User's Manual

KaKs_Calculator Toolbox 2.0

Version 2.0 (June, 1 2009)

Da-Peng Wang (wangdp@big.ac.cn)

CAS Key Laboratory of Genome Sciences and Information, Beijing Institute of Genomics, Chinese Academy of Sciences, Beijing 100029, PR China

Yu-Bin Zhang (ybzhang@big.ac.cn)

CAS Key Laboratory of Genome Sciences and Information, Beijing Institute of Genomics, Chinese Academy of Sciences, Beijing 100029, PR China

Zhang Zhang (zhangzhang.cn@gmail.com)

Plant Stress Genomics Research Center, Division of Chemical and Life Sciences & Engineering, King Abdullah University of Science and Technology, Thuwal 23955-6900, Kingdom of Saudi Arabia

Jun Yu* (junyu@big.ac.cn)

CAS Key Laboratory of Genome Sciences and Information, Beijing Institute of Genomics, Chinese Academy of Sciences, Beijing 100029, PR China

***Corresponding author**

Content

1. Introduction

KaKs_Calculator 2.0 is a toolbox that calculates nonsynonymous (Ka) and synonymous (Ks) substitution rates by means of various models or model selection and averaging. Furthermore, it can detect positive selected sites (PSS) based on chosen Ka/Ks methods, by incorporating sliding window strategy. In particular, we have added the gamma-series methods such as γ-NG, γ-LWL, γ-MLWL, γ-LPB, γ-MLPB, γ-YN and γ-MYN, which were developed by us.

The KaKs_Calculator 2.0 toolbox, including source codes, compiled executables and documentation, is freely available for academic use only at https://sourceforge.net/projects/kakscalculator2.

2. Core tool

2.1 Methods for Calculating Ka and Ks

Calculating Ka and Ks normally involves three steps. Let us assume that the number of lengths between two DNA sequences compared is *n* and the number of substitutions between them is *m*. To calculate Ka and Ks, we need to count the numbers of synonymous (S) and nonsynonymous (N) sites (S+N=*n*) and the numbers of synonymous

(S_d) and nonsynonymous (N_d) substitutions ($S_d+N_d=m$). Then it is after correcting multiple substitutions that (N_d/N) and (S_d/S) could represent Ka and Ks, respectively, since the observed number of substitutions underestimates the real number of substitutions as sequences diverge over time. Therefore, we can conclude from mentioned above that these methods normally involve three steps to estimate Ka and Ks: counting S and N, counting S_d and N_d, and correction for multiple substitutions.

Methods for calculating Ka and Ks adopt different substitution models with subtle yet significant differences. They can be classified as approximate methods and maximum-likelihood methods. Different from approximate methods, maximum-likelihood methods adopt the probability theory to finish all three steps mentioned above in one go.

2.1.1 Approximate Methods

There are several approximate methods incorporated into KaKs_Calculator, and we list their abbreviations in the program and their corresponding reference(s) as follows.

●NG: Nei, M. and Gojobori, T. (1986)

●LWL: Li, W.H., et al. (1985)

●LPB: Li, W.H. (1993) and Pamilo, P. and Bianchi, N.O. (1993)

●MLWL (Modified LWL), MLPB (Modified LPB): Tzeng, Y.H., et al. (2004)

●YN: Yang, Z. and Nielsen, R. (2000)

●MYN (Modified YN): Zhang, Z., et al. (2006)

●γ-MYN: Wang, D.P., et al. Biology Direct. (2009)

●γ-NG, γ-LWL, γ-MLWL, γ-LPB, γ-MLPB, γ-YN: Wang, D.P., et al. Geno, Prot &Bioi. (2009)

2.1.2 Maximum-Likelihood Methods

The method of GY takes account of sequence evolutionary features, such as transition/transversion rate ratio and nucleotide frequencies (reflected in the HKY Model) and incorporates these features into a codon-based model. We extend this method to a set of candidate models in a maximum likelihood framework and use the AICc for model selection and model averaging.

●GY: Goldman, N. and Yang, Z. (1994)

●MS (Model Selection), MA (Model Averaging): based on a set of candidate models defined by Posada, D. (2003) as follows.

Model	Substitution Rates	Nucleotide Frequency
JC F81	$r_{TC}=r_{AG}=r_{TA}=r_{CG}=r_{TG}=r_{CA}$	Equal Unequal
K2P HKY	$r_{TC}=r_{AG}\neq r_{TA}=r_{CG}=r_{TG}=r_{CA}$	Equal Unequal
TrNEF TrN	$r_{TC}\neq r_{AG}\neq r_{TA}=r_{CG}=r_{TG}=r_{CA}$	Equal Unequal
K3P K3PUF	$r_{TC}=r_{AG}\neq r_{TA}=r_{CG}\neq r_{TG}=r_{CA}$	Equal Unequal
TIMEF TIM	$r_{TC}\neq r_{AG}\neq r_{TA}=r_{CG}\neq r_{TG}=r_{CA}$	Equal Unequal
TVMEF TVM	$r_{TC}=r_{AG}\neq r_{TA}\neq r_{CG}\neq r_{TG}\neq r_{CA}$	Equal Unequal
SYM GTR	$r_{TC}\neq r_{AG}\neq r_{TA}\neq r_{CG}\neq r_{TG}\neq r_{CA}$	Equal Unequal

r_{ij}: substitution rate between i and j, where $i \neq j$ and $i, j \in [A, C, G, T]$

2.2 Installation

For high efficiency and compatibility with more platforms, the kernel codes of KaKs_Calculator are written in standard C++. For Windows version we use Visual C++ 6.0 for GUI (Graphics User Interface). You can download the newest package from the webpage at https://sourceforge.net/projects/kakscalculator2.

2.2.1 Linux/Unix

KaKs_Calculator 2.0 has been tested on ROCKS LINUX 4.3 X86-64 platform.
.

●Unpack the package of KaKs_CalculatorXXX.tar.gz by the following commands.

```
gzip –d KaKs_CalculatorXXX.tar.gz
tar –xf KaKs_CalculatorXXX.tar
```

●If you use other Linux/Unix OS, you have to compile the program in the source codes folder with the help of g++/gcc compiler by yourselves.

```
cd KaKs_CalculatorXXX/src
make
```

2.2.2 Windows

The Windows version of KaKs_Calculator can run on any IBM compatible computer under Windows Operating System (tested on Windows 2000/XP/Vista).

●Unpack the package of KaKs_CalculatorXXX.tar.gz.

●In the folder of "KaKs_CalculatorXXX/bin/Windows/", just click 'KaKs_CalculatorXXX.exe' for execution.

2.3 Format of Sequence

KaKs_Calculator accepts quasi-AXT sequence format as follows. Before calculation, gaps and stop codons between compared sequences will be removed. You can also see "example.axt" in the folder of "KaKs_CalculatorXXX/examples/".

For example:

```
NP_000026
ATGCTCCTGTG-CCACTGGCC
ATCCCC-TGCGCTCACTGGAC
NP_000053
ACAGaTtCTACCc-GCCcACTA--GgtGtt
---ggTTCTCCtACCcA-G-CACTACTggg
```

Each pair of sequences in an axt file contains three lines: a sequence name line and 2 sequence lines. Pairwise sequences are separated from one another by blank lines.

●Sequence name line

```
NP_000026
```

.

●Pairwise sequences lines

```
ATGCTCCTGTG-CCACTGGCC
```

```
ATCCCC-TGCGCTCACTGGAC
```

2.4 Parameters setting

2.4.1 Linux/Unix

KaKs_Calculator are more suitable for a large number of dataset to calculate Ka and Ks. It reads a pair of sequences and computes corresponding estimates one by one, so that it requires memory proportional to the maximum length among pairwise sequences. In addition, KaKs_Calculator allows user to choose more than one method to calculate Ka and Ks at one running time. The following is the parameters' setting in Linux version.

● -i AXT sequence file name for calculating Ka and Ks

● -o File name for outputting results

● -c Genetic code (Default = 1-Standard Code). For more information about the Genetic Codes, please see the link:

http://www.ncbi.nlm.nih.gov/Taxonomy/Utils/wprintgc.cgi?mode=c

● -m Methods for calculating Ka and Ks (Default = MA): NG, LWL, LPB, MLWL, MLPB, YN, MYN, GY, MS, MA, GNG, GLWL, GMLWL, GLPB, GMLPB, GYN, and GMYN, ALL (including all above methods). Note that γ-NG, γ-LWL, γ-MLWL, γ-LPB, γ-MLPB, γ-YN, and γ-MYN have been renamed as GNG, GLWL, GMLWL, GLPB, GMLPB, GYN, and GMYN for typing.

● -d File name for details about each candidate model only when using the method of MS or MA

● -h Show help information

For example:

● use MA method and standard code

```
KaKs_Calculator -i test.axt -o test.axt.kaks
```

● use MA method and vertebrate mitochondrial code

```
KaKs_Calculator -i test.axt -o test.axt.kaks -c 2
```

● use MA method and standard code and output details of model selection on each candidate model

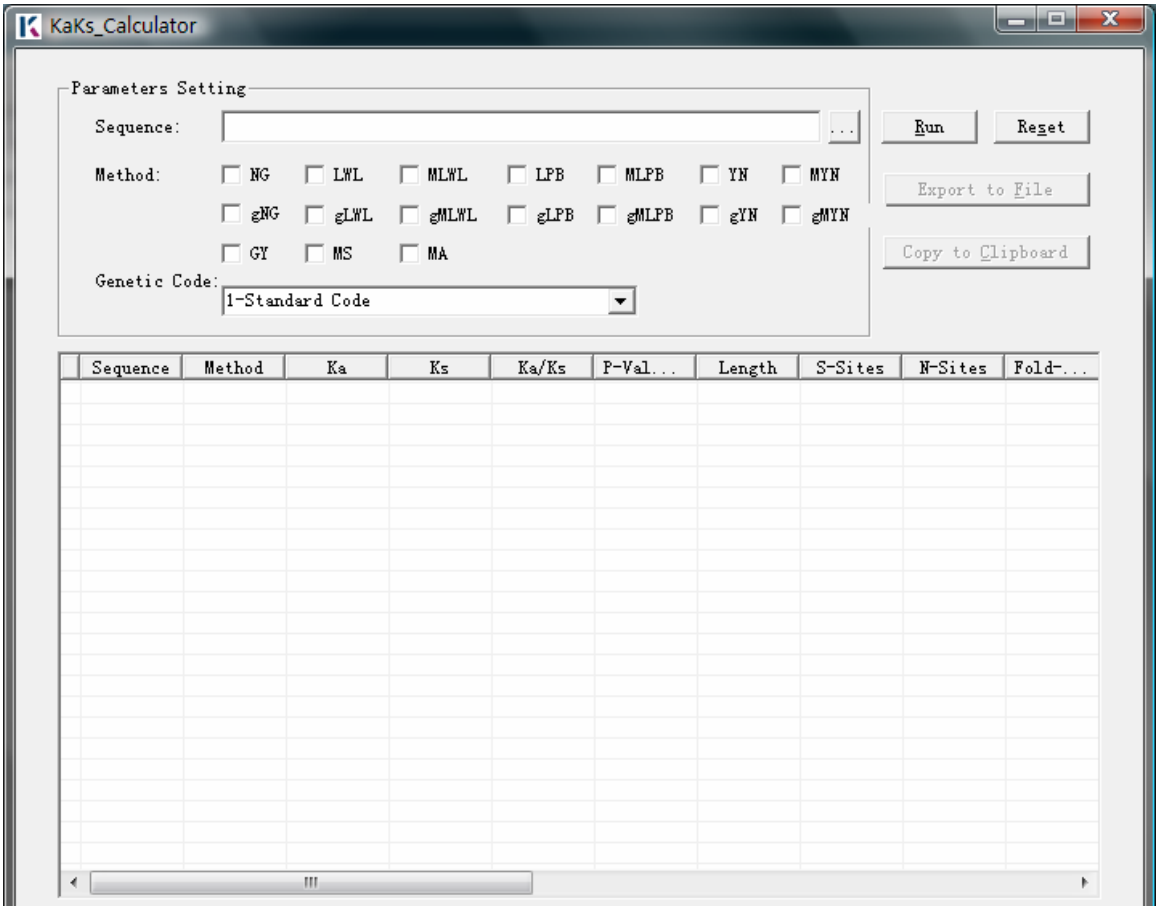
```
KaKs_Calculator -i test.axt -o test.axt.kaks –d test.axt.details
```

● use LWL, YN and MYN and standard Code

```
KaKs_Calculator -i test.axt -o test.axt.kaks -m LWL -m YN -m MYN
```

2.4.2 Windows

The Windows version provides users with a friendly interface to select input sequences' file, genetic code and method(s) for estimating Ka and Ks. During calculating you can minimize the application window and send it to tray. After finishing calculation, KaKs_Calculator allows users to exports results to file or clipboard at will. For convenience, γ-NG, γ-LWL, γ-MLWL, γ-LPB, γ-MLPB, γ-YN, and γ-MYN have been renamed as gNG, gLWL, gMLWL, gLPB, gMLPB, gYN, and gMYN.



2.5 Output Format

KaKs_Calculator provides comprehensive information estimated from compared sequences, including numbers of synonymous and nonsynonymous sites, numbers of synonymous and nonsynonymous substitutions, GC contents, maximum-likelihood score, and AICC, in addition to synonymous and nonsynonymous substitution rates and their ratio. Meanwhile, Fisher's exact test for small sample is applied to justify the validity of Ka and Ks calculated by these methods.

● Sequence: Name of Pairwise sequence

● Method: Name of method for calculation of Ka and Ks

● Ka: Nonsynonymous substitution rate

● Ks: Synonymous substitution rate

● Ka/Ks: Selective strength

● P-Value (Fisher): The value computed by Fisher exact test

● Length: Sequence length (after removing gaps and stop codon(s))

● S-Sites: Synonymous sites

● N-Sites: Nonsynonymous sites

● Fold-Sites (0:2:4): 0,2,4-fold degenerate sites

● Substitutions: Substitutions between sequences

● S-Substitutions: Synonymous substitutions

● N-Substitutions: Nonsynonymous substitutions

● Fold-S-Substitutions (0:2:4): Synonymous substitutions at 0,2,4-fold

● Fold-N-Substitutions (0:2:4): Nonsynonymous substitutions at 0,2,4-fold

● Divergence-Time: Divergence time

● Substitution-Rate-Ratio (rTC:rAG:rTA:rCG:rTG:rCA/rCA): Ratios of six substitution rates to the substitution rate between C and A

● GC(1:2:3): GC content of entire sequences and of three codon positions

● ML-Score: Maximum likelihood score

● AICc: Value of AICc

● Akaike-Weight: Value of Akaike weight for model selection

● Model: Selected model for the method of MS

3. Expanding tools

3.1 Overview

These expanding tools mainly contain three modules such as ***SPLIT***, ***PLOT*** and ***DPSS***, and the detailed descriptions are listed:

SPLIT**:** to divide the raw paired orthologs into many parts by means of sliding window strategy, of which the window length and step length should be set.

PLOT**:** to plot various figures of Ka, Ks and Ka/Ks in sliding windows, with consideration of different methods and sequences.

DPSS**:** to detect the positive selected sites according to the results of sliding windows.

3.2 Installation or compilation

Before using these tools, ***Rserve package of R statistical software*** (version>= 2.9.0) and ***java software*** should be installed by user.

R statistical software can be downloaded on the website: http://ftp.ctex.org/mirrors/CRAN/

Rserve package can be accessed from "../bin/Windows/ Rserve_0.6-0.zip" in Windows platform or "../bin/Linux/ Rserve_0.6-0.tar.tar" in Linux platform, after the software package has been decompressed.

Please note that the *Rserve server* should keep running in all procedures when the expanding tools are used.

We provide compiled class and jar files, alternatively, source code of java, in the conditions of both Linux and Windows. We only describe the compiling procedures of source code in the following:

Linux/Unix/ Windows

```
javac –classpath ./REngine.jar; ./RserveEngine.jar plot.java
javac split.java
javac dpss.java
```

3.3 Usage

All of these three modules can deal with large scale level data.

Note that user should start the ***Rserve*** server in R environment like this:

```
library(Rserve);
Rserve();
```

SPLIT

```
Java split example.axt [window length] [step length]
```

Note: this function yield a file by adding split_[window length]_ [step length] into raw

name. Later, this resulting file can be imported into KaKs_Calculator 2.0 Core tool to calculate the Ka and Ks values. In particular, [window length] and [step length] should be multiple of three to meet the requirement of codons.

PLOT

```
Java plot examplesplit_57_6.axt.kaks
Or   Java –jar plot.jar examplesplit_57_6.axt.kaks
```

Note: this file "**examplesplit_57_6.axt.kaks**" is just resulting from KaKs_Calculator 2.0 Core tool.

DPSS

```
Java dpss examplesplit_57_6.axt.kaks
```

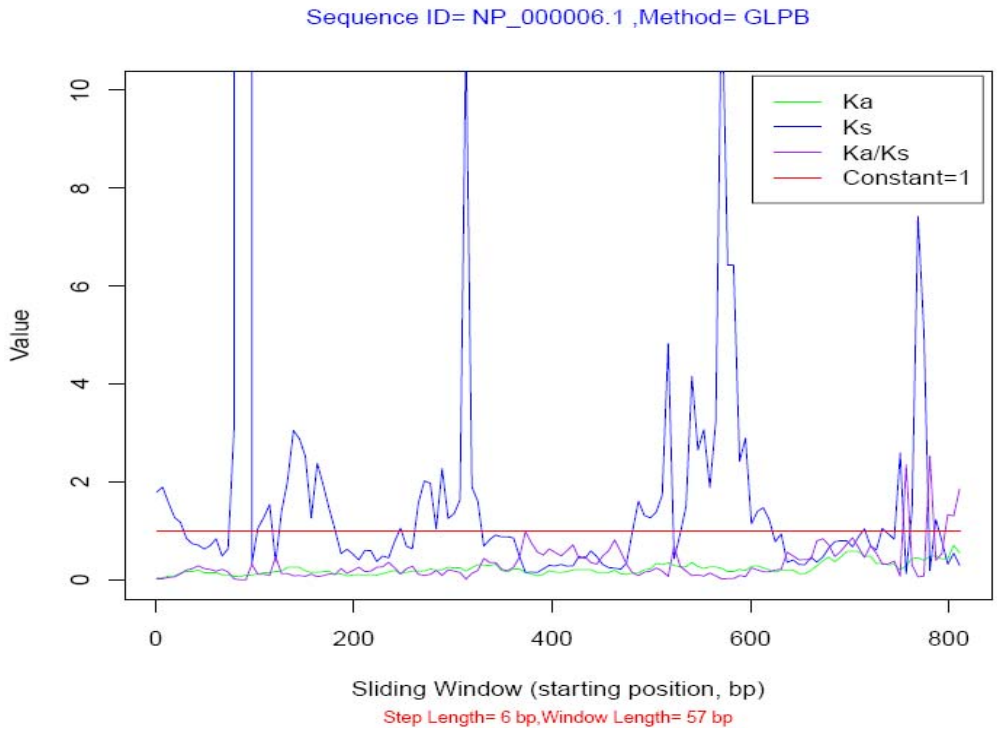
Note: please do not rename the files from all the procedures.

3.4 Output

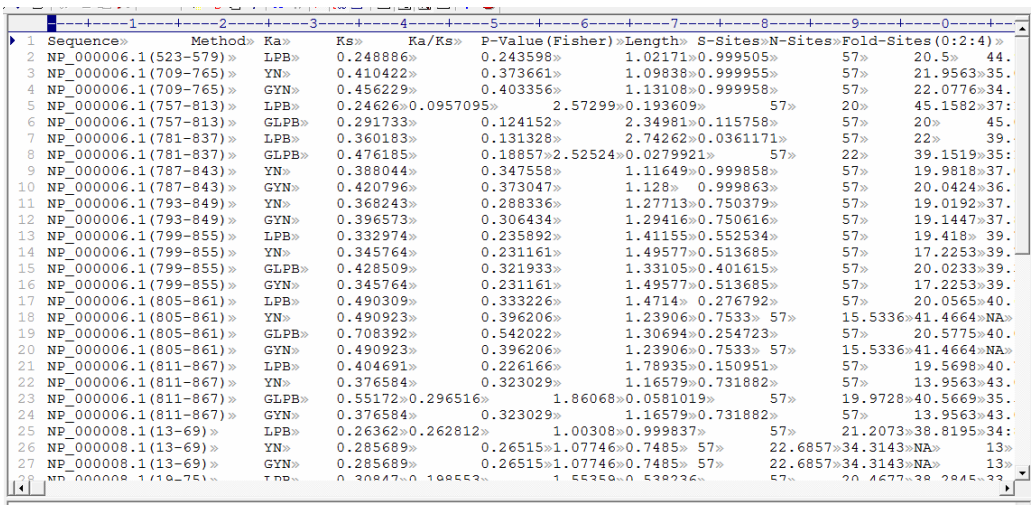
SPLIT: the results from raw orthologs pairs by window sliding, including sequence id, beginning coordinate, ending coordinate and aligned sequences. A file including cutting sequences can be achieved in the current directory.

```
NP_000006.1(1-57)
ATGGACATTGAAGCATATTTTGAAAGAATTGGCTATAAGAACTCTAGGAACAAATTG
ATGGACATCGAAGCATACTTTGAAAGGATTGGTTACAAGAACTCAGTGAATAAATTG

NP_000006.1(7-63)
ATTGAAGCATATTTTGAAAGAATTGGCTATAAGAACTCTAGGAACAAATTGGACTTG
ATCGAAGCATACTTTGAAAGGATTGGTTACAAGAACTCAGTGAATAAATTGGACTTA

NP_000006.1(13-69)
GCATATTTTGAAAGAATTGGCTATAAGAACTCTAGGAACAAATTGGACTTGGAAACA
GCATACTTTGAAAGGATTGGTTACAAGAACTCAGTGAATAAATTGGACTTAGCCACA

NP_000006.1(19-75)
TTTGAAAGAATTGGCTATAAGAACTCTAGGAACAAATTGGACTTGGAAACATTAACT
TTTGAAAGGATTGGTTACAAGAACTCAGTGAATAAATTGGACTTAGCCACATTAACT

NP_000006.1(25-81)
AGAATTGGCTATAAGAACTCTAGGAACAAATTGGACTTGGAAACATTAACTGACATT
AGGATTGGTTACAAGAACTCAGTGAATAAATTGGACTTAGCCACATTAACTGAAGTT

NP_000006.1(31-87)
GGCTATAAGAACTCTAGGAACAAATTGGACTTGGAAACATTAACTGACATTCTTGAG
GGTTACAAGAACTCAGTGAATAAATTGGACTTAGCCACATTAACTGAAGTTCTTCAG

NP_000006.1(37-93)
AAGAACTCTAGGAACAAATTGGACTTGGAAACATTAACTGACATTCTTGAGCACCAG
AAGAACTCAGTGAATAAATTGGACTTAGCCACATTAACTGAAGTTCTTCAGCACCAG
```

PLOT: this function results in a lot of figures in pdf format, which can be found in the directory of "../figure/". Also, the raw corresponding files can be found in the directory of "../temp/". Sequence id, method name, step length, window length have been shown in titles. And Ka, Ks and Ka/Ks curves have been drawn, in comparison with the level 1.



DPSS: this module can lead to potential positive selected sites, based on the former results.



	Sequence	Method	Ka	Ks	Ka/Ks	P-Value(Fisher)	Length	S-Sites	N-Sites	Fold-Sites(0:2:4)
2	NP_000006.1(523-579)	LPB	0.248886	0.243598		1.02171	0.999505	57	20.5	44.
3	NP_000006.1(709-765)	YN	0.410422	0.373661		1.09838	0.999955	57	21.9563	35.
4	NP_000006.1(709-765)	GYN	0.456229	0.403356		1.13108	0.999958	57	22.0776	34.
5	NP_000006.1(757-813)	LPB	0.24626	0.0957095	2.57299	0.193609	57	20	45.1582	37:
6	NP_000006.1(757-813)	GLPB	0.291733	0.124152		2.34981	0.115758	57	20	45.
7	NP_000006.1(781-837)	LPB	0.360183	0.131328		2.74262	0.0361171	57	22	39.
8	NP_000006.1(781-837)	GLPB	0.476185	0.18857	2.52524	0.0279921	57	22	39.1519	35:
9	NP_000006.1(787-843)	YN	0.388044	0.347558		1.11649	0.999858	57	19.9818	37.
10	NP_000006.1(787-843)	GYN	0.420796	0.373047		1.128	0.999863	57	20.0424	36.
11	NP_000006.1(793-849)	YN	0.368243	0.288336		1.27713	0.750379	57	19.0192	37.
12	NP_000006.1(793-849)	GYN	0.396573	0.306434		1.29416	0.750616	57	19.1447	37.
13	NP_000006.1(799-855)	LPB	0.332974	0.235892		1.41155	0.552534	57	19.418	39.
14	NP_000006.1(799-855)	YN	0.345764	0.231161		1.49577	0.513685	57	17.2253	39.
15	NP_000006.1(799-855)	GLPB	0.428509	0.321933		1.33105	0.401615	57	20.0233	39.
16	NP_000006.1(799-855)	GYN	0.345764	0.231161		1.49577	0.513685	57	17.2253	39.
17	NP_000006.1(805-861)	LPB	0.490309	0.333226		1.4714	0.276792	57	20.0565	40.
18	NP_000006.1(805-861)	YN	0.490923	0.396206		1.23906	0.7533	57	15.5336	41.4664	NA
19	NP_000006.1(805-861)	GLPB	0.708392	0.542022		1.30694	0.254723	57	20.5775	40.
20	NP_000006.1(805-861)	GYN	0.490923	0.396206		1.23906	0.7533	57	15.5336	41.4664	NA
21	NP_000006.1(811-867)	LPB	0.404691	0.226166		1.78935	0.150951	57	19.5698	40.
22	NP_000006.1(811-867)	YN	0.376584	0.323029		1.16579	0.731882	57	13.9563	43.
23	NP_000006.1(811-867)	GLPB	0.55172	0.296516	1.86068	0.0581019	57	19.9728	40.5669	35.
24	NP_000006.1(811-867)	GYN	0.376584	0.323029		1.16579	0.731882	57	13.9563	43.
25	NP_000008.1(13-69)	LPB	0.26362	0.262812	1.00308	0.999837	57	21.2073	38.8195	34:
26	NP_000008.1(13-69)	YN	0.285689	0.26515	1.07746	0.7485	57	22.6857	34.3143	NA	13
27	NP_000008.1(13-69)	GYN	0.285689	0.26515	1.07746	0.7485	57	22.6857	34.3143	NA	13

4. Acknowledgements

In the 1.0 version, we thank Professor Wen-Hsiung Li for providing us with his computer program and Professor Ziheng Yang for his invaluable source codes in PAML. We are grateful to Heng Li for his advice and Yafeng Hu for his help in software designing. We also thank all anonymous users for reporting bugs and sending suggestions.

In the 2.0 version, we acknowledge Mr. Yu-Bin Sun, Mr. Yun-Chao Ling and Mr. Song Zhang for their valuable technical support.

5. References

[1] Agresti, A. 1992. A Survey of Exact Inference for Contingency Tables. Statistical Science. 7, 131 -177.

[2] Akaike, H. 1973 Information theory as an extension of the maximum likelihood principle. In Petrov, B.N. and Csaki, F. (eds), Second International Symposium on Information Theory. Akademiai Kiado, Budapest, 267-281

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

[4] Bierne, N. and Eyre-Walker, A. 2003. The Problem of Counting Sites in the Estimation of the Synonymous and Nonsynonymous Substitution Rates: Implications for the Correlation Between the Synonymous Substitution Rate and Codon Usage Bias. Genetics. 165, 1587-1597.

[5] Burnham, K.P. and Anderson, D.R. 2002 Model Selection and Multimodel Inference: A Practical Information Theoretic Approach. In. Springer-Verlag, New York, 488.

[6] Burnham, K.P. and Anderson, D.R. 2004 Multimodel Inference: Understanding AIC and BIC in Model Selection, Sociological Methods Research, 33, 261-304.

[7] Comeron, J.M. 1999. K-Estimator: calculation of the number of nucleotide substitutions per site and the confidence intervals. Bioinformatics. 15, 763-764.

[8] Gillespie, J.H. 1991. The causes of molecular evolution. Oxford University Press, Oxford, England.

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

[10] Hasegawa, M., H. Kishino, and T. Yano 1985. Dating the human-ape splitting by

a molecular clock of mitochondrial DNA. J. Mol. Evol. 22, 160-174.

[11] Hurst, L.D. 2002. The Ka/Ks ratio: diagnosing the form of sequence evolution. Trends in Genetics. 18, 486-487.

[12] Jukes, T.H., and C. R. Cantor 1969. Evolution of protein molecules, 21-123. In Munro, H.N. eds., Mammalian Protein Metabolism. Academic Press, New York.

[13] Kimura, M. 1980. A simple method for estimating evolutionary rate of base substitutions through comparative studies of nucleotide sequences. J. Mol. Evol. 16, 111-120.

[14] Kimura, M. 1983. The neutral theory of molecular evolution. Cambridge University Press, Cambridge, England.

[15] Li, W.H. 1993. Unbiased estimation of the Rates of synonymous and nonsynonymous substitution. J. Mol. Evol. 36, 96-99.

[16] Li, W.H. 1997. Molecular evolution. Sinauer Associates. Sunderland, Mass.

[17] Li, W.H., Wu, C.I. and Luo, C.C. 1985. A new method for estimating synonymous and nonsynonymous rates of nucleotide substitution considering the relative likelihood of nucleotide and codon changes. Mol. Biol. Evol. 2, 150-174.

[18] Muse, S.V. 1996. Estimating synonymous and nonsynonymous substitution rates. Mol. Biol. Evol. 13, 105-114.

[19] Nei, M. and Gojobori, T. 1986. Simple methods for estimating the numbers of synonymous and nonsynonymous nucleotide substitutions. Mol Biol Evol. 3, 418-426.

[20] Pamilo, P. and Bianchi, N.O. 1993. Evolution of the Zfx and Zfy genes: rates and interdependence between the genes. Mol. Biol. Evol. 10, 271-281.

[21] Posada, D. 2003 Using Modeltest and PAUP* to select a model of nucleotide substitution. In Baxevanis, A.D. (ed), Current Protocols in Bioinformatics. JohnWiley & Sons, New York, 6.5.1-6.5.14.

[22] Posada, D. and Buckley, T.R. 2004 Model Selection and Model Averaging in Phylogenetics: Advantages of Akaike Information Criterion and Bayesian Approaches over Likelihood Ratio Tests, Syst. Biol., 53, 793-808.

[23] Sullivan, J. and Joyce, P. 2005 Model Selection in Phylogenetics, Annual Review of Ecology, Evolution, and Systematics, 36, 445-466.

[24] Torrents, D., Suyama, M., Zdobnov, E. and Bork, P. 2003. A Genome-Wide Survey of Human Pseudogenes. Genome Res. 13, 2559-2567.

[25] Tzeng, Y.-H., Pan, R. and Li, W.-H. 2004. Comparison of Three Methods for Estimating Rates of Synonymous and Nonsynonymous Nucleotide Substitutions. Mol. Biol. Evol. 21, 2290-2298.

[26] Yang, Z. 1997. PAML: a program package for phylogenetic analysis by maximum likelihood. CABIOS. 13, 555-556.

[27] Yang, Z. and Nielsen, R. 2000. Estimating Synonymous and Nonsynonymous Substitution Rates Under Realistic Evolutionary Models. Mol Biol Evol. 17, 32-43.

[28] Zhang, Z., Li, J. and Yu, J. 2006 Computing Ka and Ks with a consideration of unequal transitional substitutions, BMC Evolutionary Biology, 6, 44.

[29] Wang, D.P., Wan, H.L., Zhang, S. and Yu, J. 2009. Gamma-MYN: a new algorithm for estimating Ka and Ks with consideration of variable substitution rates. Biol Direct, 4: 20.

[30] Wang, D., Zhang, S., He, F., Zhu, J., Hu, S. and Yu, J. 2009. How Do Variable Substitution Rates Influence Ka and Ks Calculations? Genomics, Proteomics & Bioinformatics. 7: 116-127