMutSel0, 7:FMutSel

        icode = 0          * 0:universal code; 1:mammalian mt; 2-10:see below

        model = 3          * 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
                           * 6:FromCodon, 7:AAClasses, 8:REVaa_0, 9:REVaa(nr=189)
      NSsites = 0          * 0:one w;1:neutral;2:selection; 3:discrete;4:freqs;
                           * 5:gamma;6:2gamma;7:beta;8:beta&w;9:beta&gamma;
                           * 10:beta&gamma+1; 11:beta&normal>1; 12:0&2normal>1;
                           * 13:3normal>0

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

    fix_omega = 0          * 1: omega or omega_1 fixed, 0: estimate
        omega = 0.5        * initial or fixed omega, for codons or codon-based AAs

    fix_alpha = 0          * 0: estimate gamma shape parameter; 1: fix it at alpha
        alpha = 0.5        * initial or fixed alpha, 0:infinity (constant rate)
        ncatG = 5          * # of categories in dG of NSsites models

    cleandata = 0          * remove sites with ambiguity data (1:yes, 0:no)?
       method = 0          * Optimization method 0: simultaneous; 1: one branch a time

* Genetic codes: 0:universal, 1:mammalian mt., 2:yeast mt., 3:mold mt.,
* 4: invertebrate mt., 5: ciliate nuclear, 6: echinoderm mt.,
* 7: euplotid mt.   8: alternative yeast nu.  9: ascidian mt.
```

그림 15. codeml.ctl 파일. 아미노산 치환 모형. LG.

세 계통수의 비교를 위해
"12 3" 으로 다시 설정

```
12  3
((((((Chimpanzee_Mx,Human_Mx),Orangutan_Mx),Rhesus_
(((((((Chimpanzee_Mx,Human_Mx),Orangutan_Mx),Rhesu
((((((Chimpanzee_Mx,Human_Mx),Orangutan_Mx),Rhesus_
```

그림 10. 비교를 위한 세종류의 계통수

```
Tree comparisons (Kishino & Hasegawa 1989; Shimodaira & Hasegawa 1999)
Number of replicates: 10000

 tree           li       Dli     +- SE      pKH      pSH    pRELL

    1   -7472.960    -1.924     3.360    0.283    0.541    0.270
    2*  -7471.036     0.000     0.000   -1.000   -1.000    0.690
    3   -7480.352    -9.316     6.287    0.069    0.088    0.040

pKH: P value for KH normal test (Kishino & Hasegawa 1989)
pRELL: RELL bootstrap proportions (Kishino & Hasegawa 1989)
pSH: P value with multiple-comparison correction (MC in table 1 of Shimodaira & Hasegawa 1999)
(-1 for P values means N/A)
```

그림 16. LG 모형. 세종류 계통수에 대한 SH 테스트

Tree1의 스코어 차이는 1.924 ($\approx -7471.036 - (-7472.960)$)정도이고 pSH열의 결과에서 알 수 있듯이 SH 테스트의 p값(pSH)은 0.541이며 그다지 작지 않다. Tree2와 Tree3의 차이의 p값도 0.088이라 그다지 작지 않다. 즉, ML방법으로 LG 아미노산 모형을 적용시켜 얻은 ML 계통수는 Tree2이지만 Tree1, Tree3도 LG 아미노산 모형 하에서 그다지 열등하지 않은 계통수라는 것을 의미한다.

자연선택 검출을 보다 정확하게 할 수 있는 Site 모형(ω는 사이트별로 다르며 $\omega > 1$인 사이트를 특정 가능함), Branch-site 모형(ω는 사이트와 계통수 가지별로 다르며 $\omega > 1$인 사이트와 계통수 가지를 특정 가능함)등을 제반 사정을 고려하여 본 논문에서 다루지 못한 것이 무척 아쉽다. Álvarez-Carretero et al(2023)에는 Site 모형, Branch-site 모형을 이용한 프로그램 실행 방법이 상세히 설명되어 있으나 이해도를 높이기 위해서는 원저 논문(Nielsen and Yang 1998; Yang and Nielsen 2002)을 병행해서 볼 필요가 있다.