Application of mechanistic models to separate the effects of mutation, selection, and drift on protein sequence evolution

A Thesis Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Cedric Landerer

December 2018

To my mother

Acknowledgments

First and foremost I want to thank my Adviser, Michael Gilchrist for his long lasting patience, his availability and his teachings; always sharpening my focus and providing a new angle to a problem.

The faculty and students in EEB allowing me to broaden knowledge and insights with always stimulating discussions.

Nothing in Biology Makes Sense Except in the Light of Evolution.

-Theodosius Dobzhansky

Abstract

Mutation, selection, and drift are the three forces guiding evolution.

Interplay of forces is complicated.

Efficacy of one depends on the others, e.g. population mutation rate is μN_e, efficacy of selection is sN_e.

Using mechanistic models to separate effects.

Separation of mutation, and selection allows us to extract more information.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Mathematical and statistical have long been used to describe or summarize observations in genetics and genomics. Mostly without addressing the underlying biological mechanisms mutation, selection, and drift shaping DNA sequences, but as phenomelogical description. However, as researchers learn more about the underlying processes and more genetic and genomic data is available, the mathematical descriptions that allow us to extract information from this data have to keep up. For example, after the unreaveling of the degenerate genetic code by Matthaei and Nierenberg [58], Nierenberg and Matthaei [63], Maxwell [59], Leder and Nierenberg [55], and many others, researchers noticed that synonymous codons are not found in uniform proportions [24, 36, 42, 37, 74]. Models of codon usage, however, where long purley describtive and heuristic [42, 6, 73, 88]. Similarly, phylogenetic models have long been phenomelogical [46, 13, 48, 20, 1], describing the rate at which one state is transformed into another, without regards for the fundamental forces of evolution mutation, selection, and drift. Zuckerkandl and Pauling [92] described the distance between hemoglobin proteins and proposed that the evolution of proteins is constant over time and between lineages before the genetic code was fully deciphered and were protein production was barely understood. This dissertation is therefore focues on the application of mechanistic models rooted in first principles to protein coding sequences.

A wide variety of information is stored in protein and protein coding sequences, e.g. structure [2], mutation bias [71, 29], protein synthesis rate [28, 29]. Mechanistic models can be used extract these informations and to study the relative strength of mutation, selection, and genetic drift leading to the observed sequences. Mechanistic models have a long standing history throughout biology [53, 12, 15]. They provide insights into the processes and estimates of paramerers shaping the data we observe. In the case of this dissertation, mechanistic models lead to an understanding of the contributions of mutation, selection and drift on the evolution of observed sequences. This allows us to use this information to learn more about these seqeunces.

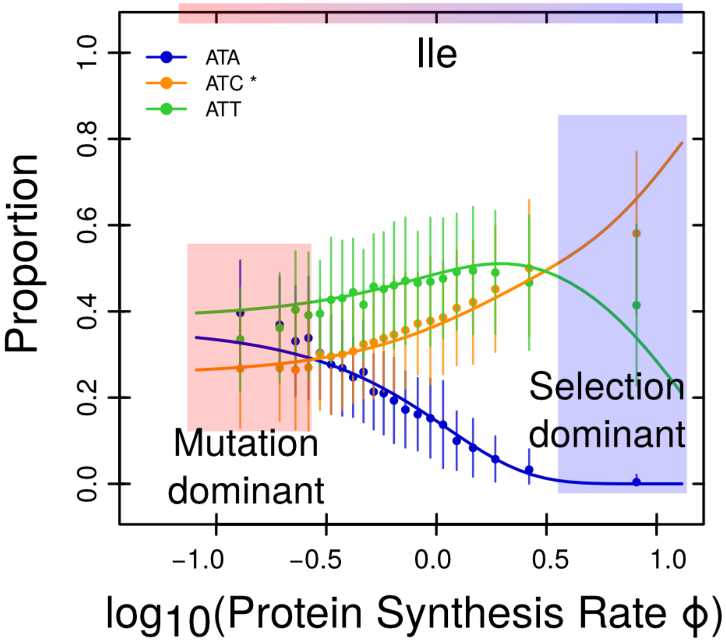


Figure 1.1:

1.1 Decomposing Codon Usage

Mutation bias in codon usage is a reflection of the cellular environment while selection on codon usage allows us to make inferences about the cellular and external evironment a genome and its genes are exposed to. The relative strength of mutation and selection on individual genes varies, allowing us to separate the effects of mutation bias and selection, specifically selection against translation overhead cost. Genes with low protein synthesis rates are thought to be under weak selection and their codon usage is therfore dominated by mutation bias. In contrast, genes with high protein synthesis rate are thought to be under strong selection and their codon usage is therfore dominated by selection. However, mutation bias and selection can differ within the genome. For example, strand specific mutation rates, differences in the tRNA pool thorughout life stages or introgressions can produce or reflect of mutliple genomic environments. To provide researchers with a software tool, AnaCoDa [50], to address intra genomic variation in codon usage, chaper 2 extends the mechanistic model ROC-SEMPPR [29] to allow for a mixture distribution of mutation and selection parameters. However, there is a significant difference to classical mixutre approaches as ROC-SEMPPR not only estimates population specific paramerers (mutation and selection) that are now modelled as mixture distibutions but also a gene specific parameter (protein synthesis rate). Therefore, the protein synthesis rate has to be estimated for each population, providing aditional insight into the adaptiveness of a gene to alternative genomic environments.

In chaper 3, I apply AnaCoDa to the yeast *L. kluyveri* which experienced a large scale introgression replacing the whole left arm of chromosome C. Applying a mechanistic models allowed me to separate the effects of mutation and selection in the exogenous *L. kluyveri* genes and the introgressed exogenous genes. This information was used to determine a potential donor lineage in *E. gossypii*, estimate a time since introgression, and estimate the genetic load the introgression introduced into the *L. kluyveri* genome.

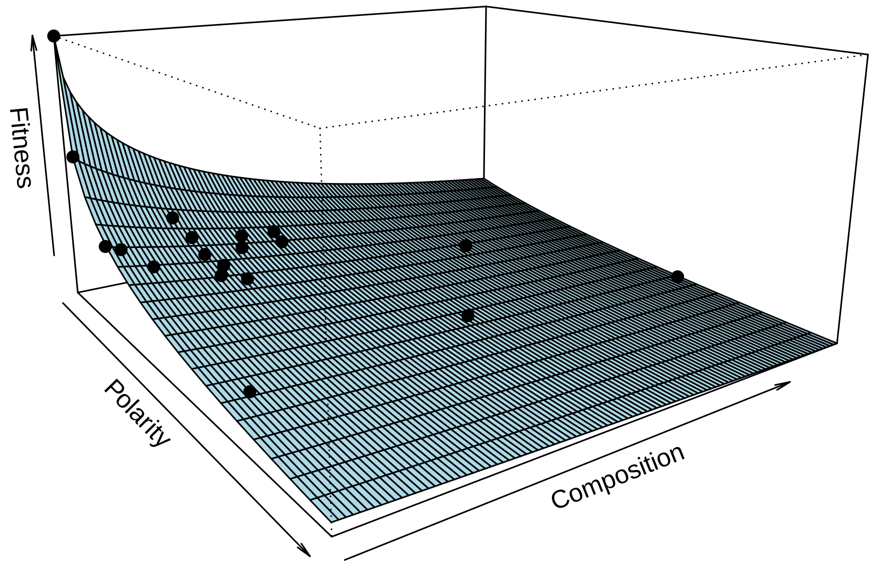


Figure 1.2: Decline in fitness with distance in physicochemical space from the optimal amino acid. Fitness decline of amino acids (black dots) relative to optimal amino acid (Alanine). Weighting of properties was obtained from Grantham [35].

I chapter 4, I shifted my focus from the cost of protein synthesis to the functionallity or benefit a protein produces. For that matter I utilized a mechanisitc phylogenetic model of stabilizing selection rooted in population genetics, SelAC [5].

Chapter 2

AnaCoDa: Analyzing Codon Data with Bayesian mixture models

This chapter is a lightly revised version of a paper by the same name published in Bioinformatics and co-authored with Alexander Cope, Russell Zaretzki, and Michael A. Gilchrist.

C. Landerer, A. Cope, R. Zaretzki, M.A. Gilchrist, AnaCoDa: analyzing codon data with Bayesian mixture models, Bioinformatics, 34, 2018, 2496-2498

Abstract

AnaCoDa is an R package for estimating biologically relevant parameters of mixture models, such as selection against translation inefficiency, nonsense error rate, and ribosome pausing time, from genomic and high throughput datasets. **AnaCoDa** provides an adaptive Bayesian MCMC algorithm, fully implemented in C++ for high performance with an ergonomic R interface to improve usability. **AnaCoDa** employs a generic object-oriented design to allow users to extend the framework and implement their own models. Current models implemented in **AnaCoDa** can accurately estimate biologically relevant parameters given either protein coding sequences or ribosome foot-printing data. Optionally, **AnaCoDa** can utilize additional data sources, such as gene expression measurements, to aid model fitting and parameter estimation. By utilizing a hierarchical object structure, some parameters can vary between sets of genes while others can be shared. Genes may be assigned to clusters or membership may be estimated by **AnaCoDa**. This flexibility allows users to estimate the same model parameter under different biological conditions and categorize genes into different sets based on shared model properties embedded within the data. **AnaCoDa** also allows users to generate simulated data which can be used to aid model development and model analysis as well as evaluate model adequacy. Finally, **AnaCoDa** contains a set of visualization routines and the ability to revisit or re-initiate previous model fitting, providing researchers with a well rounded easy to use framework to analyze genome scale data.

Availability:

AnaCoDa is freely available under the Mozilla Public License 2.0 on CRAN (https://cran.r-project.org/web/packages/AnaCoDa/).

2.1 Introduction

AnaCoDa is an open-source software implemented in R [66] that allows researchers to analyze genome-scale data like coding sequences and ribosome footprinting data using evolutionary or analytical models in a Bayesian framework. **AnaCoDa** was developed to analyze selection on synonymous codon usage in the form of ribosome overhead cost [29, 86, 72]. However, other codon metrics like the codon adaptation index [73] or the effective number of codons [**?**] are also provided as reference. In addition, three currently unpublished models to analyze coding sequences for evidence of selection against nonsense errors and estimate ribosome pausing times from ribosome footprinting data are included. **AnaCoDa** implements an adaptive Gibbs sampler within a Metropolis-Hastings Monte Carlo Markov Chain (MCMC). This allows for the incorporation of prior knowledge such as observed gene expression levels and easy sampling from the posterior distribution to estimate parameter values and quantify degree of uncertainty. **AnaCoDa** provides a mixture distribution option to all implemented models, combining genes into sets by estimating the posterior probabilities of set membership based on gene-set specific parameters shared by all genes assigned to a given set. **AnaCoDa** provides a generic, mixture distribution option to all implemented models, allowing for the estimation of condition specific parameters or the automatic categorization of data into different sets based on differences in their posterior probabilities of set membership. In addition to the four models provided, **AnaCoDa** provides a modular infrastructure such that additional genome scale or even phylogenetic models can be integrated.

The **AnaCoDa** framework works with **AnaCoDa** requires gene specific data such as codon frequencies obtained from coding sequences or position specific footprint counts. Conceptually, **AnaCoDa** allows for three different types of parameters. The first type are gene specific parameters such as protein synthesis rate or relative functionality. The second type are gene-set specific parameters, such as mutation bias terms or translation error rates. These parameters are shared across genes within a set and can be exclusive to a single set or shared with other sets. While the number of gene sets must be pre-defined by the user, set assignment of genes can be pre-defined or estimated as part of the model fitting. Estimation of the set assignment provides the probability of a gene being assigned to a set allowing the user to asses the uncertainty in each assignment. The third type are hyperparameters allowing for the construction and analysis of hierarchical model. Hyperparameters control the prior distribution for gene and gene-set specific parameters such as mutation bias or protein synthesis rate.

2.2 Features

AnaCoDa provides an interface written in R, a freely available programming language noted for its ease of use for even inexperienced programmers. As a result, **AnaCoDa** is accessible to researchers with minimal computational experience.

The interface of **AnaCoDa** is designed for quick and efficient data analysis. Generally, the only input needed for fitting a model to the data are protein-coding codon sequences in the form of a FASTA file or a flat-file containing codon counts obtained from ribosome foot-printing experiments. **AnaCoDa** also provides visualization functionality, including plotting functions to compare parameter estimates for different mixture distributions and display codon usage patterns. In addition, diagnostic functions such as those for calculating and visualizing the degree of autocorrelation in the parameter tracess are provided.

Robust and efficient model fitting

AnaCoDa has built-in features designed to improve the robustness and performance of the implemented MCMC approach. For example, the implemented MCMC automatically adapts the proposal width for sampled parameters such that an user defined acceptance range is met, improving sampling efficiency of the MCMC and computational performance. Even though **AnaCoDa** is written in C++, analysis of large datasets and/or complex models can be very computationally intensive. To protect users from computer failures or aid in the collection of additional MCMC samples, **AnaCoDa** can periodically produce output checkpoint files, which can be used to restart an MCMC chain from a previous time point. In addition, **AnaCoDa** automatically thins all parameter traces - meaning only every k^{th} sample is kept - increasing the effective number of samples and reducing its memory foorprint.

Although **AnaCoDa** is provided as an R package, the main computational work is implemented in C++. Because R does not provide native C++ support, Rcpp was employed to expose whole C++ classes as modules to R [19]. Using Rcpp eliminates time consuming data transfers between the R environment and the C++ core during model fitting, resulting in improved computational performance and allows for a fully object-oriented code design [9]. As expected, the runtime of **AnaCoDa** scales linearly with genome size and number of iterations, and scales polynomially with the number of mixture distributions in the data set. The polynomial increase in runtime with the number of mixture distributions is due to the necessity to condition the gene assignment on the estimation of gene specific parameters, such as, protein synthesis rate.

Data Simulation

In addition to fitting the models to datasets, **AnaCoDa** can be used to generate simulated data sets as well. On their own, simulated datasets are useful for model development and analysis. Simulating data under different conditions allows the user to explore model behavior and explore theoretical scenarios. Different conditions can include the addition or elimination of parameters, or simply allowing a set of parameter values to vary. Fitting models to simulated data can provide insight into potential pitfalls or shortcomings when fitting observational data and can serve as the basis for evaluating model adequacy of a model fit to observational data [60]. Significant differences between simulated and observational data suggests the current set of parameters or the model as a whole fail to include or adequately represent biological mechanisms underlying the observed data.

Available models

AnaCoDa currently provides codon models for analyzing genome scale data. The ROC model implements and extends the codon usage bias (CUB) models developed by Gilchrist et al. [29], Wallace et al. [86], Shah and Gilchrist [72], which can reliably estimate the strength of selection on ribosome overhead cost, mutation bias and allows for the inference of protein synthesis rates. This model allows for the separation of effects of mutation and selection based on gene ordering by protein synthesis rate, and the addition of a mixture distribution allows for gene clustering based on mutation bias and selection for translation efficiency. In addition to identifying the most efficient codons, ROC provides estimates of mutation bias allowing the approximation of mutation ratios between codons [29, 86].

The ability to estimate protein synthesis rates in the absence of empirical data is useful for investigating CUB of non-model organisms for which such data is lacking and enables the usage of protein synthesis rate in comparative frameworks or other analyses requiring protein synthesis rate information [17]. Use of the mixture model allows for the investigation of CUB heterogeneity at the genome or gene level. Following the same framework, additional models included in **AnaCoDa** provide estimates of codon-specific nonsense errors rates (FONSE) and ribosome pausing times (PA and PANSE).

Parameters estimated with the evolutionary models ROC and FONSE represent evolutionary averages and do not depend on experimental conditions. In contrast, PA and PANSE estimate the distribution of biologically relevant parameters like ribosome pausing times along a gene from experimental data such as ribosome footprinting data. The distribution can be dependent (PANSE) or independent (PA) of evidence for nonsense errors in the data.

2.3 Supplementary Material

AnaCoDa allows for the estimation of biologically relevant parameters like mutation bias or ribosome pausing time, depending on the model employed. Bayesian estimation of parameters is performed using an adaptive Metropolis-Hasting within Gibbs sampling approach. Models implemented in AnaCoDa are currently able to handle gene coding sequences and ribosome footprinting data.

2.3.1 The AnaCoDa framework

The AnaCoDa framework works with gene specific data such as codon frequencies or position specific footprint counts. Conceptually, AnaCoDa uses three different types of parameters.

- The first type of parameters are **gene specific parameters** such as gene expression level or functionality. Gene-specific parameters are estimated separately for each gene and can vary between potential gene categories or sets.
- The second type of parameters are **gene-set specific parameters**, such as mutation bias terms or translation error rates. These parameters are shared across genes within a set and can be exclusive to a single set or shared with other sets. While the number of gene sets must be pre-defined by the user, set assignment of genes can be pre-defined or estimated as part of the model fitting. Estimation of the set assignment provides the probability of a gene being assigned to a set allowing the user to asses the uncertainty in each assignment.
- The third type of parameters are **hyperparameters**, such as parameters controlling the prior distribution for mutation bias or error rate. Hyperparameters can be set specific or shared across multiple sets and allow for the construction and analysis of hierarchical models, by controlling prior distributions for gene or gene-set specific parameters.

Analyzing protein coding gene sequences

AnaCoDa always requires the following four objects:

- **Genome** contains the codon data read from a fasta file as well as empirical protein synthesis rate in the form of a comma separated (.csv) ID/Value pairs.

- **Parameter** represents the parameter set (including parameter traces) for a given genome. The parameter object also hold the mapping of parameters to specified sets.

- **Model** allows you to specify which model should be applied to the genome and the parameter object.

- **MCMC** specifies how many samples from the posterior distribution of the specified model should be stored to obtain parameter estimates.

2.3.2 AnaCoDa setup

Application of codon model to single genome

In this example we are assuming a genome with only one set of gene-set specific parameters, hence **num.mixtures** $= 1$. We assign all genes the same gene-set, and provide an initial value for the hyperparameter sphi (s_ϕ). s_ϕ controls the lognormal prior distribution on the gene specific parameters like the protein synthesis rate ϕ. To ensure identifiability the expected value of the prior distribution is assumed to be 1.

$$E[\phi] = \exp\left(m_\phi + \frac{s_\phi^2}{2}\right) = 1 \tag{2.1}$$

Therfore the mean m_ϕ is set to be $-\frac{s_\phi^2}{2}$. For more details see Gilchrist et al. [29].

After choosing the model and specifying the necessary arguments for the MCMC routine, the MCMC is run

```
genome <- initializeGenomeObject(file = "genome.fasta")
parameter <- initializeParameterObject(genome = genome, sphi = 1,
                        num.mixtures = 1,
                        gene.assignment = rep(1, length(genome)))
```

```
model <- initializeModelObject(parameter = parameter, model = "ROC")
mcmc <- initializeMCMCObject(samples = 5000, thinning = 10,
                        adaptive.width=50)
runMCMC(mcmc = mcmc, genome = genome, model = model)
```

runMCMC does not return a value, the results of the MCMC are stored automatically in the mcmc and parameter objects created earlier.

> **Please note that AnaCoDa utilizes C++ obejct orientation and therefore employs pointer structures. This means that no return value is necessary for such objects as they are modified within the the runMCMC routine. You will find that after a completed run, the parameter object will contain all necessary information without being directly passed into the MCMC routine. This might be confusing at first as it is not default R behaviour.**

Application of codon model to a mixture of genomes

This case applies if we assume that parts of the genome differ in their gene-set specific parameters. This could be due to introgression events or strand specific mutation difference, horizontal gene transfers or other reasons. We make the assumption that all sets of genes are independent of one another. For two sets of gene-set specific parameter with a random gene assignment we can use:

```
parameter <- initializeParameterObject(genome = genome,
                        sphi = c(0.5, 2), num.mixtures = 2,
                        gene.assignment = sample.int(2,
                                length(genome), replace = T))
gene.assignment = sample.int(2, length(genome), replace = T))
```

To accommodate for this mixing we only have to adjust **sphi**, which is now a vector of length 2, **num.mixtures**, and **gene.assignment**, which is chosen at random here.

Empirical protein synthesis rate values

To use empirical values as prior information one can simply specify an observed.expression.file when initializing the genome object.

```
genome <- initializeGenomeObject(file = "genome.fasta",
                    observed.expression.file = "synthesis_values.csv")
```

These observed expression or synthesis values (Φ) are independent of the number of gene-sets. The error in the observed Φ values is estimated and described by sepsilon (s_ϵ). The csv file can contain multiple observation sets separated by comma. For each set of observations an initial s_ϵ has to be specified.

```
# One case of observed data
sepsilon <- 0.1
# Two cases of observed data
sepsilon <- c(0.1, 0.5)
# ...
# Five cases of observed data
sepsilon <- c(0.1, 0.5, 1, 0.8, 3)
parameter <- initializeParameterObject(genome = genome, sphi = 1,
                        num.mixtures = 1,
                        gene.assignment = rep(1, length(genome)),
                        init.sepsilon = sepsilon)
```

In addition one can choose to keep the noise in the observations (s_ϵ) constant by using the **fix.observation.noise** flag in the model object.

```
model <- initializeModelObject(parameter = parameter, model = "ROC",
fix.observation.noise = TRUE)
```

Fixing parameter types

It can sometime be advantages to fix certain parameters, like the gene specific parameters. For example in cases where only few sequences are available but gene expression measurements are at hand we can fix the gene specific parameters to increase confidence in our estimates of gene-set specific parameters.

We again initialize the **genome**, **parameter**, and **model** objects.

```
genome <- initializeGenomeObject(file = "genome.fasta")
parameter <- initializeParameterObject(genome = genome, sphi = 1,
                        num.mixtures = 1,
                        gene.assignment = rep(1, length(genome)))
model <- initializeModelObject(parameter = parameter, model = "ROC")
```

To fix gene specific parameters we will set the **est.expression** flag to **FALSE**. This will estimate only gene-set specific parameters, hyperparameters, and the assignments of genes to various sets.

```
mcmc <- initializeMCMCObject(samples, thinning=1,
                        adaptive.width=100, est.expression=FALSE,
                        est.csp=TRUE, est.hyper=TRUE, est.mix=TRUE)
```

If we would like to fix gene-set specific parameters we instead disable the **est.csp** flag.

```
mcmc <- initializeMCMCObject(samples, thinning=1,
                        adaptive.width=100, est.expression=TRUE,
                        est.csp=FALSE, est.hyper=TRUE, est.mix=TRUE)
```

The same applies to the hyper parameters (**est.hyper**),

```
mcmc <- initializeMCMCObject(samples, thinning=1,
                        adaptive.width=100, est.expression=TRUE,
                        est.csp=TRUE, est.hyper=FALSE, est.mix=TRUE)
```

and gene set assignment (**est.mix**).

```
mcmc <- initializeMCMCObject(samples, thinning=1,
                        adaptive.width=100, est.expression=TRUE,
                        est.csp=TRUE, est.hyper=TRUE, est.mix=FALSE)
```

We can use these flags to fix parameters in any combination.

Combining various gene-set specific parameters to a gene-set description.

We distinguish between three simple cases of gene-set descriptions, and the ability to customize the parameter mapping. The specification is done when initializing the parameter object with the **mixture.definition** argument.

We encounter the simplest case when we assume that all gene sets are independent.

```
parameter <- initializeParameterObject(genome = genome,
                        sphi = c(0.5, 2), num.mixtures = 2,
                        gene.assignment = sample.int(2,
                                length(genome), replace = T),
                        mixture.definition = "allUnique")
```

The **allUnique** keyword allows each type of gene-set specific parameter to be estimated independent of parameters describing other gene sets.

In case we want to share mutation parameter between gene sets we can use the keyword **mutationShared**

```
parameter <- initializeParameterObject(genome = genome,
                        sphi = c(0.5, 2), num.mixtures = 2,
                        gene.assignment = sample.int(2,
                                length(genome), replace = T),
                        mixture.definition = "mutationShared")
```

This will force all gene sets to share the same mutation parameters.

The same can be done with parameters describing selection, using the keyword **selectionShared**

```
parameter <- initializeParameterObject(genome = genome,
                        sphi = c(0.5, 2), num.mixtures = 2,
                        gene.assignment = sample.int(2,
                                length(genome), replace = T),
                        mixture.definition = "selectionShared")
```

For more intricate compositions of gene sets, one can specify a custom $n \times 2$ matrix, where n is the number of gene sets, to describe how gene-set specific parameters should be shared. Instead of using the **mixture.definition** argument one uses the **mixture.definition.matrix** argument.

The matrix representation of **mutationShared** can be obtained by

```
#  [,1] [,2]
#[1,] 1 1
#[2,] 1 2
#[3,] 1 3
def.matrix <- matrix(c(1,1,1,1,2,3), ncol=2)
parameter <- initializeParameterObject(genome = genome,
                        sphi = c(0.5, 2, 1), num.mixtures = 3,
                        gene.assignment = sample.int(3,
                                length(genome), replace = T),
                        mixture.definition.matrix = def.matrix)
```

Columns represent mutation and selection, while each row represents a gene set. In this case we have three gene sets, each sharing the same mutation category and three different selection categories. In the same way one can produce the matrix for three independent gene sets equivalent to the **allUnique** keyword.

```
#  [,1] [,2]
#[1,] 1 1
#[2,] 2 2
```

```
#[3,] 3 3
def.matrix <- matrix(c(1,2,3,1,2,3), ncol=2)
```

We can also use this matrix to produce more complex gene set compositions.

```
# [,1] [,2]
#[1,] 1 1
#[2,] 2 1
#[3,] 1 2
def.matrix <- matrix(c(1,2,1,1,1,2), ncol=2)
```

In this case gene set one and three share their mutation parameters, while gene set one and two share their selection parameters.

Checkpointing

AnaCoDa does provide checkpointing functionality in case runtime has to be restiricted. To enable checkpointing, one can use the function **setRestartSettings**.

```
# writing a restart file every 1000 samples
setRestartSettings(mcmc, "restart_file", 1000, write.multiple=TRUE)
# writing a restart file every 1000 samples
# but overwriting it every time
setRestartSettings(mcmc, "restart_file", 1000, write.multiple=FALSE)
```

To re-initialize a parameter object from a restart file one can simply pass the restart file to the initialization function:

```
initializeParameterObject(init.with.restart.file="restart_file.rst")
```

Load and save parameter objects

AnaCoDa is based on C++ objects using the Rcpp [19]. This comes with the problem that C++ objects are by default not serializable and can therefore not be saved/loaded with the default R save/load functions.

AnaCoDa however, does provide functions to load and save parameter and mcmc objects. These are the only two objects that store information during a run.

```
#save objects after a run
runMCMC(mcmc = mcmc, genome = genome, model = model)
writeParameterObject(parameter = parameter, file = "parameter.Rda")
writeMCMCObject(mcmc = mcmc, file = "mcmc_out.Rda")
```

As **genome**, and **model** objects are purely storage containers, no save/load function is provided at this point, but will be added in the future.

```
#load objects
parameter <- loadParameterObject(file = "parameter.Rda")
mcmc <- loadMCMCObject(file = "mcmc_out.Rda")
```

2.3.3 File formats

protein coding sequence

Protein coding sequences are provided by fasta file with the default format. One line containing the sequence id starting with > followed by the id and one or more lines containing the sequence. The sequences are expected to have a length that is a multiple of three. If a codon can not be recognized (e.g AGN) it is ignored.

```
>YAL001C
TTGGTTCTGACTCATTAGCCAGACGAACTGGTTCAA
CATGTTTCTGACATTCATTCTAACATTGGCATTCAT
ACTCTGAACCAACTGTAAGACCATTCTGGCATTTAG
>YAL002W
TTGGAACAAAACGGCCTGGACCACGACTCACGCTCT
TCACATGACACTACTCATAACGACACTCAAATTACT
TTCCTGGAATTCCGCTCTTAGACTCAACTGTCAGAA
```

Empirical gene expression

Empirical expression or gene specific parameters are provided in a csv file format. The first line is expected to be a header describing each column. The first column is expected be the gene id, and every additional column is expected to be represent a measurement. Each row corresponds to one gene and contains all measurements for that gene, including missing values.

```
>YAL001C
ORF,DATA_1,DATA_2,...DATA_N
YAL001C,0.254,0.489,...,0.156
YAL002W,1.856,1.357,...,2.014
YAL003W,10.45,NA,...,9.564
YAL005C,0.556,0.957,...,0.758
```

Ribosome foot-printing counts

Ribosome foot-printing (RFP) counts are provided in a csv file format. The first line is expected to be a header describing each column. The columns are expected in the following order gene id, position, codon, rfpcount. Each row corresponds to a single codon with an associated number of ribosome footprints.

```
GeneID,Position,Codon,rfpCount
YBR177C, 0, ATA, 8
YBR177C, 1, CGG, 1
YBR177C, 2, GTT, 8
YBR177C, 3, CGC, 1
```

2.3.4 Analyzing and Visualizating results

Parameter estimates

After we have completed the model fitting, we are interested in the results. AnaCoDa provides functions to obtain the posterior estimate for each parameter. For gene-set specific parameters or codon specific parameters we can use the function **getCSPEstimates**. Again we can specify for which mixture we would like the posterior estimate and how many samples should be used. **getCSPEstimates** has an optional argument filename which will cause the routine to write the result as a csv file instead of returning a **data.frame**.

```
csp_mat <- getCSPEstimates(parameter = parameter, CSP="Mutation",
                        mixture = 1, samples = 1000)
head(csp_mat)
# AA Codon Posterior 0.025% 0.975%
#1 A GCA -0.2435340 -0.2720696 -0.2165220
#2 A GCC 0.4235546 0.4049132 0.4420680
#3 A GCG 0.7004484 0.6648690 0.7351707
#4 C TGC 0.2016298 0.1679025 0.2387024
#5 D GAC 0.5775052 0.5618199 0.5936979
#6 E GAA -0.4524295 -0.4688044 -0.4356677
getCSPEstimates(parameter = parameter, filename = "mutation.csv",
                        CSP="Mutation", mixture = 1, samples = 1000)
```

To obtain posterior estimates for the gene specific parameters, we can use the function **getExpressionEstimatesForMixture**. In the case below we ask to get the gene specific parameters for all genes, and under the assumption each gene is assigned to mixture 1.

```
phi_mat <- getExpressionEstimates(parameter = parameter,
                        gene.index = 1:length(genome),
                        samples = 1000)
head(phi_mat)
```

```
# PHI log10.PHI Std.Error log10.Std.Error 0.025 0.975 log10.025 ...
#[1,] 0.2729446 -0.6188447 0.0001261525 2.362358e-04 0.07331819 ...
#[2,] 1.4221716 0.1498953 0.0001669425 5.194123e-05 1.09593642 ...
#[3,] 0.7459888 -0.1512764 0.0002313539 1.529267e-04 0.31559618 ...
#[4,] 0.6573082 -0.2030291 0.0001935466 1.400333e-04 0.31591233 ...
#[5,] 1.6316901 0.2098120 0.0001846631 4.986347e-05 1.28410352 ...
#[6,] 0.6179711 -0.2286806 0.0001744928 1.374863e-04 0.28478950 ...
```

However we can decide to only obtain certain gene parameters. in the first case we sample 100 random genes.

```
# sampling 100 genes at random
phi_mat <- getExpressionEstimates(parameter = parameter,
                        gene.index = sample(1:length(genome), 100),
                        samples = 1000)
```

Furthermore, AnaCoDa allows to calculate the selection coefficient s for each codon and each gene. We can use the function **getSelectionCoefficients** to do so. Please note, that this function returns the log(sNe).

getSelectionCoefficients returns a matrix with log(sNe) relative to the most efficient synonymous codon.

```
selection.coefficients <- getSelectionCoefficients(genome = genome,
                        parameter = parameter, samples = 1000)
head(selection.coefficients)
# GCA GCC GCG GCT TGC TGT GAC GAT ...
#SAKL0A00132g -0.1630284 -0.008695144 -0.2097771 0 -0.1014373 ...
#SAKL0A00154g -0.8494558 -0.045305847 -1.0930388 0 -0.5285367 ...
#SAKL0A00176g -0.4455753 -0.023764823 -0.5733448 0 -0.2772397 ...
#SAKL0A00198g -0.3926068 -0.020939740 -0.5051875 0 -0.2442824 ...
#SAKL0A00220g -0.9746002 -0.051980440 -1.2540685 0 -0.6064022 ...
```

```
#SAKL0A00242g -0.3691110 -0.019686586 -0.4749542 0 -0.2296631 ...
```

We can compare these values to the weights from the codon adaptatoin index (CAI) citepSharp1987 or effective number of codons (N_c) [?] by using the functions **getCAIweights** and **getNcAA**.

```
cai.weights <- getCAIweights(referenceGenome = genome)
head(cai.weights)
# GCA GCC GCG GCT TGC TGT
#0.7251276 0.6282192 0.2497737 1.0000000 0.6222628 1.0000000
nc.per.aa <- getNcAA(genome = genome)
head(nc.per.aa)
# A C D E F G ...
#SAKL0A00132g 3.611111 1.000000 2.200000 2.142857 1.792453 ...
#SAKL0A00154g 1.843866 2.500000 2.035782 1.942505 1.986595 ...
#SAKL0A00176g 5.142857 NA 1.857143 1.652174 1.551724 3.122449 ...
#SAKL0A00198g 3.800000 NA 1.924779 1.913043 2.129032 4.136364 ...
#SAKL0A00220g 3.198529 1.666667 1.741573 1.756757 2.000000 ...
#SAKL0A00242g 4.500000 NA 2.095890 2.000000 1.408163 3.734043 ...
```

We can also compare the distribution of selection coefficients to the CAI values estimated from a reference set of genes.

```
selection.coefficients <- getSelectionCoefficients(genome = genome,
                        parameter = parameter, samples = 1000)
s <- exp(selection.coefficients)
cai.weights <- getCAIweights(referenceGenome = ref.genome)
codon.names <- colnames(s)
h <- hist(s[, 1], plot = F)
plot(NULL, NULL, axes = F, xlim = c(0,1),
                        ylim = range(c(0,h$counts)),
                        xlab = "s", ylab = "Frequency",
                         main = codon.names[1], cex.lab = 1.2)
lines(x = h$breaks, y = c(0,h$counts), type = "S", lwd=2)
abline(v = cai.weights[1], lwd=2, lty=2)
axis(1, lwd = 3, cex.axis = 1.2)
axis(2, lwd = 3, cex.axis = 1.2)
```

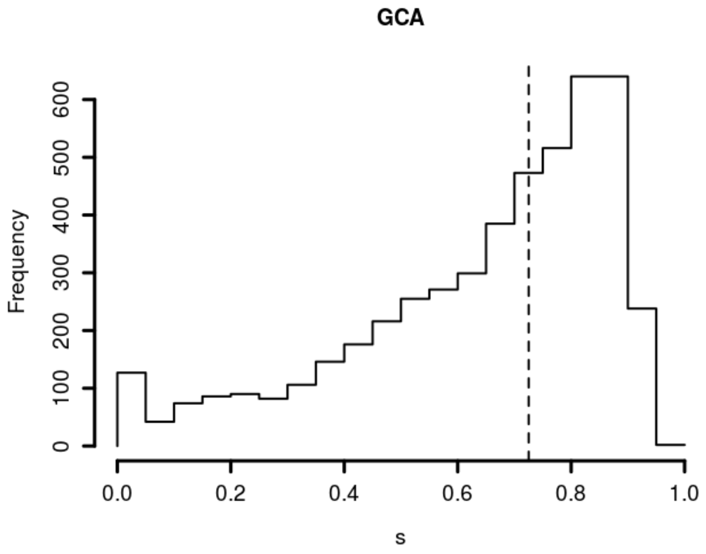


Figure 2.1: Distribution of s for codon GCA for amino acid alanine. Dashed line indicates the CAI weight for GCA. The comparisson provides a more nuanced picture as we can see that the selection on GCA varies across the genome.

Diagnostic Plots

A first step after every run should be to determine if the sampling routine has converged. To do that, AnaCoDa provides plotting routines to visualize all sampled parameter traces from which the posterior sample is obtained. First we have to obtain the **trace** object stored within our **parameter** object. Now we can simply plot the **trace** object. The argument **what** specifies which type of parameter should be plotted. Here we plot the selection parameter $\Delta\eta$ of the ROC model. These parameters are mixture specific and one can decide which mixture set to visualize using the argument **mixture**.

```
trace <- getTrace(parameter)
plot(x = trace, what = "Mutation", mixture = 1)
```

A special case is the plotting of traces of the protein synthesis rate ϕ. As the number of traces for the different ϕ traces is usually in the thousands, a **geneIndex** has to be passed to determine for which gene the trace should be plotted. This allows to inspect the trace of every gene under every mixture assignment.

```
trace <- parameter$getTraceObject()
plot(x = trace, what = "Expression", mixture = 1, geneIndex = 669)
```

We can find the likelihood and posterior trace of the model fit in the **mcmc object**. The trace can be plotted by just passing the **mcmc** object to the **plot** routine. Again we can switch between $\log(likelihood)$ and $\log(posterior)$ using the argument **what**. The argument **zoom.window** is used to inspect a specified window in more detail. It defaults to the last 10 % of the trace. The $\log(posterior)$ displayed in the figure title is estimated over the **zoom.window**.

```
plot(mcmc, what = "LogPosterior", zoom.window = c(9000, 10000))
```

Model visualization

We can visualize the results of the model fit by plotting the **model** object. For this we require the model and the **genome** object. We can adjust which mixture set we would like to visualize and how many samples should be used to obtain the posterior estimate for each parameter. For more details see Gilchrist et al. [29].

```
# use the last 500 samples from mixture 1 for posterior estimate.
plot(x = model, genome = genome, samples = 500, mixture = 1)
```

As AnaCoDa is designed with the idea to allow gene-sets to have independent gene-set specific parameters, AnaCoDa also provides the option to compare different gene-sets by plotting the parameter object. Here we compare the selection parameter estimated by ROC for seven yeast species.

```
# use the last 500 samples from mixture 1 for posterior estimate.
plot(parameter, what = "Selection", samples = 500)
```

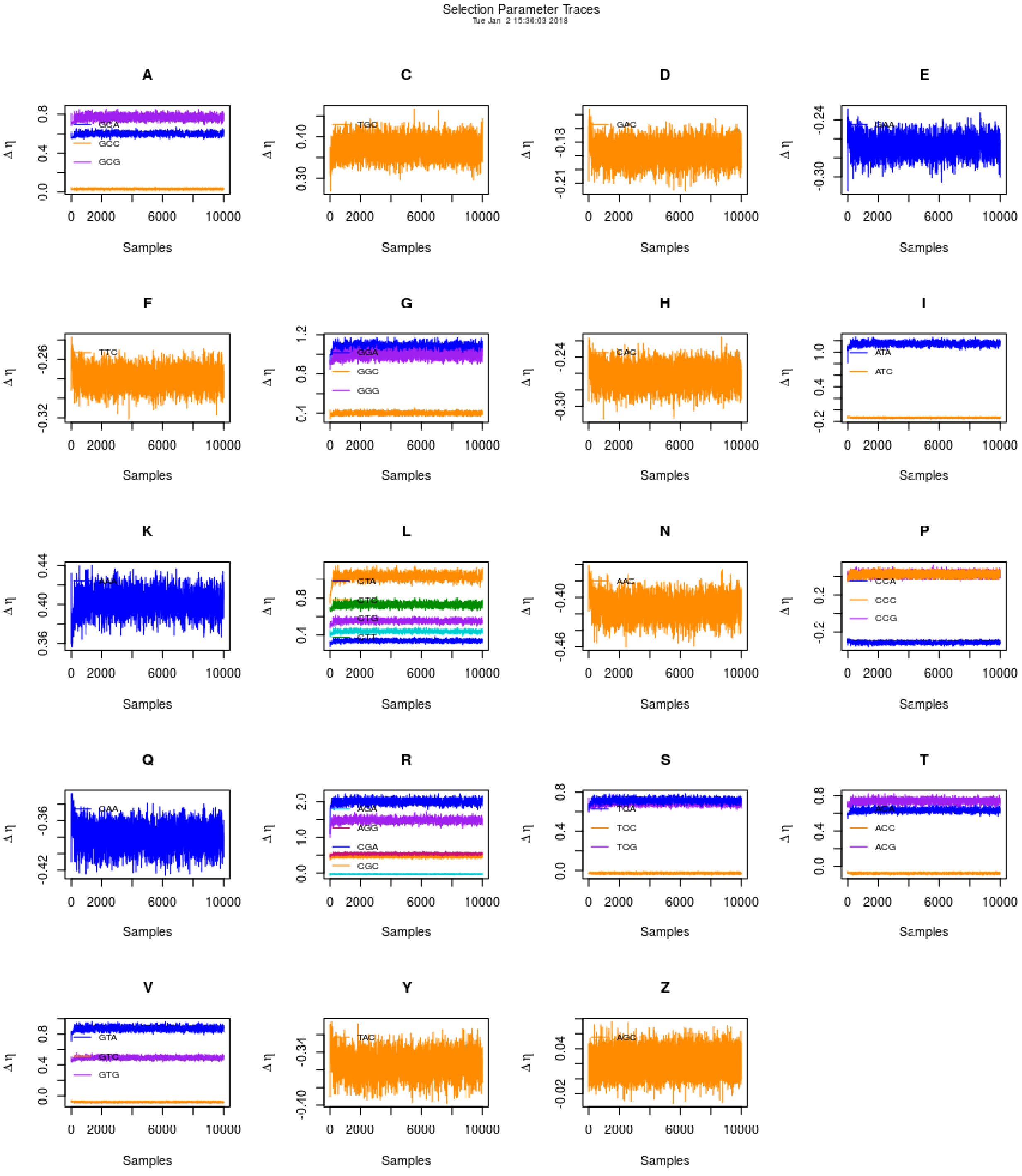


Figure 2.2: Trace plot showing the traces of all 40 codon specific selection parameters $\Delta\eta$ organized by amino acid.

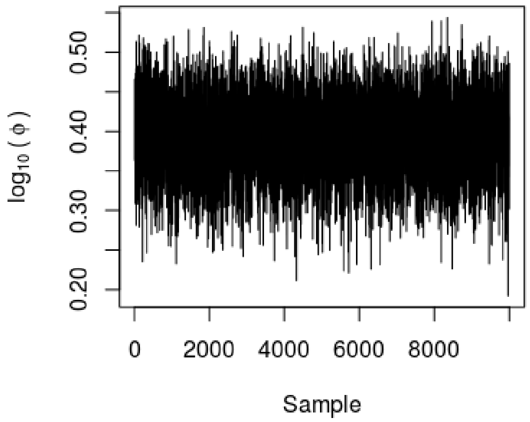


Figure 2.3: Trace plot showing the protein synthesis trace ϕ for gene 669.

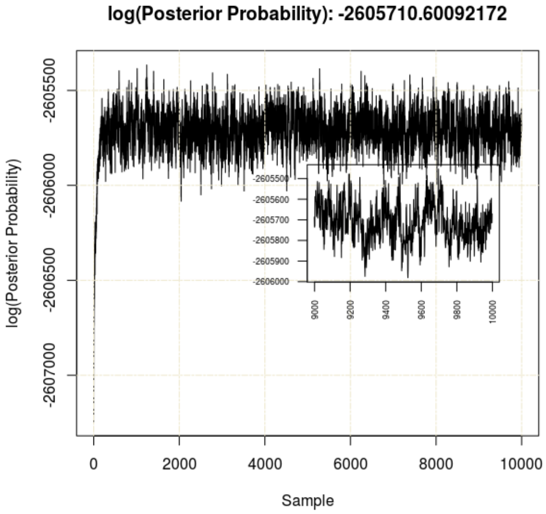


Figure 2.4: Trace plot showing the $\log(Posterior)$ trace for the current model fit. Window inset shows the last 1.000 samples

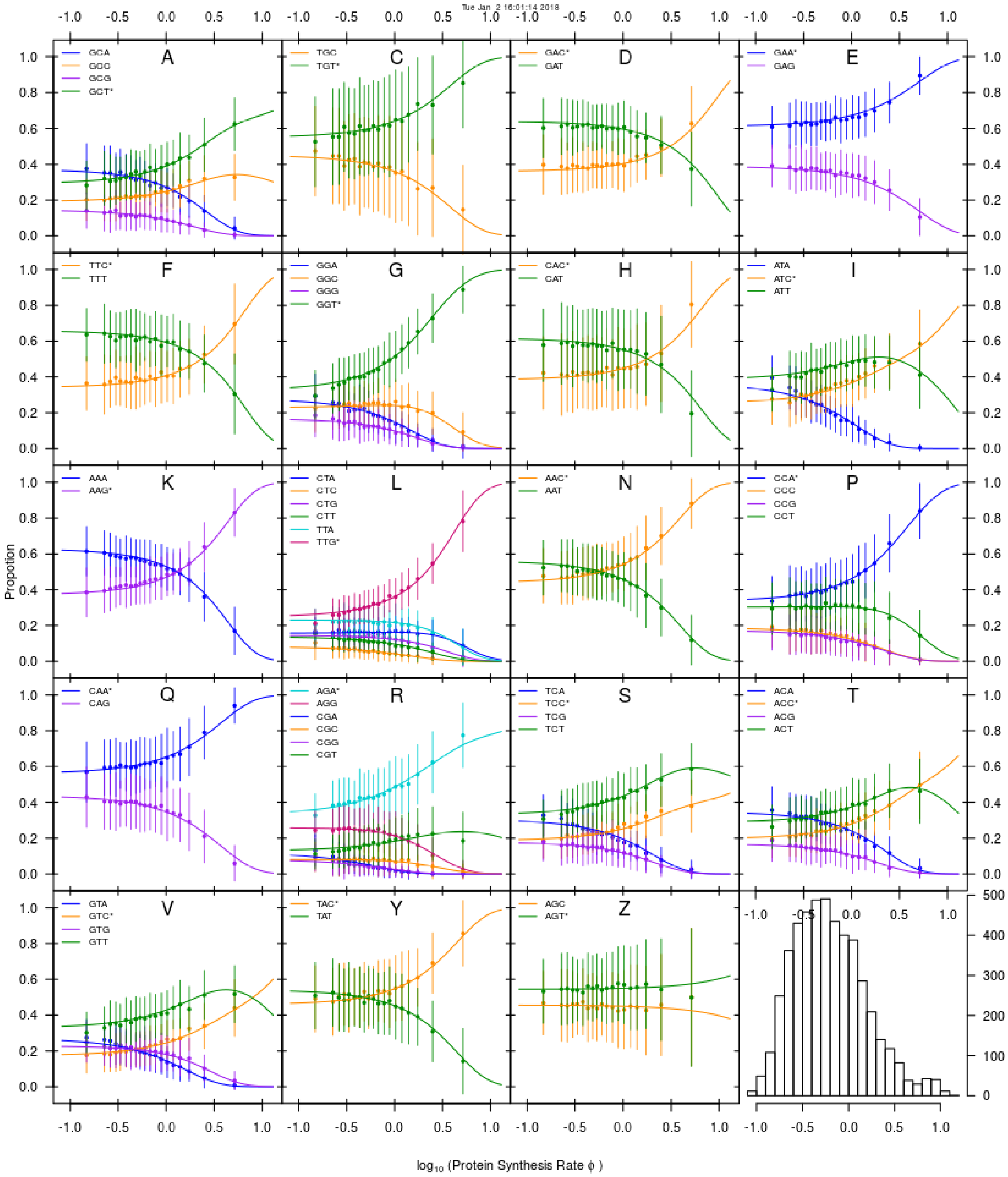


Figure 2.5: Fit of the ROC model for a random yeast. The solid line represent the model fit from the data, showing how synonymous codon frequencies change with gene expression. The points are the observed mean frequencies of a codon in that synthesis rate bin and the wisks indicate the standard diviation within the bin. The codon favored by selection is indicated by a ”*”. The bottom right panel shows how many genes are contained in heach bin

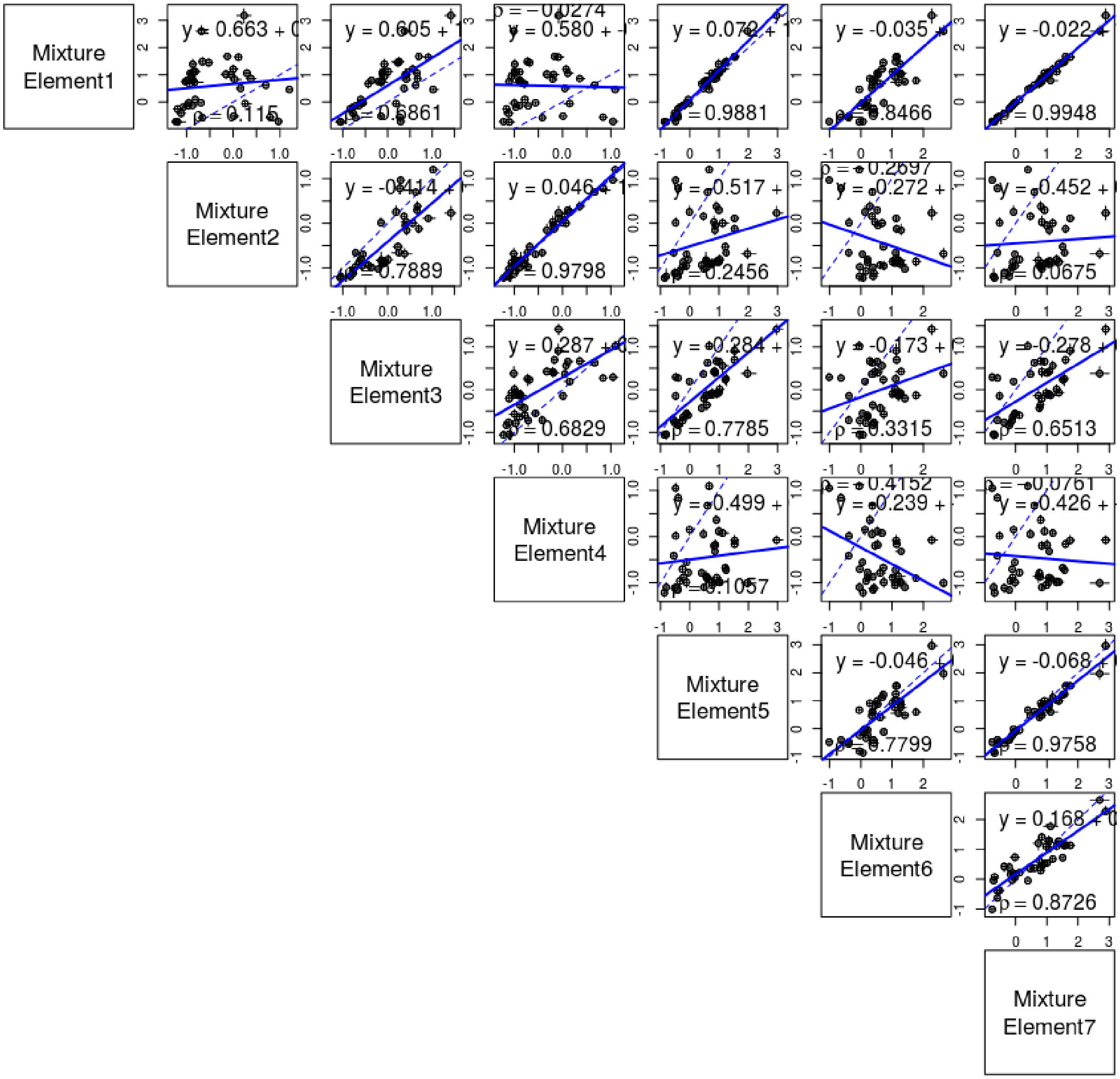


Figure 2.6: Comparisson of the selection parameter of seven yeast species estimated with ROC-SEMPPR.

Chapter 3

Decomposing mutation and selection to identify mismatched codon usage

This chapter is a lightly revised version of a paper to be submitted to Genome Biology and Evolution and co-authored with Michael A. Gilchrist and Russel Zaretzki.

C. Landerer, R. Zaretzki, M.A. Gilchrist, Decomposing mutation and selection to identify mismatched codon usage

Abstract

Codon usage has been used as a measure for adaptation of genes to their cellular environment for decades. The introgression of genes from one cellular environment to another may cause well adapted genes to suddenly be less adapted due to them having evolved in a different environment. As a result, we expect that transferred genes result in a large fitness burden for the new host organism. Here we examine the yeast *Lachancea kluyveri* which has experienced a large introgression, replacing the left arm of chromosome C ($\sim 10\%$ of its genome). The *L. kluyveri* genome provides an opportunity to study the evolution of introgressed genes to a novel cellular environment and estimate the fitness cost such a transfer imposes. We quantified the effects of mutation bias and selection against translation inefficiency on the codon usage pattern of the endogenous and introgressed exogenous genes using a Bayesian mixture framework. We found substantial differences in codon usage between the endogenous and exogenous genes, and show that these differences can be largely attributed to a shift in mutation bias from A/T ending codons in the endogenous genes to C/G ending codons in the exogenous genes. Recognizing the two different signatures of mutation and selection bias improved our ability to predict protein synthesis rate by 17% and allowed us to accurately assess codon preferences. In addition we utilize the estimates of mutation bias and selection against translation inefficiency to determine *Eremothecium gossypii* as potential source lineage, estimate the time since introgression to be on the order of 6×10^8 and assess the fitness burden across introgressed loci, showing the advantage of mechanistic models when analyzing codon data.

3.1 Introduction

Mutation, selection and genetic drift can be used to quantify the environment a genome has evolved in. Mutation bias is purely determined by the cellular environment, while the strength and efficacy of selection relative to drift is determined by the cellular environment, e.g. tRNA abundance, and the natural environment e.g. gene expression. A lineages effective population size determines the efficacy of selection relative to drift. Synonymous codon usage, the non-uniform usage of codons encoding the same amino acid, is a reflection of both, the cellular and the natural environment. Decomposing codon usage, therefore, provides us the necessary information to describe the environment a genome has evolved in in terms of its codon usage.

In general, the strength of selection on codon usage increases with gene expression [43, 34]. Conversely, the impact of mutation bias on codon usage declines with gene expression. Thus, we can easily imagine that with increasing gene expression, codon usage shifts from a process dominated mutation to a process dominated by selection. Together, the mutation process favoring specific synonymous codons - or mutation bias - and the selection for translation efficiency scaled by gene expression and effective population size - or selection bias - shape codon usage in a genome. This mutation-selection-drift balance model allows us to explicitly describe the environment in which genes evolve with respect to mutation and selection bias. Here we show that estimating the influence of mutation bias and selection bias on a gene's codon usage allows us to not only predict protein synthesis rate ϕ, but also, to infer its history and make predictions about its future with respect to these forces.

Most studies implicitly assume that synonymous codon usage of a genome is shaped by a single cellular environment. However, it is easy to think about the influence of multiple cellular environments within a cell, as genes are horizontally transferred, introgress, or as species hybridize. Genes introduced via horizontal gene transfer, introgression, or hybridization may carry the signature of a different, foreign cellular environment. These transferred genes may be less adapted to their new cellular environment, potentially imposing large fitness burdens to the organism. We expect a greater fitness burden of transferred genes if donor and recipient environment differ greatly in their selection bias, making such transfers less likely. More practically, if transferred genes are unaccounted for, they may distort parameters by biasing estimates. This can lead to the conclusion of the wrong codon preference for an amino acid when analyzing a genome that has experienced such transfer events.

In this study, we analyze the synonymous codon usage of the genome of *Lachancea kluyveri*, the earliest diverging lineage of the Lachancea clade. The Lachancea clade diverged from the Saccharomyces clade prior to the whole genome duplication, about 100 Mya ago. Since its divergence from the other Lachancea, *L. kluyveri* has experienced a large introgression of exogenous genes. The introgression replaced the left arm of the C chromosome and displays a 13% higher GC content than the endogenous *L. kluyveri* genome [65, 27]. These characteristics make *L. kluyveri* an ideal model to study the effects of an introgressed cellular environment and the resulting mismatch codon usage.

Using ROC SEMPPR, a population genetics Bayesian model, allows us to quantify the cellular environment in which genes have evolved by separating and estimating effects of mutation bias and selection bias, and predicting protein production rate [29]. We use ROC SEMPPR to describe two cellular environments reflected in the *L. kluyveri* genome, a native endogenous and an introgressed exogenous environment. Our results indicate that the difference in GC content between endogenous and exogenous genes mostly to differences in mutation bias. Recognizing the differences in codon usage between the endogenous and exogenous gene sets also improves our ability to predict protein synthesis rate from the sequence data alone.

With our improved model fits, we obtained more reliable estimates of mutation bias, selection bias and protein synthesis rate, allowing us to address more refined questions of biological importance. First we determine a potential source lineage of the exogenous genes using a combination of information in codon usage and gene synteny. We compared estimates of mutation bias (ΔM) and selection bias ($\Delta\eta$) for the exogenous genes to 38 yeast lineages and further investigated candidate lineages using synteny. Second, we estimate the time since introgression and the persistence of the signal of the exogenous cellular environment from our estimates of ΔM using an exponential model of decay. Third, we estimate the selective cost of the mismatched codon usage for the introgression, using our estimates of $\Delta\eta$ and protein synthesis rate ϕ. Thus, in addition to being able to estimate codon preference and gene expression to describe codon usage patterns, we also gain insights into the evolution of genes that have been transferred between lineages.

3.2 Results

Model selection and validation confirmed that the *L. kluyveri* genome contains signatures of at least two cellular environments. We compared model fits of ROC SEMPPR to the homogeneous *L. kluyveri* genome and the separated sets of endogenous and exogenous genes of 4864 and 497 genes respectively, using AnaCoDa [50]. We compared estimates of the cellular environment to describe differences in endogenous and exogenous codon usage. Furthermore, we utilize the differences in model fit and parameters estimated from the endogenous and exogenous genes to explore the evolution of the exogenous gene set.

AIC indicates that parameter estimates for mutation bias (ΔM) and selection bias ($\Delta\eta$) differ greatly between exogenous and endogenous gene sets. As a result, the partitioning of the *L. kluyveri* genome into an endogenous and exogenous gene set is clearly favored by model selection. The inclusion of 81 additional parameters (40 for ΔM, 40 for $\Delta\eta$, and one for s_ϕ) necessary to describe both gene sets separately improves our model fit by $\sim 75,000$ AIC units (5,311,060 for the combined gene set vs 5,235,598 for the separated gene sets).

In addition to model selection, we utilized independent information on gene expression to evaluate model fit. Recognizing differences in ΔM and $\Delta\eta$ for the endogenous and exogenous

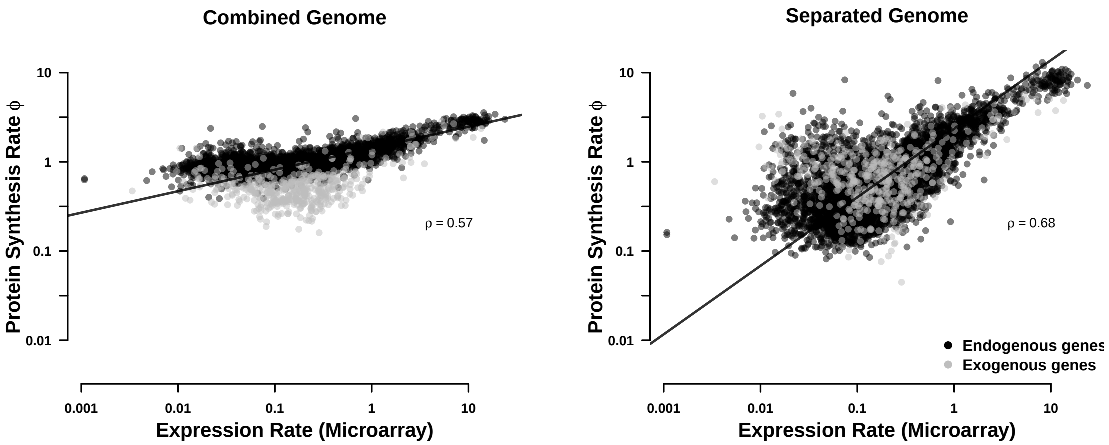


Figure 3.1: Comparison of predicted protein synthesis rate ϕ to Microarray data from Tsankov et al. [83] for (a) the combined genome and (b) the separated endogenous and exogenous genes. Endogenous genes are displayed in black and exogenous genes in gray. Black line indicates type II regression line.

gene sets substantially improves our ability to predict protein synthesis rate ϕ ($\rho = 0.69$ vs. $\rho = 0.59$ for the full genome; Figure 3.1).

3.2.1 Differences in the Endogenous and Exogenous Codon Usage

As our estimates of parameters for a codon family coding for an amino acid are relative to a reference codon, changes in the reference codon will change the order between sets. To better compare our estimates of of mutation bias (ΔM) and selection bias ($\Delta\eta$) obtained from fitting ROC SEMPPR between the endogenous and exogenous gene sets, we express our estimates relative to the mean for each codon family. As We find larger differences between ΔM than $\Delta\eta$ (Figure 3.2). Estimates of ΔM in the endogenous genes negatively correlate with the ΔM estimates for the exogenous genes ($\rho = -0.49$) indicating strong discordance in the mutation environment between *L. kluyveri* and the donor lineage of the exogenous genes. For example, $\sim$ 95% of codon families show mutation preference for A/T ending codons, in contrast, the exogenous genes display an equally strong mutation bias towards C/G ending codons. Only the two codon amino acid Phenylalanine (Phe, F) shows complete concordance between endogenous and exogenous genes in their ΔM values.

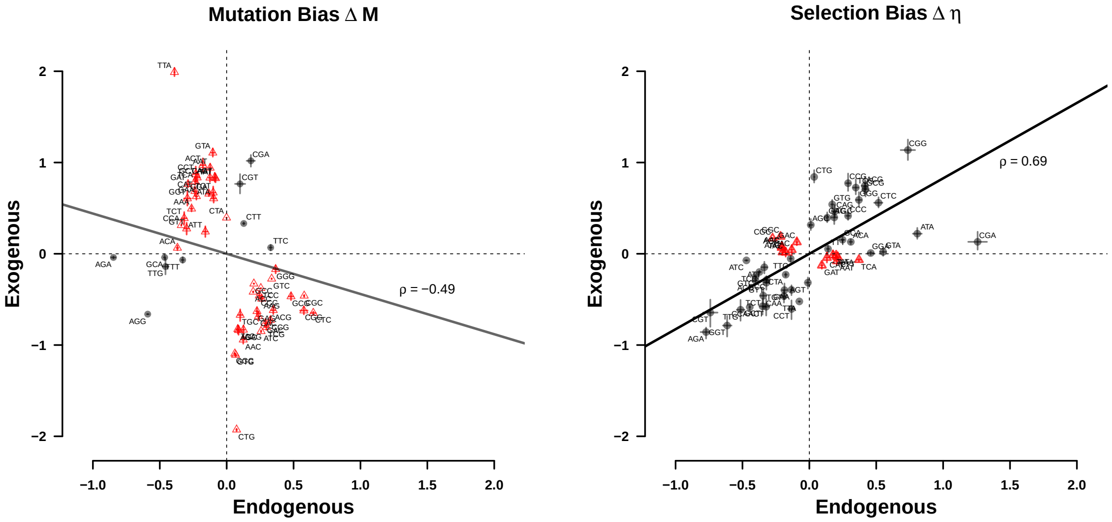


Figure 3.2: Comparison of (a) mutation bias ΔM and (b) selection bias $\Delta\eta$ of endogenous and exogenous genes. Estimates are relative to the mean for each codon family. Black dots indicate parameters with sign concordance, red dots indicate parameters with sign discordance between endogenous and exogenous genes. Black line shows the type II regression. Dashed lines mark quadrants.

Our estimates of $\Delta\eta$ for the endogenous and exogenous genes were positively correlated ($\rho = 0.69$), indicating increased concordance of $\sim 53\%$ between the two selection environments (Figure 3.2). Nevertheless, the endogenous genes only show a selection preference for C and G ending codons in $\sim 58\%$ of the codon families. In contrast, the exogenous genes display a strong preference for A and T ending codons in $\sim 89\%$ of the codon families.

The difference in codon preference between endogenous and exogenous genes is striking. Fits to the complete *L. kluyveri* genome reveal that the relatively small exogenous gene set has a disproportional effect on the model fit. We find that the complete *L. kluyveri* genome is estimated to share the mutational preference with the exogenous genes in $\sim 78\%$ of codon families with discordance between endogenous and exogenous genes. In two cases, Isoleucine (Ile, I) and Arginine (Arg, R), the strong discrodance in mutation preference results in a estiamted codon preference in the complete *L. kluyveri* genome that is not reflected by either endogenous nor exogenous genes.

The impact of the small exogenous gene set on the fit to the complete *L. kluyveri* genome is less prevalent in our estimates of selection bias $\Delta\eta$ but still strong. We find that the complete *L. kluyveri* genome is estimated to share the selection preference with the exogenous genes in $\sim 60\%$ of codon families with discordance between endogenous and exogenous genes. Therefore, it is important to recognize and treat endogenous and exogenous genes as separate sets to avoid the inference of incorrect synonymous codon preferences.

3.2.2 Determining Source of Exogenous Genes

We combined our estimates of mutation bias (ΔM) and selection bias ($\Delta\eta$) with synteny information and searched for potential source lineages of the introgressed region. We examined 38 yeast lineages of which two (*Eremothecium gossypii* and *Candida dubliniensis*) showed a strong positive correlation in codon usage (Figure 3.3).

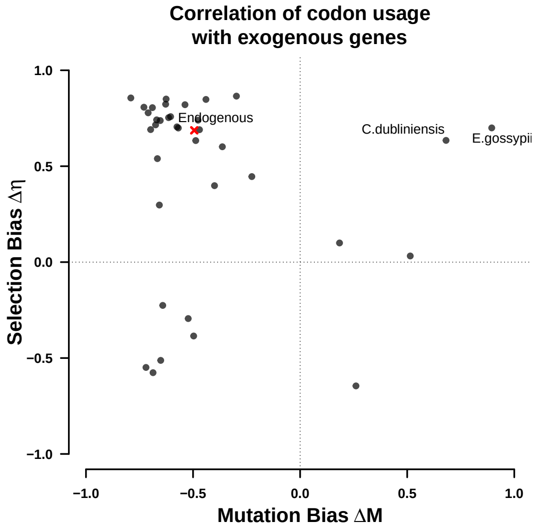


Figure 3.3: Correlation of ΔM and $\Delta\eta$ of the exogenous genes with 38 examined yeast lineages. Dots indicate the correlation of ΔM and $\Delta\eta$ of the lineages with the endogenous and exogenous parameter estimates. All regressions were performed using a type II regression.

The endogenous *L. kluyveri* genome exhibits codon usage very similar to most yeast lineages examined, indicating little variation in codon usage among the examined yeasts

(Figure 3.5). Four lineages show a positive correlation for ΔM and $\Delta\eta$ with the exogenous genes and have a weak to moderate positive correlation in selection bias with the endogenous genes; but, like the exogenous genes, tend to have a negative correlation in ΔM with the endogenous genes.

Comparing synteny between the exogenous left arm of chromosome C, and *E. gossypii* and *C. dubliniensis* as well as closely related yeast species we find that *E. gossypii* displays the highest synteny coverage (Figure 3.6, 3.7). *C. dubliniensis*, even though it displays similar codon usage does not show synteny with the exogenous region. Furthermore, the synteny relationship between the exogenous region and other yeasts appears to be limited to the Saccharomycetacease group(Figure 3.7). Given these results, we conclude that the *E. gossypii* lineage is the most likely source of the introgressed exogenous genes.

3.2.3 Estimating Introgression Age

We estimated the introgression assuming that *E. gossypii* is still representative of the mutation bias of its ancestral source lineage at the time of the introgression. We infer the age of the introgression to be on the order of $6.2 \pm 1.2 \times 10^8$ generations. *L. kluyveri* experiences between one and eight generations per day, we therefore expect the introgression to have occurred between $205,000$ to $1,600,000$ years ago. This estimates the introgression to be older than previous assumed Friedrich et al. [27]. However, our estimates are likely overestimates as they assume a purely neutral decay.

We also estimated the persistence of the signal of the foreign cellular environment. Assuming that differences in mutation bias will decay more slowly than differences in selection bias, we predict that the ΔM signal of the source cellular environment will have decayed to be within one percent of the *L. kluyveri* environment within about $5.4 \pm 0.2 \times 10^9$ generations.

3.2.4 Genetic Load of the Exogenous Genes

Estimates of selection bias for the exogenous genes show that, while well correlated with the endogenous genes, only nine amino acids share the optimal codon. We therefore expect that the introgressed genes represent a significant reduction in fitness, or genetic load for *L. kluyveri*, and even more so at the time of introgression. As the introgression occurred before the diversification of *L. kluyveri* and has fixed since then throughout the various populations, we are left without the original chromosome arm [27]. Using our estimates of ΔM and $\Delta\eta$ from the endogenous genes, we can estimate the genetic load of the exogenous genes relative to an expected gene set. We define genetic load as the difference between the fitness of an expected, replaced endogenous gene and the inferred introgressed gene relative to drift $s \propto \phi\Delta\eta$ (See Methods for details).

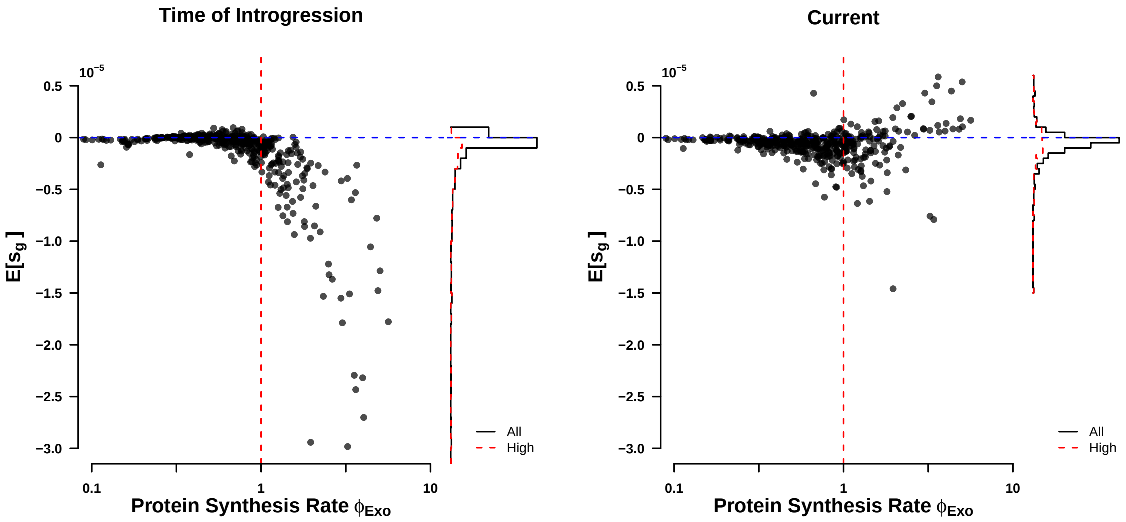


Figure 3.4: Fitness burden $\Delta s N_e$ (a) at the time of introgression ($\kappa = 5$), and (b) currently ($\kappa = 1$).

We estimate the genetic load of the exogenous genes at the time of introgression (Figure 3.4a) and currently (Figure 3.4b). As $\Delta\eta$ is defined as $\Delta\eta = 2N_e q(\eta_i - \eta_j)$, we can not distinguish if κ is a scaling on protein synthesis rate ϕ, effective population size N_e the value of an ATP q[29].

At the time of the introgression, we predict that only a few genes were weakly exapted (Figure 3.4a) with all high expression genes ($\phi > 1$) being maladapted to the novel cellular environment. However, these highly expressed genes show the greatest rate of adaptation to the *L. kluyveri* cellular environment (Figures 3.4a, 3.9).

3.3 Discussion

Using ROC SEMPPR we show that the *L. kluyveri* genome contains two distinct signatures of cellular environments, its own endogenous and a foreign exogenous one obtained by an introgression event ($\Delta AIC = 78,000$). Following Payen et al. [65], who defined the boundary of the anomalous chromosome region based on its elevated GC content, we partitioned the *L. kluyveri* genome into an endogenous and an exogenous gene set using gene location. We estimated the codon usage of the entire *L. kluyveri* genome and the separated endogenous and exogenous gene sets (Figure 3.10). Both, Mutation bias and selection bias differ between endogenous and exogenous genes. The endogenous genes show a strong mutation bias towards A/T ending codons, while the exogenous genes show mutation is bias towards G/C ending codons. We observed the reversed to be true in selection bias, leading to a strong mismatch in codon usage between the gene sets, supporting our notion of two distinct signatures of codon usage.

Only half of the codon families share the same optimal codon in the endogenous and exogenous gene sets. However, we find that the strength of selection within a codon family differs between gene sets, causing a change in rank order. Nevertheless, we find a high correlation for our estimates of selection bias $\Delta\eta$ between the two gene sets. Our estimates of the optimal codon differ in nine cases between endogenous and exogenous genes. Interestingly, when the difference in codon usage is ignored, we find that in seven out of these nine cases the exogenous codon preference is inferred as optimal(Table 3.2). We find even greater discordance in our estimates of ΔM between endogenous and exogenous gene sets(Table 3.1). Without recognizing this difference in codon preference our estimates would not have been reflective of the actual codon usage of the *L. kluyveri* genome but of a relatively small introgressed gene set. This shows that a small number of exogenous genes ($\sim 9\%$ of genes) can have a disproportional impact on our estimates of ΔM and $\Delta\eta$ when fitting ROC SEMPPR to the entire *L. kluyveri* genome. While this is surprising, it highlights the importance to recognize differences in codon usage within a genome. Our results also indicate that we can attribute the higher GC content in the exogenous genes mostly to differences in mutation bias favoring G/C ending codons rather than a novel selective force.

Separating the endogenous and exogenous genes improves our estimates of protein synthesis rate ϕ by 17% relative to the full genome estimate ($\rho = 0.59$ vs. $\rho = 0.69$, respectively). Furthermore, we find that the variation in our estimates of ϕ is more consistent with the current understanding of gene expression (compare Figure 3.1a and b). Small variation in ϕ estimates may serve as an indicator for the presents of the signature of multiple cellular environments in future work. In the case of the *L. kluyveri* genome, finding a severe mismatch in ΔM causes ϕ values for low expression genes ($\phi < 1$) to increase towards the inflection point where the dominance of mutation gives way to selection. In the case of the two codon amino acids, the inflection point represents the point at which mutation and selection are contributing equally to the probability of a codons occurrence. We find this inflection point around $\phi = 1$ for most amino acids (Figure 3.10). However, ROC SEMPPR assumes that estimates of ϕ follow a log-normal distribution with an expected value $E[\phi] = 1$. This assumption allows us to interpret $\Delta\eta$ as the strength of selection relative to drift (sN_e) for a codon in a gene with the average protein synthesis rate $\phi = 1$. However, tying the mean and standard deviation of the prior distribution together. Therefore, an increase in ϕ for low expression genes has to be meet with a decrease of ϕ for high expression genes, reducing the overall variance in ϕ (see Gilchrist et al. [29] for details).

Having shown that the introgressed exogenous genes reflect a foreign cellular environment, we used the quantitative estimates of mutation bias ΔM and selection bias $\Delta\eta$ from ROC

SEMPPR to identify potential source lineages. The comparison of the endogenous and exogenous ΔM and $\Delta\eta$ estimates to 38 other yeast lineages revealed that most yeasts examined share similarity in mutation bias (Figure 3.2). Similar, we find strong similarities in selection bias between examined yeasts, potentially indicating stabilizing selection on codon usage. However, the exogenous genes do not share this commonality (Figure 3.2a), as their mutation bias strongly deviates from the endogenous genes and most other yeast species examined. This large difference in mutation bias between endogenous and exogenous genes allowed us to limit our candidate list to only two likely lineages, *C. dubliniensis* and *E. gossypii*. Interestingly, we did not find *Lachancea thermotolerance*, a thermophilic lineage closely related to *L. kluyveri*, as a potential candidate. While *L. thermotolerance* does have a strong synteny relationship with *L. kluyveri*, it does not show similarity in codon usage with the exogenous genes and does not share their high GC content.

Inference of synteny relationships between the exogenous region and *C. dubliniensis* and *E. gossypii* as well as closely related species showed that synteny relationship is limited to the Saccharomycetaceae clade (Figure 3.7b). *E. gossypii* showed the highest syntenty coverage and is the only species with similar codon usage. Furthermore, *E. gossypii* is the only species examined with a GC content > 50% like it is observed in the exogenous region. The synteny coverage extends along the whole exogenous regions with the exception to the very 3' and 5' end of the region. The lack of synteny at the ends of the region also coincides with a drop in GC content, potentially indicating remains of the original replaced region or increased adaptation. The ancestral introgressed region may have also broken up in *E. gossypii* as we find non overlapping synteny with chromosomes *VI* and *V* as well as have indication that the C chromosome of *L. kluyveri* very robust to recombination events [65, 84].

With *E. gossypii* identified as potential source lineage of the introgressed region, we inferred the time past since the introgression occurred using our estimates of mutation bias ΔM. The ΔM estimates are well suited for this task as they are free of the influence of selection and unbiased by N_e and other scaling terms, which is in contrast to our estimates of

$\Delta\eta$ [29]. We estimated the time since introgression to be on the order of 6×10^8 generations, which is $\sim$ 10 times longer time than a previous estimate by Friedrich et al. [27] of a minimum of 5.6×10^7 generations . However, our estimate implicitly assumes all mutations are neutral, it is therefore a conservative estimate, potentially overestimating the time since introgression. Our estimate also depend on the assumption that the *E. gossypii* cellular environment reflects the ancestral environment at the time of the introgression. If the the ancestral mutation environment was more similar to the *L. kluyveri* environment at the time of the introgression than the *E. gossypii* environment is today, we would overestimate this time. On the other hand, we would underestimate the time since introgression if the two cellular environments were more dissimilar. We could have attempted to reconstruct the ancestral state of *E. gossypii*, however, as methods for ancestral state reconstruction are phenomenological, assumptions would be unclear.

The estimates of mutation bias ΔM also allow us the infer the time until the signature of the exogenous cellular environment will have decayed to be indistinguishable at about one percent difference. Our estimate of decay is an order of magnitude greater than our estimate of the time since introgression (5×10^9 and 6×10^8 generations). Estimates of decay based on ΔM are more conservative as we expect differences in $\Delta\eta$ to decay before due to selection favoring the decay.

As we have determined that the introgression event has a long persisting exogenous signature, it is important to understand the fitness consequences of such an event. We estimated the genetic load that the exogenous genes represent assuming that the replaced endogenous genes and the new exogenous genes had the same amino acid composition. This assumption, along with the assumption that the current *L. kluyveri* cellular environment is reflective of the cellular environment at the time of the introgression is necessary to estimate the expected endogenous sequence that was replaced. Our results show that individual low expression genes contribute little to the genetic load, and show less adaptation to the novel cellular environment (Figure 3.4, 3.9). A small number of low expression genes even appear exapted, likely due to the mutation bias in the endogenous genes matching the selection bias in the exogenous genes for G/C ending codons. Highly expressed genes on the other hand have greatly adapted to the *L. kluyveri* cellular environment. This, however, does not mean that these genes show a higher rate of evolution, but that small changes in their sequence have large impacts on the fitness burden these sequences represent. To this day, the exogenous genes represent a significant fitness burden on *L. kluyveri*. However, our estimates are conservative as we do not account for potential changes in the codon usage of *E. gossypii*. While divergent evolution in codon usage between *E. gossypii* and *L. kluyveri* would cause us to overestimate the genetic load, convergent evolution, on the other hand, would cause us to underestimate the genetic load. However, as the introgression appears to have reached fixation [27], the genetic load relative to the replaced chromosome arm is only of theoretical interest.

The large genetic load the exogenous genes represented at the time of the introgression indicates that the fixation of the introgression was a very unlikely event in a population with a large N_e as it is typical for yeasts. It is hard to contextualize the probability of this introgression being fixed as we are not aware of any estimates of the frequency at which such large scale introgressions of genes with very different signatures of codon usage occur. One example is *Saccharomyces bayanus*, a hybrid of *Saccharomyces uvarum*,*Saccharomyces cerevisiae*, and *Saccharomyces eubayanus*. However, unlike with *L. kluyveri* and *E. gossypii* it appears that the donor lineages show similar codon usage. *Saccharomyces cerevisiae* and *Saccharomyces eubayanus* show a very strong correlation between selection bias $\Delta\eta$ of $\rho = 0.98$ and a strong correlation between mutation bias ΔM of $\rho = 0.83$ We were unable to identify codon usage for *Saccharomyces uvarum*. However, *L. kluyveri* diverged about 85 Mya ago from the rest of the Lachancea clade. This represents between 10^{10} to 10^{11} generations. Assuming for yeasts typical effective population size on the order of 10^8, we are left with 10^{18} to 10^{19} opportunities for such an event to occur. In addition, the strong mutation bias towards G/C ending codons in the exogenous genes may have contributed to the fixation of this introgression (include figure of ΔM v $\Delta\eta$). It is, on the other hand, also possible that despite their mismatch in codon usage, the exogenous genes have represented a fitness increase due to external environmental factors resulting in the fixation of the introgression.

In conclusion, our results show the usefulness of the separation of mutation bias and selection bias and the importance of recognizing the presence of multiple cellular environments in the study of codon usage. We also illustrate how a mechanistic model like ROC SEMPPR and the quantitative estimates it provides can be used for more sophisticated hypothesis testing in the future. In contrast to other approaches used to study codon usage like CAI [73] or tAI [16], ROC SEMPPR is sensitive to differences in mutation bias. We highlight potential pitfalls when estimating codon preferences, as estimates can be biased by the signature of a second, historical cellular environment. In addition, we show how quantitative estimates of mutation bias and selection relative to drift can be obtained from codon data and used to infer the fitness cost of an introgression as well as its history and potential future.

3.4 Materials and Methods

3.4.1 Separating endogenous and exogenous genes

A GC-rich region was identified by Payen et al. [65] in the *L. kluyveri* genome extending from position 1 to 989,693 of chromosome C. This region was later identified as an introgression by Friedrich et al. [27]. We obtained the *L. kluyveri* genome from SGD Project http://www.yeastgenome.org/download-data/ (last accessed: 09-27-2014) and the annotation for *L. kluyveri* NRRL Y-12651 (assembly ASM14922v1) from NCBI (last accessed: 12-09-2014). We assigned 457 genes located on chromosome C with a location within the $\sim 1Mb$ window to the exogenous gene set. All other 4864 genes of the *L. kluyveri* genome were assigned to the exogenous genes. All genes could be uniquely assigned to one or the other gene set.

3.4.2 Model Fitting with ROC SEMPPR

ROC SEMPPR was fitted to each genome using AnaCoDa (0.1.1) [50] and R (3.4.1). ROC SEMPPR was run from multiple starting values for at least 250,000 iterations, every 50th sample was collected to reduce autocorrelation. After manual inspection to verify that the MCMC had converged, parameter posterior means were estimated from the last 500 samples.

3.4.3 Comparing codon specific parameter estimates

Choice of reference codon does reorganize codon families coding for an amino acid relative to each other, therefore all parameter estimates are relative to the mean for each codon family.

$$\Delta M^{c}_{i,a} = \Delta M_{i,a} - \Delta \bar{M}_{a} \tag{3.1}$$

$$\Delta \eta^{c}_{i,a} = \Delta \eta_{i,a} - \Delta \bar{\eta}_{a} \tag{3.2}$$

Comparison of codon specific parameters (ΔM and $\Delta\eta$) was performed using the function lmodel2 in the R package lmodel2 (1.7.3) and R version 3.4.1. Type II regression was performed with re-centered parameter estimates, accounting for noise in dependent and independent variable.

3.4.4 Synteny

We obtained complete genome sequences from NCBI (last accessed: 02-05-2017). Genomes were aligned and checked for synteny using SyMAP (4.2) with default settings [76, 75]. We assessed Synteny as percentage non-overlapping coverage of the exogenous gene region (Figure 3.7b).

3.4.5 Determining introgression timeline

We modeled the change in codon frequency over time using an exponential model for all two codon amino acids, and describing the change in codon c_1 as

$$\frac{dc_1}{dt} = -\mu_{1,2}c_1 - \mu_{2,1}(1 - c_1) \tag{3.3}$$

where $\mu_{i,j}$ is the rate at which codon i mutates to codon j and c_1 is the frequency of the reference codon. Our estimates of ΔM_{endo} can be directly related to the steady state of equation 3.3.

$$\frac{\mu_{2,1}}{\mu_{1,2} + \mu_{2,1}} = \frac{1}{1 + \exp(\Delta M_{endo})} \tag{3.4}$$

Solving for $\mu_{1,2}$ gives us $\mu_{1,2} = \Delta M_{endo} \exp(\mu_{2,1})$ which allows us to rewrite and solve equation 3.3 as

$$c_1(t) = \frac{\exp(-t(1 + \Delta M_{endo})\mu_{2,1}) \exp(t(1 + \Delta M_{endo})\mu_{2,1}) + (1 + \Delta M_{endo})K}{1 + \Delta M_{endo}} \tag{3.5}$$

where K is

$$K = \frac{-1 + c_1(0) + c_1(0)\Delta M_{endo}}{1 + \Delta M_{endo}} \tag{3.6}$$

Equation 3.5 was solved over time with a mutation rate $m_{2,1}$ of 3.8×10^{-10} per nucleotide per generation [51]. Initial codon frequencies $c_1(0)$ for each codon family where taken from our estimates of ΔM_{gos} from *E. gossypii*. Current codon frequencies for each codon family where taken from our estimates of ΔM from the exogenous genes. Mathematica (9.0.1.0) [Inc.] was used to calculate the time t_{exo} it takes for the initial codon frequencies $c_1(0)$ for each codon family to change to the current exogenous codon frequencies. The same equation was used to determine the time t_{endo} at which the signal of the exogenous cellular environment has decayed to within 1% of the endogenous environment.

Estimating Genetic Load

To estimate the fitness burden, we made three key assumptions. First, we assumed that the current exogenous amino acid composition of genes is representative of the replaced endogenous genes. Second, we assume that the currently observed cellular environment of *E. gossypii* reflects the cellular environment that the exogenous genes experienced before transfer to *L. kluyveri*. Lastly, we assume that the difference in the efficacy of selection between the cellular environments of the source lineage and *L. kluyveri* can be expressed as a scaling constant and that protein synthesis rate ϕ has not changed between the replaced endogenous and the introgressed exogenous genes.

Using estimates for $N_e = 1.36 \times 10^7$ [85] for *Saccharomyces paradoxus* we scale our estimates of $\Delta\eta$ and define $\Delta\eta' = \frac{\Delta\eta}{N_e}$. We calculated the fitness burden each gene represents assuming additive fitness effects as

$$s_g = \sum_i^C -\kappa\phi_g \Delta\eta'_i n_{g,i} \tag{3.7}$$

where s_g is the selection against translation inefficiency. ϕ_g is the estimated protein synthesis rate for gene g in the exogenous gene set. $n_{g,i}$ is the codon count of each codon i in the codon set C for each gene g. κ is a constant, scaling the efficacy of selection between cellular environments. Like stated previously, our $\Delta\eta$ are relative to the mean of the codon family. We find that the fitness burden of the introgressed genes is minimized at $\kappa \sim 5$ (Figure 3.8b). Thus, we set $\kappa = 1$ if we calculate the s_g for the endogenous and the current exogenous genes, and $\kappa = 5$ for s_g for the fitness burden at the time of introgression. Since we are unable to observe codon counts for the replaced endogenous genes and for the exogenous genes at the time of introgression, we calculate expected codon counts

$$E[n_{g,i}] = \frac{\exp(-\Delta M_i - \Delta\eta_i\phi_g)}{\sum_j^C \exp(-\Delta M_j - \Delta\eta_j\phi_g)} \times m_{a_i} \tag{3.8}$$

m_{a_i} is the number of occurrences of amino acid a that codon i codes for.

We report the genetic load of the introgression as $\Delta s = s_{Intro} - s_{Endo}$ where s_{Intro} is either the fitness burden at the time of the introgression or presently.

3.5 Acknowledgments

This work was supported in part by NSF Awards MCB-1120370 (MAG and RZ) and DEB-1355033 (BCO, MAG, and RZ) with additional support from The University of Tennessee Knoxville. CL received support as a Graduate Student Fellow at the National Institute for Mathematical and Biological Synthesis, an Institute sponsored by the National Science Foundation through NSF Award DBI-1300426, with additional support from UTK. The authors would like to thank Brian C. O'Meara and Alexander Cope for their helpful criticisms and suggestions for this work.

Amino Acid	*E. gossypii*	Endogenous	Exogenous	*L. kluyveri*
Ala A	GCG	GCA	GCG	GCG
Cys C	TGC	TGT	TGC	TGC
Asp D	GAC	GAT	GAC	GAC
Glu E	GAG	GAA	GAG	GAG
Phe F	TTC	TTT	TTT	TTT
Gly G	GGC	GGT	GGC	GGC
His H	CAC	CAT	CAC	CAC
Ile I	ATC	ATT	ATC	ATA
Lys K	AAG	AAA	AAG	AAA
Leu L	CTG	TTG	CTG	CTG
Asn N	AAC	AAT	AAC	AAT
Pro P	CCG	CCA	CCG	CCG
Gln Q	CAG	CAA	CAG	CAG
Arg R	CGC	AGA	AGG	CGG
Ser_4 S	TCG	TCT	TCG	TCG
Thr T	ACG	ACA	ACG	ACG
Val V	GTG	GTT	GTG	GTG
Tyr Y	TAC	TAT	TAC	TAC
Ser_2 Z	AGC	AGT	AGC	AGC

Table 3.1: Synonymous codon preference in the various data sets based on our estimates of ΔM

3.6 Supplementary Material

Supporting Materials for *Fitness consequences of mismatched codon usage* by Landerer *et al.*.

Amino Acid	*E. gossypii*	Endogenous	Exogenous	*L. kluyveri*
Ala A	GCT	GCT	GCT	GCT
Cys C	TGT	TGT	TGT	TGT
Asp D	GAT	GAC	GAT	GAT
Glu E	GAA	GAA	GAA	GAA
Phe F	TTT	TTC	TTC	TTC
Gly G	GGA	GGT	GGT	GGT
His H	CAT	CAC	CAT	CAT
Ile I	ATA	ATC	ATT	ATT
Lys K	AAA	AAG	AAA	AAG
Leu L	TTA	TTG	TTG	TTG
Asn N	AAT	AAC	AAT	AAC
Pro P	CCA	CCA	CCT	CCA
Gln Q	CAA	CAA	CAA	CAA
Arg R	AGA	AGA	AGA	AGA
Ser_4 S	TCA	TCC	TCT	TCT
Thr T	ACT	ACC	ACT	ACT
Val V	GTT	GTC	GTT	GTT
Tyr Y	TAT	TAC	TAT	TAC
Ser_2 Z	AGT	AGT	AGT	AGT

Table 3.2: Synonymous codon preference in the various data sets based on our estimates of $\Delta\eta$

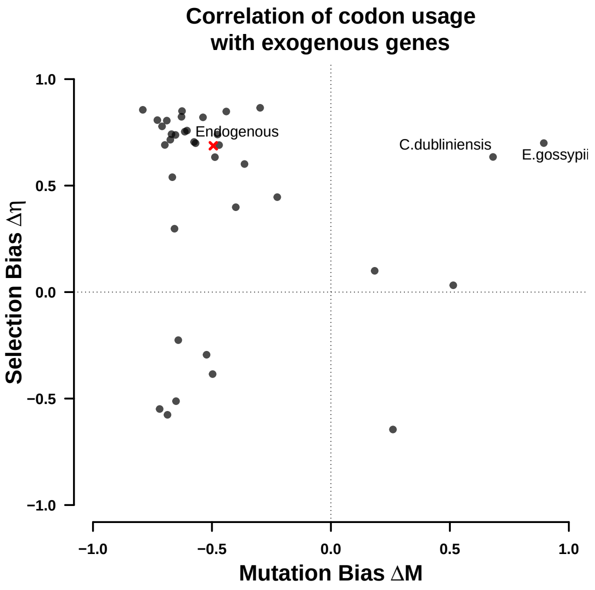


Figure 3.5: Correlation of ΔM and $\Delta\eta$ of the endogenous genes with 38 examined yeast lineages. Dots indicate the correlation of ΔM and $\Delta\eta$ of the lineages with the endogenous and exogenous parameter estimates. All regressions were performed using a type II regression.

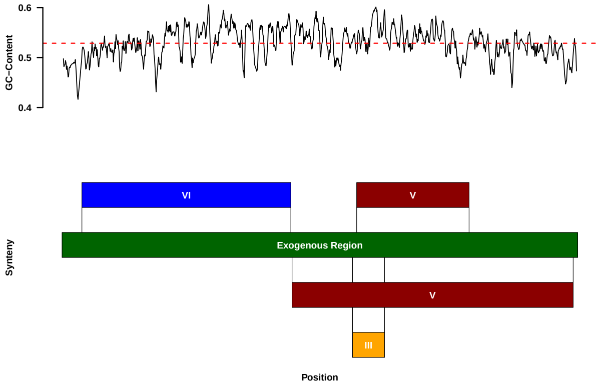


Figure 3.6: Synteny relationship of *E. gossypii* and the exogenous genes. Indicated is the GC content along the introgression.

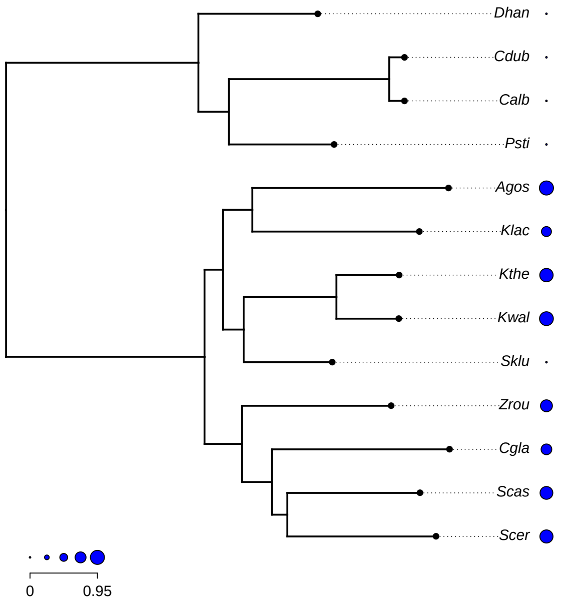


Figure 3.7: Amount of synteny for each species in units of standard deviations for selected species.

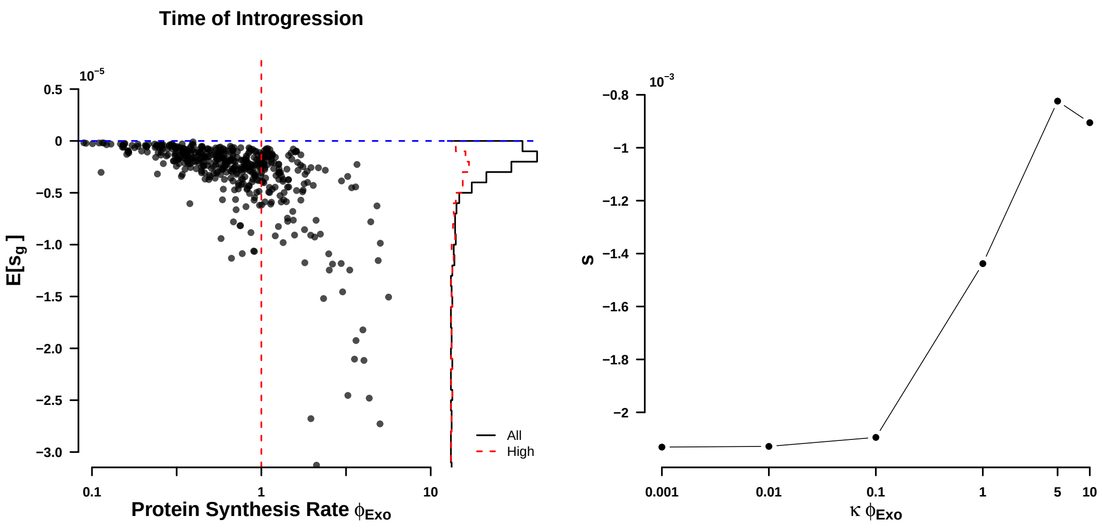


Figure 3.8: Genetic load (left) without scaling of ϕ per gene, and change of total genetic load with scaling κ between *E. gossypii* and *L. kluyveri* (right)

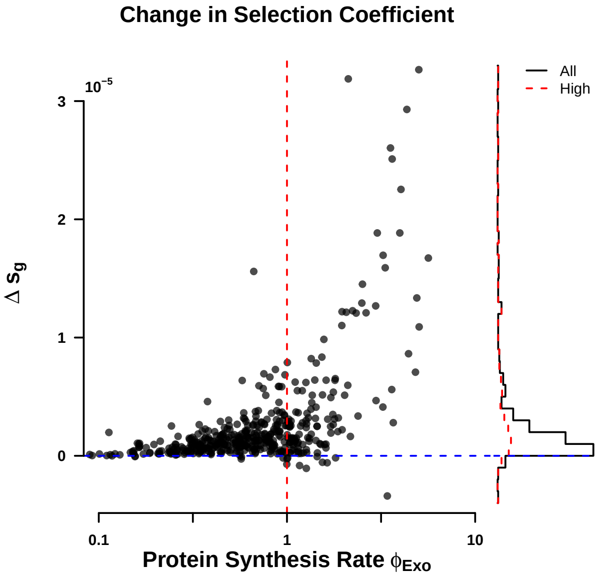


Figure 3.9: Total amount of adaptation estimated to have occured between time of introgression and currently observed per gene.

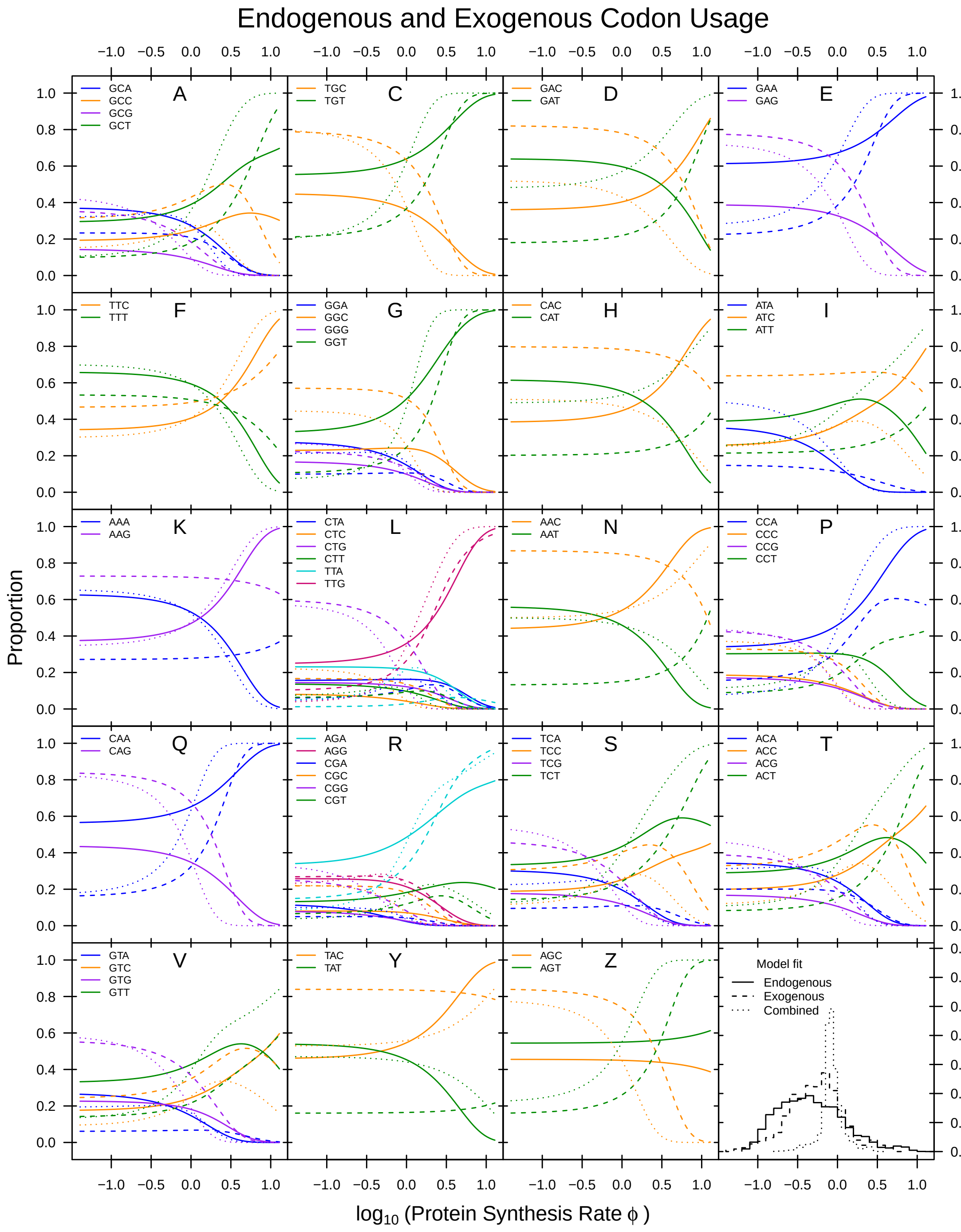


Figure 3.10: Codon usage patterns for 19 amino acids. Amino acids are indicated as one letter code. The amino acids Serine was split into two groups (S and Z) as Serine coded for by two groups of codons that are separated bymore than one mutation. Solid line indicates the endogenous codon usage, dashed line indicates the exogenous codon usage, dotted line indicates the combined codon usage.

Chapter 4

Phylogenetic model of stabilizing selection is more informative about site specific selection than extrapolation from laboratory estimates

This chapter is a lightly revised version of a paper to be submitted to Genome Biology and Evolution and co-authored with Michael A. Gilchrist and Brian C. O'Meara.

C. Landerer, B.C. O'Meara, M.A. Gilchrist, Decomposing mutation and selection to identify mismatched codon usage

Abstract

Here we examine the adequacy of experimentally inferred site specific selection for amino acids to inform phylogenetic inferences of sequence evolution. Previous work has shown that laboratory estimates of selection can improve model fit but did not assess their adequacy. We assess the adequacy of experimentally inferred site specific selection using DMS to inform phylogenetic models. We use the β-lactamse TEM for which empirical estimates of site specific selection on amino acids are readily available. We compare our results to *SelAC*, a new phylogenetic model of stabilizing selection. Using simulations to assess model adequacy, we find that experimentally inferred selection does not adequately reflect evolution in the wild. In contrast, *SelAC* improves model fit over models informed by experimentally inferred selection and provides higher model adequacy. We demonstrate the capability of *SelAC* by estimating site specific genetic load of the observed TEM variants.

4.1 Introduction

Numerous attempts to incorporat selection into phylogenetic models have been made. Early models focused on the influence of selection on the substitution rate and fixation probability between a resident and a mutant introduced into a population[32, 61, 81]. These models however, lack site specific equilibrium frequencies. The importance of site specific equilibrium frequencies has long been noted [20, 31]. Halpern and Bruno [38] first introduced a model to incorporate site specific equilibrium frequencies of amino acids. However, they had to concede that their model was too parameter rich and therefore intractable for biological data sets without additional simplifying assumptions. More recent models incorporating site specific equilibrium frequencies still require a large number of parameter to be estimated from the sequence data [52, 54, 87, 41, 89, 80]. Other approaches treat site specific selection as a random effect [69, 67, 68]. A full parameterization of site specific equilibrium frequencies for amino acids requires $19 \times N$ parameters where N is the length of the sequence in amino aicds. It is therefore an attractive option to utilize laboratory experiments to empirically estimate site specific strength of selection on amino acids and infer their equilibrium frequencies [7, 82, 8].

Incorporating empirical estimates of site specific strength of selection on amino acids has important advantages. Individual amino acid site along the protein show differences in evolutionary rates, and strong preferences for amino acids [38, 3, 18]. The usage of site specific selection acknowledges the heterogeneity in selection and amino acid preferences along the protein sequence [40]. Empirical estimates of site specific selection reduce the number of parameters that have to be estimated from the data, making it applicable for smaller data sets and allowing for the fitting of more complex models. There are, however, also shortcomings. Deep mutation scanning (DMS) has recently been recently used to generate comprehensive site specific estimates of the strength of selection on amino acids [26]. The ability to estimate site specific strength of selection on amino acids allows to estimte site specific amino acid preferences and the genetic load a mutation introduces at a particular site[7, 22, 79]. The quality of empirical estimates from DMS, however, depends on many factors including the initial library of mutants and the applied selection [23]. Mutation libraries have to be extensive and therefore produce a heterogeneous population of competing organsims usually not found in nature. In addition, estimates of selection can only be obtained for fast growing organisms that can be manipulated under laboratory conditions. This is a severe limitation of experimentally informed models as many organism can not be cultivated under laboratory conditions or have long generation times.

Even in the cases where empirical estimates of site specific selection on amino acids can be obtained, their applicability for phylogenetic reconstruction is quastionable. In this study, we assess the adequacy of experimentally inferred site specific selection using DMS to inform phylogenetic models and offer an alternative approach to deterimine site specific selection on amino aicds. We use site specific estimates of selection on amino acids for the β-lactamase TEM from Stiffler et al. [79]. We fitted 227 nucleotide and codon models using IQTree and compared their model fits to site specific models of stabilizing selection with (*phydms*, *SelAC*+DMS) and without (*SelAC*) experimentally determined site specific selection coefficients on amino acids [62, 40, 5]. We find that experimentally inferred selection, while improving model fit, does not adequately reflect observed wild type sequences. In contrast, *SelAC* [5] a mechanistic phylogenetic model of stabilizing selection rooted in first principles with site specific equilibrium frequencies improves model fit, and better reflects evolution in the wild. Because *SelAC* assumes stabilizing selection and that the distance of two amino acids in physicochemical space affects substitution probabilities it is able to infer the optimal amino acid at a site and reduce the number of site specific parameters from $19 \times L$ to L.

4.2 Results

4.2.1 Site Specific Stabilizing Selection on Amino Acids Improves Model Fit

We compared *phydms* [40] and *SelAC* [5], models of site specific stabilizing selection on amino acids, to 227 other codon and nucleotide models. We fitted all models to 49 observed sequences of the β-lactamase TEM. The *phydms* and *SelAC* models with site specific selection on amino acids improved model fits by 366 and 934 AICc units, respectively, over codon or nucleotide models without site specific selection (Table 4.1). In addition, *SelAC* outperformed the experimentally informed model *phydms* by 562 to 568 AICc units, depending whether site specific selection on amino acids was inferred by *SelAC* or experimentally informed.

SelAC utilizes a hierarchical model and estimates 263 site specific parameters, $\sim 5\%$ of the $19 \times N = 4997$ parameters necessary to fully describe the site specific selection on amino acids. In contrast, *phydms* does not infer any site specific parameters, but utilizes site specific selection on amino acids estimated from deep mutation scanning experiments. We fixed the optimal amino acid at each site to the experimentally determined one in *SelAC* and refitted the model to the 49 TEM sequences (*SelAC*+DMS). Incorporating site specific selection on amino acids estimated from deep mutation scanning experiments into *SelAC* (*SelAC*+DMS) yields a similar AICc value to *SelAC* without that information. We incorporated the experimentally inferred site specific amino acids by estimating the

However, *SelAC*+DMS is favored by AICc. This is solely due to a decrease in the number of parameters estimated, as the model log-likelihood ($\log(\mathcal{L})$) worsens from -1498 to -1768 (Table 4.1). 263 of the 374 parameters estimated are the discrete optimal amino acid state at each site. However, it is unclear if discrete parameters contribute to the Kullback-Leibler divergence like continous parameters do and have to be penelaized like such. Therefore, the number of parameter for *SelAC* is reported conservatively as the number of unique

Model	log($\mathcal{L}$)	n	AIC	ΔAIC	AICc	ΔAICc
SelAC+DMS	-1768	111	3758	14	3760	0
SelAC	-1498	374	3744	0	3766	6
phydms	-2061	102	4326	582	4328	568
SYM+R2	-2230	102	4663	919	4694	934
GY+F1X4+R2	-2243	102	4690	946	4821	1061

Table 4.1: Model selection, shown are the three models of stabilizing site specific amino acid selection (*SelAC*, *SelAC*+DMS, *phydms*) and the best performing codon and nucleotide model. Reported are the log-likelihood (log($\mathcal{L}$)), the number of parameters estiamted n including edge length, AIC, ΔAIC, AICc, and ΔAICc values. See Table 4.1 for all models tested.

site patterns in the TEM alignment is only 27 which would yield a total number of 138 parameters (96 edge length, 15 mutation/selection parameters, and 27 optimal amion acids). This however would likely be an under estimate of the number of parameteres estimated and the true number of parameters remains unclear at this point due to the inherent non-independence of the underlying data and the descrete nature of the optimized parameters.

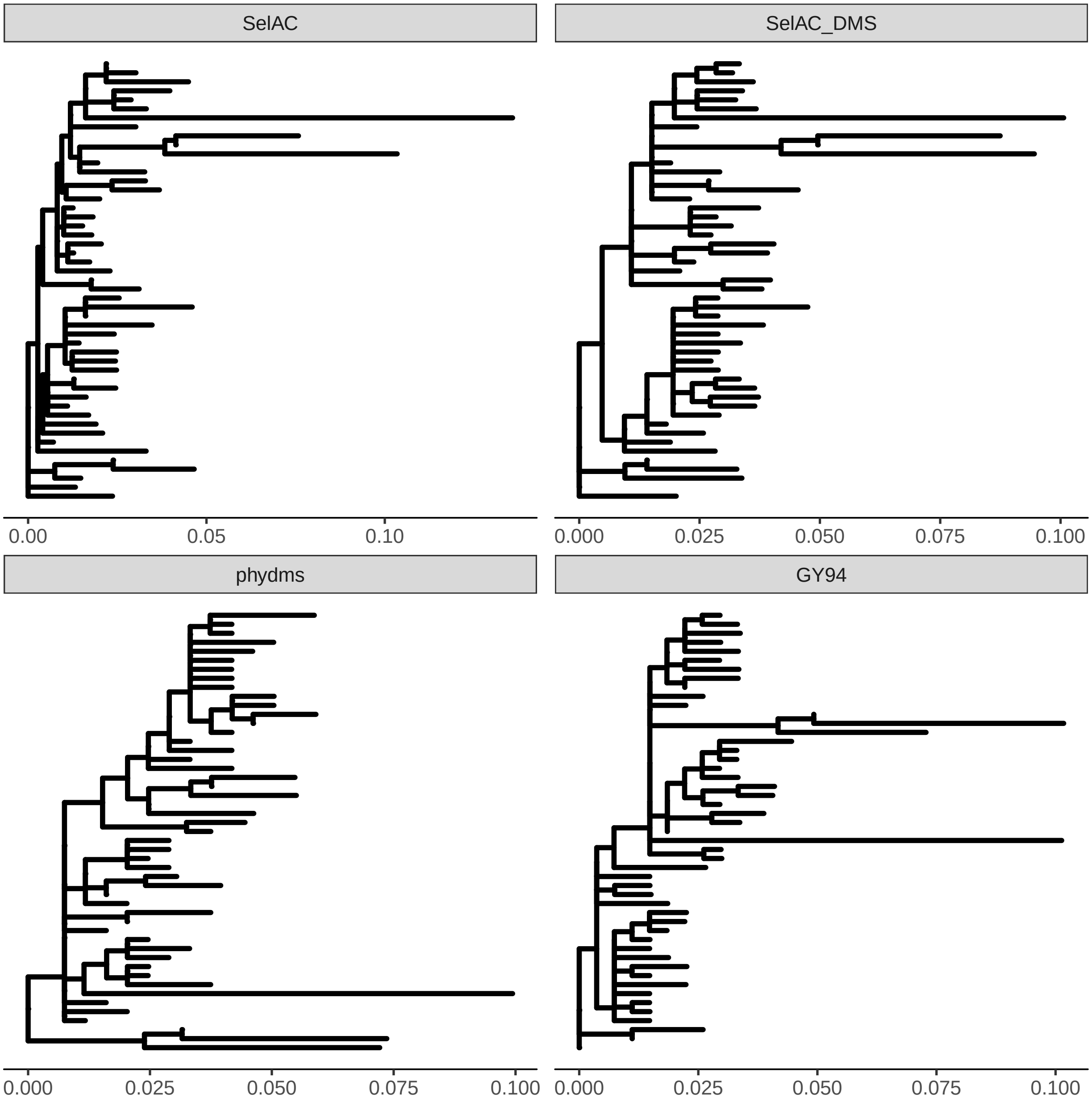
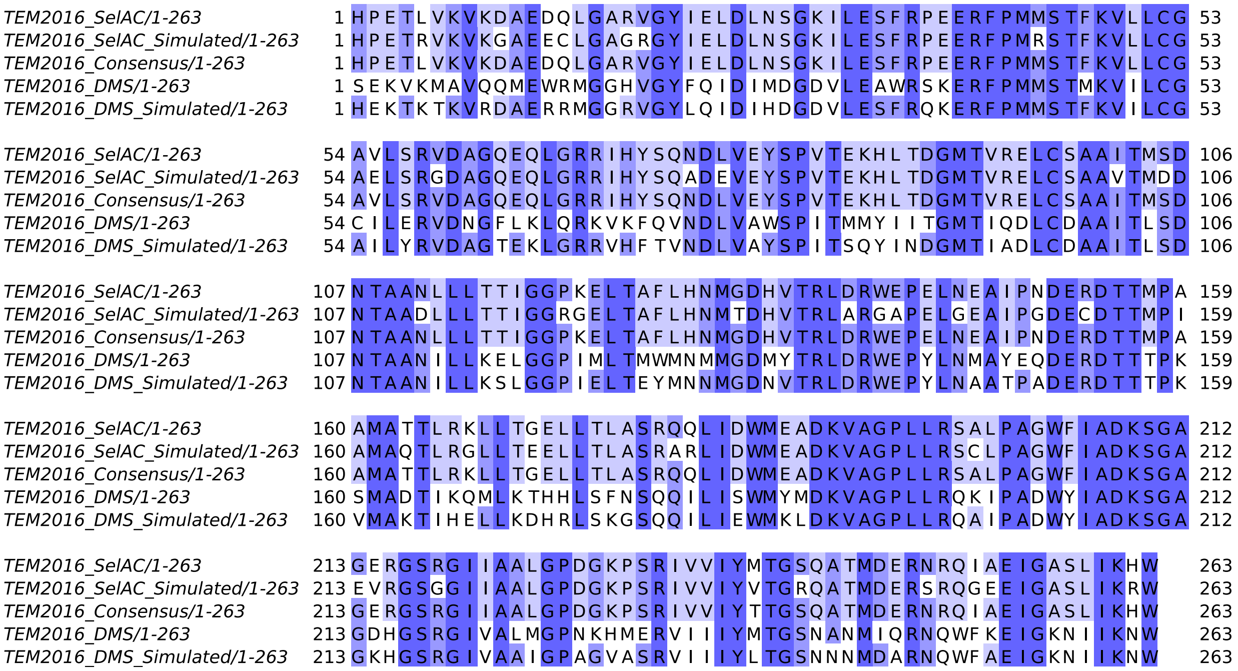


Figure 4.1: Phylogenies resulting from *SelAC*, *SelAC*+DMS, *phydms*, and *GY94*. As *SelAC* is currently to slow for the inference of tomopogies, the topology for the *SelAC* phylogenies was inferred using the codon model of Kosiol et al. [49].

We observe differences in the topology between model fits. The *SelAC* model is currently too slow to estimate the topology, therefore the topology was estimated using the codon model of Kosiol et al. [49]. At this point, it is therefore unclear if the difference in topology can be attributes to the experimentally inferred selection on amino acids. We find that the best codon model (*GY94*) [32] is outperformed by several nucleotide model e.g. *SYM* [91]. This could be an indication that negative frequency dependent selection like it is modeled in *GY94* is not appropriate for TEM [32, 5]. Figure 4.1 shows that the estimated phylogenetic trees shift from long terminal branches (*SelAC*) to longer internal branches (*phydms*). While the *SelAC* model fit shows 84% of all evolution happening at the tips, this reduces to 79% in the *SelAC*+DMS model fit, and 77% in the *phydms* and *GY94* model fits. All models produce polytomies but their location differs between models. The largest polytomies appear in the experimentally informed phylogenies of *phydms*. The position of the sequences with the longest branches differ between *SelAC* and *phydms*.

4.2.2 Laboratory Inferences Inconsistent with Observed Sequences.

The improved model fits with phydms relative to classical nucleotide and codon models are, however, deceiving. The site specific selection inferred by DMS is inconsistent with the observed TEM sequences. We find that the sequence of selectively favored amino acids has only 52% sequence similarity with the observed consensus sequence (Figure 4.2). In addition, assuimg the site specific selection estimated by DMS, the observed TEM sequences represent an unexpectedly high genetic load.



```
TEM2016_SelAC/1-263              1 HPETLVKVKDAEDQLGARVGYIELDLNSGKILESFRPEERFPMMSTFKVLLCG  53
TEM2016_SelAC_Simulated/1-263    1 HPETRVKVKGAEECLGAGRGYIELDLNSGKILESFRPEERFPMRSTFKVLLCG  53
TEM2016_Consensus/1-263          1 HPETLVKVKDAEDQLGARVGYIELDLNSGKILESFRPEERFPMMSTFKVLLCG  53
TEM2016_DMS/1-263                1 SEKVKMAVQQMEWRMGGHVGYFQIDIMDGDVLEAWRSKERFPMMSTMKVILCG  53
TEM2016_DMS_Simulated/1-263      1 HEKTKTKVRDAERRMGGRVGYLQIDIHDGDVLESFRQKERFPMMSTFKVILCG  53

TEM2016_SelAC/1-263             54 AVLSRVDAGQEQLGRRIHYSQNDLVEYSPVTEKHLTDGMTVRELCSAAITMSD  106
TEM2016_SelAC_Simulated/1-263   54 AELSRGDAGQEQLGRRIHYSQADEVEYSPVTEKHLTDGMTVRELCSAAVTMDD  106
TEM2016_Consensus/1-263         54 AVLSRVDAGQEQLGRRIHYSQNDLVEYSPVTEKHLTDGMTVRELCSAAITMSD  106
TEM2016_DMS/1-263               54 CILERVDNGFLKLQRKVKFQVNDLVAWSPITMMYIITGMTIQDLCDAAITLSD  106
TEM2016_DMS_Simulated/1-263     54 AILYRVDAGTEKLGRRVHFTVNDLVAYSPITSQYINDGMTIADLCDAAITLSD  106

TEM2016_SelAC/1-263            107 NTAANLLLTTIGGPKELTAFLHNMGDHVTRLDRWEPELNEAIPNDERDTTMPA  159
TEM2016_SelAC_Simulated/1-263  107 NTAADLLLTTIGGRGELTAFLHNMTDHVTRLARGAPELGEAIPGDECDTTMPI  159
TEM2016_Consensus/1-263        107 NTAANLLLTTIGGPKELTAFLHNMGDHVTRLDRWEPELNEAIPNDERDTTMPA  159
TEM2016_DMS/1-263              107 NTAANILLKELGGPIMLTMWMNMMGDMYTRLDRWEPYLNMAYEQDERDTTTPK  159
TEM2016_DMS_Simulated/1-263    107 NTAANILLKSLGGPIELTEYMNNMGDNVTRLDRWEPYLNAATPADERDTTTPK  159

TEM2016_SelAC/1-263            160 AMATTLRKLLTGELLTLASRQQLIDWMEADKVAGPLLRSALPAGWFIADKSGA  212
TEM2016_SelAC_Simulated/1-263  160 AMAQTLRGLLTEELLTLASRARLIDWMEADKVAGPLLRSCLPAGWFIADKSGA  212
TEM2016_Consensus/1-263        160 AMATTLRKLLTGELLTLASRQQLIDWMEADKVAGPLLRSALPAGWFIADKSGA  212
TEM2016_DMS/1-263              160 SMADTIKQMLKTHHLSFNSQQILISWMYMDKVAGPLLRQKIPADWYIADKSGA  212
TEM2016_DMS_Simulated/1-263    160 VMAKTIHELLKDHRLSKGSQQILIEWMKLDKVAGPLLRQAIPADWYIADKSGA  212

TEM2016_SelAC/1-263            213 GERGSRGIIAALGPDGKPSRIVVIYMTGSQATMDERNRQIAEIGASLIKHW    263
TEM2016_SelAC_Simulated/1-263  213 EVRGSGGIIAALGPDGKPSRIVVIYVTGRQATMDERSRQGEEIGASLIKRW    263
TEM2016_Consensus/1-263        213 GERGSRGIIAALGPDGKPSRIVVIYTTGSQATMDERNRQIAEIGASLIKHW    263
TEM2016_DMS/1-263              213 GDHGSRGIVALMGPNKHMERVIIIYMTGSNANMIQRNQWFKEIGKNIIKNW    263
TEM2016_DMS_Simulated/1-263    213 GKHGSRGIVAAIGPAGVASRVIIIYLTGSNNNMDARNQWFAEIGKNIIKNW    263
```

Figure 4.2: Alignment of TEM optimal and simulated sequences. Indicated is the percentage identity at each site.

Simulations of codon sequences under the experimentally inferred site specific selection for amino acids reveals that we would not expect to see the observed TEM sequences. We simulated under a wide range of effective population sizes N_e, and find that the experimentally inferred site specific selection is very strong. With more realistic values of $N_e = 10^7$, we find that the simulated sequences are 62% similar to the observed consensus sequence (Figure 4.3a). This is a higher similarity than the observed consensus sequence shows with the sequence of selectively favored amino acids estimated using deep mutation scanning. Only when N_e is reduced to one individual does drift overpower selection (Figure 4.3b). The genetic load of the simulated sequences decrease slowly with increasing N_e (Figure 4.3b). After simulating until the sequences reach one expected mutation per site and $N_e = 10^7$ the simulated sequences show a genetic load of 0.25, which is in contrast to the ~ 8 times higher than the estimated observed load of 2.1. Thus it appears unlikely that the observed sequences have evolved under the DMS inferred site specific selection values.

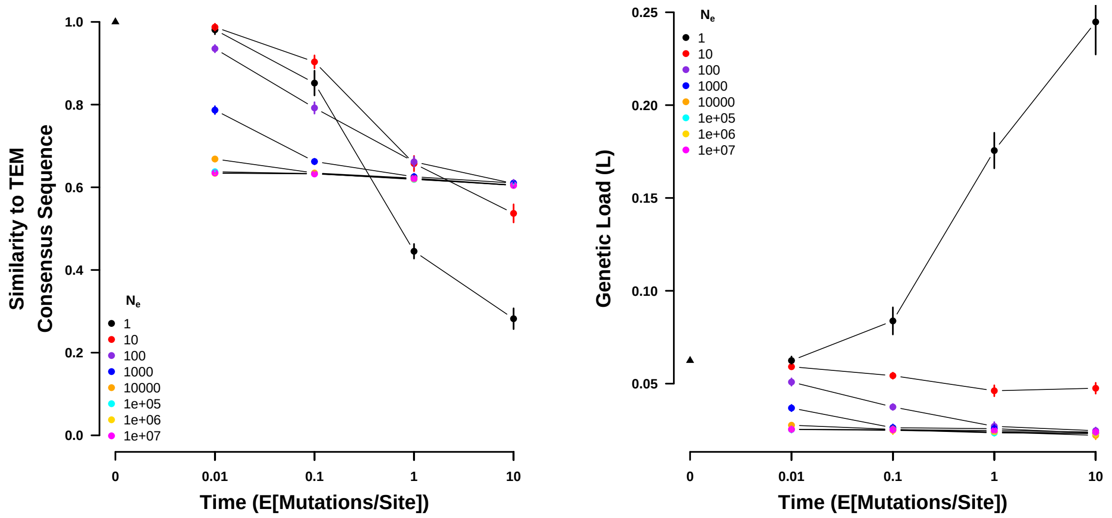


Figure 4.3: Sequences simulated from the ancestral state under the site specific selection on amino acids estimated using deep mutation scanning. (left) Sequence similarity to the observed consensus sequence at various times for a range on values of N_e. (right) Genetic load of the simulated sequences at various times for a range on values of N_e. Time is given in number of expected mutations per site, which equals the substitution rate of a neutral mutation. Points indicate sample means and vertical bars indicate standard deviations. Initial sequence is the inferred ancestral state of the TEM variants and indicated by a black triangle.

4.2.3 Stabilizing Selection for Optimal Physicochemical Properties Improves Model Adequacy

We assessed model adequacy of *SelAC* and find that it better explains the observed TEM sequences than *phydms*. The observed consensus sequence has 99% sequence similarity with the sequence of selectively favored amino acids estimated by *SelAC*. Furthermore, assuming the site specific selection estimated by *SelAC*, the observed sequences represent a very small genetic load on the order of 10^{-6}(Table 4.2, Figure 4.5).

We simulated codon sequences forward in time for various length of time, using the *SelAC* inferred site specific selection for amino acids to assess sequence similarity. We simulated the evolution of TEM from the inferred ancestral state using a wide range of effective population sizes N_e (Figure 4.4a). The ancestral state was estimated to be the observed consensus sequence. As expected, for small N_e, simulated sequences drift away from the observed consensus sequence. Because of the high similarity between the optimal amino acid sequence estimated by *SelAC* and the observed consensus sequence, the genetic load increases drastically as a result. Increasing N_e to 10^7 the simulated sequences reach a sequence similarity of 83%, this is in contrast to the observed avergage sequence similarity of 98%.

We estimated the total genetic load the sequences represent using the *SelAC* inferred selection on amino acids. The total genetic load of the simulated sequences avergages 9.8×10^{-6} (Figure 4.4b). The total estimated genetic load of the observed sequences averages 4.2×10^{-5}. Thus, the simulated sequences show a lower genetic load despite the greater divergence from the observed consensus sequence.

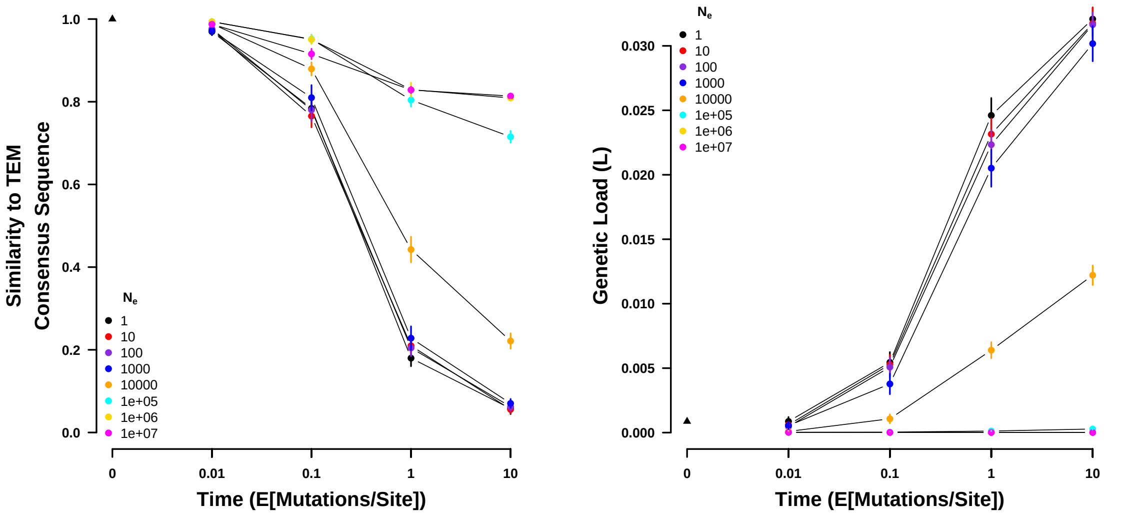


Figure 4.4: Sequences simulated from the ancestral state under the site specific selection on amino acids estimated using *SelAC*. (left) Sequence similarity to the observed consensus sequence at various times for a range on values of N_e. (right) Genetic load of the simulated sequences at various times for a range on values of N_e. Time is given in number of expected mutations per site, which equals the substitution rate of a neutral mutation. Points indicate sample means and vertical bars indicate standard deviations. Initial sequence is the inferred ancestral state of the TEM variants and indicated by a black triangle.

To further demonstrate the consistency of *SelAC*, we simulated codon sequences over the same period of time using 10 uniform samples codon sequences with 263 sites, the same length as the observed TEM variants. We find that the sequence similarity increases with effective population size N_e. The random sequences start of with a similarity of $\sim 6\%$ which increases with N_e to $\sim 28\%$ (Figure 4.8a). The same initial sequences under the site specific selection inferred by the deep mutation scanning experiment increase only to $\sim 18\%$ in sequence similarity.

4.2.4 Estimating Site Specific Selection on Amino Acids

SelAC allows for the estimation of site specific selection on amino acids and the genetic load of an observed amino acid relative to the inferred optimal amino acid. Figure 4.5 and Figure 4.6 illustrate how the genetic load varies along the TEM sequence. The region between residue 80 to 120 where three consecutive helices are located consist only of selectively favored amino acids and does not show any genetic load. The highest genetic load is found in the unstructured regions and the lowest genetic load is found in β-sheets. Despite the differences, none show statistical significance. The largest increase in genetic load is located at the beginning of the last helix. This stronly contributes to the estimate of similar genetic loads for helices and unstructured regions in the observed TEM sequences (Table 4.2). However, exclution of this site as outlier does not produce a significant difference in genetic load between unstructured and helix regions.

The highest efficacy of selection G and the lowest genetic load among the TEM secondary structure features is estimated in the β-sheet regions. Residues forming the active and substrate binding site appear to be under the strongest selection, with no accumulated genetic load. However, we find in one sequence (*Acinetobacter baumannii*, TEM-193) a Lysine, a proton donor, at the proton acceptor site 143 driving the reduced efficacy of selection G. This is in concordance with the experimental estimates, where proton acceptors

			G		Genetic Load	
Protein	Secondary Structure	# Residues	Mean	SE	Mean	SE
TEM		263	219.3	7.5	15.9×10^{-8}	6.5×10^{-8}
	Helix	113	206.1	12.4	17.5×10^{-8}	13.1×10^{-8}
	β-Sheet	48	238.6	15.8	6.8×10^{-8}	2.9×10^{-8}
	Unstructured	102	224.8	11.4	18.6×10^{-8}	8.1×10^{-8}
	Active Sites	5	202.6	62.2	0.01×10^{-8}	0.01×10^{-8}
SHV		263	244.9	6.8	4.0×10^{-8}	1.9×10^{-8}
	Helix	102	234.6	11.5	7.3×10^{-8}	4.8×10^{-8}
	β-Sheet	66	253.1	12.8	2.1×10^{-8}	1.1×10^{-8}
	Unstructured	95	224.7	11.4	1.5×10^{-8}	0.6×10^{-8}
	Active Sites	5	239.9	60.0	1.5×10^{-8}	1.5×10^{-8}

Table 4.2: Efficacy of selection (G) and Genetic Load for TEM and SHV, and separated by secondary structure. G was estimated as a truncated variable with an upper bound of 300.

are selectively favored. Again, differences between secondary structure elements are not statistically significant.

It was previously proposed that experimentally inferred site specific selection for amino acids can be used to extrapolate the fitness landscape of related proteins [7, 8]. We therefore compared the genetic load, the *SelAC* selection parameters of our *SelAC* TEM model fit to a *SelAC* model fit of SHV, and site specific efficacy of selection (G). The genetic load in SHV appears to be lower than in TEM with the exception of residues found in β-sheets and the active site (Table 4.2). This is consistent with the elevated efficacy of selection G in SHV. However, only differences in genetic load in the unstructured regions are significantly different between the TEM and SHV sequences, but only at the $\alpha = 0.05$ significant level ($p = 0.04$). While the average genetic load across secondary structures is not significantly different, the sites causing increases genetic load differ between SHV and TEM (Figure 4.7). In contrast to TEM, we find the highest genetic load in SHV secondary structure features in the helices (Table 4.2). We find the highest genetic load in SHV at the end of the first helix. However, we do find a peak of similar magnitude in the TEM sequence at the end of the first helix, but this peak is overshadowed by increased genetic load at the beginning of the last helix.

We find that site specific efficacy of selection G differs greatly between SHV and TEM ($\rho = 0.12$), despite a similar estimate of α_G describing the distribution of G values (Figure 4.10a). We generally find increased G values in SHV (Table 4.2). However, non of these increases are statistically significant. Most *SelAC* selection parameters are very similar between the TEM and the SHV model fit. An exception is the weight for the physicochemical composition property α_c (Figure 4.10b). Furthermore, we find that the sequences of selectively favored amino acids estimated by *SelAC* for TEM and SHV only show 68% sequence similarity.

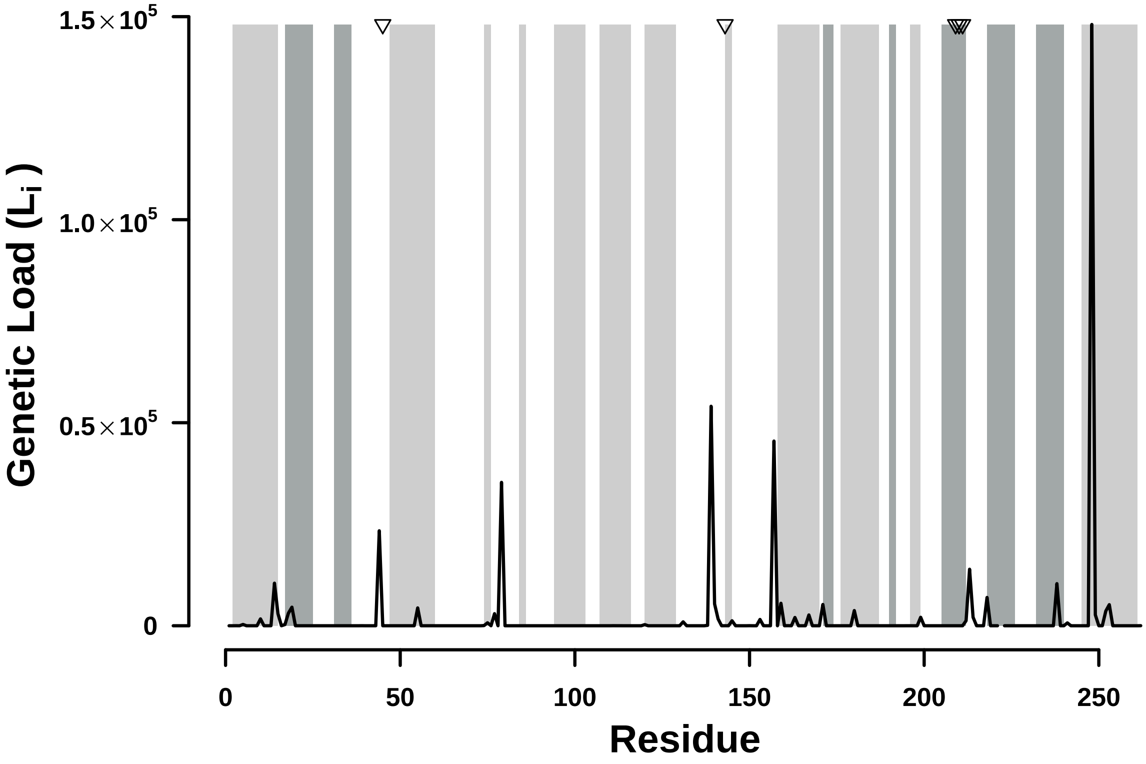


Figure 4.5: Distribution of genetic load in TEM. Average genetic load over all observed TEM variants is indicated by the black line. Light gray bars indicate where helices are found, and dark gray bars indicate β-sheets. The three residues forming the active sites are indicated by three triangles at the top of the plot.

4.3 Discussion

Here we revisited how well experimental selection estimates from laboratory experiments, specifically deep mutation scanning, explain sequence evolution and compared it to $SelAC$, a novel phylogenetic framework. Previous work has shown that laboratory estimates of selection can improve model fit over classical approaches like GY94 [7, 8]. While our study confirms this notion, we identify important shortcomings of these laboratory estimates. In contrast, *SelAC* is a more general phylogenetic model of stabilizing selection that does not require costly laboratory estimates of selection and is nevertheless favored by model selection (Table 4.1). *SelAC* does not rely on artificially induced selection in the laboratory but is a mechanistic framework rooted in first principles. It estimates site specific selection on amino acids from the sequence data based on distances between amino acids in physicochemical space [35, 5]. This allows *SelAC* to be applied to any set of protein coding sequences, eliminating the need to extrapolate from one homologous gene family to the next (e.g. from TEM to SHV).

While previous work showed the advantages of experimentally informed phylogenetics, they did not assess how adequate the estimated selection reflects observed wild-type sequences. The low sequence similarity between the observed consensus sequence and the sequence of selectively favored amino acids estimated from deep mutation scanning experiments is evidence for that. This begs the question how well the experimental selection coefficients represent evolution of sequences in nature. Deep mutation scanning experiments are performed using a comprehensive library of mutants and a strong artificial selection pressure [23, 45, 25, 26]. This results in a very large selection coefficient s and a heterogeneous population of competing individuals unlikey to occur in nature.

The selection pressure imposed during the DMS experiment was limited to ampicillin and focused solely on TEM-1 [79]. However, TEM variants can also confer resistance to a wide range of other antibiotics, including penicillins, cephalosporins, cefotaxime, ceftazidime, or aztreonam [77, 78, 33, 56, 11, 10]. Thus, the inferred selection is biased towards ampicillin and is inconsistent with the observed TEM sequences (Figure 4.3). This may very well be very appropriate to explore the selection on TEM in a hospital environment but is unlikely to be representative of the selection faced by *E. coli* in nature.

If we assume that the DMS selection coefficients underly the evolution of the observed TEM sequences we are left with two possible explanations for the observed sequences. First, the sequences are unable to reach a fitness peak, potentially due to a low selection pressure, or not enough time. Second, the observed TEM sequences are highly maladapted. Both options seem unlikely. *E. coli* has a large effective population size N_e, estimates are on the order of 10^8 to 10^9 [64, 39]. As new mutations are introduced into a population at a rate proportional to N_e, *E. coli* can effectively explore the sequence space. We, therefore, expect the observed sequence variants to be near mutation-selection-drift equilibrium. This expectation is supported by our simulations in which we observe a higher sequence similarity with the observed TEM consensus sequence and decreased genetic load even with much smaller N_e (Figure 4.3). Furthermore, previous work showed that the catalytic reaction performed by TEM of penicillin-class antibiotics is close the diffusion limit, making TEM a so-called perfect enzyme [57]. The very large effective population size, however, also raises a concern that the population mutation rate of *E. coli* $\Theta = 4N_e\mu$ exceeds 0.1 and violated *SelAC*'s weak mutation assumption [14]. If the weak mutation assumption is violated evolution is no longer mutation limited and the time between fixation events increases.

As experimental selection estimates are not readily available for most organsims and proteins, one solution is to extrapolate the estimates to homologous gene families [7, 8]. When extrapolating the selection estimates from the β-lactamase family TEM to the SHV family, the sequence similarity between the observed consensus sequence and the sequence of selectively favored amino acids estimated from deep mutation scanning experiments drops slightly from 52% to 49%. In contrast, the site specific efficacy of selection (G) revealed large differences in the site specific selection on amino acids between TEM and SHV. The mismatched in physicochemical weights also indicates differences in selection constraints. While the polarity of amino acids is of similar importance in TEM and SHV, amino acid composition appears to play a much greater role in SHV than in TEM. In contrast to the experimental selection estimates, extrapolated from TEM to SHV, the *SelAC* selection estimates are consistent with the observed sequences, e.g. the selectively favored amino acids estimated by *SelAC* shows a high sequence similarity with the observed TEM and SHV consensus sequence (99%).

While *SelAC* better explains the observed TEM sequences than the experimental estimates of site specific selection on amino acids, it is not without shortcomings itself. *SelAC* is currently to slow to be used in topology searches, therefore it is unclear if the differences in topology between *phydms* and *SelAC* can be attributed to the same inadequacies of experimentally inferred selection. As the simulation of TEM evolution from he ancestral state under the *SelAC* inferred site specific selection revealed, the formulation of *SelAC* can and should be improved upon. Starting from the ancestral sequence, the simulated sequences show initial divergence despite stabilizing selection for the optimal amino acid. While *SelAC* allows for site heterogeneity in selection for amino acids, it still ignores epistatis. This however, is a shortcoming shared with experimental estimates by deep mutation scanning, as each mutant typically only carries one mutation [23, 45]. *SelAC* is a model stabilizing selection, however, not every protein is under stabilizing selection. TEM plays a role in chemical warfare with conspecifics and other microbes, therefore some sites may be under negative frequency dependent selection. This potential heterogeneity in selection highlights another shortcoming of *SelAC*. *SelAC* assumes the same distribution for the efficacy of selection (G) and physicochemical sensitivities across the whole protein. However, it is easy to imagine that sites in different secondary structures or at active sites do not share a common distribution.

As *SelAC* assumes that the fitness of an amino acid at a site declines with its distance in physicochemical space to the optimal amino acid, the choice of physicochemical properties becomes important. In this study, we assumed the physicochemical properties estimated by Grantham [35] for all sites. However, a wide range of additional physicochemical properties of amino acids have been described [47]. A more optimal choice of physicochemical properties may be possible as well as the a relaxation of the assumptions that the same properties apply to all sites equally. Future work will attempt to address these shortcomings, however, *SelAC*'s hierarchical model structure and the open-source code base allow researchers to easily address these shortcoming if desired.

In conclusion, experimental estimates of site specific selection on amino acids have to be treated with skepticism and their adequacy should be assessed before using them to inform phylogenetic inferences. This study was initiated to assess the quality of *SelAC* with the expectation that *SelAC* could be a faster, cheaper, and more readily available alternative to experimentally inferred selection; specifically in organisms where these experiments are not feasible. Intuitively one would expect that selection coefficients estimated of mutations in living organisms would provide more information on the evolution of proteins than a model relying on many simplifying assumptions. As we show in this study, not only can *SelAC* estimate site specific selection on amino acids but our approach is a more adequat descripton of selection on amino acids in nature than experimental estimates.

4.4 Materials and Methods

4.4.1 Phylogenetic Inference and Model selection

TEM and SHV sequences were obtained from Bloom [8] already aligned. We however, separated the TEM and SHV sequences into individual alignments. Experimentally fitness values for TEM were taken from Stiffler et al. [79]. We followed [8] to convert the experimental fitness values into site specific equilibrium frequencies for *phydms*. *phydms* (version 2.5.1) was fitted using site specific selection on amino acids estimated from deep mutation scanning experiments from Stiffler et al. [79] and python (version 3.6).

SelAC (version 1.6.1) was fitted to the TEM alignment using R (version 3.4.1) [66] with and without site specific selection on amino acids estimated from deep mutation scanning experiments. We assumed the physicochemical properties estimated by Grantham [35]. We chose the contraint free general unrestricted model [90] as mutation model . All other models were fitted using IQTree [62].

We report each model's $\log(\mathcal{L})$, AIC, and AICc. Models were selected based on the AICc values.

4.4.2 Sequence Simulation

Sequences were simulated by stochastic simulations using a Gillespie algorithm [30] that was model independent. The simulation followed Sella and Hirsh [70] to calculate fixation probabilities. The fitness values were estimated using *SelAC* or experimentally inferred. We chose the fitness values of the highest concentration (2500 $\mu g/mL$) treatment of ampicillin for our comparison. We modified the experimental fitness such that the amino acid with the highest fitness at each site has a value of one. Mutation rates were taken from the *SelAC* or *SelAC*+DMS fit. The initial sequences were either a random sample of 263 codons or the ancestral sequence reconstructed using FastML [4] (last accessed: 30.09.2018). Each sequence was simulated 10 times and we report average genetic load and sequence similarity and the corresponding standard error. The sequences were sampled at times 0.01, 0.1, 1, and 10 expected mutations per site.

4.4.3 Estimating site specific efficacy of selection G

SelAC does not by default estimate site specific values for G but assusmes G values follow a gamma distribution [21]. Site specific values for G were optimized by fixing all estimated parameters and perfoming a maximum likelihood search without the usual integration over G. In contrast to *SelAC* that assumes G to be purley positive, we allowed negative values for G and constraint the search to values between -300 and 300.

4.4.4 Estimating site specific fitness values w_i

Following Beaulieu et al. [5] w_i is proportional to

$$w_i \propto \exp(-A_0 \eta \psi) \tag{4.1}$$

where A_0 describes the decline in fitness with each high energy phosphate bond wasted per unit time, and ψ is the protein's production rate. η is the cost/benefit ratio of a protein (see [5] for details). However, *SelAC* only estimates a composition parameter $\psi' = A_0 \psi N_e$. N_e describes the effective population size. *SelAC* assumes $N_e = 5 \times 10^6$. *SelAC* assumes $A_0 = 4 \times 10{-}7$ [28]. Thus,

$$\psi = \frac{\psi'}{A_0 N_e q} \tag{4.2}$$

4.4.5 Model Adequacy

Model adequacy was assessed by comparing the observed sequences and simulations under the site specific selection inferred by the deep mutation scanning experiment or *SelAC*. First, similarity between the sequence of selectively favored amino acids and the observed TEM sequences was assessed. Sequence similarity was measured as the number of differences in the amino acid sequence. Second, the genetic load of the observed and the simulated sequences was calculated using either the site specific selection inferred by the deep mutation scanning experiment or *SelAC*.

Genetic load was calculated as

$$L_i = \frac{w_{max} - w_i}{w_{max}} \tag{4.3}$$

where w_{max} is the fitness of the sequence of selectively favored amino acids estimated using the site specific selection inferred by the deep mutation scanning experiment or *SelAC*. w_i represents the fitness of the ith residue. This the genetic load L of a sequence is given by $\sum_{i=1}^{n} L_i$ where n is the number of amino acids.

4.5 Acknowledgments

This work was supported in part by NSF Award and DEB-1355033 (BCO, MAG, and RZ) with additional support from The University of Tennessee Knoxville. CL received support as a Graduate Student Fellow at the National Institute for Mathematical and Biological Synthesis, an Institute sponsored by the National Science Foundation through NSF Award DBI-1300426, with additional support from UTK. The authors would like to thank Russel Zaretzki, Jeremy Beaulieu and Alexander Cope for their helpful criticisms and suggestions for this work.

4.6 Supplementary Material

No.	Model	LnL	n	AIC	ΔAIC	AICc	ΔAICc
1	*SelAC*+DMS +G4	-1768	111	3758	14	3760	0
2	*SelAC* +G4	-1498	374	3744	0	3766	6
3	*phydms*	-2060.85	102	4326	582	4328	568
4	SYM+R2	-2229.616	102	4663.232	919.232	4693.862	933.862
5	TIMe+R2	-2232.406	100	4664.811	920.811	4694.172	934.172
6	TVMe+R2	-2232.838	101	4667.677	923.677	4697.668	937.668
7	TIM3e+R2	-2234.332	100	4668.664	924.664	4698.024	938.024
8	TIM2e+R2	-2234.381	100	4668.763	924.763	4698.123	938.123
9	K3P+R2	-2235.777	99	4669.553	925.553	4698.291	938.291
10	TNe+R2	-2236.078	99	4670.155	926.155	4698.892	938.892
11	SYM+R3	-2229.616	104	4667.232	923.232	4699.162	939.162
12	TIM+F+R2	-2230.958	103	4667.915	923.915	4699.191	939.191
13	TIMe+R3	-2232.404	102	4668.808	924.808	4699.437	939.437
14	GTR+F+R2	-2228.537	105	4667.073	923.073	4699.665	939.665
15	K3Pu+F+R2	-2232.617	102	4669.234	925.234	4699.864	939.864
16	TVM+F+R2	-2230.105	104	4668.21	924.21	4700.14	940.14
17	TVMe+R3	-2232.838	103	4671.676	927.676	4702.952	942.952
18	K2P+R2	-2239.424	98	4674.847	930.847	4702.969	942.969
19	TIM3e+R3	-2234.332	102	4672.664	928.664	4703.293	943.293
20	TIM2e+R3	-2234.381	102	4672.762	928.762	4703.391	943.391
21	TIM3+F+R2	-2233.064	103	4672.127	928.127	4703.403	943.403
22	TIM2+F+R2	-2233.114	103	4672.227	928.227	4703.503	943.503
23	K3P+R3	-2235.777	101	4673.553	929.553	4703.545	943.545
24	TN+F+R2	-2234.624	102	4673.249	929.249	4703.878	943.878
25	TPM3u+F+R2	-2234.673	102	4673.347	929.347	4703.977	943.977
26	TPM3+F+R2	-2234.674	102	4673.348	929.348	4703.978	943.978
27	TPM2u+F+R2	-2234.681	102	4673.363	929.363	4703.993	943.993
28	TPM2+F+R2	-2234.683	102	4673.365	929.365	4703.995	943.995
29	TNe+R3	-2236.077	101	4674.155	930.155	4704.146	944.146
30	TIM+F+R3	-2230.958	105	4671.915	927.915	4704.507	944.507
31	HKY+F+R2	-2236.266	101	4674.531	930.531	4704.522	944.522
32	GTR+F+R3	-2228.536	107	4671.073	927.073	4705.011	945.011
33	K3Pu+F+R3	-2232.617	104	4673.234	929.234	4705.163	945.163
34	TVM+F+R3	-2230.105	106	4672.21	928.21	4705.471	945.471
35	K2P+R3	-2239.192	100	4678.384	934.384	4707.745	947.745
36	TIM3+F+R3	-2233.063	105	4676.127	932.127	4708.718	948.718
37	TIM2+F+R3	-2233.113	105	4676.227	932.227	4708.818	948.818
38	TN+F+R3	-2234.624	104	4677.249	933.249	4709.178	949.178
39	TPM3u+F+R3	-2234.673	104	4677.347	933.347	4709.277	949.277
40	TPM3+F+R3	-2234.674	104	4677.348	933.348	4709.277	949.277

41	TPM2u+F+R3	-2234.681	104	4677.363	933.363	4709.293	949.293
42	TPM2+F+R3	-2234.682	104	4677.364	933.364	4709.294	949.294
43	HKY+F+R3	-2236.074	103	4678.148	934.148	4709.424	949.424
44	SYM+I+G4	-2243.212	102	4690.424	946.424	4721.054	961.054
45	TVMe+I+G4	-2244.533	101	4691.066	947.066	4721.057	961.057
46	TIMe+I+G4	-2246.457	100	4692.914	948.914	4722.275	962.275
47	K3P+I+G4	-2248.166	99	4694.332	950.332	4723.069	963.069
48	TVM+F+I+G4	-2241.853	104	4691.707	947.707	4723.636	963.636
49	TIM3e+I+G4	-2247.379	100	4694.758	950.758	4724.119	964.119
50	K3Pu+F+I+G4	-2245.156	102	4694.311	950.311	4724.941	964.941
51	GTR+F+I+G4	-2241.484	105	4692.968	948.968	4725.559	965.559
52	TIM+F+I+G4	-2244.418	103	4694.836	950.836	4726.112	966.112
53	TPM3u+F+I+G4	-2246.03	102	4696.06	952.06	4726.69	966.69
54	TPM3+F+I+G4	-2246.069	102	4696.138	952.138	4726.768	966.768
55	TIM2e+I+G4	-2248.934	100	4697.868	953.868	4727.228	967.228
56	TNe+I+G4	-2250.587	99	4699.174	955.174	4727.911	967.911
57	TIM3+F+I+G4	-2245.534	103	4697.068	953.068	4728.344	968.344
58	K2P+I+G4	-2252.181	98	4700.362	956.362	4728.484	968.484
59	TPM2u+F+I+G4	-2247.579	102	4699.158	955.158	4729.788	969.788
60	TPM2+F+I+G4	-2247.685	102	4699.371	955.371	4730	970
61	HKY+F+I+G4	-2249.065	101	4700.13	956.13	4730.121	970.121
62	TIM2+F+I+G4	-2247.009	103	4700.018	956.018	4731.294	971.294
63	TN+F+I+G4	-2248.511	102	4701.023	957.023	4731.652	971.652
64	TVMe+I	-2254.804	100	4709.608	965.608	4738.968	978.968
65	K3P+I	-2257.72	98	4711.439	967.439	4739.561	979.561
66	SYM+I	-2254.11	101	4710.221	966.220	4740.212	980.212
67	TIMe+I	-2257.074	99	4712.149	968.149	4740.886	980.886
68	TVM+F+I	-2252.157	103	4710.315	966.315	4741.591	981.591
69	K3Pu+F+I	-2254.856	101	4711.712	967.712	4741.704	981.704
70	TIM3e+I	-2257.796	99	4713.592	969.592	4742.33	982.33
71	TPM3+F+I	-2255.771	101	4713.543	969.543	4743.534	983.534
72	TPM3u+F+I	-2255.771	101	4713.543	969.543	4743.534	983.534
73	K2P+I	-2261.218	97	4716.436	972.436	4743.949	983.949
74	GTR+F+I	-2252.067	104	4712.133	968.133	4744.063	984.063
75	TIM+F+I	-2254.783	102	4713.566	969.566	4744.195	984.195
76	TNe+I	-2260.579	98	4717.158	973.158	4745.28	985.28
77	TIM3+F+I	-2255.684	102	4715.368	971.368	4745.998	985.998
78	HKY+F+I	-2258.352	100	4716.703	972.703	4746.064	986.064
79	TIM2e+I	-2259.878	99	4717.757	973.757	4746.494	986.494
80	TVMe+G4	-2258.853	100	4717.705	973.705	4747.066	987.066
81	SYM+G4	-2257.573	101	4717.146	973.146	4747.137	987.137
82	TPM2+F+I	-2257.712	101	4717.423	973.423	4747.415	987.415
83	TPM2u+F+I	-2257.712	101	4717.423	973.423	4747.415	987.415
84	K3P+G4	-2261.922	98	4719.844	975.844	4747.966	987.966
85	TIMe+G4	-2260.683	99	4719.365	975.365	4748.103	988.103

86	TN+F+I	-2258.28	101	4718.561	974.561	4748.552	988.552
87	TIM3e+G4	-2261.255	99	4720.51	976.51	4749.247	989.247
88	TVM+F+G4	-2256.108	103	4718.216	974.216	4749.492	989.492
89	TIM2+F+I	-2257.643	102	4719.286	975.286	4749.915	989.915
90	K3Pu+F+G4	-2258.971	101	4719.941	975.941	4749.933	989.933
91	TPM3u+F+G4	-2259.716	101	4721.433	977.433	4751.424	991.424
92	TPM3+F+G4	-2259.717	101	4721.434	977.434	4751.425	991.425
93	GTR+F+G4	-2255.75	104	4719.5	975.5	4751.43	991.43
94	TIM+F+G4	-2258.638	102	4721.276	977.276	4751.906	991.906
95	K2P+G4	-2265.454	97	4724.907	980.907	4752.421	992.421
96	TNe+G4	-2264.219	98	4724.437	980.437	4752.559	992.559
97	TIM3+F+G4	-2259.366	102	4722.732	978.732	4753.361	993.361
98	TIM2e+G4	-2263.57	99	4725.141	981.141	4753.878	993.878
99	JC+R2	-2266.233	97	4726.466	982.466	4753.98	993.98
100	F81+F+R2	-2262.327	100	4724.654	980.654	4754.015	994.015
101	HKY+F+G4	-2262.499	100	4724.999	980.999	4754.359	994.359
102	TPM2+F+G4	-2261.915	101	4725.829	981.829	4755.82	995.82
103	TPM2u+F+G4	-2261.915	101	4725.829	981.829	4755.82	995.82
104	TN+F+G4	-2262.169	101	4726.338	982.338	4756.329	996.329
105	TIM2+F+G4	-2261.585	102	4727.17	983.17	4757.8	997.8
106	F81+F+R3	-2262.028	102	4728.056	984.056	4758.685	998.685
107	JC+R3	-2265.997	99	4729.994	985.994	4758.731	998.731
108	F81+F+I+G4	-2274.845	100	4749.69	1005.69	4779.05	1019.05
109	JC+I+G4	-2279.318	97	4752.636	1008.636	4780.149	1020.149
110	F81+F+I	-2283.56	99	4765.119	1021.119	4793.857	1033.857
111	JC+I	-2287.984	96	4767.968	1023.968	4794.881	1034.881
112	F81+F+G4	-2287.834	99	4773.669	1029.669	4802.406	1042.406
113	JC+G4	-2292.095	96	4776.19	1032.19	4803.103	1043.103
114	*GY94* +F1X4+R2	-2242.963	102	4689.926	945.926	4821.251	1061.251
115	MGK+F1X4+R2	-2243.111	102	4690.221	946.221	4821.546	1061.546
116	*GY94* +F1X4+R3	-2238.022	104	4684.043	940.043	4822.271	1062.271
117	MGK+F3X4+R2	-2229.923	108	4675.846	931.846	4828.729	1068.729
118	*GY94* +F1X4+I+G4	-2247.179	102	4698.359	954.359	4829.684	1069.684
119	MGK+F1X4+I+G4	-2247.292	102	4698.583	954.583	4829.908	1069.908
120	MGK+F1X4+R3	-2241.989	104	4691.978	947.978	4830.206	1070.206
121	MGK+F3X4+R3	-2224.78	110	4669.559	925.559	4830.217	1070.217
122	*GY94* +F1X4+G4	-2251.144	101	4704.287	960.287	4832.263	1072.263
123	MGK+F1X4+G4	-2251.472	101	4704.944	960.944	4832.919	1072.919
124	*GY94* +F3X4+R3	-2227.048	110	4674.096	930.096	4834.754	1074.754
125	*GY94* +F3X4+R2	-2233.068	108	4682.136	938.136	4835.019	1075.019
126	MGK+F3X4+I+G4	-2233.539	108	4683.078	939.0781	4835.962	1075.962
127	MGK+F3X4+G4	-2237.512	107	4689.024	945.024	4838.134	1078.134
128	*GY94* +F3X4+I+G4	-2238.243	108	4692.485	948.485	4845.368	1085.368
129	*GY94* +F3X4+R4	-2227.106	112	4678.213	934.213	4846.96	1086.96
130	*GY94* +F3X4+G4	-2242.394	107	4698.789	954.789	4847.899	1087.899

131	*GY94* +F1X4+I	-2260.085	101	4722.169	978.169	4850.144	1090.144
132	MGK+F1X4+I	-2260.345	101	4722.69	978.69	4850.665	1090.665
133	MGK+F3X4+I	-2246.112	107	4706.225	962.225	4855.335	1095.335
134	MG+F1X4+R2	-2268.482	101	4738.963	994.963	4866.938	1106.938
135	*GY94* +F3X4+I	-2252.532	107	4719.064	975.064	4868.174	1108.174
136	MG+F3X4+R2	-2254.453	107	4722.906	978.906	4872.015	1112.015
137	MG+F1X4+I+G4	-2272.057	101	4746.113	1002.113	4874.089	1114.089
138	MG+F1X4+R3	-2267.523	103	4741.047	997.047	4875.789	1115.789
139	MG+F1X4+G4	-2276.171	100	4752.342	1008.342	4877.033	1117.033
140	MG+F3X4+I+G4	-2257.945	107	4729.891	985.891	4879.001	1119.001
141	MG+F3X4+G4	-2261.949	106	4735.898	991.898	4881.309	1121.309
142	MG+F3X4+R3	-2253.514	109	4725.027	981.027	4881.759	1121.759
143	SYM	-2329.878	100	4859.756	1115.756	4889.116	1129.116
144	TIMe	-2333.105	98	4862.21	1118.21	4890.332	1130.332
145	TIM3e	-2333.481	98	4862.961	1118.961	4891.083	1131.083
146	TVMe	-2333.164	99	4864.328	1120.328	4893.065	1133.065
147	GTR+F	-2328.404	103	4862.809	1118.809	4894.085	1134.085
148	K3P	-2336.391	97	4866.783	1122.783	4894.297	1134.297
149	MG+F1X4+I	-2284.946	100	4769.892	1025.892	4894.583	1134.583
150	TVM+F	-2330.086	102	4864.172	1120.172	4894.802	1134.802
151	TIM+F	-2331.48	101	4864.96	1120.96	4894.952	1134.952
152	TNe	-2336.729	97	4867.458	1123.458	4894.972	1134.972
153	K3Pu+F	-2333.162	100	4866.323	1122.323	4895.684	1135.684
154	TIM3+F	-2331.971	101	4865.942	1121.942	4895.934	1135.934
155	TPM3+F	-2333.648	100	4867.297	1123.297	4896.657	1136.657
156	TPM3u+F	-2333.648	100	4867.297	1123.297	4896.657	1136.657
157	TIM2e	-2336.292	98	4868.584	1124.584	4896.706	1136.706
158	MG+F3X4+I	-2270.442	106	4752.885	1008.885	4898.295	1138.295
159	K2P	-2340.015	96	4872.03	1128.03	4898.943	1138.943
160	TN+F	-2335.102	100	4870.204	1126.204	4899.565	1139.565
161	HKY+F	-2336.783	99	4871.566	1127.566	4900.303	1140.303
162	TIM2+F	-2334.7	101	4871.401	1127.401	4901.392	1141.392
163	TPM2u+F	-2336.381	100	4872.761	1128.761	4902.122	1142.122
164	TPM2+F	-2336.381	100	4872.762	1128.762	4902.123	1142.123
165	JC	-2366.286	95	4922.571	1178.571	4948.892	1188.892
166	F81+F	-2362.554	98	4921.108	1177.108	4949.229	1189.229
167	*GY94* +F1X4	-2315.788	100	4831.575	1087.575	4956.267	1196.267
168	KOSI07+FU+R2	-2325.725	97	4845.45	1101.45	4960.675	1200.675
169	MGK+F1X4	-2318.048	100	4836.095	1092.095	4960.787	1200.787
170	KOSI07+FU+R3	-2323.063	99	4844.126	1100.126	4965.599	1205.599
171	MGK+F3X4	-2304.357	106	4820.713	1076.713	4966.124	1206.124
172	*GY94* +F3X4	-2306.17	106	4824.339	1080.339	4969.749	1209.749
173	KOSI07+FU+I+G4	-2335.554	97	4865.108	1121.108	4980.332	1220.332
174	KOSI07+FU+G4	-2339.513	96	4871.026	1127.026	4983.218	1223.218
175	KOSI07+F3X4+R2	-2315.814	106	4843.627	1099.627	4989.038	1229.038

176	KOSI07+F3X4+R3	-2310.509	108	4837.018	1093.018	4989.901	1229.901
177	KOSI07+F1X4+R2	-2333.491	100	4866.983	1122.983	4991.674	1231.674
178	KOSI07+F1X4+R3	-2328.692	102	4861.383	1117.383	4992.708	1232.708
179	SCHN05+FU+R2	-2344.705	97	4883.411	1139.411	4998.635	1238.635
180	KOSI07+F1X4+I+G4	-2337.965	100	4875.93	1131.93	5000.621	1240.621
181	KOSI07+F1X4+G4	-2341.156	99	4880.312	1136.312	5001.784	1241.784
182	SCHN05+FU+R3	-2341.179	99	4880.358	1136.358	5001.831	1241.831
183	KOSI07+FU+I	-2349.617	96	4891.233	1147.233	5003.426	1243.426
184	KOSI07+F3X4+I+G4	-2323.767	106	4859.534	1115.534	5004.944	1244.944
185	MG+F1X4	-2342.797	99	4883.593	1139.593	5005.065	1245.065
186	KOSI07+F3X4+G4	-2327.376	105	4864.751	1120.751	5006.534	1246.534
187	MG+F3X4	-2328.539	105	4867.078	1123.078	5008.861	1248.861
188	SCHN05+F1X4+R3	-2340.927	102	4885.854	1141.854	5017.179	1257.179
189	KOSI07+F1X4+I	-2349.1	99	4896.2	1152.2	5017.672	1257.672
190	SCHN05+F3X4+R3	-2324.472	108	4864.944	1120.944	5017.827	1257.827
191	SCHN05+FU+I+G4	-2354.523	97	4903.046	1159.046	5018.27	1258.27
192	SCHN05+F1X4+R2	-2348.226	100	4896.452	1152.452	5021.143	1261.143
193	SCHN05+F3X4+R2	-2331.916	106	4875.833	1131.833	5021.243	1261.243
194	SCHN05+FU+G4	-2358.682	96	4909.365	1165.365	5021.558	1261.558
195	KOSI07+F3X4+I	-2336.826	105	4883.653	1139.653	5025.436	1265.436
196	SCHN05+F1X4+I+G4	-2351.096	100	4902.192	1158.192	5026.883	1266.883
197	SCHN05+F1X4+G4	-2353.895	99	4905.79	1161.79	5027.263	1267.263
198	SCHN05+F1X4+R4	-2340.593	104	4889.187	1145.187	5027.414	1267.414
199	SCHN05+F3X4+R4	-2324.102	110	4868.203	1124.203	5028.861	1268.861
200	SCHN05+F3X4+I+G4	-2338.345	106	4888.69	1144.69	5034.101	1274.101
201	SCHN05+F3X4+G4	-2341.811	105	4893.621	1149.621	5035.404	1275.404
202	SCHN05+FU+I	-2370.471	96	4932.943	1188.943	5045.135	1285.135
203	SCHN05+F1X4+I	-2363.696	99	4925.391	1181.391	5046.864	1286.864
204	SCHN05+F3X4+I	-2352.81	105	4915.621	1171.621	5057.404	1297.404
205	KOSI07+FU	-2394.782	95	4979.563	1235.563	5088.785	1328.785
206	KOSI07+F1X4	-2398.44	98	4992.88	1248.88	5111.197	1351.197
207	KOSI07+F3X4	-2383.159	104	4974.318	1230.318	5112.546	1352.546
208	SCHN05+FU	-2419.333	95	5028.665	1284.665	5137.887	1377.887
209	SCHN05+F1X4	-2416.544	98	5029.088	1285.088	5147.405	1387.405
210	SCHN05+F3X4	-2402.838	104	5013.675	1269.675	5151.903	1391.903
211	*GY94* +F+R2	-2208.59	159	4735.181	991.181	5229.161	1469.161
212	*GY94* +F+G4	-2217.694	158	4751.388	1007.388	5234.504	1474.504
213	*GY94* +F+I+G4	-2213.659	159	4745.319	1001.319	5239.299	1479.299
214	*GY94* +F+R3	-2202.599	161	4727.198	983.198	5243.673	1483.673
215	*GY94* +F+I	-2228.346	158	4772.691	1028.691	5255.807	1495.807
216	*GY94* +F+R4	-2202.61	163	4731.219	987.219	5271.26	1511.26
217	*GY94* +F	-2282.254	157	4878.509	1134.509	5351.004	1591.004
218	KOSI07+F+R2	-2291.643	157	4897.286	1153.286	5369.781	1609.781
219	KOSI07+F+G4	-2301.662	156	4915.325	1171.325	5377.438	1617.438
220	KOSI07+F+I+G4	-2298.418	157	4910.835	1166.835	5383.33	1623.33

221	KOSI07+F+R3	-2286.723	159	4891.446	1147.446	5385.426	1625.426
222	KOSI07+F+I	-2311.78	156	4935.559	1191.559	5397.672	1637.672
223	SCHN05+F+R2	-2310.015	157	4934.03	1190.03	5406.525	1646.525
224	SCHN05+F+G4	-2316.684	156	4945.369	1201.369	5407.482	1647.482
225	SCHN05+F+I+G4	-2313.733	157	4941.467	1197.467	5413.962	1653.962
226	SCHN05+F+R3	-2303.732	159	4925.463	1181.463	5419.444	1659.444
227	SCHN05+F+I	-2327.127	156	4966.254	1222.254	5428.367	1668.367
228	SCHN05+F+R4	-2303.45	161	4928.9	1184.9	5445.375	1685.375
229	KOSI07+F	-2357.579	155	5025.157	1281.157	5477.12	1717.12
230	SCHN05+F	-2379.264	155	5068.528	1324.528	5520.491	1760.491

Table 4.3: Model selection of 230 models of nucleotide and codon evolution.

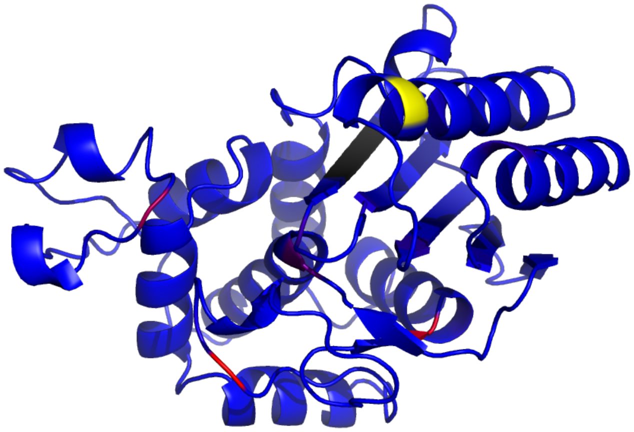


Figure 4.6: Distribution of genetic load in TEM mapped on its structure (1xpb). Average genetic load over all observed TEM variants is indicated by the color, blue low, red medium, yellow high genetic load. Active site is indicated in black.

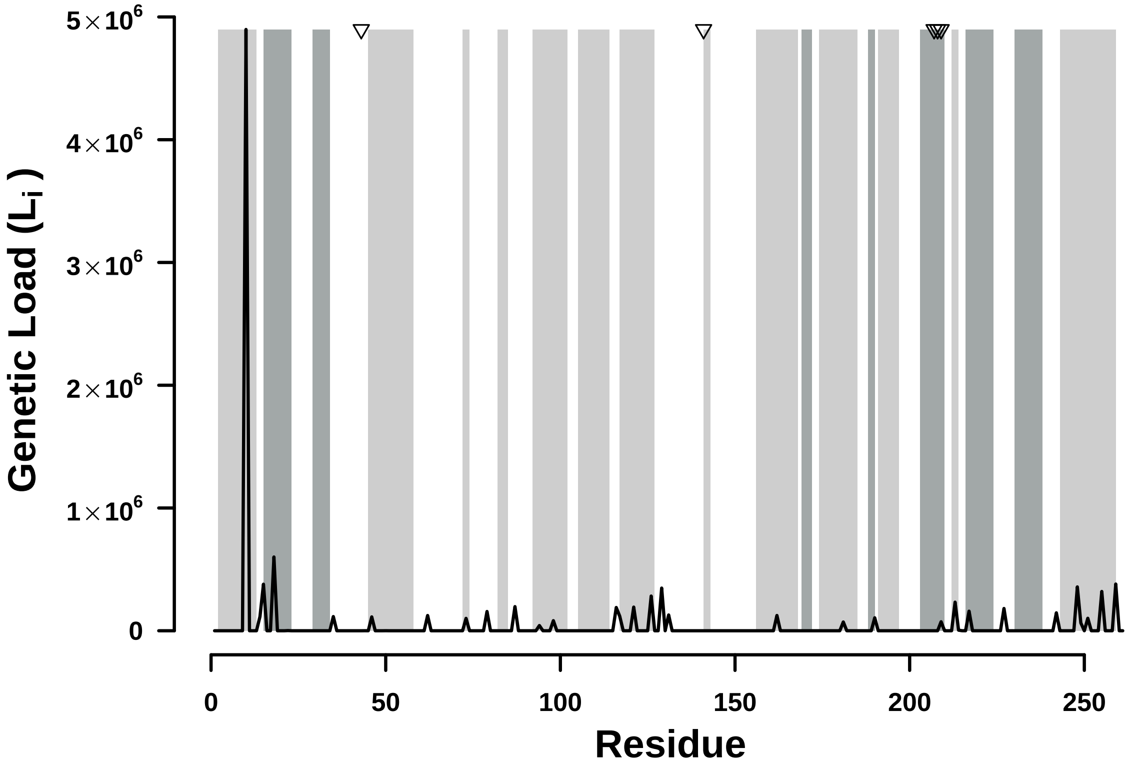


Figure 4.7: Distribution of genetic load in SHV. Average genetic load over all observed SHV variants is indicated by the black line. Light gray bars indicate where helices are found, and dark gray bars indicate β-sheets. The three residues forming the active sites are indicated by three triangles at the top of the plot.

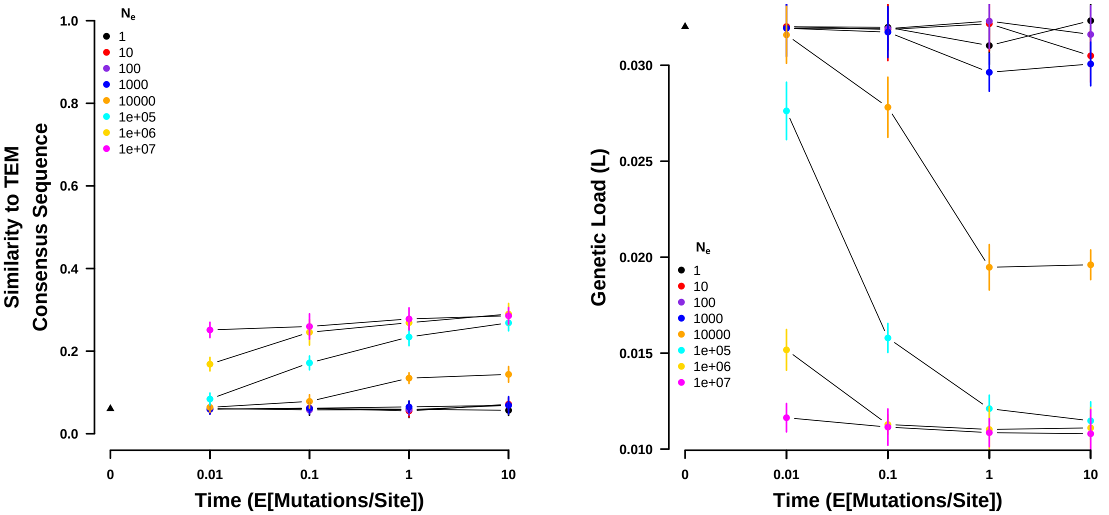


Figure 4.8: Sequences simulated from a random codon sequence under the site specific selection on amino acids estimated using $SelAC$. (left) Sequence similarity to the observed consensus sequence at various times for a range on values of N_e. (right) Genetic load of the simulated sequences at various times for a range on values of N_e. Time is given in number of expected mutations per site, which equals the substitution rate of a neutral mutation. Points indicate sample means and vertical bars indicate standard deviations. Initial sequence is the inferred ancestral state of the TEM variants and indicated by a black triangle.

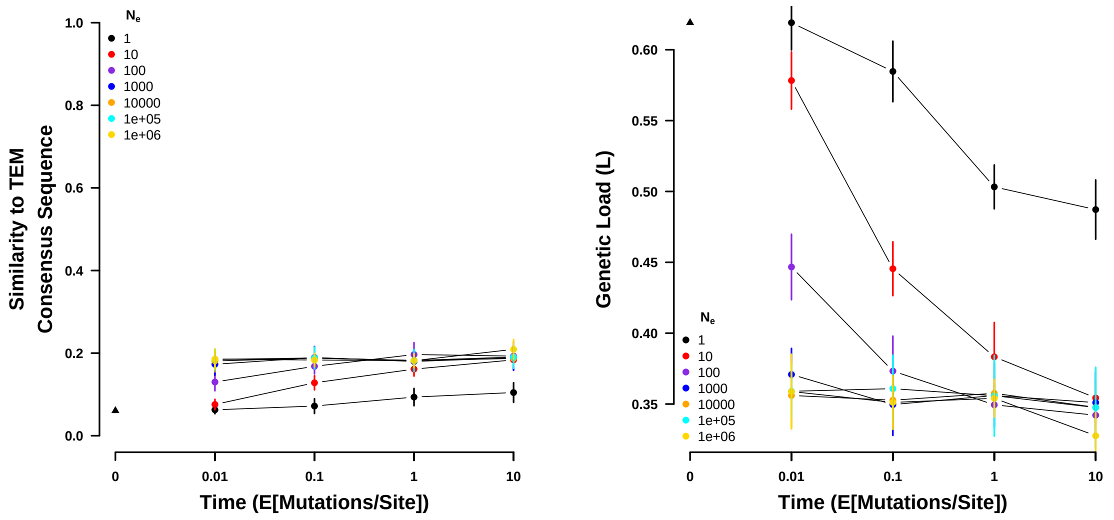


Figure 4.9: Sequences simulated from a random codon sequence under the site specific selection on amino acids estimated using deep mutation scanning. (left) Sequence similarity to the observed consensus sequence at various times for a range on values of N_e. (right) Genetic load of the simulated sequences at various times for a range on values of N_e. Time is given in number of expected mutations per site, which equals the substitution rate of a neutral mutation. Points indicate sample means and vertical bars indicate standard deviations. Initial sequence is the inferred ancestral state of the TEM variants and indicated by a black triangle.

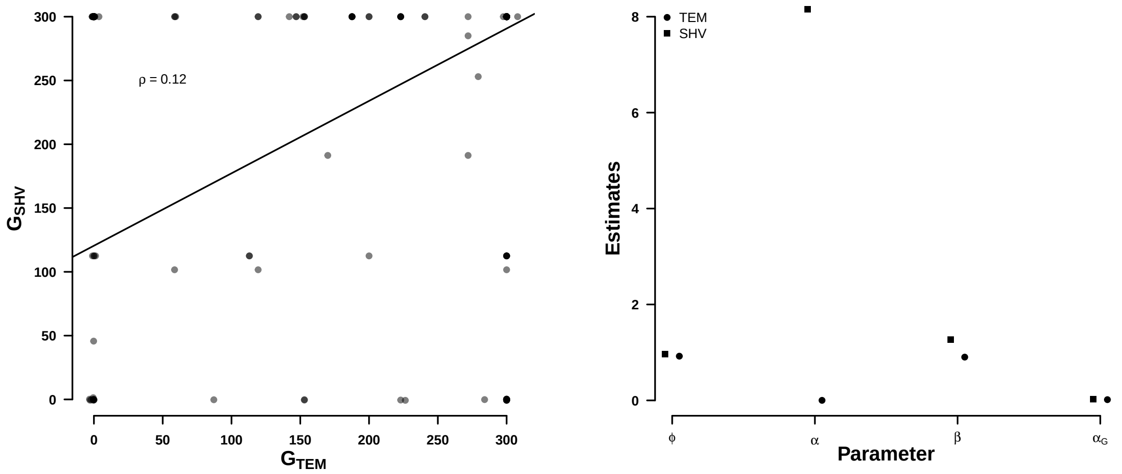


Figure 4.10: Comparison of selection related parameters between TEM and SHV. (left) Estimated site specific efficacy of selection G. (right) Selection related parameter estimates. Protein functionality production rate ψ, physicochemical weight for amino acid composition α_c, physicochemical weight for amino acid polarity α_p, and the parameter describing the distribution of G, α_G estimated by *SelAC*.

Chapter 5

Conclusion

Bibliography

[1] Altschul, S. (1991). Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology*, 219(3):555–565. 1

[2] Anfinsen, C. (1973). Principles that govern the folding of protein chains. *Science*, 181(4096):223–230. 2

[3] Ashenberg, O., Gong, L., and Bloom, J. (2013). Mutational effects on stability are largely conserved during protein evolution. *Proccedings of the National Academy of Sciences U.S.A*, 110:21071–21076. 59

[4] Ashkenazy, H., Penn, O., Doron-Faigenboim, A., Cohen, O., Cannarozzi, G., Zomer, O., and Pupko, T. (2012). Fastml: a web server for probabilistic reconstruction of ancestral sequences. *Nucleic Acids Research*, 40(Web Server Issue):W580–4. 76

[5] Beaulieu, J., O'Meara, B., Zaretzki, R., Landerer, C., Chai, J., and Gilchrist, M. (in review). Population genetics based phylogenetics under stabilizing selection for an optimal amino acid sequence: A nested modeling approach. *Molecular Biology and Evolution*, X:NA. 4, 60, 61, 64, 72, 76, 77

[6] Bennetzen, J. and Hall, B. (1982). Codon selection in yeast. *J. Biol. Chem.*, 257:3026–3031. 1

[7] Bloom, J. (2014). An experimentally informed evolutionary model improves phylogenetic fit to divergent lactamase homologs. *Molecular Biology and Evolution*, 31(10):2753–2769. 59, 60, 69, 71, 73

[8] Bloom, J. (2017). Identification of positive selection in genes is greatly improved by using experimentally infromed site-specific models. *Biology Direct*, 12:1. 59, 69, 71, 73, 75

[9] Booch, G. (1993). *Object-oriented analysis and design with applications*. Benjamin-Cummings Publishing Co, Redwood City. 9

[10] Brun, T., Peduzzi, J., Canica, M., Paul, G., Nevot, P., Barthelemy, M., and Labia, R. (1994). Characterization and amino acid sequence of irt-4, a novel tem-type enzyme with a decreased susceptibility to beta-lactamase inhibitors. *FEMS Microbiology Letters*, 120:111–117. 72

[11] Chanal, C., Poupart, M., Sirot, D., Labia, R., Sirot, J., and Cluzel, R. (1992). Nucleotide sequences of caz-2, caz-6, and caz-7 beta-lactamase genes. *Antimicrob. Agents Chemother.*, 36:1817–1820. 72

[12] Davis, M. and Pelsor, M. (2001). Experimental support for a resourcebased mechanistic model of invasibility. *Ecology Letters*, 4(5):421–428. 2

[13] Dayhoff, MO andSchwartz, R. and Orcutt, B. (1978). A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure*, 5(3):345–352. 1

[14] de Koning, A. and De Sanctis, B. (2018). The rate of molecular evolution when mutation may not be weak. *bioRxiv*. 73

[15] Doron-Faigenboim, A. and Pupko, T. (2007). A combined empirical and mechanistic codon model. *Molecular Biology and Evolution*, 24(2):388–397. 2

[16] dos Reis, M., Savva, R., and Wernisch, L. (2004). Solving the riddle of codon usage preferences: a test for translational selection. *Nucleic Acids Research*, 32(17):5036–5044. 46

[17] Dunn, C., Zapata, F., Munro, C., Siebert, S., and Hejnol, A. (2018). Pairwise comparisons across species are problematic when analyzing functional genomic data. *Proc Natl Acad Sci USA*. 10

[18] Echave, J., Spielman, S., and Wilke, C. (2016). Causes of evolutionary rate variation among protein sites. *Nature Reviews Genetics*, 17:109–121. 59

[19] Edelbuettel, D. and Francois, R. (2011). Rcpp: Seamless r and c++ integration. *Journal of Statistical Software*, 40:1–18. 9, 18

[20] Felsenstein, J. (1981). Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of Molecular Evolution*, 17:368–376. 1, 59

[21] Felsenstein, J. (2001). Taking variation of evolutionary rates between sites into account in inferring phylogenies. *Journal of Molecular Evolution*, 53(4):447–455. 76

[22] Firnberg, E., Labonte, J., Gray, J., and Ostermeier, M. (2014). A comprehensive, high-resolution map of a gene's fitness landscape. *Molecular Biology and Evolution*, 31(6):1581–1592. 60

[23] Firnberg, E. and Ostermeier, M. (2012). Pfunkel: Efficient, expansive, user-defined mutagenesis. *PLOS ONE*, 7(12):e52031. 60, 72, 74

[24] Fitch, W. (1976). Is there selection against wobble in codon-anticodon pairing? *Science*, 194:1173–1174. 1

[25] Fowler, D. and Fields, S. (2014). Deep mutational scanning: a new style of protein science. *Nature Methods*, 11:801–807. 72

[26] Fowler, D., Stephany, J., and Fields, S. (2014). Measuring the activity of protein variants on a large scale using deep mutational scanning. *Nature Protocols*, 9:2267–2284. 59, 72

[27] Friedrich, A., Reiser, C., Fischer, G., and Schacherer, J. (2015). Population genomics reveals chromosome-scale heterogeneous evolution in a protoploid yeast. *Molecular Biology and Evolution*, 32(1):184 – 192. 34, 39, 40, 44, 45, 46

[28] Gilchrist, M. (2007). Combining models of protein translation and population genetics to predict protein production rates from codon usage patterns. *Molecular Biology and Evolution*, 24(11):2362–2372. 2, 77

[29] Gilchrist, M., Chen, W., Shah, P., Landerer, C., and Zaretzki, R. (2015). Estimating gene expression and codon-specific translational efficiencies, mutation biases, and selection coefficients from genomic data alone. *Genome Biology and Evolution*, 7:1559–1579. 2, 3, 7, 10, 12, 25, 34, 40, 42, 44

[30] Gillespie, D. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434. 76

[31] Gojobori, T. (1983). Codon substitution in evolution and the ”saturation” of synonymous changes. *Genetics*, 105:1011–1027. 59

[32] Goldman, N. and Yang, Z. H. (1994). Codon-based model of nucleotide substitution for protein-coding DNA-sequences. *Molecular Biology and Evolution*, 11:725–736. 59, 64

[33] Goussard, S., Sougakoff, W., Mabilat, C., Bauernfeind, A., and Courvalin, P. (1991). An is1-like element is responsible for high-level synthesis of extended-spectrum beta-lactamase tem-6 in enterobacteriaceae. *J. Gen. Microbiol.*, 137:2681–2687. 72

[34] Gouy, M. and Gautier, C. (1982). Codon usage in bacteria: correlation with gene expressivity. *Nucleic Acids Research*, 10:7055–7074. 33

[35] Grantham, R. (1974). Amino acid differences formula to help explain protein evolution. *Science*, 185(4154):862–864. xi, 4, 72, 74, 75

[36] Grantham, R., Gautier, C., and Gouy, M. (1980). Codon frequencies in 119 individual genes confirms consistent choices of degenerate bases according to genome type. *Nucleotide Acid Research*, 8:1893–1912. 1

[37] Grantham, R., Gautier, C., Gouy, M., Jacobzone, M., and Mercier, R. (1981). Codon catalog usage is a genome strategy modulated for gene expressivity. *Nucleic Acids Research*, 9:43–74. 1

[38] Halpern, A. and Bruno, W. (1998). Evolutionary distances for protein-coding sequences: Modeling site-specific residue frequencies. *Molecular Biology and Evolution*, 15(7):910–917. 59

[39] Hartl, D., Moriyama, E., and Sawyer, S. (1994). Selection intensity for codon bias. *Genetics*, 138:227–234. 73

[40] Hilton, S., Doud, M., and Bloom, J. (2017). phydms: software for phylogenetic analyses informed by deep mutation scanning. *PeerJ*, 5:e3657. 59, 60, 61

[41] Holder, M., Zwickl, D., and Dessimoz, C. (2008). Evaluating the robustness of phylogenetic methods to among-site variability in substitution processes. *Philos Trans R Soc Lond B*, 363:4013–4021. 59

[42] Ikemura, T. (1981). Correlation between the abundance of *Escherichia coli* transfer rnas and the occurrence of the respective codons in its protein genes: a proposal for a synonymous codon choice that is optimal for the *E. coli* translational system. *Journal of Molecular Biology*, 151:389–409. 1

[43] Ikemura, T. (1985). Codon usage and trna content in unicellular and multicellular organisms. *Molecular Biology and Evolution*, 2:13–34. 33

[Inc.] Inc., W. R. Mathematica, Version 9.0. Champaign, IL, 2012. 48

[45] Jain, P. and Varadarajan, R. (2014). A rapid, efficient, and economical inverse polymerase chain reaction-based method for generating a site saturation mutant library. *Analytical Biochemistry*, 449:90–981. 72, 74

[46] Jukes, T. and Cantor, C. (1969). *Evolution of Protein Molecules*, pages 21–132. Academic Press. 1

[47] Kawashima, S., Pokarowski, P., Pokarowska, M., Kolinski, A., Katayama, T., and Kanehisa, M. (2008). Aaindex: amino acid index database, progress report 2008. *Nucleic Acids Research*, 36:D202–D205. 74

[48] Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16(2):111–120. 1

[49] Kosiol, C., Holmes, I., and Goldman, N. (2007). An empirical codon model for protein sequence evolution. *Molecular Biology and Evolution*, 24(7):1464–1479. xiii, 63, 64

[50] Landerer, C., Cope, A., Zaretzki, R., and Gilchrist, M. A. (2018). Anacoda: analyzing codon data with bayesian mixture models. *Bioinformatics*, 34(14):2496–2498. 3, 35, 47

[51] Lang, G. I. and Murray, A. W. (2008). Estimating the per-base-pair mutation rate in the yeast saccharomyces cerevisiae. *Genetics*, 178(1):67 – 82. 48

[52] Lartillot, N. and Philippe, H. (2004). A bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. *Molecular Biology and Evolution*, 21:1095–1109. 59

[53] Laureau, M. (1998). Biodiversity and ecosystem functioning: A mechanistic model. *Proccedings of the National Academy of Sciences U.S.A*, 95:5632–5636. 2

[54] Le, S., Lartillot, N., and O, G. (2008). Phylogenetic mixture models for proteins. *Philos Trans R Soc Lond B Biol Sci*, 363:3965–3976. 59

[55] Leder, P. and Nierenberg, M. (1964). Rna codewords and protein synthesis, iii. on the nucleotide sequence of a cysteine and leucine rna codeword. *Proccedings of the National Academy of Sciences U.S.A*, 52:1521–1529. 1

[56] Mabilat, C., Lourencao-Vital, J., Goussard, S., and Courvalin, P. (1992). A new example of physical linkage between tn1 and tn21: the antibiotic multiple-resistance region of plasmid pcff04 encoding extended-spectrum beta-lactamase tem-3. *Mol Gen Genet*, 235:113–121. 72

[57] Matagne, A., Lamotte-Brasseur, J., and Frere, J. (1998). Catalytic properties of class a beta-lactamases: efficiency and diversity. *Biochemistry Journal*, 300:581–598. 73

[58] Matthaei, J. and Nierenberg, M. (1961). Characteristics and stabilization of dnaase-sensitice protein synthesis in *E. coli* extracts. *Proccedings of the National Academy of Sciences U.S.A*, 47(10):1580–1588. 1

[59] Maxwell, E. (1962). Stimulation of amino acid incorporation into protein by natural and synthetic polyribonucleotides in a mammalian cell-free system. *Proccedings of the National Academy of Sciences U.S.A*, 48(9):1639–1643. 1

[60] Mi, G., Di, Y., and Schafer, D. (2015). Goodness-of-fit tests and model diagnostics for negative binomial regression of rna sequenceing data. *PLOS ONE*, 10:e0119254. 9

[61] Muse, S. and Gaut, B. (1994). A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Molecular Biology and Evolution*, 11(5):715–724. 59

[62] Nguyen, L., Schmidt, H., von Haeseler, A., and Minh, B. (2015). Iq-tree: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32(1):268–274. 60, 75

[63] Nierenberg, M. and Matthaei, J. (1961). The dependence of cell-free protein synthesis in *E. coli* upon naturally occurring or synthetic polyribonucleotides. *Proccedings of the National Academy of Sciences U.S.A*, 47(10):1588–1602. 1

[64] Ochman, H. and Wilson, A. (1987). *Evolutionary history of enteric bacterian*, pages 1649–1654. ASM Press. 73

[65] Payen, C., Fischer, G., Marck, C., Proux, C., Sherman, D. J., Coppe, J.-Y., Johnston, M., Dujon, B., and Neuvglise, C. (2009). Unusual composition of a yeast chromosome arm is associated with its delayed replication. *Genome Research*, 19(10):1710–1721. 34, 41, 43, 46

[66] R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 7, 75

[67] Rodrigue, N. (2013). On the statistical interpretation of site-specific variables in phylogeny-based substitution models. *Genetics*, 193:557–564. 59

[68] Rodrigue, N. and Lartillot, N. (2014). Site-heterogeneous mutation-selection models within the phylobayes-mpi package. *Bioinformatics*, 30:1020–1021. 59

[69] Rodrigue, N., Philippe, H., and Lartillot, N. (2010). Mutation-selection models of coding sequence evolution with site-heterogeneous amino acid fitness profiles. *Proccedings of the National Academy of Sciences U.S.A*, 107:4629–4634. 59

[70] Sella, G. and Hirsh, A. (2005). The application of statistical physics to evolutionary biology. *Proceedings of the National Academy of Sciences of the United States of America*, 102:9541–9546. 76

[71] Shah, P. and Gilchrist, M. (2011a). Explaining complex codon usage patterns with selection for translational efficiency, mutation bias, and genetic drift. *Proccedings of the National Academy of Sciences U.S.A*, 108(25):10231–10236. 2

[72] Shah, P. and Gilchrist, M. (2011b). Explaining complex codon usage patterns with selection for translational efficiency, mutation bias, and genetic drift. *Proc Natl Acad Sci USA*, 108:10231–6. 7, 10

[73] Sharp, P. (1987). The codon adaptatoin index - a meassure of directional synonymous codon usage bias, and its potential applications. *Nucleic Acids Research*, 15:1281–1295. 1, 7, 46

[74] Sharp, P., Cowe, E., Higgins, D., Shields, D., Wolfe, K., and Wright, F. (1988). Codon usage patterns in escherichia coli, bacillus subtilis, saccharomyces cerevisiae, schizosaccharomyces pombe, drosophila melanogaster and homo sapiens; a review of the considerable within species diversity. *Nucleic Acids Research*, 16:8207–8211. 1

[75] Soderlund, C., Bomhoff, M., and Nelson, W. (2011). Symap v3.4: a turnkey synteny system with application to plant genomes. *Nucleic Acids Research*, 39(10):e68. 47

[76] Soderlund, C., Nelson, W., Shoemaker, A., and Paterson, A. (2006). Symap A system for discovering and viewing syntenic regions of fpc maps. *Genome Research*, 16:1159 – 1168. 47

[77] Sougakoff, W., Goussard, S., and Courvalin, P. (1988). The tem-3 beta-lactamase, which hydrolyzes broad-spectrum cephalosporins, is derived from the tem-2 penicillinase by two amino acid substitutions. *FEMS Microbiology Letters*, 56:343–348. 72

[78] Sougakoff, W., Petit, A., Goussard, S., Sirot, D., Bure, A., and Courvalin, P. (1989). Characterization of the plasmid genes blat-4 and blat-5 which encode the broad-spectrum beta-lactamases tem-4 and tem-5 in enterobacteriaceae. *Gene*, 78:339–348. 72

[79] Stiffler, M., Hekstra, D., and R, R. (2016). Evolvability as a function of purifying selection in tem-1 β-lactamase. *Cell*, 160:882–892. 60, 72, 75

[80] Tamuri, A., Goldman, N., and dos Reis, M. (2014). A penalized likelihood method for estimating the distribution of selection coefficients from phylogenetic data. *Genetics*, 197:257–271. 59

[81] Thorne, J., Goldman, N., and Jones, D. (1996). Combinng protein evolution and secondary structure. *Molecular Biology and Evolution*, 13:666–673. 59

[82] Thyagarajan, B. and Bloom, J. (2014). The inherent mutational tolerance and antigenic evolvability of influenza hemagglutinin. *eLife*, 3:e03300. 59

[83] Tsankov, A., Thompson, D., Socha, A., Regev, A., and Rando, O. (2010). The role of nucleosome positioning in the evolution of gene regulation. *PLoS Biol*, 8(7):e1000414. xii, 36

[84] Vakirlis, N., Sarilar, V., Drillon, G., Fleiss, A., Agier, N., Meyniel, J.-P., Blanpain, L., Carbone, A., Devillers, H., Dubois, K., Gillet-Markowska, A., Graziani, S., Huu-Vang, N., Poirel, M., Reisser, C., Schott, J., Schacherer, J., Lafontaine, I., Llorente, B., Neuvéglise, C., and Fischer, G. (2016). Reconstruction of ancestral chromosome architecture and gene repertoire reveals principles of genome evolution in a model yeast genus. *Genome research*, 26(7):918–32. 43

[85] Wagner, A. (2005). Energy constraints on the evolution of gene expression. *Molecular Biology and Evolution*, 22:1365–1374. 49

[86] Wallace, E., Airoldi, E., and Drummond, D. (2013). Estimating selection on synonymous codon usage from noisy experimental data. *Molecular Biology and Evolution*, 30:1438–1453. 7, 10

[87] Wang, H., Li, K., Susko, E., and Roger, A. (2008). A class frequency mixture model that adjusts for site-specific amino acid frequencies and improves inference of protein phylogeny. *BMC Evolutionary Biology*, 8:331. 59

[88] Wright, F. (1990). The 'effective number of codons' used in a gene. *Gene*, 87(1):23–29. 1

[89] Wu, C., Suchard, M., and Drummond, A. (2013). Bayesian selection of nucleotide substitution models and their site assignments. *Molecular Biology and Evolution*, 30:669–688. 59

[90] Yang, Z. (1994). Maximum-likelihood phylogenetic estimation from DNA-sequences with variable rates over sites - approximate methods. *Journal of Molecular Evolution*, 39:306–314. 75

[91] Zharkikh, A. (1994). Estimation of evolutionary distances between nucleotide sequences. *Journal of Molecular Evolution*, 39(3):315–329. 64

[92] Zuckerkandl, E. and Pauling, L. (1962). *Molecular disease, evolution, and genic heterogeneity*, pages 189–225. Academic Press. 1

Vita

Cedric Landerer was born in Floersheim am Main, Germany on December 22, 1986 and raised in Frankfurt am Main, Germany. He graduated from Heinrich-Kleyer Highschool in Frankfurt, Germany in 2006. After that he moved to Munich, Germany to study Bioinformatics in a joined major at the University of Munich and Technical University, Munich. He recieved his Bachelor of Science in Bioinformatics in 2011 and his Masters of Science in Bioinformatics in 2013. He joined the Department of Ecology and Evolutionary Biology at the University of Tennessee, Knoxville in 2013 to persue his Ph. D. He recieved his Ph. D. in Ecology and Evolutionary Biology in 2018 and will start a postdoctoral position at the Max Planck Institute for Molecular Cell Biology and Genetics in Dresden Germany in January 2019.