New semi-empirical codon substitution models based on principal component analysis of mammalian sequences

Master Thesis

Author(s):
Zoller, Stefan

Publication date:
2010

Permanent link:
https://doi.org/10.3929/ethz-a-006215736

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Master Thesis

New semi-empirical codon substitution models based on principal component analysis of Mammalian sequences

Stefan Zoller

6 April 2010 – 6 October 2010

Computational Biochemistry Research Group
ETH Zurich

Supervisors:
Dr. A. Schneider
Prof. Dr. G. Gonnet

Abstract

Parametric codon models in combination with Markov chains have been widely used in modeling molecular evolution on the level of codons. These models need – mainly for reasons of computational feasibility – to restrict themselves to a relatively small set of parameters, and the choice of parameters used has to be a well-founded one.

In this project, we tried to use a novel approach on identifying the most relevant parameters of codon models. The most relevant model parameters are those that explain the most of the variance of the sequence data. Principal component analysis (PCA) is a statistical method to identify orthogonal vectors in data space called principal components (PC) that point in the directions of the largest variance in the data. According to our working hypothesis, the most relevant codon model parameters should therefore emerge in the first PCs when applying PCA to the codon substitution matrix data. We were able to confirm this hypothesis on a large set of *Mammalia* data and on data sets that resulted from simulations.

PCs can be interpreted as linear combinations of known or unknown parameter vectors. We designed a test to confirm whether the obtained PCs correspond to previously considered parameters. Our test vectors encoded features like ω (ratio of non-synonymous to synonymous substitutions), κ (ratio of transitions to transversions), but also properties of amino acids such as weight or hydropathy. For the first few principal components, simple interpretations have been found. PC1 corresponds mostly to ω, while PC2 seems to be a linear combination of mostly κ and the amount of multi-nucleotide substitutions. In PC3, physico-chemical properties of the amino acids such as acidity and charge become more relevant.

We then used these results to build two new models of codon substitution. PCM+nC uses linear combinations of n principal components, and ECM+ω+ν (with ν denoting a factor for multi-nucleotide substitutions) has been constructed to test the relevance of ν compared to κ. Both models have been implemented in the `BEAST` and `BEAGLE` software packages and extensively tested against M0, a parametric codon model, and ECM+ω+2κ, a semi-parametric codon model. PCM+nC outperformed M0 and ECM+ω+2κ whenever the tested MSAs where not too distant from the original MSAs the PCA matrices where computed from. ECM+ω+ν showed to be very similar to ECM+ω+2κ, but was outperformed by the latter one as soon as the mean and variance of the κ distribution in the data set increased.

In conclusion, our simulations showed that important parameters of codon substitution models emerge in the first principal components. We recommend the consideration of PCM+nC when analyzing sequences of vertebrates; likelihood ratio test (LRT) can be used to identify the optimal n. Furthermore, multi-nucleotide substitutions appear to be important in substitution models, and this parameter should be considered when building new models of codon evolution.

Contents

1. Introduction

This project consisted of two parts. First, we applied principal components analysis (PCA) on a large set of substitution matrices estimated from codon alignments and analyzed the first principal components. Our hypothesis was: The parameters used in parametric codon models show a large effect on the likelihood of the data given the specific model (e.g. Doron-Faigenboim and Pupko, 2007; Kosiol et al., 2007). The most relevant model parameters explain the most variation in the data and should thus emerge as principal components (PCs) – or in linear combination of PCs. Additionally, other important parameters for codon models might be found by analysing unidentified PCs with high eigenvalues (Zoller and Schneider, 2010).

In the second part of the project, we then applied this knowledge by building two new models of codon evolution. One model exploits the fact that a linear combination of only the first few principal components can be used to approximate the given data points (Pearson, 1901; Hotelling, 1933). The second one builds upon our understanding of the first two principal components.

But first I want to discuss the basic background to understand the rest of the project. This includes some basic biology and molecular evolution, the modeling of molecular evolution with Markov chains, the notion of parametric codon models, Bayesian inference and PCA.

1.1. Basic Biology

1.1.1. Genetic Information

Every living cell contains one or more double helices of deoxyribonucleic acid (DNA) that encode the genetic information of the whole organism. These structures are built by two polymers (called strands) with backbones made of sugars and phosphate groups. One of four different bases, adenine (A), thymine (T), cytosine (C) or guanine (G), is attached to each sugar, and in this sequence of bases the genetic information is encoded (Watson and Crick, 1953). Adenine and guanine are called **purines**, thymine and cytosine are known as **pyrimidines**.

The double helix emerges by the pairing of these bases. A-T as well as C-G use hydrogen bonds to build stable pairs of molecules. The DNA strands are complementary; this leads to the binding of one strand to the other. Other physical properties are responsible for the twist in this bonded strands, thus building the double helix shape. Figure 1.1 shows the basic structure of the two strands, their complementarity and the base pairing.

1.1.2. The Classic Central Dogma of Molecular Biology

Because of the complementarity of the base pairs and the shape and structure of DNA, the double helix can easily be copied by a special molecular machine called DNA polymerase that unwinds the DNA strands and synthesizes a new strand of DNA on top of each already

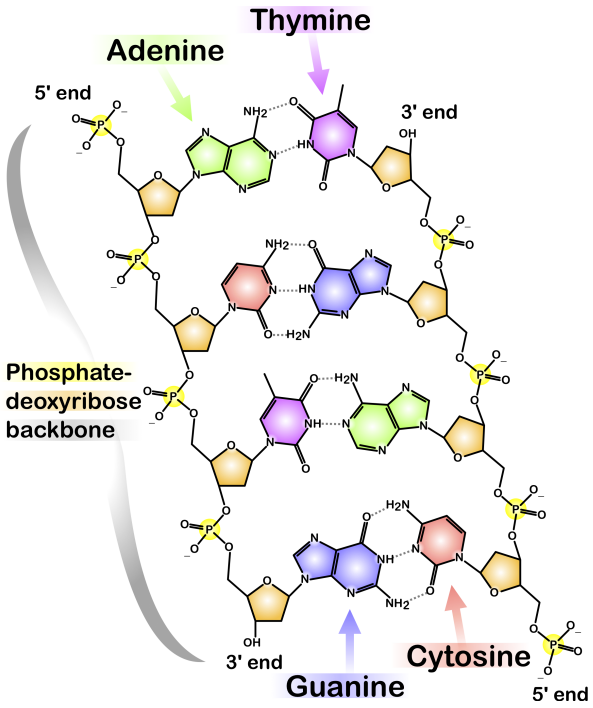


Figure 1.1. – Chemical structure of DNA (Wikipedia and Ball, 2009).

existing one. This copy mechanism leads to a new set of strands, which again form a double helix. The mechanism is called **replication** and denotes step one of the classic central dogma in molecular biology (figure 1.2, Crick, 1958). The name "dogma" has historical reasons; it is rather a rule that describes the flow of genetic information in living cells. There are exceptions to this rule, but we will not consider those in the subsequent considerations.

To read the genetic information on a certain position in the DNA, RNA polymerase, another molecular machine, copies the bases of interest into a polymer of ribonucleic acid (RNA). RNA is also built of four different nucleotides that are very similar to those of DNA. One difference is the use of ribose instead of deoxyribose; but the four bases attached to the sugar molecules are adenine, cytosine, guanine and (instead of thymine) uracil. The RNA polymerase works similar to the DNA polymerase: It unwinds the two DNA strands in a short segment of DNA and synthesizes a new RNA molecule (called messenger RNA or mRNA for short) on top of the DNA strand blueprint, using the same base pairing mechanism as before in the DNA itself. This step is called **transcription**.

In the last step, the mRNA molecule is processed by a ribosome, a third molecular machine. The ribosome reads the order of the nucleotides in the mRNA – like a tape deck reads a tape of data – and converts the sequence in a chain of amino acids. This is called **translation**. The amino acid chain, also called protein, folds into a specific shape that defines its function. Proteins are molecules of great importance in every living organism: They build support structures (e.g. both passive and active channels in the cells membranes), complex machines (e.g. the mentioned polymerases), allow chemical processes to happen (e.g. in the form of enzymes), are used for information distribution inside the cell or an organism (e.g. as receptor molecules at the outside of a cell's membrane) and much more.

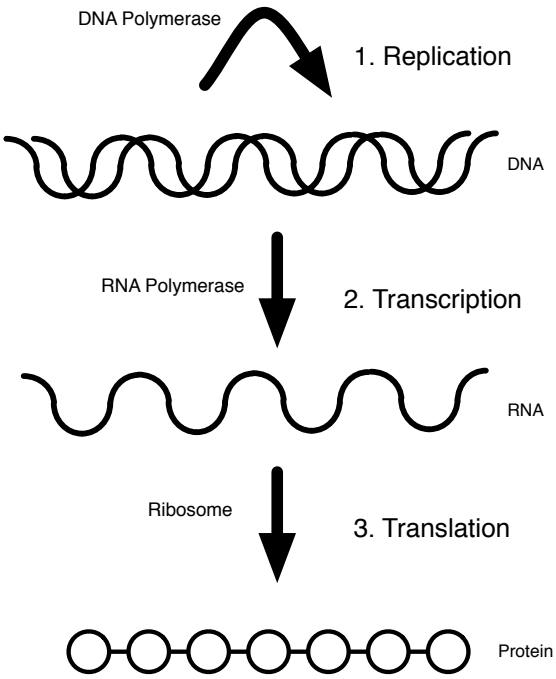


Figure 1.2. – Classic central dogma of molecular biology: 1) Replication of DNA with the help of DNA polymerase, 2) Transcription of DNA to RNA with RNA polymerase, 3) Translation of RNA to protein in the ribosome.

1.1.3. The Genetic Code

There are (normally) 20 different amino acids that are used to synthesize proteins. Using special adapter molecules, triplets of RNA nucleotides – a group that is called a **codon** – encode for one specific amino acid. A coding system with three positions and four different symbols can encode a total of $4^3 = 64$ states; therefore, some codons encode for the same amino acid. A three-character encoding scheme also introduces the concept of a **reading frame**; if the sequence loses one or two single nucleotides, the code will not just miss some single information but the reading frame will be shifted and thus all subsequent amino acids will change. One special codon is used to encode a start signal to help the RNA polymerase to find its starting position and set the correct reading frame when transcribing the DNA; three more codons encode a stop signal that tells the ribosome that it has reached the end of the useful information and shall stop the synthesis of the amino acid chain.

The encoding of the start and stop codons is not always the same but depends on the species; also, the encoding used by mitochondria (special compartments in most eukaryotic cells) differ slightly compared to the most common codes. Figure 1.3 shows the encoding of the RNA nucleotides to amino acids and includes the most common encodings for start and stop signals.

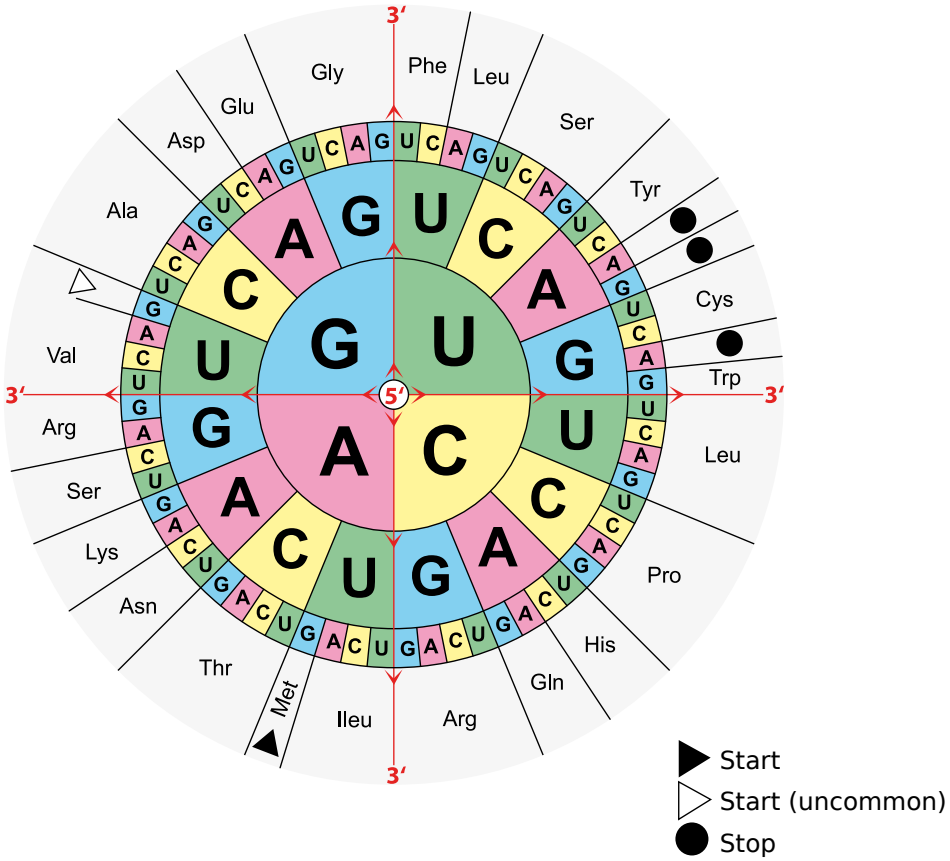


Figure 1.3. – Standard genetic code. The center nucleotides are in position one of the codons, the next ring of nucleotides provides position two, and the third ring denotes position three of the codon. For example, the codon `AGC` encodes the amino acid serine (Ser) (Wikipedia and Mouagip, 2010).

1.1.4. Molecular Evolution

The effectiveness (or even the functionality at all) of a protein is highly dependent on its structure, which in turn depends on the sequence of amino acids that build the protein. The order of those amino acids depends on the codons in the mRNA, which are determined by the codons in the DNA. Very small changes in the DNA can therefore have a strong influence on proteins and, in subsequence, on the whole organism. And these changes happen: Although the process of replication is very stable and highly secured with a variety of mechanisms such as proof reading and fixing of wrong pairings (e.g. Garmendia et al., 1992; Wood, 1996), errors still occur. Furthermore, mutagens such as chemical substances (Creppy et al., 1985), radiation (Witkin, 1976), viruses (Pilon et al., 1986) etc. can make nucleotides unstable or change their attached base. Such an error – independent on what its cause was – is called a (point-) **mutation**. A gene may accumulate changes through various evolutionary events: Point mutations, the insertion or deletion of a character, duplication of a part of the sequence, translocation of characters, recombination of genes, gene conversion or other mechanisms.

If the change of the genetic information did not substantially decrease the protein's ability to fulfill its function (i.e. the mutation is non-lethal), it might eventually be fixed in the

genome of the considered species depending on how much influence it has on the species fitness.

1.2. Modeling Codon Evolution

Models of molecular evolution (and thus also models of codon evolution) can be used for different tasks, including:

- Studying evolutionary patterns from molecular data
- Statistical testing of hypotheses
- Inferring conserved sites and those under positive selection to study the evolution of protein function
- Studying the evolution of gene families through phylogenetic inference
- Estimation of mutation rates, dating of speciation events

Since the process of molecular evolution is stochastic, we use probabilistic techniques to model it. The most successful model is the Markov model of evolutionary change.

1.2.1. The Standard Markov Model

A Markov chain is a stochastic process without memory. At each state of the chain, a probability determines how likely it is that a current state changes in a given time to some other state. Three assumptions are made:

Neighbour independence: At each position, the characters mutate randomly and independent of the other positions

Positional independence: The position of a character has no influence on the mutation probability

History independence: The model is memoryless; the probabilities only depend on the current state, not on the history of the character

A basic Markov model of nucleotide evolution models the characters at a position in the molecular sequence as states in its chain; mutations over time are modelled as a change of one state into another. The model uses a process generating matrix Q as the following:

$$Q = \begin{pmatrix} q_{TT} & q_{TC} & q_{TA} & q_{TG} \\ q_{CT} & q_{CC} & q_{CA} & q_{CG} \\ q_{AT} & q_{AC} & q_{AA} & q_{AG} \\ q_{GT} & q_{GC} & q_{GA} & q_{GG} \end{pmatrix}$$

The characters in this model A, C, G, T correspond to the four nucleotides in DNA. q_{ij} is an instantaneous rate from i to j. The process leaves state i at rate $-q_{ii} = \sum_{j \neq i} q_{ij}$.

Q determines the transition matrix $P(t)$:

$$\begin{aligned} \frac{dP(t)}{dt} &= P(t)Q \text{ with } P(0) = I \\ \Rightarrow P(t) &= e^{Qt} \end{aligned}$$

The General Time Reversible Model (GTR) gives a rate matrix that describes a reversible homogeneous Markov process Q with nine independent parameters (Tavaré, 1986):

$$Q = \begin{pmatrix} \cdot & a\pi_C & b\pi_A & c\pi_G \\ a\pi_T & \cdot & d\pi_A & e\pi_G \\ b\pi_T & d\pi_C & \cdot & f\pi_G \\ c\pi_T & e\pi_C & f\pi_A & \cdot \end{pmatrix} = \begin{pmatrix} \cdot & a & b & c \\ a & \cdot & d & e \\ b & d & \cdot & f \\ c & e & f & \cdot \end{pmatrix} \begin{pmatrix} \pi_T & 0 & 0 & 0 \\ 0 & \pi_C & 0 & 0 \\ 0 & 0 & \pi_A & 0 \\ 0 & 0 & 0 & \pi_G \end{pmatrix} = S \cdot \Pi$$

The diagonals of Q are given as $q_{ii} = -\sum_{i \neq j} q_{ij}$. π_i is the **frequency** of character i while $a \dots f$ are called **substitution rate parameters**. The model includes three assumptions: First, the process is supposed to be **homogeneous**. This means that the same character frequencies Π, the substitution rates S and therefore the rate matrix Q are constant across all sites. Second, the process is **stationary**. The same character frequencies and rates are constant through time. Third, the process is **reversible**, since $\pi_i q_{ij} = \pi_j q_{ji}\ \forall i \neq j$.

To extend this model to a model for codon evolution, we use 64 characters instead of just 4. This matrix is sometimes split into a 3×3 matrix for the start and stop codons and a 61×61 matrix for the "normal" codons. The models can be parametric, empirical or a combination of both.

1.2.2. Parametric Codon Models

Models that rely on parameters from observations and theoretical considerations are called parametric models. In terms of parametric codon models, these are often parameters that have a direct biological meaning such as κ or ω.

Not all point mutations lead to changes in the proteins. If we take a look at the codon table in figure 1.3, we see for example that `AAA` and `AAG` both encode for the amino acid lysine. If we have the sequence `...AAAAAA...`, and a single mutation occurs to change this to `...AAGAAA...`, then both codons still encode for the same amino acid as before, and the protein won't have changed its structure (and therefore its functionality). Such a mutation is called a **synonymous change**. A **non-synonymous change** would have been a mutation to `...AAUAAA...`; now the first codon encodes for asparagin, a different amino acid. Non-synonymous changes are often less likely to be fixed than synonymous changes, the substitution rate of non-synonymous changes is therefore often lower. The ratio of non-synonymous to synonymous changes is usually denoted as ω.

A non-synonymous mutation is either a **transition** – a purine changes into another purine ($A \leftrightarrow G$) or a pyrimidine into another pyrimidine ($C \leftrightarrow T$) – or a **transversion** – a purine changes into a pyrimidine or vice versa. About two thirds of all mutations on the nucleotide level are transitions (Collins and Jukes, 1994). Usually, κ denotes the rate of transitions over transversions.

Two well-known parametric codon models are M0 (Goldman and Yang, 1994) and the model of Gaut and Muse (Muse and Gaut, 1994). The model of Goldman and Yang was later refined by Yang (Yang et al., 2000). There, a whole family of parametric codon models has been described. The simplest one is called M0; its Q matrix is defined like this:

$$\forall i \neq j : Q_{ij} = \begin{cases} 0 & \text{if } i \text{ or } j \text{ is a stop codon or } i \to j \text{ require} > 1 \text{ nt substitutions} \\ \pi_j & \text{if } i \to j \text{ is a synonymous transversion} \\ \pi_j \kappa & \text{if } i \to j \text{ is a synonymous transition} \\ \pi_j \omega & \text{if } i \to j \text{ is a non-synonymous transversion} \\ \pi_j \kappa \omega & \text{if } i \to j \text{ is a non-synonymous transition} \end{cases}$$

There is only one non-synonymous/synonymous rate ratio ω and one rate of transitions over transversions, κ, for the whole model. Later models get more complicated: M7, for example, describes also an among-site variation of ω. The model of Gaut and Muse very much corresponds to M0 of Goldman and Yang. One difference is the use of two separate parameters α and β for synonymous and non-synonymous rates instead of using the ratio ω, but this is more a technical detail.

1.2.3. Empirical and Combined Models

Empirical models use large amount of sequence data to estimate the rate matrix Q. Models that use this empirical approach plus parameters (mostly with a biological meaning) are called combined models.

The first empirical codon model was CodonPAM, a log-odds matrix built by Schneider et al. to be used in multiple sequence alignment (Schneider et al., 2005). Later, combined codon models such as the Mechanistic and Empirical Combined Codon Model (MECCM) (Doron-Faigenboim and Pupko, 2007) or the Empirical Codon Model (ECM) family (Kosiol et al., 2007) have been developed. These models combine empirical matrices with well-known parameters as ω or κ.

Mechanistic and Empirical Combined Codon Model (MECCM)

The basis for the MECCM are empirical amino acid replacement matrices. These 20×20 matrices are then expanded into 61×61 codon matrices by splitting the single amino acid values into the corresponding number of codon values. Have a look at the value in the amino acid matrix A that denotes the change from methionine (Met) to aspartic acid (Asp): We can see in the codon table in figure 1.3 on page 8 that only one codon encodes for Met (AUG), while two codons encode for Asp (GAU and GAC). The single value in A that denotes Met→Asp needs to be split up into two values for AUG→GAU and AUG→GAC, respectively. For MECCM, this is done by weighting the codon changes with κ and the codon frequencies, and we end up with two new values in the codon table.

One basic assumption in this procedure is the validity of weighting the codon changes by using κ and the codon frequencies. Another assumption is that the sum of the rates between all the codons that encode two amino acids is also the replacement rate between those two amino acids (weighted by the relative rates of the amino acids and codons). Yang et al. first formulated this assumption in 1998 (Yang et al., 1998), and Doron-Faigenboim and Pupko included it in their MECCM (Doron-Faigenboim and Pupko, 2007). The weighting is formulated as follows:

$$\phi_i \cdot A_{ij} = \sum_{l:AA_l=i} \sum_{s:AA_s=j} \pi_l \cdot C_{ls}$$

The amino acids encoded by codons l and s are denoted as AA_l and AA_s, ϕ_i and π_l are the frequencies of amino acid i and codon l, and the substitution rate from codon l to codon

s is denoted as C_{ls}. The matrix C is calculated in the following way:

$$C_{ls} = \begin{cases} t_r \pi_s X_{AA_l AA_s} & l \text{ and } s \text{ differ by 1 transition} \\ t_v \pi_s X_{AA_l AA_s} & l \text{ and } s \text{ differ by 1 transversion} \\ t_{rr} \pi_s X_{AA_l AA_s} & l \text{ and } s \text{ differ by 2 transitions} \\ t_{vv} \pi_s X_{AA_l AA_s} & l \text{ and } s \text{ differ by 1 transversions} \\ t_{rv} \pi_s X_{AA_l AA_s} & l \text{ and } s \text{ differ by 1 transversion and 1 transition} \\ t_{sub} \pi_s X_{AA_l AA_s} & l \text{ and } s \text{ differ by 3 substitutions} \end{cases}$$

Here, we have six different weight factors t_r up to t_{sub} and some specific factors $X_{AA_i AA_j}$ that are calculated using the first equation. These specific factor are not free parameters but functions of the other parameters; the weight parameters are the only free model parameters which are optimized from the data.

The final substitution rate matrix C' is then constructed by multiplying ω to all non-synonymous substitutions:

$$C'_{ls} = \begin{cases} \omega C_{ls} & \text{non-synonymous substitutions} \\ C_{ls} & \text{synonymous substitutions} \end{cases}$$

Empirical Codon Model (ECM)

The ECM family of Kosiol et al. (2007) follows a similar approach as the MECCM. But instead of estimating amino acid rate matrices, it directly uses codon rate matrices as the basis for its exchange rate matrix. Kosiol et al. used 7332 protein families from the pandit database (Whelan et al., 2003) to estimate an initial codon substitution rate matrix I. The Q matrix itself is built as follows:

$$Q_{ls} = \begin{cases} 0 & \text{if } l \text{ or } s \text{ are stop codons} \\ I_{ls} \pi_s \kappa(l,s) & \text{if } l \to s \text{ is a synonymous change} \\ I_{ls} \pi_s \kappa(l,s) \omega & \text{if } l \to s \text{ is a non-synonymous change} \end{cases}$$

Again, π_s denotes the frequency of codon s; I_{ls} is the estimated exchangeability from the initial codon rate substitution matrix. $\kappa(l,s)$ denotes a term that represents the bias of transition/transversion between the two codons l and s and differs depending on which model of the ECM family is used; ω is ratio of non-synonymous to synonymous substitutions.

The ECM family includes different models. They are denoted as ECM+F+ω+$n\kappa$, where different values for n lead to the different model types. Kosiol et al. distinguish nine possible combinations of transitions and transversions, and they use these nine combinations to build different functions of $\kappa(l,s)$ to create the different models. The nine possibilities are:

1 nucleotide changes: (1ts, 0tv), (0ts, 1tv)

2 nucleotides change: (2ts, 0tv), (1ts, 1tv), (0ts, 2tv)

3 nucleotide change: (3ts, 0tv), (2ts, 1tv), (1ts, 2tv), (0ts, 3tv)

The most important models in the ECM family are then:

ECM+F+ω: $\kappa = 1$. This model assumes that the transition-transversion bias is already included in the initial rate matrix I.

ECM+F+ω+1κ(ts): $\kappa(i,j) = \kappa^{n_{ts}}$; n_{ts} denotes the number of transitions in the changes.

ECM+F+ω+1κ(tv): $\kappa(i,j) = \kappa^{n_{tv}}$; n_{tv} denotes the number of transversions in the changes.

ECM+F+ω+2κ: $\kappa(i,j) = \kappa_1^{n_{ts}} \kappa_2^{n_{tv}}$

ECM+F+ω+9κ: Each of the nine possible change combinations gets its own parameter κ.

Kosiol et al. recommend the consideration of the first four of these five models when using (and comparing) substitution models (Kosiol et al., 2007).

1.3. Estimating Likelihoods of Substitution Models

1.3.1. Maximum Likelihood Estimation

The likelihood function $L(\theta) = f_\theta(x_1, x_2, \cdots, x_n|\theta)$ denotes the likelihood of data samples $x_1, x_2, \cdots, x_n$ given a model with free parameters θ. We use likelihood values to compare different models under similar conditions. Maximum likelihood estimation (MLE) tries to find the $\hat{\theta}$ that maximizes $L(\hat{\theta})$. Assuming that all data is i.i.d., this can be done by maximizing $L(\theta) = \prod_{i=1}^{n} f_\theta(x_i|\theta)$. Since $\log L(\theta)$ is strictly monotonic, maximizing the likelihood is identical to maximizing the logarithm of the likelihood; but using logarithms leads to computationally more feasible terms. Therefore, we do not maximize the likelihood itself but its logarithm $l(\theta) = \sum_{i=1}^{n} \log f_\theta(x_i|\theta)$.

Maximum likelihood estimators are, under regular conditions, asymptotically unbiased, asymptotically efficient and consistent (Stuart et al., 1999).

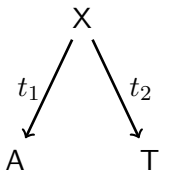


Figure 1.4. – Maximum likelihood estimation on trees: An simple example of a tree with two branches.

Evolutionary relations can be plotted as tree structures called **phylogenetic trees**. The outer nodes of the tree are called **leafs**, and the node on the origin (if there is one) is called **root**. The nodes denote different species, and the branches visualize one or more evolutionary events that changed the parent node to the child node. Figure 1.4 shows an example of such a phylogenetic tree with one root node and two leafs. We have a character A in the left leaf and a character T in the right one; the character at the root is not known and denoted with X. The branches have lengths t_1 and t_2. Felsenstein (1981) explained how to express the likelihood of such a tree given the sequence data and a model of evolution. To get the maximum likelihood, this likelihood function then has to be maximized, i.e. the set of θ that maximizes $l(\theta)$ has to be found. This can be done with various methods such as steepest descent search, Newton's method, BFGS and others (Yang, 2006).

1.3.2. Bayesian Inference

Bayesian inference uses Bayes' theorem to calculate the probability of a continuous parameter θ given a dataset D:

$$f(\theta|D) = \frac{f(D|\theta)f(\theta)}{f(D)} = \frac{f(D|\theta)f(\theta)}{\int f(D|\theta)f(\theta)d\theta}$$

$f(\theta|D)$ is called the **posterior distribution of** θ, $f(D|\theta)$ denotes the **likelihood of data**, $f(\theta)$ is known as the **prior distribution of** θ and $f(D)$ is the **marginal probability of data**.

Although Bayes' theorem was known for a long time, its application has computationally not been feasible for many problems since the calculation of the integrals needed is not trivial. Typically, we are interested in summaries of posterior distributions, such as the posterior expectations of some function $h(\theta)$. This requires the calculation of integrals like $\int h(\theta)f(\theta|D)d\theta$. With the advent of Markov chain Monte Carlo algorithms (MCMC, MC^3, Metropolis et al., 1953; Geyer, 1992) and the construction of fast computers, this problem has been alleviated. In each MCMC step, a new set θ of parameters is drawn and the new likelihood l_n is calculated. If θ leads to an improvement of l_n over the old value l_o, then it is kept; if l_n is lower than l_o, the parameter set can still be accepted with a probability of $\frac{l_n}{l_o}$. This sampling has to be done until θ converges.

1.4. Principal Components Analysis

Principal Components Analysis (PCA, Pearson, 1901; Hotelling, 1933) is a method that can be used to reduce the dimensionality of a data set (see e.g. Jackson, 1991; Bishop et al., 2006). Applying PCA allows us to find linear combinations of the parameters involved (called **principal components** or PCs) that are orthogonal to each other and point in the directions of the largest variances in the data; technically, these are the eigenvectors of the covariance matrix C of the data set. PCs can be ordered by their explanatory power of the variance in the data – this explanatory power is directly proportional to the corresponding eigenvalues of the PCs. Linear combinations of the PCs describe all points in the given data set. By choosing only the PCs with the highest eigenvalues for building these linear combinations, reasonable approximations of the original data points can be found. This leads to the aforementioned dimensionality reduction. Note that the algebraic sign of PCs is arbitrary.

Figure 1.5 shows a two-dimensional example of PCA. The principal components can also be understood as a new Cartesian coordinate system that has been rotated to fit to the directions of the largest variances in the data set.

PCA assumes:

Important dynamics have large variances: This only works when the signal in the data is larger than the noise. If the variance introduced by noise is larger than the variance in the data without any noise, PCA will not lead to useful results.

Linearity: PCA assumes that all data points can be described as linear combinations of certain vectors. Without the need for clustering via PCA, this should not pose an issue when dealing with codon data.

Normal distribution of parameters: By using the eigenvectors of the covariance matrix C, Gaussian distributions are assumed. If the parameters do not follow Gaussian distribu-

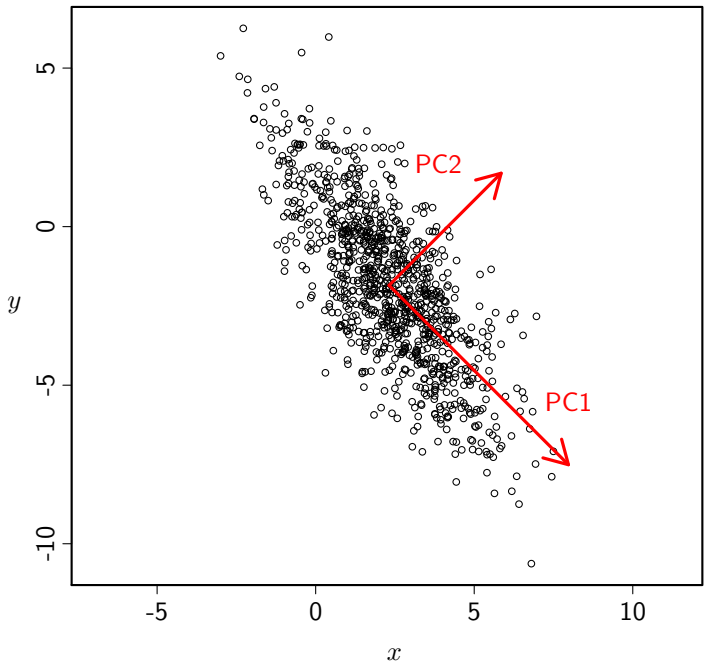


Figure 1.5. – PCA of a multivariate Gaussian distribution centered at $(2.3, -1.85)$ with a standard deviation of 3 in roughly the $(0.71, -0.71)$ direction and of 1 in the orthogonal direction. The red arrows indicate the principal components PC1 and PC2.

tions, they will be decorrelated, but not necessarily be independent after PCA. In this case, PCA will not necessarily find an optimal solution.

2. Methods

2.1. Two New Codon Models

2.1.1. The PCA Codon Model

The PCA Codon Model (PCM) builds on a rate matrix that is derived from a linear combination of principal components that were empirically found to describe important features of codon evolution. It corresponds to reversing the PCA with fewer components but free coefficients. Three data sets are needed from the original PCA: The principal components (denoted as C^i for the ith component), the scaling factors S and the means of the parameters M. `R` returns this information in the `prcomp` object that results from the `prcomp()` function used for the PCA as described later in section 2.2: Given a result object `o`, `o$center` contains the means, `o$scale` contains the scaling factors, and `o$rotation` contains the eigenvectors i.e. the principal components. The chosen number of principal components n and a vector p of size n containing free coefficients for the linear combination are used as input parameters of the model. Every element R_{uv} in the symmetric relative rate matrix (also called exchangeability matrix) corresponds to one parameter in the result of the PCA. These elements are calculated by building a linear combination of a number of principal components $C^1 \dots C^n$ and the free coefficients $p_1 \dots p_n$ and are then scaled back by the scaling factor S_{uv} to the original variance of the parameter in question. At last, the mean value M_{uv} is added to this combination in order to decenter the resulting matrix. Since some entries in the PCs are negative, it is possible to get a negative entry in the R matrix, which is not valid for later processing. Negative entries are therefore set to 0.0 after the calculation of R. For the first n PCs, this can also be written as:

$$R_{uv} = \begin{cases} 0 & \text{if } u \text{ or } v \text{ is a stop codon} \\ & \text{or if } M_{uv} + S_{uv} \sum_i^n p_i C_{uv}^i < 0 \\ M_{uv} + S_{uv} \sum_i^n p_i C_{uv}^i & \text{else} \end{cases}$$

We use PCM+nC to denote a PCA Codon Model with n principal components and an additional $+F$ if the codon frequencies are estimated from the data.

2.1.2. ECM+ω+ν

The second new model was created after the interpretation of the first few principal components of the *Mammalia* data set (see 3.2.2 for more). It bases on a given initial rate matrix – hence the ECM – and uses the well known parameter ω, the ratio of non-synonymous to synonymous substitutions. The additional parameter ν serves as a scalar for multi-nucleotide substitutions. The term for calculating the relative rates R from the initial rate matrix I

becomes therefore:

$$R_{uv} = \begin{cases} 0 & \text{if } u \text{ or } v \text{ is a stop codon} \\ I_{uv} & \text{if } u \to v \text{ is a synonymous single-nucleotide change} \\ I_{iv}\omega & \text{if } u \to v \text{ is a non-synonymous single-nucleotide change} \\ I_{uv}\nu & \text{if } u \to v \text{ is a synonymous multi-nucleotide change} \\ I_{uv}\omega\nu & \text{if } u \to v \text{ is a non-synonymous multi-nucleotide change} \end{cases}$$

2.2. Principal Component Analysis of Ortholog *Mammalia* MSAs

2.2.1. Alignments

The orthologous matrix (OMA) project (Dessimoz et al., 2005) contains groups of orthologs that provided the sequences for our *Mammalia* data set. Using orthologs is not necessary for PCA, but searching for homologs is a time consuming task, and the OMA project as a convenient source of pre-computed orthologs alleviates this problem. The data set used was the *Mammalia* set from 9 June 2009 which contains 62,156 groups covering 33 species. We chose the *Mammalia* data mainly for two reasons: First, preliminary tests with *Actinobacteria* showed that large amount of codon bias, high mutation rates and large distances within the groups lead to a higher noise level in the data set. *Mammalia* data does not show these properties. Second, the amount of groups included should allow for computational feasibility.

To ensure the quality of the data needed to be able to estimate the 1830 parameters, we applied some filters. If a coding sequence included more than one percent unknown bases, it was discarded. If a group of orthologs contained less than 6 sequences, it was excluded; if it contained more than 20 sequences, the 20 longest ones were kept. Since most of the OMA groups include less than 6 sequences (Roth et al., 2008), this step in the filtering process eliminated most of the candidate MSAs. The protein sequences of each group were then aligned with `Mafft` (Katoh et al., 2005), and by mapping the corresponding coding DNA sequences to the aligned protein sequences, a DNA MSA was produced. We then removed every position in the MSAs where a gap or a codon with unknown bases occurred. If this lead to an MSA that was shorter than 333 codons ($\approx$ 1000 bases), the alignment was excluded. Identical sequences in MSAs have been removed as well. This lead to 3666 MSAs with at least 333 aligned codons and 6 sequences.

Constructing the guide trees that were required for the matrix estimation was accomplished by using the least-squares distance tree method that is included in `Darwin` (Gonnet et al., 2000) and uses pairwise CodonPAM distance estimates (Schneider et al., 2005).

2.2.2. Substitution Rates Estimates

It is not easy to estimate a 61×61 substitution matrix from a single MSA. There are only a few observable substitutions in each MSA, and there are many parameters to estimate. Without using restrictions, the variances on the estimations can get very high. We therefore chose to fix the codon frequencies as constants. By assuming a time-reversible model, the number of free variables was nearly halved and we ended up with 1830 parameters. These are the off-diagonal values of the (symmetric) exchangeability matrix. Together with the equilibrium frequencies – which have been fixed –, these values define the substitution rate matrix.

In order to perform the PCA, estimations of all 1830 parameters from each of the 3666 used MSAs were needed. Furthermore, the codon frequencies had to be estimated from all 3666 MSAs simultaneously. We used the expectation maximization (EM) algorithm implemented in `XRate` (Klosterman et al., 2006) with a customized grammar file to estimate the codon frequencies; the first run lead to an initial substitution matrix calculated for all MSAs. Next, every MSA was again run through `XRate` with the initial substitution rate matrix as a starting point for estimating the individual matrices. In these estimations, the grammar file instructed `XRate` to keep the frequencies fixed at the initial frequencies that have been estimated in the first run.

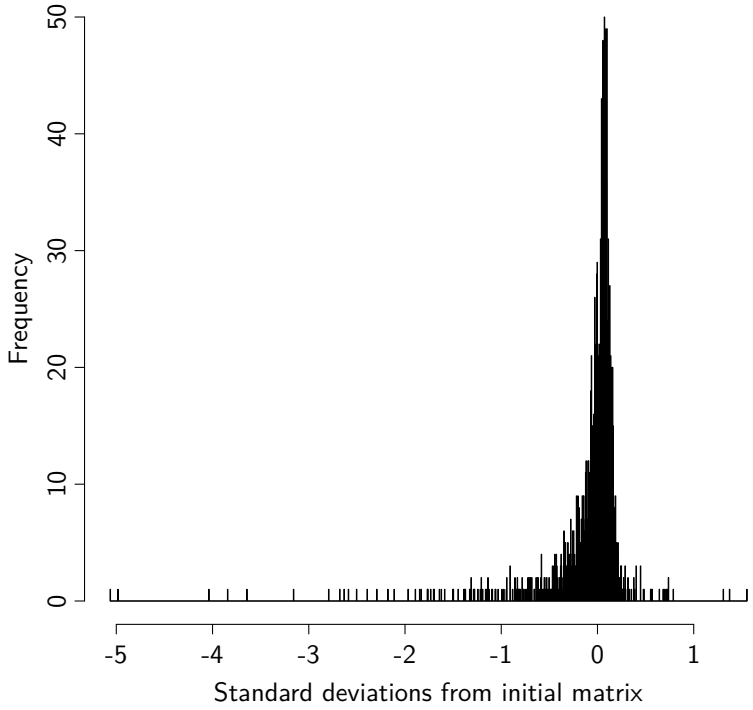


Figure 2.1. – Histogram showing the differences between the initial rate matrix and the means of the parameters estimated from single MSAs. The values are divided by the standard deviation of the corresponding parameters.

We then compared the 3666 separate matrices to the initial rate matrix that included all available sequence data. Since the amount of data used for the initial matrix is very high, its estimation is more trustworthy than that of one single MSA; it can therefore be used as a frame of reference for the single estimates. The computation of means and standard deviations for each of the 1830 parameters over all 3666 single MSAs was done in `R`, resulting in one matrix of 1830 mean values and another corresponding matrix of 1830 standard deviations. Next, the mean matrix was subtracted from the initial rate matrix, and each result was divided by its corresponding standard deviation. Figure 2.1 shows a histogram with the distribution of these standardized differences. Only 15 ($\approx 0.8\%$) of the values are larger than 2.0, more than 95% are smaller than 1.0. The parameter estimates from the single MSAs seem therefore to

be in the expected range of about one standard deviation.

As a result of all these steps, we ended up with a matrix containing 3666 rows that denote the exchangeability matrices of the single MSAs, each with a length of 1830 estimated parameters.

2.2.3. Principal Component Analysis

This resulting data matrix was processed by the `prcomp()` function of `R`. The function centers the data and then scales it to a variance of 1.0 on each parameter before it applies the PCA. The output consists – among other information – of the principal components (PCs), the corresponding eigenvalues and the data on centering and scaling. We sorted the PCs according to their eigenvalues and represented them as matrices with positive and negative values. A `ruby` script modified the PCs to be a valid input for a modified version of `visualizeRates.pl` that ships with `XRate` which was then used to create bubble plots of each interesting PC. Since the matrix that is represented by the PCs is symmetric, we used the upper triangular matrix to show only the positive values and the lower triangular matrix to show the negative ones. This lead to plots where each triangular matrix displays a group of parameters that are anticorrelated with the parameters of the other group, using the diagonal of the matrix as a separation line. The size of a bubble relates to the value of the corresponding parameter: The larger the bubble, the higher its influence in this particular PC. The bubbles are color coded for different features such as the number of nucleotides changed or the amount of transitions and transversion per substitution. The algebraic sign of a PC is arbitrary, which means that mirroring a bubble plot on its diagonal does not change the meaning of the PC; we exploited this in a few plots by mirroring the axes to make the PCs easier to compare visually.

The initial 3666×1830 data matrix resulted in 1830 individual PCs. Since their importance is proportional to the corresponding eigenvalues, the PC with the highest eigenvalue has the highest influence on the variance in the data. PCA is also a method for dimensionality reduction, and only the first few PCs have to be used to explain most of the data in observation; but what is the cutoff point where one can discard PCs with lower eigenvalues without losing too much information? This problem is also known as the stopping problem in PCA.

How many Principal Components?

There are several methods to solve the question for the cutoff position (Jackson, 1993). We chose to apply two of them: The visual analysis of the scree plot and the method of bootstrapped eigenvalues and eigenvectors.

The scree plots were drawn using the standard `R` functions. The second method was also applied in `R`; the initial data matrix was bootstrapped 250 times and PCA was performed on every bootstrap sample. The eigenvalues were stored in a new matrix and means, minima, maxima and 95% confidence intervals were calculated.

Dealing with Noise

Random noise in the data is a nuisance because it makes the PCA results less reliable. Methods that can reduce this unwanted influence can therefore be very useful. One approach is the filtering of the original data. We tried different techniques to accomplish this goal: Outlier detection, clustering, weighted PCA and the transformation of the data before the PCA.

The rational behind outlier detection is that maybe very few outliers distort the signal in the data set. Discarding these outliers could lead to a better detectable signal. An outlier in a parameter vector was defined being a value that has a reasonable high distance from the rest of the distribution. To count a whole single MSA (i.e. the estimations of a set of 1830 parameter values) as an outlier, not only one parameter but several of them needed to be identified as outliers themselves. The detection of these parameter outliers has been done manually as well as mechanically. For the manual method, plots of the values for every of the 1830 parameters were inspected and outliers have been marked and then sorted out. This is clearly a very subjective and tedious method, and therefore a mechanical approach has been tested as well. For every parameter vector with 3666 measurement values, all values that were larger than a given distance from the mean of the whole vector have been marked. In a second step, all measurements (vectors of 1830 parameters) where the amount of markers exceeded a threshold have been discarded from the dataset. PCA was then applied on the remaining data, and the eigenvalues and PCs were compared to the original data set.

Clustering denotes the aggregation of several points with similar characteristics. The idea for using this method for noise reduction is the following: Given the assumption that most of the noise results from small differences in otherwise very similar data points, clustering can smooth these differences and may help to show a clearer signal in the data. We treated each MSA estimation as a point in a 1830 dimensional space and calculated a distance matrix for all the 3666 data points. Then k-means clustering was used to generate 1222 cluster of data points. Each cluster corresponded to a group of MSAs; the largest cluster included 15 MSAs, the smallest 422 clusters included only one MSA. With `XRate`, a new parameter vector from each of those groups was estimated jointly from all MSAs belonging to the cluster. These parameter vectors were combined in a new cluster data set, which could then be used for weighted PCA. The weights of the vectors were given by the size of the corresponding cluster.

Transformation of the data before the PCA is another method to optimise PCA. The idea is that the data might not be in an "optimal" state for the method. Maybe numerical errors introduce new noise that would not have been present if the data would have been transformed before, maybe the parameters do not follow a normal distribution. Typical transformations include taking the logarithm of the original values; this is what we did as well. Since PCA works best on data that has a Gaussian distribution and taking the logarithm can help to come closer to this ideal distribution, it was reasonable to try this transformation on the data. Zero values in the original data set had first to be transformed into a "lowest threshold" of $1e-20$.

2.2.4. Simulations

Our working hypothesis about finding the most relevant parameters in codon models was that those relevant parameters show up as principal components – or, at least, as linear combinations of the PCs. To test this hypothesis, several simulations were performed next to the analysis of the OMA MSAs.

The aim of the first simulation was to test again whether the results from the PCA of the *Mammalia* data set were within reasonable ranges. We compared the original data set to a data set that should have emerged – at least theoretically – the same empirical parameter distribution as the ones measured in the original data. Estimations of ω and κ from the 3666 single MSAs with `codeml` (Yang, 2007) have been taken and `evolver` (Yang, 2007) was used to generate new MSAs with the same values for ω, κ, the corresponding trees from `Darwin`,

the same sequence lengths and the same number of species under an M0 model. The resulting new 3666 MSAs have then undergone the same procedure as the original ones, leading to a new exchangeability rate matrix for this first simulation.

The second and third simulation should help to test whether the aforementioned working hypothesis was correct or not. As in the first simulation, new MSAs were created using the exact same characteristics as before; but in the second simulation, we fixed κ at 1.0 whereas in the third simulation, κ was drawn from a normal distribution with mean 6 and variance 2.5, where values below 0.2 were discarded. This lead to a resulting κ distribution that had a much higher variance than the measured one; given the hypothesis was true and this parameter really influences the data substantially, such strong changes in κ should be visible in the first principle components.

2.3. BEAST and BEAGLE

`BEAST` (Drummond and Rambaut, 2007) is an application written in Java that uses Bayesian MCMC techniques to analyse molecular sequences. Some of the models and features in `BEAST`, taken from their website at `http://beast.bio.ed.ac.uk/What_can_BEAST_do%3F`, include:

- Constant rate molecular clock models
- Variable rate (relaxed) molecular clock models
- Local clock molecular models
- Flexible model specification
- Range of substitution models
- Flexible choice of priors on parameters
- Reconstruction of phylogenies

Its modular design allows a fast implementation of new models.

2.3.1. BEAGLE and Graphical Processing Units

`BEAGLE` is a library that can be used by `BEAST` for high-performance likelihood calculations (Suchard and Rambaut, 2009). It does not only speed up the calculations on one or more CPUs, but is also able to use Graphical Processing Units (GPUs, with support for `CUDA`, Nvidia Company, 2007) as computing platforms.

All of the model calculations were done on Brutus (`https://www1.ethz.ch/id/services/list/comp_zentral/cluster`), a high-performance cluster at ETHZ. Brutus consists not only of thousands of CPUs, but has also four experimental nodes available that each contain 12 CPUs with 6 GPUs attached to them. Each GPU contains 240 cores at 1.44GHz and 4GB of VRAM, reaching a peak performance of 1.04 TF in single precision mode.

2.4. Implementing Models for Codon Evolution

To implement new substitution models in `BEAST`, at least three tasks have to be performed: One needs to define some new XML elements for the configuration file and write a parser class for these elements, the model class itself has to be implemented, and the new model has to be included in the existing framework. If the model should also be used by `BEAGLE`, the parser and the model classes have to be copied and slightly changed. I will describe very shortly how these tasks can be done in general; the actual implementation of the two new models will be discussed later in detail.

New models can easily be included in the framework; there is one central place in `src/dr/app/beast/development_parsers.properties` where the new parser classes have to be added at the end of the file.

Defining the new XML elements might require some consideration about the design, but is also very straightforward. A good way to start is to have a look at the syntax of an existing model and then to adapt those elements to the needs of the new model. Similarly, the adaption of the copied code of an existing parser (i.e. the M0 XML parser) is a fast method to implement the parser class of a new model.

After reading and parsing the configuration, the parser class instantiates the new model class which will be implemented as subclass of an existing abstract model class. For codon substition models, this parent class is `AbstractCodonModel`. To implement the model, two things have to be accomplished: In the constructor, the free parameters that are going to be changed by the MCMC object have to be registered in the framework, and an abstract function declared in the parent class, `setupRelativeRates()`, needs to be written.

Generally, all codon models rely on a rate matrix Q where each element that denotes a possible change from codon A to codon B is a product of the frequency of codon B, π_B, times a relative rate factor $R_{A\to B}$. So two matrices are actually needed: A matrix Π with codon frequencies and a matrix R with the rate factors. In `BEAST`, there is already a model for codon frequencies present in the framework. The frequency model object sets up a frequency matrix Π, can be created by the parser and be given to the new model object in the constructor. This leaves only the rate matrix to be defined, and this is exactly what the method `setupRelativeRates()` is supposed to do. It provides the model with a relative rate matrix R; everything else (calculating Q from Π and R and then $M(t) = e^{Qt}$) is already implemented in the parent classes.

2.4.1. Empirical Models of Kosiol et al.

The Empirical Codon Model family (ECM) (Kosiol et al., 2007) consists (seen as a `BEAST` model programmer) of an empirically estimated matrix with fixed rates that are modified by some free parameters such as ω and one or more different values for κ. The XML syntax has to reflect this. A typical model definition in the configuration XML looks like the example in listing 2.1

Line 1 defines a model of the ECM family. It contains an identifier *id*, a value *ecmDataDir* for the directory where the files with the initial rates reside, and two more values *ecmRateFile* and *ecmFreqFile* with the filenames for the rates and the codon frequencies. The lines 2-7 define a parameter ω and a parameter κ_{ts-tv}. Both parameters get identifiers and initial values to start the MCMC with; for ω, this is one `double` value, for κ, there are two `double` values given, which means that this configuration describes an instance of ECM+ω + 2κ.

Listing 2.1 – Example of the definition of an ECM+F+omega+2k model in `BEAST`

```
1  <empiricalCodonModel id="ecm" geneticCode="universal" ecmDataDir="ecmdata" ecmRateFile="
       rates_k_unrest.csv" ecmFreqFile="freqs_k_unrest.csv">
2    <omega>
3      <parameter id="omega" value="0.1"/>
4    </omega>
5    <kappaTsTv>
6      <parameter id="kappatstv" value="1.0 1.0"/>
7    </kappaTsTv>
8    <frequencyModel id="freqmodel" dataType="codon-universal">
9        <alignment idref="codonAlignment"/>
10       <frequencies>
11           <parameter id="codonfrequencies" dimension="61"/>
12       </frequencies>
13   </frequencyModel>
14 </empiricalCodonModel>
```

Furthermore, the definition of the frequency model on the lines 8-13 define that the frequencies shall not be read from the given file but merely estimated from the codon alignment itself, thus creating a ECM+$F + \omega + 2\kappa$ model.

This example shows that our implementation of the ECM model is in fact an implementation of a wide range of the ECM family (Kosiol et al., 2007). The model object can distinguish between ECM+ω, ECM+$\omega + 2\kappa$, ECM+$\omega + 9\kappa$ and their corresponding types with estimated frequencies. The models ECM+ω+$1\kappa(ts)$ and ECM+ω+$1\kappa(tv)$ can be emulated by fixing one of the two κ parameters to 1.0 and setting the appropriate boundaries on the corresponding dimension of the κ_{ts-tv} parameter.

To manage the different initial matrices like the initial rate matrix I and the frequencies F, a new object `EmpiricalRateMatrix` was written that parses the corresponding data files and stores arrays with the given values. The parser instantiates this object while reading the XML configuration and passes it to the model object; it also creates the frequency object and decides if the codon frequencies shall be estimated from the given alignment or if they shall be taken from the frequency file.

2.4.2. Principal Codon Model Implementation

PCM expects as input a number of PCs to use and initial data files with mean values, PCs and scalar factors. The names of the data files are hardwired in the model, only the correct directory has to be specified in the XML configuration. This configuration is therefore rather simple (see listing 2.2): Line 1 defines the PCM model with *pcadata* as its data directory, and *pcaType* declares it of type *mammalia*, which is the only type currently provided. In lines 2-4, a parameter *pcaDimension* is configured. Its dimensionality defines the number of PCs to be used by the model. The *frequencies* tag is the same as in the ECM configuration. The complete example in the listing defines therefore PCM+F+2C.

As with the ECM model, a separate class `PCARateMatrixMammalia` serves as a container for all the data matrices needed by the model. The names of the data files are included as static strings and cannot be changed by the end user; however, the directory in which the data files are stored is defined in the XML configuration. `PCARateMatrixMammalia` expects to

Listing 2.2 – Example of the definition of PCM+2C in `BEAST`

```
<pcaCodonModel id="pca" geneticCode="universal" pcaType="mammalia" pcaDataDir="pcadata">
    <pcaDimension>
      <parameter id="pcaDimension" dimension="2"/>
    </pcaDimension>
    <frequencyModel id="freqmodel" dataType="codon-universal">
    <frequencyModel id="freqmodel" dataType="codon-universal">
        <alignment idref="codonAlignment"/>
        <frequencies>
            <parameter id="codonfrequencies" dimension="61"/>
        </frequencies>
    </frequencyModel>
</frequencyModel>
```

find at least three different files: `means.csv` containing the mean values of the PCA, `pcs.csv` containing the principal components and `scales.csv` with the scalars for each parameter. It is also possible to include a list of frequencies in a file `freqs.csv`, but this is not necessary, since the frequencies can also be estimated from the MSA. The file format is – as the name suggests – "comma separated values": Each line includes one vector of 1830 parameters, separated with commas. `means.csv` consists therefore of one line with 1830 values, `pcs.csv` will be a file with 1830 lines, each containing 1830 values. If the files are not found or do not contain the correct data format, `PCARateMatrixMammalia` will throw a `FileError`-Exception and terminate the application with an appropriate error message.

Extending the model with a rate matrix object that uses more or less parameters is straightforward. `PCARateMatrixMammalia` is a subclass of `AbstractPCARateMatrix`; its parent class includes all the functions that are needed by the `PCACodonModel` class. By creating a new subclass of `AbstractPCARateMatrix`, other rate matrices can be included in PCM; the *type* parameter in the XML configuration can be used to distinguish between different model types. Currently, only our `PCARateMatrixMammalia` is included in `BEAST`.

`PCACodonModel` calculates its substitution rate matrix R by applying the rules given in section 2.1.1.

2.4.3. Using BEAST models in BEAGLE

The changes to use these new `BEAST` models with `BEAGLE` are very minor. The parser and rate matrix objects are identical to the corresponding `BEAST` objects and will not be discussed any further. The model objects cannot be subclasses of `AbstractCodonModel`, because this class does not exist in `BEAGLE`; instead, they have to extend `BaseSubstitutionModel`. This parent class has – as far as our models are concerned – only one important difference to `AbstractCodonModel` in `BEAST`: The method `setupRelativeRates()` changes to `setupRelativeRates(double[] rates)` where the rates are not member variables of the model classes anymore but passed in by the framework.

2.5. Comparison of Different Models

2.5.1. Bayes Factor and Marginal Likelihood

Model comparison in a Bayesian framework can be done by calculating and interpreting the Bayes factor (BF, Kass and Raftery, 1995). The BF can be formulated as:

$$\frac{p(M_1|x)}{p(M_2|x)} = \frac{p(M_1)}{p(M_2)} \frac{p(x|M_1)}{p(x|M_2)} = \frac{p(M_1)}{p(M_2)} \frac{\int p(x|\theta, M_1)p(\theta|M_1)d\theta}{\int p(x|\theta, M_2)p(\theta|M_2)d\theta}$$

The terms $\int p(x|\theta, M)p(\theta|M)d\theta$ denote the **marginal likelihood** of a model M with parameters θ and data x. The BF does not depend on the parameters used in each model or on the models' degrees of freedom but only on the probability of the model by considering all possible parameter values. Given two equally performing models, BF will decide in favor of the simpler model, i.e. the one with less free parameters.

The marginal likelihoods cannot be calculated exactly in `BEAST`, because it would involve one single MCMC averaging over both models. But `BEAST` can approximate the marginal likelihood of one model at a time (Clifford, 1994; Suchard et al., 2001). Taking the difference of two of these approximate marginal likelihoods in log space results in a good approximation for the log BF of those two models. `BEAST` includes this functionality since version 1.5, and we used it to compare the performances of M0+F, PCM+F+2C, PCM+F+8C, ECM+F+ω+2κ and ECM+F+ω+ν.

Kass and Raftery (1995) gave guidelines for the interpretation of the BF which we used in our comparison. According to the authors, log BF values between 1 and 3 are considered as "not worth more than a bare mention"; a log BF value between 3 and 20 can be interpreted as positive evidence, values between 20 and 150 are considered as strong and values beyond 150 as very strong evidence. Since the concept of BF roots in Bayesian statistics, it does not include p-values which are common in tests rooting in the frequentist framework.

M0+F was included to give a base reference. Kosiol et al. compared their ECM family also to M0 and M7, but they used `codeml` from the `PAML` package (Kosiol et al., 2007). This application does not use Bayesian inference to get its likelihood values, and the results are therefore not directly comparable to the ones `BEAST` gives. Furthermore, the possibilities in the configuration of `BEAST` are much more extensive than the ones in `PAML`. By using the same MSAs from the pandit database and two of the same models (M0 and ECM+F+ω+2κ) as Kosiol et al. did, we configured `BEAST` such that the measurements reflect the results of `codeml` in the best possible way. The M0 model was already implemented in `BEAST`, and by comparing first the results of M0 and then the results of the ECM family with the values given by Kosiol et al. (2007), we were able to verify that our implementation of the ECM family subset was correct.

Only ECM+F+ω+2κ was considered from the family of traditional ECM models. Even with the use of `BEAGLE` on Brutus, the calculations were very time consuming, and we had to restrict ourselves to one single member of the whole model group. ECM+F+ω+2κ was chosen because Kosiol et al. (2007) recommended four models of the family for consideration; among these four models, ECM+F+ω+2κ performed best. The pandit MSAs have been estimated with the original initial rate matrix published by Kosiol et al. which was also estimated from pandit sequences. The PCM has its roots in the OMA project MSAs. To estimate a possible data bias, we also used the ECM models – ECM+F+ω+2κ as well as our ECM+F+ω+ν – in conjunction with the rate matrix gained from an early step of the PCA process with the OMA data sets to be able to truly compare the models and not only their initial rate matrices.

2.5.2. Alignments

Kosiol et al. (2007) used 200 MSAs from the Pandit database (Whelan et al., 2003) when they evaluated their ECM model family. We extracted those 200 MSAs and their corresponding trees from the database; 17 MSAs with less than 4 species were discarded. Next, 200 new *Mammalia* MSAs were chosen from the OMA project (version of 24 Mai 2010) and compared to the ones already gathered for the PCA. Only one of the sequences was identical; we removed it from the data set. Furthermore, 15 MSAs each were taken from the OMA groups *Archaea* (version of 6 Mai 2010) and *Cyanobacteria* (4 December 2010), 14 MSAs from *Other Eukaryota* (16 March 2010) and 9 MSAs from the group *Vertebrata* (9 January 2010). These groups have been chosen as test cases because their in-group distances differ from the ones in the *Mammalia* data set. All of the OMA MSAs and the corresponding trees were constructed in the same way as the ones described above that were used for the PCA. We then used a `ruby` script to convert the different MSA and tree formats into the XML format used by `BEAST`.

By recomputing the likelihood results of Kosiol et al. (2007) with `codeml`, we realized that this application uses an internal upper boundary for branch lengths. We used version 4.2a of the program, but this is also present in all earlier versions at least up to version 3.14. The boundary is hard-wired in the source code and limits the branches at a length of 50.000. We processed all MSAs with `codeml` and parsed its tree output for branches with length 50.000; 94 Pandit MSAs (from a total of 183) showed these truncated trees. In the other data sets, no MSA included a branch length estimation of 50.000.

The MSAs taken from the Pandit database include many different species. While the species in the OMA MSAs are relatively close to each other (all *Mammalia*, or all *Vertebrata*), Pandit does not make this distinction but keeps its focus on including every species where a given gene has been sequenced. This leads to possibly high distances between sequences within an MSA, which makes estimating model parameters and branch lengths harder.

We used `BEAST` to estimate the branch rates and likelihood values of all MSAs. Without a prior on the branch rates, some parameter estimations in `BEAST` grew uncontrollably without any reasonable limits. We therefore decided to introduce a similar restriction as in `codeml` by using a uniform prior that limited branch rates to 100.0. Branch rates above this value were considered as unrealistic.

2.5.3. BEAST Configuration

The actual model evaluation is defined in the `BEAST` configuration, an XML file that can either be constructed with the GUI tool `BEAUti` (included in the `BEAST` distribution) or written manually in any text editor. We did not include the ECM and PCM family in `BEAUti`; if one wants to use the new models, the XML output of `BEAUti` has at least to be adapted by hand.

A complete example (not including the complete codon sequences) of a `BEAST` configuration file can be found in listing S.1 in the supplementary materials. All configuration files include the same standard parts: An MSA, a corresponding tree, a tree model and model for branches, a substitution model, the definition of the MCMC operators – parameters that are allowed to be changed by the MCMC process – and of the MCMC itself as well as definitions for the logfile output.

We did not include priors for the model parameters in the configuration, leading to use uniform distributions as default priors. Choosing other, more appropriate priors for the

parameters might have been beneficial for the likelihood estimations; but since we used the same setting for all different models in the comparison, it is unlikely to have caused a bias in our analysis. If the use of a uniform prior affected some parameters, this disadvantage was shared among all models tested.

The tree topology and the branch lengths were fixed to the values of the given trees, but branch rates were estimated freely by a model of arbitrary branch rates. A uniform prior with a minimum of 0.0 and a maximum of 100.0 was applied to these rates to restrict the model estimation numerically. The chain length was set depending on the MSAs sequence length and number of species; pandit sequences generally required longer chains than the sequences from the OMA project. In the end, the results of each `BEAST` run was analyzed with `Tracer`, a graphical logfile analyser from the `BEAST` distribution. Results with effective sample sizes (ESS) lower than 200 in the important parameters (i.e. posterior, likelihood and model parameters) have been discarded and calculated again with a longer chain. All results reported have therefore at least 200 ESS in the important parameters, most of the results even reach between hundreds and thousands of ESS per parameter.

A ruby script using a `BEAST` XML wrapper class allowed for the convenient mechanical construction of valid `BEAST` configuration files (see listing 2.3).

Listing 2.3 – Example for the usage of the `BEAST` XML wrapper class. The chain length is an estimation based on the size of the MSA file.

```
1 xml = BeastXMLWrapper.new("yang_PF01201", BeastXMLWrapper::YANG)
2 xml.wFolder = "some/output/folder"
3 xml.logFolder = "some/logfolder/yang"
4 xml.omega = 0.3
5 xml.kappa = 1.2
6 xml.msaFile = "MSAs/PF01201.xml"
7 xml.chainLength = [(File.size xml.msaFile)*15, 250000].max
8 xml.logFreq = 1000
9 xml.run
```

2.6. Tools Used

The main programming languages used in this project were `ruby` (Flanagan and Matsumoto, 2008), `Darwin` (Gonnet et al., 2000), `R` (Team, 2010) and `Java`. To access the OMA data and manipulate the multiple sequence alignments (MSAs), `Darwin` was a good choice. It has been built for exactly this kind of work and is maintained by the CBRG that also takes care of the OMA project. The rate matrices of the MSAs for the PCA have been estimated with `XRate` (Klosterman et al., 2006). The PCA was performed in `R`. For the testing of the models we chose `BEAST` (Drummond and Rambaut, 2007) with its GPU library `BEAGLE` (Suchard and Rambaut, 2009) and `codeml` from the `PAML` package (Yang, 2007). Additionally, `evolver` from `PAML` has been used for the simulations. Furthermore, lots of glue code had to be written to build suitable toolchains for the different processes. We used `ruby` for this. The results of the model comparison had been collected in a `MySQL` (MySQL, 2004) database and accessed via `ActiveRecord` (Lerner, 2005), an ORM for ruby. All of the files were managed with `git` (Torvalds and Hamano, 2005), a distributed version control system.

3. Results and Discussion

3.1. Codon Model Comparison

3.1.1. Comparison of Marginal Likelihoods

One of the results of this project is the construction of two new codon substitution models based on our interpretations of the first few principal components (PCs) gained from a *Mammalia* data set (see section 2.2 for detailed information about the data set and the methods used for PCA). To test these new models, we compared the performance of our two new models with two other, widely used codon substitution models on six different data sets. A detailed description of the used sequences alignments can be found in section 2.5.2. Descriptions of all codon models used can be found in section S.1 in the supplementary materials, but here is a short list with the most important features of each model used:

PCM+F+nC: The principal component model (PCM) uses a linear combination of up to n principal components (PCs) and a mean rates matrix to build a new codon substitution matrix. These matrices – the mean rates matrix as well as the PCs – are derived from empirical analysis. In this comparison, we always used the results gained from the *Mammalia* data set described in section 2.2. The scalars of the PCs serve as free model parameters; PCM+2C has therefore two degrees of freedom.

ECM+F+ω+ν: The first PC of the *Mammalia* data set is dominated by the ratio of non-synonymous to synonymous substitutions (ω) while in the second PC, the most dominant factor seems to be a distinction between single- and multi-nucleotide substitutions (see section 3.2.2). We defined a parameter ν that models the multi-nucleotide substitutions. ECM+F+ω+ν was built to measure the influence of this parameter. The model needs an initial rate matrix: We tested it with the matrix given by Kosiol et al. gained from the Pandit database as well as with our own initial rate matrix from *Mammalia* data. When using the latter one, we denoted this as "ECM+F+ω+ν (Mam)".

ECM+F+ω+2κ: This model of the ECM family of Kosiol et al. (2007) includes separate factors for transition and transversion (hence "2κ"). Its two κ parameters model multi-nucleotide substitutions as well as the ratio of transitions to transversions. As initial rate matrix, we used the matrix given by Kosiol et al. as well as our own initial rate matrix. Similarly to ECM+F+ω+ν, the use of the *Mammalia* rate matrix has been denoted with "ECM+F+ω+2κ (Mam)".

M0+F: The widely used model of Goldman and Yang (1994) serves as a base line to our estimations. It has two free parameters, ω and κ.

Figure 3.1 gives an overview of the results of the model comparison. The marginal log-likelihood (mlogL) values have been normalized for each MSA and the mean mlogL for each model type per data set has been calculated. For each mean mlogL m per model and data

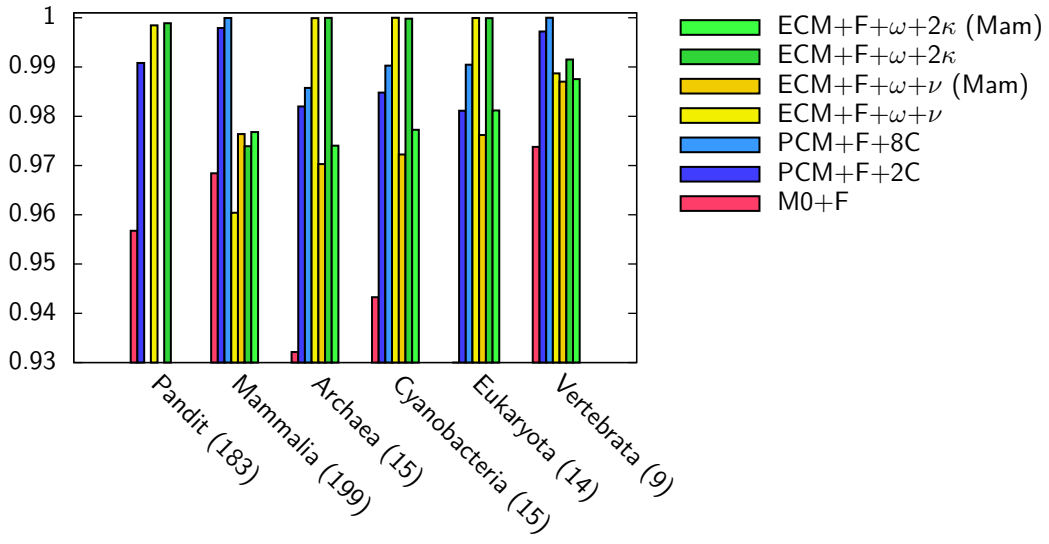


Figure 3.1. – Summarized results of the model comparison: Given the used initial data matrices are comparable, PCM+F+nC (blue) outperforms ECM+F+ω+2κ (light green) in every tested data set. ECM+F+ω+ν (yellow) and ECM+F+ω+2κ (green) show a very similar performance. Marginal log-likelihood (mlogL) values have been normalized for each MSA and the mean mlogL for each model type per data set has been calculated. For each mean mlogL m per model and data set, a bar with height $2 - m$ is shown, thus setting the theoretical maximum to 1.0. Higher bars mean better mlogL values. ECM+F+ω+2κ and ECM+F+ω+ν have been estimated with the initial rate matrix of Kosiol et al. as well as with an initial matrix estimated from the *Mammalia* data set, denoted as "ECM+F+ω+2κ (Mam)" and "ECM+F+ω+ν (Mam)". The Pandit MSAs have not been estimated with all models. The numbers in the data set labels inform about the total number of MSAs in a particular set; complete numbers (counts and percentages) can be found in table S.4.

set, a bar with height $2 - m$ is shown, thus setting the theoretical maximum to 1.0. Higher bars mean better mlogL values.

Generally, our new model PCM+nC (dark and pale blue) can compete with existing models. It outperforms M0+F in every tested data set. The best results can be seen on MSAs including species that belong to or are close to *Mammalia* (i.e. *Mammalia*, *Vertebrata*). This is no surprise, given that the initial matrices (means and principal components) have been estimated from *Mammalia* sequences. Using the original initial rate matrix by Kosiol et al., ECM+F+ω+2κ outperforms PCM+nC in the data sets Pandit, *Archaea*, *Cyanobacteria* and *Other Vertebrata*. This initial rate matrix was generated using 7,332 MSAs from the pandit database, thus involving sequences from many different groups.

Comparing PCM+nC to ECM+ω+2κ with the rate matrix of Kosiol et al. does not only measure the performance of the models but also the influence of the used initial rate matrices. By applying the same rate matrix as used in the PCA to the ECM model, we can effectively test only the models performance. As expected, using the *Mammalia* rate matrix with ECM+ω+2κ results in higher likelihood values in about 60% of all measurements in the *Mammalia* data set compared to ECM+ω+2κ with the original data matrix – yet PCM still outperforms these numbers. Furthermore, using this *Mammalia* rate matrix that roots in the

same data as the PCM initial matrices with means and principal components, PCM+nC outperforms ECM+F+ω+2κ in all data sets tested (see figure 3.1, blue and light green bars). We conclude that PCM outperforms ECM models, given the data matrices used are comparable.

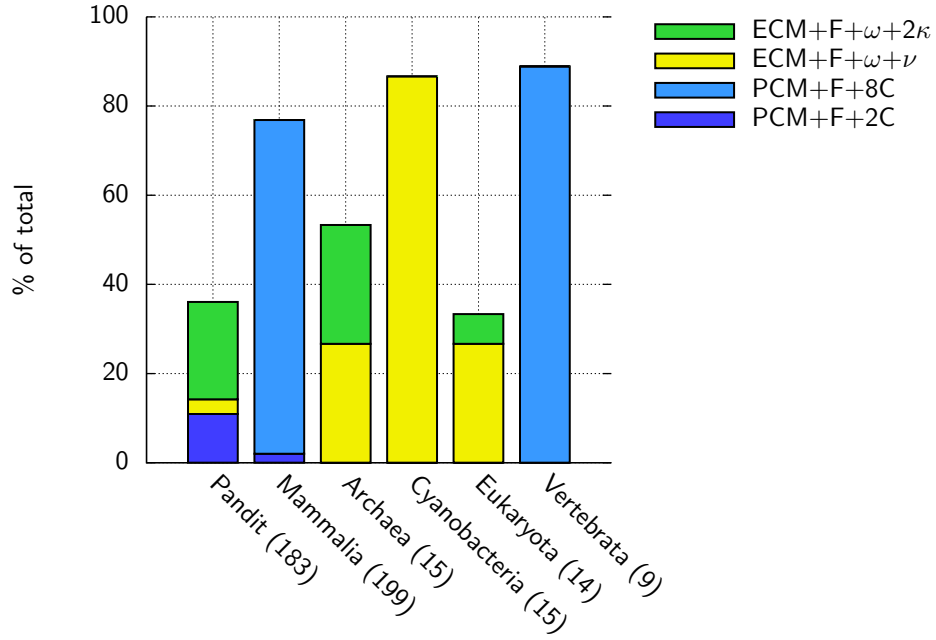


Figure 3.2. – Significant maximum mlogL values: On most MSAs in the Pandit data set, at least two models performed nearly equally well. Reviewing figure 3.1 reveals that these models are most likely ECM+F+ω+2κ and ECM+F+ω+ν.
The number of maximum marginal log-likelihood (mlogL) values per model in each data set have been counted and are shown as percentages of all MSAs in a particular set. Only significant results are counted – if the difference to the second highest mlogL value (i.e. the log Bayes Factor, see section 2.5.1) was less than 3, the MSA has been discarded (Kass and Raftery, 1995). PCM+8C has not been tested with the Pandit MSAs. ECM+F+ω+2κ and ECM+F+ω+ν were also used with an initial matrix estimated from the *Mammalia* data set, but in all measurements either the PCM+F+nC or the ECM+F+ω+2κ with the Kosiol matrix performed better; the models have therefore not been included in this figure. M0+F had no significant results and was excluded as well. The numbers in the data set labels inform about the total number of MSAs in a particular set; complete numbers (counts and percentages) can be found in table S.4.

Figure 3.2 gives another view to the results of the model comparison. The maximum marginal log-likelihood values per model in each data set have been counted and are shown as percentages of all MSAs in a particular set, but only significant results are included in the graph. *Mammalia*, *Cyanobacteria* and *Vertebrata* show a high coverage of significant maximum marginal likelihood values of around or more than 80%. Lower coverage indicates that no model has been able to clearly outperform its competitors in the given data set. Bars that include mostly one model (such as *Mammalia*, *Cyanobacteria* or *Vertebrata*) indicate a clear dominance of one model in the particular data set.

The most diverse result, i.e. a low significant coverage that is split on three models, emerges from the *Pandit* MSAs. It indicates that on most MSAs, at least two models performed nearly

equally well. Reviewing figure 3.1 reveals that these models are most likely ECM+F+ω+2κ and ECM+F+ω+ν, but also PCM+F+2C outperforms these models in some MSAs.

Selected MSAs of this data set are listed in table 3.1; these are the Pandit families also analysed in detail by Kosiol et al. in their paper from 2007. The *Pandit* families are not limited to distinct groups of species but focus on all available DNA evidence for a certain gene. PF01226, for example, includes *Haemophilus influenzae*, *Methanobacterium formicicum*, *Saccharomyces cerevisiae* (baker's yeast), *Bacillus subtilis* (hay bacillus) and *Escherichia coli*. All these species do not belong to *Mammalia*, and given the results in figure 3.1, it is not surprising that ECM fits best. PF01229, on the other hand, includes *Caldocellum saccharolyticum*, *Canis familiaris* (dog), *Mus musculus* (mouse) and *Thermoanaerobacter saccharolyticum*. Dog and mouse belong to *Mammalia* and represent 50% of the sequences in this MSA. This is likely a reason why the PCM performs better on this MSA.

Table 3.1. – Detailed results of selected Pandit MSAs: PCM+F+2C performs better on MSAs that include a higher percentage of *Mammalia* sequences. In PF01229, 50% of the sequences are *Mammalia* (PF01226: 0%, PF01233: 20%). Note that the results for ECM+F+ω+ν and ECM+F+ω+2κ in PF01226 and PF01233 are still of strong significance compared to PCM+F+2C, even if the maximum mlogL value is not marked as significant compared to the second highest mlogL.
Highest marginal log likelihood (mlogL) values per MSA are marked with a star; additional stars denote the significance to the second best value. Evidence for the higher value is positive (two stars) if the difference d between best and second best mlogL values (i.e. the log Bayes Factor, see section 2.5.1) is $3 < d < 20$, strong (three stars) if $20 < d < 150$ and very strong (four stars) if $d > 150$. Differences below 3 are not significant (Kass and Raftery, 1995).

	M0+F	**PCM+F+2C**	**ECM+F+ω+ν**	**ECM+F+ω+2κ**
PF01226	-5667.00	-5548.60	-5509.17*	-5509.85
PF01229	-6724.47	-6590.68**	-6608.41	-6602.55
PF01233	-2406.75	-2319.95	-2292.35*	-2292.54

Figure 3.3 compares and summarizes only the significant maximum mlogL values per ECM model of each data set. For each ECM+F+ω+2κ and ECM+F+ω+ν, mlogL values have been calculated using Kosiol's initial rate matrix and using a rate matrix estimated in the first steps of the PCA on the *Mammalia* data. The maximum mlogL value of both ECM+F+ω+2κ models was then compared to the maximum mlogL value of both ECM+F+ω+ν models. If the difference between the two compared mlogL values (i.e. the log Bayes Factor, see section 2.5.1) was smaller than 3, the result was discarded for lack of significance (Kass and Raftery, 1995). ECM+F+ω+ν performed clearly better than ECM+ω+2κ in our *Cyanobacteria* and *Other Eukaryota* data set. In the *Archaea* set, both models perform equally. In *Mammalia*, ECM+F+ω+2κ outperforms ECM+F+ω+ν in nearly every MSA. In *Vertebrata*, this is true for 8 out of 9 MSAs, and the missing MSA was also in favor for ECM+F+ω+2κ, although this result is not significant and therefore not included in the graph. The Pandit data set shows an interesting feature: Only one third of the mlogL values are significant; in two thirds of all estimations, ECM+F+ω+ν and ECM+ω+2κ performed equally well. This can also be observed in *Archaea* (46.6% not significant) and *Other Eukaryota* (64.3% not significant), although it is more important in the Pandit set since this group contains a larger number of MSAs.

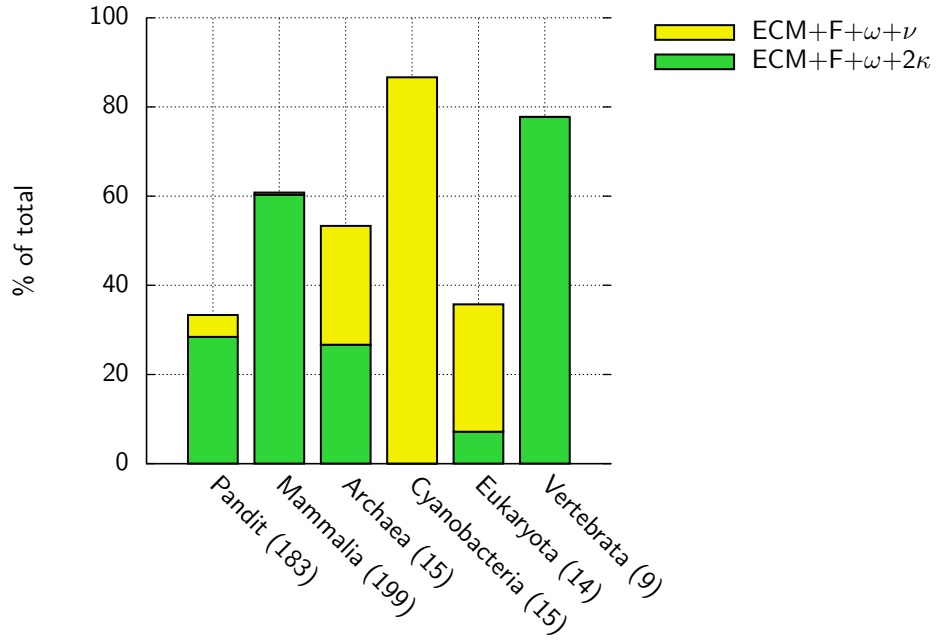


Figure 3.3. – Summarized significant results of only the four ECM models: ECM+F+ω+ν outperforms ECM+F+ω+2κ in all data sets where mean and variance of the κ distribution is low (see table 3.2). In the pandit data set, only one third of the mlogL values are significant; in two thirds of all estimations, ECM+F+ω+ν and ECM+ω+2κ performed equally well.
The number of maximum marginal log likelihoods (mlogL) per model of each data set (restricted to results of ECM models) have been counted and are shown as percentages of all MSAs in a particular set. For each ECM+F+ω+2κ and ECM+F+ω+ν, mlogL values have been calculated using Kosiol's initial rate matrix and using a rate matrix estimated in the first steps of the PCA on the *Mammalia* data. The maximum mlogL value of both ECM+F+ω+2κ models was then compared to the maximum mlogL value of both ECM+F+ω+ν models. If the difference between the two compared maximum mlogL values (i.e. the log Bayes Factor, see section 2.5.1) was smaller than 3, the result was discarded for lack of significance (Kass and Raftery, 1995). The numbers in the data set labels inform about the total number of MSAs in a particular set. Complete numbers (counts and percentages) can be found in table S.5.

ECM+F+ω+ν could be understood as a special case of ECM+F+ω+2κ: If $\kappa_{ts} = \kappa_{tv} = \kappa$, its generating rules for the substitution matrix transform to

$$Q_{ls} = \begin{cases} 0 & \text{if } l \text{ or } s \text{ are stop codons} \\ \pi_s I_{ls} \kappa^n & \text{if } l \to s \text{ is a synonymous change} \\ \pi_s I_{ls} \kappa^n \omega & \text{if } l \to s \text{ is a non-synonymous change} \end{cases}$$

with n denoting the number of nucleotides changed. ECM+F+ω+ν uses a very similar approach:

$$Q_{ls} = \begin{cases} 0 & \text{if } l \text{ or } s \text{ is a stop codon} \\ \pi_s I_{ls} \nu^m & \text{if } l \to s \text{ is a synonymous change} \\ \pi_s I_{ls} \nu^m \omega & \text{if } l \to s \text{ is a non-synonymous change} \end{cases}$$

but with $m = 0$ for a single nucleotide change and $m = 1$ for a multi-nucleotide substitution. If the κ values in ECM+F+ω+2κ are equal, the only difference between these two models are

the exponents n and m. ECM+F+ω+ν is not a subset of ECM+F+ω+2κ, but the models can – depending on the parameters of ECM+F+ω+2κ – become so similar that it is very hard to make a distinction, resulting in very similar mlogL values.

Table 3.2. – The distribution of κ in the tested data sets corresponds to the performance of ECM+F+ω+2κ compared to ECM+F+ω+ν. Shown are mean and variance of the distribution of κ for each data set as well as the significant maximum marginal likelihood values of the model comparison in figure 3.3. Low mean κ values and small variances correspond with better performance of ECM+F+ω+ν; when increasing the importance of κ in the data set, ECM+F+ω+2κ is more likely to outperform its competitor.

	mean(κ)	**var(κ)**	**ECM+F+ω+ν**	**ECM+F+ω+2κ**
Pandit	1.487457	0.3517328	9 (4.9%)	52 (28.3%)
Mammalia	2.703795	0.2958711	0	199 (100%)
Archaea	1.539323	0.00905533	4 (26.7%)	4 (26.7%)
Cyanobacteria	1.198291	0.002759878	13 (86.7%)	0
Eukaryota	1.261841	0.003330507	4 (28.6%)	1 (7.1%)
Vertebrata	2.049522	0.1136261	0	8 (88.9%)

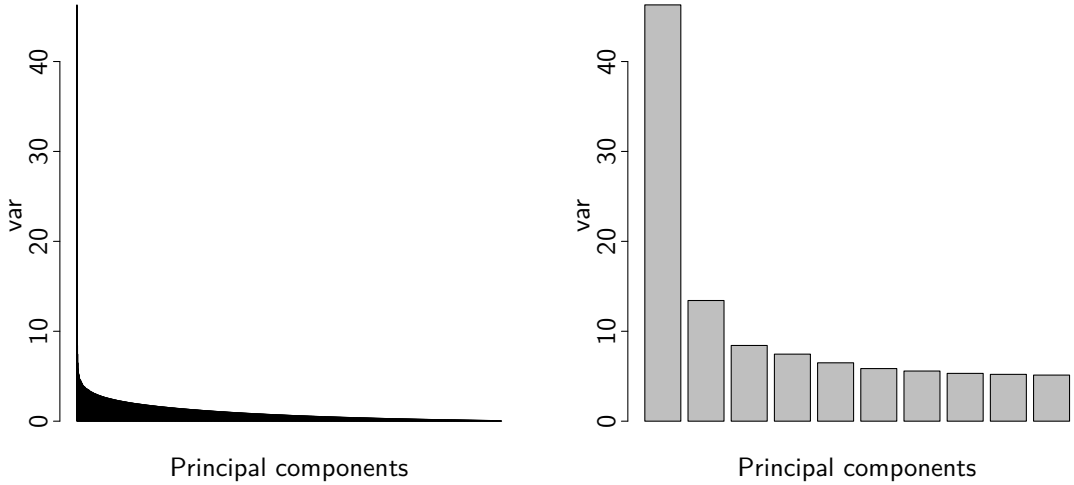
The biggest difference between both models is the introduction of transitions and transversions in ECM+F+ω+2κ vs. the lack of it in ECM+F+ω+ν. We would therefore expect ECM+F+ω+ν to perform as well as ECM+F+ω+2κ on data sets where the mean of all κ values, the ratios of transitions to transversions per MSA, is near 1 and the variance on its distribution is very small. This is exactly what we observe in our results. Table 3.2 shows for each data set the mean and variance of the distribution of κ in the particular MSAs, as well as the numbers of the model comparison shown in figure 3.3. Low mean κ values and small variances correspond with better performance of ECM+F+ω+ν; when increasing the importance of κ in the data set, ECM+F+ω+2κ is more likely to outperform its competitor.

A complete overview of all likelihood values and summaries can be found in tables S.4, S.5, S.6, S.7, S.8, S.9, S.10 and S.11 in the supplementary materials.

3.1.2. Computational Performance

Graphical processing units (GPUs) in computers nowadays not only render graphics animations and desktops but can also help to speed up certain calculations traditionally performed on the central processing unit (CPU). GPUs are highly parallelized computing units with up to hundreds of individual specialized computing cores. Nearly all calculations that can be parallelized – such as calculating the likelihood of a codon substitution model given a phylogenetic tree and an MSA – can benefit from this architecture. We used `BEAST` (Drummond and Rambaut, 2007) with its GPU library `BEAGLE` (Suchard and Rambaut, 2009) to estimate the likelihood values in a Bayesian framework. Calculating MCMCs is a computationally expensive task; the performance boost gained by using the GPU compared to the CPUs of our machines was welcomed.

Suchard and Rambaut (2009) report speedup between 12 and 150, depending on the organisation of the problem and the amount of GPUs used. We performed some short tests on a laptop (i7 Mobile 2.66GHz CPU with 8GB RAM and an NVIDIA GeForce GT 330M with 512MB VRAM) and could get a mean speedup of ≈ 15. If calculating the MCMC chain for



(a) Complete scree plot **(b)** First ten principal components

Figure 3.4. – Scree plots of the *Mammalia* data set after PCA.

a single, average MSA took about one day on the CPU, the same calculation needed only about 1.5 hours if done on the GPU.

When we started our simulations with `BEAST` on the Brutus cluster (see 2.3.1), we scheduled a first batch of jobs that took 3 days to complete. Three of the available nodes were used, and on every node, six jobs ran simultaneously. If these simulations had to be done simultaneously on the four conventional machines we had access to (Pentium4 Dualcore, 3GHz, no CUDA, no GPU), even with a conservative speedup of 10, this would have taken about 67.5 days.

It took about 3 weeks to perform all the needed calculations on the GPU cluster. Without the use of `BEAGLE` and the GPUs, we would never have been able to gather as much data in the given time.

3.2. PCA of the Mammalia MSAs

3.2.1. Analysis of the Eigenvalues – The Stopping Problem

PCA is also a method for dimensionality reduction, and only the first few PCs have to be used to explain most of the data in observation; but what is the cutoff point where one can discard PCs with lower eigenvalues without losing too much information?

A scree plot (as in figure 3.4) shows the eigenvalues of the covariance matrix in rank order, highest eigenvalues first. Normally, the first few PCs show a steep decrease in their corresponding eigenvalues, whereas the last components show eigenvalues that are more or less equal, which leads to a characteristic curve (e.g. figure 3.4a). The cutoff point is chosen to be the position where the first few eigenvalues depart from the more or less straight line of the last eigenvalues, including one point to the right of this position. In figure 3.4b, this would mean that up to three, maybe four principal components are to be called “important”, the rest of the 1830 components have to be discarded. This result reveals a problem of the

scree plot method: It's a good way to gain some first insight, but it is not always very clear which position exactly has to be chosen for the cutoff point.

As a second method to determine a cutoff point, we used bootstrapping of the original data, applying PCA to the samples and then comparing the confidence intervals of the distributions of eigenvalues (Jackson, 1993). Overlapping confidence intervals between successive eigenvalues indicate that the these eigenvalues are indistinguishable from one another. Table 3.3 shows the results of this analysis. However, none of the calculated intervals (for components up to the first twenty) showed an overlap. This analysis did not reveal a reasonable cutoff point amongst the first twenty PCs that was valid for all MSAs. We did not expand our analysis to more than those first twenty components, because using PCM+nC with $n > 20$ will probably not be computationally feasible, depending on the size of the MSA and the length of its sequences. Although there probably is a general cutoff point beyond the first twenty PCs, it would not help restricting PCM to reasonable complexity.

Table 3.3. – Confidence intervals (on a 95% level) of the standard deviations (i.e. square roots of the eigenvalues of the covariance matrix) of the first ten principal components from 250 bootstrapped data matrices, rounded to five digits. None of the intervals (additional intervals up to the first twenty components have been calculated) overlap. This analysis is not suitable as a stopping criterion.

	Lower bound	**Upper bound**
PC1	0.02566	0.02577
PC2	0.00796	0.00806
PC3	0.00573	0.00590
PC4	0.00507	0.00517
PC5	0.00470	0.00478
PC6	0.00443	0.00451
PC7	0.00422	0.00429
PC8	0.00405	0.00411
PC9	0.00390	0.00396
PC10	0.00379	0.00384

Instead of expanding our search for a general cutoff point to larger values of n, we applied a third method to answer the stopping problem individually for each MSA. Five different MSAs from the *Mammalia* data set were selected randomly, and marginal log-likelihood (mlogL) values have been calculated with `BEAST` using between one and ten PCs. The optimal number n of PCs used for an individual MSA was defined such that PCM+$(n+m)$C with $0 < m < 10 - n$ had a significantly better result than PCM+nC. The most significant result was found by calculating p-values and choosing n and m such that the p-value was minimal. We used the likelihood ratio test (LRT) to measure the significance of the mlogL results.

It turned out that the results were highly dependent on the chosen MSA. LRT did not show a consistent behaviour. All MSAs gained from more PCs used in the PCM model, but the optimal number of PCs used was very different. Table 3.4 shows the optimal number of PCs for each tested MSA as well as the $2\Delta l$ statistics to the next higher number of PCs and the corresponding p-values.

For computational reasons, we chose to use PCM+2C and PCM+8C for our model comparisons. Two components already give a good estimation of the performance, and models

Table 3.4. – Results of LRT on the optimal number n of PCs used in PCM where $n+m$ PCs used did not result in a significantly better result. The most significant result was found by calculating p-values and choosing n and m such that the p-value was minimal. All six MSAs show a very different behaviour.

	Optimal Nr.	$2\Delta l$	p**-value**
MSA 1 (mam_bu)	3	32.94	$9.496e{-09}$
MSA 2 (mam_cy)	3	15.83	0.0003650
MSA 3 (mam_dm)	8	16.38	$5.170e{-05}$
MSA 4 (mam_fl)	7	21.81	$7.138e{-05}$
MSA 5 (mam_gr)	10	24.09	$7.674e{-05}$
MSA 6 (mam_hg)	2	28.17	$1.112e{-07}$

with more than eight PCs were computationally too expensive; although this has to be seen in the context of the amount of MSAs we wanted to test. When estimating only few MSAs, PCM+10C and above is still feasible in a realistic amount of time if the calculation speed can be increased by using GPUs.

3.2.2. Analysis and Interpretation of the Principal Components

Principal components can be interpreted as axes of a new Cartesian coordinate system. This new coordinate system is rotated such that its axes point in the directions of the maximum variance in the data enclosed by the system. Figure 1.5 shows this in a two-dimensional example. The position of every point in the graph can either be described as a linear combination of x and y or as a linear combination of PC1 and PC2. But more important: The PCs themselves can also be described as linear combinations of the two parameter vectors $[x, 0]$ and $[0, y]$. Similarly, the PCs we find in a set of substitution matrices can be understood as linear combinations of a number of unknown parameter vectors.

Table 3.5. – Feature vectors used in the analysis of *Mammalia* PCs. Each vector consists of 1830 parameters corresponding to the substitution rate parameters in the PCs. "AA" denotes "amino acid".

Name	**Encoding**
ω	Non-synonymous substitutions $\rightarrow 1.0$, synonymous substitutions $\rightarrow -1.0$
ν	Single nucleotide substitutions $\rightarrow 1.0$, multi nucleotide substitutions $\rightarrow -1.0$
κ	(Number of transitions) - (number of transversions)
Weight	Weight of AA [Da], scaled to $[-1.0, 1.0]$
Acidity	Acidity of AA (acidic, neutral, basic) changes per substitution $\rightarrow -1.0$, keeps constant $\rightarrow 1.0$
Charge	Charge (positive, neutral, negative) changes per substitution $\rightarrow -1.0$, keeps constant $\rightarrow 1.0$
Hydropathy	Hydropathy (Kyte and Doolittle, 1982) changes algebraic sign $\rightarrow -1.0$, keeps algebraic sign $\rightarrow 1.0$
Polarity	Polarity (polar, non-polar) changes per subst. $\rightarrow -1.0$, keeps constant $\rightarrow 1.0$
Random	A test vector filled randomly with 1.0 and -1.0

If PCA can be used to find relevant parameters of codon substitution models, these parameters were expected to be part of linear combinations that result in the emerged PCs. By encoding common model parameters and features of amino acids in vectors of 1830 parameters corresponding to the 1830 substitution rate parameters in the PCs, feature vectors have been constructed that can be compared to the PCs. The absolute value of the scalar product of a normed feature vector and a PC reveals the similarity between the two vectors, thus allowing to test if a feature vector is a possible relevant part in some linear combination describing the PC (scalar projection analysis). Table 3.5 explains the feature vectors used in this analysis. The feature vectors have been tested for correlations between themselves. κ correlates mostly with ω (scalar product: 0.4465), ν also with ω (0.7639), and *Weight* correlates mostly with ω as well (0.6280). *Acidity* and *Charge* have a high correlation of 0.8798, which is no surprise since the level of acidity depends on the charge of the molecule. Also, polarity and hydropathy show a correlation (0.318).

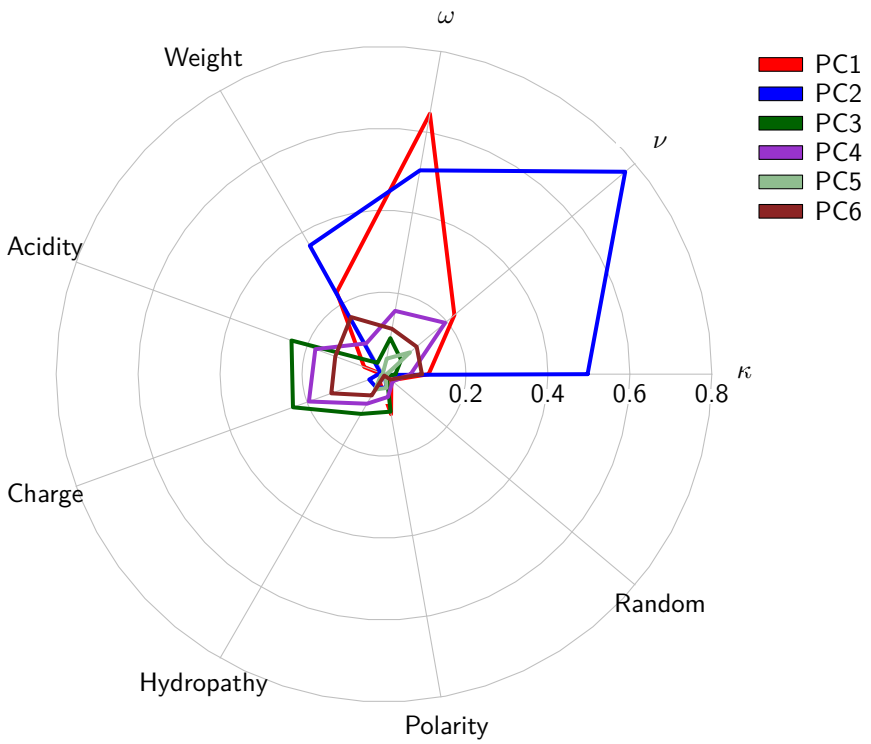


Figure 3.5. – Scalar products of normed feature vectors and the first PCs of the *Mammalia* data set. Higher numbers denote a higher correlation between a parameter and a PC; 1.0 would denote a perfect correlation. All analyzed PCs were found to model to some extent parameters commonly considered in molecular evolution such as the feature parameters discussed in table 3.5. The "Random" parameter serves as a noise threshold; scalar products smaller than this noise level are considered as insignificant. Complete numbers can be found in table S.3 in the supplementary material, an excerpt for the first 3 PCs can be found in table 3.7.

Figure 3.5 shows the scalar products of the feature vectors and the first six PCs of the *Mammalia* data set. Higher numbers denote a higher correlation between a PC and a feature vector. All analyzed PCs were found to model to some extent parameters commonly considered in molecular evolution such as the discussed feature parameters, although our list is of course not exhaustive: We can only test for features we define in advance. Other unknown parameters might have a relevant influence, too. The maximum scalar product of the random feature vector with any of the analyzed PCs is 0.0370; any scalar product equal or below this result cannot be distinguished from noise.

According to this analysis, we expect to find certain patterns in the first PCs: In PC1, ω should be clearly visible. PC2 should be a mixture of ν, κ and ω with ν being the most relevant feature. Beginning with PC3, physico-chemical features of the amino acids start to play a more important role; these features do not occur at all in the first two PCs.

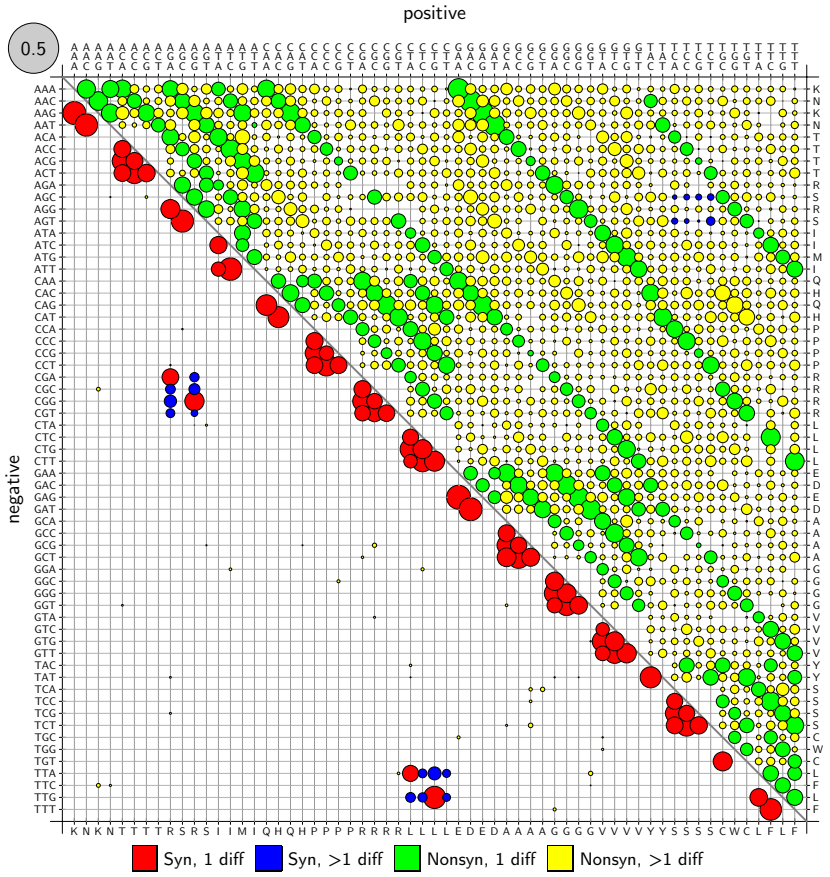


Figure 3.6. – First principal component of *Mammalia* data set; positive values are shown in the upper right half, negative values in the lower left half of the matrix. The algebraic sign of PCs is arbitrary. The diagonal clearly separates synonymous (red and blue) and non-synonymous (green and yellow) substitutions which corresponds to the scalar projection analysis that predicted a pattern corresponding mainly with ω.

The first principal component of the *Mammalia* data set is shown in figure 3.6. Positive values are shown in the upper right half, while negative values are drawn in the lower left half of the matrix. The diagonal mostly separates synonymous (red and blue) from non-

synonymous substitutions (green and yellow). There are some exceptions to this observation, but the values corresponding to these substitutions are very small. Their influence on the component is therefore almost negligible. This corresponds to the scalar projection analysis that predicted a pattern corresponding mainly with ω.

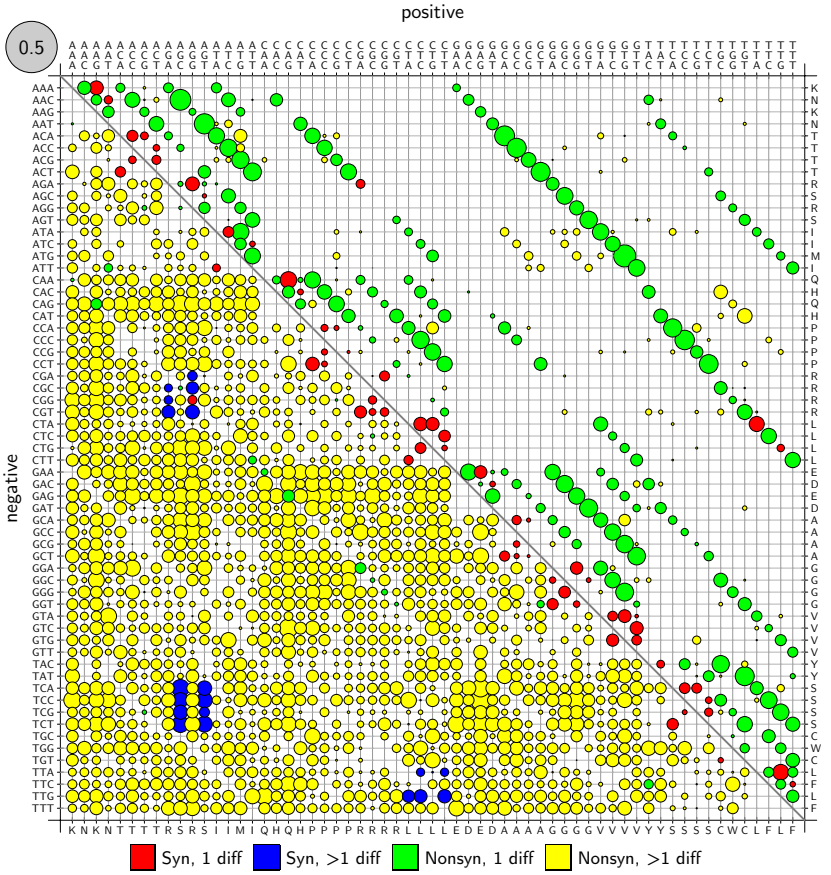


Figure 3.7. – Second principal component of *Mammalia* data set with positive values in the upper right and negative values in the lower left half of the matrix. The algebraic sign of PCs is arbitrary. The diagonal seems to separate substitutions involving two or three nucleotide changes (blue and yellow) from single-nucleotide substitutions (red and green), which corresponds to the scalar projection analysis of PC2 that predicted a mixed pattern corresponding mostly to ν combined with κ and ω.

Figure 3.7 shows the second principal component of the *Mammalia* data. Here, the diagonal seems to separate substitutions involving two or three nucleotide changes (blue and yellow) from single-nucleotide substitutions (red and green). This separation is not as close to perfect as in the first component; several substitutions do not follow this observed pattern. Nevertheless, the multi-nucleotide pattern seems to be the most prominent in the given component. This finding corresponds again to our predictions in the scalar projection analysis (figure 3.5). The existence of positive instantaneous rates for multi-nucleotide substitutions is well documented (Averof et al., 2000; Whelan and Goldman, 2004; Kosiol et al., 2007), but the topic is not fully understood and currently debated (e.g. Anisimova and Kosiol, 2009).

Figure S.3 in the supplementary materials shows the next four components; they mostly

consist of non-synonymous substitutions. According to our scalar projection analysis, these PCs include physico-chemical features of the amino acids such as acidity, charge, hydropathy and weight.

3.2.3. Validation of Assumptions

PCA makes (as mentioned in section 1.4) two assumptions that need to be tested: The signal in the data is stronger than the noise and the parameters follow a normal distribution. To test the first assumption, 250 new data matrices were sampled from the initial one where the 3666 values for each parameter were shuffled; this idea originated from G. Gonnet in a discussion about the validation of the PCA. The distribution of these sampled parameters is exactly the same as in the original data set, but the rows (i.e. the data points) itself have been randomized and thus became independent from each other. The confidence intervals of the matrices' eigenvalues were assumed to give a lower limit of what is considered to be random noise.

Table 3.6. – Confidence intervals (on a 95% level) of standard deviations (square roots of the eigenvalues) of the first five eigenvalues from the 250 randomized data matrices, rounded to six digits. This implies that principal components with eigenvalues equal or smaller to 0.001620 in the *Mammalia* data set cannot be distinguished from random noise.

	Lower bound	**Upper bound**
PC1	0.001611	0.001620
PC2	0.001588	0.001591
PC3	0.001576	0.001579
PC4	0.001568	0.001570
PC5	0.001560	0.001562

The results are shown in table 3.6. The highest value is 0.001620; if we encounter principal components in the *Mammalia* data whose corresponding standard deviation is equal or smaller to this lower bound, they cannot be distinguished from random noise and have to be discarded. This affects the last 1743 components of our *Mammalia* data, which accounts for 79% of the total variance in the data.

The 2nd assumption (the normal distribution of the parameters) does not hold for all parameters. Figure 3.8 shows three typical parameter distribution that could be found in our 3666 data points. Note the high peaks of values nearly zero in all plots. While the first distribution in figure 3.8a could be interpreted as a zero-inflated normal distribution (although the data fails the Shapiro-Wilk test for normality), the second and third distribution in figures 3.8b and 3.8c are clearly not Gaussian.

These results do not indicate that PCA is incorrect when applied to our data, but rather show it is not optimal. Non-normal distributions in the data lower the performance of PCA, but do not necessarily prevent the application of the method (Dudzinski et al., 1975).

3.2.4. Results of the Simulations

As mentioned in section 2.2.4, three different simulations were performed. In the first simulation, new MSAs have been generated with the same values for ω, κ, the corresponding trees from `Darwin`, the sequence length and the number of species under an M0 model as in

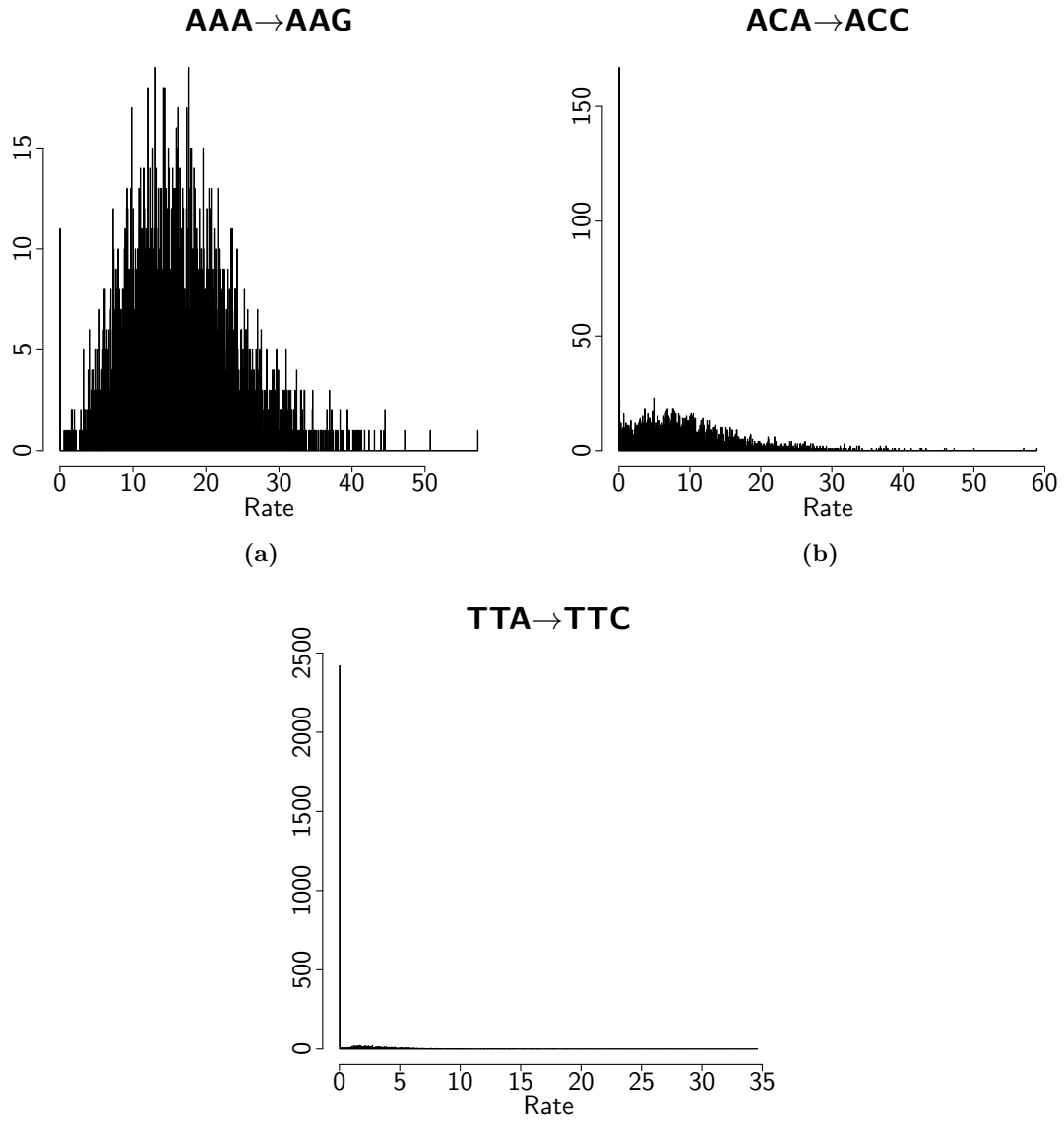


Figure 3.8. – Three typical parameter distributions in the *Mammalia* data set. 24 parameters show a similar distribution as in **a)**, 33 are similar to **b)**, the rest of the parameters are close to the distribution in **c)**. Note the high peaks of values nearly zero in all plots; in **c)**, most of the values are close to zero.

the 3666 original *Mammalia* MSAs. The resulting new 3666 MSAs have then undergone the same procedure as the original ones to gain a new exchangeability rate matrix.

The goal of this first simulation was to test whether the results of the PCA with *Mammalia* data were within reasonable range. Since the MSAs of the simulation were produced by some model of molecular evolution that was parametrized with estimations from the *Mammalia* data, we expected to find two basic features in the new results. First, at least the first few principal components should be similar to the PCs from the original *Mammalia* data, making the corresponding eigenvalues to those PCs comparable. Second, since models are always simplifications, we expected to get less noise in the simulated data compared to the *Mammalia* data set. Less noise should lead to higher eigenvalues in the first PCs.

Table 3.7. – Scalar products of PCs and normed feature vectors. The feature vectors are explained in detail in table 3.5. "MPC1" denotes the first PC of the *Mammalia* data set, "S3PC3" the third PC of the third simulation. Feature vector names have been abbreviated for better readability of table (weight, acidity, charge, hydropathy, polarity, random).

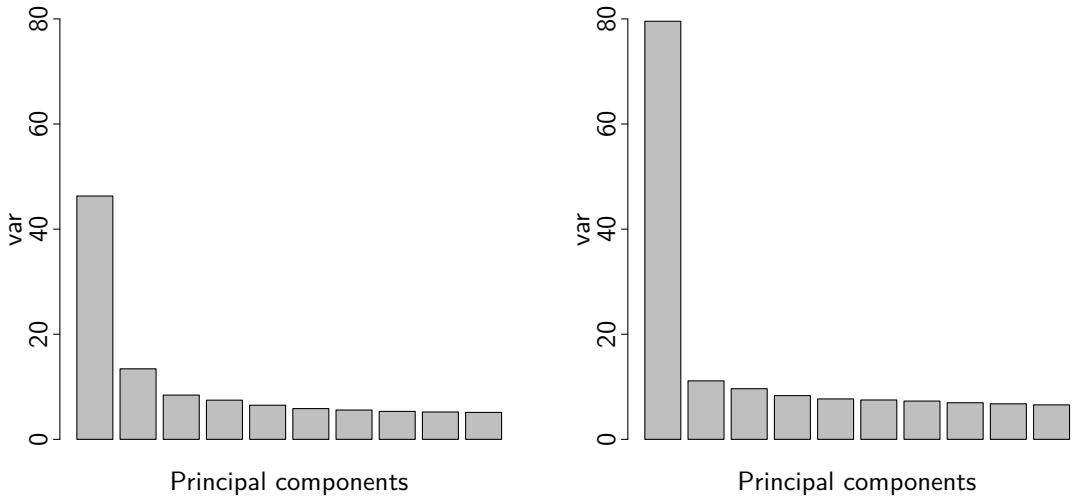
PC	κ	ν	ω	**W.**	**Ac.**	**Ch.**	**Hy.**	**Pol.**	**Rnd**
MPC1	0.1089	0.2243	0.6452	0.2308	0.0514	0.0032	0.0212	0.0993	0.0234
MPC2	0.4968	0.7684	0.5050	0.3621	0.0090	0.0389	0.0375	0.0339	0.0029
MPC3	0.0280	0.0545	0.0893	0.0318	0.2402	0.2364	0.1120	0.0930	0.0089
S1PC1	0.1840	0.2199	0.7037	0.3840	0.0587	0.1258	0.0004	0.0698	0.0173
S1PC2	0.2822	0.6809	0.4584	0.3975	0.1031	0.1568	0.0247	0.0452	0.0085
S1PC3	0.1102	0.1025	0.0525	0.0606	0.0220	0.0300	0.0130	0.0023	0.0370
S2PC1	0.2528	0.2245	0.7069	0.3579	0.0368	0.1036	0.0146	0.0818	0.0138
S2PC2	0.2654	0.8153	0.5301	0.4059	0.0951	0.1415	0.0179	0.0598	0.0214
S2PC3	0.1145	0.1232	0.0720	0.0482	0.0196	0.0211	0.0190	0.0210	0.0316
S3PC1	0.2232	0.2072	0.7003	0.4269	0.0826	0.1521	0.0225	0.0472	0.0181
S3PC2	0.4203	0.0045	0.0150	0.2065	0.0539	0.0807	0.0702	0.0839	0.0257
S3PC3	0.1862	0.3571	0.2404	0.1761	0.0109	0.0207	0.0512	0.0636	0.0272

Figure 3.10 shows the first two principal components of both data sets. The plots seem to represent very similar vectors, and this observation corresponds to the feature vector analysis (see table 3.7). This also makes the corresponding eigenvalues comparable and allows us to estimate if the values gained from the original data set are reasonable.

The scree plots in figure 3.9 show the variances of the ordered PCs (i.e. the corresponding eigenvalues of the ordered eigenvectors) of both the original data and the first simulated data set; the actual numbers can be found in table S.2 in the supplementary materials. As we expected, the first eigenvalues of the simulation are slightly higher than the ones of the *Mammalia* data.

We have shown with the randomized data set that the first principal components are indeed not random artefacts. On the other hand, comparing the eigenvalues with the results of the first simulation tell us that they are not unreasonably high, and comparing the first component vectors we see very similar characteristics. We concluded therefore that the resulting principal components of the *Mammalia* data set emerge from a real signal.

The second and third simulation should help testing the validity of our working hypothesis. We assumed that free parameters of M0 should show up in the first PCs, since these parameters account for the variability in the data. To test this hypothesis, we fixed one of



(a) *Mammalia* data set **(b)** Simulation 1

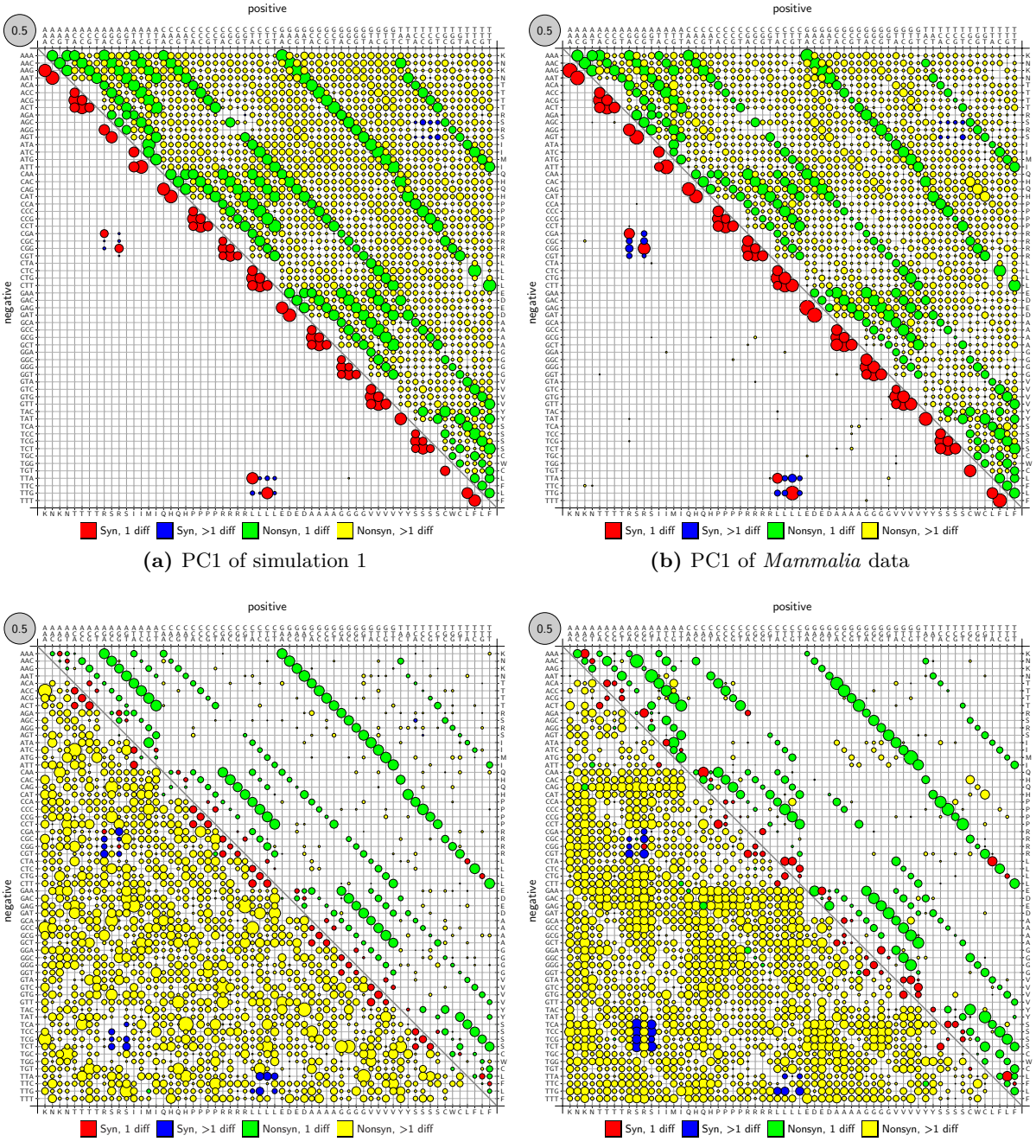
Figure 3.9. – Scree plots of the first ten components of the *Mammalia* data set and the first simulation. Complete numbers can be found in tableS.2.

the parameters – κ, the ratio of transitions to transversion – at 1.0 in the second simulation while drawing it from a normal distribution with big variance ($\kappa \sim N(6, 2.5)$; values below 0.2 have been discarded) for the third simulation.

The first PCs of both the second and the third simulation are very similar to the first components of the *Mammalia* data and the first simulation (see table 3.7). As we did not randomize the single parameter in the model that could influence this component – ω, which balances synonymous and non-synonymous substitutions – but used the values estimated from the original data, this result was not surprising.

PC2 of the second simulation (figure 3.11a) resembles the second PCs of the first simulation and the original *Mammalia* data. The separation is mainly between substitutions with single nucleotide changes (red and light red) and substitutions with two or three nucleotide changes. The fixed parameter κ, the ratio between transitions and transversions, is not directly visible; but scalar projection analysis shows that it is part of the linear combination that forms PC2. The absolute value of the scalar product of PC2 and the κ feature vector is in simulation 2 even higher than in simulation 1 and the original data set; but this has to be seen with respect to the other scalar product values for that PC, since not the absolute values but their difference determine the result of the linear combination. This taken into account, the absolute contribution of κ in the second simulation has clearly decreased.

In the third simulation, we varied κ strongly. The second PC (figure 3.11b) visually reflects this variation: It does no longer separate between single and multi nucleotide changes (red and light red bubbles are on opposite sides of the diagonal), but distinguishes between transitions (shown in strong colors) and transversion (shown in pale colors). The scalar projection analysis supports this observation: κ is by far the most strongest influence in this PC. Interestingly, the pattern we found in PC2 of the original data set and in PC2 of the first and second simulation – namely a distinction between single- and multi-nucleotide substitutions



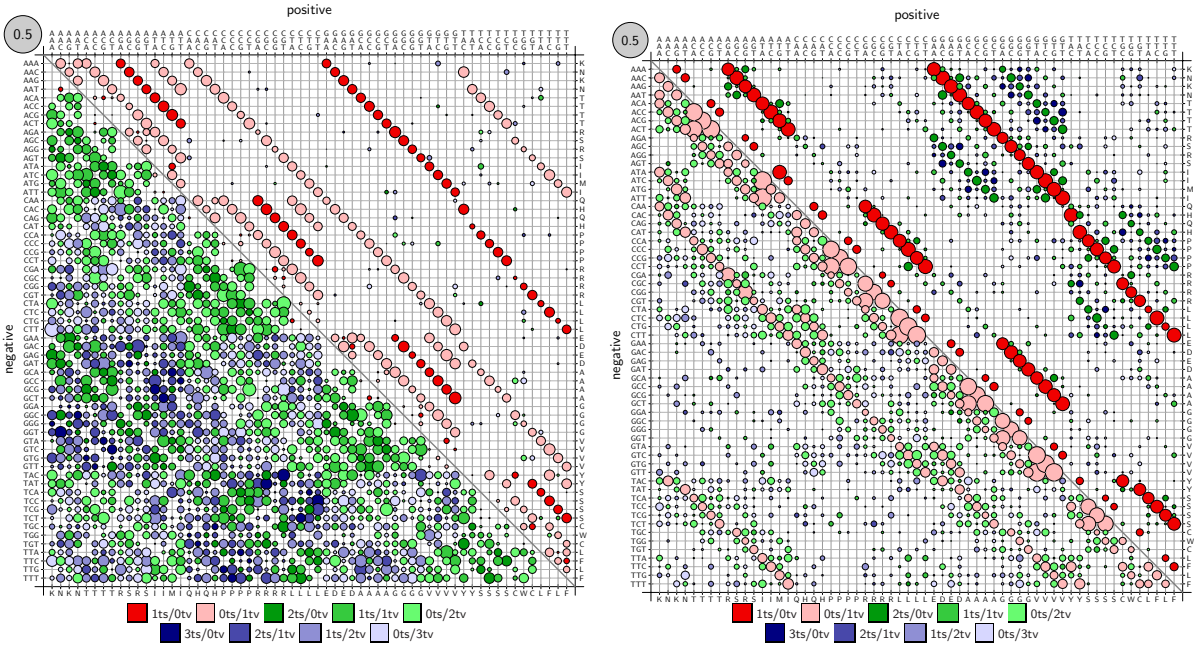
(a) PC1 of simulation 1

(b) PC1 of *Mammalia* data

(c) PC2 of simulation 1

(d) PC2 of *Mammalia* data

Figure 3.10. – Bubble plots of the first two principal components from simulation 1 and the original *Mammalia* data.



(a) PC2 of simulation 2 **(b)** PC2 of simulation 3

Figure 3.11. – Second principal components of the second and third simulation. Positive values are shown in the upper right half, negative values in the lower left half of the matrix. The algebraic sign of PCs is arbitrary. All findings correspond to the scalar projection analysis given in table 3.7. **a)** The diagonal separates mostly between single nucleotide substitutions (light red and red, 1ts/0tv and 0ts/1tv substitutions) and substitutions involving two or three nucleotides. It is very similar to the second component of the *Mammalia* data. **b)** The separation mainly takes place between transitions (strong colors) and transversions (pale colors).

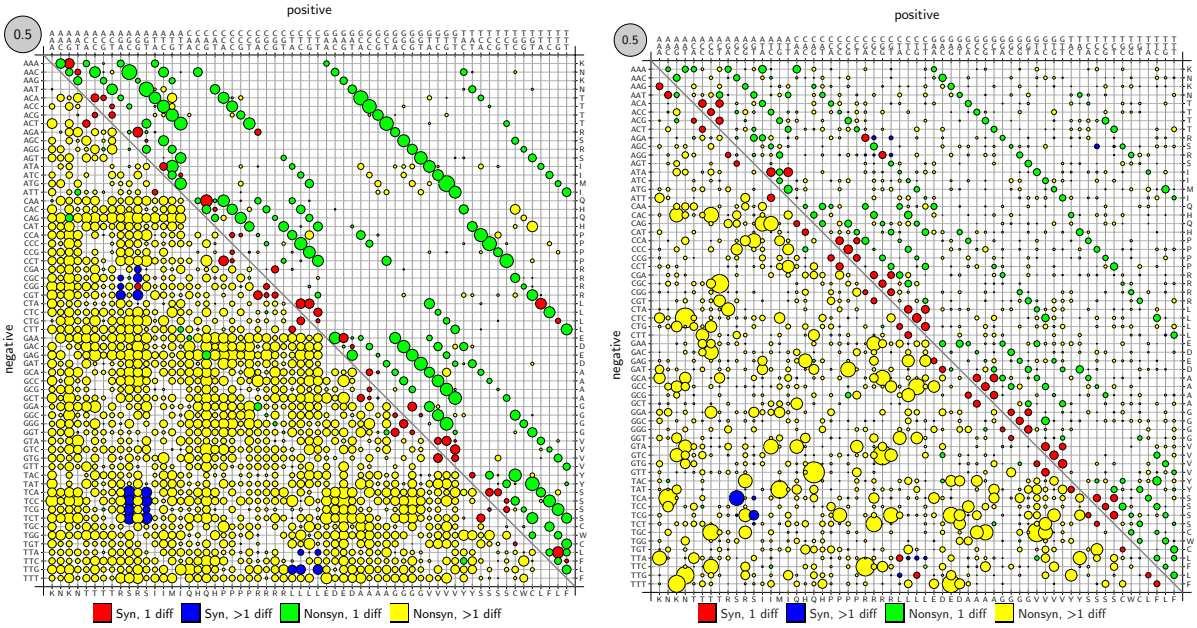
– seems to emerge in the third simulation as well, but not in PC2 but in PC3 (figure 3.12b and table 3.7).

Together, the results of these simulations support our hypothesis. Parameters for which a high variation was simulated have also a high influence on the variation in the simulated data. They clearly show up in the first few principal components of the PCA.

3.2.5. Noise Reduction

Outlier detection

To measure the influence of possible outliers in the data, we first calculated the means of all parameters. Then, all values per parameter p that had a larger distance from the mean than $v \times \mathrm{var}(p)$ were marked. MSAs with more than m marked parameter estimations were defined as outliers and have been discarded from the data set. PCA was then applied on the resulting data, and the eigenvalues and PCs were compared to the original data set. Table 3.8 summarizes the results of this procedure. The eigenvalues of the first principal components increase by discarding more outliers, but we also lose variance in the data. This is represented by the cumulative proportions whose values do not increase when applying stronger filter criteria. We discarded therefore the idea of using filtering outliers to get a



(a) PC2 of original data set **(b)** PC3 of simulation 3

Figure 3.12. – PC2 of the original data set compared to PC3 of the third simulation with positive values in the upper right and negative values in the lower left half of the matrix. The algebraic sign of PCs is arbitrary. The patterns are similar: The diagonal seems to separate substitutions involving two or three nucleotide changes (blue and yellow) from single-nucleotide substitutions (red and green), corresponding to the scalar projection analysis in table 3.7.

stronger signal in the data.

Clustering

Given the assumption that most of the noise results from small differences in otherwise very similar data points, clustering can smooth these differences and maybe help to show a clearer signal in the data. We treated each original MSA estimation as a point in a 1830 dimensional space and calculated a distance matrix for all the 3666 data points. Then k-means clustering was used to generate 1222 cluster of data points. Substitution rates in each cluster have been estimated with `XRate` and PCA was applied on the resulting data set. The results of this procedure can be seen in table 3.9. Since the PCA resulted in only 1222 PCs, the data in this table cannot be compared directly to the results of the original data set. Keeping this difference in mind, the results do not seem to be substantially different. Figures S.1a and S.1b in the supplementary materials show the bubble plots of the first two components. These components were also not substantially different from the ones gained from the original data set. Since the amount of parameter estimations that have to be done with `XRate` increases with the amount of clusters, we decided to skip the computationally expensive task of building clusters before doing the PCA.

Table 3.8. – Standard deviation (*std*) and cumulative proportion (*Cum. Prop.*) of the four first principal components of the *Mammalia* data set. Eigenvalues of PC1 increase by filtering more outliers, but cumulative proportion does not increase. Discarding outliers does not seem to have the desired effect on the data.

		PC1	PC2	PC3	PC4
All MSAs	**std**	6.80429	3.66353	2.90198	2.73076
	Cum. Prop.	0.02530	0.03263	0.03724	0.04131
$v = 20$, $m = 20$	**std**	6.93223	3.19916	2.79772	2.39387
	Cum. Prop.	0.02626	0.03185	0.03613	0.03926
$v = 30$, $m = 10$	**std**	7.02815	3.22833	2.80471	2.40450
	Cum. Prop.	0.02699	0.03269	0.03699	0.04014
$v = 30$, $m = 20$	**std**	6.85776	3.43184	2.82012	2.37658
	Cum. Prop.	0.02570	0.03213	0.03648	0.03957

Table 3.9. – Standard deviations, proportions of variance and cumulative proportions of the five first principal components of the clustered *Mammalia* data. Values for proportion of variance and cumulative proportion cannot be directly compared to the values of the original data set, since this set consists of only 1222 PCs.

	Standard deviation	Proportion of variance	Cumulative Proportion
PC1	6.7744	0.02510	0.02510
PC2	4.28730	0.01000	0.03510
PC3	3.84031	0.00806	0.04318
PC4	3.68575	0.00742	0.05060
PC5	3.63118	0.00721	0.05781

Data Transformation before PCA

Often, before applying PCA, the data set is transformed in some means, the idea being that the data might not be in an "optimal" state for PCA; maybe numerical errors introduce new noise that would not have been present if the data would have been transformed before, or maybe the parameters do not show an optimal (i.e. Gaussian) distribution. We used a log transformation with the intention to obtain a more normal distribution of the input parameters for the PCA. After the transformation, the eigenvalues of the first principal components were much better than the ones of the original *Mammalia* data. Using the scalar projection analysis, PC1 of the log-transformed data could be interpreted as a linear combination of ω, ν and, to a slightly less degree, κ; PC2 showed mostly features of ν and κ, and PC3 and beyond were dominated by physico-chemical features of the amino acids (see table 3.10).

When we did this analysis of the log-transformed data, we did not have the method of scalar projection at hand. At this time, our analysis of PCs consisted mostly in visual analysis of the bubble plots, and we decided to abandon taking the logarithm and chose clarity of interpretation over higher eigenvalues. But it might be beneficial to take a closer look at this transformation. The results of the scalar projection analysis (table 3.10) indicate a higher correlation to the feature vectors we tested; together with the higher eigenvalues this could

Table 3.10. – Scalar products of normed feature vectors and the first PCs of the log-transformed *Mammalia* data set. Higher numbers denote a higher correlation between a parameter and a PC; 1.0 would denote a perfect correlation. All analyzed PCs were found to model to some extent parameters commonly considered in molecular evolution such as the feature parameters discussed in table 3.5. Feature vector names have been abbreviated for better readability (weight, acidity, charge, hydropathy, polarity, random).

PC	κ	ν	ω	**W.**	**Ac.**	**Ch.**	**Hy.**	**Pol.**	**Rnd**
PC1	0.4347	0.7088	0.9270	0.7222	0.1715	0.2613	0.0708	0.0378	0.0070
PC2	0.4965	0.3792	0.0108	0.0143	0.0231	0.0220	0.0614	0.0538	0.0081
PC3	0.2131	0.1491	0.0417	0.1030	0.3419	0.3262	0.0772	0.0952	0.0105
PC4	0.0689	0.0024	0.1003	0.3576	0.5059	0.4933	0.0313	0.1700	0.0125
PC5	0.0301	0.0866	0.1050	0.0166	0.2301	0.2604	0.2680	0.1924	0.0115
PC6	0.0377	0.0247	0.0331	0.0751	0.1926	0.2048	0.0785	0.0733	0.0029

mean taking the log indeed shaped the distributions of the rates to a more Gaussian form, potentially resulting in a new PCM with a better performance.

4. Conclusions

We have applied PCA on a set of 3666 codon substitution matrices estimated from *Mammalia* sequences. Because sensible parameters for codon substitution models were found in the *Mammalia* data set as well as in simulations, it can be concluded that PCA is applicable to this problem despite non-normal distributions of the rate parameters. The first few PCs of our *Mammalia* data set can be identified as linear combinations of previously described features of codon evolution. These features include the ratio of non-synonymous to synonymous substitution rates ω, the ratio of transitions and transversions κ and the rate of multi-nucleotide substitutions ν.

The most relevant component of the first PC is the selection coefficient ω, which is already widely used in parametric codon models. This supports the importance of this parameter. Intra-serine substitutions that require multi-nucleotide changes seem to be an interesting exception to the correspondance of PC1 and ω. Parametric models might benefit by modeling these substitutions separately.

PC2 consists mostly of a linear combination of ν, κ and, to a lesser degree, ω. The most relevant factor is ν. Although the mechanisms of multi-nucleotide substitutions are not yet fully understood, they appear to be important, and one might want to include this parameter in new parametric models of codon substitution. This finding is consistent with the conclusion given by Kosiol et al. (2007). Furthermore, our own model comparisons showed that models including solely ω and ν can compete with models including κ, although it is beneficial to model κ as well.

In the next few PCs, physico-chemical features of the amino acids such as acidity, charge, weight and hydropathy could be identified. More complex codon models would probably benefit from including parameters that describe these features.

We built a new parametric PCA Codon Model (PCM) which uses linear combinations of a given number of principal components to approximate the empirically found variation of codon substitution rate matrices. Our results show that this approach outperforms any other tested model when dealing with sequences that are not too distant from the MSAs used to gain the initial matrices. This also holds when comparing PCM directly to ECM+ω+2κ with an adapted initial rate matrix. We recommend to redo PCA on an appropriate data set and use the results as new initial matrices for our model when applying PCM on species further away from *Mammalia*. LRT testing can be used to determine the optimal amount of principal components used in the model.

A new variant of the ECM model, ECM+ω+ν, was built to test the relevance of the multi-nucleotide substitution rate parameter ν. Its performance was very similar to the ECM+ω+2κ model (see figures 3.1 and 3.3 for details). But on data sets with a high mean and variance in the distribution of κ, the lack of modeling transitions and transversions made ECM+ω+ν less accurate than ECM+ω+2κ. We recommend considering ECM+ω+ν as an addition to the established ECM family only when dealing with sequences that are known to have low κ values.

Scalar projection analysis enabled us to test new hypothesis on PCs that could not be

interpreted solely by graphical means. Especially taking the logarithm of the data matrix before applying PCA seems, analyzed with this method, to be an interesting extension to PCM. Future work could reveal if this approach leads to even more powerful codon models.

5. Acknowledgements

I want to thank Prof. Dr. Gaston Gonnet for giving me the opportunity to write my Master thesis about this very interesting topic. Special thanks go to Dr. Adrian Schneider for his support, advice, motivation and great mentorship throughout all stages of this project.

Also, I want to thank Hannes Röst for his most valuable feedbacks during proofreading my thesis.

And last but not least, thanks to Manuela for supporting me during my whole time at ETHZ.

Nomenclature

κ	Rate of transitions over transversions
ω	Ratio of non-synonymous to synonymous changes
BF	Bayes Factor
DNA	Deoxyribonucleic acid
ECM	Empirical Codon Model
ESS	Effective Sample Size
GTR	General time reversible model
LRT	Likelihood ratio test
MECCM	Mechanistic and Empirical Combined Codon Model
MLE	Maximum Likelihood Estimation
mlogL	Marginal log-likelihood
mRNA	Messenger RNA
OMA	Orthologous matrix project
PC	Principal Component
PCA	Principal component analysis
PCM	PCA Codon Model
RNA	Ribonucleic acid

List of Figures

List of Tables

Bibliography

M. Anisimova and C. Kosiol. Investigating protein-coding sequence evolution with probabilistic codon substitution models. *Molecular Biology and Evolution*, 26(2):255, 2009.

M. Averof, A. Rokas, K. Wolfe, and P. Sharp. Evidence for a high frequency of simultaneous double-nucleotide substitutions. *Science*, 287(5456):1283, 2000.

C. Bishop et al. *Pattern recognition and machine learning*. Springer New York, 2006.

P. Clifford. In discussion of "Approximate Bayesian inference with the weighted likelihood bootstrap" by MA Newton and AE Raferty. *J. Roy. Statist. Soc. B*, 56:34–35, 1994.

D. Collins and T. Jukes. Rates of transition and transversion in coding sequences since the human-rodent divergence. *Genomics*, 20(3):386–396, 1994.

E. Creppy, A. Kane, G. Dirheimer, C. Lafarge-Frayssinet, S. Mousset, and C. Frayssinet. Genotoxicity of ochratoxin A in mice: DNA single-strand break evaluation in spleen, liver and kidney. *Toxicology letters*, 28(1):29–35, 1985.

F. Crick. On protein synthesis. In *Symposia of the Society for Experimental Biology*, volume 12, page 138, 1958.

C. Dessimoz, G. Cannarozzi, M. Gil, D. Margadant, A. Roth, A. Schneider, and G. Gonnet. OMA, a comprehensive, automated project for the identification of orthologs from complete genome data: Introduction and first achievements. *Comparative Genomics*, pages 61–72, 2005.

A. Doron-Faigenboim and T. Pupko. A combined empirical and mechanistic codon model. *Molecular biology and evolution*, 24(2):388, 2007.

A. Drummond and A. Rambaut. BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology*, 7(1):214, 2007.

M. Dudzinski, J. Norris, J. Chmura, and C. Edwards. Repeatability of principal components in samples: normal and non-normal data sets compared. *Multivariate Behavioral Research*, 10(1):109–117, 1975.

J. Felsenstein. Evolutionary trees from DNA sequences: A maximum likelihood approach. *Journal of molecular evolution*, 17(6):368–376, 1981.

D. Flanagan and Y. Matsumoto. *The ruby programming language*. O'Reilly, 2008. ISBN 9780596516178.

C. Garmendia, A. Bernad, J. Esteban, L. Blanco, and M. Salas. The bacteriophage phi 29 dna polymerase, a proofreading enzyme. *Journal of Biological Chemistry*, 267(4):2594, 1992.

C. Geyer. Practical markov chain monte carlo. *Statistical Science*, 7(4):473–483, 1992.

N. Goldman and Z. Yang. A codon-based model of nucleotide substitution for protein-coding DNA sequences. *Molecular Biology and Evolution*, 11(5):725, 1994.

G. H. Gonnet, M. T. Hallett, C. Korostensky, and L. Bernardin. Darwin v. 2.0: An interpreted computer language for the biosciences. *Bioinformatics*, 16(2):101–3, Feb 2000.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417–441, 1933.

D. Jackson. Stopping rules in principal components analysis: A comparison of heuristical and statistical approaches. *Ecology*, 74(8):2204–2214, 1993.

J. Jackson. *A user's guide to principal components*. Wiley-Interscience, 1991.

R. Kass and A. Raftery. Bayes factors. *Journal of the American Statistical Association*, 90 (430), 1995.

K. Katoh, K. Kuma, H. Toh, and T. Miyata. Mafft version 5: Improvement in accuracy of multiple sequence alignment. *Nucleic acids research*, 33(2):511, 2005.

P. S. Klosterman, A. V. Uzilov, Y. R. Bendaña, R. K. Bradley, S. Chao, C. Kosiol, N. Goldman, and I. Holmes. Xrate: A fast prototyping, training and annotation tool for phylogrammars. *BMC Bioinformatics*, 7:428, 2006. doi: 10.1186/1471-2105-7-428.

C. Kosiol, I. Holmes, and N. Goldman. An empirical codon model for protein sequence evolution. *Molecular biology and evolution*, 24(7):1464, 2007.

J. Kyte and R. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of molecular biology*, 157(1):105–132, 1982.

R. Lerner. At the forge: Working with ActiveRecord. *Linux Journal*, 2005(140):12, 2005.

N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, E. Teller, et al. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087, 1953.

S. Muse and B. Gaut. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Molecular Biology and Evolution*, 11(5):715, 1994.

MySQL. MySQL manual, 2004.

Nvidia Company. Compute unified device architecture programming guide. *NVIDIA: Santa Clara, CA*, 2007.

K. Pearson. LIII. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 6*, 2(11):559–572, 1901.

L. Pilon, Y. Langelier, and A. Royal. Herpes simplex virus type 2 mutagenesis: Characterization of mutants induced at the hprt locus of nonpermissive XC cells. *Molecular and cellular biology*, 6(8):2977, 1986.

A. Roth, G. Gonnet, and C. Dessimoz. Algorithm of OMA for large-scale orthology inference. *BMC bioinformatics*, 9(1):518, 2008.

A. Schneider, G. Cannarozzi, and G. Gonnet. Empirical codon substitution matrix. *BMC bioinformatics*, 6(1):134, 2005.

A. Stuart, J. Ord, and S. Arnold. *Kendall's advanced theory of statistics vol. 2a: Classical inference and the linear model*. Oxford University Press; NY: Arnold Publishers, 1999.

M. Suchard and A. Rambaut. Many-core algorithms for statistical phylogenetics. *Bioinformatics*, 25(11):1370, 2009.

M. Suchard, R. Weiss, and J. Sinsheimer. Bayesian selection of continuous-time markov chain evolutionary models. *Molecular Biology and Evolution*, 18(6):1001, 2001.

S. Tavaré. Some probabilistic and statistical problems in the analysis of DNA sequences. *Some mathematical questions in biology—DNA sequence analysis*, 17:57–86, 1986.

R. D. C. Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

L. Torvalds and J. Hamano. Git – fast version control system, 2005.

J. Watson and F. Crick. Molecular structure of deoxypentose nucleic acids. *Nature*, 1953.

S. Whelan and N. Goldman. Estimating the frequency of events that cause multiple-nucleotide changes. *Genetics*, 167(4):2027, 2004.

S. Whelan, P. de Bakker, and N. Goldman. Pandit: A database of protein and associated nucleotide domains with inferred trees. *Bioinformatics*, 19(12):1556, 2003.

Wikipedia and M. P. Ball. DNA chemical structure — Wikipedia, CC-BY-SA, 2009. URL `http://commons.wikimedia.org/w/index.php?title=File:DNA_chemical_structure.svg&oldid=26234525`.

Wikipedia and Mouagip. Table of amino acids, public domain, 2010. URL `http://commons.wikimedia.org/w/index.php?title=File:Aminoacids_table.svg&oldid=43282540`.

E. Witkin. Ultraviolet mutagenesis and inducible DNA repair in Escherichia coli. *Bacteriological reviews*, 40(4):869, 1976.

R. Wood. DNA repair in eukaryotes. *Annual review of biochemistry*, 65(1):135–167, 1996.

Z. Yang. *Computational Molecular Evolution*. Oxford series in ecology and evolution. Oxford University Press, 2006.

Z. Yang. PAML 4: Phylogenetic analysis by maximum likelihood. *Molecular biology and evolution*, 24(8):1586, 2007.

Z. Yang, R. Nielsen, and M. Hasegawa. Models of amino acid substitution and applications to mitochondrial protein evolution. *Molecular Biology and Evolution*, 15(12):1600, 1998.

Z. Yang, R. Nielsen, N. Goldman, and A. Pedersen. Codon-substitution models for heterogeneous selection pressure at amino acid sites. *Genetics*, 155(1):431, 2000.

S. Zoller and A. Schneider. Empirical analysis of the most relevant parameters of codon substitution models. Jan 2010.

S. Supplementary materials

S.1. Codon Substitution Models

S.1.1. Symbols Used

Table S.1. – Symbols used in the description of the models.

Q_{ls}	Element (l, s) of substitution matrix Q
π_s	Element s of Frequency vector π
ω	Ratio of non-synonymous to synonymous substitutions
κ	Ratio of transitions and transversions
κ_{ts}, κ_{tv}	Factors for transition and transversion
n_{ts}, n_{tv}	Number of transitions/transversion in a particular substitution
I_{ls}	Elements (l, s) of initial rate matrix I
ν	Factor for multi-nucleotide substitutions
M_{ls}	Element (l, s) of mean rate matrix M
C^i_{ls}	Element (l, s) of ith principal component C
p_i	Coefficient for principal component i
S_{ls}	Scalar factor for substitution $l \to s$

S.1.2. M0 (Goldman and Yang, 1994)

$$\forall l \neq s : Q_{ls} = \begin{cases} 0 & \text{if } l \text{ or } s \text{ is a stop codon or } l \to s \text{ require} > 1 \text{ nt substitutions} \\ \pi_s & \text{if } l \to s \text{ is a synonymous transversion} \\ \pi_s\kappa & \text{if } l \to s \text{ is a synonymous transition} \\ \pi_s\omega & \text{if } l \to s \text{ is a non-synonymous transversion} \\ \pi_s\kappa\omega & \text{if } l \to s \text{ is a non-synonymous transition} \end{cases}$$

This model is explained in detail in section 1.2.2.

S.1.3. ECM+ω+2κ (Kosiol et al., 2007)

$$Q_{ls} = \begin{cases} 0 & \text{if } l \text{ or } s \text{ are stop codons} \\ \pi_s I_{ls} \kappa_{ts}^{n_{ts}} \kappa_{tv}^{n_{tv}} & \text{if } l \to s \text{ is a synonymous change} \\ \pi_s I_{ls} \kappa_{ts}^{n_{ts}} \kappa_{tv}^{n_{tv}} \omega & \text{if } l \to s \text{ is a non-synonymous change} \end{cases}$$

This model is explained in detail in section 1.2.3.

S.1.4. ECM+ω+ν

$$Q_{ls} = \begin{cases} 0 & \text{if } l \text{ or } s \text{ is a stop codon} \\ \pi_s I_{ls} & \text{if } l \to s \text{ is a synonymous single-nucleotide change} \\ \pi_s I_{ls}\omega & \text{if } l \to s \text{ is a non-synonymous single-nucleotide change} \\ \pi_s I_{ls}\nu & \text{if } l \to s \text{ is a synonymous multi-nucleotide change} \\ \pi_s I_{ls}\omega\nu & \text{if } l \to s \text{ is a non-synonymous multi-nucleotide change} \end{cases}$$

This model is explained in detail in section 2.1.2.

S.1.5. PCM+nC

$$Q_{ls} = \begin{cases} 0 & \text{if } l \text{ or } s \text{ is a stop codon} \\ & \text{or if } M_{ls} + S_{ls} \sum_i^n p_i C_{ls}^i < 0 \\ \pi_s(M_{ls} + S_{ls} \sum_i^n p_i C_{ls}^i) & \text{else} \end{cases}$$

This model is explained in detail in section 2.1.1.

S.2. Principal Components of Mammalia and Simulation 1

Table S.2. – Standard deviations, proportions of variance and cumulative proportions of the five ten principal components of the *Mammalia* data (white background) and the data of simulation 1 (grey background).

	Standard deviation	**Proportion of variance**	**Cumulative Proportion**
PC1	8.91901	0.04347	0.04347
	6.80429	0.02530	0.02530
PC2	3.33587	0.00608	0.04955
	3.66353	0.00733	0.03263
PC3	3.10430	0.00527	0.05482
	2.90198	0.00460	0.03724
PC4	2.88417	0.00455	0.05936
	2.73076	0.00407	0.04131
PC5	2.77313	0.00420	0.06356
	2.54812	0.00355	0.04486
PC6	2.73688	0.00409	0.06766
	2.41919	0.00320	0.04806
PC7	2.69856	0.00398	0.07164
	2.36179	0.00305	0.05110
PC8	2.63938	0.00381	0.07544
	2.30586	0.00291	0.05401
PC9	2.59994	0.00369	0.07914
	2.28203	0.00285	0.05686
PC10	2.56162	0.00359	0.08272
	2.26443	0.00280	0.05966

S.3. Scalar Products of Feature Vectors and Principal Components

Table S.3. – Scalar products of PCs and normed feature vectors. Feature vectors are explained in table 3.5. "MPC1" denotes PC1 of the *Mammalia* data set, "S3PC3" PC3 of simulation 3, "LogPC1" PC1 of the log-transformed *Mammalia* data set. Feature vector names have been abbreviated for better readability (weight, acidity, charge, hydropathy, polarity, random).

PC	κ	ν	ω	**W.**	**Ac.**	**Ch.**	**Hy.**	**Pol.**	**Rnd**
MPC1	0.1089	0.2243	0.6452	0.2308	0.0514	0.0032	0.0212	0.0993	0.0234
MPC2	0.4968	0.7684	0.5050	0.3621	0.0090	0.0384	0.0375	0.0339	0.0029
MPC3	0.0280	0.0545	0.0893	0.0318	0.2402	0.2364	0.1120	0.0929	0.0089
MPC4	0.0651	0.1948	0.1568	0.0867	0.1783	0.1950	0.0832	0.0564	0.0273
MPC5	0.0079	0.0827	0.0384	0.0047	0.0027	0.0078	0.0435	0.0342	0.0054
MPC6	0.0925	0.1041	0.1123	0.1609	0.1265	0.1368	0.0599	0.0035	0.0203
S1PC1	0.1840	0.2199	0.7037	0.3840	0.0587	0.1258	0.0004	0.0698	0.0173
S1PC2	0.2822	0.6809	0.4584	0.3975	0.1031	0.1568	0.0247	0.0452	0.0085
S1PC3	0.1102	0.1025	0.0525	0.0606	0.0220	0.0300	0.0130	0.0023	0.0370
S1PC4	0.0183	0.2098	0.1573	0.1291	0.0346	0.0558	0.0018	0.0083	0.0515
S1PC5	0.0110	0.0921	0.0435	0.0285	0.0115	0.0251	0.0176	0.0018	0.0005
S1PC6	0.1048	0.0238	0.0141	0.0090	0.0046	0.0124	0.0369	0.0102	0.0011
S2PC1	0.2528	0.2245	0.7069	0.3579	0.0368	0.1036	0.0146	0.0818	0.0138
S2PC2	0.2654	0.8153	0.5301	0.4059	0.0951	0.1415	0.0179	0.0598	0.0214
S2PC3	0.1145	0.1232	0.0720	0.0482	0.0196	0.0211	0.0190	0.0210	0.0316
S2PC4	0.1231	0.0604	0.0215	0.0230	0.0077	0.0042	0.0121	0.0007	0.0189
S2PC5	0.0107	0.0347	0.0291	0.0388	0.0368	0.0295	0.0114	0.0587	0.0174
S2PC6	0.0278	0.0919	0.0540	0.0182	0.0276	0.0394	0.0180	0.0506	0.0341
S3PC1	0.2232	0.2072	0.7003	0.4269	0.0826	0.1521	0.0225	0.0472	0.0181
S3PC2	0.4203	0.0045	0.0150	0.2065	0.0539	0.0807	0.0702	0.0839	0.0257
S3PC3	0.1862	0.3571	0.2404	0.1761	0.0109	0.0207	0.0512	0.0636	0.0272
S3PC4	0.0483	0.0288	0.0109	0.0176	0.0197	0.0161	0.0170	0.0397	0.0000
S3PC5	0.0560	0.3081	0.2001	0.1455	0.0676	0.0857	0.0180	0.0028	0.0026
S3PC6	0.0598	0.1703	0.1061	0.1096	0.0027	0.0275	0.0226	0.0132	0.0080
RND	0.0160	0.0116	0.0068	0.0197	0.0434	0.0414	0.0142	0.0082	0.0371
LogPC1	0.4347	0.7088	0.9270	0.7222	0.1715	0.2613	0.0708	0.0378	0.0070
LogPC2	0.4965	0.3792	0.0108	0.0143	0.0231	0.0220	0.0614	0.0538	0.0081
LogPC3	0.2131	0.1491	0.0417	0.1030	0.3419	0.3262	0.0772	0.0952	0.0105
LogPC4	0.0689	0.0024	0.1003	0.3576	0.5059	0.4933	0.0313	0.1700	0.0125
LogPC5	0.0301	0.0866	0.1050	0.0166	0.2301	0.2604	0.2680	0.1924	0.0115
LogPC6	0.0377	0.0247	0.0331	0.0751	0.1926	0.2048	0.0785	0.0733	0.0029

S.4. Summarized Marginal Likelihood Results

Table S.4. – Summarized results of the model comparison. Values in the first rows (white background) denote the number of maximum marginal likelihood values for all MSAs as absolute counts and percentages of the whole data set family. In the second rows (grey background), only significant results are considered (difference to the second best likelihood $d > 3$). All models included frequencies estimated from the alignments, the $+F$ notation has been omitted due to readability. PCM+8C has not been tested with the Pandit MSAs. ECM+F+ω+2κ and ECM+F+ω+ν were also estimated with a initial rate matrix gained from the *Mammalia* data set, but in all measurements either the PCM+F+nC or the ECM+F+ω+2κ with the Kosiol matrix performed better; these models have therefore not been included in this table. Pandit family PF01287 showed twice the same result, once for ECM+ω+ν and once for ECM+ω+2κ; it has been counted for both of the models, thus increasing the total number of counts from 183 to 184 in this particular set.

	M0	**PCM+2C**	**PCM+8C**	**ECM+ω+ν**	**ECM+ω+2κ**
Pandit:	1 (0.5%)	22 (12.0%)	N/A	58 (31.5%)	103 (56.0%)
	0 (0.0%)	20 (10.9%)	N/A	6 (3.3%)	40 (21.7%)
Mammalia:	0	20 (10.1%)	179 (89.9%)	0	0
	0	4 (2.0%)	149 (74.9%)	0	0
Archaea:	0	0	0	5 (33.3%)	10 (66.7%)
	0	0	0	4 (26.7%)	4 (26.7%)
Cyano-bacteria:	0	0	0	14 (93.3%)	1 (6.7%)
	0	0	0	13 (86.7%)	0 (0.0%)
Other Eukaryota:	0	0	0	8 (57.1%)	6 (42.9%)
	0	0	0	4 (28.6%)	1 (7.1%)
Vertebrata:	0	1 (11.1%)	8 (88.9%)	0	0
	0	0 (0.0%)	8 (88.9%)	0	0

S.5. Complete Marginal Log-likelihood Results

The following tables include the estimated marginal log-likelihood values for all MSAs. The frequencies have always been estimated from the MSA; the notation "+F" for the models has been discarded due to spacing issues. "ECM+ω+2κ (Mam)" denotes the ECM+ω+2κ model, using the initial rate matrix estimated from 3666 *Mammalia* MSA in the first step of the PCA process. Highest values per MSA are marked with a star; additional stars denote the significance to the second best value. Evidence for the higher value is positive (two stars) if the difference d between best and second best marginal logL values is $3 < d < 20$, strong (three stars) if $20 < d < 150$ and very strong (four stars) if $d > 150$. Differences below 3 are not significant (Kass and Raftery, 1995). At the end of each table, a summary is included with two rows: First, the total count of highest likelihood values is given as absolute numbers and as percentages on the whole data set; second, only those highest values are counted where the result is significant (i.e. $d > 3$).

Table S.5. – Summarized results of only the three ECM models. Values in the first rows (white background) denote the number of maximum marginal likelihood values for all MSAs as absolute counts and percentages of the whole data set family. In the second rows (grey background), only significant results are considered (difference to the second best likelihood $d > 3$). All models included frequencies estimated from the alignments, the $+F$ notation has been omitted due to readability. "ECM+F+ω+ν (Mam)" and "ECM+F+ω+2κ (Mam)" denote the use of an initial rate matrix estimated in the first steps of the PCA on the *Mammalia* data instead of the original one from Kosiol et al.. Pandit MSAs have not been estimated with these two model variations. Pandit family PF01287 showed twice the same result, once for ECM+ω+ν and once for ECM+ω+2κ; it has been counted for both of the models, thus increasing the total number of counts from 183 to 184 in this particular set.

	ECM+ω+ν	**ECM+ω+ν (Mam)**	**ECM+ω+2κ**	**ECM+ω+2κ (Mam)**
Pandit:	65 (35.3%)	0	119 (64.7%)	0
	9 (4.9%)	0	52 (28.3%)	0
Mammalia:	0	28 (14.0%)	77 (38.5%)	95 (47.5%)
	0	1 (0.5%)	69 (34.5%)	44 (22%)
Archaea:	5 (33.3%)	0	10 (66.7%)	0
	4 (26.7%)	0	4 (26.7%)	0
Cyano-bacteria:	14 (93.3%)	0	1 (6.7%)	0
	13 (86.7%)	0	0	0
Other Eukaryota:	8 (57.1%)	0	6 (42.9%)	0
	4 (28.6%)	0	1 (7.1%)	0
Vertebrata:	0	0	8 (88.9%)	1 (11.1%)
	0	0	7 (77.8%)	1 (11.1%)

Table S.6. – Estimated mlogL values for all Pandit MSAs.

MSA	**M0**	**PCM+2C**	**ECM+ν**	**ECM+ω+2κ**
PF01201	−15951.20	−15268.60	−15176.70	−15162.70**
PF01202	−49969.10	−47783.60	−46659.60	−46638.00***
PF01203	−4380.13	−4258.50	−4217.70	−4217.19*
PF01204	−37677.30	−36024.80	−35501.10	−35500.40*
PF01205	−4490.14	−4308.71	−4236.69	−4236.33*
PF01206	−1725.53	−1683.36	−1661.69*	−1661.80
PF01207	−19390.40	−18804.60	−18686.90	−18681.70**
PF01208	−16719.50	−16122.10	−15943.50	−15941.50*
PF01209	−4061.57	−3900.02	−3858.53	−3857.67*
PF01210	−24509.00	−23725.70	−23497.80	−23491.30**
PF01212	−48404.50	−46129.30	−45015.60	−44974.00***
PF01213	−9711.89	−9412.78	−9357.31*	−9358.53
PF01214	−3808.92	−3611.29	−3572.20*	−3573.10
PF01217	−3373.44	−3241.43	−3221.08*	−3221.60
PF01218	−5746.02	−5478.15	−5406.45	−5405.70*
PF01219	−2308.30	−2258.66	−2237.95*	−2238.21
PF01220	−3241.81	−3088.97	−3046.26	−3041.79**
PF01221	−1598.57	−1531.68	−1515.37	−1512.91*
PF01222	−7803.75	−7571.96	−7515.68**	−7518.89

MSA	**M0**	**PCM+2C**	**ECM+ν**	**ECM+ω+2κ**
PF01223	−41947.50	−40431.90	−40196.90	−40196.40*
PF01226	−5667.00	−5548.60	−5509.17*	−5509.85
PF01227	−4646.57	−4484.19	−4402.09*	−4404.68
PF01228	−3001.48	−2937.05	−2894.08	−2893.92*
PF01229	−6724.47	−6590.68**	−6608.41	−6602.55
PF01230	−7436.82	−7160.37	−7098.83	−7097.15*
PF01231	−5436.80	−5329.96	−5295.30	−5294.41*
PF01232	−7396.11	−7142.81	−7025.81	−7023.34*
PF01233	−2406.75	−2319.95	−2292.35*	−2292.54
PF01234	−5238.04	−5068.60	−5040.50*	−5041.39
PF01237	−9753.20	−9573.47	−9513.51*	−9513.94
PF01238	−9776.23	−9431.80	−9345.84	−9345.53*
PF01241	−2337.02	−2280.92	−2266.03	−2265.81*
PF01244	−50951.30	−48702.90	−48025.30	−48022.10**
PF01245	−2919.93	−2839.79	−2781.54	−2781.27*
PF01246	−1280.02	−1213.00	−1203.19*	−1204.67
PF01248	−13817.40	−13222.50	−13070.20	−13069.60*
PF01249	−958.18	−900.46	−887.79	−886.19*
PF01250	−3494.27	−3425.82	−3370.52*	−3371.49
PF01251	−2624.05	−2496.34	−2466.16*	−2466.82
PF01253	−5603.37	−5428.11	−5368.44	−5366.40*
PF01255	−5984.36	−5774.50	−5702.05	−5701.57*
PF01256	−8230.07	−8050.28	−7979.22	−7974.51**
PF01257	−2548.69	−2480.90	−2449.87*	−2450.92
PF01258	−1582.69	−1563.30	−1540.58	−1539.30*
PF01259	−7312.33	−7116.69	−7024.61	−7022.46*
PF01263	−72943.60	−70167.10	−69280.30	−69270.10**
PF01264	−14514.50	−13864.90	−13653.30**	−13666.40
PF01265	−4731.14	−4580.61	−4542.22*	−4543.93
PF01267	−4543.58	−4435.56	−4367.15*	−4367.78
PF01268	−12734.20	−12374.30	−12137.30	−12135.70*
PF01270	−9760.11	−9439.31	−9351.80*	−9351.81
PF01271	−14401.80	−13997.10***	−14291.80	−14253.40
PF01272	−2012.80	−1970.30	−1932.37	−1931.98*
PF01273	−20493.30	−19867.90	−19931.90	−19867.60*
PF01274	−25848.90	−24623.20	−24359.20	−24357.00*
PF01275	−4868.37	−4702.94*	−4707.39	−4705.04
PF01276	−12950.40	−12478.30	−12353.80	−12353.30*
PF01277	−5160.88	−5000.74	−4981.47*	−4981.62
PF01278	−6616.37	−6464.07	−6452.37	−6440.83**
PF01279	−1800.43	−1759.92**	−1767.80	−1765.17
PF01280	−2983.50	−2905.87	−2870.90	−2870.85*
PF01281	−1671.52	−1606.18	−1584.88	−1584.50*
PF01282	−1596.83	−1564.37	−1547.63	−1546.30*
PF01283	−1837.98	−1768.55	−1755.19*	−1755.72
PF01284	−22262.60	−21612.90***	−21767.80	−21760.50
PF01285	−7986.87	−7841.75	−7839.35*	−7842.41
PF01286	−592.49	−574.49	−567.80*	−569.11
PF01287	−1267.70	−1231.17	−1197.53*	−1197.53*
PF01288	−7134.91	−6865.42	−6805.24	−6804.07*
PF01289	−7705.39	−7436.02	−7331.15	−7328.76*
PF01291	−3001.61	−2932.68***	−2996.53	−2963.49

MSA	**M0**	**PCM+2C**	**ECM+ν**	**ECM+ω+2κ**
PF01292	−17629.00	−17138.80	−16936.00	−16928.80**
PF01293	−14305.80	−13619.60	−13375.00	−13373.00*
PF01294	−3181.01	−3070.94	−3037.68	−3033.40**
PF01295	−12740.80	−12235.00	−12189.10	−12188.30*
PF01297	−21384.30	−20561.90	−20436.10	−20429.50**
PF01299	−9896.39	−9648.29***	−9699.27	−9672.19
PF01300	−12240.00	−11854.10	−11755.70	−11755.10*
PF01301	−32364.20	−30906.20	−30563.70	−30563.00*
PF01302	−9248.65	−8809.31	−8663.81	−8660.23**
PF01303	−4000.29	−3848.63	−3810.39	−3807.06**
PF01304	−566.20	−561.16**	−569.43	−571.13
PF01305	−3098.47	−3028.58	−2977.65	−2971.76**
PF01306	−5314.46	−5126.83	−5098.23	−5097.57*
PF01307	−9289.40	−8952.56	−8952.42*	−8952.71
PF01309	−1622.83	−1646.72	−1663.91	−1618.91**
PF01310	−5452.04	−5324.10	−5289.23	−5288.80*
PF01311	−13705.20	−13432.50	−13296.70*	−13297.80
PF01312	−18551.80	−17988.20	−17870.20	−17865.20**
PF01313	−3478.97	−3364.62	−3316.31	−3315.88*
PF01314	−12215.00	−11842.80	−11764.00*	−11766.70
PF01315	−10703.50	−10236.00	−10079.60	−10073.00**
PF01316	−2447.01	−2397.56	−2386.57*	−2388.41
PF01320	−1433.04	−1366.04**	−1370.02	−1370.41
PF01322	−1882.73	−1847.01	−1829.43*	−1830.79
PF01323	−17522.90	−16876.70	−16835.00*	−16836.40
PF01325	−2009.92	−1958.35	−1921.88	−1921.57*
PF01326	−26881.40	−25913.10	−25688.40	−25685.60*
PF01329	−1810.23	−1767.59	−1756.75	−1754.37*
PF01330	−6248.30	−6072.54	−5969.70*	−5970.27
PF01331	−7605.65	−7424.79	−7389.74*	−7389.88
PF01333	−3517.18	−3418.30	−3412.06	−3411.86*
PF01335	−10497.10	−10038.60	−10031.80	−10030.80*
PF01338	−2523.79	−2428.98	−2412.45*	−2413.71
PF01339	−4778.85	−4600.72	−4535.36	−4533.15*
PF01341	−13236.40	−12404.90	−12163.90	−12158.80**
PF01342	−2743.94	−2657.69	−2641.33*	−2642.32
PF01343	−10368.20	−9986.61	−9872.40	−9870.49*
PF01345	−8991.40	−8793.78	−8732.19	−8723.91**
PF01346	−3623.39	−3487.36*	−3489.38	−3487.61
PF01347	−85816.70	−83027.90	−82579.90	−82562.10**
PF01349	−4894.40	−4768.74	−4743.77*	−4744.02
PF01350	−2657.20	−2588.20	−2554.65*	−2555.42
PF01351	−9293.87	−8948.85	−8863.81	−8855.56**
PF01352	−6707.40	−6304.68****	−6493.48	−6645.02
PF01353	−11521.10	−11240.90	−11169.40	−11161.30**
PF01354	−3397.91	−3287.26	−3218.20	−3216.65*
PF01357	−10462.30	−9925.84	−9751.41	−9748.77*
PF01358	−5320.32	−5181.74	−5163.85*	−5163.98
PF01359	−10470.80	−10098.80	−10049.90	−10043.10**
PF01361	−5811.55	−5617.87	−5566.71*	−5569.23
PF01363	−4680.22	−4551.88	−4519.48	−4517.23*
PF01365	−9047.05	−8755.60	−8718.39	−8716.66*

MSA	**M0**	**PCM+2C**	**ECM+ν**	**ECM+ω+2κ**
PF01366	−30599.50	−29382.80	−29219.10	−29212.50**
PF01367	−8571.11	−8196.54	−8034.38	−8026.74**
PF01369	−7031.81	−6795.47	−6740.54	−6740.45*
PF01371	−4530.06	−4415.21	−4362.57*	−4364.02
PF01373	−6045.97	−5852.83	−5792.02*	−5792.85
PF01374	−3926.19	−3796.37	−3764.57*	−3765.46
PF01375	−2402.12	−2358.30	−2343.10	−2341.04*
PF01376	−629.27*	−629.85	−630.51	−631.04
PF01378	−638.70	−604.71	−604.48	−602.57*
PF01379	−12161.90	−11778.30	−11599.80	−11595.90**
PF01380	−18694.60	−18219.40	−18067.80*	−18068.40
PF01382	−1559.53	−1529.58	−1514.23	−1513.51*
PF01383	−3466.54	−3400.49	−3352.18*	−3354.45
PF01384	−32315.90	−31610.20	−31387.60	−31376.40**
PF01385	−13761.80	−13306.40	−13182.00*	−13183.10
PF01386	−1831.97	−1804.99	−1771.18*	−1771.41
PF01388	−3826.05	−3734.20	−3706.13	−3705.32*
PF01389	−6345.71	−6019.97	−5958.74	−5949.25**
PF01390	−6890.10	−6723.27***	−6758.01	−6744.03
PF01391	−6551.78	−6453.17	−6341.46**	−6344.76
PF01392	−12055.70	−11599.10	−11580.40	−11564.80**
PF01393	−1496.11	−1458.45	−1448.29	−1445.11**
PF01394	−7556.98	−7301.83	−7146.58	−7142.62**
PF01395	−34652.80	−33484.30***	−33535.50	−33512.20
PF01396	−3106.26	−2928.59	−2906.37	−2903.97*
PF01397	−39436.60	−38546.10	−38583.10	−38538.40**
PF01398	−13230.30	−12739.50	−12604.80	−12602.20*
PF01399	−29729.00	−28763.50	−28486.60*	−28486.70
PF01400	−13874.90	−13271.30	−13162.80**	−13166.80
PF01401	−6311.52	−6118.16***	−6184.98	−6142.95
PF01402	−6418.22	−6301.41	−6189.71	−6185.93**
PF01403	−34351.40	−33121.30***	−33198.40	−33195.50
PF01404	−6441.03	−6155.14**	−6158.20	−6200.90
PF01407	−5148.40	−5012.08	−4952.96*	−4953.29
PF01408	−19121.00	−18481.70	−18270.50	−18267.50*
PF01410	−5796.09	−5502.00	−5491.48**	−5494.66
PF01412	−8635.52	−8181.41	−8156.01*	−8158.70
PF01413	−7191.79	−6813.94	−6760.66	−6758.38*
PF01414	−2723.09	−2591.71**	−2601.69	−2601.38
PF01415	−1664.95	−1649.72**	−1660.50	−1660.71
PF01417	−7923.94	−7583.78	−7538.64*	−7540.27
PF01418	−3071.99	−2976.96	−2961.56*	−2961.61
PF01419	−17090.50	−16544.20	−16484.00	−16471.70**
PF01421	−17644.50	−17127.10	−17109.30	−17078.00***
PF01422	−1304.95	−1296.01	−1286.82	−1283.35**
PF01424	−6659.96	−6488.48	−6442.13*	−6443.23
PF01426	−19802.00	−19162.90	−19102.10*	−19103.60
PF01427	−3482.12	−3403.36**	−3407.87	−3408.91
PF01428	−1184.65	−1145.39	−1143.39	−1143.31*
PF01429	−5232.22	−5114.40	−5092.91	−5092.82*
PF01431	−12493.70	−12044.70	−11992.30	−11991.60*
PF01433	−45184.20	−43502.70	−43189.20*	−43189.40

MSA	**M0**	**PCM+2C**	**ECM+ν**	**ECM+ω+2κ**
PF01434	−19011.90	−18232.50	−17927.70	−17927.10*
PF01435	−13674.00	−13375.90	−13170.50**	−13174.10
PF01436	−9408.67	−9211.46	−9004.96*	−9006.94
PF01439	−4094.15	−3858.63***	−3930.07	−3914.86
PF01440	−6974.32	−6782.95	−6774.03	−6764.07**
PF01441	−14933.20	−14479.20***	−14819.00	−14692.10
PF01442	−12221.10	−11559.90***	−11687.00	−11641.00
PF01443	−51103.40	−48979.20	−48534.20	−48511.90***
Total:	1(0.5%)	22(12.0%)	58(31.5%)	103(56.0%)
	0(0.0%)	20(10.9%)	6(3.3%)	40(21.7%)

Table S.7. – Estimated mlogL values for all *Mammalia* MSAs.

MSA	**M0**	**PCM+2C**	**PCM+8C**	**ECM+ν**	**ECM+ν (Mam)**	**ECM+ω+2κ**	**ECM+ω+2κ (Mam)**
mam-aa	−39022.20	−37038.70	−37028.70**	−38136.90	−37829.40	−37953.10	−37783.50
mam-ab	−12023.00	−11755.30	−11722.30***	−12199.00	−12014.50	−11935.90	−12005.70
mam-ac	−8597.63	−8380.21	−8359.05***	−8716.80	−8545.57	−8545.51	−8542.94
mam-ad	−7938.52	−7572.84	−7548.54***	−7860.81	−7766.64	−7788.67	−7755.17
mam-ae	−7199.71	−7032.48	−7008.03***	−7196.82	−7202.16	−7126.94	−7199.57
mam-af	−14020.40	−13686.70	−13624.30***	−14115.50	−13889.40	−13874.50	−13886.60
mam-ag	−5062.09	−5003.06	−5000.09*	−5110.08	−5117.65	−5040.23	−5115.11
mam-ah	−7735.77	−7444.10	−7431.48**	−7688.11	−7624.34	−7603.20	−7624.58
mam-ai	−10727.30	−10368.70	−10358.80**	−10799.90	−10636.50	−10588.90	−10632.60
mam-aj	−15593.40	−15259.00	−15222.70***	−16145.80	−15626.30	−15719.90	−15600.60
mam-ak	−9267.91	−9032.64	−9024.40**	−9448.47	−9305.23	−9244.31	−9288.12
mam-al	−9998.41	−9756.30	−9725.78***	−10053.50	−9915.32	−9909.87	−9914.18
mam-am	−6915.11	−6734.25*	−6736.83	−6939.03	−6899.47	−6872.87	−6899.19
mam-an	−12491.90	−12055.00	−12023.40***	−12582.30	−12234.00	−12451.10	−12230.90
mam-ao	−12533.30	−12130.30	−12129.50*	−12496.90	−12448.30	−12409.30	−12447.50
mam-ap	−4547.50	−4477.54**	−4492.09	−4666.18	−4640.47	−4558.45	−4639.90
mam-aq	−6933.40	−6762.29	−6749.94**	−7062.51	−6898.97	−6933.25	−6893.30
mam-ar	−14494.90	−14164.90	−14152.30**	−14858.70	−14446.80	−14630.70	−14443.00
mam-as	−7513.33	−7312.26	−7311.58*	−7515.01	−7432.20	−7462.15	−7428.06
mam-at	−7364.00	−7056.63	−6976.40***	−7476.07	−7298.85	−7382.64	−7295.91
mam-au	−8244.65	−7882.73	−7882.18*	−8210.47	−8063.87	−8159.42	−8060.85
mam-av	−6864.34	−6599.65*	−6600.63	−6871.54	−6821.11	−6805.41	−6820.67
mam-aw	−7606.13	−7364.99	−7353.42**	−7577.65	−7535.23	−7496.07	−7536.68
mam-ax	−8446.23	−8237.02	−8228.60**	−8600.68	−8531.91	−8408.45	−8522.22
mam-ay	−9107.15	−8667.72	−8662.43**	−8972.36	−8870.49	−8935.13	−8866.70
mam-az	−9677.09	−9400.74	−9399.66*	−9881.53	−9638.60	−9704.12	−9631.02
mam-ba	−9998.05	−9834.37	−9822.50**	−10084.40	−10082.90	−9947.98	−10083.80
mam-bb	−10316.50	−9988.24	−9975.72**	−10532.70	−10232.30	−10322.10	−10231.80
mam-bc	−8555.74	−8329.53	−8328.69*	−8639.80	−8508.66	−8487.25	−8505.08
mam-bd	−6643.21	−6519.36	−6504.45**	−6734.04	−6642.43	−6678.24	−6642.45
mam-be	−8326.58	−8092.17	−8074.30**	−8538.69	−8319.13	−8385.12	−8318.57
mam-bf	−10796.20	−10420.30	−10374.90***	−10623.50	−10581.80	−10539.80	−10580.20
mam-bg	−8478.46	−8084.43	−8059.55***	−8395.84	−8206.63	−8354.64	−8209.26
mam-bh	−5936.28	−5808.76	−5798.79**	−6012.79	−5930.01	−5934.88	−5929.70
mam-bi	−9281.74	−9050.78	−9021.71***	−9233.93	−9156.98	−9162.86	−9152.81

MSA	**M0**	**PCM+2C**	**PCM+8C**	**ECM+ν**	**ECM+ν (Mam)**	**ECM+ω+2κ**	**ECM+ω+2κ (Mam)**
mam-bj	−6874.33	−6730.60	−6721.92**	−6908.90	−6946.68	−6815.68	−6944.99
mam-bk	−5533.85	−5470.46	−5435.39***	−5604.69	−5558.25	−5554.89	−5558.60
mam-bl	−6848.02	−6634.54	−6621.70**	−6768.02	−6779.79	−6724.88	−6773.34
mam-bm	−5043.09	−4946.87	−4944.70*	−5121.90	−5095.82	−5076.32	−5096.41
mam-bn	−10837.30	−10543.40	−10535.70**	−10922.20	−10792.20	−10805.20	−10793.60
mam-bo	−8033.60	−7928.12	−7921.54**	−8196.32	−8140.15	−8130.54	−8141.18
mam-bp	−6895.14	−6649.61	−6636.93**	−6888.66	−6841.27	−6812.90	−6840.82
mam-bq	−5097.29	−4940.97	−4928.44**	−5053.31	−4997.08	−5029.19	−4995.11
mam-br	−5669.79	−5582.47	−5576.37**	−5868.57	−5731.27	−5717.55	−5719.53
mam-bs	−4664.09	−4626.66	−4605.90***	−4713.82	−4701.17	−4677.18	−4700.37
mam-bt	−9494.56	−9094.64	−9050.20***	−9207.89	−9244.08	−9173.15	−9223.51
mam-bu	−5677.79	−5597.28	−5574.28***	−5731.79	−5728.02	−5658.91	−5725.54
mam-bv	−6732.51	−6456.63	−6447.59**	−6757.23	−6570.39	−6713.08	−6569.31
mam-bw	−8223.97	−8011.80*	−8012.24	−8356.25	−8206.90	−8287.17	−8206.44
mam-bx	−10007.60	−9728.76*	−9729.11	−10135.50	−9895.42	−10014.30	−9896.04
mam-by	−6078.66	−5954.57*	−5956.18	−6176.34	−6109.02	−6108.40	−6108.59
mam-bz	−6398.63	−6228.06	−6227.68*	−6366.72	−6474.53	−6319.96	−6473.69
mam-ca	−6078.76	−5834.69*	−5837.41	−6044.50	−5971.28	−5994.08	−5969.77
mam-cb	−5356.30	−5313.07	−5308.39**	−5451.75	−5437.89	−5405.12	−5439.09
mam-cc	−6174.18	−6030.73	−6030.56*	−6184.40	−6127.23	−6160.95	−6126.40
mam-cd	−5398.29	−5243.58	−5232.72**	−5425.14	−5364.17	−5379.23	−5365.20
mam-ce	−5957.89	−5825.57	−5812.84**	−6036.74	−5893.92	−5960.40	−5894.64
mam-cf	−5521.52	−5355.83	−5349.09**	−5639.11	−5563.00	−5523.14	−5558.69
mam-cg	−6722.14	−6580.32**	−6583.54	−6871.60	−6712.30	−6733.19	−6710.56
mam-ch	−7646.03	−7448.57	−7439.01**	−7794.58	−7599.31	−7709.01	−7597.93
mam-ci	−5847.70	−5832.83	−5785.76***	−5947.90	−5903.51	−5862.79	−5903.52
mam-cj	−5347.25	−5295.49	−5267.87***	−5478.50	−5380.31	−5383.16	−5381.32
mam-ck	−7790.59	−7679.08	−7665.42**	−7866.63	−7793.55	−7771.79	−7792.73
mam-cl	−6978.35	−6795.90	−6790.21**	−6967.04	−6925.71	−6950.15	−6919.25
mam-cm	−5713.48	−5537.00	−5531.89**	−5709.26	−5658.64	−5658.36	−5659.74
mam-cn	−6486.67	−6333.22	−6330.61*	−6505.55	−6481.70	−6454.07	−6483.29
mam-cp	−10813.30	−10527.40	−10516.40**	−11083.40	−10777.00	−10865.20	−10777.00
mam-cq	−10566.40	−10386.90	−10350.70***	−10714.40	−10599.60	−10558.10	−10599.30
mam-cr	−5921.96	−5763.56	−5711.32***	−5991.93	−5909.86	−5945.77	−5908.71
mam-cs	−5554.84	−5458.45*	−5459.22	−5702.66	−5571.17	−5597.20	−5565.01
mam-ct	−5610.88	−5367.30	−5367.15*	−5528.30	−5450.35	−5520.53	−5449.16
mam-cu	−6744.28	−6545.05*	−6545.78	−6791.48	−6698.65	−6746.22	−6698.17
mam-cv	−11477.10	−11135.00	−11130.10**	−11633.40	−11446.50	−11441.60	−11441.20
mam-cw	−4678.67	−4418.82*	−4421.31	−4513.51	−4502.92	−4507.55	−4494.81
mam-cx	−5029.56	−4952.03	−4948.85**	−5042.59	−5013.15	−4996.14	−5009.68
mam-cy	−4079.52	−4003.07*	−4003.39	−4210.02	−4062.17	−4102.49	−4060.31
mam-cz	−7946.45	−7673.56	−7597.54***	−8007.42	−7887.26	−7918.74	−7885.63
mam-da	−4712.40	−4660.77	−4634.25***	−4895.65	−4740.97	−4783.90	−4740.33
mam-db	−6650.23	−6453.42**	−6458.04	−6667.58	−6588.86	−6596.27	−6590.15
mam-dc	−6344.84	−6252.36	−6234.76**	−6435.20	−6367.56	−6287.10	−6360.89
mam-dd	−9347.83	−9104.83	−9096.11**	−9515.54	−9346.41	−9242.18	−9323.36
mam-de	−5529.47	−5421.64	−5418.05**	−5609.43	−5573.17	−5541.00	−5570.36
mam-df	−7441.74	−7191.36	−7184.76**	−7530.71	−7326.18	−7388.76	−7325.46
mam-dg	−5244.01	−5116.91	−5093.29***	−5250.98	−5206.99	−5202.53	−5206.56
mam-dh	−6273.77	−6018.34	−6016.93*	−6203.32	−6143.02	−6152.19	−6142.84
mam-di	−5091.43	−4911.28	−4906.12**	−5066.05	−5019.50	−5039.13	−5012.49

MSA	**M0**	**PCM+2C**	**PCM+8C**	**ECM+ν**	**ECM+ν (Mam)**	**ECM+ω+2κ**	**ECM+ω+2κ (Mam)**
mam-dj	−5687.70	−5562.18	−5545.34**	−5737.25	−5708.87	−5671.61	−5709.62
mam-dk	−7592.45	−7372.58	−7369.93*	−7603.48	−7509.80	−7557.65	−7508.24
mam-dl	−4938.53	−4921.56	−4902.32**	−5051.09	−5021.88	−4974.66	−5022.59
mam-dm	−7017.48	−6765.34	−6756.48**	−7001.92	−6848.65	−6957.37	−6848.52
mam-dn	−6035.07	−5835.29	−5832.80*	−5988.95	−5992.54	−5959.93	−5992.34
mam-do	−5627.24	−5480.65	−5475.31**	−5710.29	−5603.17	−5657.15	−5603.61
mam-dp	−4920.93	−4830.98	−4817.24**	−4941.54	−4890.44	−4879.01	−4890.10
mam-dq	−5275.19	−5110.51	−5105.09**	−5373.91	−5228.72	−5270.55	−5220.98
mam-dr	−6450.87	−6302.27	−6292.40**	−6612.02	−6451.11	−6477.06	−6449.14
mam-ds	−36477.80	−35141.50	−35009.90***	−35750.80	−35572.70	−35571.30	−35540.30
mam-dt	−21401.80	−20734.00	−20717.20**	−21577.60	−21420.80	−21210.00	−21413.80
mam-du	−13909.40	−13444.30	−13427.40**	−13806.60	−13681.00	−13677.90	−13678.20
mam-dv	−15757.50	−15285.20	−15266.80**	−16094.80	−15654.60	−15821.60	−15653.40
mam-dw	−9475.03	−9101.78	−9061.11***	−9563.41	−9208.58	−9438.23	−9201.56
mam-dx	−12707.20	−12468.90	−12395.80***	−12967.10	−12701.60	−12716.30	−12698.20
mam-dy	−11732.20	−11522.20	−11502.90**	−11857.50	−11754.40	−11733.80	−11753.70
mam-dz	−18804.00	−18487.10	−18469.20**	−19430.20	−18794.50	−18935.00	−18783.00
mam-ea	−22403.10	−21237.50	−21103.20***	−22308.70	−21628.30	−22106.90	−21620.00
mam-eb	−9904.94	−9846.66	−9825.79***	−10153.30	−9984.27	−10126.00	−9970.63
mam-ec	−9138.52	−8864.09*	−8866.86	−9156.02	−9024.28	−9054.07	−9022.04
mam-ed	−15590.40	−15048.10	−15032.40**	−15664.40	−15360.90	−15487.40	−15361.90
mam-ee	−15374.90	−14931.50	−14901.10***	−15834.10	−15289.30	−15474.10	−15282.60
mam-ef	−13251.30	−12913.20	−12904.70**	−13453.60	−13189.70	−13179.80	−13179.70
mam-eg	−10430.70	−10158.30	−10149.30**	−10459.70	−10422.30	−10312.00	−10422.60
mam-eh	−12721.30	−12135.30	−12105.30***	−12636.60	−12335.60	−12406.50	−12321.80
mam-ei	−10573.20	−10446.10	−10390.20***	−11051.10	−10653.60	−10776.40	−10641.90
mam-ej	−8737.66	−8517.75	−8481.18***	−8756.81	−8736.71	−8632.42	−8734.25
mam-ek	−17495.30	−16954.10	−16898.00***	−17882.70	−17323.70	−17529.20	−17321.60
mam-el	−18646.70	−18095.40	−18080.70**	−18938.40	−18612.90	−18503.10	−18581.10
mam-em	−12352.10	−12067.10	−11998.60***	−12477.90	−12297.10	−12210.40	−12264.30
mam-en	−5800.67	−5710.91*	−5711.16	−5963.36	−5840.37	−5849.16	−5831.59
mam-eo	−5453.77	−5272.88	−5263.57**	−5564.70	−5455.35	−5449.96	−5452.17
mam-ep	−18247.00	−17684.60	−17671.30**	−18141.40	−18201.90	−17946.00	−18200.40
mam-eq	−6479.70	−6279.44	−6257.77***	−6507.72	−6456.09	−6377.27	−6454.17
mam-er	−13134.10	−12664.40	−12632.70***	−13367.10	−12941.40	−13170.60	−12932.70
mam-es	−10934.30	−10567.10	−10549.80**	−10888.30	−10731.70	−10730.50	−10733.00
mam-et	−9568.77	−9313.54	−9295.16**	−9481.38	−9445.56	−9420.20	−9443.62
mam-eu	−8914.77	−8659.25	−8646.34**	−9167.90	−8815.42	−8921.16	−8812.55
mam-ev	−13556.10	−12989.90	−12984.80**	−13447.10	−13321.80	−13349.60	−13321.20
mam-ew	−8340.50	−7964.21	−7941.77***	−8400.73	−8117.86	−8244.54	−8113.08
mam-ex	−6842.05	−6579.01	−6577.13*	−6781.59	−6687.70	−6707.98	−6688.45
mam-ey	−7478.12	−7122.83	−7121.82*	−7322.98	−7209.36	−7312.63	−7207.53
mam-ez	−13823.10	−13365.50	−13276.90***	−14136.40	−13658.60	−13909.50	−13647.70
mam-fa	−7146.81	−6897.36	−6884.86**	−7190.49	−7035.35	−7097.59	−7032.02
mam-fb	−6479.78	−6236.29	−6230.93**	−6446.40	−6396.82	−6366.81	−6390.21
mam-fc	−7223.29	−7004.26	−6990.16**	−7190.42	−7176.75	−7109.98	−7176.97
mam-fd	−8094.87	−7839.99	−7837.48*	−8088.76	−7982.12	−8009.90	−7977.85
mam-fe	−6422.64	−6301.13	−6278.90***	−6410.14	−6385.68	−6398.42	−6380.12
mam-ff	−5910.81	−5784.41	−5776.10**	−5979.31	−5876.30	−5891.13	−5877.46
mam-fg	−8606.73	−8295.58	−8276.58**	−8753.80	−8530.97	−8558.48	−8530.17
mam-fh	−7470.35	−7353.35	−7342.16**	−7606.57	−7497.01	−7484.60	−7498.11

MSA	**M0**	**PCM+2C**	**PCM+8C**	**ECM+ν**	**ECM+ν (Mam)**	**ECM+ω+2κ**	**ECM+ω+2κ (Mam)**
mam-fi	−7133.92	−6914.81	−6914.73*	−7118.70	−7104.67	−7033.81	−7105.06
mam-fj	−7485.51	−7211.95	−7198.06**	−7460.29	−7388.61	−7376.98	−7385.15
mam-fk	−12823.20	−12299.60	−12275.60***	−12666.40	−12487.60	−12529.80	−12488.50
mam-fl	−6069.83	−5896.00	−5885.77**	−6131.06	−5980.23	−6058.70	−5981.33
mam-fm	−6941.10	−6853.10	−6837.57**	−7050.34	−6946.82	−6946.43	−6943.51
mam-fn	−7920.92	−7738.26	−7649.15***	−8191.26	−7926.36	−7885.59	−7907.74
mam-fo	−6672.20	−6390.85	−6390.03*	−6686.72	−6504.88	−6621.67	−6506.44
mam-fp	−12629.10	−12318.80	−12278.70***	−12632.70	−12535.60	−12459.90	−12536.10
mam-fq	−10368.50	−10036.20	−9973.74***	−10520.90	−10265.90	−10402.80	−10269.00
mam-fr	−6308.76	−6134.20	−6132.56*	−6379.08	−6332.67	−6294.85	−6333.38
mam-fs	−10816.90	−10579.00	−10561.30**	−11000.80	−10805.20	−10834.50	−10800.50
mam-ft	−11125.20	−10724.60*	−10724.80	−11083.00	−10955.00	−10990.00	−10952.90
mam-fu	−7135.63	−6969.13	−6968.24*	−7185.67	−7074.92	−7131.19	−7076.31
mam-fv	−10619.10	−10302.00	−10299.90*	−10615.10	−10480.00	−10547.10	−10480.60
mam-fw	−11700.40	−11381.70	−11334.30***	−11657.60	−11569.20	−11492.00	−11569.30
mam-fx	−8832.21	−8596.96	−8586.56**	−8988.72	−8789.72	−8867.45	−8789.63
mam-fy	−7098.35	−6875.61	−6866.34**	−7053.20	−6972.96	−7009.38	−6967.26
mam-fz	−11751.40	−11184.20	−11180.90**	−11500.80	−11385.10	−11469.30	−11372.50
mam-ga	−5667.70	−5478.71	−5467.42**	−5599.14	−5549.67	−5576.69	−5548.15
mam-gb	−8251.12	−8052.15	−8041.61**	−8359.13	−8222.95	−8277.98	−8222.07
mam-gc	−6394.24	−6307.87	−6269.92***	−6468.79	−6420.58	−6385.35	−6419.70
mam-gd	−8559.08	−8363.06	−8354.46**	−8891.16	−8551.65	−8722.55	−8547.94
mam-ge	−9827.61	−9507.99	−9412.44***	−10006.00	−9685.32	−9707.08	−9652.64
mam-gf	−11776.80	−11549.80	−11543.80**	−12219.10	−11778.60	−11942.90	−11773.00
mam-gg	−10081.90	−9750.81	−9748.07*	−10057.40	−10039.80	−9949.39	−10040.90
mam-gh	−8202.80	−7909.56	−7886.87***	−8266.19	−8108.54	−8161.39	−8107.17
mam-gi	−9915.39	−9618.55	−9614.99**	−9956.30	−9875.92	−9828.25	−9873.06
mam-gj	−7846.49	−7673.91	−7671.34*	−8027.69	−7885.52	−7931.89	−7884.37
mam-gk	−6286.25	−6159.18*	−6161.57	−6335.61	−6268.19	−6277.33	−6268.82
mam-gl	−8630.70	−8291.49	−8086.55****	−8783.33	−8480.89	−8518.75	−8467.28
mam-gm	−6562.65	−6374.29	−6352.12***	−6560.66	−6469.89	−6475.72	−6471.52
mam-gn	−7452.35	−7233.29	−7197.62***	−7588.20	−7471.27	−7469.60	−7470.81
mam-go	−7748.74	−7540.67	−7514.72***	−7960.09	−7686.74	−7824.66	−7686.72
mam-gp	−9517.09	−9220.98	−9214.40**	−9522.72	−9404.17	−9418.44	−9403.72
mam-gq	−12582.30	−12153.40	−12131.70***	−12760.20	−12398.80	−12628.70	−12398.60
mam-gr	−4261.01	−4195.12	−4184.93**	−4305.11	−4251.34	−4286.97	−4252.19
mam-gs	−5356.99	−5212.60*	−5215.00	−5460.79	−5361.89	−5368.77	−5360.61
mam-gt	−12052.00	−11685.80	−11676.40**	−12313.20	−11926.70	−12113.10	−11923.10
mam-gu	−7376.04	−7219.26	−7210.63**	−7537.95	−7429.10	−7426.34	−7430.37
mam-gv	−8378.78	−8035.67	−8032.68*	−8471.39	−8234.25	−8372.92	−8233.08
mam-gw	−7805.89	−7442.36	−7435.07**	−7792.64	−7669.50	−7746.79	−7667.82
mam-gx	−7764.79	−7360.60	−7315.52***	−7703.76	−7465.52	−7659.22	−7466.12
mam-gy	−8004.91	−7743.41	−7732.27**	−7946.15	−7861.47	−7866.48	−7862.87
mam-gz	−10769.50	−10467.50	−10466.60*	−10842.60	−10654.60	−10685.80	−10652.80
mam-ha	−8139.11	−7826.33	−7807.13**	−8121.61	−8079.18	−8030.01	−8075.71
mam-hb	−7901.17	−7641.47	−7621.61**	−8038.36	−7821.48	−7906.42	−7814.04
mam-hc	−5771.26	−5595.76	−5593.19*	−5755.17	−5719.45	−5687.00	−5720.40
mam-hd	−4528.51	−4385.76*	−4388.54	−4516.77	−4449.24	−4484.75	−4450.55
mam-he	−13408.70	−12810.10	−12758.20***	−13275.30	−13059.10	−13247.80	−13046.70
mam-hf	−6216.22	−6000.47	−5998.09*	−6290.38	−6143.04	−6201.38	−6142.90
mam-hg	−8170.72	−7898.97	−7895.20**	−8058.28	−8058.99	−8022.17	−8053.39

MSA	M0	PCM+2C	PCM+8C	ECM+ν	ECM+ν (Mam)	ECM+ω+2κ	ECM+ω+2κ (Mam)
mam-hh	−5732.48	−5718.77	−5706.15**	−5933.26	−5801.83	−5772.11	−5787.83
mam-hi	−8185.57	−7972.77	−7966.43**	−8285.98	−8106.41	−8161.49	−8106.68
mam-hj	−12456.70	−12050.90	−12021.00***	−12719.00	−12332.90	−12416.90	−12317.10
mam-hk	−4702.46	−4562.20**	−4566.21	−4705.16	−4652.19	−4673.30	−4652.66
mam-hl	−12101.60	−11665.50	−11661.70**	−12087.70	−11883.30	−11943.30	−11881.70
mam-hm	−5020.69	−4872.42	−4854.72**	−5070.95	−5015.41	−5048.20	−5008.22
mam-hn	−6579.63	−6327.35	−6326.29*	−6555.09	−6505.49	−6445.44	−6495.72
mam-ho	−5555.41	−5373.22	−5369.78**	−5547.50	−5509.28	−5497.92	−5511.14
mam-hp	−6441.14	−6270.03	−6266.09**	−6453.81	−6360.52	−6431.28	−6357.88
mam-hq	−9199.04	−8876.68	−8875.79*	−9251.89	−9098.76	−9095.40	−9092.72
mam-hr	−14306.40	−13752.00	−13738.70**	−14173.40	−13931.00	−14085.10	−13931.10
Total:	0	20(10.1%)	179(89.9%)	0	0	0	0
	0	4(2.0%)	149(74.9%)	0	0	0	0

Table S.8. – Estimated mlogL values for all *Archaea* MSAs.

MSA	M0	PCM+2C	PCM+8C	ECM+ν	ECM+ν (Mam)	ECM+ω+2κ	ECM+ω+2κ (Mam)
archaea-aa	−56226.50	−53755.20	−53510.80	−52712.90	−54237.30	−52700.70**	−54076.30
archaea-ab	−51046.40	−48638.80	−48500.50	−47909.30	−49153.50	−47907.80*	−48989.00
archaea-ac	−40743.50	−38532.00	−38461.60	−37870.80	−38895.10	−37870.10*	−38799.80
archaea-ad	−30432.30	−29117.50	−29043.20	−28646.60	−29398.20	−28645.00*	−29317.10
archaea-ae	−27965.30	−26847.00	−26752.50	−26354.40	−27029.70	−26345.30**	−26940.70
archaea-af	−32708.60	−31266.50	−31162.30	−30730.50**	−31659.30	−30734.00	−31507.90
archaea-ag	−24788.40	−23420.50	−23309.90	−22888.40	−23690.90	−22885.50*	−23549.50
archaea-ah	−29268.40	−27788.20	−27728.30	−27486.30*	−28315.50	−27488.00	−28195.50
archaea-ai	−26053.20	−24875.10	−24720.00	−24348.70	−25130.50	−24344.40**	−25059.00
archaea-aj	−26097.40	−24876.20	−24783.70	−24382.30**	−25210.50	−24385.90	−25149.40
archaea-ak	−59913.10	−56718.80	−56641.10	−55875.00	−57448.20	−55874.50*	−57281.90
archaea-al	−55878.70	−53158.00	−53025.10	−52323.50	−53846.90	−52319.90**	−53617.90
archaea-am	−45959.80	−43785.30	−43510.70	−42898.90**	−44334.60	−42903.20	−44084.80
archaea-an	−32526.30	−31256.60	−31065.40	−30662.00**	−31493.90	−30665.50	−31400.10
archaea-ao	−39316.50	−37712.80	−37564.40	−37082.60	−38240.10	−37082.10*	−38107.50
Total:	0	0	0	5(33.3%)	0	10(66.7%)	0
	0	0	0	4(26.7%)	0	4(26.7%)	0

Table S.9. – Estimated mlogL values for all *Cyanobacteria* MSAs.

MSA	M0	PCM+2C	PCM+8C	ECM+ν	ECM+ν (Mam)	ECM+ω+2κ	ECM+ω+2κ (Mam)
cyano-aa	−59827.60	−57406.20	−57047.10	−56532.90**	−58205.70	−56543.70	−57923.50
cyano-ab	−51499.70	−49479.80	−49195.20	−48646.00**	−50001.30	−48653.50	−49784.00
cyano-ac	−46942.10	−45093.00	−44792.50	−44234.60**	−45495.60	−44239.90	−45303.60
cyano-ad	−32991.10	−31547.60	−31377.60	−31155.60**	−31995.80	−31168.60	−31809.50
cyano-ae	−49264.50	−47104.80	−46888.30	−46612.30**	−47746.10	−46620.00	−47484.10
cyano-af	−56236.20	−54056.60	−53802.30	−53549.50**	−54986.40	−53556.50	−54559.10

MSA	M0	PCM+2C	PCM+8C	ECM+ν	ECM+ν (Mam)	ECM+ω+2κ	ECM+ω+2κ (Mam)
cyano-ag	−29007.10	−28102.40	−27998.90	−27588.30	−28388.30	−27587.80*	−28311.50
cyano-ah	−42809.50	−41073.10	−40694.00	−40317.20*	−41455.30	−40318.30	−41281.10
cyano-ai	−39390.30	−38159.80	−38036.40	−37751.20**	−38639.00	−37764.20	−38451.00
cyano-aj	−34419.80	−33229.20	−32977.20	−32577.60**	−33556.20	−32586.30	−33436.00
cyano-ak	−38092.80	−36475.10	−36285.90	−35899.40**	−37010.80	−35908.50	−36846.80
cyano-al	−34770.50	−33424.80	−33190.80	−32960.70**	−33734.30	−32967.20	−33626.90
cyano-am	−42250.80	−40500.60	−40297.00	−39822.90**	−41058.40	−39828.20	−40748.50
cyano-an	−33013.70	−31608.50	−31525.40	−31164.60**	−32011.90	−31174.20	−31864.30
cyano-ao	−41334.40	−39629.30	−39483.60	−39099.80**	−40226.80	−39108.90	−40014.40
Total:	0	0	0	14(93.3%)	0	1(6.7%)	0
	0	0	0	13(86.7%)	0	0(0.0%)	0

Table S.10. – Estimated mlogL values for all *Eukaryota* MSAs.

MSA	M0	PCM+2C	PCM+8C	ECM+ν	ECM+ν (Mam)	ECM+ω+2κ	ECM+ω+2κ (Mam)
other-aa	−27340.00	−25905.70	−25774.80	−25530.00*	−26250.30	−25530.50	−26085.40
other-ab	−22012.10	−20915.60	−20667.30	−20470.20**	−20953.50	−20475.30	−20841.50
other-ad	−18273.60	−17267.40	−17129.90	−17042.80	−17438.80	−17034.70**	−17366.40
other-ae	−17893.00	−17096.50	−16849.90	−16628.20*	−17140.70	−16629.80	−17054.30
other-af	−14312.50	−13624.40	−13509.50	−13297.90**	−13487.80	−13303.20	−13444.10
other-ag	−55606.20	−53071.90	−52652.10	−52353.00*	−53530.40	−52355.40	−53250.70
other-ah	−36070.10	−34324.70	−34118.20	−33810.20	−34576.00	−33808.90*	−34372.70
other-ai	−47727.50	−45643.80	−45273.20	−44796.40**	−45796.10	−44801.10	−45581.10
other-aj	−35869.90	−34374.60	−34250.00	−34116.10	−34777.50	−34113.50*	−34680.40
other-ak	−30456.10	−28840.30	−28561.10	−28252.70**	−28987.60	−28259.80	−28844.40
other-al	−21330.60	−20234.90	*N/A*	−19840.60	−20315.70	−19838.80*	−20241.10
other-am	−18224.70	−17295.10	−17180.20	−17036.90*	−17433.50	−17037.00	−17343.90
other-an	−23560.80	−22464.20	−22122.90	−21893.30	−22473.30	−21891.70*	−22357.00
other-ao	−22243.10	−21348.00	−21069.10	−20852.80	−21388.20	−20852.40*	−21249.00
Total:	0	0	0	8(57.1%)	0	6(42.9%)	0
	0	0	0	4(28.6%)	0	1(7.1%)	0

Table S.11. – Estimated mlogL values for all *Vertebrata* MSAs.

MSA	M0	PCM+2C	PCM+8C	ECM+ν	ECM+ν (Mam)	ECM+ω+2κ	ECM+ω+2κ (Mam)
vert-aa	−11972.90	−11732.10	−11696.30***	−11823.80	−11852.20	−11779.60	−11850.40
vert-ab	−10068.90	−9920.45	−9855.67***	−9971.65	−9997.02	−9913.33	−9996.36
vert-ac	−8326.55	−8017.30	−8002.69**	−8104.91	−8104.46	−8070.75	−8105.77
vert-ae	−7187.92	−7035.18*	−7035.64	−7076.54	−7123.42	−7074.84	−7109.17
vert-ag	−11358.30	−11111.00	−11098.30**	−11176.90	−11273.40	−11163.70	−11263.20
vert-ah	−8446.11	−8273.06	−8252.70***	−8354.56	−8350.53	−8327.26	−8347.87
vert-ai	−10801.50	−10582.60	−10553.50***	−10633.20	−10663.00	−10622.50	−10650.30
vert-aj	−8819.93	−8549.16	−8500.02***	−8704.45	−8588.58	−8687.66	−8588.07
vert-ak	−6460.28	−6340.32	−6330.73**	−6390.21	−6427.72	−6366.91	−6427.70

MSA	M0	PCM+2C	PCM+8C	ECM+ν	ECM+ν (Mam)	ECM+ω+2κ	ECM+ω+2κ (Mam)
Total:	0	1(11.1%)	8(88.9%)	0	0	0	0
	0	0(0.0%)	8(88.9%)	0	0	0	0

S.6. BEAST XML Example

Listing S.1 – Example configuration file for `BEAST`. The XML defines a PCM+F+2C model. The sequences have been cut after the second codon for better reading.

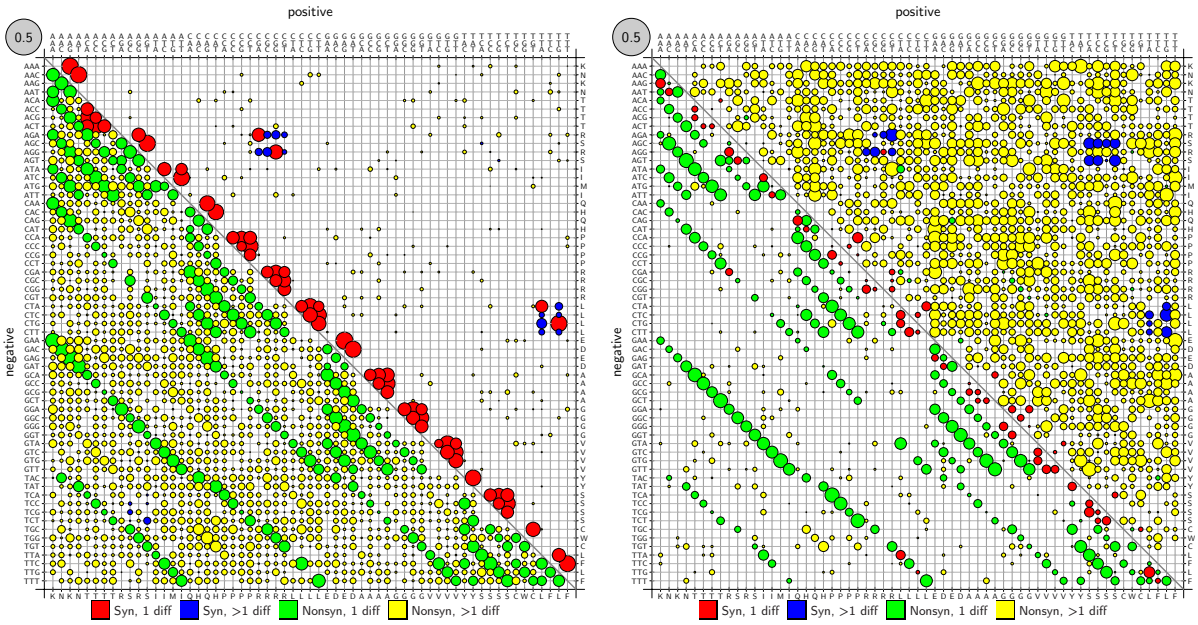
```
<?xml version="1.0" standalone="yes"?>
<beast>
  <!-- Define alignment -->
  <taxa id="taxa">
    <taxon id="IL9_HUMAN/1-142"/>
    <taxon id="IL7_SHEEP/28-172"/>
    <taxon id="IL7_BOVIN/28-172"/>
    <taxon id="IL7_HUMAN/28-173"/>
    <taxon id="IL7_MOUSE/28-152"/>
  </taxa>
  <alignment id="alignment" dataType="nucleotide">
    <sequence>
      <taxon idref="IL9_HUMAN/1-142"/>
      ATGCTT...
    </sequence>
    <sequence>
      <taxon idref="IL7_SHEEP/28-172"/>
      GATTTT...
    </sequence>
    <sequence>
      <taxon idref="IL7_BOVIN/28-172"/>
      GATATT...
    </sequence>
    <sequence>
      <taxon idref="IL7_HUMAN/28-173"/>
      GATATT...
    </sequence>
    <sequence>
      <taxon idref="IL7_MOUSE/28-152"/>
      CACATT...
    </sequence>
  </alignment>
  <newick id="startingTree">
(IL7_SHEEP/28-172:0.01322,(IL7_BOVIN/28-172:0.01494,IL9_HUMAN/1-142:2.29798):0.00000,(
      IL7_MOUSE/28-152:0.14822,IL7_HUMAN/28-173:0.10619):0.08037);
  </newick>

  <convert id="codonAlignment" dataType="codon" geneticCode="universal">
    <alignment idref="alignment"/>
  </convert>
  <patterns id="patterns" from="1">
```

```
41      <alignment idref="codonAlignment"/>
42    </patterns>
43
44    <!-- Define tree and branch rate models -->
45
46    <treeModel id="treeModel">
47      <newick idref="startingTree"/>
48      <rootHeight>
49        <parameter id="treeModel.rootHeight"/>
50      </rootHeight>
51      <nodeHeights internalNodes="true">
52        <parameter id="treeModel.internalNodeHeights"/>
53      </nodeHeights>
54      <nodeHeights internalNodes="true" rootNode="true">
55        <parameter id="treeModel.allInternalNodeHeights"/>
56      </nodeHeights>
57    </treeModel>
58
59    <arbitraryBranchRates id="branchRates">
60      <treeModel idref="treeModel"/>
61      <rates>
62        <parameter id="branch.rates"/>
63      </rates>
64    </arbitraryBranchRates>
65
66    <rateStatistic id="meanRate" name="meanRate" mode="mean" internal="true" external="true">
67      <treeModel idref="treeModel"/>
68      <arbitraryBranchRates idref="branchRates"/>
69    </rateStatistic>
70
71    <!-- Define substitution model -->
72
73    <pcaCodonModel id="pca" geneticCode="universal" pcaType="mammalia" pcaDataDir="pcadata"
          >
74      <pcaDimension>
75        <parameter id="pcaDimension" dimension="2"/>
76      </pcaDimension>
77      <frequencyModel id="freqmodel" dataType="codon-universal">
78        <alignment idref="codonAlignment"/>
79        <frequencies>
80          <parameter id="codonfrequencies" dimension="61"/>
81        </frequencies>
82      </frequencyModel>
83    </pcaCodonModel>
84
85    <siteModel id="siteModel">
86      <substitutionModel>
87        <pcaCodonModel idref="pca"/>
88      </substitutionModel>
89    </siteModel>
90
91    <treeLikelihood id="treeLikelihood" useAmbiguities="false">
92      <patterns idref="patterns"/>
```

```
93      <treeModel idref="treeModel"/>
94      <siteModel idref="siteModel"/>
95      <arbitraryBranchRates idref="branchRates"/>
96    </treeLikelihood>
97
98    <!-- Define operators -->
99    <operators id="operators">
100     <randomWalkOperator windowSize="1.0" weight="1.0" autoOptimize="true">
101       <parameter idref="pcaDimension"/>
102     </randomWalkOperator>
103
104     <scaleOperator scaleFactor="0.75" weight="15.0">
105       <parameter idref="branch.rates"/>
106     </scaleOperator>
107   </operators>
108
109   <!-- Define MCMC -->
110   <mcmc id="mcmc" chainLength="734070" autoOptimize="true">
111     <posterior id="posterior">
112       <prior id="prior">
113         <uniformPrior lower="0.0" upper="100.0">
114           <parameter idref="branch.rates"/>
115         </uniformPrior>
116       </prior>
117       <likelihood id="likelihood">
118         <treeLikelihood idref="treeLikelihood"/>
119       </likelihood>
120     </posterior>
121     <operators idref="operators"/>
122
123     <!-- write log to screen -->
124     <log id="screenLog" logEvery="50000">
125       <column label="Posterior" dp="4" width="12">
126         <posterior idref="posterior"/>
127       </column>
128       <column label="Likelihood" dp="4" width="12">
129         <likelihood idref="likelihood"/>
130       </column>
131       <column label="meanRate" sf="6" width="12">
132         <rateStatistic idref="meanRate"/>
133       </column><column label="pcaDimension" sf="6" width="12">
134       <parameter idref="pcaDimension"/>
135     </column></log>
136     <!-- write log to file -->
137     <log id="fileLog" logEvery="500" fileName="pca2.log">
138       <posterior idref="posterior"/>
139       <likelihood idref="likelihood"/><parameter idref="pcaDimension"/>
140       <rateStatistic idref="meanRate"/>
141       <parameter idref="branch.rates"/>
142       <treeLikelihood idref="treeLikelihood"/>
143     </log>
144   </mcmc>
145
```

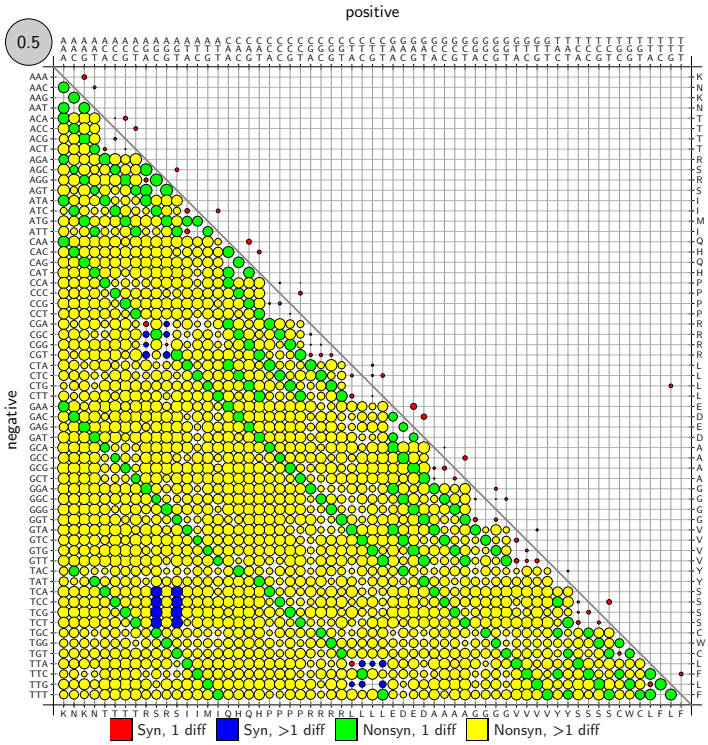
```
146   <report>
147     <property name="timer">
148       <mcmc idref="mcmc"/>
149     </property>
150   </report>
151
152 </beast>
```

S.7. Additional Bubble Plots of Principal Components

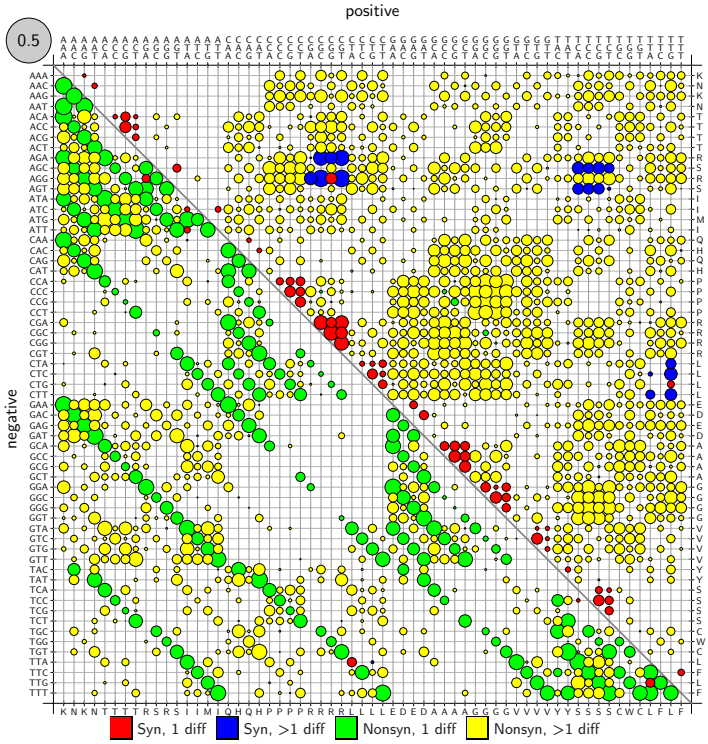


(a) PC1 of clustered *Mammalia* data **(b)** PC2 of clustered *Mammalia* data

Figure S.1. – PC1 and PC2 of clustered *Mammalia* data set. The color coding and the arrangement of positive and negative values corresponds to figures 3.6 and 3.7.

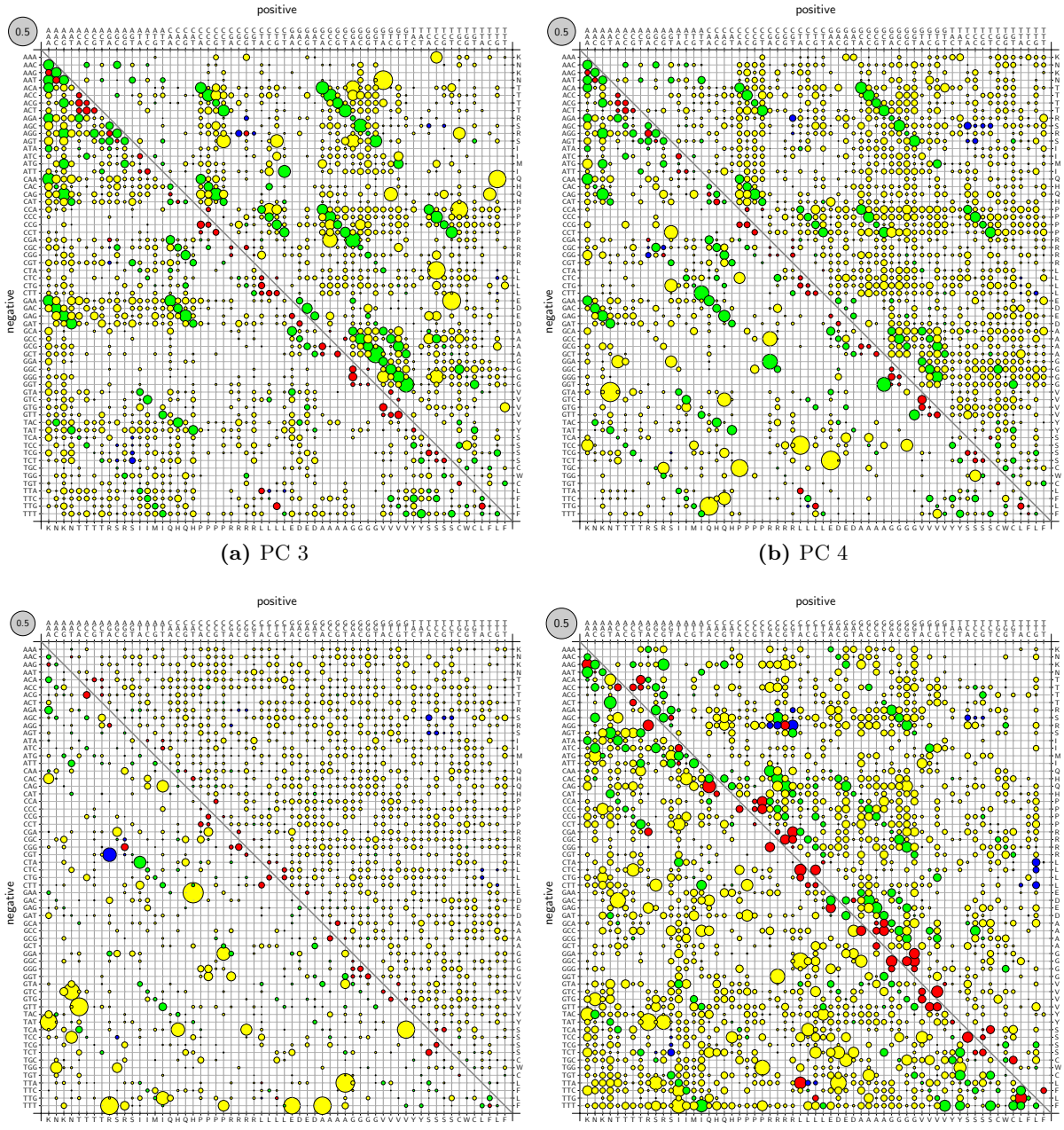


(a) PC1 of log-transformed *Mammlia* data



(b) PC2 of log-transformed *Mammalia* data

Figure S.2. – PC1 and PC2 of log-transformed *Mammalia* data set. The color coding and the arrangement of positive and negative values corresponds to figures 3.6 and 3.7.



(a) PC 3 **(b)** PC 4

(c) PC 5 **(d)** PC 6

Figure S.3. – Principal components 3 to 6 of *Mammalia* data set. The color coding and the arrangement of positive and negative values corresponds to figures 3.6 and 3.7. We were not able to determine a simple description of any of these components.