아미노산 서열과 코돈 서열의 진화 모형

서태건[1,*]

요약**:** 단백질을 코딩하는 유전자에서는 DNA 염기 세 개가 코돈이라는 기본 단위를 이루며 하나의 아미노산을 지정한다. 따라서 단백질 코딩 유전자의 분자진화학 연구에 있어서, (1) 코돈이라는 기본단위를 무시하고 단순히 DNA 염기 치환에 초점을 둔 DNA 치환모형, (2) 코돈으로부터 번역된 아미노산의 변화에 초점을 둔 아미노산 치환모형, (3) 세개의 염기세트의 변화에 초점을 둔 코돈 치환모형, 이렇게 크게 나누어 세가지 카테고리의 모형을 생각할 수 있다. 이전 논문(서태건 2022)에서 DNA 치환모형과 정량적 비교에 대해 살펴본 바 있다. 본 논문에서는 이전 논문에 이어서 아미노산 모형과 코돈 모형에 대해 설명한다. 또한, 세 종류의 모형을 이용한 분자계통수의 추정과 비교, 분자수준에서 작용한 자연선택의 추정에 대해 논의하며 분석의 예시를 IQ-TREE, PAML 프로그램을 이용하여 보여준다.

키워드**:** DNA 모형, 아미노산 모형, 코돈 모형, Shimodaira-Hasegawa 검정, 자연선택, ω, d_N/d_S

[1] *인천광역시 연수구 송도동 송도미래로26 극지연구소*

* **Corresponding author**: seo.taekun@gmail.com

서론

이전 논문(서태건 2022)에서 설명한 바와 같이 DNA 염기치환 모형은 A, C, G, T, 네 종류 염기 사이의 치환율을 규정하고 이를 4×4 치환율 행렬로 표현한다. 아래는 Tamura and Nei (1993) 모델과 GTR (General Time Reversible model; Tavaré 1986) 모형의 예이다.

$$\mathbf{R}^{(TN)} = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{bmatrix} - & \pi_C & \kappa_1\pi_G & \pi_T \\ \pi_A & - & \pi_G & \kappa_2\pi_T \\ \kappa_1\pi_A & \pi_C & - & \pi_T \\ \pi_A & \kappa_2\pi_C & \pi_G & - \end{bmatrix} \end{array}, \quad \mathbf{R}^{(GTR)} = \begin{array}{c} \\ A \\ C \\ G \\ T \end{array} \begin{array}{c} \begin{array}{cccc} A & C & G & T \end{array} \\ \begin{bmatrix} - & a\pi_C & b\pi_G & c\pi_T \\ a\pi_A & - & d\pi_G & e\pi_T \\ b\pi_A & d\pi_C & - & \pi_T \\ c\pi_A & e\pi_C & \pi_G & - \end{bmatrix} \end{array} \tag{1}$$

여기서 네 종류의 π는 각 염기의 빈도를 나타내는 모수이고, κ_1, κ_2는 두 종류의 transition에 다른 치환률을, a,b,c,d,e는 두 종류의 transition과 세 종류의 transversion에 각각 다른 치환률을 할당하는 모수이다. 이러한 모형을 이용하여 가능도(likelihood)[1]를 계산할 수 있고, 가능도를 이용하여 AIC (Akaike Information Criterion; Akaike 1974), BIC (Bayesian Information Criterion; Schwarz 1978)등을 계산하여 모형을 정량적으로 비교할 수 있음을 지난 논문에서 설명했다.[2]

단백질을 코딩하는 유전자에서는 DNA 염기 세개가 하나의 코돈(codon)을 이루고 코돈은 번역(trans-

[1] 본 논문에 등장하는 영문 통계학 용어는 한국통계학회 홈페이지와 김우철(2021)을 참고하여 번역하였다. http://www.kss.or.kr/bbs/board.php?bo_table=psd_sec

[2] 본 논문을 읽는 독자는 이전 논문 (서태건 2022)의 내용을 숙지했다고 가정한다. 따라서 AIC, BIC 같은 정보량기준이나 감마분포를 이용한 사이트간 진화속도 이질성 모형등, 기본적인 사항은 자세한 설명없이 기술하도록 하겠다.

lation)과정에서 하나의 아미노산을 지정한다. 코돈과 아미노산과의 관계를 규정하는 코돈 테이블은 생물에 따라 매우 다양한 양상을 보여 염기 서열 데이터뿐만 아니라 코돈 테이블도 다양하게 진화함을 알 수 있다.[3] 표준 코돈 테이블(standard codon table)의 경우 세개의 정지코돈과 61개의 센스코돈 (sense codon; 아미노산을 지정하는 코돈)이 있는데 61개의 센스코돈이 20개의 아미노산을 지정하므로 일부 복수의 코돈이 동일 아미노산을 지정하게 된다. 동일 아미노산을 지정하는 코돈사이의 치환을 동의치환 (synonymous substitution; 同義置換), 서로 다른 아미노산을 지정하는 코돈 사이의 치환을 비동의치환 (nonsynonymous substitution; 非同義置換)이라 한다.[4] 동의치환을 무시하고 비동의치환만을 고려하는 모형이 아미노산치환 모형이고, 동의치환과 비동의치환 모두를 고려하는 모형이 코돈치환 모형이다. 비동의 치환이 동의치환에 비해 얼마만큼 빠르게 일어나는가를 코돈 모형은 ω (혹은 d_N/d_S) 모수를 이용하여 모형화하고 이를 이용하여 분자수준에 작용한 자연선택의 세기를 측정한다 (자세한 내용은 Felsenstein 2004, Yang 2006을 참조하라).

DNA 모형에 대해 설명했던 지난 논문에 이어 본 논문에서는 아미노산치환 모형, 코돈치환 모형을 설명하고, 예제파일을 통해 ML 계통수 추정, 계통수들의 정량적인 비교, 자연선택의 세기를 측정하는 분석 사례를 간략하게 기술한다.

본론

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]

초기에 만들어진 Dayhoff et al.(1978) 모형은 아미노산 서열 데이터의 얼라인을 위한 스코어 계산에 흔히 사용되었다. 70여개 그룹의 단백질 서열 데이터로부터 유사도가 85 % 이상인 서열을 선별한 후 이들간의 비교로 얻은 1500여개의 아미노산 치환 양상을 이용하여 모형을 구성하였다 (Mount 2004). 이

[3] 2023년 11월 현재 https://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi 에는 26종의 코돈 테이블이 등재되어 있다.

[4] 참고로 synonymous/nonsynonymous substitution을 같은 한자 문화권인 일본에서는 同義/非同義 置換, 중국에서는 同義/非同義 替換(혹은 置換) 으로 번역하고 있다.

[5] $s_{a_i a_j}$를 흔히 exchangeability coefficient라고 부른다. 염기서열 데이터의 규모가 크면 $s_{a_i a_j}$를 직접 추정할 수도 있지만 일반적으로는 고정된 값을 사용한다.

[6] 시간가역성은 어디까지나 계산상의 편의를 위한 것일뿐, 생물학적으로 그럴듯해서 채택된 가정은 아니다. 시간 가역성을 가정하지 않으면 엄청나게 많은 계산 시간이 요구된다.

후 데이터의 규모가 증가하며 보다 신뢰성이 높은 아미노산 치환 모형이 개발되었고 (Jones et al. 1992; Whelan and Goldman 2001;Le and Gascuel 2008), 미토콘드리아나 엽록체등 세포 소기관 특이적인 아미노산 치환 모형도 개발되었다(Adachi et al. 1996, 2000). 이들은 공통적으로 식 (2)의 형태로 정의되며 모형에 따라 구제적인 $s_{a_i a_j}$ 값들이 달라지는 형태를 갖게 된다.

그림 1은 PAML 프로그램(Yang 2007)에 포함된 lg.dat 파일의 내용으로 아미노산 모형중에 비교적 최근에 개발된 LG 모형(Le and Gascuel 2008)의 $s_{a_i a_j}$ 값들을 나타낸 것이다. 20×20 행렬중 좌측하단 부분의 일부를 표시한 것으로 $s_{a_i a_j} = s_{a_j a_i}$ 이므로 우측상단도 대칭적으로 같은 값들이 위치하게 된다. 하삼각 행렬 형태의 값들이 표시된 후에 LG 모형이 디폴트로 가정하는 아미노산 빈도가 나열되어 있다. 이 빈도들은 대량의 아미노산 서열 데이터로부터 $s_{a_i a_j}$ 값들과 함께 얻어진 값이다. 아미노산 모형이 디폴트로 제시하는 아미노산 빈도는 분석 데이터의 빈도와는 상당한 차이가 있어 분석 데이터의 빈도를 사용하는 것이 보다 현실적이고 성능 좋은 모형이 되는 경우가 많다. 이처럼 주어진 데이터에서 직접 얻은 빈도를 사용할 때 '+F' 태그를 모형이름에 추가한다(예: LG+F).

```
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.

대표적인 아미노산 치환모형 네 종류를 임의로 선정하고 $s_{a_i a_j}$ 값들을 시각화 하여 그림 2에 표시하였다.[7] 모형간에 다소 변이가 보이지만, 전체적인 치환양상은 상당히 유사함을 알 수 있다. 물리,화학적인 성질이 비슷한 아미노산끼리는 쉽게 치환될수 있고 이러한 성질은 DNA모형에서 퓨린, 피리미딘끼리의 구조적 유사성이 transition 치환을 활발하게 하는 상황과 유사하다고 할 수 있다.[8]

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

[7] 단위시간에 사이트당 아미노산 치환이 1회가 되도록 $s_{a_i a_j}$ 값들을 표준화 하였다.

[8] $s_{a_i a_j}$ 값들을 유도하는데 사용된 데이터들이 중복되었다는 점도 유사한 패턴의 한가지 원인이라 생각할 수 있다.

[9] 정지코돈의 갯수는 코돈 테이블의 종류에 따라 다르다. 예컨데, 포유동물의 미토콘드리아의 경우 4개의 정지코돈을 가지고 있다. 본 논문에서는 편의상 표준 코돈을 상정하고 설명하지만, 기술하는 내용은 다른 코돈 테이블에도 동일하게 적용된다.

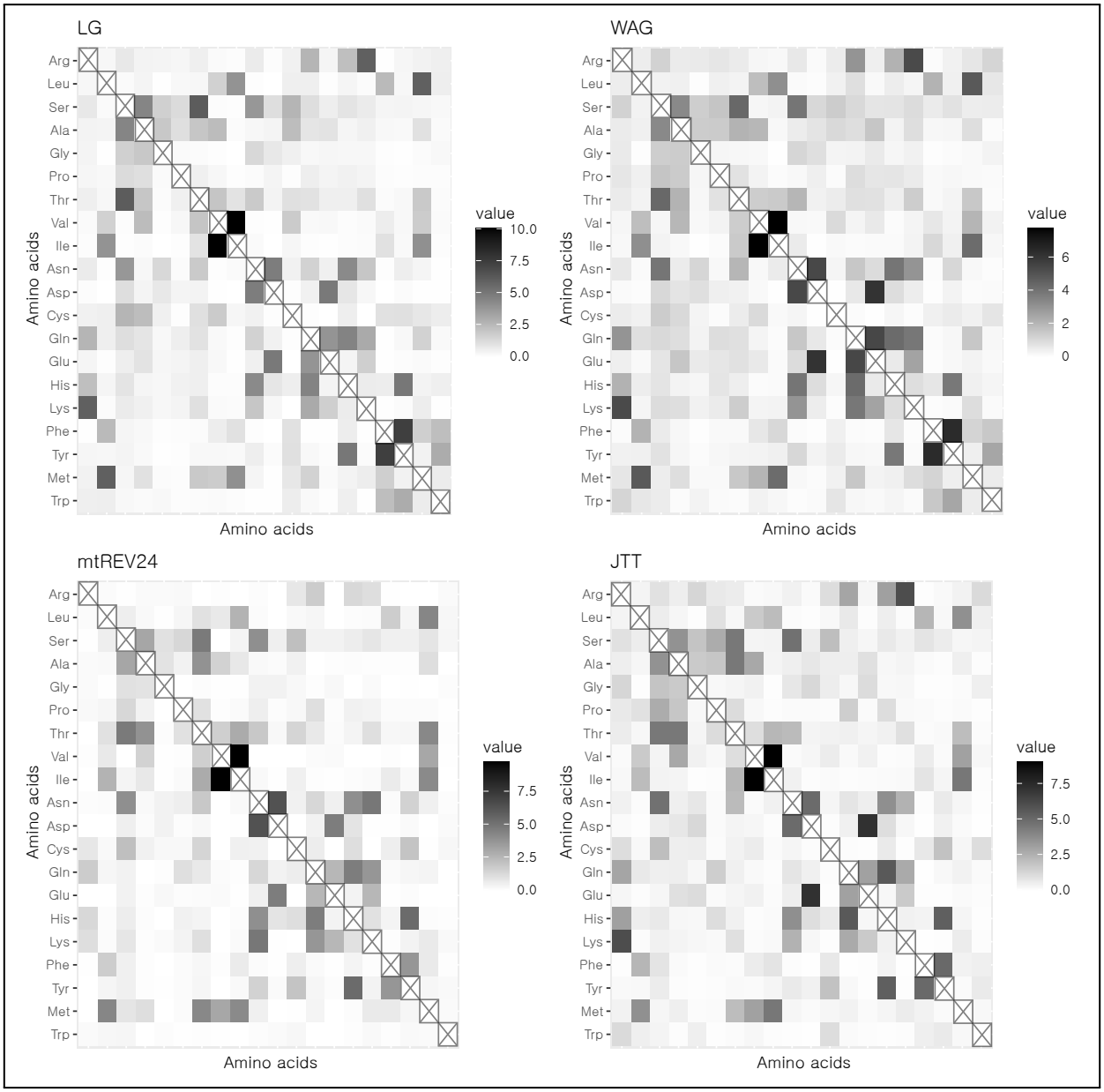

그림 2. 아미노산 모형의 $s_{a_i a_j}$ 값들. $s_{a_i a_j}$ 값이 크면 해당 아미노산 쌍은 치환이 빈번하게 일어남을 의미한다. 임의로 LG(Le and Gascuel 2008), WAG(Whelan and Goldman 2001), mtREV24(Adachi et al. 1996), JTT(Jones et al. 1992) 네 종류 아미노산 치환 모형을 선택했다. 가로축 아미노산 순서 (좌에서 우)는 세로축 아미노산 순서(위에서 아래)와 같다.

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 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.

식 (3)의 이해를 돕기 위해 61×61 행렬 중 극히 일부분, 아르기닌을 코딩하는 여섯개의 코돈 사이의 치환율을 아래와 같이 나타내었다. 예를 들어 CGT로부터 CGC로의 치환은 세번째 T가 C로 바뀌는 치환이고 이는 transition 이므로 치환율은 CGC의 빈도 π_{CGC}와 κ의 곱의 형태로 나타내어진다. CGT로부터 AGA로의 치환은 두개의 염기치환을 수반하므로 치환율은 0이된다. 또한, 모두 동의치환이므로 ω 모수는

[10] 엄밀히 말하면 동의코돈끼리도 선호도의 차이가 있기 때문에(codon bias) 완벽하게 자연선택의 영향이 없다고는 할 수 없으나(예를 들어, Plotkin and Kudla 2011) 비동의치환에 비하면 그 영향은 미미하다고 가정할 수 있다.

포함되지 않는다.

$$
\begin{array}{c} \\ CGT \\ CGC \\ CGA \\ CGG \\ AGA \\ AGG \end{array}
\begin{array}{c}
\begin{array}{cccccc} CGT & CGC & CGA & CGG & AGA & AGG \end{array} \\
\left[\begin{array}{cccccc}
- & \kappa\pi_{CGC} & \pi_{CGA} & \pi_{CGG} & 0 & 0 \\
\kappa\pi_{CGT} & - & \pi_{CGA} & \pi_{CGG} & 0 & 0 \\
\pi_{CGT} & \pi_{CGC} & - & \kappa\pi_{CGG} & \pi_{AGA} & 0 \\
\pi_{CGT} & \pi_{CGC} & \kappa\pi_{CGA} & - & 0 & \pi_{AGG} \\
0 & 0 & \pi_{CGA} & 0 & - & \kappa\pi_{AGG} \\
0 & 0 & 0 & \pi_{CGG} & \kappa\pi_{AGA} & -
\end{array}\right]
\end{array}, \tag{4}
$$

세 그룹 모형의 비교와 장단점

인트론이나 rDNA 같이 단백질을 코딩하지 않는 염기서열은 DNA 치환 모형밖에 선택의 여지가 없지만, 단백질 코딩 유전자의 경우에는 세 그룹(DNA, 아미노산, 코돈 모형)의 모형이 적용 가능해 모형 비교의 문제가 자연스럽게 등장한다. 이전 논문 (서태건 2022)에서 AIC 혹은 BIC등의 정보량 기준을 이용하여 다양한 DNA 모형을 비교하는 것을 설명하였다. 같은 방법으로 여러가지 아미노산 모형, 혹은 코돈 모형들도 로그가능도, 모수의 수, 데이터의 갯수(얼라인된 서열데이터의 열의 갯수)등을 이용하여 AIC, BIC를 정의할 수 있고 이를 통해 모형 비교가 가능하다. 이처럼 DNA 모형그룹내에서, 혹은 아미노산 모형 그룹내에서, 혹은 코돈 모형 그룹내에서의 모형 비교는 이론의 적용이 자명하여 비교적 쉽게 할 수 있다. 하지만, 서로 다른 그룹의 모형 비교는 그리 단순하지 않다. 세 그룹은 근본적으로 데이터의 구조가 다르다. 세 그룹의 모형은 각각 4, 20, 61개의 염기, 아미노산, 코돈간의 치환을 정의하고 있어 치환율 행렬의 차원도 다르며, 특히 아미노산 모형은 코돈을 아미노산으로 변환하는 작업이 수반되어 이를 가능도로 정량화하지 않으면 AIC, BIC등의 비교는 가능하지 않다. Seo and Kishino (2008, 2009)는 4-state DNA 모형, 20-state 아미노산 모형, 61-stae 코돈 모형이 각각 적절한 변환을 거치면 64-state 모형으로 간주될 수 있고 이런 변환을 통하여 계산된 AIC, BIC가 동일 선상에서 비교 가능하다는 것을 보였다.

DNA 모형은 네 종류 염기 사이의 치환을 고려하기 때문에 데이터의 차원이 낮아 계산을 빨리 할 수 있고, 따라서 대량의 데이터 분석도 실행할 수 있다는 장점이 있다. 하지만, DNA 모형은 암묵적으로 두가지 비현실적인 가정을 한다.[11] 첫째, 정지코돈의 빈도가 0이 아니라고 가정한다. 둘째 정지코돈과 센스코돈 사이에, 그리고 정지코돈끼리, 치환이 발생할 수 있다고 가정한다. 만약 DNA 모형의 가정에 따라서 코돈의 세 사이트에서 랜덤하게 독립적으로 염기 치환이 일어난다면 어느 순간에는 정지코돈이 등장하기도 하고, 정지코돈↔센스코돈의 치환이 발생하기도 할 것이다. 하지만 이런 현상은 실제 진화과정에서 발생하기 어렵다. 이처럼 두 가지 암묵적이고 비현실적인 가정이 DNA 모형의 성능을 저해하는 요인으로 작용한다(Seo and Kishino 2009).

아미노산 치환 모형은 대량의 데이터로부터 경험적으로 얻은 아미노산 치환 정보($s_{a_i a_j}$값들)를 반영한다는 장점이 있다. 또한, 비동의치환만 고려하기 때문에 동의치환의 포화(saturation)에 영향을 받지 않는

[11] 명시적으로 가정하지는 않지만 모형의 수학적인 구조가 암묵적으로 이를 가정하는 모형이 된다(Seo and Kishino 2009).

다. 진화적 거리가 먼 경우 동의치환은 매우 많이 일어나 포화상태에 가까울 것이므로(Maynard Smith and Smith 1996) 동의치환은 오히려 노이즈로 작용할 가능성이 있다. 이러한 이유로 진화적으로 먼 유전자를 분석할때 아미노산 모형이 자주 사용되곤 한다. 하지만 진화적 거리가 멀지 않을 경우에는 동의치환도 중요한 정보를 가지고 있어 이를 일률적으로 무시하고 비동의치환만 고려하는 것은 상당한 양의 정보손실을 가져 올 수 있다(Seo and Kishino 2008).

코돈 모형은 염기치환을 동의/비동의 치환으로 구분하고 둘 사이의 상대적인 발생율을 정량적으로 비교하여 분자수준에서 작용한 자연선택의 세기를 측정할 수 있다는 점에서 DNA 모형이나 아미노산 모형보다 우월하다. 또한, 많은 경우 모형의 데이터 적합도도 우수하고 아미노산 치환 정보를 반영하면 성능이 더욱 향상됨이 알려졌다(Seo and Kishino 2008, 2009). 하지만, 61개의 코돈사이의 치환을 고려하기 때문에 데이터의 차원이 압도적으로 증가하여 계산시간이 많이 걸린다는 단점이 있다.

세 그룹의 모형의 장단점을 표 1에 정리하였다. 서열데이터의 규모, 계산을 위한 리소스의 한계, 데이터 분석의 목적등, 상황을 종합적으로 고려하여 모형을 비교/선택해야 할 것이다.

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

데이터 분석의 예

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]

Codeml 프로그램을 이용한 데이터 분석 과정을 설명해 주는 좋은 리뷰논문이 최근 발표되었다 (Álvarez-Carretero et al. 2023). Álvarez-Carretero et al.은 선행연구 (Hou et al. 2007)에서 분석한 유전자를 예시로 하여 자연선택의 세기와 자연선택이 작용한 코돈사이트, 자연선택이 적용된 계통수 상의 위치등을 codeml 프로그램을 이용하여 추정하는 방법을 설명하고 있다. 본 논문에서는 Álvarez-Carretero et al.이 생성한 염기서열 데이터를 이용하여 DNA 모형, 아미노산 모형, 코돈 모형으로 분자계통수를 추정하는 방법을 간략히 설명하고, 또한 간단한 코돈 모형을 이용해 ω 모수를 추정하는 분석예를 보이겠다. Álvarez-Carretero

[12]DNA 치환 모형으로 ML 추정을 하는 baseml, 분기연대를 추정하는 mcmctree, 서열데이터의 시뮬레이션을 수행하는 evolver 등 다양한 프로그램이 패키지에 포함되어 있다. http://abacus.gene.ucl.ac.uk/software/paml.html

et al.이 그들의 논문에서 보여준 자연선택 평가를 위한 다양한 분석 (예컨데, GY94모형 하부의 M1a, M2a, M7, M8 서브 모형등)은 내용도 방대하고 추가로 설명해야 할 사항도 많아 본 논문에서는 생략한다. [13]

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)는 Hou et al.의 논문에 기술된 절차를 따라 염기서열을 데이터 베이스로부터 입수, 정렬하여 12종 1989 염기의 데이터 세트를 결정했다. 본 논문에서는 Álvarez-Carretero et al.(2023)이 생산한 얼라인 데이터를 이용해 분석을 진행한다. [14]

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.

일반적으로 단백질 코딩 유전자는 코돈의 세번째 사이트가 동의치환인 경우가 많아 진화속도가 매우 빠르다. 따라서 코돈의 세번째 사이트를 별도의 파티션으로 간주하고 분석하는 것이 일반적이다. 그림 3의 내용을 파일명 partition_info.txt에 텍스트 모드로 저장하자. 이는 Mx_aln.phy 에 저장된 염기서열 데이터의 파티션 정보를 설정한다. '# nexus'는 설정 양식이 넥서스[16] 양식임을 알리는 키워드이다.[17]

```
#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.

'begin sets;'와 'end;' 키워드 사이에 파티션 설정을 입력한다. 'charset'와 'charpartititon' 키워드를 이용하여 파티션을 정의한다. '1-1989\3'은 사이트 $1,4,7,\cdots,3k+1,\cdots,1987$를 의미하고 '2-1989\3'은 사이트

[13] Álvarez-Carretero et al.이 자신들의 논문에서 공개한 PAML 튜토리알 https://github.com/abacus-gene/paml-tutorial 에 코돈 모형을 이용한 다양한 분석이 소개되어 있다. 본 논문에서 다루는 M0 모형을 이용한 분석은 Álvarez-Carretero et al.의 분석과 세부 옵션 설정 등에서 약간의 차이가 있으나 결과에 크게 영향은 없다.

[14] 데이터 출처: https://github.com/abacus-gene/paml-tutorial/tree/main/positive-selection/00_data

[15] http://www.iqtree.org/ 본 논문에서는 version 1.6.12을 사용하였다.

[16] 넥서스 포맷은 분자진화 데이이터 분석에 흔히 사용되는 설정 양식이다.

[17] Álvarez-Carretero et al. 논문에서는 RAxML 프로그램(Stamatakis 2014)을 이용하여 파티션 설정 없이 ML 계통수를 추정하는 과정이 설명되어 있다. 그림 3을 RAxML에 사용하는 것도 가능하다. 구체적인 사용법은 RAxML 설명서를 참조하라.

$2,5,8,\cdots,3k+2,\cdots,1988$를 의미하며 '3-1989\3'도 비슷한 방식으로 정의된다. 이런 형식은 코돈의 세 개 사이트들을 파티션화 하는데 자주 쓰이는 설정 방식이다.[18] 이후 part1 파티션에 GTR+F+G 치환모형, part2 파티션에도 GTR+F+G 치환 모형을 적용시킨다는 설정이다.[19] 네 종류 염기의 빈도는 데이터에서 얻어진 값을 사용하며(+F 옵션), 감마분포로 사이트간 진화속도의 변이를 모형화 한다(+G 옵션).[20]

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.

서 "-seed 1"은 랜덤 넘버의 씨드를 지정해주는 옵션이다. '1' 대신 적당한 양의 정수 설정이 가능하다. 서열데이터의 수 (taxa의 수)가 크면 ML 방법으로 계통수 공간을 모두 탐색하는 것은 현실적으로 불가능하다. 대안으로 계통수 공간의 일부만 탐색하는 heuristic 방법을 사용하게 된다.[21] 씨드 넘버를 지정하여 실행하면 완전히 동일한 상황을 반복할 수 있어 실행 과정중 발생하는 에러나 프로그램 버그에 효율적으로 대처할 수 있다. "-s Mx_aln.phy"는 서열데이터 파일명을 지정하는 옵션이다. "-spp partition_info.txt" 옵션으로 그림 3에서 작성한 파티션 정보를 설정한다. "-bb 1000"는 붓스트랩(bootstrap) 반복횟수를 지정하는 옵션을 나타낸다. 붓스트랩 방법은 얼라인된 각 사이트의 열을 랜덤하게 복원 추출하고 계통수를 추정하는 절차를 반복하여 주어진 clade의 신뢰도를 붓스트랩 확률(bootstrap probability;BP)로 정량화 한다(Felsenstein 1985). IQ-TREE는 Ultrafast bootstrap 방법을 사용하는데 이는 ML 계통수 탐색 과정중에 거쳐간 계통수를 활용하는 방법으로 전통적인 붓스트랩 방법에 비하여 빠르다는 장점이 있다 (Minh et al. 2013). "-redo" 옵션은 과거에 이 명령을 실행한 적이 있는 경우 이전에 생성된 파일에 덧씌우고 다시 결과 파일을 생성하도록 하는 설정이다.

실행결과의 자세한 내용은 partition_info.txt.iqtree 파일에 저장된다. 그림 3에서 설정한 두 파티션에 대한 GTR모형의 모수(식 1의 a,b,c,d,e), 상대적인 진화속도등을 확인할 수 있다. ML 계통수는 partition_info.txt.treefile 파일에 저장되며 이를 FigTree(Rambaut 2010) 같은 프로그램으로 읽어들이면 그림 5과 같다. 내부노드에 표시된 수치는 붓스트랩 확률이다.

다음으로 아미노산 치환 모형으로 계통수를 추정해보자. 그림 4의 실행으로 얻은 결과 파일들을 적절히 백업한 후에 그림 6과 같이 실행한다. 여기서 "-st NT2AA" 옵션은 DNA로부터 아미노산으로 번역하는데 표준 코돈 테이블 (standard codon table) 을 사용한다는 의미이다. 'NT2AA'에 적절한 번호를 덧붙여 다양한 코돈 테이블을 지정할 수 있다. 가령, 포유동물의 미토콘드리아의 경우 NT2AA2를, 효모의 미토콘드리아의 경우 NT2AA3를, 이런식으로 코돈 테이블을 지정한다 (자세한 설정은 IQ-TREE

[18] 연속된 사이트를 파티션으로 지정할 때는 '1-200_400-500' 이런 식으로 스페이스로 띄어 쓰며 구간을 지정한다. 본 데이터는 단백질 코딩 유전자이므로 구간보다는 3으로 나눈 나머지를 기준으로 그룹화 하는 것이 더 합리적이다.

[19] 파티션마다 다른 모형을 지정하는 것도 가능하며, 다른 염기서열 파일을 지정하는 것도 가능하다. 본 분석에서는 두 파티션에 일괄적으로 GTR+F+G 모형을 적용했는데 이전 논문(서태건 2022)처럼 모형비교를 통하여 최적의 모형을 선택/지정하는 방법도 있다.

[20] GTR', '+F', '+G' 옵션에 대한 설명은 이전 논문 (서태건 2022)을 참고하라; IQ-TREE는 '+G'를 '+G4'로 인식한다.

[21] NNI(Nearest Neighbour Interchange), SPR(Subtree Pruning and Regrafting), TBR(Tree Bisection and Reconnection)등 다양한 전략이 있다. https://en.wikipedia.org/wiki/Tree_rearrangement

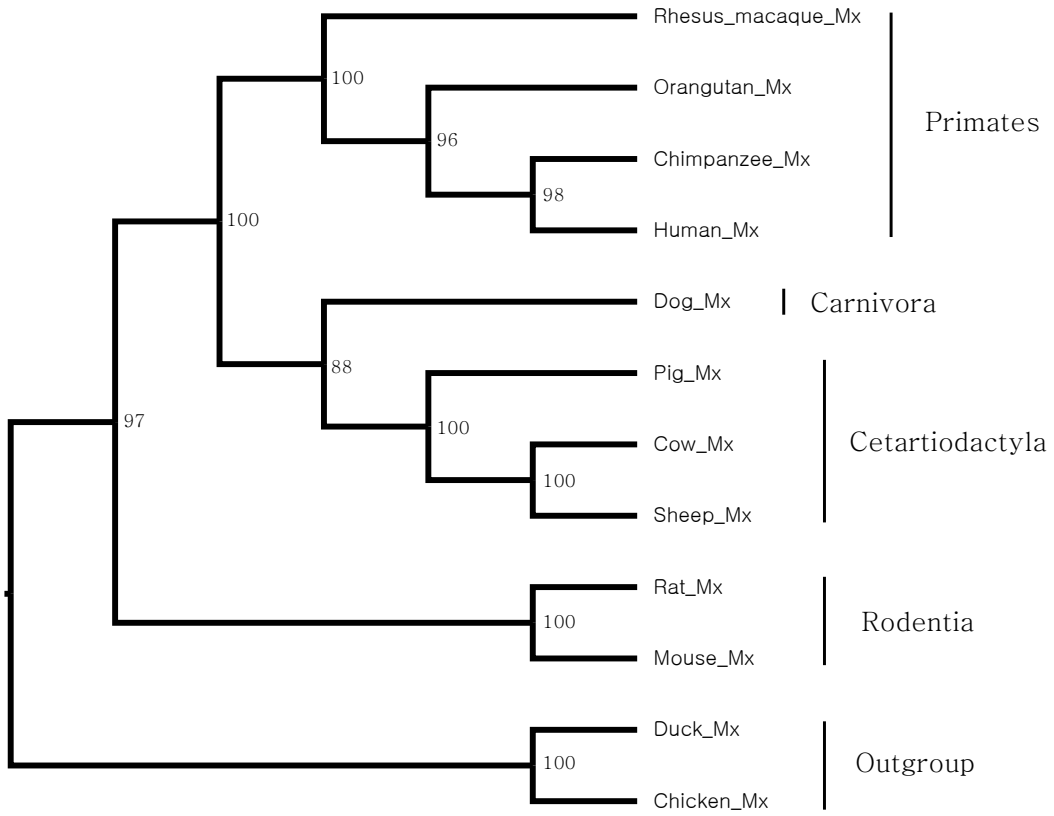

그림 5. partition_info.txt.treefile 파일에 저장된 계통수. 내부노드의 수치는 붓스트랩 확률이다. 계통수 가지의 길이는 무시하고 계통관계만 표현한 cladogram임에 주의하라.

```
 C:\temp>  iqtree -seed 1 -s Mx_aln.phy -st NT2AA -m LG+F+G4 -bb 1000
-redo  ↵
```

그림 6. Mx 염기서열 데이터에 LG+F+G4 아미노산 모델을 적용, ML 계통수를 추정하는 방법.

설명서를 참조하라). “-m LG+F+G4” 설정으로, LG 아미노산 치환모형(LG), 아미노산 빈도는 데이터에서 얻어진 값을 사용 (+F 옵션), 사이트간 진화속도의 이질성 (Rate Heterogeneity Among Site, RHAS; Yang 1994)은 네개의 카테고리를 가지는 이산형 감마분포(+G4)로 지정한다. 분석결과의 상세한 내용은 Mx_aln.phy.iqtree파일에, 추정된 계통수는 Mx_aln.phy.treefile 파일에 저장된다. 계통수를 확인하면 그림 5와는 다른 ML 계통수가 얻어진다. Dog_Mx의 위치만 다른데 목(目) 레벨의 분류군으로 요약하자면 그림 8의 Tree2의 형태이며 Carnivora(Dog_Mx)가 연결된 노드의 BP는 62%이다.

다음으로 GY94 코돈 모형으로 계통수를 추정해보자. 그림 6의 실행으로 얻은 결과 파일들을 적절히 백업한 후에 그림 7과 같이 실행한다. 여기서 “-st CODON1” 옵션은 표준코돈을 사용한다는 의미이다.

```
 C:\temp>  iqtree -seed 1 -s Mx_aln.phy -st CODON1 -m GY+F+G4 -bb
1000 -redo   ↩
```

그림 7. Mx 염기서열 데이터에 GY94+F+G4 코돈 모델을 적용, ML 계통수를 추정하는 방법.

포유동물의 미토콘드리아의 경우 CODON2를, 효모의 미토콘드리아의 경우 CODON3를, 이런 식으로 CODON뒤에 번호를 붙여 다양한 코돈테이블을 지정할 수 있다. “-m GY+F+G4” 설정으로, GY94 코돈 모형, 코돈 빈도는 데이터에서 얻어진 값을 사용 (+F 옵션), 사이트간 진화속도의 이질성 (RHAS)은 네개의 카테고리를 가지는 이산형 감마분포(+G4)로 지정한다. 프로그램 실행이 끝나면 추정된 계통수는 Mx_aln.phy.treefile 이라는 파일에 저장된다. 이를 읽어들이면 그림 5과 동일한 계통관계가 얻어짐을 알 수 있다. Carnivora가 연결된 부위의 BP는 56%로 GTR 모형을 사용하여 얻은 그림 5의 88%보다는 다소 낮다.

이상의 결과를 요약하면, DNA 모형과 코돈 모형을 사용했을때는 그림 8의 Tree1이, 아미노산 모형을 사용했을때는 Tree2가, ML 계통수로 얻어졌다.[22] 이처럼 적용하는 모형에 따라서, 혹은 같은 모형이라도 세부 옵션 설정에 따라서 (예컨데 +G의 설정 유무, 아미노산 모형중에서도 LG, WAG, JTT등 선택에 따라) 다른 계통수가 얻어질 수 있다.[23] 그렇다면 이들 계통수 간의 우열관계가 통계적으로 유의미하다고 볼 수 있을까? 가령 DNA 모형에서는 Tree1이 가장 좋은 계통수이지만, 이것이 Tree2보다 확실히 좋다고 말할 수 있을까? 만약 Tree2가 아슬아슬하게 패하여 1등을 하지 못한 것이라면 Tree2도 Tree1 못지 않게 비중있게 고려되어야 할 것이다.

ML방법으로 추정된 계통수들의 우열관계가 통계적으로 유의한지 아닌지 검정하는 방법으로 여러가지가 있으나 본 논문의 주된 내용이 아니므로 자세한 설명은 생략하고 여기에서는 Kishino-Hasegawa test (KH 검정; Kishino and Hasegawa 1989)과 Shimodaira-Hasegawa test (SH 검정; Shimodaira and Hasegawa 1999)의 기본 아이디어 및 프로그램 결과 해석 방법만 간략하게 소개한다. KH 검정은 사전에 우열관계가 결정되지 않은 두개의 계통수의 로그가능도 (log-likelihood) 스코어 차이가 통계적으로 유의한지 조사하는 방법이고, SH 검정은 여러개의 계통수 중 최대 스코어를 가진 계통수와 나머지 계통수의 스코어의

[22]본 논문의 주제는 통계적인 모형을 이용한 데이터 분석이므로 해당 포유류의 계통관계에 대한 상세한 논의는 생략한다.

[23]Hou et al.(2007)은 아미노산 서열의 p-distance(아미노산 서열의 불일치 비율을 거리로 계산)에 Neighbor-Joining 방법(Saitou and Nei 1987)을 이용하여 Tree2를 최적의 계통수로 추정했다. Álvarez-Carretero et al.(2023)은 DNA 모형을 이용한 ML 방법(본 논문과는 달리 파티션을 고려하지 않음)으로 Tree1을 추정했고 다양한 자연선택 분석을 Tree1을 기반으로 실행했다.

차이가 통계적으로 유의한지 조사하는 방법이다. 일반적으로 계통수간의 우열관계를 정한 후 '1등 계통수'와 그 외의 계통수와의 스코어 차이가 유의한지 알아보기 때문에, KH 검정보다는 SH 검정이 보다 실용적이라 할 수 있다. KH 검정과 SH 검정을 실행할 수 있는 프로그램은 다수 존재하는데 다음 절에서 다루는 codeml 프로그램도 이 중 하나이다.

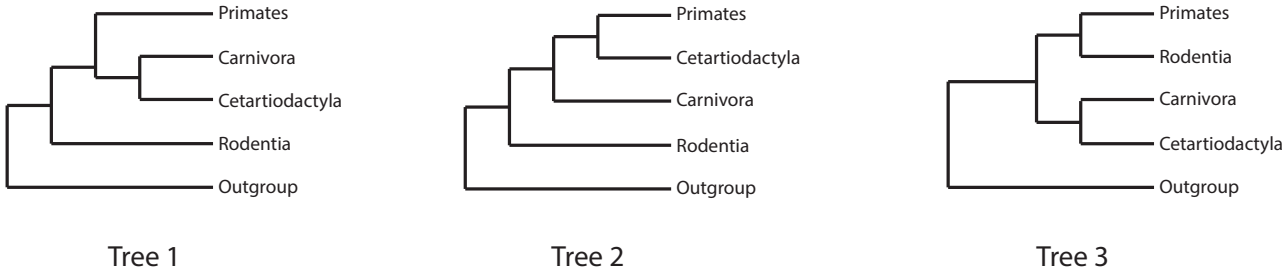

그림 8. 세가지 대표적인 포유류 유전자 계통관계. Tree1은 *Mx* 유전자를 DNA모형, 코돈모형으로 분석하여 얻은 ML 계통수이고 Tree2는 아미노산 모형으로 얻은 ML 계통수이다. Tree3은 Song et al. (2012)이 보고한 종 계통수와 합치하는 계통수이다.

PAML(codeml)을 이용한 자연선택 추정

Codeml 프로그램을 이용하여 *Mx* 유전자에 작용한 자연선택의 세기를 추정해보자. GY94모형 내에서도 ω의 변화 양상에 따라 여러가지 서브모형이 있으나 여기에서는 비교적 단순한 모형인 M0 모형과 Branch 모형만 간략히 설명하겠다. M0 모형은 식 (3)의 ω가 계통수상의 모든 가지에서 그리고 모든 사이트에서 동일하다고 가정한다. 단백질에서 일어나는 아미노산 치환의 대부분은 단백질의 기능을 손상시키기 때문에 음의 선택($\omega < 1$)이 작용하고 치환의 극히 일부분만이 특정 사이트 주위에서 기능을 향상시켜 양의 선택($\omega > 1$)이 작용하는 것이 일반적이다. 또한 양의 선택도 계통수상의 일부분에서 한정적으로 일어나는 경우가 많다. 이러한 ω의 변화 양상을 고려하지 않고 계통수 전체에 대해서 그리고 사이트 전체에 대해서 하나의 공통된 ω값을 구하는 것은 평균값을 구하는 것과 비슷하므로 M0 모형으로 추정된 ω가 1보다 큰 경우는 좀처럼 관찰하기 어렵다. Branch 모형은 M0 모형보다는 더 현실적인 모형으로 계통수 가지 각각이 서로 다른 ω를 갖는다고 가정한다. 하지만 각 사이트는 해당 가지위에서 동일한 ω를 갖는다고 가정하기 때문에 자연선택의 검출 성능은 여전히 제한적이다. 이를 보완하기 위해 Site 모형, Branch-Site 모형이 개발되었다 (Nielsen and Yang 1998; Yang and Nielsen 2002). 이에 대한 상세한 설명은 생략한다.[24]

먼저 M0 모형으로 자연선택을 측정해보자. 분석에 사용될 codeml.exe, lg.dat 파일을 C:\temp 에 복사한 후, 분석에 필요한 모든 설정을 그림 9처럼 C:\temp\codeml.ctl 파일에 저장한다.[25] 각 라인에서 '*'의 우측 부분은 주석부분으로 간주되어 실행에서 무시된다. 'seqfile', 'treefile', 'outfile'에는 각각 서열 데이터 파일명, 계통수 파일명, 결과 파일명이 지정된다. Codeml 프로그램은 계통수를 탐색하는 기능이 없어 계통수를 사용자가 직접 입력해야 한다. 본 예시에서는 Mx_unrooted_trees.txt라는 파일(그림 10)에 3개의

[24] Álvarez-Carretero et al.(2023)은 M0 모형과 Branch 모형뿐만 아니라, Site 모형 Branch-Site모형의 분석 사례도 설명하고 있다.

[25] 본 논문의 분석은 PAML version 4.9j로 수행되었다.

```
      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.

계통수를 지정하였다. 첫 번째 줄에는 taxa의 수와 분석 대상 계통수의 수를 입력하고[26] 그 다음 줄부터 차례대로 계통수를 Newick포맷으로 입력한다. Dog_Mx (Carnivora)의 위치가 다른 세 종류의 계통수를 그림 10처럼 입력하였다. 위에서 설명한 바와 같이 Tree1과 Tree2는 각각 DNA 모형(그리고 코돈모형)과 아미노산 모형으로 얻은 계통수이다. Tree3은 Song et al.(2012)이 보다 많은 taxa와 유전자 데이터 (37 taxa, 447 유전자)를 이용하여 추정한 계통 관계와 합치하는 계통수이다.[27] 그림 10에는 계통수 가지의 길이(branch length)가 포함되어 있지 않지만, 포함된 계통수를 입력해도 무방하다. 입력된 길이는 무시되고 codeml이 ML 방법으로 가지 길이를 추정해준다. 코돈 서열데이터를 분석하므로 codeml.ctl 파일에서

```
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

'seqtype=1'로 설정하고 코돈의 빈도는 'CodonFreq=3'로 설정하여 식 (3) 의 π_s의 값을 서열데이터에서 관찰되는 코돈의 빈도로 설정한다.[28] 'icode=0'으로 표준 코돈테이블을 지정하고 'model=0'으로 M0 모형, 즉 모든 사이트, 모든 계통수가지에서 공통의 ω를 갖는다는 가정을 한다. 'fix_omega' 와 'fix_alpha' 는 0

[26]만약 분석대상인 계통수가 하나라면 '12␣1' 이런식으로 입력한다. 이 경우 첫 번째 계통수 외에 다른 계통수는 무시된다. Branch 모형을 이용한 분석에서 참고하자.

[27]유전자 계통수(gene tree)와 종 계통수(species tree)는 여러 가지 요인에 의해 다를 수 있다. Song et al.이 보고한 계통수는 종 계통수이다.

[28]이를 ML 방법으로 추정하는 것도 가능하지만 60개의 모수를 추정해야하므로 많은 계산시간을 요한다. F1X4, F3X4는 코돈의 빈도가 네 종류 염기 빈도의 곱에 비례하여 결정된다는 설정이다. PAML 설명서를 참조하라.; 'CodonFreq=3' 설정은 Álvarez-Carretero et al.의 설정과 다르므로 결과의 비교에 있어서 주의를 요한다.

으로 설정 (즉, ω 와 α 값을 고정된 값으로 지정하지 않고 데이터로부터 추정함)하여 ML 방법으로 추정을 하도록 하고 각각 그 다음 라인에 초기값을 적당히 지정해준다. 그 외의 사소한 설정은 PAML 설명서 (Yang 2007)를 참조하라.

Codeml.ctl 파일 설정이 모두 끝나면 명령 프롬프트에 아래와 같이 codeml을 실행한다.

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

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

실행 결과는 codeml.ctl 파일에서 지정한대로 'out_M0.txt'에 저장된다. 결과 파일의 가장 마지막 부분을 보자 (그림 12). 그림 8과 10에서 설정한 세 종류 계통수 각각의 로그 가능도(log-likelihood) 스코어가 저장되어 있고 (li 열), 셋 중 최적의 계통수('*'로 표시; Tree1)와 나머지 두 계통수와의 스코어 차이가 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 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

'out_M0.txt' 파일에는 Tree1, Tree2, Tree3 세종류의 계통수 각각에 대해서 추정된 모수와 계통수 가지의 길이가 저장되어 있다. 그림 13에는 Tree1의 결과만을 나타내었다. ω의 추정치는 0.25425이고 이 값은 모든 가지에 대해, 모든 사이트에 대해 동일하다. 그림 13의 하단에서 dN/dS열은 계통수 가지 각각에 대해 추정된 ω값을 나타낸다. M0 모형이므로 모두 동일한 값(0.2542)을 갖고 있다.

이번에는 Branch 모형으로 ω값을 추정해보자. 그림 9에서 'model=1'로 수정하여 Branch 모형을 선택하고 'fix_alpha = 1'과 'alpha = 0'을 설정하여 사이트간 진화속도의 변이를 가정하지 않는다.[29] 또한 'method=1'로 수정하여 계산이 빠르게 되도록 설정한다. 계산 시간을 줄이기 위해 그림 10의 첫째 줄을 '12␣1'로 수정하여 세 계통수의 비교는 생략하고 Tree1에 대해서만 분석을 하도록 하자. 수정이 끝난후 명

[29]Branch 모형은 각 가지마다 ω값을 추정하기 때문에 M0 모형보다 많은 계산 시간이 소요된다. Codeml 프로그램은 Branch 모형과 'fix_alpha=0'을 동시에 허용하지 않는다. 계산 부담을 줄이기 위한 어쩔 수 없는 선택이다.

```
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: 0

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.

령프롬프트에 그림 11와 같이 입력하면 프로그램이 실행되며 그림 14 같은 결과가 얻어진다. 다른 부분의

```
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.

고찰은 생략하고 하단의 dN/dS열을 보자. 그림 13과 달리 각 가지별로 서로다른 ω값을 가짐을 알 수 있다. 가령, 노드 13과 노드 14를 연결하는 가지는 Chicken_Mk와 Duck_Mk의 공동조상에 연결된 내부 가지인데 이 가지의 ω값이 0.1077임을 의미한다.[30] 다른 모든 가지도 ω값이 1보다 작아 Branch 모형으로는 여전히 자연선택의 검출이 쉽지 않음을 알 수 있다.

다음으로 염기서열을 아미노산으로 번역한 후에 아미노산 모형으로 분석을 해보자. 그림 9의 codeml.ctl 파일을 그림 15과 같이 수정하자. 'seqtype=3'으로 코돈에서 아미노산으로 번역을 한다는 것을 명시하고,

```
      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.

'model=3'를 지정하여 식 (2)의 아미노산 빈도를 데이터에서 관찰된 빈도로 구한다. 'aaRatefile=lg.dat' 라인을 추가하여 아미노산 치환 행렬은 그림 1에 나타난 LG 아미노산 모형을 이용한다는 것을 명시한다. 설정 파일 codeml.ctl의 주석 부분에도 설명되어 있듯이 lg.dat외에 wag.dat, jones.dat, dayhoff.dat등의 다양한 아미노산 치환 모형을 설정할 수 있다. 계통수 파일 Mx_unrooted_trees.txt (그림 5)에서 '12␣3'으로 다시 설정하여 세 종류 계통수의 로그가능도 스코어를 통계적으로 비교할 수 있게 설정한다. 설정이 끝난 후 그림 11과 같이 실행한다.

분석결과는 'outfile='에 설정한 바와 같이 out_AA_lg.txt 파일에 저장된다. 세 종류 계통수 각각에 대한 ML 추정된 모수가 표시되고 파일의 마지막에 세 계통수의 로그가능도 스코어차이가 통계적으로 유의한지 그림 16과 같이 표시된다. ML 계통수는 Tree2이고 가능도 스코어가 가장 높다 (-7471.036). Tree2와

[30] 해당 가지 위에서 모든 사이트가 동일하게 0.1077을 ω값을 갖는다는 설정이다. 일반적으로 일부 사이트만 자연선택의 영향을 받는 상황을 생각하면 검출력이 떨어지는 것이 쉽게 이해될 것이다.

```
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.

결론

본 논문에서는 아미노산 치환 모형과 코돈 치환 모형에 대해 설명하고 코돈 모형의 강점인 자연선택의 측정에 대해 분석예제를 통하여 살펴보았다. 또한 자연선택 분석의 선행작업으로 필요한 계통관계 추정을 DNA 치환 모형, 아미노산 치환 모형, 코돈치환 모형 각각을 이용해 수행하였고, ML 계통수와 그 외의 계통수의 차이가 유의한지 통계적으로 검정하는 방법을 대략적으로 설명하였다. 본 논문에서는 다루지 않았지만 PAML 패키지에는 baseml 프로그램이 있어 DNA 모형으로 ML 분석을 하는 것이 가능하다. Baseml.ctl 파일이 설정 파일이며 codeml.ctl과 비슷한 포맷으로 구성되어 있고 중복되는 항목도 많아 이해가 어렵지 않을 것이다. 또한 그림 10처럼 복수의 계통수를 지정하여 SH 검정도 실행 가능하다.[31]

표 1에서 설명한 바와 같이 코돈모형은 계산시간이 많이 소요된다는 단점이 있다. 따라서 염기서열 수가 많은 대용량 데이터의 경우, 코돈모형을 이용하여 계통수를 추정하는 것은 현실적으로 어려움이 많을 것이다. 이 경우 DNA 모형이나 아미노산 모형을 사용하여 계통수 탐색을 하게 된다. 계통수의 뿌리쪽에 가까운 deep branch의 경우 치환의 포화상태를 우려하여 아미노산 모형을 흔히 사용하지만, DNA 모형도 아미노산 모형 못지 않게 좋은 성능을 보여준다는 최근 보고도 있다(Kapli et al. 2023). DNA 모형을 이용한 분석에서는 첫번째 사이트와 두번째 사이트를 분리할지 혹은 병합할지, 세번째 사이트를 데이터 분석에서 아예 제외할지 혹은 포함할지, 포함할 경우 진화속도는 나머지 두 사이트에 비례할지 혹은 독립적일지, 등등 여러가지 설정의 조합을 생각해 볼 수 있고 그에 따라 얻어지는 추정 결과에도 다소 편차가 있을 수 있다. 추정된 계통관계에서 공통적으로 변하지 않는 부분도 있겠지만 설정에 민감하게 영향을 받는 부분도 있을 것이다. 계통 관계의 추정은 그 자체가 연구의 주 목적인 경우도 있지만 다른 연구를 하기 위한 중간 단계인 경우도 많다. 추정된 계통 관계가 다른 추가적인 분석의 재료로 사용된다면 ML 계통수뿐만 아니라 근소한 차이로 ML 계통수가 되지 못한 다른 계통수도 다음 단계의 분석에 이용해

[31] 본 논문에서 계통수 추정에 사용했던 IQ-TREE 프로그램도 SH검정을 할 수 있다. IQ-TREE 프로그램 설명서에서 “tree topology tests” 키워드로 검색하면 해당 내용을 찾을 수 있다.

봄으로써 최종적으로 얻는 결과가 얼마나 로버스트(robust)한지 살펴보는 것이 좋다.[32]

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. 본 논문에서 소개한 내용이 원저 논문의 이해와 분자진화 데이터 분석의 기초 형성에 조금이나마 도움이 되기를 기대한다.

관련 자료

본 논문의 분석에서 사용된 설정/결과 파일등은 한국진화학회 홈페이지에서 찾아 볼 수 있다.

감사의 글

본연구는 해양수산부의 재원으로 극지연구소의 지원을 받아 수행되었다 (과제번호: PE23140).

참고문헌

Adachi J., Hasegawa M. 1996. Model of amino acid substitution in proteins encoded by mitochondrial DNA. J. Mol. Evol. 42:459-468.

Adachi J., Waddell P.J., Martin W., Hasegawa M. 2000. Plastid genome phylogeny and a model of amino acid substitution for proteins encoded by chloroplast DNA. J. Mol. Evol. 50:348-358.

Álvarez-Carretero S., Kapli P., Yang Z. 2023. Beginner's Guide on the Use of PAML to Detect Positive Selection. Mol. Biol. Evol. 40(4):msad041

Akaike H. 1974. A new look at the statistical model identification. IEEE Trans. Autom. Contr. 19:716-723.

Dayhoff M.O., Schwartz R.M., Orcutt B.C. 1978. A model of evolutionary change in proteins. Atlas of protein sequence and structure. Vol 5, 345-352, National Biomedical Research Foundation, Washington DC.

Felsenstein J. 1985. Confidence Limits on Phylogenies: An Approach Using the Bootstrap. Evolution. 39(4):783-791.

Felsenstein J. 2004. Inferring Phylogenies. Sinauer Associates, Sunderland, MA.

Goldman N., Yang Z. 1994. A codon-based model of nucleotide substitution for protein-coding DNA sequences. Mol. Biol. Evol. 11:725-736.

Hasegawa H., Kishino H., Yano T. 1985. Dating the human-ape splitting by a molecular clock of mitochondrial DNA. J. Mol. Evol. 22:160–174.

[32] 데이터 분석이 여러 단계로 나뉘어져 있을 경우 각 단계에서의 결과의 불확실성(uncertainty)은 흔히 간과된다. 예컨데 염기서열 얼라인 단계에서 불확실성을 고려하지 않고 정렬된 데이터 세트 하나만 채택한다든지, ML 계통수 하나만 채택하고 나머지 계통수는 무시하며 다음 단계로 진행한다든지 하는 것이다. 각 단계의 결과의 불확실성에도 불구하고 연구자가 얻는 최종 결과가 로버스트함을 보여준다면 최종 연구결과의 신뢰도는 그만큼 높아질 것이다.

Hou Z.-C., Xu G.-Y., Su Z., Yang N. 2007. Purifying selection and positive selection on the myxovirus resistance gene in mammals and chickens Gene 396(1):188-195.

Jones D.T., Taylor W.R., Thornton J.M. 1992. The rapid generation of mutation data matrices from protein sequences. CABIOS 8:275-282.

Kapli P., Kotari I., Telford M.J., Goldman N., Yang Z. 2023. DNA Sequences Are as Useful as Protein Sequences for Inferring Deep Phylogenies. Syst. Biol. 72(5):1119–1135.

Kishino H., Hasegawa M. 1989. Evaluation of the maximum likelihood estimate of the evolutionary tree topologies from DNA sequence data and the branching order in hominoidea. J. Mol. Evol. 29:170-179.

Le S.Q., Gascuel O. 2008. An improved general amino acid replacement matrix. Mol. Biol. Evol. 25(7):1307-1320.

Maynard Smith J. Smith N.H. 1996. Synonymous nucleotide divergence: what is ”saturation”? Genetics 142(3):1033-1036.

Minh B.Q., Nguyen M.A.T., Haeseler A. 2013. Ultrafast Approximation for Phylogenetic Bootstrap. Mol. Biol. Evol. 30(5):1188-1195.

Mount D.W. 2004. Bioinformatics:Sequence and Genome Analysis(2/e). pp 94–102. Cold Spring Harbor Laboratory Press.

Muse S.V., Gaut B.S. 1994. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome Mol. Biol. Evol. 11(5):715-724.

Nielsen R., Yang Z. 1998. Likelihood models for detecting positively selected amino acid sites and applications to the HIV-1 envelope gene. Genetics 148:929-936.

Plotkin J.B., Kudla G. 2011. Synonymous but not the same: the causes and consequences of codon bias. Nature Reviews Genetics 12:32-42.

Rambaut A. 2010. FigTree v1.3.1. Institute of Evolutionary Biology, University of Edinburgh, Edinburgh. http://tree.bio.ed.ac.uk/software/figtree/

Saitou N., Nei M. 1987. The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4(4):406-425.

Schwarz G. 1978. Estimating the dimension of a model. Ann. Stat. 6:461-464.

Seo T.-K., Kishino H. 2008. Synonymous Substitutions Substantially Improve Evolutionary Inference from Highly Diverged Proteins. Syst. Biol. 57(3):367-377.

Seo T.-K., Kishino H. 2009. Statistical Comparison of Nucleotide, Amino Acid, and Codon Substitution Models for Evolutionary Analysis of Protein-Coding Sequences. Syst. Biol. 58(2):199-210.

Shimodaira H, Hasegawa M. 1999. Multiple comparisons of log-likelihoods with applications to phylogenetic inference. Mol. Biol. Evol. 16:1114-1116.

Song S., Liu L., Edwards S.V., Wu S. 2012. Resolving conflict in eutherian mammal phylogeny using

phylogenomics and the multispecies coalescent model. Proc. Natl. Acad. Sci. USA 09(37):14942-14947.

Stamatakis A. 2014. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies Bioinformatics. 30(9): 1312-1313.

Tamura K., Nei M. 1993. Estimation of the number of nucleotide substitutions in the control region of mitochondrial DNA in humans and chimpanzees. Mol. Biol. Evol. 10:512-526.

Tavaré, S. 1986. Some probabilistic and statistical problems on the analysis of DNA sequences. Lect. Math. Life Sci. 17:57-86.

Whelan S., Goldman N. 2001. A general empirical model of protein evolution derived from multiple protein families using a maximum likelihood approach. Mol. Biol. Evol. 18:691-699.

Yang Z. 1994. Maximum likelihood phylogenetic estimation from DNA sequences with variable rates over sites: approximate methods. J. Mol. Evol. 39:306-314.

Yang Z. 2006. Computational Molecular Evolution. Oxford University Press.

Yang Z. 2007. PAML 4: a program package for phylogenetic analysis by maximum likelihood. Mol. Biol. Evol. 24:1586-1591

Yang Z., Nielsen R. 2002. Codon-substitution models for detecting molecular adaptation at individual sites along specific lineages. Mol. Biol. Evol. 19:908-917.

김우철 2021. 수리통계학(개정판). 민영사.

서태건 2022. DNA 염기치환 모형의 비교. 한국진화학회지 1:88-104.

영문초록

Title: Evolutionary models of amino acid and codon sequences

Abstract: In the evolutionary analyses of protein-coding DNA sequences, 20-state models of amino acid replacement and/or 61-state models of codon substitution are widely adopted. In this review article, we briefly overview the features of amino acid and codon models and discuss their advantages and disadvantages. By using example sequence data from mammalian species, we also show how to infer and compare phylogenies and how to measure positive selection applied during molecular evolution.

Authors: Seo, Tae-Kun [1, *]

Affiliation: [1] Division of Life Sciences, Korea Polar Research Institute, 26 Songdomirae-ro, Yeonsu-gu, Incheon 21990,Republic of Korea

***Corresponding author:** seo.taekun@gmail.com