



HAL
open science

Phylogenetic detection of protein sites associated to a phenotype, at the genome scale

Louis Duchemin

► **To cite this version:**

Louis Duchemin. Phylogenetic detection of protein sites associated to a phenotype, at the genome scale. Molecular biology. Université Claude Bernard - Lyon I, 2023. English. NNT : 2023LYO10022 . tel-04481543

HAL Id: tel-04481543

<https://theses.hal.science/tel-04481543>

Submitted on 28 Feb 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



**THÈSE de DOCTORAT DE
L'UNIVERSITÉ LYON 1**

**Ecole Doctorale 341
Écosystèmes Évolution Modélisation
Microbiologie**

Discipline : Génomique évolutive

Soutenue publiquement le 01/03/2023, par :

Louis Duchemin

**Détection phylogénétique de
sites protéiques associés à un
phénotype, à l'échelle
génomique**

Devant le jury composé de :

Céline Brochier-Armanet

Professeure, CNRS/LBBE, Université Claude Bernard Lyon 1

Présidente

Maria Anisimova

Directrice de Recherche, ZHAW Zurich University of Applied Sciences (Suisse)

Rapporteure

Nicolas Galtier

Directeur de Recherche, CNRS/ISEM, Université de Montpellier

Rapporteur

Sophie Abby

Chargée de Recherche, CNRS/TIMC, Université Grenoble-Alpes

Examinatrice

Bastien Boussau

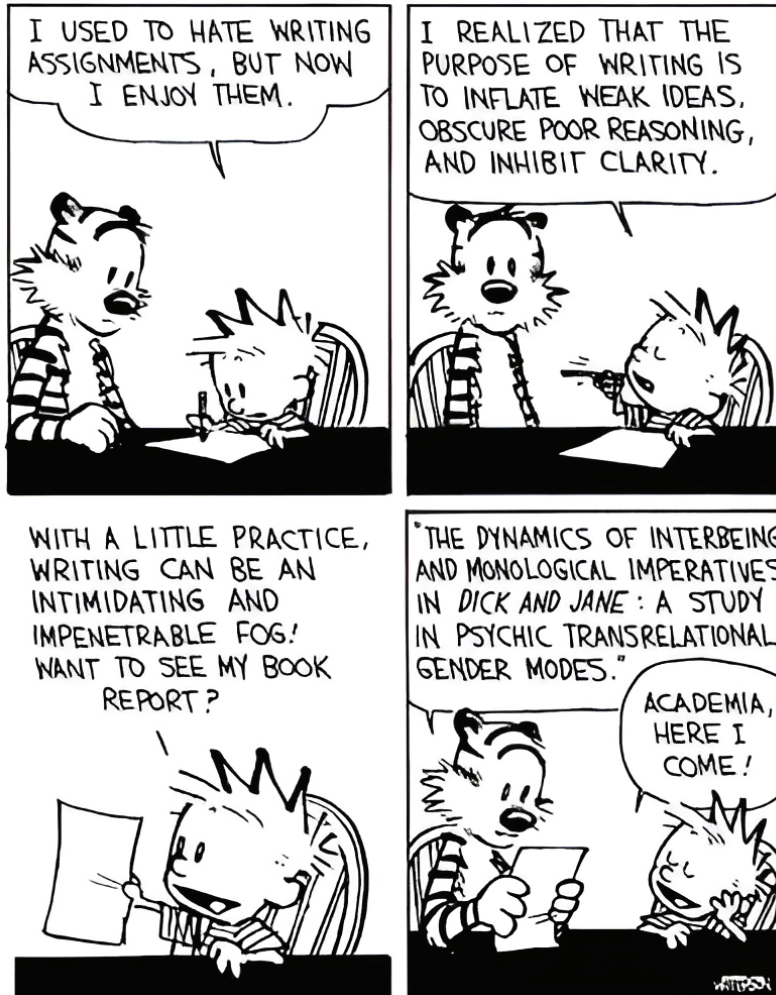
Directeur de Recherche, CNRS/LBBE, Université Claude Bernard Lyon 1

Directeur de thèse

Philippe Veber

Ingénieur de Recherche, CNRS/LBBE, Université Claude Bernard Lyon 1

Co-directeur de thèse



Really tried to do the opposite while writing this thesis — and hope I managed to.

Résumé

Les différences de phénotypes — les caractéristiques physiques, physiologiques et fonctionnelles — entre espèces sont une manifestation de variations qui se sont produites dans leur génome, à l'échelle moléculaire. Ces changements à long terme sont la conséquence de l'interaction entre processus de différentes natures. La mutation, communément considérée comme aléatoire, est le moteur initial de la diversification, et se produit à l'échelle d'un individu. Au fil des générations, un nouveau variant génétique se diffuse au sein d'une population sous l'action combinée d'un processus non-adaptatif, la dérive génétique, et de la sélection qui favorise ou non sa transmission selon l'avantage reproductif que ce variant procure. L'adaptation du vivant à un environnement changeant émerge de la combinaison de ces processus, à partir de laquelle son immense diversité se déploie.

Les espèces actuelles, et donc leurs génomes, partagent une histoire commune de par leur descendance d'une même espèce ancestrale, qui s'est séparée au fil de l'accumulation de divergences entre populations. En associant les séquences génomiques issues d'une même séquence ancestrale, et en examinant leur divergence, il est possible d'interpréter les traces laissées par leur histoire évolutive pour la reconstruire en partie. Parmi les événements de modification du génome, je m'intéresse au cas des substitutions, c'est à dire des remplacements ponctuels à une position, au sein des gènes codants pour des protéines, dont la structure et la fonction peut en être modifiée et donc avoir un effet adaptatif. En confrontant le signal porté par ces substitutions à l'histoire d'un trait phénotypique on peut tenter de déceler une corrélation entre l'histoire évolutive d'un site codant et celle du phénotype. L'identification de telles corrélations pourrait être le signal qu'une position génotypique est impliquée dans l'émergence ou le maintien du phénotype considéré, et plus largement témoigner de son implication dans l'adaptation d'une espèce à un environnement donné.

De nombreux modèles du processus de substitution basés sur ce genre d'approches comparatives existent déjà et sont largement utilisés pour construire et améliorer nos connaissances de l'évolution moléculaire. Il est toutefois difficile de les appliquer à l'échelle génomique pour effectuer une détection systématique des sites associés à un phénotype, du fait de la quantité de données que cela représente et de la limitation de la puissance de calcul existante. Dans cette thèse, je cherche à proposer une solution pour permettre ce genre d'analyse à large échelle à moindre coût en temps, tout en préservant la qualité des prédictions obtenues.

Après des premières tentatives infructueuses d'adapter des modèles linéaires utilisés en GWAS à l'échelle des populations pour étudier les associations génotype-phénotype, pour les appliquer à l'échelle inter-espèces, j'ai identifié une approche qui semble constituer une solution satisfaisante. Celle-ci se base sur un modèle d'évolution des séquences protéiques — le produit de la traduction des séquences d'ADN — publié précédemment, mais dont le potentiel n'avait pas été bien reconnu. J'ai montré, sur la base de simulations, que l'implémentation que nous avons faite de ce modèle permet de déceler des changements dans la dynamique de substitution en association avec des variations du phénotype aussi bien que plusieurs modèles plus complexes et plus coûteux en calculs. Bien qu'elle

ne soit peut-être pas plus rapide que d'autres implémentations de modèles phylogénétiques, ce qu'il faudrait évaluer, elle apparaît comme la plus rapide des méthodes dites "à profils" qui permettent d'estimer une direction pour la sélection.

Une partie de cette thèse est consacrée à détailler cette méthode, que nous appelons Pelican, son modèle, son implémentation et quelques unes de ses limites. Une stratégie alternative pour l'estimation des paramètres du modèle, en déportant les calculs sur GPU pour exploiter leur capacité de parallélisme, est aussi explorée pour tenter d'améliorer la vitesse des analyses. J'ai également proposé une extension du modèle basée sur des phénotypes continus, et non plus catégoriels. Celle-ci demande encore davantage de travail pour évaluer sa validité. Enfin, j'ai cherché à identifier une manière de prédire les gènes associés à un phénotype à partir des prédictions individuelles réalisées à chacune des positions de leur séquence. C'est un objectif difficile à atteindre, du fait de problèmes statistiques inhérents à la méthode, mais j'ai identifié une approche qui permet d'exploiter les prédictions par sites avec une puissance suffisante, bien qu'elle puisse manquer de robustesse dans certains cas.

Afin de valider notre approche sur des données empiriques, je l'ai appliquée à des alignements de gènes de mammifère pour identifier des sites et des gènes associés à divers phénotypes discrets. Les prédictions obtenues, comparées aux annotations et à la littérature existantes, suggèrent que la méthode est capable d'identifier des sites associés au trait considéré de manière relativement fiable. Le résultat de ce travail est l'implémentation logicielle de Pelican, qui bien qu'elle soit encore à un stade précoce, propose une solution pour détecter des associations génotype-phénotype inter-espèces à l'échelle génomique.

Remerciements

Un très grand merci à Bastien et Philippe, pour m’avoir guidé dans cette thèse ! J’ai eu la chance grâce à vous de participer à la recherche avec des gens enthousiastes et bienveillants, qui m’ont donné envie de faire un doctorat (et c’était pas gagné) et ne m’ont donné aucune raison de le regretter, bien au contraire. Merci pour vos conseils et encouragements, pour m’avoir laissé le temps d’apprendre et de comprendre, pour les “après-midis” de développement, pour votre confiance que *mais si ça va marcher*, et pour avoir été les deux parties complémentaires d’une super direction de thèse.

Merci aux membres de mon jury, Sophie Abby, Maria Anisimova, Céline Brochier-Armanet et Nicolas Galtier pour votre intérêt porté à mon travail, et pour avoir consacré un peu de votre temps au contenu de cette thèse, que j’espère vous aurez plaisir à lire.

Merci à la communauté du LBBE, et en particulier aux membres de l’équipe Le Cocon pour m’avoir accueilli durant mon doctorat, dans d’excellentes conditions à tous points de vue, ainsi qu’à Vincent Lanore, Anamaria Necşulea et Nicolas Lartillot pour leur contribution à ce travail. Merci aux thésard·es (et docteurs) Théo, Thibault, Alexandre, Émilie, Alexia, Hugo, Julien, Marco, Djivan... pour avoir partagé cette expérience à un moment ou à un autre — et à Théo et Thibault pour m’avoir montré les ficelles du doctorat. Merci à Vincent Lacroix, qui m’a toujours encouragé à poursuivre dans mes études, et sans qui j’aurais peut-être suivi un chemin différent; et à Arnaud Mary et Marie-France Sagot qui ont participé de cet élan à un moment où c’était nécessaire.

Merci aux membres de mon comité de suivi de thèse, Tamara Ben-Ari, Laurent Jacob, Marie Sémon et Clémentine François, pour leur bienveillance, l’intérêt qu’ils ont porté au déroulement de ma thèse, et leurs questions et conseils lors des comités de suivi. Je remercie tout particulièrement Clémentine pour avoir été la meilleure tutrice au cours de ma thèse, en étant attentive à ce que ça se passe bien à tous les niveaux.

Merci à Florian Malard, avec qui j’ai eu le plaisir de travailler ces dernières années en parallèle de mes études et de mon doctorat, et qui a tout organisé pour que ça ne devienne jamais un poids trop important; merci aussi à Lara Konecny et Christophe Douady pour y avoir collaboré.

Merci aux copains et copines du master bioinfo qui ont fait des deux années avant ma thèse des vrais bons moments de vie étudiante. Merci à mes ami·es, les porteurs du stick solaire, les mineurs d’astéroïdes, les nouveaux parents, celles et ceux qui ont supporté mes radotages, qui sont toujours là quand même et m’ont aidé à tenir le coup. Merci à Laetitia pour avoir traversé les galères avec moi au cours de cette année difficile à tous points de vue. Chaque pas compte, même les plus petits, alors keep it up ! Merci à mon père pour avoir maintenu un intérêt sincère à ce qui façonne ma vie, et merci enfin à ma mère et mes soeurs pour leur soutien et franchement bravo à nous parce que au bout du compte, ben on s’en sort pas si mal !

Phylogenetic detection of protein sites associated to a phenotype, at the genome scale

Abstract

Phenotype variations across species — morphological, physiological, or functional traits — are a manifestation of variations in their genomes at the molecular level. Long-term variations between species are a consequence of multiple interacting processes of different nature. Mutation is the source of diversification that occurs at the level of individual organisms, and is generally considered to be a stochastic process. Through generations, a novel genetic variant diffuses within a population under the combined effect of a non-adaptive process that is genetic drift, and selection that promotes or represses its transmission, depending on the reproduction advantage it provides. Adaptation of species to a perpetually changing environment emerges from the interaction of these processes, from which the massive diversity of life unfolds.

Extent species and their genomes share a common history that stems from the ancestral ascent they share, and separated into distinct species through the accumulation of divergences between populations of the ancestral species. By gathering genomic sequences that originate from the same ancestral sequence, and analyzing their divergence, it is possible to interpret traces left by their evolutionary history and infer parts of it. Among the variety of modifications that may alter genome sequences, I focus on substitutions, i.e. point modifications at one position within protein coding genes, that may result in changes in the structure and function of the protein they encode, with consequences in terms of adaptation. By analyzing the signal in these substitutions, in combination with the history of a phenotypic trait, one may attempt to detect correlations between the evolutionary history of a coding site, and that of the phenotype. The identification of such correlations might then be the signature that a genotype site is involved in the emergence or the stability of the trait under consideration, and more generally hint at its implication in the adaptation of a species to a particular environment.

Many models of substitutions in gene sequences that exploit this comparative approach already exist, and are widely used to develop our knowledge of molecular evolution. However, they are

difficult to apply at the scale of whole genomes for systematic detection of sites associated to a phenotype, because of the large amount of data involved, and limited computing power. In this thesis, I search for a solution to allow this kind of analyses at large scale, that would involve shorter computation times, while preserving the quality of the resulting predictions.

After some unfruitful attempts at adapting linear models used in GWAS at the population scale to study genotype-phenotype associations, in order to make them applicable at the level of species, I identified an approach that seems to be a satisfactory solution. It is based on a model of amino acid sequence evolution — thus working directly at the level of protein sequences, after translation from DNA — that was previously published, but whose potential had not been recognized yet. I have shown, using simulations, that our implementation of this model enables fast and accurate detection of changes in the substitution dynamics that are associated to phenotype variations, just as well as several other more complex and computationally intensive models. Although it might not be a lot faster than other implementations based on phylogenetic models, that we could also evaluate, it appears to be the fastest among so-called “profile” methods, which provide estimates for the direction of selection at one site.

A part of this thesis is dedicated to exposing the details of this method, which we call Pelican, including its model, implementation and some of its limitations. An alternative strategy for fitting the model, using GPU computation to exploit the highly parallel nature of the problem, was also explored to attempt improving the throughput of analysis further. I then describe an extension of the model based on continuous traits, which were initially limited to discrete categories; more efforts are yet required to evaluate the validity of this alternative model. I also investigate several ways to predict genes associated to a phenotype, using site-level predictions obtained at each position of their sequence. This is not an easy task, because of statistical issues inherent to the method, but I came up with an approach that allows to exploit site-level predictions with good statistical power, although its robustness may be lacking in some cases.

Finally, to further validate our approach using empirical data, I applied it to a genome-scale dataset of coding sequence alignments of mammals, to identify sites and genes associated to several discrete phenotypes. The predictions we obtained, when compared to the existing gene annotations and literature, suggest that this method is able to identify sites associated to the trait quite reliably. The result of this work is a software implementation for Pelican that, although it is in an early-stage, is proposed as a solution to detect inter-species genotype-phenotype associations at the genome scale.

Foreword

This thesis is a bit lengthier than I originally wanted. This is mainly because I took the time to elaborate on the research process, and relate all of my explorations of the subject, even those that were unfruitful. The other reason is that, at several points in the manuscript, I recap some fundamental elements in the field of molecular evolution, either on the subject of the evolutionary theory itself, or commonly used models and algorithms. I attempted to limit such explanations to situations where they are relevant and actually helpful to understand the rest of this work.

Each chapter starts with a short introduction, which aims to recall how we got to this point, and the motivations for the work that is presented. Each of them also ends with a summary of the main points that were discussed in the chapter. I hope this will be useful to you as a reader, for quickly assessing which parts of this manuscript are relevant for you depending on the nature of your interest in this work, and for navigating its content with ease.

Contents

Introduction	13
1 Framework for the evaluation of detection methods	19
1.1 Modeling the evolution of sequences using continuous-time Markov chains	20
1.2 The substitution process at the level of populations	22
1.2.1 The mechanism of substitution in population genetics	23
1.3 Simulating alignments in the mutation-selection framework	25
1.3.1 A mutation-selection model to simulate coding sequence alignments	26
1.3.2 Simulations	29
1.3.3 Extensions of the simulator	31
1.3.4 Empirical phylogenies	33
1.4 Measuring the detection performance of a statistical method	33
1.5 Evaluation pipeline: technical aspects	36
2 Is this basically inter-species GWAS?	38
2.1 Linear Mixed Models	39
2.1.1 GEMMA: Genome-wide Efficient Mixed Model Association	40
2.1.2 Linear Mixed Model	42
2.2 Comparison of performance	43
2.2.1 Multinomial as a baseline reference for performance comparison	44
2.2.2 Linear mixed models do not improve on Multinomial	46
2.2.3 The cost of ignoring the phylogeny	47
2.2.4 Multinomial with phylogenetic random effect: it's complicated	48
3 Evaluation of methods to detect shifts in directional selection at the genome scale	51
3.1 Introduction	53
3.2 New Approaches	57
3.3 Results	57
3.3.1 Detection performances on synthetic trees	59
3.3.2 Detection performances on empirical phylogenies	61
3.4 Discussion	65
3.4.1 Mutation-selection models for simulating coding sequences	65
3.4.2 Methods working at the amino acid level perform as well as codon-based methods	66
3.4.3 Features of a data set that affect performances	67
3.4.4 GC-biased gene conversion is an important confounding factor for both dN/dS and profile methods	67

3.4.5	Persistent positive selection is an important confounding factor for dN/dS methods, less so for profile methods	68
3.4.6	Interpreting screens for changes in directional selection	68
3.4.7	Looking forward	69
3.5	Conclusion	70
3.6	Methods	70
3.6.1	Detection of dN/dS variations using <code>codeml</code>	70
3.6.2	Multinomial method	70
3.6.3	Pelican: improvements on TDG09	71
3.6.4	Simulations	71
3.6.5	gBGC simulation	72
3.6.6	CpG simulation	72
3.6.7	Simulation of persistent positive selection	73
4	Pelican: a fast phylogenetic method to identify selective pressure changes	75
4.1	Technicals	77
4.1.1	Inputs and outputs	77
4.1.2	Other features	79
4.2	The original model: TDG09	80
4.2.1	A general time-reversible model of sequence evolution	80
4.2.2	Model parameters and empirical exchangeabilities	80
4.2.3	A condition specific model of sequence evolution	81
4.2.4	Hypothesis testing	82
4.3	Implementation and fitting of the model	83
4.3.1	Felsenstein's algorithm for phylogenetic likelihood computation	83
4.3.2	Numerical optimization	86
4.3.3	Efficient matrix exponentiation	87
4.3.4	State space reduction	89
4.4	On hypothesis testing and the applicability of LRT	91
4.4.1	Wilks' asymptotic null distribution: the problem of the effective sample size	92
4.4.2	Degrees of freedom	95
4.5	Improving the detection of relaxed selection	98
4.5.1	Motivation: a pathological case of Influenza	98
4.5.2	A better approximation of the null distribution of log-ratios	100
4.5.3	Adapting the model to distinguish neutral from purifying selection	101
4.5.4	Performances obtained with the Pelican variants	103
4.6	Filtering sites using Multinomial, a fast non-phylogenetic method	104
4.7	An alternative approach for fitting the model using automatic differentiation	105
4.7.1	Vectorized implementation of Felsenstein algorithm	105
4.7.2	Optimisation of the parameters using L-BFGS	108
4.7.3	Results	109
4.7.4	Discussion	111

5	Continuous trait associations	114
5.1	Introduction	116
5.2	Methods	116
5.2.1	Models of sequence evolution	116
5.2.2	Model of codon sequence evolution	117
5.2.3	Model of protein sequence evolution	117
5.2.4	Sigmoid function used to link phenotypic value and amino acid preference	118
5.2.5	Sparse parameter estimation at each site	118
5.2.6	Likelihood ratio test in model C	118
5.2.7	Benchmark simulations	120
5.3	Results	122
5.3.1	Simulations	122
5.3.2	Longevity in mammals	123
5.4	Conclusion	126
5.5	Acknowledgements	126
5.6	Supplementary material	126
5.7	Additional remarks and discussion	126
6	Gene-level predictions	129
6.1	How many positive sites are there within each gene ?	131
6.2	Which genes are most likely to be associated to the phenotype ?	132
6.2.1	Wilkinson's method	132
6.2.2	Fisher's method and derivatives	133
6.2.3	A mixture model for the distribution of p-values	137
6.2.4	Influenza strikes back	140
6.3	How many genes are associated to a given phenotype ?	143
6.4	Detection of genotype-phenotype associations in the Orthomam database	146
6.4.1	Echolocation	147
6.4.2	Diet	151
6.4.3	Aquatic and marine	153
6.4.4	Subterranean	156
6.4.5	Other phenotypes	159
7	Relevance and limitations of the approach	163
7.1	What problem are we solving ?	164
7.2	On the interpretation of Pelican results	165
7.2.1	Pelican identifies sites whose evolution <i>correlates</i> to variations of the phenotype	165
7.2.2	On the interpretation of estimated parameters	166
7.2.3	On the choice of the phenotypic trait	166
7.2.4	The issue of calibration	167
7.3	On input data, and uncertainties	168
7.3.1	Uncertainties in the sequences, and their use as representations of the genotype of species	168
7.3.2	Uncertainties in the alignment	169
7.3.3	Uncertainties in phylogenetic trees	169
7.3.4	Uncertainties in the phenotype	170

7.4	Hypotheses of the Pelican model, and their biological implications	171
7.4.1	Adaptation happens through coding sequences modifications	171
7.4.2	Sites evolve independently from each other	171
7.4.3	Similar traits involve similar genes across species	172
7.4.4	Modeling substitutions at the level of amino acids	172
7.5	Perspectives	173
	Bibliography	175
	Appendices	191
A	Summarized remarks on the usage of Pelican	192
B	Empirical phylogenies	194
C	Evaluation of methods: supplementary material	200
C.1	Synthetic trees	201
C.2	Study of the calibration of the methods	204
C.3	Benchmark results with confounding factors	205
C.4	Evaluation of Pelican using different degrees of freedom in the LRT	207
D	Gene aggregation	209
D.1	Null distribution for the Gene-wise Truncated Fisher (GTF) method	210
D.2	Mixture model: derivation of EM equations	211
D.2.1	Expectation step	211
D.2.2	Maximisation step	213
E	Scans of Orthomam for genotype-phenotype associations	214
E.1	Echolocation	215
E.2	Diet	217
E.3	Adaptation to life in aquatic environments	219
E.4	Adaptation to life in marine environments	221
E.5	Adaptation to life in subterranean environments	223
E.6	Diurnality and nocturnality	225
E.7	Vocal learning	227
E.8	Domestication	229
F	Other work	231

Acronyms

AUC area under the curve. 35, 37, 46, 58, 60, 62–64, 68, 132, 134

CDF cumulative distribution function. 98

ECDF empirical cumulative distribution function. 98, 134

EM expectation-maximisation. 48, 138

FDR false discovery rate. 97, 129, 131, 143, 144, 146, 161, 167

gBGC GC-biased gene conversion. 16, 20, 31, 32, 37, 55, 64, 65, 67, 69, 70, 72, 74, 165, 172

GO gene ontology. 146, 148, 158–160, 162

GTF Genewise Truncated Fisher. 140, 144, 146, 161, 210

GTR general time-reversible. 27, 71, 80, 81, 91

GWAS genome-wide association studies. 38, 40, 50, 58

LRT likelihood ratio test. 43–45, 54, 55, 57, 58, 64, 69–71, 82, 83, 91, 95, 110, 113, 139, 140

MCEM Monte-Carlo expectation-maximisation. 49

MCMC Markov-chain Monte-Carlo. 49

MLE maximum likelihood estimate. 90

PPS persistent positive selection. 31, 64, 65, 68, 69, 74

Introduction

Phenotypic variation is a cornerstone of the evolution of species. Since Darwin's ideas on "descent with modification", it is apprehended as both the consequence of the evolutionary process, and the object of natural selection. With the advent of molecular biology throughout the 20th century, the molecular basis of heredity has been better understood as the notion of genotype was materialized from an abstract notion of inheritance, through the identification of DNA as the support of heredity. With time and efforts, our understanding of the structure of DNA improved, first uncovering its fragmentation into chromosomes, then the existence within them of genes: sub-sequences of DNA that encode amino acid sequences, and are translated into the primary effectors of biological functions called proteins¹. This led to the establishment of the central hypothesis of molecular biology, describing the tight one-way relationship between nucleic acid sequences and protein sequences.

Understanding the relation between genotype and phenotype has continued to be, and still is, a major challenge for life sciences. It is indeed a complex task, for many reasons. The manifestation of the potentiality encoded in the genotype into biological structures does not happen in a vacuum: phenotypic traits are shaped by the interaction of the genotype with the environment, at many levels. The expression of genes is regulated, in interaction with other genes and in response to the environment, thus varying through tissues and developmental stages of an organism, and throughout the rest of its life. Proteins, the product of gene expression and the main agents of biological functions, also depend on their immediate environment to properly fold and exert a structural or chemical role. Moreover, organisms are intricate, complex systems, in which biological structures, be they molecules, cells, tissue or organs, inter-operate at multiple levels to develop and maintain life functions. The idea that the genome is akin to a textual code that would define all the organization and properties of biological structures is thus misleading: it merely holds the information necessary to their reproduction².

All in all, the complexity of the processes leading to the emergence of higher level phenotypes makes it difficult to establish a mechanistic relationship between the genotypic information and phenotypic traits. We can however resort to correlative approaches, that do not immediately attempt to determine a causal chain between the genotype and phenotype. They focus instead on the identification of genes, or variations within genes, that may be linked to a possibly complex phenotype; understanding the nature of the putative relationships is then left to further investigation. We generally can not directly observe evolutionary events unfolding, but rather work with instantaneous snapshots of their end product, observable on extant species. Understanding the association between the evolution of genes and phenotypic traits at the level of species thus requires some form of

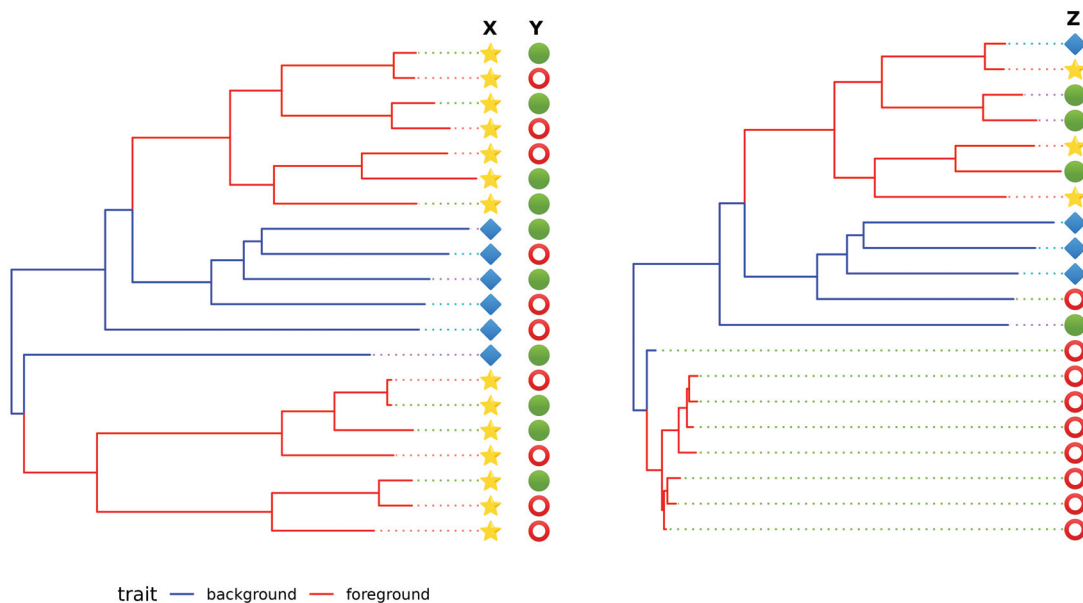
¹Although proteins are not the only functional agents of life: RNA can also have a functional role, in ribosomes for example.

²This is translated from an eloquent formula from [Morange, 2000]: "L'ADN n'est pas le support d'une information qui définirait la structure des composants élémentaires du vivant: il est simplement l'information qui permet aux cellules de reproduire la structure de ces composants."

inference of past events, where the phylogenetic relationships between species come into play. The present work focuses on detecting traces that remain within gene sequences from their evolutionary history, to better understand their relation with a given phenotype. The term phenotype is used in a broad sense, here and throughout this manuscript: it encompasses morphological and metabolic traits, life-history traits such as longevity, or the ability to live in a particular environment. The underlying assumption is that when phenotypic variations are the result of changes in the selective pressure that is exerted, they may leave traces that are observable in extant genomes.

Using comparative approaches, we take advantage from observations of homologous sequences somewhat considered as so many statistical replicates, to identify evolutionary trends among coding sites that correlate with some phenotypic trait. However, unlike experimental replicates, observations of species and their genomes are not independent, due to their common evolutionary history: homology both enables comparative analyses, and is a major confounding factor that must be accounted for in their application. Gene sequences are the product of evolution as they are both the result and the subject of an ongoing process, that involves a large diversity of events such as point mutations, insertions, deletions, duplications, loss, recombination, horizontal transfers... Among all the modifications that a gene can go through on its evolutionary path, this work focuses on punctual changes in the sequence at the level of species, i.e. *substitutions*, denoting the result of an individual mutation spreading through a population until it becomes fixed as the only remaining variant.

I propose to materialize the problem at hand through simple illustrations of coding sites, which we aim to determine whether or not they are possibly associated with a phenotype. Situations depicted are kept voluntarily abstract, with arbitrary trees and phenotype annotations. Sequence alphabet is depicted using symbols, that we can freely choose to represent nucleotides, codons, or amino acids.



(a) (X) shows patterns for association with the trait; (b) Association of (Z) to the trait is more
the composition of (Y) seems to be independent from it. ambiguous.

Figure 1: Panel (a) shows examples of sites that can easily be identified as associated or not to the trait, while panel (b) shows a more ambiguous site, where its association to the phenotype is harder to evaluate.

Figure 1a depicts example sites, X and Y , attached to a phylogeny of species with annotations of a phenotypic trait — we assume here that the history of the trait to be known with certainty. From the observation of each site composition, compared to the phenotypic condition, it seems likely that the site X is associated to the phenotypic trait, while Y is likely not. We may postulate that the composition of site X results from substitutions from an ancestral state \blacklozenge , that is conserved among lineages having the background trait, towards a new state \blackstar coincidentally to the presence of the foreground trait. On the other hand, the composition of Y seems independent from the phenotypic trait: both \circ and \bullet substitutions are observed across conditions, in a seemingly random fashion, suggesting that the substitution dynamics between alleles is unrelated to this particular phenotype.

On the other hand, figure 1b displays a situation for which it is much more difficult to assess whether the site bears traces of a history associated to the phenotypic trait. For example, a large number of species displaying the red (foreground) phenotype have genotypic state \circ , but it is also present among some blue (background) species. Moreover, evolutionary distances are very short within the sub-tree that carries most of these alleles, and should be accounted for: a scenario where only one substitution towards \circ occurred early by random chance and remained in this state is perfectly plausible. The complexity of the site makes it much more difficult to conclude on the matter.

How likely is it that the substitution dynamics in these examples depends on the phenotypic trait? Just by looking at them, one could make qualitative answers in obvious situations, but not in general. This illustrates the need for more formal approaches, for two reasons: (1) the necessity to produce a quantitative assessment on the likelihood that a site has a substitution history related to the phenotype, so that we can accurately analyze more complex situations; (2) to enable automating the analysis so that it is applicable on large scale datasets of aligned sequences, as relying on human evaluation to scan even the few hundreds of positions in one gene family alignment would be laborious, not to mention the millions of sites that compose genome scale datasets.

For this purpose, formalization of the analysis can be achieved by using probabilistic models of the processes that would be relevant to explain the data, combined with statistical inference to estimate parameter values that fit the observations. In our case, it is the evolutionary process underlying substitution events that ought to be represented. An actual model could be strongly *mechanistic*, and incorporate with great precision theoretical elements from the current knowledge on the theory of evolution, or rather *phenomenological*, relying on an empirical explanation of the data without necessarily being anchored in the theory; it may as well stand anywhere in-between. Mechanistic models thus have the benefit that variables and parameters typically have direct biological interpretation. On the other hand, more phenomenological ones are typically less parameter-heavy³, thus making them easier to apply, at the cost of renouncing to some extent the direct equivalence between parameters and biological concepts. The choice of the model formulation depends on the question that is asked, as well as computational or statistical constraints, as fitting complicated models typically requires larger data samples and more computing power. In this thesis, we explore the use of models that lie at different positions in this spectrum, and evaluate their predictive abilities while accounting for their computational cost.

Comparative approaches have been widely used to investigate the causes and effects of adaptation, and search for associations between the evolution of genes and phenotypic traits. Studies have highlighted the degeneracy of genes after environmental transitions, as relaxation of the selective pressure permitted loss of their functions. For example, transitions from surface to underground

³Although neural networks are a good counter-example: they are phenomenological by essence, but commonly involve thousands of parameters.

habitats in mammals have been found to be systematically accompanied by increased substitution rate in vision-related genes, suggesting relaxed selective constraints [Partha et al., 2017]. Similar results have been obtained on cave-dwelling freshwater crustaceans, and supported by the significantly reduced expression of pigmentation and vision-related genes [Lefébure et al., 2017]. Adaptive evolution also typically occurs in such situations, leading to the emergence of specialized traits that contribute to the fitness of the species in a new environment (e.g [Chikina et al., 2016]). Host-switching events among viruses entail changes in selective pressure [Tamuri et al., 2009], and traces of persistent positive selection can be found in the viral genome, especially in genes having critical role in pathogen-host interactions [Hou et al., 2022]. Convergent amino acid substitutions have also been reported in relation to complex phenotypic traits, such as echolocation that was found to be associated to independent substitutions in the sequence of the Prestin protein expressed in the auditory system, across distinct mammalian clades [Liu et al., 2010]. Another example of convergence is the independent emergence of the C4 metabolic pathway many times within the plant kingdom: it has involved molecular adaptations in multiple enzymes [Besnard et al., 2009, Kapralov et al., 2012], and led to increased photosynthesis efficiency in warm and arid environments. The predominance of convergent adaptations among these few examples is no coincidence. Convergence cases are particularly useful to understand the evolution of species, as they often provide an unambiguous signal for adaptation: the repeated observation of independent but similar genomic changes on the same locus, correlated to the presence of a phenotypic trait, suggests these changes were promoted by directional selection⁴.

Across the variety of methods that have been employed to identify substitutions associated to a phenotypic trait, there is currently a lack of satisfactory solutions to scan genome scale datasets in a reasonable time. Previously published literature establishes that codon-level models of mutation-selection are performing well at this task [Rey et al., 2019], but are computationally expensive; a comparison to so-called ω models is currently lacking. This calls for novel approaches that offer a good trade-off between high throughput and reliability of predictions and enable conducting analyses at the genome scale, and should be properly evaluated and compared to existing ones. Because non-adaptive processes such as CpG hypermutability or GC-biased gene conversion (gBGC) do leave perceptible traces in genome sequences that can be misleading, the robustness of detection methods to such confounding factors should also be evaluated. Although the initial scope of this work is to investigate methods for the detection of sites within genes that are associated to a phenotype, we are also interested in gene-level predictions for phenotype association from site-level results. Altogether, the questions that motivate such methodological developments are: which sites in genome-wide alignments have evolved in relation to a given phenotype? Among them, how did the evolutionary dynamics change concomitantly to variation of the trait? Can we distinguish changes in its direction? And which genes carry a differential signature between traits? The purpose of this work is to propose methodological tools that help answering these questions, and are applicable at the scale of genomes.

As an opening matter, I describe in [chapter one](#) the methodological framework in which this work is conducted. This is motivated by the need we have to evaluate methods that we investigate to perform the detection of genotype-phenotype associations: this requires to have access to a “ground truth”, that can be obtained by relying on simulations. This chapter thus focuses on the simulation model that was chosen, and the measurement of detection performance to be compared across methods.

⁴Note however that convergent adaptation is a special case of positive selection towards the same direction, and the two notions should not be equated. Successful attempts have been made to distinguish positive selection from convergent adaptation, e.g in the instance of the molecular evolution of Rubisco (a key enzyme in photosynthesis) towards C4 metabolism [Parto and Lartillot, 2018].

I then report in [chapter two](#) our initial attempts using linear mixed models, to identify sites associated with a binary trait based on their composition. These models lay far on the phenomenological side, as they do not model substitution events at all, but rather look for correlations between site composition and phenotypic trait, while somewhat accounting for the phylogenetic relationship between observations.

At this point, we suspect that these simple approaches are lacking, and turn to exploring the use of phylogenetic substitution models. In [chapter three](#), an evaluation of several methods is presented, that encompass a linear mixed model, amino acid substitution models, a mutation-selection model, and a model of codon substitution rates. Using simulations conducted on multiple empirical and synthetic phylogenies, we show that our implementation of a previously published model of amino acid substitutions [[Tamuri et al., 2009](#)] has good performance, comparable to more complex models operating at the level codons. It is also faster than the other phylogenetic methods that were evaluated, and offers what seems to be a good trade-off between speed and accuracy for the purpose of detecting genotype-phenotype associations. The content of this chapter has been accepted for publication in *Molecular Biology and Evolution* (MBE).

On the basis of these findings, more efforts were dedicated to improving this implementation, that we named Pelican, as well as identifying and understanding its limitations. [Chapter four](#) is an in-depth presentation of the model and implementation of Pelican, that details the underlying substitution model and the procedure by which hypothesis testing is performed, as well as statistical limitations with a particular focus on the calibration of the test. I also discuss a particular case that was encountered during our benchmark experiments on the detection of relaxed selection, and propose workarounds to tackle this specific case.

An extension of Pelican to handle continuous phenotypic traits is described in [chapter five](#). The search for associations between genotypes and continuous traits is typically done by binning the trait values into discrete categories, somewhat arbitrarily. We propose a model that does not require such discretization of the trait and enables the analysis of a continuous trait as-is. Using simulated datasets against several discretization strategies, we report that this model is an improvement over discretization approaches. We also scan several mammalian genes that were found to be associated with longevity, and compare our predictions to published results.

All variations of Pelican operate at the site level, and produce site-wise predictions. While this level of information is already useful, we strive to exploit the set of predictions made within each gene to provide diagnostics at the level of genes, and enable the identification of candidate genes most likely to be associated to a phenotype. We explore in [chapter six](#) different ways to aggregate the site-wise predictions resulting from Pelican runs into a gene-wise score, so that a list of best gene candidates can be established. We apply one of the method we designed to finally conduct analyses on an empirical dataset, Orthomam, a database of coding sequence alignments from 116 mammalian species. Scans for associations were performed against several phenotypes, producing site- and gene-level predictions for each of them, that we review and confront to both the literature and our expectations.

Finally, this work is concluded in [chapter seven](#) with a discussion on limitations and critical points that one should be aware of when using Pelican. In particular, I discuss of the general biological interpretation of results obtained with Pelican, and some pitfalls that should be watched out for. Some underlying hypotheses of the model are also made explicit, which is an opportunity to identify the kind of associations that can be detected, and those that can not. I then propose several perspectives to further this work, that consist either in improvements on the implementation of the current model, or extensions of the model to broaden its application scope as well as hopefully

improve its throughput and accuracy.

Note: a study we published during my time as a PhD student, but that is unrelated to the subject of this work since it focuses on the epidemiology of Sars-CoV-2 in early 2020, is also included in [appendix F](#).

Chapter 1

Framework for the evaluation of detection methods

Since we aim to develop methods to scan for sites under differential pressure, we need to determine and quantify how accurate they are. A perfect method should correctly distinguish every “positive” site that underwent selective pressure change coincidentally to phenotypic variation, from “negative” sites whose evolution is independent from the phenotype. Determining the ability to discriminate between the two requires prior knowledge of the composition of the two sets, so that predictions made by the method can be confronted to a ground truth. It is possible to achieve this using real data, e.g. from already published analyses, although this is quite inconvenient: such datasets are often limited in size, are not always readily available, and there is generally some uncertainty on the results that are published. Comparing, and eventually corroborating results obtained on real world data is certainly informative and an important step in the validation of methodological developments — and we do rely on this to validate some of our approaches — but it is not practical for systematic evaluation of performances. An other way is to use a model to simulate synthetic alignments, so that the generative process for each site is known, since it is controlled by the model parameters. The particular model that is used in the simulation can be arbitrary, although it is typically desired to be as realistic as possible. To that end, we use a simulation model derived from the mutation-selection (MutSel) framework, a rather mechanistic modeling approach where most parameters are identifiable to biological processes.

As you may be already familiar with Markov chains, the Wright-Fisher model or mutation-selection models, feel free to skip directly to section 1.3.3 that exposes specific extensions of the mutation-selection models we used in some of our simulations. Section 1.4 follows, describing our methodology for measuring detection performances.

The first part of this chapter (section 1.1) focuses on Markovian processes, and particularly continuous-time Markov chains, to expose the modeling framework that we use for simulating, as well as for inference throughout this work. They are the mathematical foundation of many models of sequence evolution, among which are mutation-selection models, as well as most phylogenetic models that are discussed in this thesis.

After these mathematical notions have been exposed, I describe in section 1.2 the essential biological components of the substitution process that are modeled in the MutSel framework. I present a short historical overview of the research that led to their identification and a better understanding of their interactions. A description of mutation-selection models is then provided,

building on these supporting elements, as well as extensions of the base model to include additional biological processes, namely GC-biased gene conversion and CpG hypermutability. A procedure to simulate persistent positive selection is also presented.

In this work, we define the performance of a method by its ability to make accurate predictions, and measure it using two metrics, precision and recall. I describe in the second part of this chapter (section 1.4) the methodology we use to compute a score that integrates these metrics, and associate a confidence interval to it.

The MutSel simulator and the estimation of precision-recall scores make up the two essential tools that we use in our search for methods for detecting differential selection correlated with a phenotypic variation.

Contents

1.1 Modeling the evolution of sequences using continuous-time Markov chains	20
1.2 The substitution process at the level of populations	22
1.2.1 The mechanism of substitution in population genetics	23
1.3 Simulating alignments in the mutation-selection framework	25
1.3.1 A mutation-selection model to simulate coding sequence alignments	26
1.3.2 Simulations	29
1.3.3 Extensions of the simulator	31
1.3.4 Empirical phylogenies	33
1.4 Measuring the detection performance of a statistical method	33
1.5 Evaluation pipeline: technical aspects	36

1.1 Modeling the evolution of sequences using continuous-time Markov chains

Modelling sequence evolution is typically done using a Markovian process that describes the dynamics of substitutions between states (e.g. nucleotides or amino acids) along a phylogeny. Markov chains are used to model stochastic processes as a set of possible states of a system, and the probabilities of transition between each pair of states. Given an initial distribution of states, a Markov process can be unfolded through time, either in a discrete step-by-step manner, or continuously, to determine the probabilities for future states of the system. A preeminent characteristic of Markov processes is that they are memory-less: future states only depend on the present one, and the complete history of previous states does not have to be known to make predictions. Because the evolution of sequences is best modelled as a continuous process through time, only continuous-time Markov chains is presented in this section.

In this framework, the components of a model of sequence evolution are a discrete state space, that represents the sequence alphabet, and a *transition rate matrix* Q . It is a square matrix having the same dimension as the state space, that holds the instantaneous rates of transitions that can occur between each state. The diagonal terms are minus the total rate of change away from each state, so that the sum of terms in each row is equal to 0. A diagonal term q_{ii} thus represents the rate of departure from state i . Equation 1.1 illustrates the general form of rate matrices, for a state space of dimension n , where q_{ij} is the instantaneous transition rate from state i to j . It is oriented with initial states as lines, and transition states as columns. This matrix defines a substitution model,

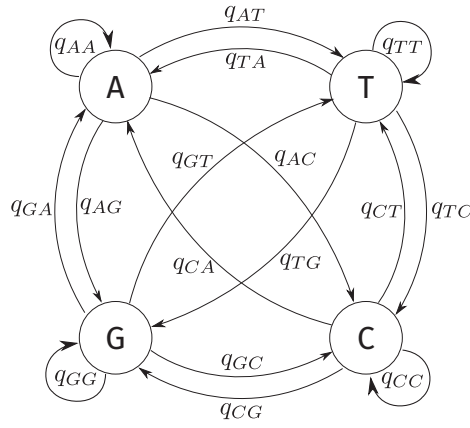


Figure 1.1: Representation of a nucleotide substitution model as a graph. The state alphabet is the set of possible nucleotides: A, T, C or G; transitions between states occur at different instantaneous rates q , which are generally gathered in a matrix.

and can also be represented as an oriented graph, such as depicted in figure 1.1 for the nucleotide alphabet.

$$Q = \begin{matrix} & \begin{matrix} 1 & 2 & \dots & n \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ \vdots \\ n \end{matrix} & \begin{bmatrix} -\sum_j q_{1j} & q_{12} & & q_{1n} \\ q_{21} & -\sum_j q_{2j} & & q_{2n} \\ & & \ddots & \\ q_{n1} & & & -\sum_j q_{nj} \end{bmatrix} \end{matrix} \quad (1.1)$$

From the rate matrix, a stochastic matrix or transition probability matrix $P(t)$ is derived to obtain time-dependent probabilities of changes between states. It is the solution to the differential equation $P'(t) = P(t)Q$ with respect to t

$$P(t) = e^{Qt} \quad (1.2)$$

where $p_{ij} = \mathbb{P}[x(t) = j | x(0) = i]$ is the probability of transition from state i to state j after a time t . Each row describes all the possible transitions from one state, including the probability to remain in that same state, hence the sum of probabilities at each row is 1

$$\sum_j p_{ij} = 1 \quad (1.3)$$

The probability distribution of states $\lambda(t)$ after a time t , is given by a vector-matrix product of $P(t)$ and an initial distribution $\lambda(0)$. $\lambda(t)$ is a vector of probabilities, with the dimension of the state space, holding the probability for each state at time t . The sum of its components $\sum \lambda(t) = 1$ for any t .

$$\lambda(t) = \lambda(0)P(t) \quad (1.4)$$

In the context of sequence evolution, this has a direct application in finding the probability distribution of states $\lambda(t)$ at the tip of a branch with length t , given the distribution $\lambda(0)$ at the origin of the branch.

Markov chains are memory-less processes, meaning that the future states of the system that is modelled only depend on the current state at any point. This implies that predictions of state

distribution can be made on each node, only knowing the distribution on the upstream node, without the need to look further up towards the root of the tree. It also makes possible to stop and restart the process at any point, and compute the corresponding state distribution

$$\begin{aligned}\lambda(u + v) &= \lambda(0)P(u + v) \\ &= \lambda(0)e^{Q^u}e^{Q^v} \\ &= \lambda(0)P(u)P(v) \\ &= \lambda(u)P(v)\end{aligned}$$

A state probability distribution that is unchanged by the application of the stochastic matrix, is known as a *stationary distribution*, defined as $\pi = \pi P(t)$. Some evolution models assume the process is stationary, and are parameterized using a stationary distribution which determines a general direction for transitions — an example is given in section 4.2.1. A Markov process is called time-reversible when the direction of the change between two states is identical, so that

$$\pi_i p_{ij} = \pi_j p_{ji} \tag{1.5}$$

It is a convenient assumption for computing the transition probability matrix, as the rate matrix of a time-reversible process has real eigenvalues and eigenvectors, a property that makes its exponentiation easier. An application of this is presented in section 4.3.3.

The continuous-time Markov chain framework is applicable to model the evolution of both nucleotide and amino acid sequences, with adjustments to the dimensions of the rate and probability matrices. Its realism is however limited for this purpose, as it is not well suited to model the evolution of sequences as a whole, but rather model each site independently — although some parameters may be shared across sites.

1.2 The substitution process at the level of populations

Although they are often reduced to be considered as point events in large-scale evolutionary models, substitutions are by essence the result of a process unfolding within populations, that involves multiple components. Figure 1.2 depicts a simplified overview of the substitution process within a population, without giving much details on the forces at play for now.

When a mutation occurs within a population, the long-term fate of the mutant allele is either to reach 100% frequency and become fixed within the population, or to completely disappear¹. The diffusion of the mutant allele partly depends on the reproductive success of individuals that carry it. The characterization of mutations as deleterious, neutral, or advantageous has thus to be understood as a difference in reproductive success between the new allele and the non-mutated allele, i.e. their relative fitness. Allele frequencies are also subject to stochastic variations between generations through genetic drift, a process that may be responsible for the random fixation or loss of alleles in the population, at the condition that they are not too strongly deleterious. Fixation and loss of alleles are both absorbing states, in that once they have occurred, the frequency of allele remains constant at 0 or 100%. However, new mutations can still arise at the same locus, and reintroduce variability in the population eventually leading to other substitutions.

¹Long-term polymorphism can still be maintained e.g. in the case of frequency dependent selection which favors low-frequency alleles and penalizes high-frequency alleles.

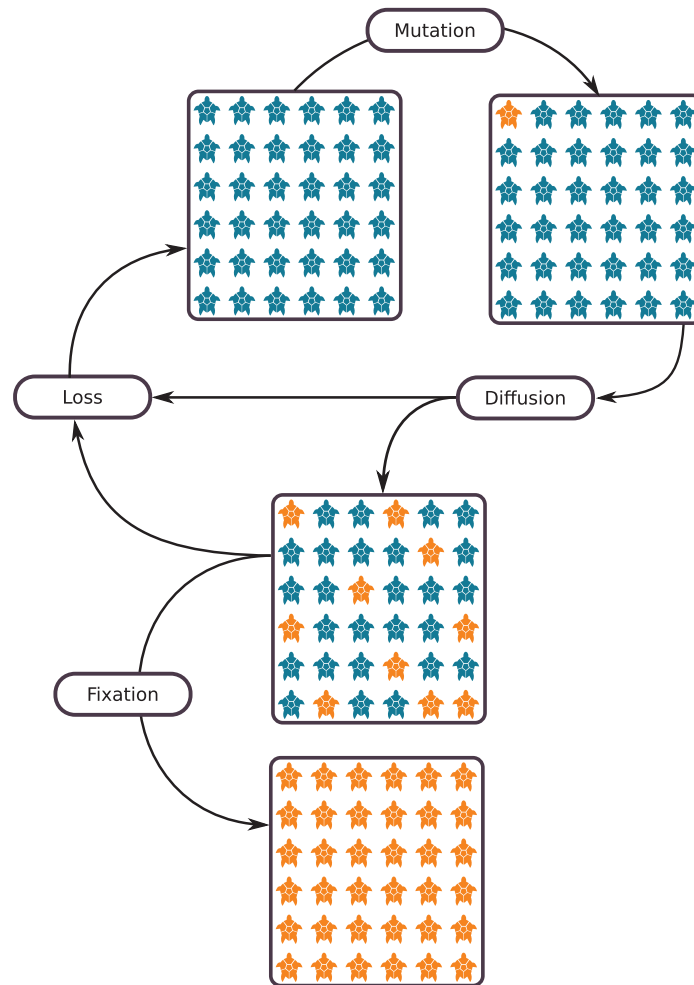


Figure 1.2: Schematic depiction of the stages in the process of allele substitution within a population. Arrows symbolize the passing of at least one generation. An initial population where all individuals share the same ancestral allele (blue turtles) is considered. From this state, a mutation may occur and introduce a new (orange) variant. The frequency of this variant then fluctuates across generations, under the combined effects of random drift and selective forces. The ultimate fate of the allele is either to be entirely removed from the population, or to become fixed in the population as the only remaining variant.

Selective forces, actualized as fitness differentials, and genetic drift, responsible for random variations of allele frequencies, are the primary forces² that determine the diffusion dynamics of alleles within populations, and ultimately the occurrence of substitutions at larger time scales. The following sections introduce in further details classical models from population genetics, that describe the diffusion of alleles in populations using these two components. I hope it might be helpful, both in establishing the grounds on which substitution models are built, and exploring the interplay between selection and genetic drift from a historical perspective.

1.2.1 The mechanism of substitution in population genetics

Starting early in the 20th century, the combined works of Wright, Fisher and Haldane have been foundational to the establishment of population genetics and, later on, of molecular evolution. Fisher

²Other processes may also alter the diffusion of alleles; they are left out for now, but some of them will be discussed later on.

proposed a mathematical model for the variation of allele frequencies within a population, assuming Mendelian segregation of alleles at each generation [Fisher, 1922]. The population is considered almost “ideal” in the sense of Hardy-Weinberg: it is infinite in size, with random mating and non-overlapping generations; furthermore, mutation or migration events are non-existent. Fisher’s model mimicked the effect of selection — which is null in Hardy-Weinberg populations — on the variation of allele frequencies, by assuming that the genotype of diploid individuals determined their survival until maturity. Depending on the relative advantage that each genotype had on the probability of survival, the steady state for the system could be determined, either leading to the fixation of one allele, or the coexistence of the two alleles in the form of three possible genotypes (homozygous AA , aa , or heterozygous Aa).

Using this model, Haldane derived the probability that an advantageous mutant allele is sustained within the population. He showed that, “after only a single appearance in an adult zygote”, the mutant allele has probability $2s$ to reach fixation, where s is a value denoting a selective advantage [Haldane, 1927]. Haldane defines $(1 + s)$ as the mean expected number of descendant for each individual that carry the allele and survive to produce a new one.

Wright then adapted the model to consider finite population sizes, with the number N of individuals ($2N$ alleles for diploids) remaining constant between generations [Wright, 1931]. The probability of fixation within a diploid population for a single advantageous mutation, derived in [Haldane, 1927] to be equal $2s$, is thus generalized to give $2s/(1 - e^{-4Ns})$. For large values of N , the probability is approximately $2s$, consistently with the result obtained by Haldane for infinite populations. This modification had a major consequence on the prediction of allele frequencies: in the absence of selection, Fisher’s model was at the Hardy-Weinberg equilibrium and the frequencies were constant through time; whereas considering a finite population induced random fluctuations of allele frequencies between generations, that were not caused by selective effects. This stochastic process, known as genetic drift, eventually leads to the fixation of one of the alleles in the population. This model, describing the fluctuation of allele frequencies in the absence of selection, was later named the Wright-Fisher model.

In his article, Wright mentioned that deviations from the hypotheses made on the population were likely to occur, such as non random mating or unbalanced sex-ratio. He thereby introduced the idea of effective population size, representing the number of breeding individuals in an idealized population that would behave like the real population under consideration. A direct implication is that the effective population number N_e is commonly lower than the census population, so that the importance of random drift relative to selection is higher than it was initially thought to be. This was foundational to the development of the neutral theory of evolution, of which Motoo Kimura was one of the pioneer. Regarding the probability of fixation for a single mutation, Kimura refined Wright’s equation to be $(1 - e^{-2s})/(1 - e^{-4N_e s})$ [Kimura, 1962]. When $|s|$ is small, Wright’s formula gives a good approximation of this expression, which thus remains in conceptual agreement with the previous results.

The combined efforts of Wright, Haldane, Fisher, and then Kimura greatly contributed to the field of population genetics³, and the development of the Modern Synthesis of the theory of evolution, by bridging the gap between Darwinian selection and Mendelian laws of inheritance using mathematical modelling.

³And many others, whose efforts and contributions I am unfortunately not able to credit without dedicating an unreasonable amount of time to it. This historical overview is meant to be concise, but was informed by reviews [Crow, 1987, Patwa and Wahl, 2008, Bacaër, 2011] on the subject which can be referred to for more details.

1.3 Simulating alignments in the mutation-selection framework

As described in the previous sections in this chapter, the probability of fixation is a balance between the effect of selection depending on the relative fitness s of the mutant, and genetic drift that depends on population parameters that are captured in N_e , the effective population size. The core concept of mutation-selection models is to describe the substitution process as two components, as established in the Wright-Fisher model: random mutations events that occur at the level of individuals, and the probability that they become fixed in the population (see [Teufel et al., 2018] for a review). Codon-level modeling allows for accurate representation of mutations that occur at the level of nucleotides, distinguishing synonymous from non-synonymous substitutions, and comparing fitnesses between amino acids in the latter case. Other factors influencing the probability of fixation can also be accounted for, examples of which are given in the following sections.

The general definition of MutSel models is in the form of Markovian substitution processes, where the transition rates between codons are defined as

$$\begin{cases} q_{xy}^i = \sigma \times \mu_{xy} \times \mathbb{P}_{fix}^i(x, y) & x \neq y \\ q_{xx}^i = - \sum_{y \neq x} q_{xy}^i \end{cases} \quad (1.6)$$

where μ_{xy} is the instantaneous rate of mutation from codon x to y , and $\mathbb{P}_{fix}(x, y)$ is the probability for its fixation. The σ constant is often incorporated to scale the rates, allowing to adjust the measurement unit. This gives the rate matrix Q , with dimension 64×64 when all codon transitions are represented, or 61×61 if stop codons are removed from the alphabet as nonsense or nonstop mutations typically result in nonfunctional proteins.

The mutation rate between codons is typically derived from a mutational process modelled at the nucleotide level. This process can be defined using classical models of nucleotide mutations, like e.g. HKY85 [Hasegawa et al., 1985] in [Halpern and Bruno, 1998, Yang and Nielsen, 2008, Tamuri et al., 2012]. In the simplest case, mutations only occur between neighboring codons, that is codons that differ from only one nucleotide, and replacement rates between single nucleotides can be directly incorporated. Mutations between non-neighboring codons can also be modeled using probabilities of nucleotide mutation at each position [Halpern and Bruno, 1998, Tamuri et al., 2012].

$$\mu_{xy} \propto \prod_{j=1}^3 \mu_{x_j y_j} \quad (1.7)$$

The probability of fixation $\mathbb{P}_{fix}(x, y)$ is typically defined as a function of the relative fitness difference between ancestral codon x and mutant codon y , and the effective population size N_e , as established in the population genetics theory [Wright, 1931, Kimura, 1962]. The effective population size is generally assumed constant across lineages, although branch-specific values for N_e could be represented as in [Nielsen et al., 2006]. Importantly, the corresponding parameter in mutation-selection models is often the effective *chromosomal number* N , allowing to generalize the equation to any ploidy. Although the actual specification varies across models, sometimes incorporating other factors, the general definition is

$$\mathbb{P}_{fix}(x, y) = \frac{1 - e^{-2s}}{1 - e^{-2N_s}} \approx \frac{2s}{1 - e^{-2N_s}} \quad \text{when } |s| \text{ is small} \quad (1.8)$$

where s is the relative fitness difference between codons $f(y) - f(x)$.

Finally, fixation times are typically assumed to be negligible compared to the time between mutations, so that polymorphism caused by new mutations during the diffusion of a mutant allele in the population can be neglected.

As established by population genetic theory, the fixation probability is influenced by the competing forces of genetic drift and natural selection, the relative effect of which depending on the effective population size N_e . A large effective population size makes the relative fitness of a mutation the predominant factor in its fixation probability. On the other hand, random fixation of alleles independently of their relative fitnesses through genetic drift is dominating when the effective population size is low.

In this way, small fitness differences between alleles have little influence on the fixation probability, while mutations inducing marked reductions in fitness are very unlikely to be fixed. Mutations may also be beneficial when the new allele has greater fitness, increasing the chance that it becomes fixed in the population. As a result, slightly deleterious mutations are allowed to be fixed in the population, while fixation of lethal or strongly disadvantageous alleles is rare; positive selection can also occur, raising the probability of fixation of favored alleles.

The first model of this class was proposed in [Halpern and Bruno, 1998], to perform an inference of evolutionary distances between aligned sequences. The model does not include explicit parameters for N_e and s , instead deriving the probabilities of fixation from its definition in [Kimura, 1962] to be a function of mutation probabilities between codons and their equilibrium frequencies. [Yang and Nielsen, 2008] is an example of extension of the base MutSel model that distinguishes synonymous from non-synonymous codon substitutions. It was used to investigate the effect of selection on codon usage, by modeling fitness differences at the level of codons. The instantaneous rates for non-synonymous substitutions are scaled by a positive parameter ω^4 , to represent selection at the protein level. The value of ω have been found to be increased when “the substitution process is not dominated by selection or drift, but admits interplay between the two” [Jones et al., 2016], using a mutation-selection model to highlight this phenomenon they named shifting balance. Other mutation-selection models have been proposed to estimate amino acid fitness distributions [Rodrigue et al., 2010], that can be contrasted between phenotypic traits, e.g. different hosts species for the Influenza virus [Tamuri et al., 2012] or C3/C4 photosynthesis [Parto and Lartillot, 2017, Parto and Lartillot, 2018]. These models introduce new approaches to investigate selective effects in coding sequences⁵, that differ from the widespread ω -based framework by modeling individual amino acid fitnesses, and have clearer population genetics interpretation.

1.3.1 A mutation-selection model to simulate coding sequence alignments

The model we use to simulate alignments, including sites under differential selection, belongs to the latter sub-family of mutation-selection models and involves a representation of amino acid fitness profiles. It is the same as in [Rey et al., 2019], with some extensions that I describe in section 1.3.3. Substitutions between codons are represented as a site-independent process, by defining a 61×61

⁴ ω is a central parameter in codon models that investigate selective effects through the comparison of rates of synonymous and non-synonymous substitutions ($\omega = d_N/d_S$). This class of model precedes the MutSel framework, and is explained with more details in the introduction of chapter three.

⁵This kind of applications for mutation-selection models was already foreseen by [Halpern and Bruno, 1998]:

“Without denying the existence of rate variation due to positive selection (e.g., as result of immune pressure on an antigenic region of a pathogen) or purifying selection at levels other than the amino acid level (e.g., RNA secondary structure, same-residue codon bias, etc.), it would be interesting to see the extent to which observed rate heterogeneity in coding regions could be reduced to variation in the strength of residue preferences as indicated by their frequencies.”

matrix Q^i of transition rates at each site i . Transitions involving any of the three stop codons are not allowed, by removing them from the alphabet of states altogether.

$$\begin{cases} q_{xy}^i = \mu_{xy} \times \mathbb{P}_{fix}^i(x, y) & x \neq y \\ q_{xx}^i = - \sum_{y \neq x} q_{xy}^i \end{cases} \quad (1.9)$$

Transition rates are simply expressed as the product of codon mutation rates μ_{xy} , and probabilities of fixation $\mathbb{P}_{fix}^i(x, y)$ from codon x to y . These two components of mutation and fixation that make up the substitution model are illustrated in figure 1.3, along with the parameters that they involve.

Mutational component

We define a site-invariant mutational component where multiple nucleotide mutations cannot occur at once, therefore prohibiting mutations between non-neighboring codons as in [Thorne et al., 2007, Yang and Nielsen, 2008, Rodrigue and Lartillot, 2016]. The nucleotide mutation process is described using a general time-reversible (GTR) process [Tavaré et al., 1986, Lanave et al., 1984].

π : nucleotide equilibrium frequencies

a, b, c, d, e, f : nucleotide exchangeabilities

$$R = \begin{matrix} & \begin{matrix} A & C & G & T \end{matrix} \\ \begin{matrix} A \\ C \\ G \\ T \end{matrix} & \begin{bmatrix} * & a\pi_C & b\pi_G & c\pi_T \\ a\pi_A & * & d\pi_G & e\pi_T \\ b\pi_A & d\pi_C & * & f\pi_T \\ c\pi_A & e\pi_C & f\pi_G & * \end{bmatrix} \end{matrix} \quad (1.10)$$

Among classical models of nucleotide mutations, this is the most general in that it involves the most parameters, which can be constrained to reproduce other commonly used models such as HKY85 if desired. Using such a parameter-rich specification for the mutational component would typically increase the difficulty of fitting the model to limited data but is not an issue in the context of simulation, where parameter values are controlled. The mutation rate in our model is then defined as

$$\mu_{xy} = \begin{cases} R_{uv} & \text{if codon } y \text{ can be reached from } x \text{ by mutating one nucleotide } u \rightarrow v \\ 0 & \text{otherwise} \end{cases} \quad (1.11)$$

Fixation component

The fixation probability of novel mutations from codon x to y is a function of the selection coefficient S_{xy}^i at site i

$$\mathbb{P}_{fix}(x, y) = \frac{S_{xy}^i}{1 - e^{-S_{xy}^i}} \quad (1.12)$$

For numerical reasons, a first order approximation $\mathbb{P}_{fix}(x, y) = (1 + S_{xy}^i/2)$ is used whenever $|S_{xy}^i| < 10^{-30}$. The scaled selection coefficient S_{xy}^i is null if x and y are synonymous codons. For non-synonymous codons, it is calculated as the difference of their scaled fitnesses F

$$S_{xy}^i = F_{AA(y)}^i - F_{AA(x)}^i = 2N \left(f_{AA(y)}^i - f_{AA(x)}^i \right) \quad (1.13)$$

Model specification	Parameters
$Q_{61 \times 61} = \begin{cases} q_{xy}^i = \mu_{xy} \times \frac{S_{xy}^i}{1 - e^{-S_{xy}^i}} & x \neq y \\ q_{xx}^i = - \sum_{y \neq x} q_{xy}^i \end{cases}$	π : nucleotide equilibrium frequencies a, b, c, d, e, f : nucleotide exchangeabilities ρ : scaling factor for the intensity of selection β : a vector of amino acid preferences
$\mu_{xy} : \text{GTR}(\pi, \{a, b, c, d, e, f\})$	<p>Optional parameters</p> B : intensity of GC-biased gene conversion H : CpG hypermutability rate Z : intensity of persistent positive selection
$S_{xy}^i = \rho \log \frac{\beta_{AA(y)}^i}{\beta_{AA(x)}^i}$	

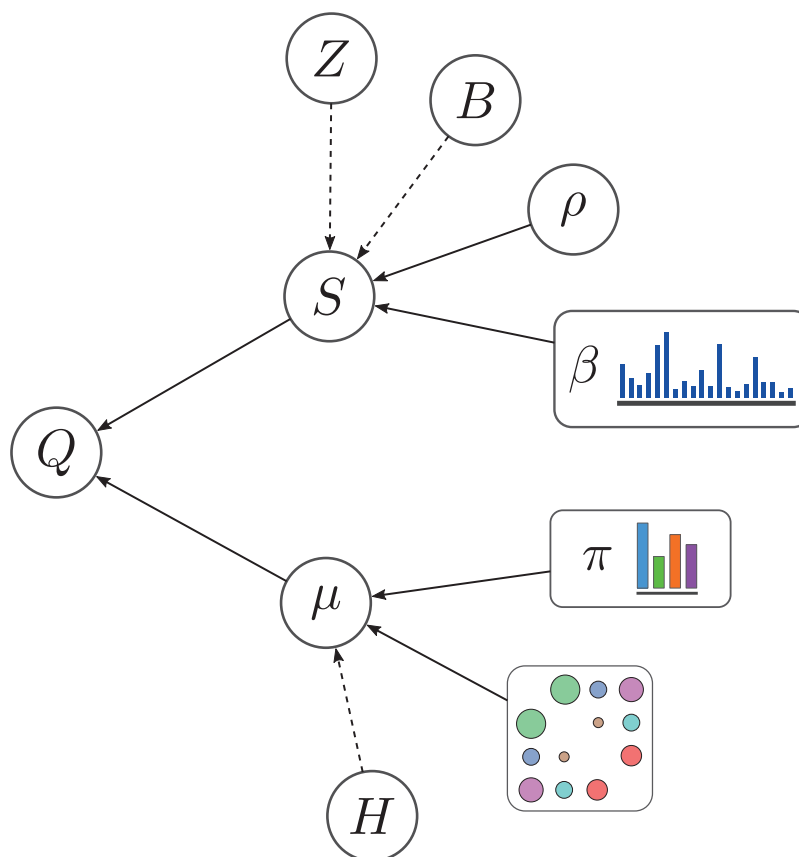


Figure 1.3: Representation of the mutation-selection model as a directed acyclic graph. Arrows indicate that a variable or parameter affects the value of the variable that it points to. Dashed arrows indicate optional parameters that are involved in extensions of the model.

where $f_{AA(c)}^i$ is the fitness of the amino acid encoded by codon c at site i . Ultimately, the substitution rate from codon x to y for haploid species is expressed as the expected number of mutants at each generation ($N\mu_{xy}$), multiplied by the probability of fixation of the mutation:

$$q_{xy}^i = N\mu_{xy} \frac{2s_{xy}^i}{1 - e^{-2Ns_{xy}^i}} \quad (1.14)$$

The same specification is used in many other mutation-selection models [Yang and Nielsen, 2008, Rodrigue et al., 2010, Parto and Lartillot, 2017, Parto and Lartillot, 2018, Tamuri et al., 2012, Tamuri, 2021].

To improve the realism of simulation, we use distributions of amino acid preferences that are estimated experimentally from [Bloom, 2017]. They consist in 263 vectors of frequencies V with size 20 (each summing to 1), that can be randomly sampled to act as a proxy for fitness values of each amino acid at one site. As a consequence, we do not dispose of actual fitness values for amino acids, as described in equation 1.13.

Denoting $\beta_{AA(c)}^i$ the experimental frequency at site i of the amino acid encoded by c , we specify the selection coefficient at site i to be

$$S_{xy}^i = \rho \log \frac{\beta_{AA(y)}^i}{\beta_{AA(x)}^i} \quad (1.15)$$

where ρ is an arbitrary scaling constant, that controls the intensity of selection and is proportional to N , which is not explicitly represented. For the purpose of simulating alignments, incorporating N would also involve setting arbitrary values for it, since we do not attempt to represent the evolution of a particular population where an accurate value of N would be relevant.

Equations 1.13 and 1.15 are thus equivalent with regard to the process that is represented:

$$\begin{cases} F_{AA(x)}^i = \rho \log \beta_{AA(x)}^i = 2N f_{AA(x)}^i \\ \log \beta_{AA(x)}^i \neq f_{AA(x)}^i \end{cases} \quad (1.16)$$

Because this model describes evolution independently at each site, the realism of simulations using it is limited. Proteins are not just ordered collections of amino acid, but they have multiple levels of structure that emerge from the interactions between amino acids, and are essential to their functional role. Modelling the co-evolution of interacting sites within one or between multiple proteins would therefore be an improvement of the realism of simulations, just like accounting for the genomic architecture in general. The evolutionary mechanisms that are represented are also reduced to the most basic components, and other phenomenons could be described more precisely, like codon usage bias, recombination and biased gene conversion, insertions-deletions, gene duplications... We propose some extensions of the model to account for some of these molecular mechanisms in section 1.3.3. It is done with the intent to evaluate their impact on the predictions made by detection methods, rather than attempt to produce perfectly realistic alignments, which is beyond the scope of this work — we are satisfied with “realistic enough”.

1.3.2 Simulations

The model is used to simulate codon alignments from the definition of the substitution rates in Q^i at each site i , using Gillespie’s algorithm for stochastic simulations[Gillespie, 1976] running along a phylogenetic tree with branch lengths. Gillespie’s algorithm allows to simulate the evolution of a

system that is driven by a stochastic process, such as the one we specified in our mutation-selection model. The main idea is that the number of transition events that occur within a time window Δt follows a Poisson process with parameter $\nu(x)\Delta t$, where $\nu(x)$ is the total rate of departure from the initial state x . Consequently, the waiting time for the next occurrence of a state transition follows an exponential distribution

$$\begin{aligned}\nu(x) &= \sum_{y, y \neq x} Q_{xy} \\ \tau &\sim \text{Exp}(1/\nu(x))\end{aligned}\tag{1.17}$$

Simulating the evolution a position in a sequence starting from a state x , along a branch with length $\Delta t = l$, can thus be done by drawing a value for τ and comparing it to the value of l . If $\tau \leq l$, then a transition had time to occur, and the nature of the transition is left to be determined; otherwise the system remains at its current state and the simulation on this branch has ended. The probability to reach an other state y from x , given that a transition has occurred, is proportional to the rate of transition from x to y . Therefore, the value for y can simply be sampled from a discrete distribution, with probabilities equal $Q_{xy}/\nu(x)$, for each possible state y . This process can then be repeated, taking y as the new starting state and the remaining branch length $l - \tau$, until the end of the branch is reached.

The algorithm requires that an initial state is set at the start of the simulation, that is the root of tree. We assume that the process is stationary, so that the distribution of codon states at the root is given by their equilibrium frequencies R , which are calculated as in e.g. [Yang and Nielsen, 2008]

$$R_x^i \propto \prod_{i=1}^3 \pi_{(x_i)} \times \exp\left(\rho \beta_{AA(x)}^i\right)\tag{1.18}$$

A codon state is drawn at the root of the tree from this distribution, that defines the initial state for the simulation that is propagated from branch to branch along the tree.

Sites can either be simulated under a neutral regime, where amino acid preferences β^i remain constant across branches, or undergo differential selection by switching β^i depending on the branch, and the phenotypic trait that is attached to it. Amino acid profiles β^i at each site are randomly drawn among the 263 empirical frequency vectors measured by [Bloom, 2017]. When simulating sites under differential selection, one profile per phenotypic trait value is drawn, with the constraint that the most frequent amino acid must be different between profiles on a given site. This helps preventing the simulation of sites under differential selection that lack the signal for it.

The parameter ρ controls the balance between the influences of genetic drift and natural selection on the diffusion of mutant alleles within the population, and is fixed to $\rho = 4$ in all experiments, unless specified otherwise. This value is the same as in [Rey et al., 2019], and was chosen to provide good discriminating power between methods (see [Rey et al., 2019, supplementary figure S4]). Simply put, this allows the simulation of sites which are neither too easy nor too difficult to identify by detection methods.

As for parameters of the mutational component, nucleotide equilibrium frequencies π are randomly drawn from a Dirichlet distribution, and exchangeabilities between nucleotides are symmetrical and drawn from a Gamma(1, 1) distribution.

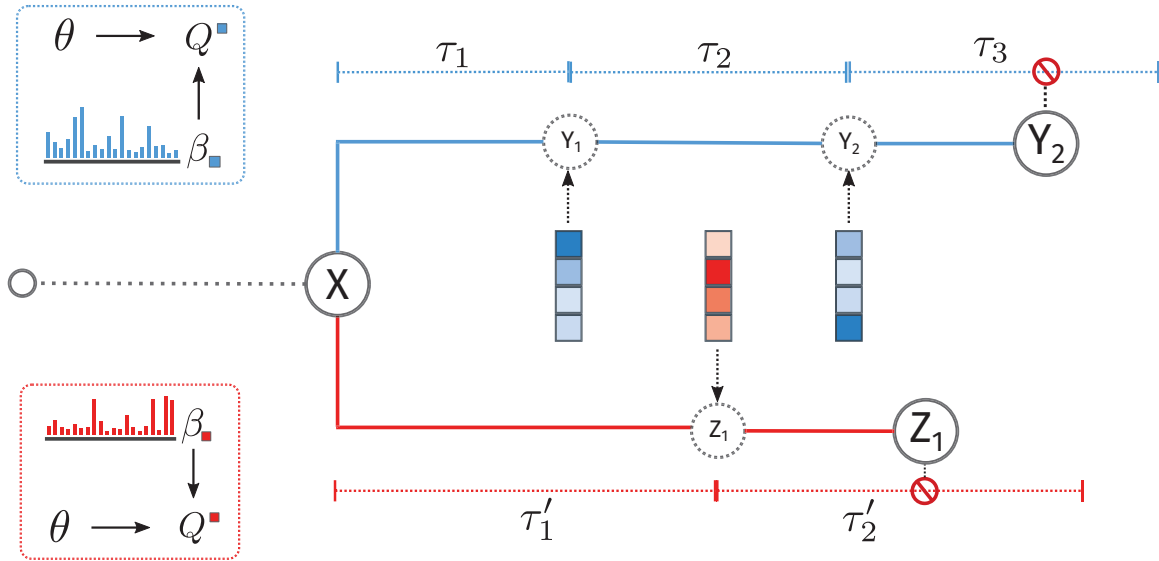


Figure 1.4: Simulation of codon state transitions using Gillespie’s algorithm, under different selection pressures described by pseudo-fitness profiles β . Segments labelled as τ_i depict randomly drawn waiting times between substitutions. Colored tile vectors are probability distributions for the new state. The set of parameters that are not dependent on the branch annotation is denoted by θ .

1.3.3 Extensions of the simulator

The base simulation model represents the substitution process by its essential components: a mutation rate and a probability of fixation, that accounts for the effect of selection and drift. However, other biological mechanisms do affect the dynamics of substitutions, and may interfere with the signal for directional direction in real sequence alignments. We included two of them in the simulation model, to evaluate the robustness of detection methods to these confounding factors: (1) GC-biased gene conversion (gBGC) that alters the probability of fixation of some mutations; (2) CpG hypermutability that increases the mutation rate of CpG dinucleotides. We also extended the model to allow the simulation of persistent positive selection (PPS), which may be challenging to distinguish from directional selection.

GC-biased gene conversion

GC-biased gene conversion (gBGC) is a non-selective process that occurs during meiotic recombination, by which GC alleles are favored compared to AT when repairing mismatches in AT/GC heterozygotes [Duret et al., 2002]. As a consequence, this increases the rate of fixation of GC alleles, and tends to increase the overall GC content of genomes, particularly near recombination hotspots. The effect of gBGC mimics positive selection, in that it promotes the diffusion of certain alleles within populations [Nagylaki, 1983]; however it is not an adaptive process, as it depends on the biochemical nature of the alleles, but not on their fitness. Although it can be distinguished from positive selection [Galtier and Duret, 2007] by accounting for the location within genomes and the nature of alleles that are fixed, they may be confounded by gBGC-naive methods performing systematic screens for directional selection.

We extend the model defined in section 1.3.1 to incorporate gBGC within the calculation of the

selection coefficient S

$$S_{xy}^i = \rho \left(\log \beta_{AA(y)}^i - \log \beta_{AA(x)}^i + \gamma(x, y) \right)$$

$$\gamma(x, y) = \begin{cases} B & \text{if } y \text{ is reachable from } x \text{ through } \{A, T\} \rightarrow \{G, C\} \\ -B & \text{if } y \text{ is reachable from } x \text{ through } \{G, C\} \rightarrow \{A, T\} \\ 0 & \text{otherwise} \end{cases} \quad (1.19)$$

We assume that B , the intensity of gBGC, is constant across all positions in the sequence, which is an unrealistic assumption as it typically is stronger near recombination hotspots — that our model can not simulate either. Nonetheless, we believe that this is sufficient to assess the robustness of methods to the general influence of gBGC as a confounding factor when searching for directional selection.

CpG hypermutability

Cytosine-guanine pairs on a DNA strand, or CpG dinucleotides, are often methylated at C, a configuration which favors the spontaneous mutation from methyl-C to T. CpG hypermutability denotes the fact that the frequency of such mutations is an order of magnitude higher than the frequency of other nucleotide mutations [Bulmer, 1986]. As such, it can be a force that alter the dynamics of substitutions at positions that are in this configuration.

The rate of mutation is scaled by a parameter $H \geq 1$ that controls the hypermutability rate of CpG dinucleotides, when applicable. In order to properly identify CpG dinucleotides on each strands of DNA, the simulator accounts for the flanking codons $\{l(x), r(x)\}$ of the mutating codon x

$$\begin{aligned} \text{forward strand } x_i^+ &= \begin{cases} (x_i, x_{i+1}) & \text{if } i < 3 \\ (x_i, r(x)_1) & \text{if } i = 3 \end{cases} \\ \text{reverse strand } x_i^- &= \begin{cases} (x_{i-1}, x_i) & \text{if } i > 1 \\ (l(x)_3, x_i) & \text{if } i = 1 \end{cases} \end{aligned} \quad (1.20)$$

This allows to conditionally scale the mutation rate for codon mutations that can be achieved by a single nucleotide mutation $C \rightarrow T$ on either DNA strand, when the mutated nucleotide site is a cytosine in a CpG dinucleotide

$$q_{xy} = \mu_{xy} \times \nu(x, y, l_x, r_x) \times \mathbb{P}_{fix}^i(x, y)$$

$$\nu(x, y) = \begin{cases} H & \text{if } (x_i = C) \rightarrow (y_i = T) \text{ and } x_i^+ = CG \\ & \text{or } (x_i = G) \rightarrow (y_i = A) \text{ and } x_i^- = CG \\ 1 & \text{otherwise} \end{cases} \quad (1.21)$$

Persistent positive selection

Persistent positive selection, also known as diversifying selection, occurs when substitutions in sequences are constantly promoted, without reaching an optimal state. This is typical of Red-Queen dynamics, where lineages that share ecological interactions undergo constant changes in response to adaptations from their partner. A common example is host-pathogen interactions, where selection favors mutations in the pathogen that improve its ability to exploit the host, and in return promotes mutations in the host that help evade the pathogen. In the case of virus-host interactions, such

situations have been highlighted at the molecular level, as the reproduction of the virus relies on its ability to interact with specific host cell receptors, and to evade the host immune system.

Similarly to [Tamuri, 2021], site-specific parameter $Z^i > 0$ is introduced in the expression of the selection coefficient S^i , controlling the increase in fitness obtained by simply changing the amino acid

$$S_{xy}^i = \begin{cases} 0 & \text{if } x \text{ and } y \text{ are synonymous} \\ \rho \left(\log \beta_{AA(y)}^i - \log \beta_{AA(x)}^i + Z^i \right) & \text{otherwise} \end{cases} \quad (1.22)$$

1.3.4 Empirical phylogenies

To improve the realism of the simulated datasets, we use a set of empirical phylogenies. Because the representation of these trees requires a lot of space and would break the flow of the document, I present them in the appendix. Four of them were previously used in [Rey et al., 2019] for similar purposes, and two more (HIV and Influenza) were included as well to cover a broader range of configurations.

Rodents [Rey et al., 2019]

Appendix figure B.1.

A small phylogeny of rodent species, annotated with clades adapted to life in arid environments.

Cyperaceae C_3/C_4 [Besnard et al., 2009]

Appendix figure B.2.

Cyperaceae, also known as sedges, are a family of flowering plants similar to grasses, that include the papyrus plant. The phylogeny was reconstructed by [Besnard et al., 2009] to investigate the molecular bases for the convergent adaptation in sedges to C_4 photosynthesis, which is more efficient in warm environments than the ancestral C_3 pathway.

Orthomam Echolocation [Scornavacca et al., 2019]

Appendix figure B.3.

A phylogeny inferred from the Orthomam database, which consists in a set of 14 509 curated alignments of coding sequences from 116 mammalian species. The phylogeny was reconstructed at maximum likelihood from the alignments, using the IQ-TREE software [Nguyen et al., 2015]. A phenotype trait annotation was built from a subset of echolocating species, with ancestral state inferred at maximum parsimony using Fitch’s algorithm [Fitch, 1971].

Amaranthaceae C_3/C_4 [Kapralov et al., 2012]

Appendix figure B.4.

Amaranthaceae are another family of flowering plants that convergently evolved the C_4 photosynthesis pathway. Examples of commonly known species from this family are spinach and quinoa. This dataset was used in [Kapralov et al., 2012] to identify traces of positive selection in the sequence of Rubisco — a key enzyme in the photosynthesis pathway —, as well as in [Parto and Lartillot, 2018] that aimed to distinguish positive selection from convergent evolution.

HIV [Murrell et al., 2012a]

Appendix figure B.5.

A large phylogeny of HIV strains, with a phenotype annotation that represents an experimental treatment where HIV strains were exposed to antiretroviral drugs, to investigate the acquisition of drug resistance mutations. The resulting partition of the tree involves one transition event on every odd terminal branch, amounting to a total of 238 transitions.

Influenza [Tamuri et al., 2009]

Appendix figure B.6.

This is a phylogeny of influenza strains, divided into two clades depending on the host species of the virus: avian species that are the ancestral reservoir, and humans which are a secondary

host. This phylogeny is the only one among those we consider that does not involve convergent adaptation as there is a single transition event from background (avian) to foreground (human) conditions, that represents the switching between hosts.

1.4 Measuring the detection performance of a statistical method

When performing hypothesis testing with statistical methods, predictions come in the form of probabilities, whether they are p -values or posterior probabilities. Given such scores, an actual prediction can then be made by setting a threshold at which a value is extreme enough to be a signal for a positive test. For example, p -values are a probability to draw a value for a test statistic under the null hypothesis, that is equal or more extreme than the observed one. It thus represents the risk that is taken to wrongfully reject the null hypothesis. In our day-to-day usage of statistical tests, we are constantly confronted to the choice of the threshold for p -value significance. Lower threshold values make the test more precise, as the production of false positive is less likely to occur; however the test is then more prone to fail to detect positives, and produces more false negatives instead. This trade-off is often represented using a combination of two metrics: *sensitivity* and *specificity*. Sensitivity is the *true positive rate*, the fraction of true positive that were correctly predicted as positive, while specificity is the *true negative rate*, the fraction of true negative that were predicted as negative.

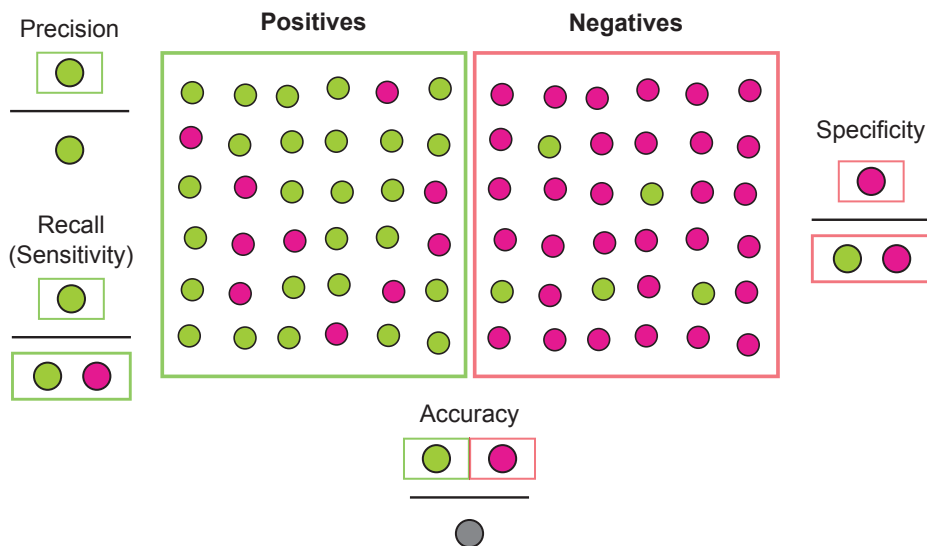


Figure 1.5: Example dataset, comparing predictions to true categories. The color of a circle represents a prediction on one data point, while the square it is located in indicates which category it actually belongs to.

They can be combined in a single *accuracy* metric, which is the fraction of correct assessments among all predictions. This indicator is well adapted to balanced data, where the amount of positive and negative elements is roughly equal, but it may be misleading when it is not the case. Indeed, when most elements in a dataset are negatives, the accuracy of a method that always predicts negatives would still be quite good. This is typically the case of the kind of data we are interested in: the amount of positive sites associated to a phenotypic trait is expected to be low compared to the amount of neutral sites. Therefore we can use instead *precision* and *recall* to measure the quality of predictions. Precision measures the fraction of true positives among all positive predictions, while

recall measures the proportion of actual positives that were identified. In this way the focus is set on reliably detecting positive elements, and the assessment of performance is more robust to imbalanced datasets.

When we need to compare the quality of predictions between methods, we do not want to make such arbitration. Instead, we would rather have an overview of the behavior of the method regardless of a particular threshold that would determine the trade-off between recall and precision. Indeed, a method that has better predictive power than another is expected to have better precision and recall across the board. This kind of comparison can be done by applying each method to a set of observations, where the outcome for each datum is known (e.g by simulating the data under different regimes, and using the methods to predict the regime from the data). Any threshold can then be applied to the resulting probabilities, providing a set of predictions for each method. By comparing these predictions to the known true state, we can compute the proportion of true positives within the predicted positives (the precision), and the proportion of predicted positives within the true positives (the recall). Repeating this for all the possible thresholds within the interval $[0; 1]$, we can produce a *precision-recall curve* by plotting the measured precision as a function of the recall. The *area under the curve (AUC)* can then be computed to summarize the detection performance of a method.

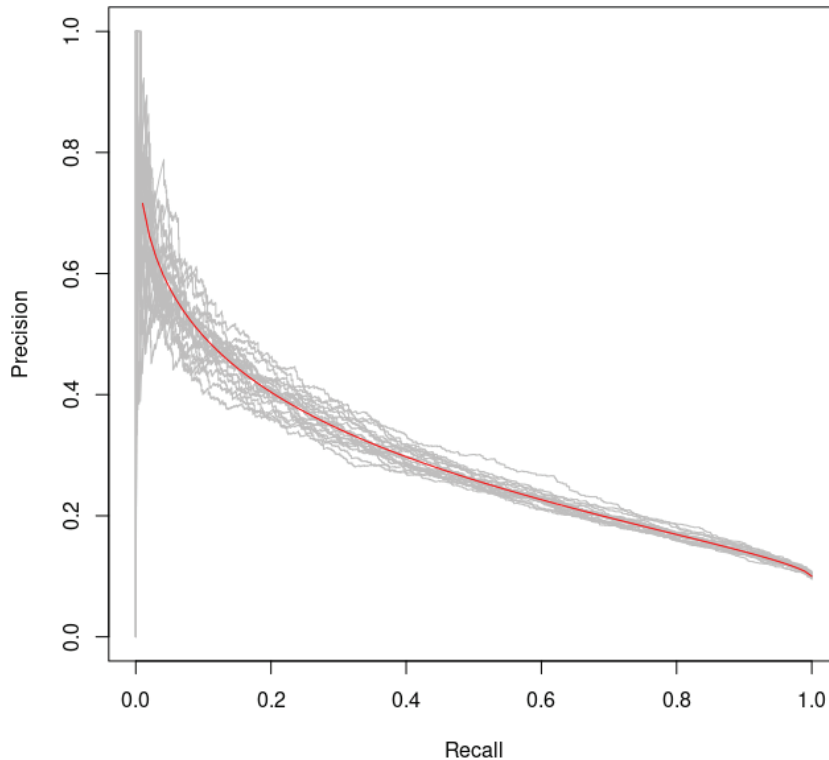


Figure 1.6: Precision-Recall curves are variable between runs (grey lines). The average trajectory is shown as the red curve. The predictions that were used to generate this figure are kept abstract here, and simply illustrate the variation of the AUC across repetitions.

A major pitfall is that precision-recall curves of one method are variable between runs, particularly for low recall values, as shown in figure 1.6. Instead of a single precision-recall curve, a more robust estimate of the method's performance would be the average curve calculated from multiple runs. A method was proposed in [Boyd et al., 2013] to compute the expected average AUC and the

associated confidence interval at a chosen risk threshold α , without the need for repeated evaluation of the method. All PR AUC results in this thesis are presented with a 95% confidence interval ($\alpha = 5\%$). A convenient property of this procedure is that it only requires that the methods to be evaluated provide predictions in the form of scores that can be sorted. In that manner, different methods producing different kinds of prediction, like p -values or posterior probabilities, can have their performance compared regardless.

An OCaml implementation is available at <https://github.com/pveber/prc/>, and a mirror implementation as an R package is at <https://github.com/lsdch/prauc>.

1.5 Evaluation pipeline: technical aspects

We developed an evaluation pipeline for the evaluation of detection methods, that implements the simulator and performance measurement that were presented in previous sections of this chapter. This pipeline is implemented in the OCaml programming language, building on a previous implementation of a pipeline dedicated to running programs that detect convergent molecular evolution. It allows running computation tasks — functions or programs — concurrently, using the Bistro library [Veber, 2017]. External programs can be launched from Docker images. An important feature that is provided by Bistro in this implementation is the caching of results, that prevents the repetition of previously done computations; it is combined to a dependency system that identifies parts of the pipeline that must be computed again after a change upstream that would affect the result. Because the pipeline is directly implemented in a full-fledged programming language, a lot of control is given on each of its steps and extensions can be added in a flexible way.

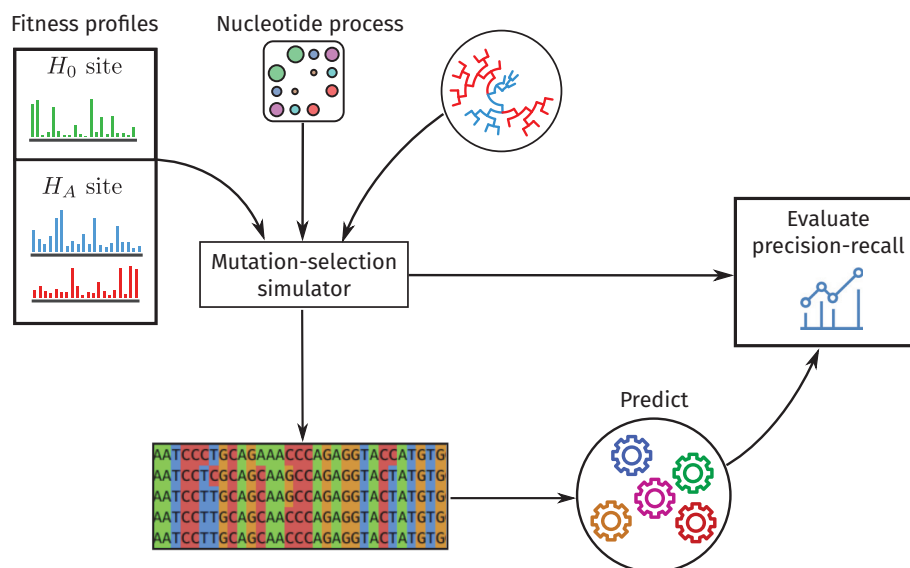


Figure 1.7: Illustration of the evaluation pipeline.

As depicted figure 1.7, the process of evaluation consists in three steps:

1. **simulation** using randomly drawn fitness profiles, that orient the selective pressure depending on the regime of each site and the phenotype annotation in the tree
2. **prediction**: scan alignments using a set of detection methods that ought to be evaluated

3. **evaluation:** compare predictions to the simulation regime of each sites, and compute precision-recall estimates to measure the performance of each method

Chapter 1 summary: *Framework for the evaluation of detection methods*

The main objective of this work being to enable efficient and reliable detection of sites with coding sequences alignments that are associated to a phenotype, we require some tools to **evaluate the quality of predictions** for each method that we investigate. For this purpose, we should be able to simulate sequence alignments, including sites that are associated or not to a phenotypic trait, so that we can confront predictions to a “ground truth”. This evaluation of predictions should give a synthetic measure of the reliability of a method, and be well-suited to the characteristics of the problem.

Our answer to these requirements is an implementation of an **evaluation pipeline**, that integrates the simulation of alignments containing both negative and positive sites along phylogenies, their analysis using a set of detection methods, and the calculation of performance measurements.

We developed a simulator based on a **mutation-selection model of codon substitutions**, with extensions of the base model to incorporate evolutionary mechanisms that may interact with the signal for differential selection: **GC-biased gene conversion (gBGC)**, **CpG hypermutability**, and **persistent positive selection**.

A set of six **empirical phylogenies** was picked from the literature, in an attempt at making the simulations more realistic, and to represent a diversity of tree topologies and annotations. The resulting alignments are of course not perfectly realistic, but the simulation model has strong theoretical foundations and gives a mechanistic representation of the substitution process.

The procedure to evaluate the quality of predictions by confronting them to the “truth” established by the simulation relies on **precision-recall**, a two-faceted measure that emphasizes on the ability to predict positive sites correctly, and essentially quantifies how the ranking of predictions is accurate. We leverage a statistical procedure published in the literature to estimate confidence intervals on the PR AUC, accounting for the variability between runs.

The large majority of the experiments in this work are thus conducted on simulations, and evaluations are systematically done by computing and comparing precision-recall AUC, although results of analyses on empirical data are also presented at the end of [chapter six](#).

Chapter 2

Is this basically inter-species GWAS?

Genome-wide association studies (GWAS) scan sequence alignments for associations between genotype markers and phenotypes (see e.g. [Uffelmann et al., 2021] for a review) within populations of the same species. Because these approaches are designed to analyze polymorphism data, they are not originally intended to be applied to inter-specific data. However, when looking at some GWAS models that account for the relatedness between individuals in the sample, it might be tempting to investigate how such approaches could be applied to our problem, which could be somewhat described as “inter-specific association studies”. Indeed, they enable the detection of site modifications that are associated to a phenotype, are already applicable to quickly scan large scale datasets, and are widely used in medical research, suggesting they work well in their intended application scope. The throughput of GWAS is a benefit from their implementation based on linear models, which can be fitted at maximum-likelihood by solving a linear system of equations — linear models are simple enough that the maximum of their likelihood function can be determined analytically.

This chapter investigates the use of several approaches based on so-called linear mixed models, linear models that incorporate a random effect to describe the relatedness between observations. The question we ask is rather naive: can GWAS approaches be translated to the inter-specific scale for efficient detection of genotype-phenotype associations ?

We first attempt to adapt an existing GWAS model named GEMMA to our problem, and find that it is not very appropriate to the type of data that we manipulate: specifically, genotypes can only be represented as binary markers, when we need to model genotypes as amino acids, involving 20 possible states at most. This motivates the design of a linear mixed model better suited to these needs, but we do not find that this model is an improvement over GEMMA when comparing them using simulations in our evaluation pipeline. Remaining in the generalized linear modeling framework, we then explore the application of a multinomial model for the amino acid composition of alignment sites, that does not account for the phylogenetic structure at first. This phylogeny-unaware model improves the detection performance when compared to both previous approaches. We consider extending the model to incorporate a random effect for the relatedness between observations, but find that it can not be done easily and would involve a considerable increase in its computational footprint.

Contents

2.1 Linear Mixed Models	39
2.1.1 GEMMA: Genome-wide Efficient Mixed Model Association	40
2.1.2 Linear Mixed Model	42
2.2 Comparison of performance	43
2.2.1 Multinomial as a baseline reference for performance comparison	44
2.2.2 Linear mixed models do not improve on Multinomial	46
2.2.3 The cost of ignoring the phylogeny	47
2.2.4 Multinomial with phylogenetic random effect: it's complicated	48

2.1 Linear Mixed Models

For our purposes, the general idea is to model at each site the phenotype as a response to a linear function of the amino acid observed at each sequence, while accounting for the relatedness of phenotypes. The latter may be done by introducing a “random effect” in the model, a random variable that is shared across observations and induces a correlation structure between them. If the covariance structure is known or can be estimated, it can also be directly be incorporated in the model. Before describing actual models and their application in our setting, I propose a short and general introduction on linear mixed models, how they are specified and how to perform maximum likelihood inference in that context.

Gaussian linear models represent the distribution of a response variable y as a normal distribution, whose mean is determined by a linear combination of experimental factors x_1, \dots, x_m . The model is then

$$y = \theta_1 x_1 + \dots + \theta_m x_m + \varepsilon \text{ where } \varepsilon \sim \mathcal{N}(0, \sigma^2) \quad (2.1)$$

When several observations $Y = (y_1, \dots, y_n)$ are available, this can be put in vector form

$$Y \sim \mathcal{N}_n(X\theta, \sigma^2 I_n) \quad (2.2)$$

where Y is an n -vector, X is a so-called design matrix of observations with size $n \times m$, θ is an m -vector and I_n is the identity matrix. Each row of X corresponds to an observation, and each column to an observed variable. A coefficient θ_j is affected to each column $X_{.j}$, and is a parameter to optimize when fitting the model. X may include a column of 1, with a matching coefficient in θ to represent an intercept. This model can be solved for θ at maximum-likelihood using the following estimator:

$$\hat{\theta} = (X^\top X)^{-1} X^\top Y \quad (2.3)$$

In this model, observations of Y are assumed independent and have equal variance σ^2 . Linear mixed models let go of the assumption of independence by describing the covariance structure between observations as a matrix Σ

$$Y \sim \mathcal{N}_n(X\theta, \Sigma) \quad (2.4)$$

They can also be written as

$$Y = X\theta + ZU + \varepsilon \quad \text{where } \begin{cases} U \sim \mathcal{N}_n(0, \gamma^2 I_m) \\ \varepsilon \sim \mathcal{N}_n(0, \sigma^2 I_n) \end{cases} \quad (2.5)$$

where Z is a matrix that describes the relatedness between observations, and U is the associated vector of random effects. The variance is separated in two terms, one term γ^2 that scales the covariance associated to the random effect and a general term σ^2 associated to the residual error ϵ . Equations 2.4 and 2.5 are equivalent, and can be identified to each other knowing that

$$\Sigma = \gamma^2 Z Z^\top + \sigma^2 I_n \quad (2.6)$$

The covariance matrix Σ is generally not known. However a scaled approximation of Σ as a correlation matrix C is sometimes available

$$\Sigma = \sigma^2 C \quad (2.7)$$

where σ^2 is a general scaling factor for the correlation structure defined in C . When applied to phylogenetic data, C represents a measure of the relatedness between each observation, that decreases with longer evolutionary distances for example and can be used to obtain the covariance matrix Σ .

This specification strays from the formalism where independence between observations is assumed, and can not directly be fitted at maximum likelihood. However in such situations, it is possible to reduce the model to a classical form of Gaussian linear model, by performing a Cholesky decomposition of C [Bel et al., 2016, chapter 5, p. 170] using the fact that correlation matrices are symmetric

$$C = C^{1/2} (C^{1/2})^\top \quad (2.8)$$

$$Y \sim \mathcal{N}_n(X\theta, \sigma^2 C) \implies C^{-1/2} Y \sim \mathcal{N}_n(C^{-1/2} X\theta, \sigma^2 I_n) \quad (2.9)$$

where I_n is the identity matrix with dimension n . A classical Gaussian linear model can then be identified from this expression

$$\tilde{Y} \sim \mathcal{N}_n(\tilde{X}\theta, \sigma^2 I_n) \quad \text{where} \quad \begin{cases} \tilde{Y} &= C^{-1/2} Y \\ \tilde{X} &= C^{-1/2} X \end{cases} \quad (2.10)$$

which in turn can be solved for θ at maximum likelihood using the established estimator in this context, as in equation 2.3

$$\hat{\theta} = \left(\tilde{X}^\top \tilde{X} \right)^{-1} \tilde{X}^\top \tilde{Y} \quad (2.11)$$

Briefly, linear mixed models are variations on the classical form of linear models to include a specific structure of covariance between observations. When fitting a linear mixed model, it is transformed to be identifiable to a linear model with a constant variance that can then easily be solved at maximum-likelihood.

2.1.1 GEMMA: Genome-wide Efficient Mixed Model Association

As a first attempt, we explore the application of GEMMA [Zhou and Stephens, 2012], a tool that implements a linear mixed model originally designed for GWAS analysis of polymorphism data, to perform inter-species association studies instead. It is based on the following model:

$$Y = W\alpha + X\theta + u + \epsilon \quad \text{where} \quad \begin{cases} u \sim N_n(0, \gamma^2 C) \\ \epsilon \sim N_n(0, \sigma^2 I_n) \end{cases} \quad (2.12)$$

In this model, the phenotype Y is modelled as a linear function of fixed effects W (which is only an intercept term in our case), marker genotypes X , and random effects u . C is a relatedness matrix between observations, that represents the phylogenetic structure between observations. γ^2 is a scaling factor on the random effect, and σ^2 is the residual variance.

GEMMA is intended to work on same-species nucleotide sequence alignments. The vector X encodes the presence or absence of a SNP variant, using 1 to signify its presence, or 0 for its absence. We divert this encoding to represent instead the presence or absence of an amino acid at the site for each species. This means that at each site, we are constrained to repeatedly test for association between the phenotype and a single amino acid. In equation 2.13, observations of the phenotype are encoded as a binary vector Y , whose values denote the trait observed for each species. In a similar way, X encodes the observation (presence or absence) of an amino acid AA at the site under consideration, for each species.

$$\begin{matrix} & & & & AA \\ Y = & \begin{matrix} sp_1 \\ sp_2 \\ \vdots \\ sp_n \end{matrix} & \begin{bmatrix} 0 \\ 1 \\ \vdots \\ 1 \end{bmatrix} & X = & \begin{matrix} sp_1 \\ sp_2 \\ \vdots \\ sp_n \end{matrix} & \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \end{matrix} \quad (2.13)$$

As for the relatedness matrix C , it is estimated as the mean observed covariance in the genotype data

$$C = \frac{1}{m} \sum_{i=1}^m (X^{(i)} - I_n \bar{x}^{(i)}) (X^{(i)} - I_n \bar{x}^{(i)})^\top \quad (2.14)$$

where $X^{(i)}$ is the genotype vector at site i , and m the total number of sites; $\bar{x}^{(i)}$ is the mean of genotypes encoding at site i . We adapt our usage of this expression, by first computing at each site i the matrix of covariance $C^{(i)}$, as the mean of the observed covariance for each individual amino acid. The global matrix of covariance is then obtained as the mean of site-level covariance matrices.

$$\begin{aligned} C^{(i)} &= \frac{1}{k} \sum_{j=1}^k (X_j^{(i)} - I_n \bar{x}_j^{(i)}) (X_j^{(i)} - I_n \bar{x}_j^{(i)})^\top \\ C &= \frac{1}{m} \sum_{i=1}^m C^{(i)} \end{aligned} \quad (2.15)$$

where k is the number of distinct amino acid at site i , and $X_j^{(i)}$ is the vector of observations across species of amino acid j at site i .

Because of the binary representation of genotype markers, one test has to be performed for each distinct amino acid that is observed at each site. GEMMA implements three different tests for association: the likelihood ratio test, the Wald test, and the Lagrange multiplier or score test. All of these procedures are designed to test whether a model fits the data significantly better than a constrained version of it that involves fewer free parameters¹. Finally, $(3 \times k)$ p -values are produced at each site, with k the number of distinct amino acids observed at the site. In order to provide a

¹The inner workings of each test are not detailed here for brevity, and because I believe it would not help to understand the protocol we used here, nor would it help justifying its admittedly shaky statistical bases. Nevertheless, the likelihood ratio test is presented in the following sections in this chapter, and more extensively discussed in sections 4.2.4 and 4.4. For an explanation of the three kinds of test and the relation between them, see: <https://stats.oarc.ucla.edu/other/mult-pkg/faq/general/faqhow-are-the-likelihood-ratio-wald-and-lagrange-multiplier-score-tests-different-and-or-similar/>

single p -value for a given site, we simply take the minimum value from the set of all p -values that were computed using GEMMA. The result of this is actually not a p -value anymore, but provides us with a score that we can use to rank sites and compute a precision-recall estimate as described in the previous chapter.

As it clearly appears that the descriptors in this models are not well adapted to our data, this motivates the design of a similar model that would enable to encode multiple amino acids instead of a single one. I describe our attempt at making such a model in the following section.

2.1.2 Linear Mixed Model

Using GEMMA, we can only test the association of the phenotype with the presence or absence of one amino acid, because the method is intended to work with a binary alphabet: the presence or absence of a SNP marker. We misused the model to test the effect of each amino acid independently at each site, which was a bit awkward. We can transform the model to test for the association of the phenotype to multiple amino acids at once, by specifying a design matrix where each distinct amino acid observed at the site is encoded for its presence in each species, instead of only one amino acid as in GEMMA. Equation 2.16 illustrates how the design matrix G in our model differs from that of GEMMA, which is actually a collection of design matrices X_{AA} for each observed amino acid.

$$X_{AA} = \begin{matrix} & AA \\ \begin{matrix} sp_1 \\ sp_2 \\ \vdots \\ sp_n \end{matrix} & \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \end{matrix} \longrightarrow G = \begin{matrix} & A & C & I & \dots & K & R & V \\ \begin{matrix} sp_1 \\ sp_2 \\ \vdots \\ sp_n \end{matrix} & \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 & 0 \\ \vdots & & \vdots & & & \vdots & \\ 1 & 0 & 0 & \dots & 0 & 0 & 0 \end{bmatrix} \end{matrix} \quad (2.16)$$

In this manner, we avoid repeating the test for each amino acid at one site like we did with GEMMA. Phenotype observations are modeled as a random variable distributed as a multivariate normal distribution, with an n -dimensional mean vector that is composed of an intercept term μ and a fixed effect $G\theta$, and a covariance matrix Σ

$$Y \sim \mathcal{N}_n(\mu + G\theta, \Sigma) \quad \text{where} \quad \begin{cases} Y : n\text{-vector of binary phenotype} \\ G : \text{design matrix of genotypes} \end{cases} \quad (2.17)$$

As for the covariance matrix Σ , it is derived from a matrix of correlation estimates as $\Sigma = \sigma^2 C$, where C is a matrix of correlation estimates. To obtain C , we could use the estimation provided by GEMMA as described in the previous section. However, our data are not well suited to a binary representation as genotype markers, and might deteriorate this estimation.

We try an alternative way to estimate the relatedness between species, by using a Brownian model of evolution along the tree. Under this assumption, the correlation between two leaves of the tree is the extent of their common evolutionary history, i.e. the time separating their most recent common ancestor from the root of the tree, added to the variance at the root node σ_R^2 , as illustrated figure 2.1. Although this term of variance at the root should be estimated, we neglect it as it makes the estimation procedure more complicated, and is expected to have an impact on the significance of our results, but not on their ordering. This is an attempt at finding a better heuristic for the determination of the correlation structure than our hijacking of GEMMA's features.

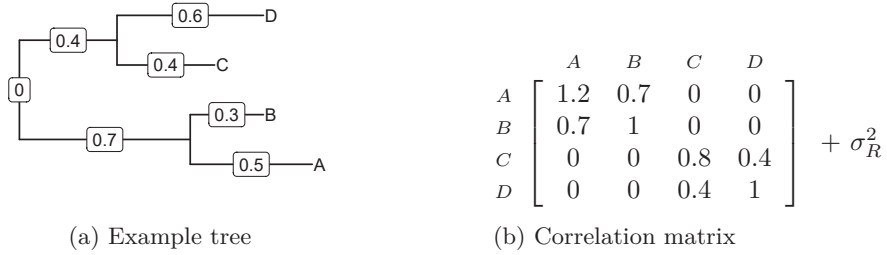


Figure 2.1: Correlation between observations at the tip are calculated as the amount of evolutionary history they share in the tree: that is the distance from the root to their most recent common ancestor.

Hypothesis testing

We compare two alternative models at each site, a reduced model where the amino acid genotype has no effect on the phenotype, and a full model where each observed amino acid i is associated to a coefficient θ_i .

$$\begin{aligned} \text{reduced model: } Y &\sim \mathcal{N}_n(\mu, \Sigma) \\ \text{full model: } Y &\sim \mathcal{N}_n(\mu + G\theta, \Sigma) \end{aligned} \quad (2.18)$$

where μ is the intercept parameter, capturing a mean effect common to all species. Parameters can be estimated at maximum likelihood, using Cholesky decomposition to reduce the model to a Gaussian linear model without random effect, which can then be easily solved for θ as described in the introduction of this chapter.

These two models are nested within each other, as the reduced model is equivalent to the full model where all θ_i are null. This allows comparing their fits using a likelihood ratio test (LRT), a statistical tool that tests the null hypothesis that a richer model does not significantly fit the data better than a more simple one, while accounting for its increased complexity. As the name implies, this is done by comparing the likelihoods of the two models, which is equivalent to comparing the sum of square errors (SSE) between each model. This likelihood ratio defines a test statistic D , which follows a chi-square distribution under the null hypothesis that the full model does not significantly fit the data better than the reduced model when accounting for its additional parameters.

$$\begin{aligned} D &= -2 \log \frac{L(Y, M_r)}{L(Y, M_f)} \\ &= 2 \left(\log L(Y, M_f) - \log L(Y, M_r) \right) \\ &= 2 \left(SSE(Y, M_f) - SSE(Y, M_r) \right) \\ D &\sim \chi^2(\dim(\theta)) \end{aligned} \quad (2.19)$$

where M_f denotes the full model, and M_r the reduced one. The number of degrees of freedom of the chi-square distribution is the number of added parameters, which is the dimension of θ .

2.2 Comparison of performance

We can use our evaluation pipeline that I described in [chapter one](#) to compare performance between the two methods based on linear models that were presented above. However, it would be even

more informative if we could compare them to a baseline method, which we choose to be a non-phylogenetic approach called Multinomial, and was previously included in [Rey et al., 2019]. It is a good candidate to provide a reference point, as it is a simple method that does not account at all for the phylogenetic structure in the data, and assumes independence between observations; nonetheless its predictive capabilities are not too bad, as reported in [Rey et al., 2019], even though they do not reach the level of phylogenetic approaches. I therefore give a brief description of the Multinomial model before presenting the results of our evaluation of performance for the three methods.

2.2.1 Multinomial as a baseline reference for performance comparison

This model was originally implemented and evaluated in [Rey et al., 2019] in the context of detecting convergent sites within coding sequence alignments. It does not account for the phylogenetic structure that ties the observations together, and was found to be less reliable than phylogenetic approaches. The multinomial distribution describes the outcome of repeated, independent draws with replacement from a categorical distribution, by counting the number of times each category is drawn. It is to the categorical distribution what the binomial distribution is to the Bernoulli distribution. In turn, the categorical distribution is a generalisation of the Bernoulli distribution to k categories, instead of two.

	2 categories	k categories
parameter	p	$\pi = p_1, \dots, p_{(k-1)}$
1 trial	Bernoulli	Categorical
n trials	Binomial	Multinomial

Table 2.1: Relation between the multinomial distribution and other discrete probability distributions

The outcome of n multinomial trials with k categories can be represented as a k -vector of counts for the number of times each category was drawn. We can thus use the multinomial distribution as a simple model for the observed amino acid counts at a site. Each observed amino acid defines a category, amounting to a maximum of $k = 20$ if all amino acids are observed, although this quantity is typically lower as protein sites tend to favor a restricted subset of amino acids. We define a base model where the amino acid counts are distributed as a single multinomial distribution as shown equation 2.20, and compare it to a model where the amino acid counts depends on the phenotype as in equation 2.21. In this model, the relationship between genotype and phenotype is reversed compared to the two previous models: instead of modeling the phenotype conditionally to the genotype, it is the distribution on the genotype that depends on the phenotype.

$$\text{reduced model: } C \sim \text{Multinomial}(\pi) \quad (2.20)$$

$$\text{full model: } C_j \sim \text{Multinomial}(\pi^{(j)}) \quad \text{where } j \in \{A, B\} \quad (2.21)$$

where C is a vector of amino acid counts at one site, and π is a vector of amino acid frequencies; A and B denote phenotype categories — that we make binary for simplicity — so that C_A is the vector of amino acid counts for the subset of species having phenotype A .

Once again, these models are nested within each other: the reduced model can be expressed using the full model by constraining its parameters so that $\pi^A = \pi^B$, which enables their comparison through a LRT. To do this, both models are first fitted at maximum likelihood. The likelihood of a

vector of counts C at one site, with regard to the reduced model and its parameter π is

$$L(C, \pi) = n! \prod_i^k \frac{\pi_i^{c_i}}{c_i!} \quad (2.22)$$

The log likelihood of the heterogeneous model is then easily computed as the sum of the log likelihoods obtained for each condition:

$$\log L(C_A, C_B, \pi^A, \pi^B) = \log L(C^A, \pi^A) + \log L(C^B, \pi^B) \quad (2.23)$$

where π^A and π^B are the probability parameters for the multinomial distribution under the phenotypic condition A and B respectively.

Both of these expressions can be derived to show that the maximum likelihood estimate for any frequency π_i in π is calculated as

$$\hat{\pi}_i = \frac{x_i}{\sum_i^k x_i} = \frac{x_i}{n} \quad (2.24)$$

which is simply the proportion of draws of each amino acid i among the total number of trials, i.e. the observed amino acid frequencies at the site — or within each phenotype category in the case of the full model.

LRT can then be performed by comparing the ratio of likelihoods of the two models and test its significance using a null chi-square distribution, which is shaped using the number of additional parameters ($k - 1$), where k is the number of distinct amino acids observed at the site. One degree of freedom is removed here, because frequencies in a vector π sum to 1 and one of them can be deduced from the others.

$$\begin{aligned} D &= 2 \left(\log L(C_A, \pi^A) + \log L(C_B, \pi^B) - \log L(C, \pi) \right) \\ D &\sim \chi^2(k - 1) \end{aligned} \quad (2.25)$$

2.2.2 Linear mixed models do not improve on Multinomial

Precision-recall performance was measured for each of the three methods, using alignments simulated with the six empirical phylogenies described in section 1.3.4. Average AUC estimate and 95% confidence intervals were computed using the procedure introduced in section 1.4, and are shown in figure 2.2. The linear mixed model depicted in section 2.1 was fitted using either the Brownian correlation matrix (LMM) or the correlation matrix as calculated by Gemma (LMM Gemma).

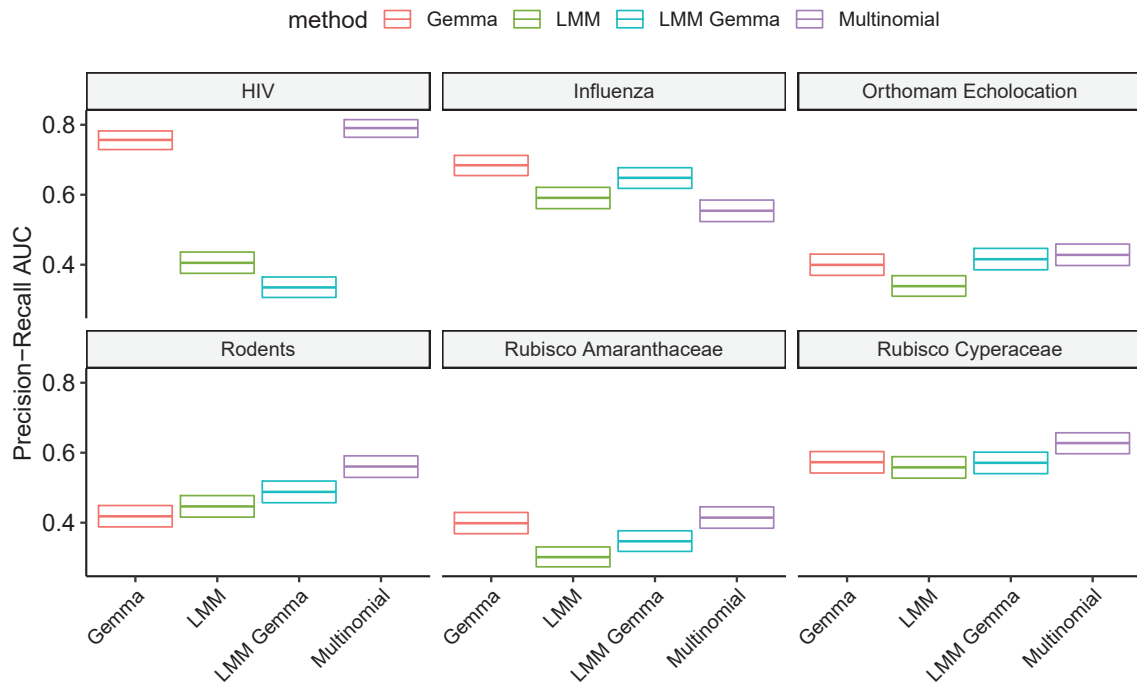


Figure 2.2: Comparison of detection performance measured as precision-recall AUC between Gemma, our linear mixed model (LMM) either using a Brownian approximation for the relatedness between observations or the correlation matrix estimated by Gemma. Performance of Multinomial acts as reference.

We find that the multinomial model is generally better than methods based on linear mixed models, except on one dataset simulated with the Influenza phylogeny. This poor performance is consistent with the high false positive rate observed for the method on this dataset, as shown in table 2.2. The same issue on this dataset will resurface later on when using another detection method, and our investigations have led us to think it has to do with the number of transitions in the phylogeny. I refrain from discussing this matter here, but it is explored in section 6.2.4.

Dataset	Gemma	LMM	LMM Gemma	Multinomial
Cyperaceae	0.02	0.03	0.01	0.07
Amaranthaceae	0.00	0.02	0.01	0.05
Rodents	0.00	0.02	0.00	0.00
Echolocation	0.01	0.02	0.01	0.02
HIV	0.00	0.14	0.07	0.00
Influenza	0.04	0.01	0.00	0.23

Table 2.2: Observed false positive rate at the 0.05 threshold. A well calibrated method should yield a proportion of 0.05 false positives.

Looking back at the benchmark results, they are somewhat surprising: the multinomial model that does not account for the phylogeny appears to be generally better than linear models with a phylogenetic random effect. A possible explanation would be that the modeling of the phenotype as a linear response to the presence or absence of amino acids is not adequate, and that working with amino acid frequencies gives more power to discriminate sites independent from the phenotype to those that are associated to it. Nonetheless, the multinomial method is not satisfactory because it is unaware of the phylogeny, which can be the source of false predictions. The following section illustrates the kind of issues that may arise because of this.

2.2.3 The cost of ignoring the phylogeny

To highlight the shortcomings of multinomial when the observations are strongly correlated due to the phylogenetic structure, we generated synthetic trees that are composed of two kinds of sub-trees. Sub-trees of the first kind have very short terminal branches, and have homogeneous phenotypic condition. They are the labelled A and B in figure 2.3. The other kind of sub-tree, labelled as C, has longer branches, and transitions from background to foreground condition occur on every odd terminal branch. As a result, when simulating sites along these trees, amino acids at the tips in clades A and B are expected to be very similar, while those at the tips of clade C are more independent. A few trees with variable size are generated this way.

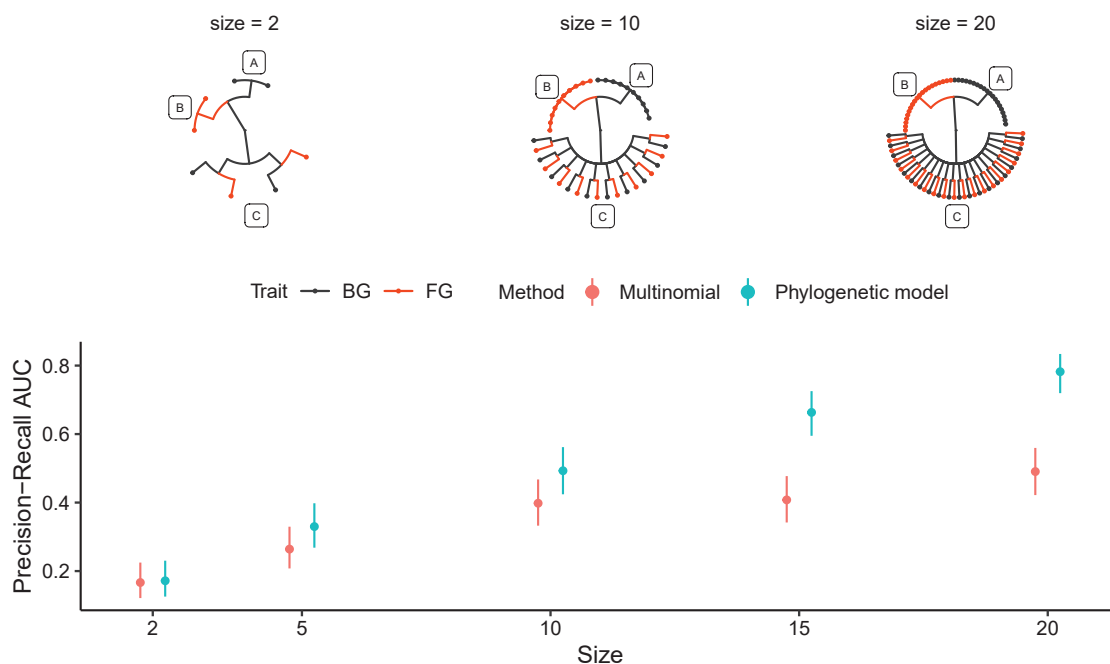


Figure 2.3: Multinomial gives equal weight to each observation, not considering how correlated they are. This leads to an increased quantity of false predictions compared to a phylogenetic approach.

The predictions made by the multinomial method make the hypothesis that observations are independent, which is not true in general because of their phylogenetic relationship, particularly when the phylogenetic divergence between observations is small. The trees that we just designed are thus expected to be very confusing for Multinomial, which will treat observations at the leaves of the sub-trees A + B in the same way as those at the tips of sub-tree C. We measure the quality of

its predictions on each tree, and compare them to the performance measured using a method that also models site composition as a response to amino acid frequency vectors, through a substitution process unfolding along the phylogeny. The identity of the phylogenetic method is not relevant to understand the point of this experiment and is not specified here for brevity².

Bottom panel of figure 2.3 shows that the performance of the phylogenetic method increases with the size of the tree and the quantity of data that is used for inference. Multinomial on the other hand does not benefit as much from increasing the number of observations. This highlights the fact that the phylogeny can not simply be ignored without harming the detection. Although our attempts at accounting for the phylogenetic structure using linear mixed models were not fruitful, they also suggest that there is potential in the Multinomial approach which makes better predictions without even considering the dependency between observations. Could we then draw inspiration from linear mixed models to represent the phylogenetic structure in Multinomial as a random effect to improve its predictions ?

2.2.4 Multinomial with phylogenetic random effect: it's complicated

Introducing a random effect in the Multinomial model to account for the phylogenetic relationship in our data is quite difficult in practice. To demonstrate why this would be hard to achieve, I present a simpler but similar model involving a random effect and attempt to perform maximum likelihood estimation of its parameters.

Let us consider an m -vector of binary variables B , which are modeled as following m Bernoulli distributions

$$B_i \sim \text{Bern}(p_i) \quad \forall i \in \{1, \dots, m\} \quad (2.26)$$

The probability of success for each variable is p_i , which can be defined as a linear response to some observed variables X_i , and a random effect α . This is done by using a logit transformation to ensure that each probability p_i is bounded within $[0; 1]$

$$\log \frac{p_i}{1 - p_i} = X_{i\bullet} \beta + \alpha \quad \alpha \sim N(0, \gamma^2 C) \quad (2.27)$$

where α is a hidden random variable whose distribution depends on the parameter γ and a correlation matrix C between observations. Parameters of the model that must be estimated are noted $\theta = \begin{pmatrix} \beta \\ \gamma \end{pmatrix}$.

The likelihood of B for one value of θ is

$$\begin{aligned} L(\theta; B) &= \log \mathbb{P}(B|\theta) \\ &= \log \int d\alpha \mathbb{P}(B, \alpha|\theta) \\ &= \log \int d\alpha \mathbb{P}(B|\alpha, \theta) \mathbb{P}(\alpha|\theta) \end{aligned} \quad (2.28)$$

An integral immediately appears, because the random effect coefficient α is a hidden random variable that we must integrate over all of its possible values. This computation is not tractable, unless an analytical expression for the value of this integral term is known, which we do not have.

This kind of inference problem where a latent variable is involved can usually be tackled using expectation-maximisation (EM) algorithms, that work in two steps: define a function for the expectation (phase E) of the log-likelihood, which requires the identification of a posterior distribution

²It is however presented with details in chapter 4.

for the latent variable, and depends on current parameter values; then compute new parameter estimates that maximize (phase M) this function. Repeating these two steps until stabilization of parameter values provides an efficient way to fit the model at maximum likelihood.


Using Jensen’s inequality, stating that $f(\mathbb{E}(X)) \geq \mathbb{E}(f(X))$ for a concave function $f(X)$, we can define a lower bound for the likelihood function such that

$$L(B, \theta) \geq \mathbb{E}_{\alpha \sim q}(\log \mathbb{P}(B|\theta, \alpha)\mathbb{P}(\alpha|\theta)) \quad (2.29)$$

which depends on a distribution q for the variable α . The closest bound is obtained when q is determined as the posterior distribution for α .

When dealing with discrete latent variables, it is sometimes possible to analytically derive the posterior distribution. Continuous latent variables, such as α in this model, make it harder because we have to deal with integrals instead of sums. As a consequence, we have to resort to other approaches to derive the posterior probability of the latent variable, such as Markov-chain Monte-Carlo (MCMC) sampling which involve computational costs larger than analytical solutions by orders of magnitude. This modification of the algorithm, known as Monte-Carlo expectation-maximisation (MCEM), has been implemented in previous works and possible optimizations have been researched (see e.g. [Levine and Casella, 2001]), but it remains computationally intensive.

Considering the cost of fitting the multinomial model with phylogenetic random effect — that does not describe the mechanisms of sequence evolution — I believe it is not worth it to further explore this idea. The main advantage of using linear models in our context is the high throughput that they offer, thanks to analytical solutions to fit them at maximum likelihood. However, this benefit would be lost when introducing random effects in a generalized linear model such as Multinomial. Should we consider more computationally heavy solutions, the use of more mechanistic models of the evolution process might then be more attractive than linear models, as the additional complexity is more likely to improve the quality of predictions, with stronger theoretical foundations.

 Chapter 2 summary: *Is this basically inter-species GWAS?*

We asked one naive question: can methods inspired from genome-wide association studies (GWAS) be applied at the inter-specific scale to perform high throughput detection of genotype-phenotype associations?

To answer it we first attempted to apply GEMMA, an existing implementation of a linear mixed model used in GWAS, to our problem and found it was not well suited to this purpose. This model works with binary genotype markers, which makes it difficult to represent amino acid states. We designed a variation on the model of GEMMA with the expectation that it would be better adapted to handle amino acids, but found that it does not improve on the performance obtained with GEMMA.

These approaches based on linear mixed models were compared to a model of the amino acid composition at one site using a multinomial distribution, which was initially described in [Rey et al., 2019]. We find that the Multinomial model is generally better than both linear mixed models. After highlighting how its lack of accounting for phylogenetic relationships can be a source of false predictions, we explored the possibility to improve it by incorporating the phylogenetic structure into it as a random effect, but found that it can not be done easily. This would also involve prohibitive computation costs, which defeats the purpose of using such simple approaches in the first place, and motivates an exploration of phylogenetic models of sequence evolution.

Chapter 3

Evaluation of methods to detect shifts in directional selection at the genome scale

After some attempts at using linear models with phylogenetic random effect, we do not have yet a satisfactory solution to reliably detect genotype-phenotype associations at large scale. We found that a model based on a multinomial distribution of amino acid counts that is completely unaware of the phylogeny works better than linear mixed models, but that ignoring the phylogeny is a non-negligible source of false predictions. This motivates us to delve into the use of more mechanistic models that describe the evolution of sequences along the phylogeny.

Phylogenetic models of sequence evolution are diverse, therefore we need to assess which kind of approach provides us with the best trade-off between the quality of predictions and throughput so that large scale datasets can be handled. Several models have been previously evaluated on their ability to detect sites that have undergone convergent evolution within alignments [Rey et al., 2019]. Among them, the Multinomial model was included, as well as phylogenetic models that operate either at the level of amino acids (TDG09 [Tamuri et al., 2009], PCOC [Rey et al., 2018]), or at the level of codons in the mutation-selection framework (Diffsel [Parto and Lartillot, 2017]). All of these models build on representations of amino acid profiles: fitness profiles in the case of Diffsel, frequency profiles for Multinomial, TDG09 and PCOC. In these investigations, Diffsel was reported to be largely better at the task of detecting convergent sites, but involves computation times too high to enable its application at large scale. PCOC and TDG09 were found to be better than Multinomial, which in turn was better than other simple methods.

We compare these methods to those based on linear mixed models using our evaluation pipeline, and include as well `codeml` [Yang, 2007], an implementation of a codon-based model that compares synonymous and non-synonymous substitution rates (d_N/d_S). Precision-recall performance is evaluated for each method in a variety of settings, using both empirical and synthetic phylogenies, and assessing the robustness of each model to the presence of confounding factors.

Our own implementation of the TDG09 model, which we name Pelican, was also included in this benchmark. It was made as a first step towards increased complexity, starting from Multinomial: TDG09 can be summarized as a phylogenetic version of Multinomial. It was originally intended to be progressively modified to better understand the reported performance gap between TDG09 and Diffsel, but the results we obtained made us change our plans: the performance of Pelican was

surprisingly good, reaching levels comparable to those of Diffsel and `codeml`, as we report in the following article.

Contents

3.1 Introduction	53
3.2 New Approaches	57
3.3 Results	57
3.3.1 Detection performances on synthetic trees	59
3.3.2 Detection performances on empirical phylogenies	61
3.4 Discussion	65
3.4.1 Mutation-selection models for simulating coding sequences	65
3.4.2 Methods working at the amino acid level perform as well as codon-based methods	66
3.4.3 Features of a data set that affect performances	67
3.4.4 GC-biased gene conversion is an important confounding factor for both dN/dS and profile methods	67
3.4.5 Persistent positive selection is an important confounding factor for dN/dS methods, less so for profile methods	68
3.4.6 Interpreting screens for changes in directional selection	68
3.4.7 Looking forward	69
3.5 Conclusion	70
3.6 Methods	70
3.6.1 Detection of dN/dS variations using <code>codeml</code>	70
3.6.2 Multinomial method	70
3.6.3 Pelican: improvements on TDG09	71
3.6.4 Simulations	71
3.6.5 gBGC simulation	72
3.6.6 CpG simulation	72
3.6.7 Simulation of persistent positive selection	73

Louis Duchemin¹, Vincent Lanore¹, Philippe Veber¹, Bastien Boussau¹

¹Univ Lyon, Univ Lyon 1, CNRS, VetAgro Sup, UMR5558, Laboratoire de Biométrie et Biologie Evolutive, F-69100, Villeurbanne, France

<https://doi.org/10.1093/molbev/msac247>

Abstract

Identifying the footprints of selection in coding sequences can inform about the importance and function of individual sites. Analyses of the ratio of non-synonymous to synonymous substitutions (d_N/d_S) have been widely used to pinpoint changes in the intensity of selection, but cannot distinguish them from changes in the direction of selection, *i.e.*, changes in the fitness of specific amino acids at a given position. A few methods that rely on amino acid profiles to detect changes in directional selection have been designed, but their performance have not been well characterized. In this paper, we investigate the performance of 6 of these methods. We evaluate them on simulations along empirical phylogenies in which transition events have been annotated, and compare their ability to detect sites that have undergone changes in the direction or intensity of selection to that of a widely used d_N/d_S approach, `codeml`'s branch-site model A. We show that all methods have reduced performance in the presence of biased gene conversion but not CpG hypermutability. The best profile method, Pelican, a new implementation of [Tamuri et al., 2009], performs as well as `codeml` in a range of conditions except for detecting relaxations of selection, and performs better when tree length increases, or in the presence of persistent positive selection. It is fast, enabling genome-scale searches for site-wise changes in the direction of selection associated with phenotypic changes.

3.1 Introduction

The genomes and phenotypes of extant species bear traces of past adaptations that occurred in their ancestors. A lot of research in molecular evolution has been devoted to detecting and interpreting these traces, both in non-coding and coding sequences (e.g., [Moretti et al., 2014, Zhang et al., 2014, Merényi et al., 2020, Partha et al., 2019, Marcovitz et al., 2019]). In protein-coding genes in particular, several approaches have been developed to study evolution at the level of whole genes or at the level of single sites [Goldman and Yang, 1994, Yang and Nielsen, 2008, Penn et al., 2008, Pupko and Galtier, 2002, Abhiman and Sonnhammer, 2005]. Studies have found that amino acid changes at a single position could create an active site *de novo* [Risso et al., 2017], that amino acid changes at a few positions could change the affinity of an hormone receptor for its ligand [Bridgham et al., 2006], that changes in rates of evolution accompanied the appearance of new HIV subtypes [Penn et al., 2008], that convergent evolution could be detected at single sites in proteins in mammals [Li et al., 2010], in grasses [Christin et al., 2007], in insects [Zhen et al., 2012], and that amino acid changes at a single position could alter the dynamic of a worldwide viral epidemic [Korber et al., 2020]. Identifying traces of past and current adaptations at the level of single amino acid sites can thus be very insightful. In this article, we investigate the performance of several methods aiming to do just that. These include one commonly-used d_N/d_S method, but also methods that have been more recently developed, based on *amino acid fitness profiles*. Although other approaches based on the detection of shifts in the rate of sequence evolution have also been used to identify coding sites that have undergone selective pressure changes [Penn et al., 2008, Gu et al., 2013, Pupko and Galtier, 2002, Abhiman and Sonnhammer, 2005], they have not been evaluated in this study. They

have been used less often than d_N/d_S methods, and have not been specifically designed to study the type of changes investigated in this study.

In proteins, amino acids that are never or seldom encountered at a particular site in a group of related species may have been selected against in the past. Those that are frequent may have been favored by selection. One can study these differences in frequency to infer differences in fitness between amino acids. A *fitness profile* is then used to represent the relative fitness of each amino acid at a given site (fig. 3.1a: A, B, C and C'). When used within models of sequence evolution, a fitness profile determines the fixation probability of arising mutations during the process of evolution through mutation and selection [Halpern and Bruno, 1998, Yang and Nielsen, 2008, Rodrigue et al., 2010, Tamuri et al., 2012]. It also provides a *direction* for selection, which pushes evolution at the site away from low-fitness amino acids and towards high-fitness amino acids.

The shape of a fitness profile derives from selective pressures that operate at a particular site of a protein. These pressures can be related to phenotypic traits or environmental constraints, which could change over time. In such a case, the pressures would change, and so would the fitness profile. Selection may vary in intensity, for instance as a trait becomes more or less important for the global fitness of the organism; and in direction, when changing the value of a trait leads to higher fitness. These different kinds of changes in selective pressure can be captured by variations of the fitness profile: changes in intensity through the pointedness or flatness of the profile (fig. 3.1a, transition from profile A to profile B), and changes in direction through the variation of the overall shape of the profile (fig. 3.1a, transition from profile A to profiles C and C'). In this manuscript we will focus on trait changes and the associated fitness profile at a site that occur discretely, at once, but progressive, continuous changes certainly occur in nature and would be important to consider.

Approaches to detect variations of selection on single sites of protein-coding sequences all require an annotation of the branches of a phylogeny, whereby each branch is associated to a phenotypic state or environmental condition. Given this annotation, either d_N/d_S or profile methods can be used (fig. 3.2). Other approaches that do not require such an annotation of branches have been proposed (e.g. [Guindon et al., 2004, Dutheil et al., 2012, Murrell et al., 2012a, Murrell et al., 2015]), but they are not evaluated in this work.

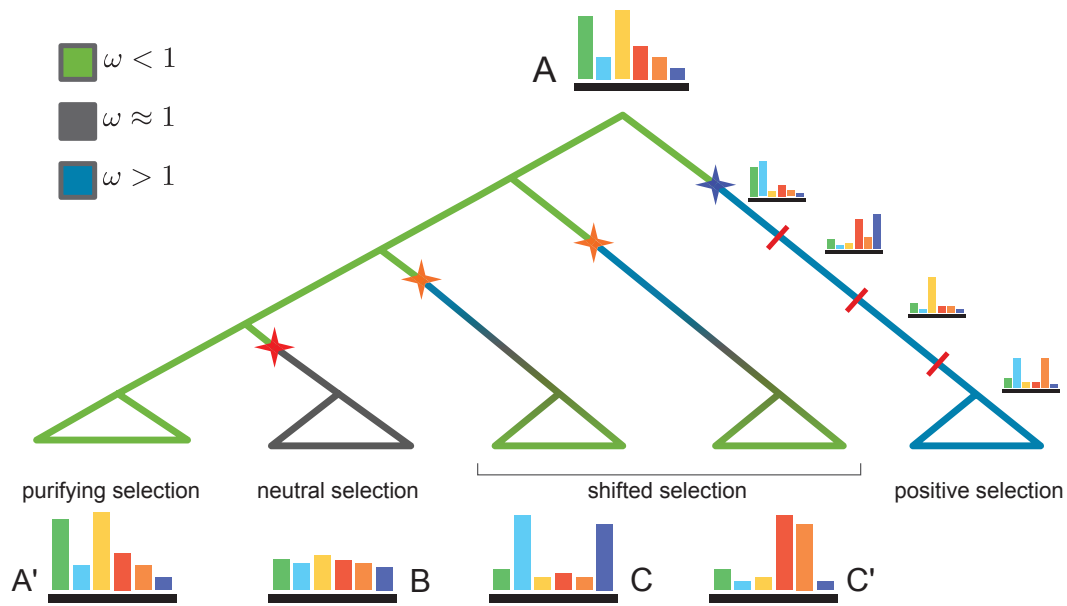
Approaches relying on the $\omega = d_N/d_S$ metric have been widely used to capture variations in selective pressure [Kosiol and Anisimova, 2019a], including in the context of genome screening (e.g. [Nielsen et al., 2005, Kosiol et al., 2008, Studer et al., 2008, Moretti et al., 2014, Zhang et al., 2014]). These methods can show good reliability, either at the level of whole gene sequences or of single sites, when the generating process matches the inference model (e.g., [Zhang et al., 2005]). The ω metric is defined as the ratio of rates between non-synonymous (d_N) and synonymous (d_S) substitutions. The underlying assumption is that selection operates at the amino acid level, so that synonymous codons provide the same fitness, while non-synonymous substitutions induce a variation in fitness. In the popular "branch-site model A" available in PAML, inference is performed at the level of a gene by comparing the likelihood of a model with one set of d_N/d_S values per condition, against a model having one global set of d_N/d_S values through a likelihood ratio test (LRT) [Yang, 2007]. At the site level, the gene-wise parameter estimates are used to identify sites whose d_N/d_S has changed in a manner correlated with the annotation of the phylogeny [Yang et al., 2005]. However, other implementations have been proposed (e.g., [Kosakovsky Pond et al., 2011, Murrell et al., 2015]). All these d_N/d_S methods should be particularly effective at inferring changes in the intensity of negative selection: weaker (respectively stronger) selection should result in higher (resp. lower) d_N/d_S values. In that sense, d_N/d_S values have been used as a proxy of selection efficiency, even though in some cases this can be misleading [Spielman and Wilke, 2015, Jones et al., 2019]. In

particular, under a constant amino acid fitness profile, shifts between amino acids with non-0 fitness can result in cases of transient $d_N/d_S > 1$ [Jones et al., 2017], often interpreted as signalling positive selection on a particular branch. In addition, d_N/d_S methods should have good power to detect cases of persistent positive selection, *i.e.*, positive selection operating on all branches of the phylogeny, (rightmost branch, fig. 3.1a), which should result in high d_N/d_S values. However, they might be less effective at detecting changes in the direction of selection [Parto and Lartillot, 2018], as they may fail to detect some sites that have undergone episodic changes in directional selection on top of a background of strong purifying selection (see fig. 3.1 and [dos Reis, 2015]). Further, they do not output estimates of the direction of selection, but only d_N/d_S values.

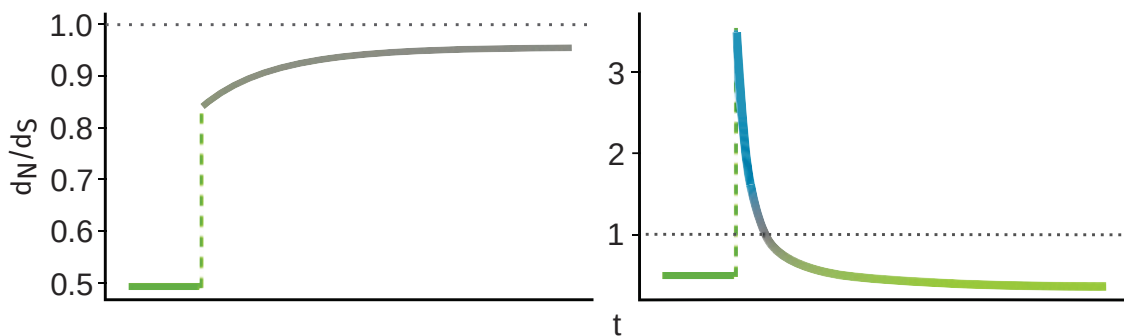
Profile methods have been developed more recently than d_N/d_S methods, and have yet to be used at a genomic scale. They all rely on amino acid profiles to identify sites that correlate with a phenotype along a phylogeny, but vary in the complexity of their underlying models. Some methods operate at the codon level and can explicitly use amino acid fitness profiles by distinguishing between the mutation process, operating at the nucleotide level, and the selection process operating at the amino acid level (e.g., [Murrell et al., 2012a, Parto and Lartillot, 2018]). These methods build on the *mutsel* framework [Halpern and Bruno, 1998, Yang and Nielsen, 2008, Rodrigue et al., 2010, Tamuri et al., 2012, Bloom, 2014] that provides a better description of coding sequence evolution than d_N/d_S approaches [Spielman and Wilke, 2016, Bloom, 2014]. Other methods operate at the amino acid level and thus cannot model the mutation process. They use amino acid *frequency* profiles as a proxy to *fitness* profiles [Tamuri et al., 2009], and may thus be less powerful than methods that operate at the codon level. In both cases, inference can be performed with a likelihood ratio test (LRT) at the site level, comparing the likelihood of a model with one profile per condition, against a model having one single profile that applies on all branches of the phylogeny. Such an approach can be problematic [Rodrigue, 2013]: there is a limited amount of information available in a single site to estimate one parameter per amino acid, or even several parameters if several profiles need to be considered. Unobserved amino acids typically are assigned a fitness or equilibrium frequency of 0, which is unrealistic. Further, they do not contribute to the computation of the degrees of freedom when doing the LRT chi-square test, which can make it anti-conservative. To mitigate some of these problems, approaches based on penalized likelihood have been proposed [Tamuri et al., 2014]. In addition, Bayesian approaches that treat site-wise amino acid profiles as a mixture distribution have also been used [Rodrigue et al., 2010, Rodrigue, 2013, Rodrigue and Lartillot, 2017, Rodrigue et al., 2020], including for a branch-heterogeneous *mutsel* model [Parto and Lartillot, 2018]. However, when comparing Maximum Likelihood, penalized, and Bayesian approaches, it was found [Spielman and Wilke, 2016] that these methods often agreed in their estimates of site-wise selective constraints. In this manuscript, we evaluate two Maximum Likelihood methods (TDG09 and Pelican), and one Bayesian mixture approach (Diffsel).

Both d_N/d_S and profile methods to detect changes in selective pressures could be misled by non-adaptive processes, or by confounding between different selection regimes [Jones et al., 2019]. Non-adaptive processes notably include GC-biased gene conversion (gBGC) [Ratnakumar et al., 2010, Bolívar et al., 2019] and CpG hypermutability [Meunier et al., 2005]. gBGC occurs during recombination and mimics natural selection by favoring the fixation of G and C alleles. CpG hypermutability increases the mutation rate of CG dinucleotides. These processes can generate patterns in the sequence data that could lead to false positives or false negatives, as has been shown for d_N/d_S methods with respect to both gBGC [Ratnakumar et al., 2010, Guéguen and Duret, 2018, Rousselle et al., 2019], and CpG hypermutability [Saunders and Green, 2007, Suzuki et al., 2009]. Confounding between different selection regimes could happen if a test aiming to find changes in the direction of

Figure 3.1: Schematic representation of various evolution scenarii of a protein site involving profile changes. Colored stars indicate transition events that trigger profile changes. The color gradient along branches show the variation of d_N/d_S a.k.a ω values. The green sub-tree is a case of purifying selection, with fixed profile (A) and $\omega < 1$. The grey sub-tree illustrates relaxed pressure subsequent to the transition in red, resulting in a flattened profile B and $\omega \approx 1$. Two cases of shifted selection are represented, each one driven by a different fitness profile (C and C'). In both cases, there is a transient increase in the value of ω , followed by a decrease towards $\omega < 1$, as represented on the right panel in 3.1b. Blue sub-tree is an example of persistent positive selection [Tamuri, 2021], where the fitness profile rapidly changes along the branch, at intervals marked with red bars. In this case, the value of d_N/d_S remains greater than 1 while positive selection continues.



(a) Cases of selection regime changes, with their representation as fitness profile changes, and their equivalence with the d_N/d_S metric.



(b) d_N/d_S variations over time. The curve on the left represents the simulated value of d_N/d_S when transition from purifying selection to relaxed selection occurs (transition between profiles A and B above). On the right is the variation of d_N/d_S during a shift in selection direction (transition between profiles A and C or C' above).

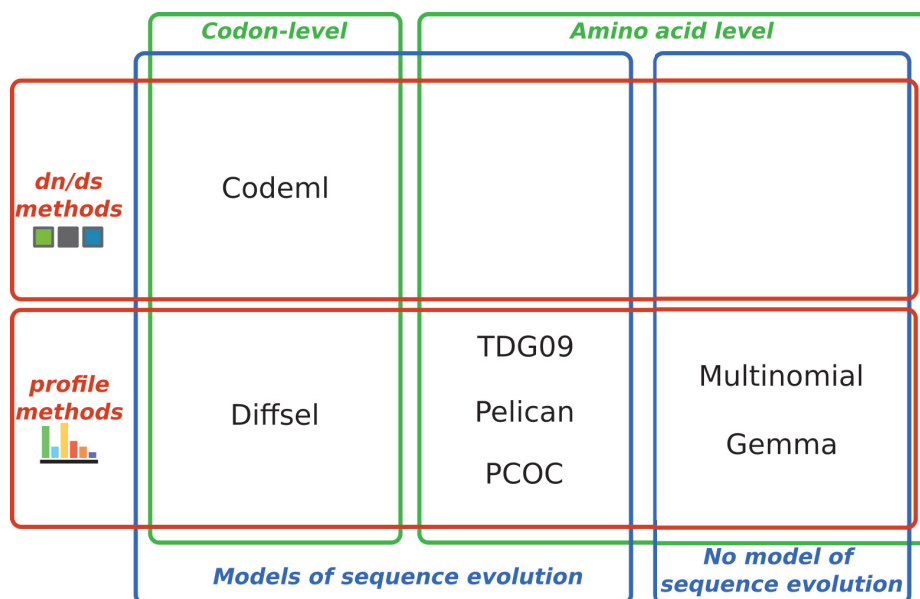


Figure 3.2: Methods evaluated in the manuscript. Methods have been positioned based on whether they are based on d_N/d_S or amino acid profiles, whether they work at the codon or amino acid level, and whether they rely on a model of sequence evolution running along a phylogeny or not.

selection detected sites under persistent positive selection. It is unclear how sensitive profile methods would be to these problems.

Genome-scale detection of changes in selective pressure requires a fast method. Firstly, there can be thousands of gene families that each need to be analyzed with the method. Secondly, using a large number of species can increase the power of an analysis, but also increases its computational cost. In fact, it has been suggested that the high computational cost of d_N/d_S methods may be a hurdle to their more widespread use [Davydov et al., 2019]. It is unclear how efficient profile methods could be.

In this article, we evaluate several profile and d_N/d_S methods to detect changes in selective pressures operating on individual positions of a protein-coding gene, on specific branches of a phylogeny. We consider several profile methods that have been published or that we have developed *de novo*, and compare them to a widely-used d_N/d_S method. In particular, we ask whether profile methods can be as powerful as the d_N/d_S method, including in the presence of confounding factors, and pay particular attention to the computational costs of all methods.

Performance measurements are done on simulated datasets, allowing us to characterize the behaviour of the methods on a range of tree shapes, branch lengths, and number of transitions along the phylogeny. We also investigate whether the detection methods are sensitive to confounding signal generated by non-adaptive processes of molecular evolution [Ratnakumar et al., 2010, Bolívar et al., 2019, Meunier et al., 2005], or by persistent positive selection [Tamuri, 2021].

3.2 New Approaches

In this article, we introduce Pelican, an improved implementation of the model from [Tamuri et al., 2009]. This implementation was found to have better sensitivity and specificity than the original, and is also faster thanks to optimisations on linear algebra computation.

Multinomial is a fast non-phylogenetic profile method that is also evaluated in this paper. It

models observed amino acid frequency profiles as multinomial distributions, and compares the likelihoods at a given site of a single frequency profile versus multiple profiles through a likelihood ratio test.

Both of these methods are implemented as a single program, that is made available to detect differential selection in protein sequence alignments. In this context, Multinomial can be used as a fast filter on the alignment to reduce the amount of candidate sites to be evaluated through Pelican.

3.3 Results

We evaluated the performance of detection methods using simulated datasets. The methods that were considered are represented in fig. 3.2 and include:

- `codeml`, a widely used d_N/d_S method for detecting positive selection, provided in the PAML toolkit [Yang, 2007]. `codeml` was configured to use the branch-site model A [Zhang et al., 2005, Yang et al., 2005], and works at the codon level.
- Multinomial, the simplest and fastest profile method, does not rely on a model of sequence evolution and works at the amino acid level. It uses a likelihood ratio test (LRT) to compare two models, one in which a single amino acid profile is used to describe amino acid frequencies observed at a site across all tip sequences, and one where different amino acid profiles are used depending on the condition associated to the tip. Multinomial ignores the shape of the phylogeny and could thus be misled by phylogenetic inertia.
- Gemma [Zhou and Stephens, 2012], based on a linear mixed model, was originally developed for genome-wide association studies (GWAS). It does not use a model of sequence evolution, but can take into account the structure of the phylogeny, encoded as a correlation matrix, which is introduced as a random effect in the mixed model. We used it at the amino acid level, by encoding the protein alignment as an alignment of binary characters (see Methods). The phenotypes of species were encoded from the state of each tip taxon.
- TDG09 [Tamuri et al., 2009], a profile method that can be considered as a refinement over the Multinomial method, in that it also works at the amino acid level but takes into account the phylogeny by relying on a model of sequence evolution. It uses a LRT to compare a model with one profile per condition and a model with one single global profile.
- Pelican, a new implementation of the model underlying TDG09 [Tamuri et al., 2009], originally motivated by the observed discrepancy reported between the performances of Diffsel and TDG09.
- PCOC [Rey et al., 2018], a profile method working at the amino acid level. It is at its base similar to TDG09 but works with a limited set of pre-existing profiles, and further expects to observe substitutions at every transition between conditions in the phylogeny.
- Diffsel [Parto and Lartillot, 2018], a profile method working at the codon level and based on a mutation-selection model in a Bayesian framework. Diffsel has performed significantly better than the other methods in a previous benchmark [Rey et al., 2019].

All simulations were done under a codon-based, time-reversible, mutation-selection model with site-specific amino acid fitness profiles. The model was run along a phylogeny whose branches are annotated with two conditions that we refer to as *background* and *foreground*. A simulation generates

codon and corresponding amino acid alignments of arbitrary length. Sites in the alignment may be either: (1) H_A sites, that are the result of a simulation where changes in the selection dynamic occur between background and foreground branches; (2) H_0 sites, resulting from an evolutionary process where selection is constant. The number of sites of each type was controlled in the simulation, allowing the comparison of predictions on the nature of each site (H_0 or H_A) with its known type, to estimate the performances of the prediction method.

Performance estimates in all the benchmarks were done using two metrics: *precision* and *recall*. Precision is the proportion of true positive sites among all sites identified as positive. Recall, also known as sensitivity, is the proportion of H_A sites that are identified as positive. These metrics were summarized by computing the area under the precision-recall curve (PR AUC). Confidence intervals for the PR AUC were computed according to [Boyd et al., 2013].

In this section we compared the detection methods using our simulation model in several contexts: (1) synthetic trees with variable branch lengths and numbers of transitions; (2) empirical trees in the presence or absence of confounding factors in the simulation. In the following, all branch length values are given in expected numbers of codon substitutions.

3.3.1 Detection performances on synthetic trees

To characterize the behaviour of the methods with respect to the number of transitions, the time spent in a condition, and branch lengths, we relied on synthetic trees with carefully controlled features.

Detection performances increase with the number of transitions

We investigated whether the number of transitions from background to foreground conditions had an effect on detection performances. We generated a balanced tree of 128 tips in which all branch lengths equal 0.01, and generated a variable number of transitions on terminal branches (tree topology shown in sup. fig. C.1). In this setting, both the number of foreground leaves and the total time in the foreground condition increase with the number of transitions. Results shown in panel a of fig. 3.3 show that all methods take advantage from such increases.

The amount of time spent in a condition has a large effect on detection performance for phylogenetic methods

We next evaluated the relative importance of the number of transitions and the amount of time spent in the foreground condition on the phylogeny. We used a different set of trees with the same general features (128 tips, branch lengths equal 0.01), varied the numbers of transitions, but kept the number of foreground leaves and total foreground length constant across trees. This was done by normalizing the branch lengths to achieve equal total times between foreground and background conditions, and across trees. As a result the number of tips in each foreground sub-tree is variable, depending on the depth of the transition event. For a given number of transitions, all transitions occur at the same depth in the tree (sup. fig. C.2).

Panel b of figure 3.3 shows that the performances of Gemma and Multinomial increase with the number of transitions, even when the amount of time spent in the foreground condition is kept constant. They become the best performing methods at 64 transitions. However, the phylogenetic methods `codeml` and Pelican seem to be less sensitive to this parameter in this experiment, suggesting that the determining factors for their performances in the previous experiment were the total foreground time and/or the number of foreground leaves, which are kept constant in this experiment.

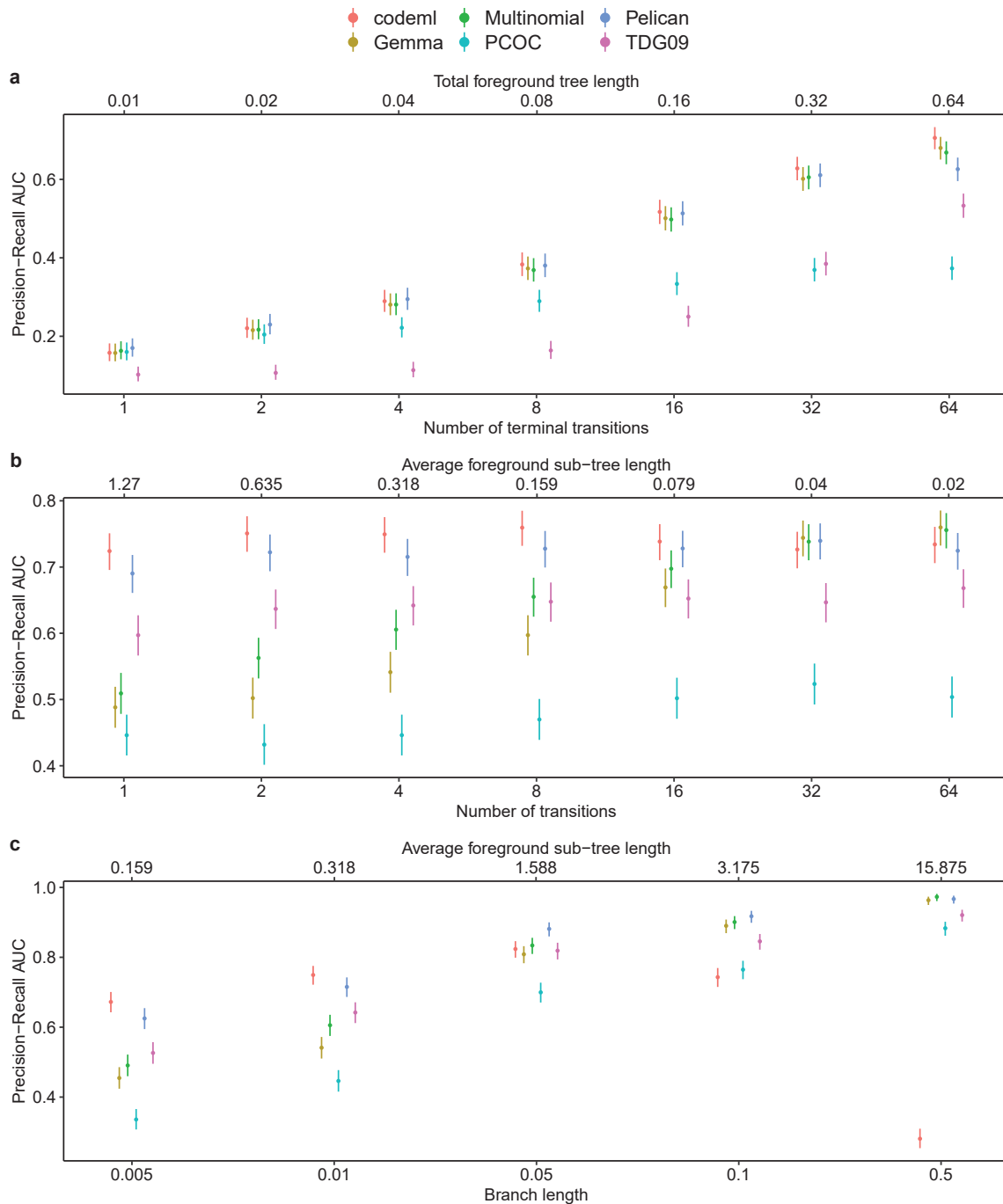


Figure 3.3: Detection performances evaluated on synthetic trees. 95% confidence intervals accounting for the variability of the PR AUC estimates are shown. (a) Performance increases with the number of transitions on terminal branches. (b) The number of transitions is not the determining factor for the performance of the phylogenetic methods but has a strong effect on the performance of Gemma and Multinomial. (c) Performances of the profile methods are positively correlated to the branch lengths, while the performance of *codeml* decreases on longer branches.

Profile methods improve as branch lengths increase

In order to assess the effect of branch lengths and of the distance between transition events and foreground leaves on method accuracy, while keeping the number of transitions constant, we evaluated each method on a balanced tree with 4 transition events where a scaling factor was applied to the branch lengths (sup. fig. C.3). As a side-effect, this scaling factor also applies to the total foreground tree length.

Results in panel c of figure 3.3 highlight two opposite trends between profile methods and the d_N/d_S method `codeml`, in relation with the branch length scaling. Profile methods tend to be more accurate in detecting selection shifts when the branch lengths increase, while the performance of `codeml` decreases. We suspect that as branch lengths increase, the number of synonymous substitutions increases, which reduces d_N/d_S and makes it harder to detect H_A sites (see fig. 3.1b, right).

Among profile methods, the performance gap tends to decrease with longer branches.

3.3.2 Detection performances on empirical phylogenies

To benchmark the methods in a more realistic context, we evaluated their performances on empirical phylogenies that differ in their size, depth and number of transitions (Table 3.1). The corresponding phylogenetic trees are shown as supplementary material (supplementary figures B.4, B.2, B.1, B.3, B.5, B.6).

Alignments were simulated as in the previous experiments, using the simulation model running along the empirical phylogenies. These alignments were used to measure the statistical calibration and the throughput of each method, and to evaluate each method as in the previous section.

Dataset	Depth	Size	Transitions	Avg branch length		Avg sub-tree length
				Global	Foreground	Foreground
Rodents [Rey et al., 2019]	11	32	10	0.0192	0.0252	0.0353
<i>Cyperaceae</i> [Besnard et al., 2009]	25	79	5	0.0207	0.0239	0.196
Echolocation [Scornavacca et al., 2019]	18	116	3	0.0081	0.0061	0.0828
<i>Amaranthaceae</i> [Kapralov et al., 2012]	22	179	15	0.0045	0.0035	0.0356
HIV RTi [Murrell et al., 2012a]	34	476	238	0.0063	0.0051	0.0051
Influenza H1 segment [Tamuri et al., 2009]	61	434	1	0.0603	0.0608	49.0709

Table 3.1: Summary statistics on empirical trees. Tree depth is defined here as the highest number of branches between a leaf and the root. Size is the number of leaves in the tree. Transitions are defined as changes from the background to the foreground condition.

Pelican performs well on empirical phylogenies

We assessed whether the methods were well calibrated, *i.e.*, how accurate was their reported false positive rate under the null (H_0) model. To this end, we simulated 9 000 sites under H_0 , and counted the number of false positives for each method, at the 0.05 p -value threshold. Under this setting, a well calibrated method should produce on average a number of false positives equal to

5% of the total number of sites. Results shown in sup. table C.1 indicate that most methods are overly conservative, *i.e.*, their observed false positive rate is lower than their advertised (5% here) false positive rate. Multinomial is the only method that can yield a higher rate of false positives, particularly on the Influenza phylogeny. To further assess how conservative the methods were, we computed the observed false positive rate on non-constant sites only, given that constant sites cannot be classified as positive. Sup. table C.2 indicates that even on this subset of sites, most methods still have low rates of false positives. This indicates that all methods except Multinomial are overly conservative.

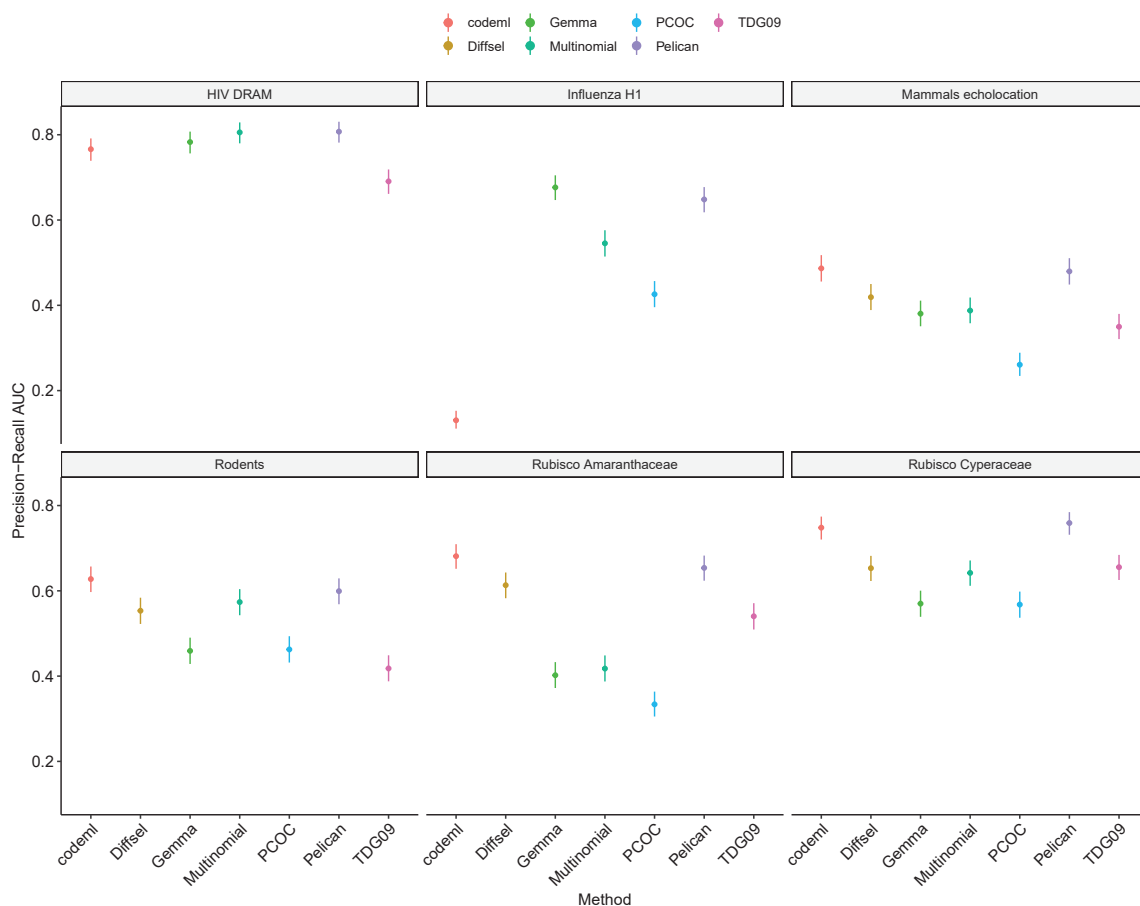


Figure 3.4: Precision-Recall area under the curve (AUC) estimates on simulated datasets using 6 empirical phylogenies, under changes in the direction of selection. Performances of TDG09 on the Influenza H1 dataset were not successfully measured. Diffsel was not evaluated on the HIV and Influenza dataset due to the large computation times involved. PCOC had an underflow error on the HIV data set.

We then assessed the performance of the methods to detect H_A sites by simulating 1000 H_A sites and 9000 H_0 sites. Pelican, `codeml` and Diffsel consistently show the best performances on all datasets (fig. 3.4), with the exception of the Influenza H1 dataset. It is worth noting that Diffsel is one of the best performing methods, even though it estimates branch lengths and does not get them as input, like most other methods.

We note that, while Pelican is essentially a reimplemention of TDG09, it shows significantly better performances on every dataset. `codeml` and Pelican have similar performances in general. However, on the Influenza H1 dataset, which has the highest average foreground sub-tree length

($\bar{d} = 49.0709$), `codeml` incurs a large drop in its performances. These observations are consistent with the results obtained on synthetic trees (fig. 3.3c).

Even though the HIV dataset has the lowest average foreground sub-tree length ($d = 0.0051$), Pelican performs better than `codeml` on this dataset. Performances are strongly increased for all methods on this dataset, compared to the other empirical phylogenies. Our explanation for the results on the HIV dataset involves multiple effects: (1) the large number of transitions ($n = 238$) on terminal branches yields a strong signal for all methods, which benefits profile methods the most (see fig. 3.3a); (2) figure 3.3c seems to indicate that there is an optimal branch length for `codeml`: the signal for d_N/d_S falls off on longer branches, but branches can also be too short to allow reliable d_N and d_S estimations because of the insufficient number of substitutions occurring in such a short time span.

We showed that some characteristics of the phylogenies had a major effect on method performance, particularly the time spent in the foreground condition, as well as the number of transitions in the phylogeny. It is likely that variations in the detection performances are the results of interactions between the features of the phylogeny, possibly including more than the two we identified, as well as the sensitivity of the detection method to these features.

On a side note, we remark that Multinomial shows some surprisingly good performances despite its simplicity. As it does not take any information from the phylogeny, it is the simplest profile method, and also the fastest (table 3.2). However, experiments on synthetic trees show that in cases where phylogenetic structure creates a lot of phylogenetic inertia, the performance of Multinomial can be strongly reduced (fig. 2.3).

Performances in the detection of changes in the intensity of selection

Profile methods are in principle particularly appropriate for detecting changes in the direction of selection, and in practice perform as well as `codeml` and better on long branches (see above). We evaluated how they perform in the presence of a change in the intensity of selection, by simulating a scenario of relaxation of selection. In this scenario, H_A sites are simulated such that all amino acids have equal fitness on foreground branches. This corresponds to a complete relaxation of selection.

Fig. 3.5 indicates that profile methods, and Pelican in particular, can also detect relaxations of selection, but that their performance depends on the phylogeny. In particular, we find that in some cases the detection is unreliable (fig. 3.5, Influenza panel). We suspected that this lower performance was due to a lack of sensitivity, and tested this hypothesis by changing the computation of degrees of freedom in the likelihood ratio test performed in Pelican (sup. section C.4). Sup. fig. C.8 shows that much better performances can be obtained on the Influenza data set, but with some cost on the performance of the method on other data sets (notably Mammals echolocation). Future work on the LRT may result in an improved performance of Pelican across data sets, in settings of changes in both the direction and the intensity of selection.

Performances in the presence of confounding factors

In order to assess the robustness of the detection to other evolutionary processes, we executed a benchmark on simulations including confounding factors: (1) CpG hypermutability, which induces a higher mutation rate on methylated CpG dinucleotides; (2) GC-biased gene conversion (gBGC), a non-adaptive process that increases the overall GC content in the genome and may be mistaken as a selective force [Ratnakumar et al., 2010]; (3) persistent positive selection (PPS), as modeled by [Tamuri, 2021], which favors non-synonymous substitutions over synonymous ones on the branches

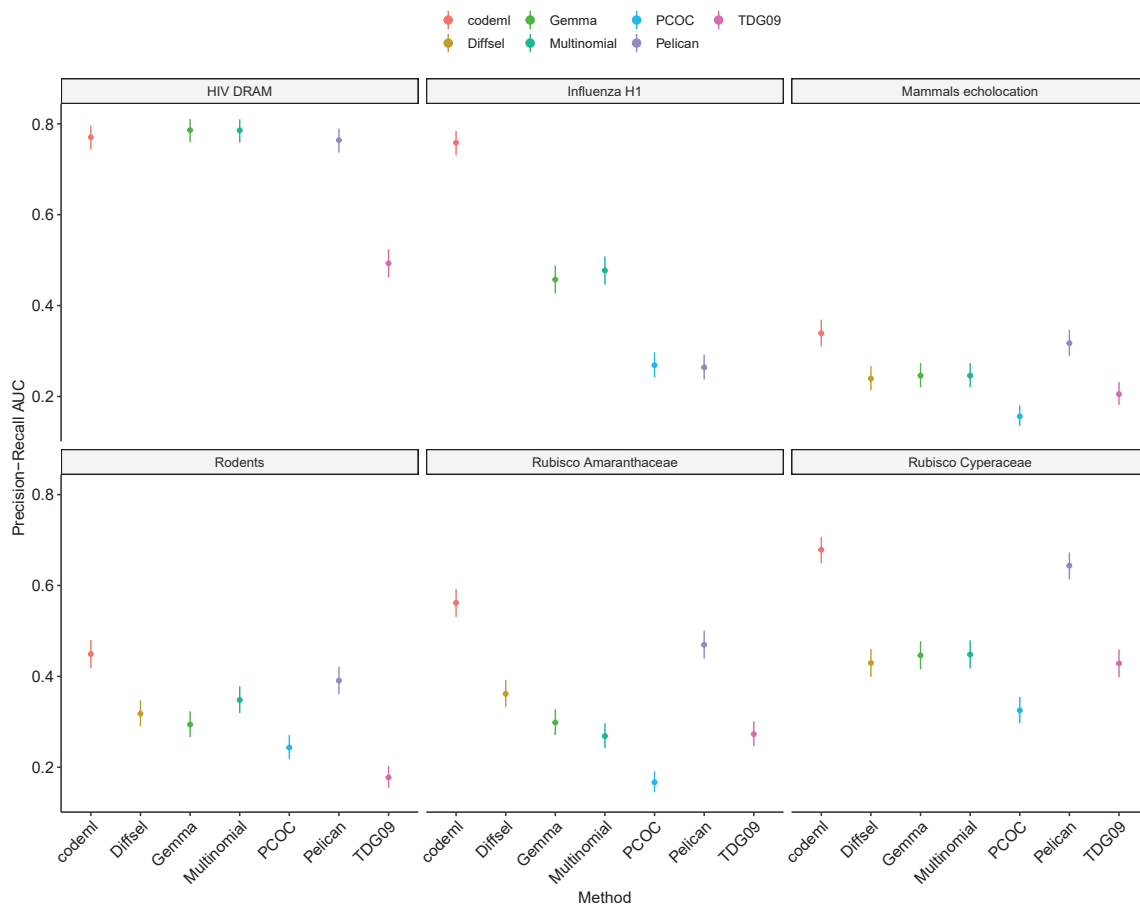


Figure 3.5: Precision-Recall area under the curve (AUC) estimates on simulated datasets using 6 empirical phylogenies, under relaxation of selection. Performances of TDG09 on the Influenza H1 dataset were not successfully measured. Diffsel was not evaluated on the HIV and Influenza dataset due to the large computation times involved. PCOC had an underflow error on the HIV data set.

where it occurs. We used strong but realistic intensities for each of these processes, with two intensities for gBGC, and two intensities for PPS. In simulations of CpG hypermutability and GC-biased gene conversion (gBGC), the processes were applied on foreground branches for both H_0 and HA sites. In the simulation of PPS, the process was applied on all branches, but only on H_0 sites, to assess the propensity of each method to generate false positives. Results are shown in figure 3.6 for the Echolocation phylogeny, and are available as supplementary material for the other phylogenies.

We find that the presence of CpG hypermutability has no influence on the detection performance in most cases.

In contrast, on simulations including gBGC, we notice a strong decrease of the performance for every method. While gBGC happens at the nucleotide level, it generates selection-like signal at the codon (or amino acid) level, that is not the result of an adaptive process. This signal was strong enough to directly interfere with the signal for selection on genomic sites.

At a fixed effective population size N_e , an increase in PPS results in a decrease in the performance of all methods. Fig. C.7 shows that this decrease is mostly due to an increase in the number of false positives. Under conditions of strong PPS and large N_e , the performance of *codeml* is strongly reduced, but the performance of profile methods can be improved. Overall, profile methods seem less prone to generating false positives in the presence of PPS, the effect of which is largely compensated

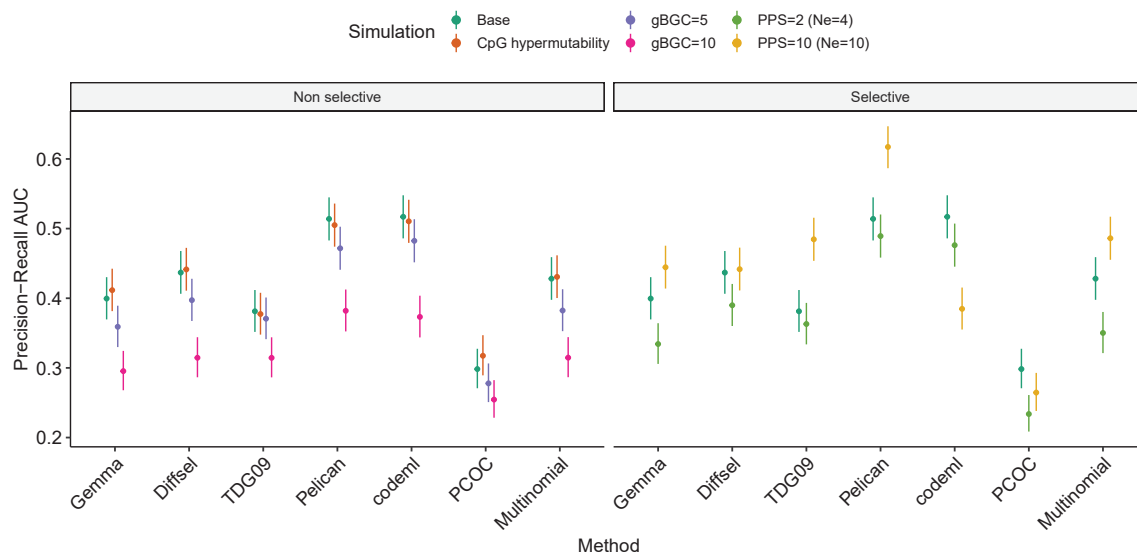


Figure 3.6: Effects of GC-biased gene conversion (gBGC), CpG hypermutability and persistent positive selection (PPS) on precision-recall AUC on the Echolocation dataset.

by an increased value of N_e .

Throughput varies greatly between methods

We measured execution time for each method on six simulated datasets. Simulations were made using our simulation model on each empirical tree to generate an alignment of 100 H_0 and 100 H_A sites. Execution times were measured as the elapsed time at completion of a run for each method using a single CPU, and are presented in table 3.2. The throughput of phylogenetic methods can vary by a large factor depending on the size of the phylogeny.

Method	Execution time (s)					
	<i>Cyperaceae</i>	<i>Amaranthaceae</i>	Rodents	Echolocation	HIV	Influenza
Multinomial	0.01	0.02	0.01	0.02	0.04	0.03
Gemma	1.79	1.90	1.72	1.81	1.96	2.03
Pelican	10.93	19.42	2.58	6.72	87.72	266.84
TDG09	22.21	40.56	6.84	12.56	369.87	
codeml	60.60	172.50	27.37	100.76	443.66	614.33
PCOC	65.56	129.01	27.80	76.48	346.25	436.84
Diffsel	1253.00	1497.84	946.79	1083.48	2659.78	3982.00

Table 3.2: Execution times for one alignment containing 100 H_0 and 100 H_A sites generated using our collection of empirical phylogenies. Result for TDG09 on the Influenza dataset is not available due to the program not terminating within a reasonable amount of time.

Multinomial and Gemma are the fastest methods by a large factor. None of these two methods require parameter estimations for a model of sequence evolution, allowing faster execution. At the other end, the two codon-level methods `codeml` and `Diffsel` are the slowest. `Pelican` is the fastest of the phylogenetic methods by a non negligible factor on all datasets.

3.4 Discussion

In this paper, we used simulations to compare the performance of methods that detect changes in the direction and intensity of selection, given an annotation of a phylogeny. These simulations rely on mutation-selection models of codon sequence evolution running along phylogenies.

3.4.1 Mutation-selection models for simulating coding sequences

Our choice to rely on mutation-selection models stems from the fact that these models have been shown to be more realistic for coding sequences than d_N/d_S methods [Spielman and Wilke, 2016, Bloom, 2014]. They distinguish between processes occurring at the mutation level, and processes occurring at the selection level among codons. This flexibility allowed us to implement in our simulations CpG hypermutability and gBGC. In addition, we have made the choice to use site-heterogeneous amino acid fitness profiles to emulate the heterogeneity among positions in protein sequences. For improved realism, the profiles we used come from [Rey et al., 2019], and are based on laboratory mutagenesis experiments [Bloom, 2017]. However, we assumed no fitness difference between synonymous codons, even though this can be implemented in the mutation-selection framework [Yang and Nielsen, 2008, Pouyet et al., 2016] to reflect selective pressures at the RNA or DNA level. Such selective pressures could be mistaken for selection at the protein level [Rubinstein et al., 2011, Spielman and Wilke, 2015], and therefore act as another confounding factor for the detection methods, but they were not investigated here. Despite this, and given the fact that we simulated along empirical phylogenies, we expect our results are informative about the performance of the methods on empirical data sets.

3.4.2 Methods working at the amino acid level perform as well as codon-based methods

Some of the methods in the benchmark rely on models that are similar to our simulation model. In particular, Diffsel is also based on a mutation-selection model, and `codeml` works at the codon level. Expectedly, these two methods perform well on our simulations. In agreement with previous results [Spielman and Wilke, 2015], `codeml`, which relies on d_N/d_S and does not use amino acid fitness profiles, is very effective except on long branches and trees (fig. 3.3c and fig. 3.4, the Influenza H1 phylogeny). All the other methods work at the amino acid level. Among those, the models based on a phylogenetic model (Pelican, TDG09, PCOC) vary in their performance, with Pelican standing out as the best performer. The lower performance of PCOC is likely due to two of its characteristics. Firstly, its reliance on a predefined set of amino acid frequency vectors, which may prevent it from accurately fitting the sites under study. Secondly, its “One-Change” component, which requires an amino acid change at each transition between background and foreground branches. This second limitation by design reduces the number of positive sites it can detect. TDG09 has lower performance than its reimplementations Pelican. The two implementations agree in the majority of cases, but disagree on some sites, likely due to optimization problems on boundary cases, which penalize TDG09. The fact that Pelican’s performance is similar to the performance of codon-based models suggests that its reliance on a WAG exchangeability matrix, not used in the simulation model, is not harmful. Further, it suggests that no information present only at the codon level is of much use to `codeml` or Diffsel, even when sequences are simulated with a model of CpG hypermutability (fig. 3.6). This may seem surprising, but probably relates to how we specified the detection problem we addressed. It is entirely centered around the amino acid profiles, so the codon level does not

provide much useful information. Finally, the non-phylogenetic methods perform quite well despite their simplicity. Multinomial, the simplest of our methods, performs better than Gemma, which has the ability to include the shape of the phylogeny as a covariate. This may be because Gemma was designed to handle binary characters, and we had to transform the amino acid data before feeding it into Gemma (see methods).

Beyond detection efficacy, the d_N/d_S and profile methods that we discuss in this manuscript vary in their execution speed. Methods that rely on models of sequence evolution typically have large computational footprints due to the use of the pruning or sum-product algorithm [Felsenstein, 1981], and the need for frequent matrix exponentiations. The computational footprints of these operations become larger as the state space grows: methods that work at the codon level (61 states) are more demanding than methods that work at the amino acid level (20 states) (fig. 3.2). Therefore, the profile methods that work at the amino acid level benefit from a computational advantage compared to codon-level profile methods or d_N/d_S methods. Diffsel is the slowest method despite a thoroughly optimized code base, for several reasons. Firstly, it works at the codon level. Secondly, it attempts to estimate more parameters than the other methods, and notably branch lengths. Thirdly, it is the only Bayesian method here, and as such is the only one providing a credible interval for each parameter, at each position, where the other methods only provide point estimates. Pelican’s speed is better than TDG09’s, due to the reliance on high performance computing libraries (see methods). It also uses diagonalization for matrix exponentiations, or the contraction of sparse substitution matrices to matrices of lower sizes as in the original method [Tamuri et al., 2009]. It has thus already been extensively optimized, but further improvements might be obtained by using substitution mapping and summary statistics as in Diffsel [Parto and Lartillot, 2018].

3.4.3 Features of a data set that affect performances

Results obtained in this benchmark highlight that profile and d_N/d_S methods perform differently in detecting changes in directional selection, depending on the features of a dataset. We identified a set of tree features that appear to have an effect on the performances: the number of transitions from background to foreground condition, the total time in the foreground condition, the number of foreground leaves, and the average length of foreground sub-trees. The variations in the detection performances observed on empirical phylogenies (fig. 3.4) likely are the result of interactions between these features, and possibly others that have yet to be identified.

Both Pelican and `codeml` benefit from an increased number of leaves in the foreground condition (fig. 3.3a), but not from an increased number of transitions (fig. 3.3b). However, non-phylogenetic methods (Multinomial, Gemma) benefit from increasing any of these features, including the number of transitions. `codeml` tends to perform better than profile methods on phylogenies with shorter foreground sub-trees on average. This conforms to our understanding of the two types of methods, and the kind of signal they rely on, as presented in the Introduction. In the case of a change in the direction of selection, the resulting burst of the d_N/d_S ratio occurs over a short time period, and quickly decreases back to a purifying selection regime ($d_N/d_S < 1$, fig. 3.1b). This implies that on longer branches more time is spent in a purifying selection regime, reducing the signal for high d_N/d_S as the rate of non-synonymous substitutions decreases. In contrast, profile methods rely on amino-acid frequencies to detect positive selection. In this case, the signal is strongest when the amino-acid frequencies have reached an equilibrium and differ the most from the ancestral frequency distribution. Since reaching the foreground equilibrium distribution through substitutions takes time, detection performances tend to increase on longer branches (fig. 3.3c).

Profile methods that do not take into account the phylogeny have a reduced performance on short branches. In that case observations at the leaves of the phylogenetic tree are more strongly correlated and this may mislead methods that assume independent observations (like Multinomial) or rely on a less accurate model (like Gemma). On longer branches, observations at the leaves of the tree tend to become more independent, and non phylogenetic methods exhibit performances similar to their more complex counterparts.

3.4.4 GC-biased gene conversion is an important confounding factor for both d_N/d_S and profile methods

In an effort to make our simulations more realistic, we introduced two non-adaptive confounding factors in our model: CpG hypermutability, which affects the mutation component, and GC-biased gene conversion (gBGC), which affects the selection component. We have found that introducing gBGC on foreground branches induces a significant drop in performances for all the methods, with higher values of gBGC resulting in larger decreases (fig. 3.6). gBGC mimics selection, independently of the underlying fitness profiles, and scrambles the signal used to detect changes in the selection regime. This corroborates previous studies on the role of gBGC in disrupting the detection of selection in genome sequences [Ratnakumar et al., 2010, Rousselle et al., 2019, Guéguen and Duret, 2018, Ho and Hurst, 2022]. Mechanistic codon-level models such as DiffSel could be extended to account for this effect, and untangle it from directional selection.

On the other hand, strong CpG hypermutability was not found to induce changes in the performance in most cases. It is possible that codons that contain CG dinucleotides are not frequent enough in our simulations based on the mutsel framework to reduce the AUC metric.

3.4.5 Persistent positive selection is an important confounding factor for d_N/d_S methods, less so for profile methods

Protein sites may be subject to a variety of selection regimes (fig. 3.1a). It may be difficult to distinguish sites undergoing changes in the direction of selection from sites evolving under a different selection regime, in particular persistent positive selection. In our simulations under the model of [Tamuri, 2021], we found that `codeml` had difficulty distinguishing the two processes, in agreement with [Parto and Lartillot, 2018]. PPS results in elevated (> 1) d_N/d_S values throughout the phylogeny, which is not well modelled by `codeml`'s branch-site model A, which assumes that positive selection only occurs on foreground branches. `codeml` has to choose between two alternatives, none of which fits the data very well: either consider that PPS sites never have $d_N/d_S > 1$, or consider that PPS sites have $d_N/d_S > 1$ on foreground branches only. The second alternative is closer to the truth, and therefore is chosen in a large number of cases, resulting in many false positives, and a low AUC. On the other hand, profile methods seem to suffer less from PPS, although it is harder to distinguish from episodic positive selection than purifying selection is (sup. fig. C.7). The effect of PPS can be compensated by increasing the effective population size N_e , which acts as a scaling factor for the intensity of selection: as a result, observed amino acid frequencies are more representative of the actual fitness profile with higher values of N_e , and constitute a stronger signal for profile methods.

3.4.6 Interpreting screens for changes in directional selection

The methods we discussed in this paper can be used to detect sites in alignments whose selection regime has changed coincidentally to a punctual transition event. Such transitions are typically changes in the environment, for example when a virus switches between hosts, and might also induce cases of convergent evolution (e.g the multiple transitions of mammals to the marine environment [Chikina et al., 2016]). In this context, these models can be used to give insights into the relation between the genotype and a given binary phenotype (e.g ancestral vs convergent, marine vs terrestrial, ...), even though some methods can handle more than two categories of phenotypes (e.g., Pelican and [Wertheim et al., 2015]). The fact that all methods except Multinomial are conservative, *i.e.*, have low rates of false positives, indicates that the positives they output are likely to be worthy of further study.

The d_N/d_S and profile methods that we discuss in this manuscript all make similar assumptions. Firstly, they can only handle a single phenotype or environmental condition at a time. This implicitly assumes that other phenotypes or conditions are unimportant for the evolution of the site under consideration. Such a strong assumption is likely to be incorrect in many cases: for instance a site may be important for several phenotypes, or its evolution may be more strongly associated to another phenotype or condition that has not been tested. Secondly, they assume that the evolution of the phenotype is known without uncertainty. d_N/d_S approaches that can handle uncertainty in the evolution of the phenotype when reconstructing the evolution of gene sequences have recently been proposed, but remain to be extended to the site level [Halabi et al., 2021]. Thirdly, they rely on the comparison of two scenarios, one of which assumes homogeneity of the process across the phylogeny. In the d_N/d_S method we consider, this means that the same d_N/d_S parameter applies to the site throughout the phylogeny. In the profile methods we consider, this means that the same profile applies to the site throughout the phylogeny. This is likely to be incorrect: the site may be evolving heterogeneously because of non-adaptive processes (e.g., CpG hypermutability or gBGC), or because it is correlated to unaccounted-for phenotypes or conditions. The use of homogeneous null scenarios can result in model confounding whereby an incorrect model is chosen in the absence of the true generating model [Jones et al., 2019]. This is what occurred in the gBGC simulations where the gBGC model generated data that was better fitted under our H_A model than under our homogeneous H_0 model. However, our simulations of persistent positive selection show that profile methods are robust to this particular confounding process.

Our results show that a site found as positive with a profile method could result from a change in the direction (fig. 3.4) or intensity (fig. 3.5) of selection, as well as from a change in gBGC or PPS (fig. 3.6). At this stage, distinguishing between these processes requires looking at the profiles estimated by the method at the site. These profiles have been shown to be inferred accurately by several mutsel models [Spielman and Wilke, 2016]. Since codon-based methods do not perform better than amino acid-based methods in our hands, we suspect that the latter should also infer accurate profiles, although this will have to be verified in a future study. Given accurate profiles, one could distinguish between the different processes. Relaxation (respectively intensification) of selection should result in a flatter (resp. more heterogeneous) profile (fig. 3.1), which could be detected by computing its entropy and comparing it to the entropy of the other profiles at the site. gBGC should result in a shift towards GC-rich amino acids. PPS should result in a high amino acid diversity (large number of amino acids with non-zero frequencies).

3.4.7 Looking forward

The profile methods presented here have all been evaluated in the same setting, where the evolution of a site depends on two conditions that have been assigned to branches of a phylogeny. Not all phenotypes or conditions of interest can be known without uncertainty along a phylogeny, or can be accurately described by such a binary classification. Pelican can handle more than two conditions, but does not handle continuous annotations along a phylogeny, or uncertainty in the extant or ancestral states. Such extensions would be very useful. Similarly, the results show that accounting for gBGC in profile methods could be important. This could be done in codon models by following the approach that [Guéguen and Duret, 2018] used in d_N/d_S models.

[Tamuri et al., 2014] and [Spielman and Wilke, 2016] showed that a penalized version of mutsel models performed better than the unpenalized version. We suspect that Pelican might also perform better with similar penalties. However, the use of penalized likelihoods would prevent us from relying on likelihood ratio tests to compute p -values and detect positive sites. Instead, [Tamuri, 2021] relied on simulations to compute p -values, which is more resource intensive and would compromise Pelican’s scalability. More work is needed to investigate the benefits of using penalization in Pelican, and, if any, come up with a fast method to compute p -values or scores. Such a method might also improve on the LRT that we have used here, as we saw that tinkering with its degrees of freedom improved the performance of the method in some cases (sup. fig. C.8). It would not however change the ML estimate of a 0 fitness value for unseen amino acids, which is unrealistic [Rodrigue, 2013]. It will be important to develop a method that yields more realistic amino acid-specific parameter values but remains fast enough for use at the genome scale.

Overall, the results show that profile methods constitute a solid alternative to d_N/d_S methods to screen for substitutions associated to changes in a phenotype or condition of interest. This opens new possibilities to better understand the link between a substitution, the structure of the protein where it occurs, and the phenotype or condition to which it is correlated. The amino acid profiles inferred by a profile method at a site can be used to investigate the effect that having a high fitness or a low fitness amino acid has on a protein structure, in a particular condition. Profile methods could thus pair very well with the recent improvements in protein structure prediction [Jumper et al., 2021] to yield new insights into the molecular basis of adaptation.

3.5 Conclusion

In this paper we evaluated on simulations a series of methods aiming to detect changes in selective pressures in coding sequences along a phylogeny. We found that some profile methods compare favourably to a commonly used d_N/d_S method, both in terms of power and in terms of speed, including in the presence of confounding factors. In particular, profile methods can readily distinguish changes in directional selection from persistent positive selection, something that the d_N/d_S method we tested cannot do. Among profile methods, we found that Pelican, a method operating at the amino acid level, can be used to detect selective pressure changes efficiently. This makes genome-wide searches for sites correlating with a phenotype or condition of interest doable on a single computer within a few days.

Further extensions of Pelican are envisioned, for example to handle continuous phenotypes. Integrating the effect of gBGC in the model would also be a major improvement, as we have found that it has a strong confounding effect on the detection of selection.

3.6 Methods

3.6.1 Detection of ω variations using `codeml`

We used the `codeml` tool from the PAML package to detect variations of d_N/d_S as a proxy for variations of selective pressure, as was done in [Thiltgen et al., 2017]. Branch lengths were re-estimated by `codeml`. `codeml` was configured to use the branch-site model A [Zhang et al., 2005, Yang et al., 2005]. This model assumes there are three categories of sites in the alignment, whose proportions are estimated. Categories 0 and 1 have a homogeneous ω value throughout the phylogeny. Category 2 has one ω value estimated per branch condition: on background branches, the ω is between 0 and 1 (subcategory 2a), characteristic of purifying selection, or at 1 (subcategory 2b), characteristic of neutral evolution. On foreground branches, $\omega \geq 1$, characteristic of neutral or positive selection. A site is declared "positive" if it belongs to this category 2. The probability for each site to be positive as inferred by the method was computed from the Bayes empirical Bayes probabilities, resulting from running `codeml` with parameter `fix_omega = 0` and summing up the probabilities to belong to categories 2a and 2b in the model.

3.6.2 Multinomial method

The multinomial method models each site of an alignment as a collection of independent categorical variables, thus completely ignoring the phylogeny. It compares two models using a likelihood ratio test (LRT), the first one assumes a single probability vector of length 20 (one frequency per each amino acid), the second a pair of vectors, one for each condition. Computing a p -value is however difficult in our setting, as at a given site, most of the amino acids are not observed and as a consequence their frequency estimated by maximum likelihood is zero, and thus lies at the boundary of the parameter space. In that case the usual convergence of the likelihood log-ratio to a χ^2 distribution known as Wilks theorem does not hold. While there exists literature on the subject (see [Mitchell et al., 2019] for a recent result), existing results are difficult to apply. We reused a heuristic we found in [Tamuri et al., 2009], consisting in approximating the likelihood log-ratio distribution under the null by a χ^2 distribution with number of degrees of freedom equal to the number of amino acids observed at the leaves of the tree minus one.

3.6.3 Pelican: improvements on TDG09

Pelican is a reimplementation of the TDG09 method, originally published by [Tamuri et al., 2009]. TDG09 relies on a site-independent model of amino acid sequence evolution and the WAG exchangeability matrix. The model involves two kinds of parameters: stationary distributions of amino acids and branch scale.

Inference of selective pressure shifts is based on the postulate that stationary distributions of amino acids reflect the fitness profile in a condition (e.g. foreground or background). In a similar way to the multinomial method, the likelihoods of two models are compared using the LRT procedure, where one model assumes a single stationary distribution of amino acids shared between both conditions, and the other model assumes a specific stationary distribution per condition.

Parameters of the model, such as stationary distributions and branch scale, are optimized to maximum likelihood using the Nelder-Mead algorithm [Nelder and Mead, 1965]. We implemented an alternative approach using automatic differentiation, made available through the PyTorch library [Paszke et al., 2019], that converges to the same optima as the Nelder-Mead implementation. This

alternative optimisation algorithm is currently not used, but might be useful in future extensions of the method.

Pelican is implemented in the OCaml language [Leroy et al., 2021]. The underlying mutation-selection model implementation takes advantage of LAPACK [Anderson et al., 1999] bindings for fast linear algebra computation, and optimisations for transition matrices exponentiation through diagonalisation [Yang, 2006]. Pelican is available at <https://gitlab.in2p3.fr/phoogle/pelican>.

3.6.4 Simulations

In all our experiments, simulations were used to generate amino-acid or codon alignments with a constant number of sites $N = 10\,000$. The simulator was configured to generate 90% of H_0 sites (no changes in selective pressure) and 10% of H_A sites (different selective pressure between background and foreground condition). Simulations were done using a general time-reversible (GTR) mutation-selection model at the codon level. The model allows for two different regimes: one modeling selection in the background condition, and the other in the foreground condition. Selective pressure changes on H_A sites are simulated using either the foreground or background regime, depending on the condition of each branch in the phylogenetic tree. H_0 sites are generated using only the background regime, indicating no change in the selective pressure through the tree for these sites. Each regime is represented as a matrix of substitution rates between codons, which can be run along the phylogeny using Gillespie’s algorithm [Gillespie, 1976].

The substitution rates are the result of a mutation probability and a relative fixation probability, which depends on a selection coefficient associated with the transition to the mutated state. Mutation probabilities for the GTR model of nucleotide substitutions are based on exchangeabilities drawn from a *Gamma*(1, 1) distribution, and equilibrium frequencies from a *Dirichlet*(10, 10, 10, 10) distribution, and are shared across sites. The selection coefficient S (eq. 3.1) is defined as the difference in fitness between the ancestral state X and the mutated state Y in a condition c .

The relative fixation probability $u(S)$ for a mutation is computed from the selection coefficient S as per [Kimura, 1983]:

$$S_{X \rightarrow Y}^c = \text{fitness}(X, c) - \text{fitness}(Y, c) \quad (3.1)$$

$$u(S) = \frac{S_{X \rightarrow Y}^c}{1 - e^{-S_{X \rightarrow Y}^c}} \quad (3.2)$$

Fitness values are determined from amino acid frequency profiles, which are randomly picked at each site from a set of 263 preset profiles [Rey et al., 2019] for each condition. These frequency profiles are transformed into fitness profiles by multiplying them by a factor $\rho = 4^1$. As a result, values $S_{X \rightarrow Y}^c$ are between -4 and 4 .

Codon substitution rates σ are the product of mutation rates μ and the relative probability of fixation:

$$\sigma_{X \rightarrow Y}^c = \mu_{X \rightarrow Y} \times u(S) \quad (3.3)$$

3.6.5 gBGC simulation

GC-biased gene conversion (gBGC) acts as a fixed increase in fixation probability for mutations from either A or C nucleotides to G or C; conversely it is modeled as a probability decrease when

¹The notation used in the published version of this article was originally N_e instead of ρ ; in hindsight, I find this to be confusing, as this parameter is not the effective population size, but a scaling factor that *represents* the effect of N_e . I therefore use ρ instead, consistently with the notation used in [chapter one](#).

mutating the other way around. We included GC-biased gene conversion in our simulation model as a bias term in the selection coefficient S :

$$S_{X \rightarrow Y}^c = B_{GC}(X, Y) + \text{fitness}(X, c) - \text{fitness}(Y, c) \quad (3.4)$$

Based on [Glémin et al., 2015], we chose an intensity of $B_{GC} = 10$, that is applied on foreground branches, which is a strong effect for this process. Transition rates were not affected on background branches. This way, in H_A sites, the change in selective pressure between background and foreground branches that has to be detected is driven both by the shifted fitness profile, and the effect of gBGC. In H_0 sites, gBGC affects foreground branches.

3.6.6 CpG simulation

CpG hypermutability is introduced in the simulation model as a scaling factor ν^2 for the mutation probability:

$$\sigma_{WXZ \rightarrow WYZ}^c = \mu_{X \rightarrow Y} \times \nu(W, X, Y, Z) \times u(S) \quad (3.5)$$

where W and Z are the states at the surrounding sites. This context is necessary because CpG dinucleotides can occur across two codons. As a consequence, the evolution of a whole sequence is not site-independent anymore, which led us to develop a dedicated Gillespie simulator. CpG hypermutability only occurs on methylated CpG dinucleotides, and induces an increased probability of mutation from C to T in this context (or G to A on the reverse strand). We assume that any CpG dinucleotide in our simulation is methylated. If the conditions for hypermutability are not verified when comparing changes from X to Y , or the current branch is background, $\nu(W, X, Y, Z) = 1$ and has no effect. Otherwise, on foreground branches, we set $\nu(W, X, Y, Z) = 10$ based on [Meunier et al., 2005], both on H_A and H_0 sites.

3.6.7 Simulation of persistent positive selection

PPS is introduced in the simulation model as a constant increasing the fitness of all other amino acids except the current one [Tamuri, 2021]. This is achieved by modifying equation 3.1 as:

$$S_{X \rightarrow Y}^c = \text{fitness}(X, c) - \text{fitness}(Y, c) + PPS \quad (3.6)$$

where $PPS \geq 0$ is a constant and describes the strength of positive selection. To simulate data, we relied on two parameter settings. In the first setting, we simulate sequences under a mild selection strength, setting $\rho = 4$ and $PPS = 2$. This setting ensures that differences in amino acid fitnesses are between -4 and 4 , as in the rest of the manuscript. In the second setting, we simulate under a strong selection regime, with $\rho = 10$ (*i.e.*, differences in amino acid fitnesses between -10 and 10), and $PPS = 10$. This second setting resembles parameter values observed on the sites showing the strongest positive selection in [Tamuri, 2021], and is also similar to their own simulation settings. H_A sites were simulated with different profiles for background and foreground branches, and H_0 sites were simulated with PPS running both on background and foreground branches.

Code and data availability

Source code to reproduce the results in this paper is publicly available at

²This notation changed for similar reasons.

<https://gitlab.in2p3.fr/phoogle/spcd-benchmark>.


The implementation of Pelican is also made available at

<https://gitlab.in2p3.fr/phoogle/pelican>.

Plots were produced in R [R Core Team, 2021] using the packages *ggplot2* [Wickham, 2016] and *ggtree* [Yu, 2020].

Acknowledgements

We thank Nicolas Lartillot and Anamaria Necşulea for their comments on an early version of the manuscript and fruitful discussions. The research presented here was funded in part by the Convergencix project (ANR-15-CE32-0005). L.D. was supported by a PhD fellowship from the Université Lyon 1 Claude Bernard.

 **Chapter 3 summary:** *Evaluation of methods to detect shifts in directional selection at the genome scale*

In this article, we used simulations to evaluate several methods to detect changes in the selective pressure in a variety of settings. We included in this benchmark two methods described in the previous chapter, Multinomial and GEMMA, that are not based on a model of sequence evolution, and others implementing phylogenetic substitution models, at the level of amino acids (TDG09, PCOC) or codons (`codem1`, Diffsel). Our implementation of the TDG09 model under the name Pelican was also considered.

We found that some methods based on amino acid profiles to represent fitnesses in the case of Diffsel, or equilibrium frequencies in the case of Pelican, had performance level similar to that of `codem1` that compares synonymous to non-synonymous substitutions rates (d_N/d_S). Despite the model of Pelican being more simple and less realistic, it appears that for the purpose of detecting changes in directional selection, modeling the substitution process at the amino acid level is sufficient. However, our results suggest that d_N/d_S methods could have more power to detect relaxed selection, although profile methods can generally manage to do so.

We also highlighted that all methods were sensitive to the confounding effect of gBGC. Persistent positive selection could be distinguished from directional selection by profile methods, but not by `codem1`; however that conversely implies that profile methods would have a harder time distinguishing PPS from purifying selection. CpG hypermutability did not have a meaningful impact on the general performance.

As it turns out, our results showed that our implementation of the TDG09 model in Pelican fixes some issues from the original implementation, and provides a good trade-off between speed and accuracy to identify sites that have undergone differential selection between conditions in the tree. This motivated us to invest more efforts on extending and improving Pelican, but also to investigate its robustness and limitations. The following chapter describes its model with implementation details, and explores as well potential statistical issues in the hypothesis testing procedure.

Chapter 4

Pelican: a fast phylogenetic method to identify selective pressure changes

Pelican originated as a reimplementations of the TDG09 model from [Tamuri et al., 2009]. The initial motivation for this work was the observed performance discrepancy between TDG09 and DiffSel [Parto and Lartillot, 2018], previously reported in [Rey et al., 2019]. DiffSel is a Bayesian implementation of a mutation-selection model of sequence evolution, at the level of codons. It can be applied to the same purpose as TDG09, i.e. identify sites in coding sequences that are associated with a phenotypic trait, and was found to be better at that task [Rey et al., 2019]. Because of the complexity of the mutation-selection model, the large dimension of the codon state space and the computation costs involved in the estimation of posterior probability distributions for model parameters, DiffSel is too computationally intensive to analyze large scale datasets — even though the implementation was thoroughly optimized. According to the results from [Rey et al., 2019], the method was reported to outperform all of the other models that were evaluated. Our motivation was to investigate the main factors that could explain the ascendant of DiffSel over TDG09 specifically, as both models share some similarities. Some hypotheses we had were:

- the mutation-selection framework could be determinant in the quality of the detection: TDG09 simplifies the evolution of sequences by modelling amino acid substitution directly, overshadowing the mutational process, and using far fewer parameters overall
- DiffSel estimates branch lengths from the alignment. In TDG09, the branch lengths are fixed, and a scaling factor is estimated separately for each site. DiffSel thus has more flexibility to fit the observed data, and integrates information from the alignment instead of a single site
- hypothesis testing in TDG09 is performed using tools from the frequentist paradigm, namely estimations at maximum likelihood and likelihood ratio tests. One could imagine that the Bayesian framework that is leveraged in DiffSel could be more robust in our setting, potentially resulting in better predictions

We therefore made our own implementation of the TDG09 model, as a foundation to progressively implement some features of DiffSel. Surprisingly, we found on the basis of predictions made on simulated datasets that this reimplementations largely outperforms the original one, as reported in [chapter three](#). Moreover, the performance measured was comparable to that of DiffSel and `codeml`,

an observation that defeated all our expectations.

In this chapter, I go into further details on the model underlying Pelican and TDG09, and its software implementation. As a foreword, I give a short overview of the features implemented in Pelican, and present the technical framework that we used. I then describe the original model proposed in TDG09, and the procedure to perform hypothesis testing at each site using likelihood ratio tests, with details on the implementation of this model in Pelican and an emphasis on optimization of its computational performance. The limited applicability of the likelihood ratio test in our setting is also investigated, first using simulations on synthetic trees, then illustrated using a problematic example reported in the previous chapter (see figure 3.5). I conclude this chapter with propositions to improve the throughput of analyses, using the Multinomial model that I described in [chapter two](#) as a first-pass filtering heuristic, and exploring the use of an alternative strategy for fitting Pelican’s model using GPU computation and automatic differentiation.

Contents

4.1	Technicals	77
4.1.1	Inputs and outputs	77
4.1.2	Other features	79
4.2	The original model: TDG09	80
4.2.1	A general time-reversible model of sequence evolution	80
4.2.2	Model parameters and empirical exchangeabilities	80
4.2.3	A condition specific model of sequence evolution	81
4.2.4	Hypothesis testing	82
4.3	Implementation and fitting of the model	83
4.3.1	Felsenstein’s algorithm for phylogenetic likelihood computation	83
4.3.2	Numerical optimization	86
4.3.3	Efficient matrix exponentiation	87
4.3.4	State space reduction	89
4.4	On hypothesis testing and the applicability of LRT	91
4.4.1	Wilks’ asymptotic null distribution: the problem of the effective sample size	92
4.4.2	Degrees of freedom	95
4.5	Improving the detection of relaxed selection	98
4.5.1	Motivation: a pathological case of Influenza	98
4.5.2	A better approximation of the null distribution of log-ratios	100
4.5.3	Adapting the model to distinguish neutral from purifying selection	101
4.5.4	Performances obtained with the Pelican variants	103
4.6	Filtering sites using Multinomial, a fast non-phylogenetic method	104
4.7	An alternative approach for fitting the model using automatic differentiation	105
4.7.1	Vectorized implementation of Felsenstein algorithm	105
4.7.2	Optimisation of the parameters using L-BFGS	108
4.7.3	Results	109
4.7.4	Discussion	111

4.1 Technicals

Pelican is a command-line software implemented in OCaml, a functional, statically compiled language from the Meta-Language family. It is well suited for the development of robust applications, by helping to reduce the amount of potential bugs that are left undetected, and offers good expressiveness as a relatively high-level programming language. Its functional programming paradigm aims to avoid the use of functions with side-effects and keep the complexity of the resulting code as low as possible. The strong typing system, coupled with automatic type inference provided by the compiler, helps to enforce invariants through the definition of dedicated data types: the definition of a type may express that it verifies a certain property, and calls to functions that require this property can not be applied without the correct type. For example, we define specific types for manipulating data related to amino acids; one of them is the amino acid vector. In this case, the invariant is that every amino acid vector has length 20, and that the order of elements always matches the amino acid alphabet¹, although the underlying data structure is a simple array. It has also a semantic function, that makes it obvious for the programmer whether they are manipulating amino acid vectors or other array-like variables. All in all, preventing side-effects and enforcing type consistency are two features that help to prevent apparitions of unexpected behaviours in the program.

Garbage collection is another feature that alleviates the burden of memory management, although it has a cost in terms of performance. Nonetheless it remains fast enough, with computation times about three times higher than native C programs and comparable to that of Java — in comparison, Python is about 50 times slower than C². In the implementation of Pelican, the most heavy computations are expressed as linear algebra operations, which are performed efficiently using OCaml bindings to LAPACK [Anderson et al., 1999], a low-level library for linear algebra implemented in C. The choice of such a language is also well suited to working with recursive data structures such as phylogenetic trees. Both the functional programming paradigm which encourages recursion, and the flexible system for type definitions, make it natural to represent and manipulate such structures. A typical example for this is Felsenstein’s algorithm which I describe in section 4.3.1 and is recursive by definition.

4.1.1 Inputs and outputs

Input file formats for Pelican

The main inputs of Pelican are a phylogenetic tree in the New Hampshire eXtended (NHX) format³, and one or more sequence alignments in the FASTA format. Sequences may be either in the amino acid alphabet, or the nucleotide alphabet, in which case they are internally translated to amino acids using the standard genetic code. Translation using alternative genetic codes is also available. Amino acid sequences are parsed using the IUPAC nomenclature of amino acids, including symbols for partly determined sequences⁴.

Tips of the NHX tree must be labelled, and are matched to the names of the sequences that are present in the alignment. Tips that could not be matched to a sequence name in the alignment are trimmed from the tree when performing the analysis of said alignment. Gaps in the alignment are

¹We could then refine this type further and define one for vectors of amino acid frequencies, adding the invariant that the sum of elements is equal 1.

²Although this is very dependent on the kind of algorithm and actual implementation. Benchmark available at <https://benchmarksgame-team.pages.debian.net/benchmarksgame/box-plot-summary-charts.html>.

³The full NHX format specification is described at: <http://www.phylosoft.org/NHX/>

⁴Symbols correspondence as defined in [JCBN, 1984, table 5] is available at <https://www.ddbj.nig.ac.jp/ddbj/code-e.html#amino-1>

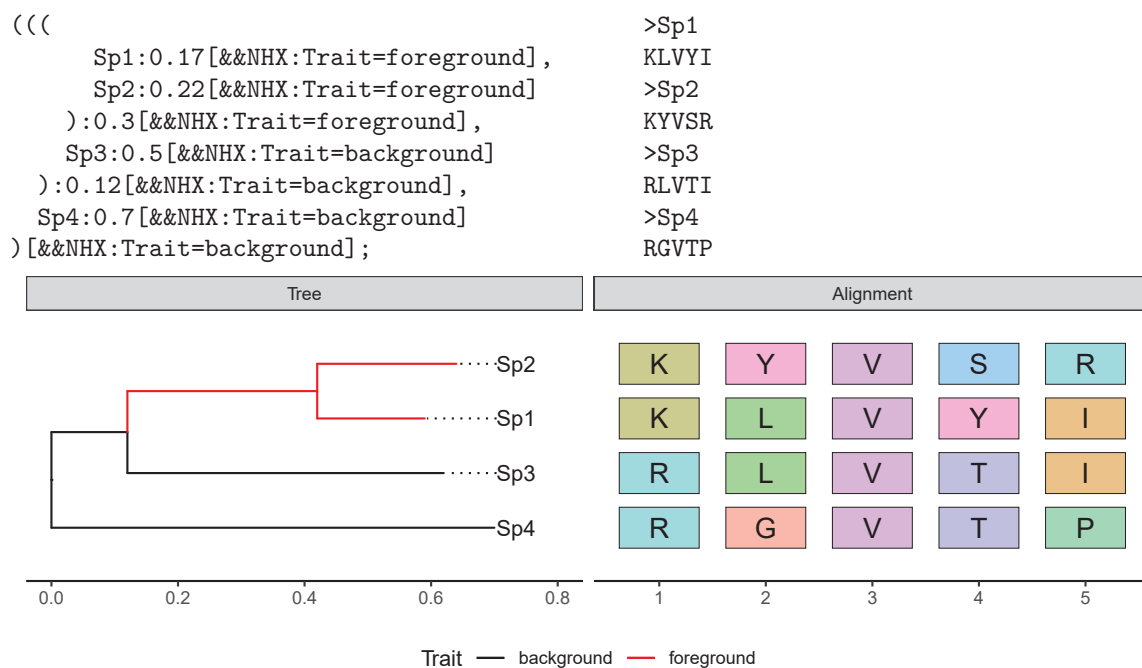


Figure 4.1: Example input for Pelican (top panel) and its graphical representation (bottom panel). Trees are represented in NHX format, and alignments in FASTA format. Trait annotation as **background** and **foreground** is given as an example, but can be arbitrarily chosen.

treated the same but on a site-wise basis, as the model can not handle insertion or deletion events as of yet.

All tips and internal nodes in the tree must be annotated with a trait value using an NHX tag, which simply consists in a tag name and value pair. The default tag name is **Trait**, but the use of an alternative tag name can be defined using an option in the command line. Trait values can be arbitrarily defined, e.g. 0 and 1, **background** and **foreground**, or X and Y, and can also be non-binary (e.g. **condA**, **condB** and **condC**). They can even be continuous values: in this case, they are treated using a specific model for continuous phenotypes that is presented in [chapter five](#). A value for the length of the parent branch must be provided for each node, with the exception of the root node, when there is one. The method does not require the tree to be rooted.

An exchangeability rate matrix can also be provided, in place of the default WAG matrix [Whelan and Goldman, 2001] of empirical exchangeabilities. It must be contained in a file with the same format⁵ that is used in the PAML package [Yang, 2007]. An option to use the LG replacement matrix [Le and Gascuel, 2008] instead is already embedded in the program.

Output from Pelican analysis

The output of Pelican consists in one p -value per site across all genes. Constant sites where a single amino acid is observed across all species are assigned p -value equal 1. Results are presented as tab-separated values (TSV format) with one line per site. Each line consists in 8 values:

alignment alignment identifier that is derived from the corresponding FASTA file name

site a site identifier that is its position in the alignment, starting from 0

nseq the number of sequences in the alignment

⁵For an example, see <https://www.ebi.ac.uk/goldman-srv/WAG/>

naa the number of distinct amino acids that were found at the site

multinomial_pval a p -value from the Multinomial test

aagtr_pval a p -value from the likelihood-ratio test between the two alternative models of Pelican

reduced_loglk the log likelihood of the reduced model

full_loglk the log likelihood of the full model

Analysis results are progressively stored in a local database during a run, that can be interrogated using the CLI to retrieve results from past analyses. Commands are also provided to extract parameter estimates at each site, as TSV format; additional plotting functions for amino acid profiles are provided. This also permits to interrupt an in-progress analysis, and resume it instead of restarting it from the beginning.

4.1.2 Other features

Parallel computation

To enable systematic screening of genome-scale datasets, we implement parallel processing across gene alignments, where the number of processes can be controlled by the user. The memory footprint of Pelican is small, which allows to perform a large number of alignment scans in parallel, the main limitation being the number of available computation cores.

Multinomial filter

The Multinomial method that was presented in [chapter two](#) was incorporated to the Pelican software, to be used as an optional first-pass filter. It is provided as a mean to quickly detect sites with a strong signal for associations to the phenotype, by avoiding to analyze every sites in a dataset with Pelican. Only a subset of candidate sites is passed on to Pelican, resulting in a massive speed-up, at the expense of an increased false negative rate (see [section 4.6](#)).

Phenotype annotation

Pelican features some utility sub-commands to help with the annotation of phylogenies, and infer ancestral traits from extant observations using simple models. Inference of discrete ancestral traits is done at maximum parsimony using Fitch's algorithm [[Fitch, 1971](#)]. A separate sub-command is provided to perform inference of ancestral continuous traits using a Brownian model of evolution along the tree [[Felsenstein, 1985](#)]. They are provided for convenience, but I would recommend that trait annotation be adapted to each specific dataset, using dedicated models and human expertise whenever possible.

4.2 The original model: TDG09

In 2009, Tamuri, dos Reis, Hay & Goldstein published the results of their effort to identify amino acid mutations in the genome of the Influenza A virus that enabled its switching from avian hosts to human hosts [Tamuri et al., 2009]. Like all pathogens, viruses are subject to selective pressures that are specific to their host, and therefore have to be highly adapted with regard to the host immunity in order to maintain themselves and spread in the population. This implies that a virus strain can not easily transfer between distant hosts without adaptive changes to a different host environment.

Tamuri et al. proposed an approach to search for protein sites that were likely to be involved in this adaptive process: that is, sites whose composition changed through substitutions in response to the environmental change concomitantly to host switching. To do so, they designed a model where the amino acid preferences at a site are dependant on the host condition. As a consequence, two distinct substitution models are inferred per site. Either one is used to describe the substitution process occurring on a branch, depending on the condition attached to it.

4.2.1 A general time-reversible model of sequence evolution

The model is expressed as a general time-reversible (GTR) model [Tavaré et al., 1986, Lanave et al., 1984], originally used to describe processes of nucleotide substitution, but was adapted to the amino acid alphabet. I briefly discussed this kind of model in [chapter one](#), as it makes up the mutational component in the mutation-selection we use for simulations, but go into more details here. The GTR model defines a Markov process as a matrix of transition rates Q , which is assembled from a vector of equilibrium frequencies π and an exchangeability rate matrix S , such that

$$\begin{cases} Q_{ij} = S_{ij}\pi_j & \text{if } i \neq j \\ Q_{ii} = -\sum_{j,j \neq i} Q_{ij} & \text{otherwise} \end{cases} \quad (4.1)$$

the vector π of equilibrium frequencies is the Markovian stationary distribution — I will use both terms interchangeably from now on.

A Markov process is time-reversible when the transition flow at the equilibrium between two states is equal in both ways, so that

$$\pi_i Q_{ij} = \pi_j Q_{ji} \quad (4.2)$$

This property is verified in the GTR model, since the exchangeability matrix S is symmetric

$$S_{ij} = S_{ji} \Rightarrow \frac{Q_{ij}}{\pi_j} = \frac{Q_{ji}}{\pi_i} \Rightarrow \pi_i Q_{ij} = Q_{ji} \pi_j \quad (4.3)$$

Time-reversibility allows choosing any node as the root of the tree by using Felsenstein’s pulley principle [Felsenstein, 1981]. This is particularly convenient for efficient inference of phylogenetic trees at maximum likelihood. However, inference using non-reversible models can also be made more efficient simply by reorganizing the computation of likelihood [Boussau and Gouy, 2006].

4.2.2 Model parameters and empirical exchangeabilities

Among classical models of sequence evolution, the GTR model is the most general, with the largest number of distinct parameters compared to other Markovian models of evolution with the same state space dimension. In the context of nucleotide sequence evolution, 4 states are defined to represent

the ATCG alphabet. The stationary distribution π sums to 1, thus introducing 3 parameters, since one of the 4 can be deduced from the others. The exchangeability matrix S is symmetric and can be reduced to a triangular matrix: it defines 6 additional parameters. This amounts to 9 parameters to be estimated in total, that remains manageable with regard to the limited data available when fitting the model on a single alignment site.

Other models further reduce the number of parameters by making some assumptions on either the transition rates or the equilibrium frequency. As an example, the Jukes-Cantor model [Jukes et al., 1969] is the simplest substitution model, that consists in a single parameter μ for the overall substitution rate and assumes equal equilibrium frequencies.

The number of parameters in the GTR model drastically increases when working with larger state spaces, such as amino acid (20 states, 209 parameters) or codon (64 states, 633 parameters) alphabets. Fitting parameter rich models can be difficult when the quantity of data is limited, as is generally the case when sites are considered independently. Empirical estimations of these parameters have been proposed, using reference datasets, which are often used as plug-ins in evolution models. Empirical amino acid exchangeability matrices are typically inferred from protein sequence alignments, initially using a maximum parsimony approach to assemble the PAM matrix [Dayhoff et al., 1978]. Improved matrices estimated at maximum likelihood were later proposed, such as the WAG [Whelan and Goldman, 2001] or LG [Le and Gascuel, 2008] matrices. These empirical estimates provide general exchangeability rates and equilibrium frequencies. In TDG09, fitting the model is thus made easier by using empirical replacement rates from the WAG matrix, while equilibrium frequencies are fitted at maximum likelihood to model individual amino acid preferences at each site. This is also the case in Pelican, where empirical exchangeabilities from the WAG matrix are used by default, although any other replacement rate matrix can be plugged in the model at the user's choice.

4.2.3 A condition specific model of sequence evolution

To model changes in amino acid preferences in relation to a phenotypic trait, TDG09 defines a heterogeneous GTR model, where amino acid frequency profiles are dependent on the trait. In the formalism that was used in the previous section, this heterogeneous model can be represented through the definition of two rate matrices, each describing a substitution process.

$$M_3 : \quad \begin{cases} Q_{ij}^{\text{Av}} = \sigma S_{ij} \pi_j^{\text{Av}} \\ Q_{ij}^{\text{Hu}} = \sigma S_{ij} \pi_j^{\text{Hu}} \end{cases} \quad (4.4)$$

where Av and Hu refer to the avian host and the human host condition, respectively, and the WAG matrix is noted S . This model is depicted in a more intuitive fashion in figure 4.2.

In addition to the 19 free parameters⁶ for each of the two stationary distributions π , one scale parameter σ is introduced to adjust the general substitution rate for the site. The evolution rate σ is site specific, and assumed to be constant in time, regardless of the host condition. This model describes the evolution of sites whose substitution process varies in response to changes in the environment, in this case the host species.

However, not all sites in an alignment are associated to a given trait or environmental condition. A homogeneous GTR model with only one stationary distribution is used to describe the evolution

⁶A frequency distribution sums to 1 and the value of one of the 20 frequencies can therefore be deduced from the others.

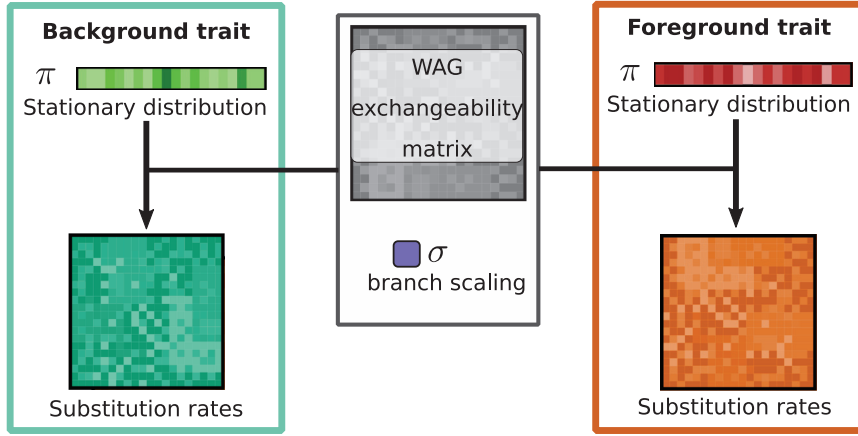


Figure 4.2: Schematic representation of the heterogeneous model (M_3) in TDG09 and Pelican. A different substitution process is defined for each phenotypic trait, depending on the stationary distribution that is specific of one trait.

of these sites, independently of the considered environmental condition.

$$M_2 : \quad Q_{ij}^0 = \sigma S_{ij} \pi_j^0 \quad (4.5)$$

As a result, two alternative models can describe the process responsible for the state of a protein site: one is the homogeneous model M_2 , where the stationary distribution is independent from conditions in the tree; the other is an heterogeneous model M_3 where two different stationary distributions of amino acid frequencies drive the evolution of a site depending on the phenotype. In our implementation, we extended this model to allow more than two conditions, so that more complex categorisations of phenotypes can be represented.

4.2.4 Hypothesis testing

In order to establish whether a site is associated with a given phenotype, one must decide which of the two available models provides the best explanation for the site composition. The homogeneous model M_2 serves as the null model that describes sites that evolved independently from the phenotype. It is also *nested* within M_3 , since it can be seen as a special case of the latter where $\pi^{\text{Av}} = \pi^{\text{Hu}}$. This nesting relationship makes it possible for us to use the likelihood ratio test (LRT) to test whether the more complex model fits the data significantly better than the null model.

The LRT test statistic is, as its name implies, a ratio of the likelihood of the full model to that of the reduced model. The null distribution is established by Wilks' theorem [Wilks, 1938] to be a chi-squared distribution, with a number of degrees of freedom equal to the difference in the number of free parameters between the two models. Importantly, this is only an asymptotic distribution, which implies that the actual distribution might be quite different from it when working with smaller samples.

$$\begin{aligned} D &= 2 \log \frac{L(M_A)}{L(M_0)} \\ &= 2 \left(\log L(M_A) - \log L(M_0) \right) \\ D &\sim \chi^2(\dim(\theta^{M_A}) - \dim(\theta^{M_0})) \end{aligned} \quad (4.6)$$

Given this null distribution, we are capable of computing a p-value that will measure the probability that the difference in likelihood is the result of chance, and not of statistical support in the data for the more complex model. Additionally, the computation of degrees of freedom of the null distribution is accurate at the condition that the true value under H_0 of any parameter does not lie on an edge of the parameter space. Parameter values may be bound within a definition domain. For example, a probability can only take values within the interval $[0; 1]$. In that case, if the true value of the parameter in the null model is either 0 or 1, it lies on a boundary of the parameter space and Wilks' theorem does not apply. The LRT takes into consideration the difference in the number of parameters between models. The richer model is penalized for each additional parameter that is fitted, so that the likelihoods of the two model can be performed in a meaningful way. The application of Wilks' theorem in the context of TDG09 and Pelican is discussed later on, in section 4.4.

4.3 Implementation and fitting of the model

In the maximum likelihood framework that we use, fitting the model consists in estimating the combination of parameter values that maximize the likelihood of the data assuming it was generated from the model under consideration. In that regard it is somewhat the inverse process of what is done when simulating: instead of generating data from parameters as input, parameters are estimated from the data. In simple models, such as linear models that I presented in the early chapters, the maximum likelihood estimate can be determined analytically; this is generally not the case when working with phylogenetic models, because the likelihood function is quite more complicated. Instead, parameter estimation is done using numerical optimization methods, that explore the parameter space using various strategies, searching for those that maximize the likelihood — ideally corresponding to a global maximum, but most commonly a local one. Although optima of the likelihood function are hard to identify analytically in a phylogenetic context because the state of ancestral nodes are not observed, the function itself can still be computed by integrating over each possible ancestral state. This is achievable by using an algorithm proposed by Joseph Felsenstein in the early 80's, which has remained a cornerstone of phylogenetic studies. I briefly present this algorithm as it may help to clarify the following sections.

4.3.1 Felsenstein's algorithm for phylogenetic likelihood computation

Computing the likelihood of a site, given a phylogenetic model with a set of parameters and a tree, can be achieved using Felsenstein's tree-pruning algorithm [Felsenstein, 1981]. It is a dynamic programming algorithm that recursively computes the likelihood of each state in the sequence alphabet at every node in the tree. As opposed to Gillespie's algorithm that I described in section 1.3.1, which instantiates one possible substitution story and propagates from the root of the tree, Felsenstein's algorithm integrates over all possible states at each node, starting for the tips of the tree. This is because the site that is observed at the tips can be the result of many possible paths of substitution events, that must be considered when computing its likelihood under a model. The likelihood is the probability to observe the data $x_{\mathcal{L}}$ at the leaves of the tree, under a model with parameters θ .

$$L_{\theta} = \mathbb{P}_{\theta}[X_{\mathcal{L}} = x_{\mathcal{L}}] \quad \text{where } X_{\mathcal{L}} \text{ is the set of leaves} \quad (4.7)$$

Because of the phylogenetic relationship, these observations are not independent from each other, and this probability can not be decomposed as a product of individual probabilities. However,

conditionally to the state of one node, the leaves of the sub-tree branching to the left are independent from those of the sub-tree branching to the right. Therefore, at the condition that the root node X_R is in state r , the likelihood of the site is

$$\begin{aligned}
 L_\theta &= \mathbb{P}_\theta[X_{\mathcal{L}} = x_{\mathcal{L}}] \\
 &= \sum_r \mathbb{P}_\theta[X_{\mathcal{L}} = x_{\mathcal{L}} | X_R = r] \mathbb{P}_\theta[X_R = r] \\
 &= \sum_r L_R(r) \mathbb{P}_\theta[X_R = r]
 \end{aligned} \tag{4.8}$$

In this expression, the term $L_R(r)$ is the likelihood of the tree that has R as a root, conditionally to the state of R being r — the parameter θ is involved but is omitted here and in the following equations as it remains constant throughout the application of the algorithm. This function can be expressed recursively at each node and leaf in the tree, which is the core idea of Felsenstein's algorithm.

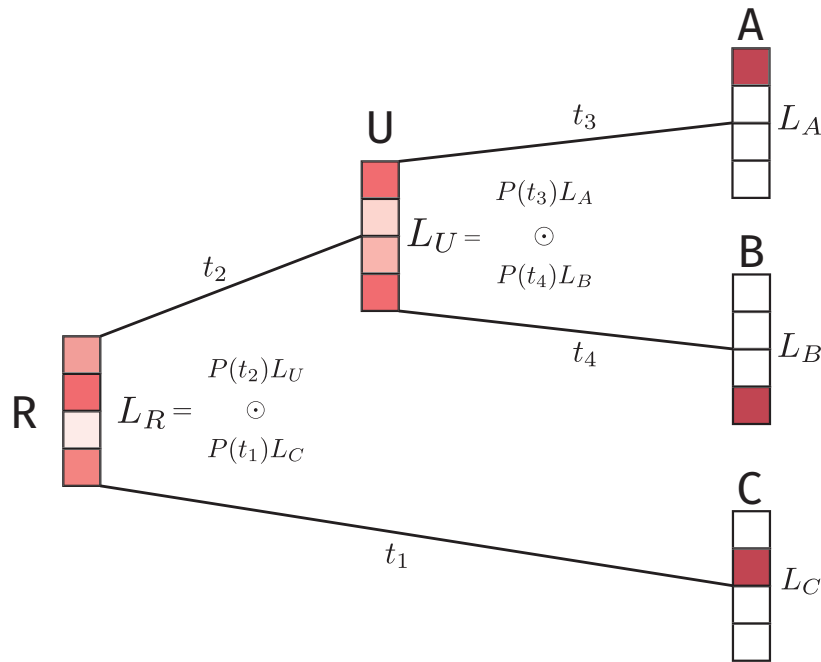


Figure 4.3: Illustration of Felsenstein's pruning algorithm with a 4 dimensional state space, such as the nucleotide alphabet. The picture can either be read left to right with a recursive point of view, or right to left with an incremental point of view. Likelihood values are symbolized by the intensity of the coloring in each cell.

At each leaf l , the likelihood conditionally to each possible state x is either 1, if it coincides to the observation ($x_l = x$) or 0 otherwise⁷. Then for an internal node n , the likelihood conditionally

⁷Although there might be uncertainty in the data, that can be reflected in the likelihood vector. For example, because sequencing methods are not perfectly accurate, the nature of nucleotides may not be completely determined at some positions. Information might be partial, e.g. limited to the identification as a purine (A or G, symbolized by R) or pyrimidine (T or C, symbolized by Y). This uncertainty can be represented by assigning a probability equal 1 to each of the two possible states [Yang, 2006]. The same applies to amino acid sequences, where similar amino acid are not always distinguished: e.g. leucine and isoleucine, symbolized by J.

to n being in state x is

$$\begin{aligned}
L_n(x) &= \mathbb{P}[X_{\mathcal{L}} = x_{\mathcal{L}} | n = x] \\
&= \prod_{c \in \mathcal{C}(n)} \sum_{y \in k} \mathbb{P}[X_{\mathcal{L}} = x_{\mathcal{L}} | c = y] \mathbb{P}[c = y | n = x, t_{nc}] \\
&= \prod_{c \in \mathcal{C}(n)} \sum_{y \in k} L_c(y) \mathbb{P}[c = y | n = x, t_{nc}]
\end{aligned} \tag{4.9}$$

where $\mathcal{C}(n)$ is the set of children of n , k is the state space, and t_{nc} is the branch length between n and c . It can also be written in the matrix form, as in figure 4.3

$$L_n = P(t_{nc_1})L_{c_1} \odot \dots \odot P(t_{nc_m})L_{c_m} \quad \text{where} \quad \begin{cases} \mathcal{C}(n) = \{c_1, \dots, c_m\} \\ P(t) \text{ is the transition probability matrix} \\ \odot \text{ is the element-wise vector product} \end{cases} \tag{4.10}$$

In this calculation, each child of n is considered independent conditionally to the state of n : therefore the likelihood at node n is the product of the likelihoods for each child node, conditionally to the parent state. The likelihood of a child node is integrated over each possible state x , for the probability of transition from the parent state x to y , in the time separating the two nodes (i.e. the branch length t_{nc}). This probability is weighted with the likelihood at the child node, conditionally to its state y , which is either observed if c is a leaf, or computed recursively using the same formula if it is an inner node in the tree.

The probability of transition from state x at the parent node towards state y at the child node $\mathbb{P}[y|x, t_{nc}]$ is determined by the transition probability matrix $P(t_{nc})$ that is derived from the transition rate matrix Q , defined in the model (see section 1.1).

The conditional likelihood with regard to each state in the alphabet is computed in this way at each node, starting from the tips of the tree, up to the root. The likelihood across the whole tree can then be computed as the sum of the conditional likelihoods at the root, weighted with a prior probability for each state. By making the assumption that the process is stationary, this prior probability can be chosen to be the stationary distribution.

$$L = \sum_{x \in k} \pi_x L_R(x) \tag{4.11}$$

Felsenstein's algorithm provides an efficient way to evaluate the likelihood function of comparative data, at one point of the parameter space that is considered in an evolution model. Most of the computation time is spent on performing matrix products, and on the determination of the matrix of transition probabilities P , a matter that is discussed in section 4.3.3.

Implementation detail: preventing underflow errors

The direct computation of the likelihood becomes problematic when working with larger trees, due to the repeated multiplication of probabilities, that produces numbers too small to be accurately manipulated in the limited representation of floating point numbers in computers⁸. In Pelican, this limitation is overcome using a strategy similar to that described in [Guindon, 2003, chapter 2;

⁸Most processing units implement the IEEE 754 standard to represent floating point numbers, typically over 32 or 64 bits of memory. As a consequence of the finite memory space, the precision is limited, and very small numbers can not be represented accurately.

section 2.4.7]. We scale the likelihood vectors by the inverse of their maximum value whenever their minimum value is below a threshold equal 1×10^{-6}

$$\begin{aligned} m &= \max(L) \\ S &= L/m \end{aligned} \tag{4.12}$$

where S is the scaled likelihood vector. The scaling factor is kept track of throughout the execution of the pruning algorithm. This is done by using a specialized data structure that we call a “shifted vector”, which contains the scaled vector and a “carry” term, that is the sum of all $\log m$ terms that have been used to scale likelihood vectors up to the current stage of the algorithm. A shifted vector can be easily transformed back to a vector of log likelihoods, by the following operation:

$$L = \log S + \text{carry} \tag{4.13}$$

This strategy thus prevents underflow issues resulting from the manipulation of very small numbers that are likely to occur when working with probabilities. It was implemented by Vincent Lanore in the OCaml library `phylogenetics`⁹, on which Pelican depends.

4.3.2 Numerical optimization

Felsenstein’s algorithm provides a way to evaluate the likelihood value associated to a model, at a given point in the parameter space. Since we aim to find the parameters that maximize the likelihood of the data within the scope of the model, this defines the *objective* function of an optimization problem.

Now as I mentioned at the beginning of this section, solving the equation that nullify the derivative of this function can not be analytically done for any arbitrary tree and model¹⁰, which prohibits finding maximum likelihood estimators of parameters. For this reason, parameter estimation is generally performed using numerical optimization, that scan the parameter space for values that maximize the likelihood. There is a large diversity in strategies for exploring the parameter space, and since numerical optimization is a huge field of research in itself, I will not attempt to give an overview of the subject. Our implementation relies on the Nelder-Mead algorithm [Nelder and Mead, 1965] for numerical optimisation, which is the one used in the original implementation of TDG09.

Parameters are represented in log scale during the optimisation. This is both to avoid reaching negative values, which make no sense in our setting, and to improve the optimizer behaviour on small values. The scale parameter σ is transformed from its log representation to linear by taking its exponential. To enforce that amino acid frequency profiles sum to 1, the corresponding parameter vector that is optimized is transformed using a so-called softmax function to obtain the actual vector of frequencies which can then be used in the pruning algorithm.

$$\pi_i = \frac{e^{v_i}}{\sum_j e^{v_j}} \quad \text{where } v \text{ is the vector of log parameters} \tag{4.14}$$

During the optimization, frequency parameters are bounded to avoid reaching frequencies values too low that could interfere with the eigen-decomposition of the resulting rate matrix. The scale parameter is also restricted to plausible values determined after the literature (see e.g. [Mayrose,

⁹<https://github.com/biocaml/phylogenetics>

¹⁰Some analytic solutions have been identified in very specific settings, on trees having few taxa, and only for a restricted class of models. See e.g. [Chor and Snir, 2007].

2004]), within the interval roughly defined in $[1 \times 10^{-2}; 10]$ — we are a bit more permissive and allow values within the interval $[2 \times 10^{-3}; 20]$. This is done by assigning the log-likelihood at out-of-bound parameters to be negative infinity, which causes the optimizer to reject such parameter points.

4.3.3 Efficient matrix exponentiation

In the formalism of continuous Markov chains, an evolution model defines the transition rate matrix Q , which represents the intensity of the “flow” of changes between states per unit of time. As I explained in the previous section, Felsenstein’s pruning algorithm works with probabilities of changes between states in a given time frame t — that is generally the length of a branch. These probabilities can be assembled in a square matrix $P(t)$. In order to compute the likelihood associated to a model, we therefore need to determine this transition probability matrix from the rate matrix, accounting for the length of the branch. This can be done by solving the equation $P'(t) = P(t)Q$ with respect to t , the solution of which is $P(t) = e^{Qt}$ as I previously exposed in [chapter one](#).

Now we need to compute the exponential of the rate matrix Q , after it was scaled by the evolution time, which is typically a costly operation. The exponent of a matrix can be computed using the Taylor series expansion, a convergent sum of powers

$$\begin{aligned} P(t) = e^{Qt} &= I + Qt + \frac{1}{2!}(Qt)^2 + \frac{1}{3!}(Qt)^3 + \dots \\ &= \sum_{i \rightarrow \infty} \frac{1}{i!}(Qt)^i \end{aligned} \quad (4.15)$$

Computing $P(t)$ using this expression would be extremely inefficient, especially when working with larger state spaces such as the amino acid or codon alphabets. A better approach relies on the diagonalization of Q through spectral decomposition

$$\begin{aligned} Q &= XDX^{-1} \quad \text{where } D \text{ is diagonal} \\ P(t) = e^{Qt} &= e^{XDtX^{-1}} = \sum_{i \rightarrow \infty} \frac{1}{i!}(XDtX^{-1})^i = X \left(\sum_{i \rightarrow \infty} \frac{1}{i!}(Dt)^i \right) X^{-1} = Xe^{Dt}X^{-1} \end{aligned} \quad (4.16)$$

In this expression, the calculation of e^{Dt} is trivial since D is diagonal, and e^{Qt} can then be obtained at the expense of only two matrix multiplications. The values in D are the eigenvalues of Q , and X is assembled from the corresponding eigenvectors.

The matrix decomposition itself is not always trivial: not all matrices are diagonalizable, and when they are, the operation requires solving a linear system to determine the eigenvalues and eigenvectors of Q . Fortunately, this problem is made easier when working with matrices having some special properties. In particular, real symmetric matrices are always diagonalizable, and efficient algorithms exist to derive the corresponding eigenvalues and eigenvectors. Importantly, the resulting change-of-basis matrix is orthogonal, so that $X^{-1} = X^T$; this property will come in handy to solve our problem. The decomposition of a symmetric matrix A is thus

$$A = XDX^T \quad (4.17)$$

This is all fine and well, however the rate matrix Q in the GTR model is generally not symmetric. This becomes manifest if we recall that the rate matrix Q is computed from the exchangeability

matrix S (which is symmetric) and the stationary distribution π :

$$Q = S \cdot \text{diag}(\pi) \Rightarrow Q_{ij} = S_{ij}\pi_j, \quad i \neq j \quad (4.18)$$

Although Q is not symmetric, we can still perform a change of basis to conjure up a similar symmetric matrix Q' [Yang, 2006, p.68] [Schabauer et al., 2012], that has identical eigenvalues. The expression of Q' is

$$Q' = \sqrt{\Pi}Q\sqrt{\Pi}^{-1} \quad \text{where } \Pi = \text{diag}(\pi) \quad (4.19)$$

As a diagonal matrix, the expression of $\sqrt{\Pi}$ and its inverse is trivial, provided that no stationary frequency is null – otherwise, Π^{-1} would be undefined. This condition can be ensured by using a sparse specification for the model, where only observed amino acids at one site are included in the state alphabet, thus preventing the presence of null frequencies in the stationary distribution. This strategy is described and discussed in the next section. We can verify that Q' is symmetric, using that S is symmetric:

$$Q'_{ij} = \frac{\sqrt{\pi_i}}{\sqrt{\pi_j}}Q_{ij} = \frac{\sqrt{\pi_i}}{\sqrt{\pi_j}}S_{ij}\pi_j = \sqrt{\pi_i}\sqrt{\pi_j}S_{ij} = \sqrt{\pi_j}\sqrt{\pi_i}S_{ji} = Q'_{ji} \quad (4.20)$$

We then exploit that fact to perform this decomposition using an efficient routine¹¹ implemented in the LAPACK library [Anderson et al., 1999], and obtain

$$Q' = LDL^T = LDL^{-1} \quad (4.21)$$

where L is an orthogonal matrix. It follows that

$$Q = \sqrt{\Pi}^{-1}Q'\sqrt{\Pi} = \sqrt{\Pi}^{-1}LDL^T\sqrt{\Pi} \quad (4.22)$$

The expression of Q can be simplified, by defining

$$\begin{aligned} U &= \sqrt{\Pi}^{-1}L \\ U^{-1} &= \left(\sqrt{\Pi}^{-1}L\right)^{-1} = L^{-1}\sqrt{\Pi} = L^T\sqrt{\Pi} \end{aligned} \quad (4.23)$$

We finally get

$$\begin{aligned} Q &= UDU^{-1} \\ P(t) &= e^{Qt} = e^{UDtU^{-1}} = Ue^{Dt}U^{-1} \end{aligned} \quad (4.24)$$

In summary, the calculation for the matrix exponential in the expression of $P(t)$ is made more efficient by leveraging spectral decomposition, in a two-step process: first perform a well chosen change-of-basis on the Q matrix to bring up a symmetric matrix Q' , that can be efficiently diagonalized into an LDL^T form. Importantly, the decomposition of Q' , which is the most costly operation in this procedure, only needs to be performed once for each rate matrix Q in the model. The TDG09 model consists in one stationary distribution π^k per condition k in the tree, and each of them determines one transition rate matrix Q^k . Consequently, a single computation of D , U and U^{-1} can be done for each condition, and used to evaluate $P(t)$ on any branch with length t , choosing the set

¹¹The LAPACK routine `dsyevr` requires a real symmetric matrix as input, and computes its eigenvalues and eigenvectors. See <http://www.netlib.org/lapack/explore-3.1.1-html/dsyevr.f.html>

of precomputed matrices that match the condition found on the branch. This is also true regarding the homogeneous model, where the decomposition can be done once and for all, and harnessed to speed up the computation of the probability matrices throughout the whole tree.

Efficient exponentiation in the context of the tree pruning algorithm

We made a small improvement on the original implementation of TDG09, that already relied on the eigen-decomposition of the rate matrix to efficiently compute its exponential. It consists in refraining ourselves from immediately computing the transition probability matrix using the eigen-decomposition, so that a more efficient computation of the likelihood at internal nodes in the tree can be performed during the pruning. This was suggested by our colleague Nicolas Lartillot, whom I thank for that.

Let us recall that the transition probability matrix $P(t)$ is involved in the computation of the likelihood of a tree, at one point of the model parameter space, using Felsenstein's pruning algorithm. The vector L_n of conditional likelihoods at a node n , with children $\mathcal{C}(n)$, is

$$L_n = P(t_{nc_1})L_{c_1} \odot \dots \odot P(t_{nc_m})L_{c_m} \quad \mathcal{C}(n) = \{c_1, \dots, c_m\} \quad (\text{from eq 4.10})$$

Replacing $P(t)$ by its expression as a matrix decomposition, we get the following expression for each partial likelihood term with regard to a child c

$$L_n^c = P(t)L_c = Ue^{Dt}U^{-1}L_c \quad (4.25)$$

A naive way to compute the partial conditional likelihood L_n^c at node n with regard to child c would be to first compute the transition probability matrix $P(t)$, and then perform the matrix-vector product with the likelihood vector of the child node L_c . However, computing $P(t)$ involves multiple matrix product operations which can be avoided. Instead, L_n^c can be computed by performing only matrix-vector products using a strategy where calculation are done from right to left:

$$L_n^c = U(e^{Dt}(U^{-1}L_c)) \quad (4.26)$$

In this manner, only $3n^2$ multiplication operations are performed, instead of $2n^3 + n^2$ in the naive form (where n is the dimension of the state space). This allows for a more efficient computation of the likelihood vector, by avoiding the calculation of matrix products and performing matrix-vector products instead which are less computationally intensive.

4.3.4 State space reduction

The dimension of the state space in the TDG09 model has a large impact on the computational cost of its optimization. The naive specification involves 20 free parameters in the reduced model M_2 and 39 in the full model M_3 : one scale parameter, and 19 parameters per stationary frequency vector. The size of the transition rate matrix increases with the square of the number of frequency parameters, with repercussions on the amount of work required for matrix computations throughout the execution of Felsenstein's tree pruning. Notably, the computation of the transition probability matrix $P(t) = e^{Qt}$ along a branch, involves the exponentiation of the rate matrix Q , which has $O(n^3)$ complexity. Any reduction in the dimension of the state space is thus a mean to substantially decrease the computational cost.

Just like the original one for TDG09, our implementation reduces the dimension of the state space by ignoring unobserved amino acids altogether, exploiting that sites in alignments commonly display a restricted number of distinct amino acids. This is a heuristic that does not work in the general case, but only when the exchangeability rates between states are neither null nor too low. Indeed, when the exchangeability between two states is too low, transitions occur preferably by going through an intermediary state that is easier to reach; in this case, removing states that could potentially have such a role is problematic. This typically happens when working with codon models: substitutions between codons that differ by more than two nucleotides are generally not allowed. In such cases, reduction of the state space is preferably done by collapsing unobserved states into a single one [Davydov et al., 2016]¹². However, this is not an issue in our context since we use empirical amino acid exchangeabilities, that do not include such low replacement rates.

This truncation of the state space helps to reduce the computational cost of the pruning algorithm, and improves the stability of the numerical optimisation of parameters, as it is difficult to reach very low frequencies — and impossible to attain null ones — in the amino acid profiles. We are thus working with a *sparse* state space, as opposed to the *dense* representation were all amino acid states are included. Figure 4.4 illustrates that, with regard to the maximum likelihood that is obtained, both approaches are equivalent. It was obtained by fitting the stationary model of Pelican on a simple binary tree with amino acid *L* or *V* at the leaves, first using the dense representation, then the sparse one. The scale parameter σ was made variable within an interval to better emphasize the adequacy between the two specifications.

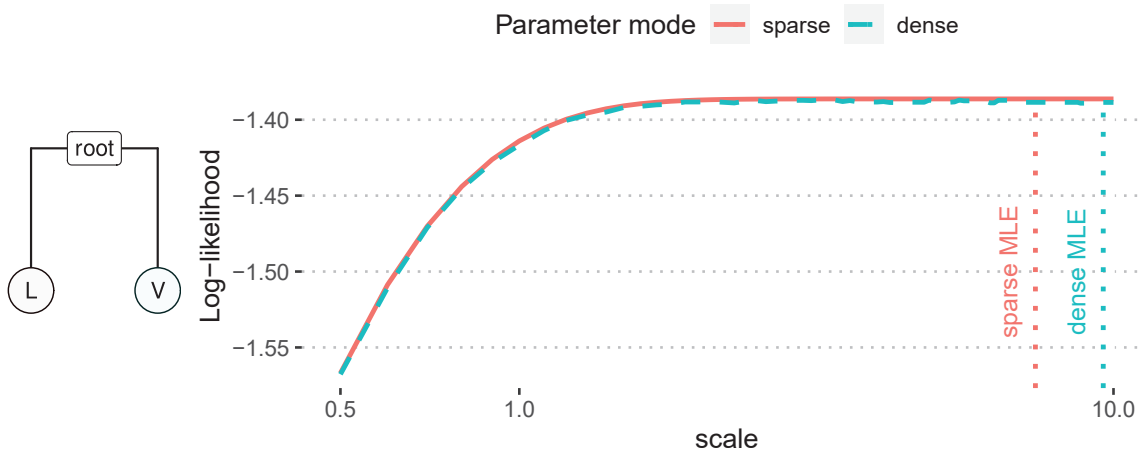


Figure 4.4: The maximum likelihood obtained when fitting the model is equivalent when using the sparse and dense specifications. Dotted lines indicate the MLE of the scale parameter when it is left free.

We can verify the equivalence between the sparse and dense representations more formally, by checking that they end up describing the same Markov process when the unobserved amino acid frequencies are null. Denoting U the set of unobserved amino acids, and C the set of conditions across the tree:

$$\pi_j^k = 0 \quad \forall j \in U, k \in C \quad (4.27)$$

In order to prove this equivalence between the two representations, let us recall the definition of the

¹²We implemented this technique but found that it did not improve the fit of the model, and was slower than the truncated representation.

transition rate matrix in our GTR model:

$$Q_{ij}^k = \sigma s_{ij} \pi_j^k \quad (4.28)$$

It follows that the transition rates towards any unobserved state is null. This implies that, in the dense representation of the substitution process, unobserved states at the tip of tree can not be reached from other states.

Now let us also bring up the fact that under our model, the amino acid frequencies at the root of the tree are at the equilibrium π^r , with r denoting the condition at the root of the tree. This means unobserved amino acids can not be the ancestral state in the tree. Two implications can be derived from this:

- From a top-down point of view, the frequencies of amino acids from the set of unobserved is null at the root. Combining this with the impossibility to reach any amino acid in U from other states, it follows that the amino acids in U are never seen throughout the tree if we were to simulate under the generative model. In Markovian terms, this is like modeling a state for unobserved amino acid that can be moved away from, but never reached, and is not the initial state of the process: its presence or absence makes no difference regarding the process that is modelled.
- From a bottom-up point of view, the likelihood across the whole tree using Felsenstein's pruning algorithm is $L = \sum_i \pi_i L_{root}(i)$. In general, π is a prior probability for each state at the root of the tree; in our case this coincides with the equilibrium frequencies $\pi^r | \pi_i^r = 0, \forall i \in U$. As a consequence, the likelihood of the whole tree is not affected by the conditional likelihood values obtained at the root for any unobserved amino acid.

In conclusion, the truncated representation behaves in the same way as a dense model, where the equilibrium frequencies are set to 0 for non-observed amino acids. The use of this representation allows drastically reducing the computational footprint of the estimation of parameters in some cases, and avoids numerical issues during the optimization. Importantly, this heuristic can be applied to our GTR model of amino acid sequence evolution, where any state transition can be achieved in one step. It would not be feasible in a model where some exchangeabilities are null, as in the instance of codon substitution models where multiple nucleotide substitutions are generally not allowed. In such situations, a strategy based on state aggregation would be a better choice. Also, the fact that we do not observe some amino acids does not imply that they are not allowed at all in the profile, i.e. given a larger sample of sequences some previously unobserved amino acid could appear. A Bayesian approach could better account for this uncertainty by estimating a posterior distribution for the frequency of each amino acid, even those that were not observed.

4.4 On hypothesis testing and the applicability of LRT

We test at each site the hypothesis that the substitution process is dependent on a phenotypic trait, or an environmental condition. To do so, a likelihood ratio test (LRT) is employed to compare the likelihood of two nested models: a condition-dependent model of evolution, and a homogeneous model. The latter can be seen as a special case of the former, where all the stationary distributions π of amino acid frequencies are equal across all conditions, thus defining one substitution process

across the whole tree. Because the model representation is sparse (see section 4.3.4), the dimension of any stationary distribution π is the number of distinct amino acids observed at the site.

Likelihood ratio tests are based on Wilks' theorem to provide an asymptotic null distribution for the D statistics derived from the likelihood ratio. Under Wilks' theorem, the null distribution for D is a chi-squared distribution with a number of degrees of freedom equal to the quantity of additional parameters introduced in the heterogeneous model. That is the dimension of π – i.e. the number of distinct amino acids observed at the site, since we are working in a sparse representation as explained in section 4.3.4 – minus one to account for the fact that π sums to 1 so that one parameter can be deduced from the others

$$D \sim \chi^2(\dim(\pi) - 1) \quad (4.29)$$

4.4.1 Wilks' asymptotic null distribution: the problem of the effective sample size

Being asymptotic, the null distribution of the test statistic is known for infinite samples, and can be used in practice when the sample size is large enough. Now this may be problematic in the context of phylogenetic analysis, where the sample size is limited, and the observations are correlated to each other due to their phylogenetic relationship. This translates to a reduction in what I call the “effective” sample size which can be enough to make the null distribution drift away from the theoretical χ^2 defined in Wilks' theorem. The adequacy of the actual null distribution to the theoretical χ^2 is thus dependent on the number of leaves in the tree (or equivalently the number of sequences in the alignment), and the correlation structure between the observations.

Let us bring this to light using a few examples. In each of these experiments, we simulate 1000 sites along a tree, using the null model where the π parameter consists in 3 non-zero stationary frequencies, with value 1/3. Both the null and condition-dependent models are then fitted at maximum likelihood, and the corresponding likelihood ratio statistics is computed at each site. The distribution of these statistics is an empirical null distribution, that we can compare to the theoretical chi-square distribution, with 2 degrees of freedom.

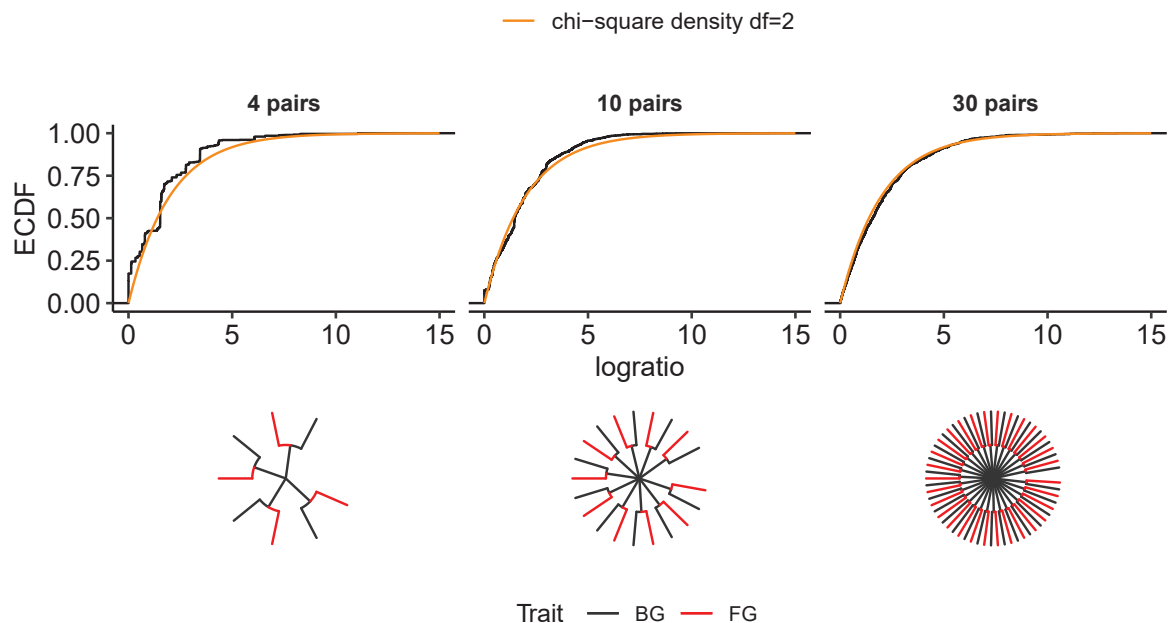
Experiment 1: increased number of leaves improves asymptotic convergence

Figure 4.5: Asymptotic convergence is improved with the number of tips in the tree.

In the first experiment, we generate three trees made of pairs (Fig. 4.5), with all branch lengths equal 0.5, and increasing size: 4 pairs, then 10, then 30. Simulations are done as described earlier, with a constant scaling of the substitution rate ($\sigma = 1$). The empirical distribution approaches the theoretical distribution as the total number of leaves increases.

Experiment 2: faster evolution rate improves asymptotic convergence

As a second experiment, we use the same 10 pairs tree to simulate sites with variable scale parameter $\sigma \in \{0.1, 1, 10\}$. Decreasing σ is equivalent to shortening all branches in the tree, thus increasing the correlation between each observation at a site: the null distribution deviates further from the asymptotic χ^2 . In contrast, increasing σ in the simulation parameter has the effect of making each state at the site more independent from the others: thus the effective sample size is increased, and the actual null distribution is better adjusted to the asymptotic distribution.

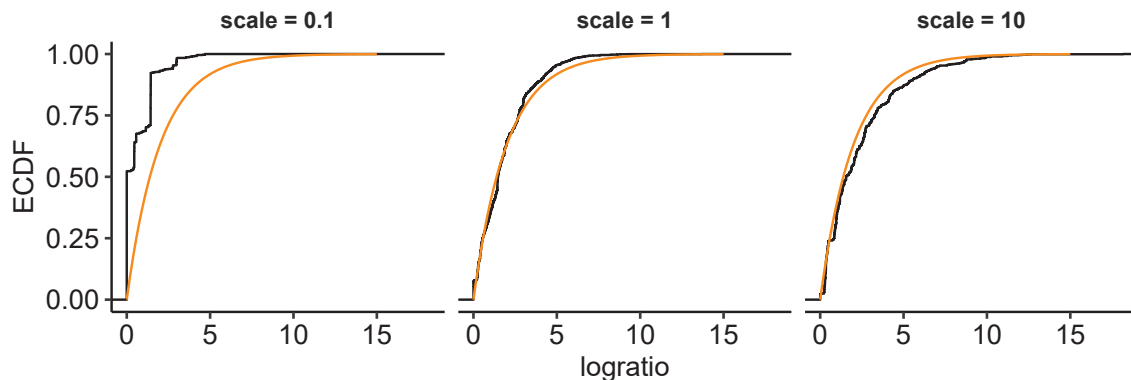


Figure 4.6: Asymptotic convergence is improved with the branch length.

Empirical phylogenies

This dependency on the evolution rate is also observable on real phylogenies. The Orthomam tree has 116 leaves and short branches (its length from root to tip is about 0.3 substitutions). The leaves are thus highly correlated to each other, and the effective sample size is very limited. As a consequence, the distribution of log-ratios under the null when simulating with lower scaling values for the branch lengths (e.g 0.1; 1) does not match the theoretical distribution under the null as established by Wilk’s theorem. Increasing further — and beyond reasonable evolutionary times — the length of branches brings the distribution of logratios closer to the theoretical χ^2 distribution, as depicted figure 4.7. In comparison, the HIV tree is both larger, with 476 leaves, and longer than the Orthomam tree: the null distribution better matches the asymptotic chi-square at lower scaling values of the branch lengths.

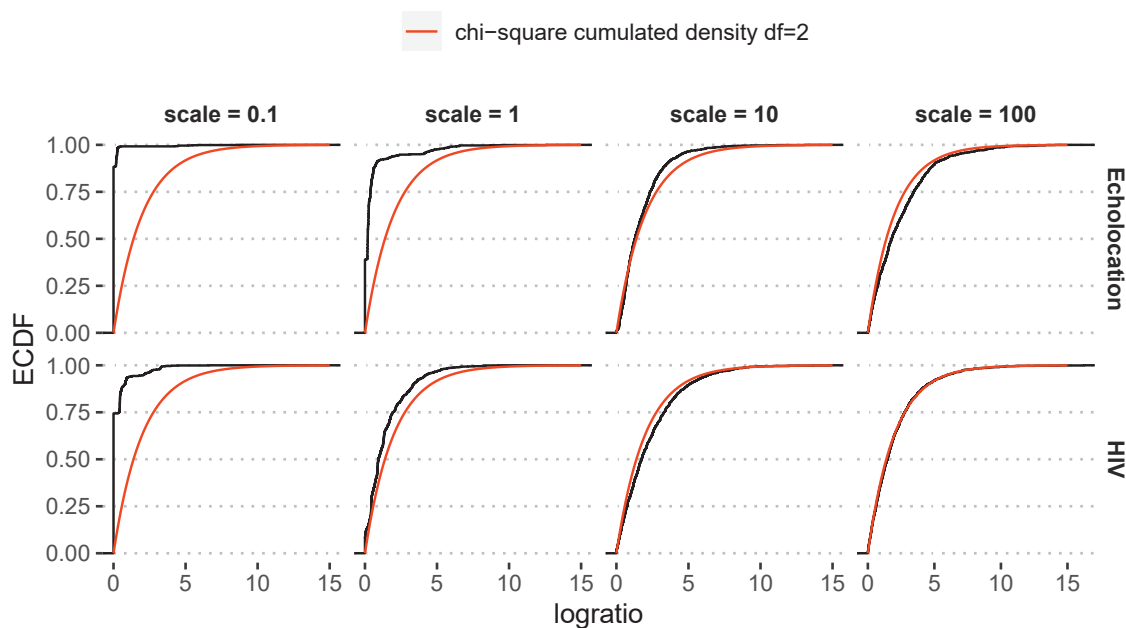


Figure 4.7: Asymptotic convergence to the null distribution improves with the length of the tree, when using empirical phylogenies. The convergence is also improved with the number of tips: the empirical distribution is closer to the theoretical one with the HIV tree than with the Echolocation tree.

In conclusion, the asymptotic chi-square distribution used to test which of the two model fits best has to be expected to be quite different from the actual null distribution, because of the correlation structure underlying the data that reduces the effective sample size. In practice, relying on the asymptotic distribution might be “good enough” in most cases, but we have to be wary of these considerations before interpreting Pelican’s results on a dataset.

An important consequence of using the chi-square approximation when the true null distribution strongly deviates from it, is that the p-values resulting from the test are not calibrated: their distribution under the null hypothesis is not uniform, and they can not accurately be used as a significance level when setting an alpha risk threshold.

4.4.2 Degrees of freedom

The problem with the limited effective sample size is not limited to the lack of convergence of the null distribution towards the asymptotic chi-square: it may also distort the computation of the number of degrees of freedom used as the chi-square parameter. As a reminder, the number of degrees of freedom in the LRT is the number of additional free parameters introduced in the condition-dependent model: that is one frequency parameter for each observed amino acid, minus one because the stationary distribution sums to 1.

The number of degrees of freedom tends to be under-estimated

In our model, the amino acid composition observed at one site is a reflection of the stationary distribution of amino acid frequencies that drives its substitution process. This reflection gets more accurate with greater effective sample sizes: trees with more tips, and longer branches will exhibit sites where the distribution of amino acid frequencies better matches the stationary distribution.

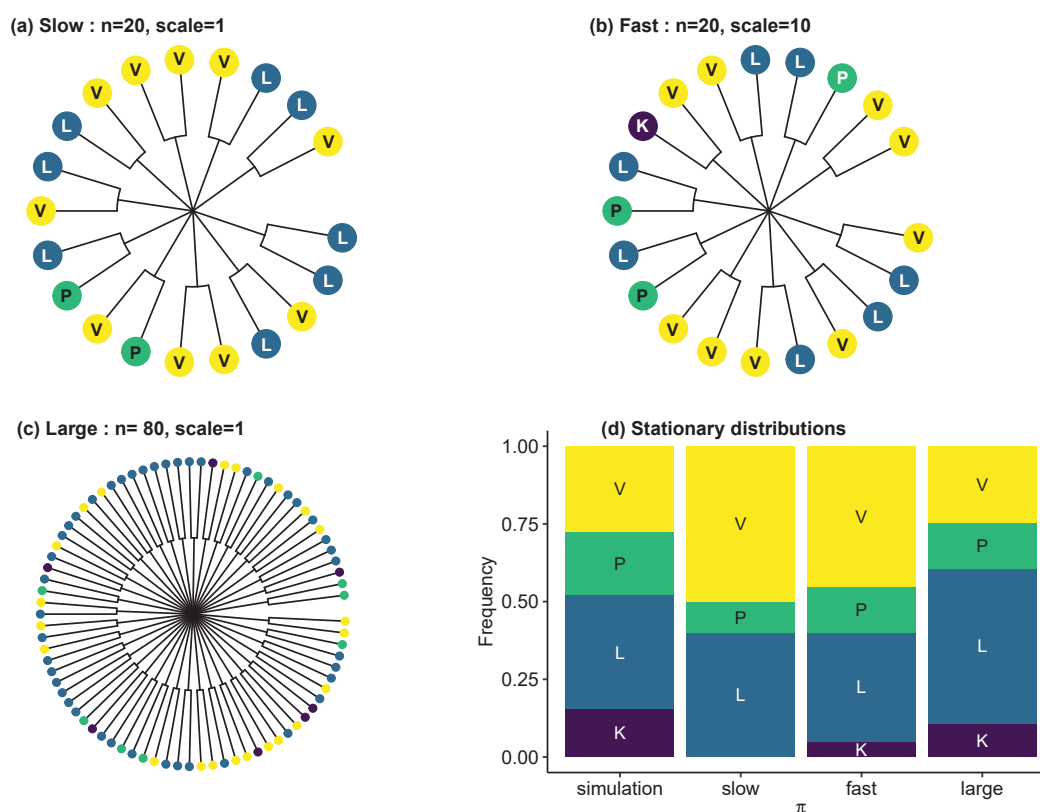


Figure 4.8: When simulating with a profile containing non-null amino acid frequencies, some of them may be unobserved if the tree is short and small.

Importantly, due to low effective sample sizes, some amino acids may not be observed in a tree at all, although they would be observed given more evolution time in the tree. Figure 4.8 illustrates this situation, using a protein site that is simulated under the homogeneous model. The simulation parameter π has non-null stationary frequencies for 4 different amino acids (K, L, P, V), represented as a stacked bar plot (figure 4.8(d)).

This parameter is first used to simulate a site from a pair tree with 20 tips, with no change to the substitution rate ($\sigma = 1$; figure 4.8(a)). Because of the small number of tips, and the low evolution time separating the root from the tips, no lysine (K) is observed at the site, although it is present

in the true stationary distribution. In consequence, the maximum likelihood estimate (MLE) of the stationary distribution has null frequency for K (see section 4.3.4 for an in-depth explanation), and there is thus one less free parameter compared to the true stationary distribution. The vanishing of K from the estimate is the result of the small effective sample size on this site.

When increasing the substitution rate ($\sigma = 10$; figure 4.8(b)), more substitutions are allowed to occur along the tree branches, and the resulting site better reflects the stationary distribution. In particular, a lysine is observed at the site, and has non-null frequency in the MLE of the stationary distribution.

We can also play with the number of tips while keeping the substitution rate low ($n = 80, \sigma = 1$; figure 4.8(c)) and take note that all 4 amino acids are observed, and that the corresponding frequencies match the simulation parameter.

In summary, low effective sample sizes may induce the absence of amino acids that would otherwise be observed given a large enough sample, i.e. longer branches and/or more tips.

Consequences on hypothesis testing

I have discussed in section 4.4.1 how low effective sample sizes degrade the adequacy of the null distribution of log-ratios with the theoretical χ^2 distribution, when performing the likelihood ratio test to compare the fit of the two alternative models. The problem at hand is of different nature, although the root cause is the same, in that it affects the parameterization of the asymptotic distribution. The number of degrees of freedom of the χ^2 is determined by the number of additional free parameters in the more complex (heterogeneous) model: that is the number of equilibrium frequencies in the sparse representation of the stationary distribution, minus one. Any “potential” amino acid that is present in the *true* null distribution, though unobserved at the site, will have null frequency in the maximum likelihood estimate, and be absent from the sparse representation of the stationary distribution. In consequence, one free parameter is lost in comparison to the “true” model, and one degree of freedom is missing in the null distribution.

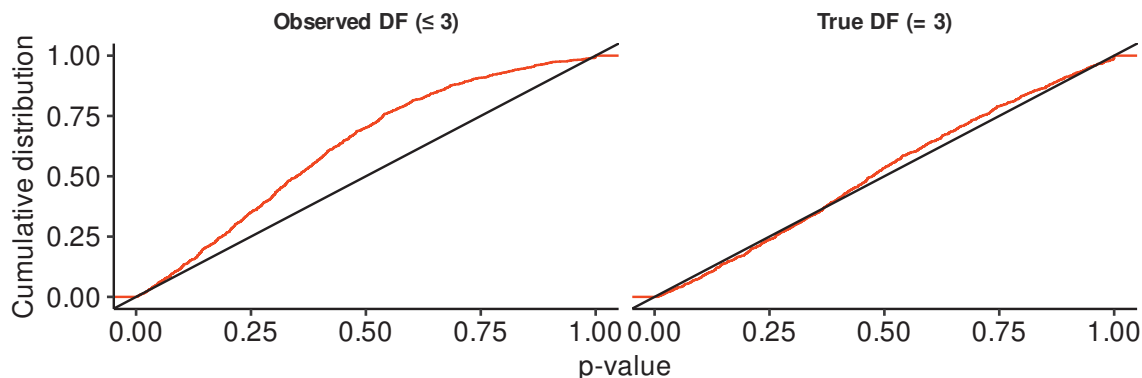


Figure 4.9: P-values under the null hypothesis deviate from the uniform distribution when the number of degrees of freedom is underestimated as a consequence of the limited sample size induced by the dimension of the tree. Using the known true number of degrees of freedom restores the uniformity of the distribution of p -values under the null.

Ultimately, the inappropriate reduction in the number of degrees of freedom makes the test more sensitive. For a given value of log-ratio, the corresponding p -value is lower when degrees of freedom are decreased. To illustrate this, we simulated 1 000 sites using a sparse profile including four amino acids with non null frequencies, along a random birth-death tree with 20 tips. We then computed

the empirical distribution of p -values, using either the observed number of distinct amino acids at each site to determine the degree of freedom of the null distribution (figure 4.9 left panel), or the number of non null frequency parameters in the simulation that gives $(4 - 1) = 3$ degrees of freedom for all sites (right panel). As expected, we find that the distribution of p -values computed with the observed number of degrees of freedom is skewed towards lower p -values in comparison the one using the true number of degrees of freedom.

Now seems to be a good time to introduce the notion of *calibration* for a statistical test. A test has the property of calibration when the p -values resulting from its application under the null hypothesis are uniformly distributed. This property, when verified, guarantees that a p -value effectively represents the risk of type 1 error, which is the probability that a null hypothesis, knowing it is true, is rejected nevertheless. It depends entirely on the condition that the null distribution of the test statistic is accurate — in our case, this is the chi-square distribution for the log-ratio of likelihoods, which is accurate when asymptotic conditions are met.

In practice, we can evaluate the calibration of a test by comparing the distribution of p -values under the null hypothesis to the uniform. This is what was done in the previous experiment: figure 4.9 shows that the distribution of p -values is not uniform when the chi-square distribution of log-ratio under the null is badly parameterized; but it is so when we know the true number of degrees of freedom that shape the distribution.

In a frequentist testing framework, calibration is an important property as it gives us the ability to quantify to some extent the error that we make. Without it, we may not consider the resulting p -values as representative of a probability for type 1 error, which prevents us from even setting a significance threshold — which is their most common use-case. In a sense, p -values from an ill-calibrated test should not even be referred to as such. Furthermore, this makes the usual procedures for meta-analysis inapplicable, such as those aimed at computing the false discovery rate (FDR) from multiple independent test. This is indeed problematic in our case, as we would benefit from integrating over the site-wise results provided by Pelican to identify genes that are associated to a given phenotype; this is the object of [chapter six](#), where I discuss of this issue and our attempts at working around it.

Assessing the calibration on simulations using empirical phylogenies

For now, as I have presented some rather artificial cases that disrupt the calibration of the test, we may ask how that translates to more realistic settings. Using the now familiar six empirical phylogenies to simulate alignments under the null mutation-selection model where fitnesses are not tied to the phenotype, we can gather the p -values obtained from the scan using Pelican, and compare their distribution to the uniform.

Figure 4.10 makes it obvious that the method is not well calibrated in most cases. There is generally an accumulation of p -values within the interval roughly delimited by $[0.5; 0.7]$. On simulation using the Influenza phylogeny, the deviation from the uniform seems to be more limited, possibly thanks to the dimension of the tree. Note however that there is an enrichment in low p -values, that we do not observe on other datasets: this has consequences on the aggregation of p -values across genes, that I discuss in the corresponding [chapter six](#).

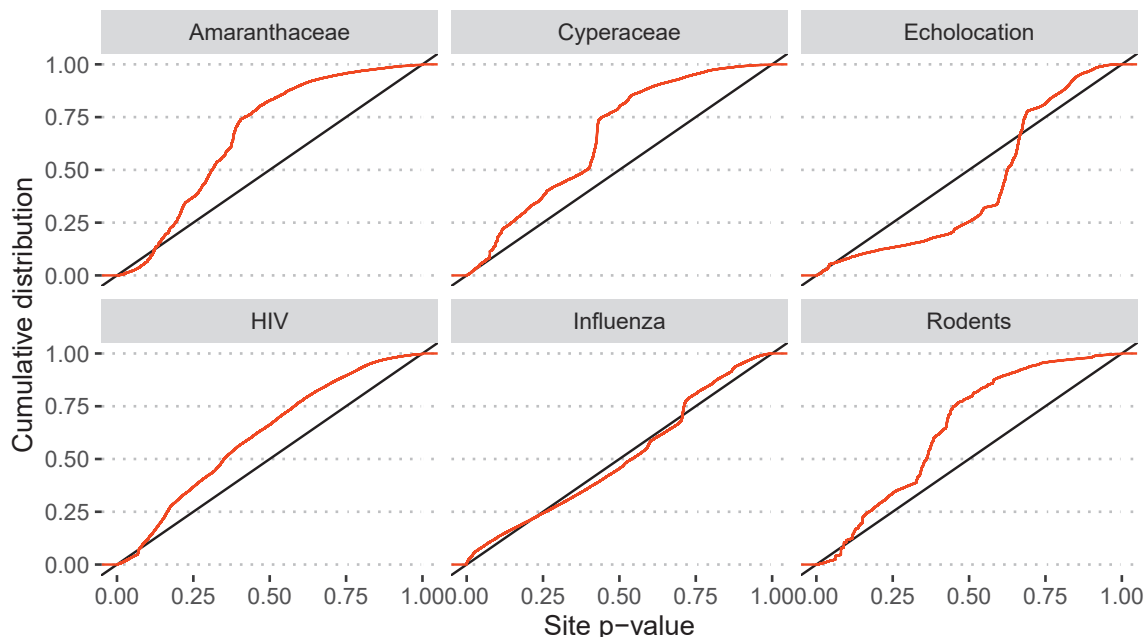


Figure 4.10: Calibration of Pelican evaluated using simulation under the null on empirical phylogenies. The distribution of p -values is shown as an empirical cumulative distribution function (ECDF) (red), compared to the uniform CDF (black).

4.5 Improving the detection of relaxed selection

4.5.1 Motivation: a pathological case of Influenza

Among the scenarios that were simulated in chapter 3 to assess the ability of methods to detect changes in the fitness landscape, one of them involved the relaxation of purifying selection. It is one particular situation of differential selection, where the fitness profile in the MutSel simulation model shifts from favoring a few amino acids in the ancestral condition under purifying selection, to a flat profile where all amino acids have equal fitnesses in the relaxed condition. This should produce some signal that Pelican could exploit, by inferring distinct amino acid stationary distributions between conditions.

We found that, in the majority of the datasets subjected to the benchmark, Pelican could distinguish sites that went through a shift towards neutral selection from sites under constant purifying selection almost as well as `codeml`, which was found to be the best at detecting them (figure 3.5). However, the precision-recall measured for Pelican on the Influenza dataset under relaxed selection was surprisingly bad, positioning the method as the worst one in this situation. I explore in this section why Pelican fails so hard on this particular dataset, and what improvements could be done to the method to make it more robust.

Diagnosis of the problem

The ancestral condition in the influenza tree spans the smaller sub-tree of strains residing in avian hosts. It is under purifying selection in this experiment, where a few amino acids are favored, while the others are strongly selected against. The other sub-tree depicts the phylogeny of human-specific strains, under relaxed selection in our simulation with a flat fitness profile where all 20 amino acids have equal fitness. The influenza tree (appendix figure B.6) is both large (432 leaves) and long as

depicted in figure 4.11, with the two sub-trees defined by the host condition diverging from the root. Because of these features, the sites simulated under differential selection nicely reflect the fitness

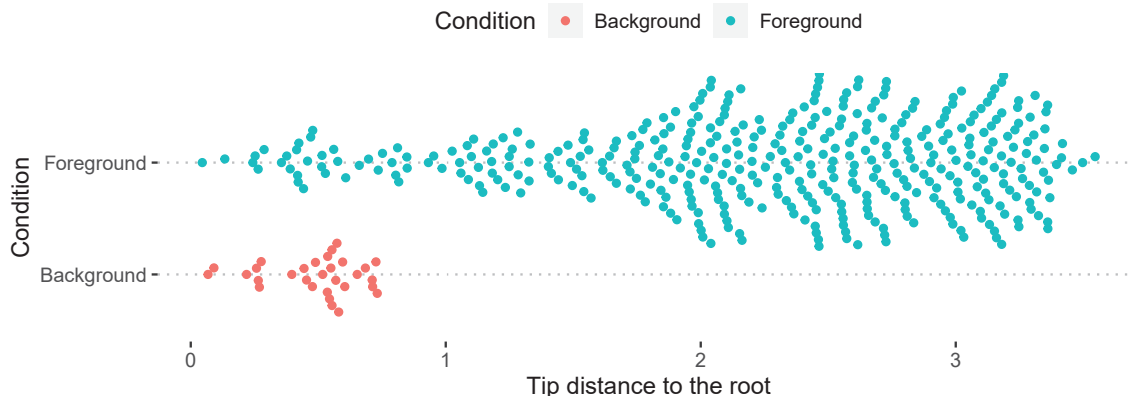


Figure 4.11: The foreground sub-tree in the Influenza phylogeny is both larger and longer than the background tree.

profiles: few distinct amino acids are displayed among avian-host strains¹³, while human-host strains regularly exhibit the almost complete amino acid alphabet, as depicted in figure 4.12.

On the basis of what I have discussed about Pelican so far, especially concerning the asymptotic condition in section 4.4.1, this dataset appears to be ideal for analysis using Pelican: the effective sample size is large enough so that we can expect that the null approximation using Wilks' chi-square distribution is accurate, and the stationary distributions are well represented in the sites. Nevertheless, something is undoubtedly confusing the detection of selective pressure shifts. We need to relate multiple factors to explain this: the features of the influenza tree, the nested models constraint and parameter sparsity.

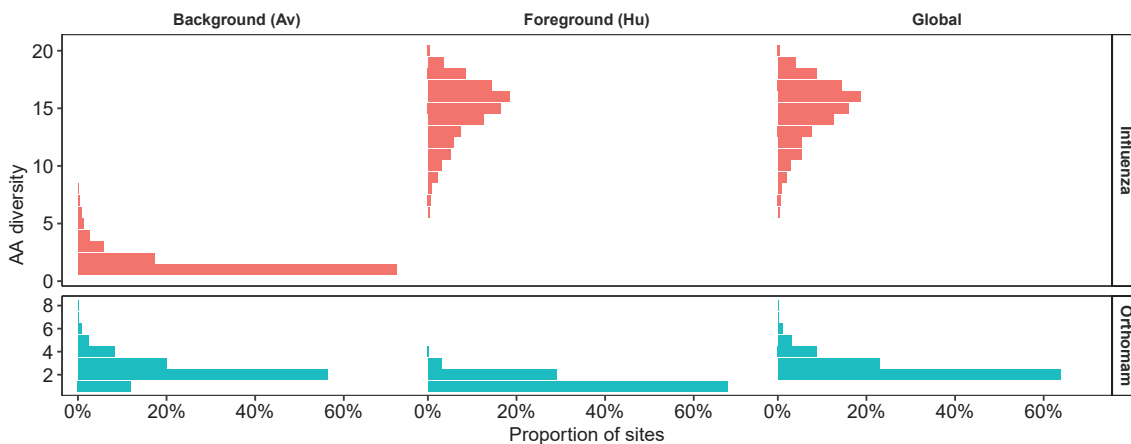


Figure 4.12: Amino acid diversities observed across sites simulated under differential selection along the Influenza and Orthomam phylogenies.

Under the homogeneous null model, the sparse representation involves a stationary distribution π^{H_0} , with frequencies for all amino acids observed at one site: that is the union of the sets of amino

¹³The avian (background) sub-tree is quite small and may not be large enough to give a clear signal for accurate estimation of the stationary distribution. However, it should be sufficient to detect differences between the background and foreground amino acid frequencies.

acids observed at each host condition. Since the condition dependent model has to encapsulate the null model so that hypothesis testing can be performed using Wilks' theorem, the dimensions of both condition-specific stationary distribution π^{Hu} and π^{Av} have to be equal to that of π^{H_0} .

As a result, in the avian host sub-tree where only a small subset of amino acids is tolerated, it is thus likely that the substitution model is over-parameterized: most frequency parameters in the stationary distribution do not contribute to better fit the model to the data. We are coerced into fitting a model that is too rich in parameters (15 on average as seen figure 4.12) on the ancestral sub-tree, where a simpler model would be sufficient to describe the process happening in this sub-tree (2 equilibrium frequencies on average).

The unpleasant conclusion of this, is that when the likelihood ratio test is conducted, an overly large number of degrees of freedom is used to shape the null chi-square distribution, resulting in a drastic loss of power in the test and a large number of false negatives.

4.5.2 A better approximation of the null distribution of log-ratios

The problem seems to be that the null distribution of log-ratios as established by Wilks' theorem generally does not match the actual distribution of log-ratios when we compare the two alternative models in Pelican. In the general case it is close enough, but the Influenza dataset magnifies this divergence from the asymptotic theoretical distribution. Searching for a better approximation of the null distribution of likelihood ratios, we attempted to mitigate the performance reduction by altering the calculation of the number of degrees of freedom in the null χ^2 distribution.

This simply consists in using the amino acid diversity observed in each condition as a measure for the number of *effective* parameters in the alternative model. The number of degrees of freedom is then obtained by calculating the difference between the sum of condition-specific diversities, and the global diversity observed at the site¹⁴.

$$df = div(fg) + div(bg) - div(total) \quad (4.30)$$

By using this heuristic, we drift further away from Wilks' theorem, but that appears to solve the issue we had with the influenza dataset. Figure 4.13 compares the detection performance measured with Pelican, when using the standard definition of degrees of freedom or using the heuristic. Evaluation was performed in the context of relaxed selection, and differential selection ("shift"), to assess how the heuristic performs in general. Precision-recall is greatly improved on the Influenza dataset in the context of relaxation, but also when simulating directional selection. Performance remains quite similar when switching to the heuristic on the other datasets; it is slightly improved on the HIV dataset, and slightly decreased on the Echolocation dataset.

I have described in 4.4.2 how some amino acids are sometimes not observed at a site just because the effective sample size is low: a small number of tips in the tree coupled with short branches. This gives a plausible explanation for the slight performance drops induced by the use of the heuristic. The alternative calculation that I describe gives, on average, a lower number of degrees of freedom than the initial specification. As a consequence, the number of degrees of freedom calculated for datasets with a small effective sample size is even more depleted when using the heuristic, which would result in a further increased false positive rate.

¹⁴This is equivalent to measuring the size of the intersection of the sets of amino acids in the background and foreground conditions. The idea is that each amino acid that is observed in both conditions should involve an additional frequency parameter in the alternative model.

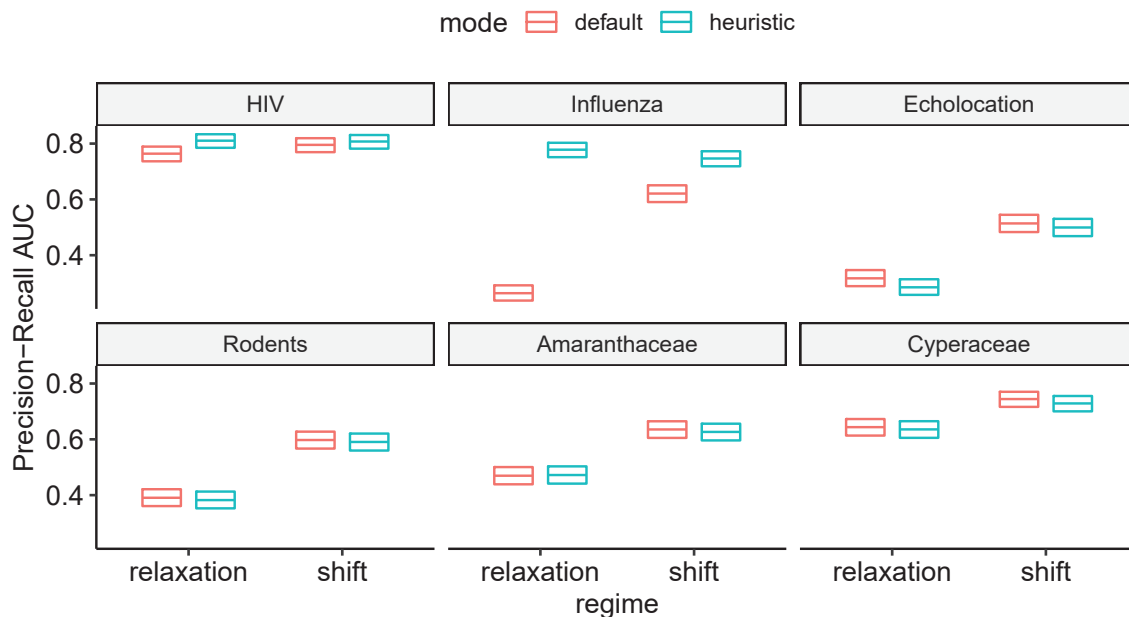


Figure 4.13: Calculation of degrees of freedom using the heuristic restores good performance on the Influenza dataset.

4.5.3 Adapting the model to distinguish neutral from purifying selection

The root cause of the problem that we attempted to mitigate by tinkering with the degrees of freedom to come up with a heuristic, is actually that our model is not best suited for detecting relaxation. Indeed, under the alternative hypothesis where one condition is relaxed, we nonetheless estimate equilibrium frequencies for each observed amino acid at the site. In that regard, relaxation is modelled no differently than other selective regimes: we just expect to infer a flatter profile at the relaxed condition.

In fact, we do not need so many parameters to detect relaxation, and we have shown a situation where it is actually harmful, with the example of the Influenza dataset. Actually we expect that the stationary distribution under the relaxed condition may look like a flat distribution where all frequencies are equal, so that we do not need to infer each of its frequencies at maximum likelihood.

We can adapt Pelican’s models for detection of selective pressure shifts in general to specifically detect shifts between neutral selection and purifying selection. Ideally, we would like to test the null hypothesis H_0 : “all conditions share the same stationary distribution” versus the alternative hypothesis H_A : “each condition has a specific stationary distribution and the stationary distribution of the relaxed condition is flat”. Unfortunately, this is not feasible in our framework, because of the constraint that the model for H_0 has to be nested within the model for H_A in order to perform likelihood ratio tests. The specification we propose does not enable us to observe this constraint: we can not express the H_0 model as a particular case of H_A , because there are no free parameters controlling the equilibrium frequencies in the relaxed condition.

We are then forced to test another null hypothesis, that we define as H_0 : “all conditions are under neutral selection, i.e. have a flat stationary distribution”. Using the same definition for H_A ,

we design the following two nested models

$$\begin{aligned}
 \text{reduced} : \beta_i^k &= 0, \quad \forall i \in U \\
 \text{full} : \begin{cases} \beta_i^k = 0 & \forall i \in U \setminus V \\ \beta_i^k \text{ free} & \forall i \in V \end{cases} \\
 \pi_i^k &= \frac{\exp(\beta_i^k)}{\sum_i \exp(\beta_i^k)}
 \end{aligned} \tag{4.31}$$

where k denotes a phenotypic condition in the tree, U the set of amino acids observed at the site, and V the set of amino acids observed across non-relaxed conditions. Both models include a parameter σ controlling the substitution rate, in an analogous way to the original model.

The nesting of the two models is apparent, as the reduced model can be expressed as the full model where all β_i^k parameters are fixed to be null. Stationary distributions π^k are derived from each β^k vectors, using a softmax function, which guarantees that π^k sums to one, while giving the necessary leeway during the inference of β_i parameters to adjust the relative importance of each equilibrium frequency.

This model does not accurately describe relaxed selection, but somewhat does the reverse by assuming that the site is in a neutral selection regime as a null hypothesis. Despite this relative inadequacy, we implemented and evaluated this model on simulations conducted under relaxed selection, to find out whether it would improve the detection performance on the Influenza dataset.

Parameter sparsity

Note that the parameter specification is also sparse in this model: the state alphabet is restricted to the observed amino acids at the site. In a dense version of this model, the stationary distribution under H_0 would have equilibrium frequencies equal $1/20$. In the sparse representation, the frequency profile under relaxation has values equal $1/|U|$, where U is the set of observed amino acids at the site. Focusing on the definition of the H_A model, we have a second layer of parameter sparsity. β parameters are only inferred for amino acids in V , the set of observed amino acid in all phenotypic conditions, excluding the relaxed condition. The rationale for this is that amino acids only observed in the relaxed condition are the outcome of the neutral substitution process, and their stationary frequency is likely negligible in the other conditions. As a matter of practicality, we have to ignore amino acids specific to the relaxed condition: otherwise the issues caused by over-parameterization would manifest in this model, as they did in the original profile shift model.

Degrees of freedom

Under Wilks' theorem, the number of degrees of freedom is the number of additional free parameters in the nesting model. In the present situation, the null model has only one free parameter, the scale σ . The alternative model introduced one additional parameter for each amino acid in V . Consequently, the number of degrees of freedom is calculated as the cardinality of V , the set of observed amino acids outside of the relaxed condition. There is an exception when $U = V$, i.e. when all amino acids in the tree are found in the non-relaxed conditions. In that situation, one parameter β_i can be set to 0, because the softmax function has the same output when every coordinate in its argument is translated by the same value. As a result, one of the β_i parameter is null and does not need to be

inferred.

$$df = \begin{cases} |V| - 1 & \text{if } |U| = |V| \\ |V| & \text{otherwise} \end{cases} \quad (4.32)$$

4.5.4 Performances obtained with the Pelican variants

Let us wrap this up with a comparison between Pelican and its two variants that I presented above, when attempting to detect sites under relaxed selection. As a reference, `codeml` is also included in the comparison. Figure 4.14 shows that good predictions are achieved on the Influenza dataset under relaxed selection when using either of the two variants of Pelican, which was the initial motivation for these developments. They also both have equal or better precision-recall when compared to `codeml`.

However, it is not clear which of the two variants could be preferable to the other when searching for sites under relaxed selection: indeed, neither of them is always better than the other independently of the dataset. Moreover, their performance is sometimes even decreased when compared to that of the original model of Pelican, and of `codeml`. For these reasons, these variants of the model are not currently included in the main implementation of Pelican. Their use is for now limited to highlighting and understanding pitfalls of the method when attempting to detect relaxed selection, as we discussed in the present section.

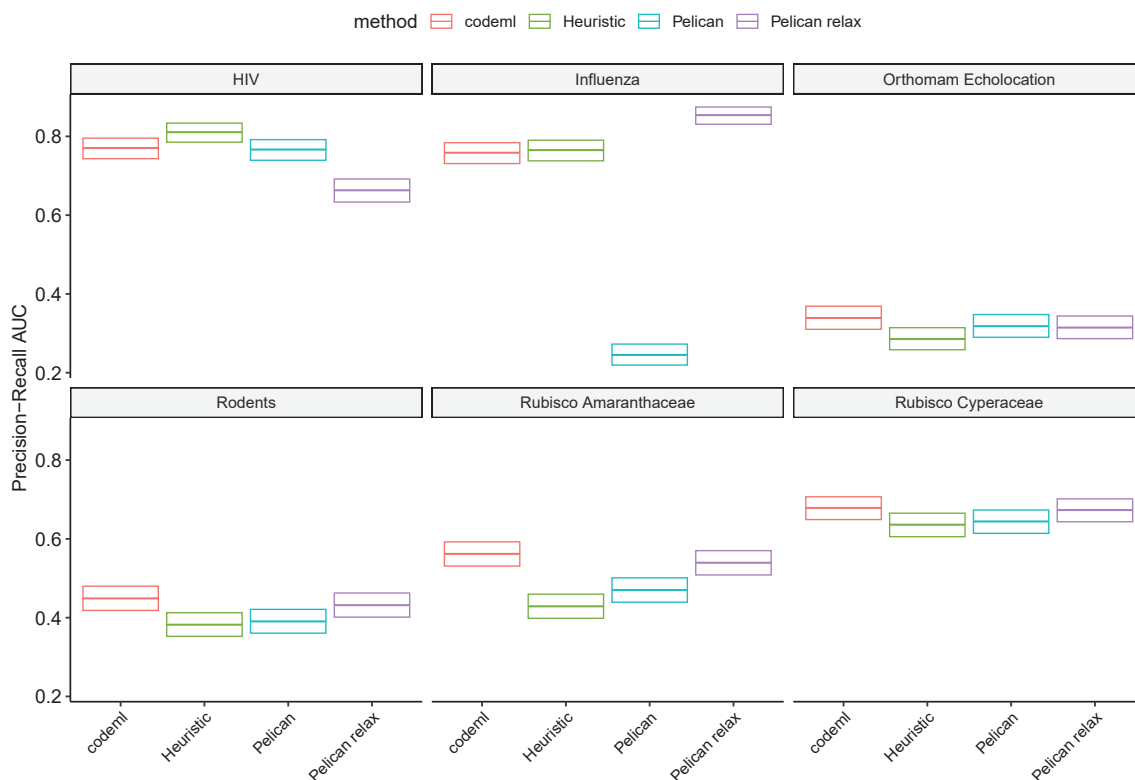


Figure 4.14: Performance in the detection of sites under relaxed selection, measured as precision-recall. Using the heuristic or a specific model of relaxation for Pelican restores the quality of prediction in the Influenza dataset.

4.6 Filtering sites using Multinomial, a fast non-phylogenetic method

In the previous chapters, I have described and evaluated the Multinomial model in several contexts. It is a non-phylogenetic method, which is faster than phylogenetic methods that we have investigated so far by orders of magnitude when analysing sequence alignments. It is of course less reliable, but shows pretty good performance considering its simplicity.

It is expected that only a few sites within an alignment are associated to a phenotype. That implies that when performing analyses with Pelican a large amount of computations is spent on negative sites, and possibly that some fraction of them is very obviously not associated to the phenotype — this includes, but is not limited to constant sites. Sorting out which sites are worth investigating using a somewhat computationally costly phylogenetic model such as Pelican, seems like a good use-case for a method such as Multinomial. It is extremely fast, and can easily be applied on genome-scale dataset in a negligible time to act as a filter and identify a pool of candidate sites. The stringency of the filter can be controlled simply by choosing a threshold on the p -values resulting from the filtering phase.

This strategy is implemented in the Pelican software, and greatly increases the throughput when it is employed, by reducing the number of sites that have to be run through the phylogenetic model. This is however not a perfect strategy, because more stringent filtering is more prone to eliminate sites that would have been potential good candidates for association to the phenotype. Figure 4.15 illustrates this trade-off for different filter thresholds, using measures obtained from the scan of the Orthomam database for sites associated with echolocation.

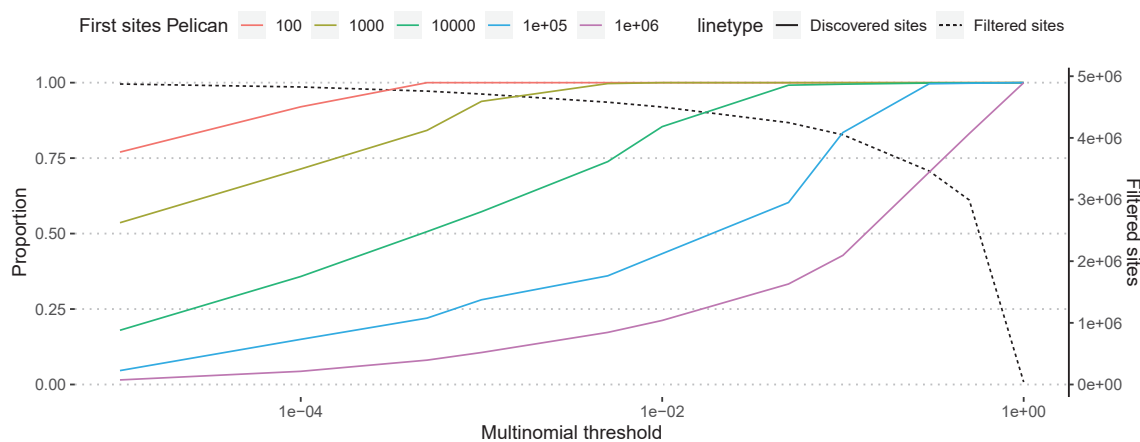


Figure 4.15: Proportion of best ranking sites retrieved (colored lines) and quantity of sites filtered globally (dashed lines) depending on the multinomial threshold used in the first pass filtering. These results were computed on empirical alignments from the Orthomam database. Each colored line corresponds to a quantity of best ranking sites, a proportion of which will be filtered out when using the multinomial filter. For example, using a threshold equal 0.1, about 80% of the top 10 000 sites in Orthomam (blue line) pass the filter to be analyzed by Pelican; about 80% ($\approx 4M$) of the Orthomam sites overall are discarded after filtering (dashed line).

In the current implementation, a scan using Multinomial is systematically performed, but no filtering of sites is done except for constant sites which are assigned a p -value equal 1. The filtering strategy can be tuned using the optional command-line flag `--multinomial-filter [thr]`, where `thr` is an optional value for the filter threshold, which is equal to 0.1 by default.

4.7 An alternative approach for fitting the model using automatic differentiation

When analysing alignments using Pelican, the overwhelming majority of the time is spent evaluating the likelihood function at variable points in the parameter space of the model, in order to estimate the parameter values that maximize the likelihood — in brief, fitting the model to the data is the most computationally expensive task. Because alignment sites are modelled independently of each other, the problem is also embarrassingly parallel¹⁵: a typical gene contains hundreds of sites, and genome scale alignments may contain millions. The current software implementation of Pelican features parallel computation between genes, which helps increase the throughput of the method when working with multiple alignments of gene families. However, this only leverages a small fraction of the parallel nature of the problem, since ideally all sites should be treated in a parallel fashion. This is easier said than done, simply because most computers do not dispose of such a large number of CPUs.

Graphical Processing Units (GPU) are specialized hardware pieces that are specifically designed to handle these “embarrassingly parallel” problems, such as 3D graphics rendering which is their most common application. The recent advances in applications of deep-learning have leveraged GPU computation to deal with the massive data inputs and models that are involved. In this context, the problem under consideration is generally the optimization of an objective function, which is generally a measure of prediction errors that must be minimized. The task of fitting this kind of model is generally done using automatic differentiation and gradient descent.

The purpose of automatic differentiation is to compute the gradient of an objective function to be optimized, along each dimension in the parameter space. This is done by calculating the partial derivatives of the function with regard to each parameter in the model — which is hard to determine analytically for complicated models. In practice, it consists in evaluating the partial derivatives of a function at one point in the parameter space, by breaking it down as a computational graph which is a structured representation of all of its internal elementary operations.

The result is a gradient along all parameter dimensions, which gives a linear approximation of the derivative of the function, and can simply be “followed”: parameter values are updated by adding to them a fraction of their respective gradient, in a process that is called gradient descent. These steps can be repeated until the objective function stabilizes, hopefully indicating that parameters have reached an optimum.

Several libraries have been made available that implement automatic differentiation and leverage GPUs to perform high-dimensional computations. We use PyTorch [Paszke et al., 2019], a Python library that provides data structures and bindings to efficient C implementations of a large variety of operations, that can be executed either on CPU or GPU. In our problem, the objective function is the likelihood of the alignment under a phylogenetic model, that we aim to maximize by adjusting the parameters of the model. We made a differentiable implementation of Felsenstein’s algorithm that computes this likelihood function using PyTorch’s API, and leverages parallel computation by vectorizing operations across sites.

4.7.1 Vectorized implementation of Felsenstein algorithm

The primary data structure that we work with is the *tensor*, which is similar to an n -dimensional array. Parameters of the model, inputs, and all intermediary variables are represented using tensors,

¹⁵https://en.wikipedia.org/wiki/Embarrassingly_parallel

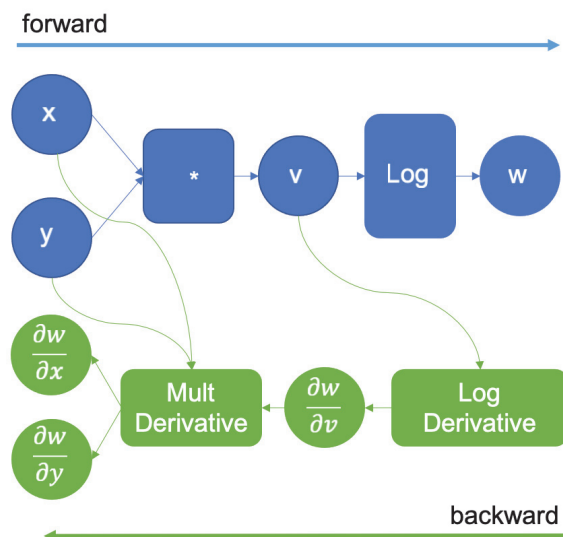


Figure 4.16: Illustration of the computational graph (in blue) representing the function $f(x, y) = \log(x \times y)$, with the corresponding reverse-mode automatic differentiation (in green). In the forward phase, the function is calculated progressively by following the computation graph, and the input of each elementary operation is stored, to be used afterwards during the backward phase which computes the gradient associated to each operation. This figure was extracted from the blog post “Overview of PyTorch Autograd Engine” at PyTorch’s official website.

where the first dimension ranges across sites. For example, the scale parameter σ in the model is a scalar for each site, which gives a vector of n parameters in the vectorized representation for n sites. Similarly, all stationary frequency parameters π_i at each site i are gathered in a single 2-dimensional tensor. However, this forces all sites to be described by models having the same dimension: the sparse representation that I described in section 4.3.4 can not be fully implemented because all elements in a tensor must have the same dimension. However, we can enforce some kind of sparsity by constraining frequency parameters of unobserved states to be close to 0, as shown in figure 4.17, as well as reduce the state space to have only the minimum size required to match the most diverse site (4 in this case). As was discussed earlier, stationary frequencies can not be set equal to 0, because that would prevent the computation of the matrix of transition probabilities. Each sequence is also represented as a tensor, with 2 dimensions: it encodes as a one-hot vector on the second dimension the observed state at each site on the first dimension.

Transition rate matrices are computed by taking the subset of the exchangeability matrix (e.g. the WAG matrix) corresponding to the state space at each site, and padding it to obtain the same dimension across all matrices. These pseudo-exchangeability matrices are then combined with scales and stationary frequencies, in a vectorized way, to obtain the transition rate matrices that define the Markov model at each site. In turn, eigen-decomposition is performed on each rate matrix, enabling to compute its scaled exponential, i.e. a transition probability matrix for each site. As soon as the transition probability matrix is available, Felsenstein’s tree pruning algorithm can be used to compute the log-likelihood at each site, using an adequate implementation based on operations on matrices and vectors.

Likelihood computation for large trees and underflow errors

As I mentioned in section 4.3.1, the direct computation of likelihoods using Felsenstein’s pruning algorithm is prone to numerical errors, due to the limited memory allocated to represented floating

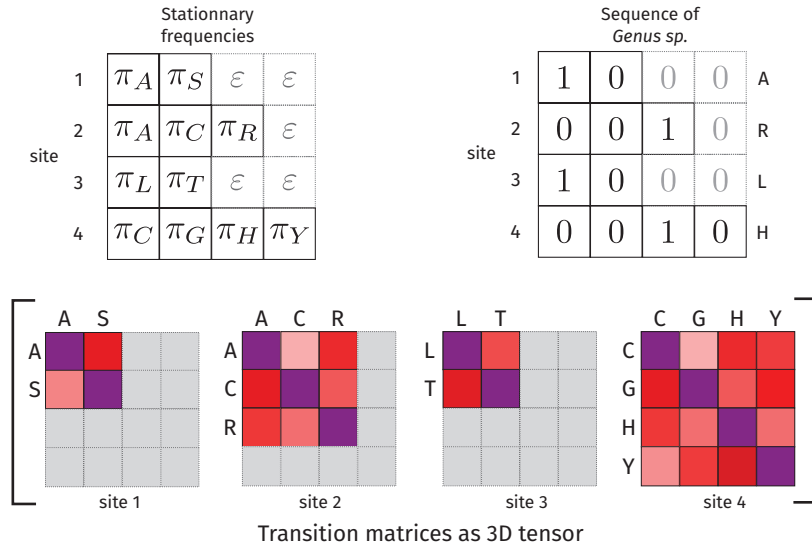


Figure 4.17: Pseudo-sparse representation of parameters, sequences, and transition matrices as PyTorch tensors for a dataset containing 4 sites which respectively display 2, 3, 2, and 4 distinct amino acids. Top left is the tensor containing the 4 stationary frequency profiles, one line for each site. Top right is a one-hot encoding of one sequence (ARLH), where the observed amino acid is assigned probability 1 and the others 0. Each sequence in the alignment is encoded in the same way. On the bottom are the corresponding transition rate matrices for each of the 4 sites, gathered in a tridimensional tensor. All matrices are padded to have the same dimensions, using a small value ε for stationary profiles as null frequencies are not allowed in our framework, or zeros in the case of sequences. The amino acid alphabet may differ between sites, but is kept consistent among all tensors at a given site: e.g. columns at the first site always correspond to amino acids A, then S.

point numbers. In the current implementation of Pelican, this issue is tackled by scaling the likelihoods whenever they are too low, as I explained as well in the section I am referencing to. However, this strategy can not be applied in the context of automatic differentiation, because it is not differentiable. We then resort to another approach classically used to overcome the issue of underflow error, which consists in working with log likelihoods, so that product operations are turned into sums, and the numbers that we manipulate can be accurately represented in memory.

The expression of the log likelihood at a node can easily be derived from equation 4.9

$$\begin{aligned}
 \log L_n(x) &= \log \prod_{c \in \mathcal{C}(n)} \sum_{y \in k} \mathbb{P}[c = y | n = x, t_{nc}] L_c(y) \\
 &= \sum_{c \in \mathcal{C}(n)} \log \sum_{y \in k} \mathbb{P}[c = y | n = x, t_{nc}] L_c(y)
 \end{aligned} \tag{4.33}$$

At this point, we are still stuck with a sum of product of probabilities on the right hand term. Another trick we can use is to introduce an exp log operation in the sum over states

$$\begin{aligned}
 \log L_n(x) &= \sum_{c \in \mathcal{C}(n)} \log \sum_{y \in k} \exp \log (\mathbb{P}[c = y | n = x, t_{nc}] L_c(y)) \\
 &= \sum_{c \in \mathcal{C}(n)} \log \sum_{y \in k} \exp (\log \mathbb{P}[c = y | n = x, t_{nc}] + \log L_c(y))
 \end{aligned} \tag{4.34}$$

Now let us define $\alpha_y(x) = \log \mathbb{P}[c = y | n = x, t_{nc}] + \log L_c(y)$, so that

$$\log L_n(x) = \sum_{c \in \mathcal{C}(n)} \log \sum_{y \in k} \exp \alpha_y(x) \quad (4.35)$$

Looking at the definition of $\alpha_y(x)$, we note that it is a sum of log probabilities, and would be typically expected to be a large negative number. As a consequence, $\exp \alpha_y(x)$ can still cause underflow issues. To address this, let us define $a = \max_y \alpha_y(x)$, so that:

$$\begin{aligned} \log L_n(x) &= \sum_{c \in \mathcal{C}(n)} \log \sum_{y \in k} \exp(\alpha_y(x) - a + a) \\ &= \sum_{c \in \mathcal{C}(n)} \log \sum_{y \in k} \exp(\alpha_y(x) - a) \exp a \\ &= \sum_{c \in \mathcal{C}(n)} \log \left(\exp a \sum_{y \in k} \exp(\alpha_y(x) - a) \right) \\ &= \sum_{c \in \mathcal{C}(n)} \left(a + \log \sum_{y \in k} \exp(\alpha_y(x) - a) \right) \end{aligned} \quad (4.36)$$

The term in the exponential is now the difference between one value $\alpha_y(x)$ and a , the maximum value in the $\alpha(x)$ vector. The magnitude of this number is thus reduced in comparison to $\alpha_y(x)$, and underflow issues are less likely to happen when computing the exponential term. Nevertheless, the procedure can still be somewhat prone to underflow errors when the value of one $\alpha_y(x)$ is a lot smaller than the maximum a . This does not hurt too much though, since the resulting approximation error is then considered negligible with regard to the result of the sum over all y states.

The same trick can be applied to the calculation of the global likelihood across the tree, upon reaching the root node.

$$\begin{aligned} \log L &= \log \sum_{x \in k} \pi_x L_R(x) \\ &= \log \sum_{x \in k} \exp \log(\pi_x L_R(x)) \\ &= \log \sum_{x \in k} \exp(\log \pi_x + \log L_R(x)) \\ &= b + \log \sum_{x \in k} \exp(\beta_x - b) \quad \text{where } \begin{cases} \beta_x = \log \pi_x + \log L_R(x) \\ b = \max_x \beta_x \end{cases} \end{aligned} \quad (4.37)$$

This log-sum-exp formulation is commonly used for computations involving products of probabilities. In our case, the resulting definition of the log likelihood function only consists in sums of numbers having a magnitude that can be represented as floating point numbers without important loss of precision in the result.

4.7.2 Optimisation of the parameters using L-BFGS

The output of our implementation of the pruning algorithm is a vector of log-likelihoods, one for each site. Since we aim to maximize each of the individual log-likelihoods, and because sites are independent in our model, we can simply maximise the sum of log-likelihoods across sites. In fact, we actually use the mean log-likelihood across sites as a cost function — the sum, but divided by

the number of sites — because it appears to work better with the optimization algorithm. This cost function is minimized using the L-BFGS algorithm [Liu and Nocedal, 1989], which is a quasi-Newton method that leverages the knowledge of the gradient at several points of the objective function to find a local optimum.

Initial parameters of the reduced model are set to a scale equal 1, and stationary frequencies equal to the observed frequencies at each site. As for the full model, initial scale is also set to 1, and we choose at each site which stationary distribution gives the best log-likelihood, between (1) using at both conditions the one estimated after fitting the reduced model (2) using the observed frequencies at each site, after grouping amino acids by condition. This strategy hopefully gives a better starting point for the estimation, and may help prevent getting stuck in local optima. During the optimisation, parameters are represented in log scale, and are bounded in the same intervals as in the reference implementation of Pelican.

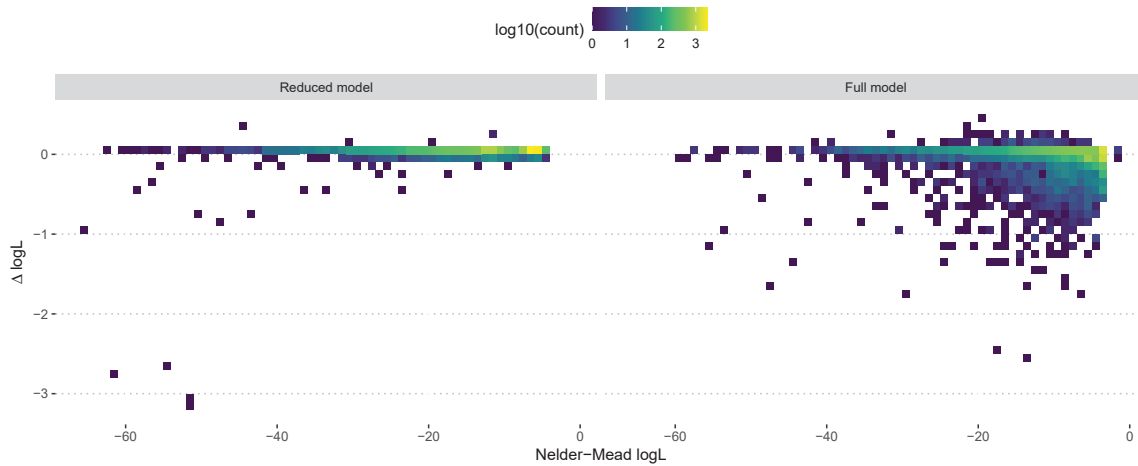
After the full model is fitted, the resulting maximum log-likelihoods are compared to those from the reduced model, so that sites where the full log-likelihood is lower than the reduced log-likelihood can be identified. Indeed such situations are symptoms of optimisation gone wrong, as the full model should be able to have at least equal likelihood with the reduced model. A “catch-up” optimisation round is done on these sites, which are all reset to have stationary distributions as estimated by the corresponding reduced models. This does help improve the fit of the sites identified in this manner.

4.7.3 Results

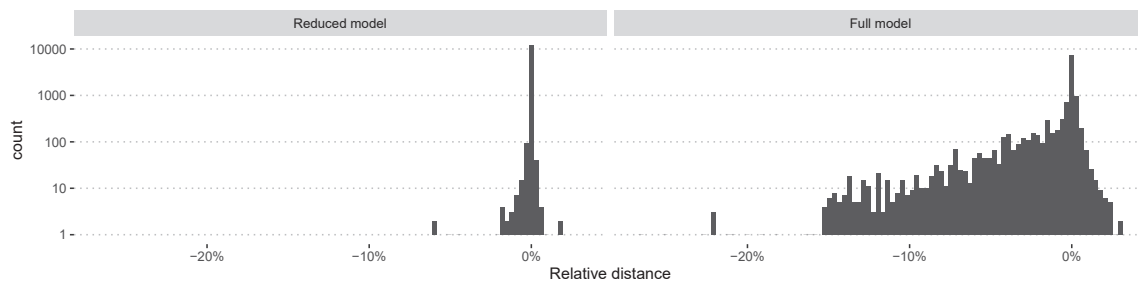
In order to evaluate the quality of the fit, I compare the maximum log-likelihood that were attained using the reference implementation based on the Nelder-Mead algorithm, and the alternative one that I described. To do so, an alignment of 20 000 sites was simulated using the Orthomam phylogeny annotated with the echolocation phenotype, among which 12 050 sites were found to be non-constant, i.e. had at least two distinct amino acid states. The full and reduced models of Pelican were fitted on these sites using both implementations, so that the maximum likelihood and corresponding p -values obtained for each of them could be compared.

Figure 4.18a shows that on most sites, the maximum log-likelihood of the reduced model matches closely between the two implementations, except for a few sites where the optimisation seems to have got stuck in a local optimum. The agreement is not as good when it comes to the full model, where the fit of the alternative implementation is worse on a large fraction of sites. Even though the absolute difference in the log-likelihood is quite small, the relative difference is sometimes more noticeable depending on the composition of the site, as highlighted in panel 4.18b. Because of the degraded fit on some sites compared to the Nelder-Mead procedure, the resulting p -values are not in perfect agreement between the two methods, as seen figure 4.18c. In most cases, it is the full model that is not as well adjusted, resulting in a bias towards higher p -values when using automatic differentiation and L-BFGS to fit the model. The optimisation also gets stuck on non-optimal parameters when fitting the reduced model, which results in lower p -values on a few sites compared to the Nelder-Mead estimation. The comparison of p -values in a log scale on the right panel of figure 4.18c highlights that the agreement between the two optimisation methods seems quite good when it comes to p -values of a lower order of magnitude. This means that both methods are able to correctly identify positions where there is a strong signal for genotype-phenotype association.

In terms of speed, the PyTorch implementation effectively leverages the parallel computing capabilities of the GPU, and achieves a complete scan of the alignment in 2m51s where the reference implementation does so in 17m35 when executed sequentially on one CPU. Profiling of the imple-



(a) Absolute difference in maximum log-likelihood using gradient descent compared to Nelder-Mead.



(b) Relative difference in maximum log-likelihood using gradient descent compared to Nelder-Mead.

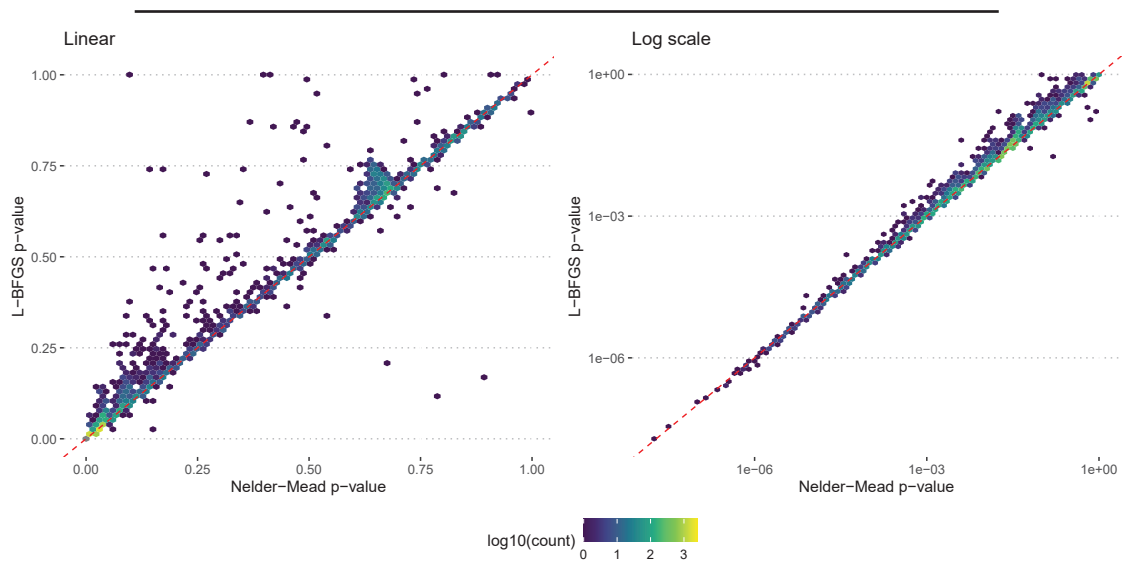
(c) Comparison of p -values obtained after fitting the model using automatic differentiation and L-BFGS or using the reference implementation based on the Nelder-Mead algorithm. Left panel displays the p -values in a linear scale; right panel uses a log scale to give a better appreciation of the agreement on small p -values.

Figure 4.18: The maximum log-likelihood attained using automatic differentiation and gradient descent tends to be lower than when using the Nelder-Mead optimizer (*a* and *b*). As a consequence, p -values computed from the LRT are not always in good agreement between the two optimisation approaches (*c*), and tend to be higher when fitting the model using L-BFGS.

mentation was conducted to detect and remove bottlenecks caused by unnecessary data transfers between CPU and GPU — a very common issue in this context, leading to the under-exploitation of the GPU — until it was clear that all necessary inputs and parameters were transferred on the GPU once at the start of the program.

4.7.4 Discussion

The difficulty to consistently achieve a fit of the model at least as good as when using Nelder-Mead to optimize the parameters might be tied to at least two reasons. First, it might be that using a gradient descent does not allow enough freedom to explore the likelihood landscape, and the algorithm could get more easily stuck in local optima compared to the Nelder-Mead approach. Second, I suspect that the pseudo-sparse implementation that is illustrated in figure 4.17 could deteriorate the quality of the fit. As I discussed in section 4.3.4, fitting Pelican’s model with a dense parameterization where unobserved amino acids are accounted for is more difficult, and parameter estimates are less stable even when using the Nelder-Mead optimizer. Because we are currently forced to maintain the same state space dimension across all sites, so that parameters of the model and all intermediary calculations in the pruning algorithm can be respectively gathered in a tensor, we can not fit a model that is “truly” sparse for each site. Instead, the dimension of the model at every site is that of the highest dimension, which is determined by the maximum amino acid diversity across all sites. Even though we mitigate this problem by fixing the value of the stationary frequencies for unobserved amino acids at a small value ε , this might still have an effect on the estimation of the other parameters.

This constraint brings another issue, which is more of a technical one. In the current pseudo-sparse implementation, we instantiate tensors that are larger than they need be, as an important fraction of their content ties to modeling substitutions involving unobserved amino acids and is thus not relevant to us. Now, the computation of the gradient using automatic differentiation does require that the input of each elementary operation in the computation graph is kept in memory, so that the corresponding local gradient can be computed in the backward pass (as shown figure 4.16). As a consequence, the memory footprint of this implementation of the pruning algorithm quickly becomes difficult to manage: it grows rapidly with the number of nodes in the tree, the maximum amino acid diversity observed across sites, and of course the number of sites¹⁶. A truly sparse implementation could help to reduce drastically the memory requirements for the execution of the pruning algorithm, as most sites have a lower diversity than the maximum in the alignment. This problem could also be addressed by distributed computing on several GPUs, but this kind of equipment is not easily available.

The PyTorch community is currently working on a feature that would precisely allow the representation of sparse tensors, which they call “nested tensors”¹⁷, and that I believe would significantly improve the viability of this implementation. For now, PyTorch’s nested tensors are in a very early stage of development, and support a set of operations that is too restrictive to implement Felsenstein’s algorithm.


I expect that the speed improvement using this implementation would scale with the number of sites in the dataset. In the setting from which the above results were obtained, the throughput

¹⁶As an example, the maximum amino acid diversity was 8 in the alignment on which the experiment for which results are presented figure 4.18. This sets the dimension of the tensor containing parameters for the stationary frequencies to $12\,050 \times 8$. The GPU memory required to perform the pruning algorithm in this setting is about 5 GB, which makes fitting larger alignments impracticable on a single GPU.

¹⁷See <https://pytorch.org/docs/stable/nested.html>

was increased by a factor about equal 6.2. This is encouraging, but must be put in perspective with the fact that this is a comparison to a sequential execution of the current implementation of Pelican, that does have the capability to run in parallel across alignments. Moreover, because of the memory limitations that were discussed, increasing the number of sites to process in parallel would be difficult as of yet, while exploiting the parallelism of the current implementation is rather easy: computing facilities generally have a large number of CPUs available, and the CPU memory footprint of Pelican is very manageable.

All in all, I consider this work to be a proof of concept, more than a credible contender to the current implementation. Given more time, it could still be interesting to evaluate a sparse version when the relevant features are implemented in PyTorch. Another possible direction to explore would be to let down automatic differentiation and gradient descent, which are the main source of the memory issues that I exposed. This would require making an implementation of another optimisation algorithm, e.g. Nelder-Mead, in a manner that would be suitable to run in parallel on the GPU. The BEAGLE library [Ayres et al., 2012] is dedicated to performing computations commonly involved in phylogenetic models, and might provide useful tools for this task. Finally, a possible application for such parallel approaches could be fitting models that incorporate parameters shared across sites: for example, a general bias towards a restricted set of amino acids among hyperthermophile species [Zeldovich et al., 2005], the estimation of which could benefit from a joint estimation between all sites.

 Chapter 4 summary: *Pelican: a fast phylogenetic method to identify selective pressure changes*

This chapter gives more details on the **model, implementation, features and limitations of Pelican**, which was introduced in the previous chapter. In a sense, it is complementary to the previous chapter: after we have established that the approach gives a good answer to our problem, it goes deeper into methodological aspects to better understand its inner workings.

After I explain the original motivation for developing this method as an implementation of the previously published model TDG09, I detail the model itself. It is in fact a pair of **nested models** of amino acid substitution: a **reduced model** where the substitution process is homogeneous across all branches in the tree, and a **full model** where the substitution process depends on an annotation of phenotypic traits on each branch. They are fitted at maximum-likelihood using Nelder-Mead's algorithm and compared using a likelihood ratio test. Some modeling choices are discussed as well, in particular the **reduction of the state space** in the Markov process underlying the substitution model.

Looking back on a particular results in the previous chapter, that is the very **low performance** of Pelican when detecting sites under **relaxed selection** on one specific simulated dataset (Influenza), I explore the cause of these underwhelming results and propose two workarounds to restore good predictive capability for Pelican. One consists in adjusting the degrees of freedom in the LRT to account for the specificity of the dataset, the other is a variation of the model specifically designed to detect relaxed selection. Both approaches fulfill the intended goal, but are limited in terms of potential for generalization.

The end of this chapter focuses on improvements on the throughput of Pelican. Based on the good trade-off between reliability and speed that is offered by the **Multinomial** approach, we propose to use it as a **first-pass filter** with low stringency to eliminate sites that are not likely to be good candidates. Remaining sites that are worth investigating with a more accurate (and costly) method are then analyzed using Pelican. With similar goals in mind, I explored the possibility to leverage the **highly parallel** nature of the problem of fitting Pelican's model independently on each site within alignments. I made an **alternative implementation** of Felsenstein's algorithm based on the PyTorch library, and massively parallel computations on the GPU. This implementation computes the likelihood function associated to a point in the parameter space of the model at one site, as well as the gradient associated to each parameter using **automatic differentiation**, which can be followed to reach the nearest optimum of the function. I show the potential of this approach as a **proof-of-concept** to speed-up analyses conducted with the Pelican model, and possibly exploit simultaneous optimisation of sites to incorporate parameters shared across sites in the model; it is however limited for now by some missing features in PyTorch.

Chapter 5

Continuous trait associations

In Pelican’s implementation of the TDG09 model, we broaden the application scope of the original model by allowing an arbitrary number of phenotype conditions to be accounted for, instead of restricting analyses to binary traits. However, some phenotypic traits are continuous, and are not naturally represented as a discrete variable without loss of information and without making somewhat arbitrary choices in the discretization procedure. This is a motivation to propose a variation on Pelican’s model that would be adapted to work directly with continuous traits. I describe such a model in this chapter, that preserves the main formalism from the discrete model by modeling the substitution process as a GTR model, driven by stationary distributions of amino acid frequencies.

An important concern here is to keep the number of additional parameters low: we are working indeed with a limited quantity of information — a single site in an alignment — to estimate parameters, which is already challenging in the discrete case. Our answer consists in a model which estimates a stationary distribution for each extreme value of the phenotype, and interpolates amino acid frequencies between them for intermediary trait values, using a sigmoid function. This model thus introduces only two additional parameters, that determine the shape of the sigmoid interpolation: one slope parameter that controls the steepness of the sigmoid, and one so-called “shift” parameter that controls its point of inflexion¹. The underlying hypothesis is that, on sites associated to a continuous phenotype, amino acid preferences are correlated to the value of the trait. For example, one amino acid could be very disadvantageous for species exhibiting the lowest trait values, and strongly favored when the trait value is highest, and its frequency would continuously increase with the value of the trait between these two extremes under this model.

As it is not immediately obvious that this approach would improve the quality of the detection of sites associated to the phenotype, its performance is compared to the discrete model of Pelican using several discretization strategies. This benchmark is conducted on simulations, using an adaptation of the mutation-selection model that incorporates the sigmoid interpolation between amino acid fitnesses, and the Orthomam phylogeny annotated with the adult body mass phenotype. Several simulation settings are considered, where the steepness of the interpolation between fitnesses varies up to a point where the simulation is akin to a discrete case.

To further validate our approach in an empirical setting, we also compare our findings to published results [Farré et al., 2021] from a study of genomic factors tied to differences in longevity

¹A small web application that illustrates this model and the interaction between its parameters is online at <https://lsdch.shinyapps.io/continuous-model/>.
Code repository: <https://github.com/lsdch/pelican-sigmoid>.

among mammal species, that used a specific approach based on a discretization of this continuous trait.

Contents

5.1 Introduction	116
5.2 Methods	116
5.2.1 Models of sequence evolution	116
5.2.2 Model of codon sequence evolution	117
5.2.3 Model of protein sequence evolution	117
5.2.4 Sigmoid function used to link phenotypic value and amino acid preference	118
5.2.5 Sparse parameter estimation at each site	118
5.2.6 Likelihood ratio test in model C	118
5.2.7 Benchmark simulations	120
5.3 Results	122
5.3.1 Simulations	122
5.3.2 Longevity in mammals	123
5.4 Conclusion	126
5.5 Acknowledgements	126
5.6 Supplementary material	126
5.7 Additional remarks and discussion	126

Abstract

Comparative genomic data provides the ability to look for substitutions that are associated to particular phenotypic traits. A few methods have been proposed to deal with discrete phenotypic traits, but not with continuous traits. Here we extend the method Pelican, which has been shown to be powerful for detecting sites associated with discrete traits, to work with continuous traits. We call model C the new model, and the base discrete version model D. Model C links a continuous trait with amino acid preference at a site using a sigmoid function. We evaluate its performance on simulated data and compare it to model D, working on data discretized into two or three categories. We find that model C performs better than model D, even when the simulated sigmoid relationship between the variable and the sites is really stepwise. We apply model C to three empirical alignments and find that it identifies sites associated to variation in lifespan in mammals, with an enrichment in functional protein domains. Our results show that Pelican can now be used with both discrete and continuous phenotypes to search for sites associated with phenotypic variation. Pelican is available at <https://gitlab.in2p3.fr/phoogle/pelican>.

5.1 Introduction

Comparative genome data offers the possibility to associate substitutions to changes in the phenotypes of the species. This has motivated a lot of methodological development, and a lot of empirical studies. For example, many models have been developed to search for signs of adaptation in coding sequences (e.g., [Goldman and Yang, 1994, Yang, 2007]), using the d_N/d_S approach. These models can be used in a first step with a phylogeny annotated with species phenotypes to search for sites or genes that undergo different types of selection depending on the phenotype. In a second step, one can look at the amino acid substitutions at these sites to investigate if the likelihood of observing particular amino acids depends on the phenotype. This two-step process can be replaced by a single analysis using *profile* methods that directly identify an association between preferred amino acids and phenotypic state (e.g., [Tamuri et al., 2009, Parto and Lartillot, 2018, Duchemin et al., 2022]). We have shown that some of these methods, in particular the method Pelican, could be as powerful and faster than d_N/d_S methods [Duchemin et al., 2022].

So far profile methods have been used with discrete annotations of a phylogeny, whereby each branch is linked to a particular phenotypic state. Such annotations are ill-suited for continuous phenotypes, because they require discretizing a continuous distribution into categories. There are several ways to do so: for instance, to discretize into two categories, one could split at the mean or the median, or any other quantile. A priori, there is no reason to pick one method rather than another. To circumvent this difficulty, some researchers have relied on two steps. Firstly, putative sites where particular amino acids are associated to phenotypic values are identified by focusing on species with the most extreme phenotypic values. Secondly, species with less extreme phenotypic values are used to validate the sites identified in the first step. This approach has been used to identify sites and genes associated with increased longevity in primates and mammals [Muntané et al., 2018, Farré et al., 2021], but still depends on thresholds to identify “extreme” phenotypic values.

In this report, we introduce a continuous variant for Pelican, called model C, that can work naturally with continuous trait values. We compare its performances against the discrete variant where trait values are discretized in different ways, using simulations to generate protein sequence alignments under a mutation-selection model. Simulations are run along the Orthomam phylogeny including 116 mammalian species, using body mass values as an example for a continuous trait. We show that the continuous variant performs better than the discrete one, while exempting from the need to determine a discretization procedure for the trait values. We apply model C to three gene alignments that had been found to contain sites associated to variation in lifespan in mammals [Farré et al., 2021], using a two-step approach relying on a discretization of the lifespan data. We found that model C finds sites associated to lifespan in all three genes, including some that had been identified previously [Farré et al., 2021]. When comparing to results obtained after shuffling the lifespan data, we find that sites are enriched in functional domains of the proteins, suggesting that they are indeed related to lifespan.

5.2 Methods

5.2.1 Models of sequence evolution

We use a model of codon sequence evolution to simulate sequences, and a model of protein sequence evolution for inference. Both models are continuous time Markov processes running along

the branches of a phylogeny whose branch lengths are in units of expected number of substitutions [Felsenstein, 1981]. At each branch b , substitution rates are described by a substitution matrix Q_b , which, combined with a branch length l_b is used to compute a substitution probability matrix P_b as follows:

$$P_b = e^{Q_b l_b} \quad (5.1)$$

Individual sites are associated to specific amino acid preference parameters to account for heterogeneity in the process of evolution across sites. Different amino acid preferences may also be used across branches, if the site is assumed to be associated to the continuous phenotype under consideration. Such sites are called H_A sites, whereas H_0 sites are not associated to the continuous phenotype. As a result, for H_0 sites, $Q_b = Q$ for all branches b .

5.2.2 Model of codon sequence evolution

We use a mutation-selection model of codon sequence evolution [Halpern and Bruno, 1998, Yang and Nielsen, 2008, Rodrigue et al., 2010, Tamuri et al., 2012, Bloom, 2014] to simulate sequence evolution using the Gillespie algorithm [Gillespie, 1976]. Mutation-selection models handle mutations at the DNA level, and selection at the amino acid level. Both are combined into a 61×61 codon substitution matrix. A matrix of mutation probabilities μ between individual nucleotides is derived from exchangeabilities and equilibrium frequencies [Lanave et al., 1984, Tavaré et al., 1986]. Exchangeabilities are drawn from Gamma(1,1) distributions, and equilibrium frequencies from a Dirichlet(10,10,10,10) distribution, and are shared across sites. A 61×61 codon mutation matrix is built from this nucleotide mutation matrix by considering that double and triple substitutions between codons are not allowed.

Selection coefficients S_b^c (eq. 5.2) associated to the amino acids are defined as the difference in scaled fitness F between the ancestral amino acid X and the descendant amino acid Y , according to the parameters applying on the current branch b .

The relative fixation probability $u_b^c(S)$ for a mutation at site c and branch b is computed from the selection coefficient S as per [Kimura, 1983]:

$$S_{X \rightarrow Y, b}^c = F(X, c, b) - F(Y, c, b) \quad (5.2)$$

$$u_b^c(S) = \frac{S_{X \rightarrow Y, b}^c}{1 - e^{-S_{X \rightarrow Y, b}^c}} \quad (5.3)$$

Fitness values are drawn from a set of 263 preset frequency profiles [Rey et al., 2019] for each site. These frequency profiles are transformed into fitness profiles by multiplying them by a factor $Ne = 4$. As a result, values $S_{X \rightarrow Y, b}^c$ are between -4 and 4 . One profile needs to be drawn for H_0 sites, whereas two such profiles need to be drawn when simulating H_A sites. Further, in H_A sites, at each branch, vectors of amino acid fitnesses $u_b^c(S)$ are determined by a sigmoid function linking them to a continuous trait (see section 5.2.4).

Codon substitution rates Q_b are the product of mutation rates μ and the relative probability of fixation:

$$Q_{X \rightarrow Y, b}^c = \mu_{X \rightarrow Y} \times u_b^c(S) \quad (5.4)$$

5.2.3 Model of protein sequence evolution

For H_0 sites, the model used for inference in Pelican is similar to commonly used models of protein sequence evolution [Whelan and Goldman, 2001, Le and Gascuel, 2008]. In particular, it can be

expressed as a combination of a 20×20 exchangeability matrix and a vector of 20 equilibrium amino acid frequencies. However, our model has site-specific amino acid equilibrium frequencies. For H_A sites, it also has branch-specific amino acid equilibrium frequencies. These equilibrium frequencies are estimated in the maximum likelihood framework for model D, or are computed with the sigmoid function as explained below.

5.2.4 Sigmoid function used to link phenotypic value and amino acid preference

In H_A sites, we model amino acid preference at a site as depending on a continuous phenotypic trait. In a mutation-selection framework operating at the codon level, this results in different amino acid fitnesses depending on the phenotypic trait value. In a model of protein sequence evolution, this results in different amino acid frequencies depending on the phenotypic trait value. We use the mutation-selection framework for simulating sequences, and rely on the model of protein sequence evolution for inference inside Pelican.

We model amino acid preferences Y at a particular site as depending on the trait value t according to a logistic function $Y(t)$ (figure 5.1a). The logistic function depends on a left asymptote value of the amino acid preference (A), a right asymptote value (K), an inflexion point (t_0), and a slope (S) (Equation 5.5).

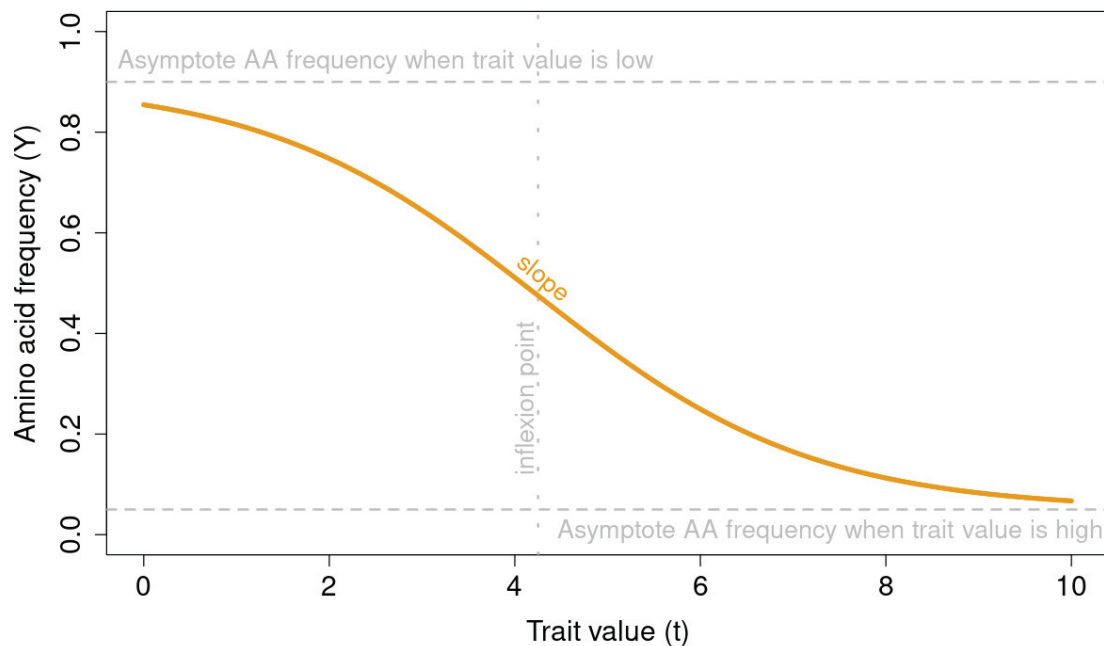
$$Y(t) = A + \frac{K - A}{1 + e^{-S(t-t_0)}} \quad (5.5)$$

Each amino acid preference is modelled independently from the other amino acid preferences, resulting in two (left and right) asymptote parameters per amino acid, *i.e.*, up to 40 parameters total. However, in the model of protein sequence evolution, only 19 amino acid frequencies are modelled, since the frequencies have to sum to 1.0, resulting in up to 38 parameters. In practice, we set the frequencies of unobserved amino acids to 0.0 as in [Tamuri et al., 2009, Duchemin et al., 2022], which strongly reduces the number of parameters (see section 5.2.5). The slope and inflexion point parameters that control the shape of the sigmoid are shared across amino acids, adding two parameters in the model. The 2 frequency parameters per amino acid combine with the slope and inflexion parameter per site to provide flexibility in the shape of the function. This is used to suit different trends in amino acid preferences as the phenotypic trait value varies, as shown figure 5.1b. For instance, the three first plots describe functions estimated for an amino acid that would be preferred for low values of the phenotype rather than for high values. However, the top right plot corresponds to a situation where an abrupt shift is observed around phenotypes of value 4, whereas the change in preference is much more gentle in the top left plot. In the bottom left plot, the change is still abrupt, but is now observed around phenotypes of value 7. Finally, the bottom right plot describes a situation where the amino acid is only slightly preferred for high values of the phenotype than for low values.

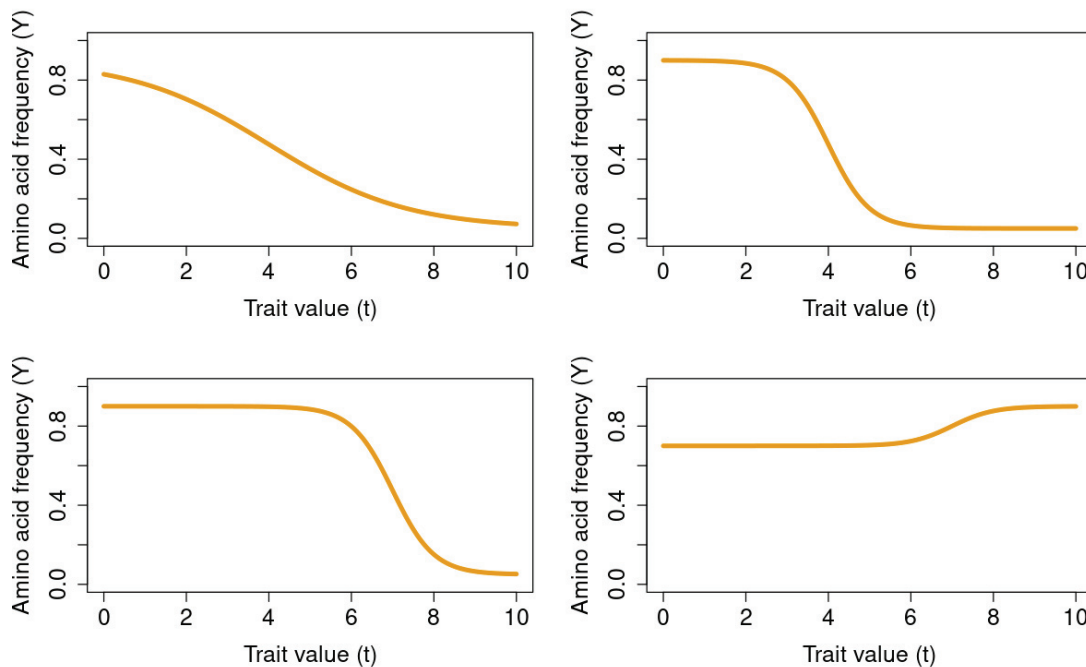
5.2.5 Sparse parameter estimation at each site

At each site, not all 20 amino acids are observed. We set the equilibrium frequency for an amino acid that is not observed at a site to 0. We use this to reduce the number of parameters to consider, by only working with the subset of amino acids that are observed at the site. Therefore, during estimation, fewer parameters need to be estimated per site. For instance, for a site that has only 5 different amino acids, model C includes only 11 parameters instead of 41. This technique reduces

Figure 5.1: Interpolation of amino acid frequency profiles along a continuous trait using a sigmoid function.



(a) Sigmoid function $Y(t)$ (orange) to describe the link between trait value and amino acid (AA) frequency. For low values of the trait, the amino acid frequency tends towards $A = 0.9$, and for high values of the trait towards $K = 0.05$. The slope is set to be $S = 0.6$, and the inflexion point is at $t_0 = 4.0$.



(b) Sigmoid functions $Y(t)$ (orange) as the parameters vary, as could be inferred at different sites. Top left: left asymptote $A = 0.9$, right asymptote $K = 0.05$, slope $S = 0.6$, and inflexion point $t_0 = 4.0$. Top right: as in top left, except that the slope is 2. Bottom left: same as top right, except inflexion point $t_0 = 7$. Bottom right: same as bottom left, except left asymptote $A = 0.7$, right asymptote $K = 0.9$.

the number of free parameters in the likelihood ratio test, and improves the speed of the method.

5.2.6 Likelihood ratio test in model C

The discretized version of Pelican relies on a likelihood ratio test between a homogeneous model, where the same parameters apply throughout the tree, and a model where different amino acid equilibrium frequencies apply depending on the condition associated with the branch [Duchemin et al., 2022]. We adopt the same framework to assign p -values to sites, because the homogeneous model is also nested within model C . The number of parameters that need to be considered in this test is the two parameters in the sigmoid function, plus the number of amino acids present at the site, minus one. For a site where 4 amino acids are observed, the likelihood ratio test (LRT) is computed as follows. First, compute the log-ratio LR :

$$LR = -2 \left(\log L(\text{model } H) - \log L(\text{model } C) \right) \quad (5.6)$$

where model H is the homogeneous model. Then compare LR to a χ^2 distribution with $4 - 1 + 2 = 5$ degrees of freedom [Wilks, 1938].

5.2.7 Benchmark simulations

Tree and trait values

We chose a phylogeny of mammals present in the Orthomam database [Scornavacca et al., 2019] and used in [Duchemin et al., 2022]. To annotate this phylogeny with a continuous trait, we chose to use body mass. We gathered body mass values for every species in the tree from the databases PanTheria [Jones et al., 2009] and EltonTraits [Wilman et al., 2014], and from [Farré et al., 2021]. We provide a table of body mass values as supplementary material. Leaves were annotated with the \log_{10} of the species body mass, and internal nodes were annotated by reconstructing ancestral body mass using a Brownian motion model [Felsenstein, 1985]. Here we want to stress that this ancestral reconstruction is not meant as reliable estimates to discuss body mass evolution in mammals, but only as a framework to simulate sites that depend on a continuous trait. The annotated phylogeny is represented figure 5.2.

Simulations

Simulations were performed in a manner similar to [Duchemin et al., 2022], but adapted to continuous phenotypes. More precisely, simulations were based on a mutation-selection model, where each site is associated to site-specific amino acid fitness profiles. Two types of sites were simulated: sites not associated to the phenotypic trait, and sites associated to the phenotypic trait. The former were simulated under a constant mutation-selection model applying to all branches of the phylogeny. A single vector of amino acid fitnesses was drawn as in [Duchemin et al., 2022] and applied throughout the tree. The latter were simulated under mutation-selection models that changed at each branch of the phylogeny. More precisely, equilibrium amino acid frequencies changed depending on the phenotype value at the tip of the branch. To this end, 2 vectors of amino acid fitnesses were drawn per site, and were used to provide left (A in Equation 5.5) and right (K in Equation 5.5) asymptotic amino acid fitness values, for low and high values of the continuous phenotype respectively. Then, at each branch, the end phenotypic value was used to compute branch-specific amino acid fitnesses, according to the sigmoid link function between phenotype and amino acid frequency (figure 5.1a,

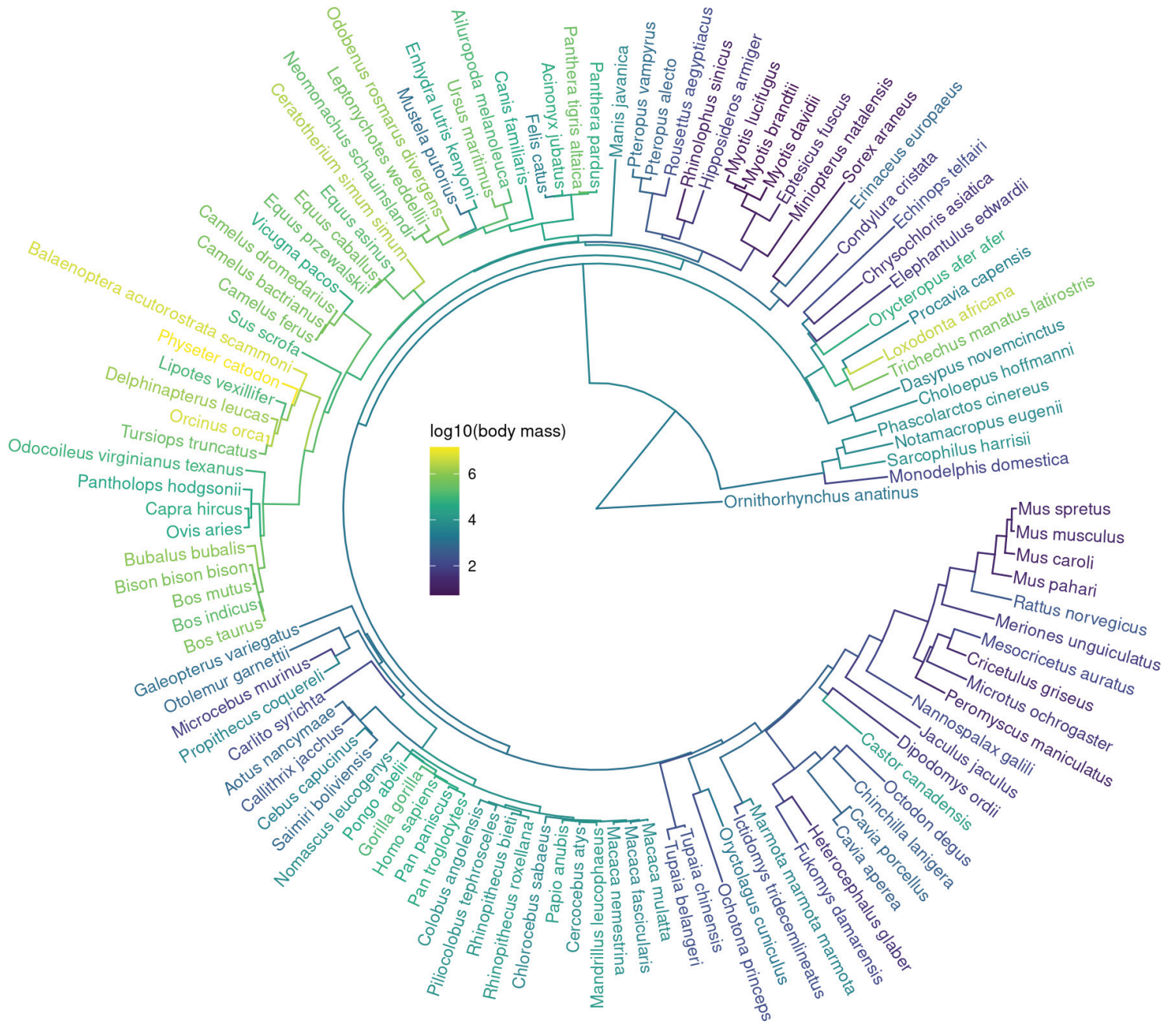


Figure 5.2: Orthomam tree with log₁₀ body mass value annotations

Equation 5.5). Other parameters of the sigmoid were:

$$\begin{aligned} t_0 &\sim N(\mu, 1) && \text{where } \mu \text{ is the trait value at the root of the tree} \\ S &= \lambda/\text{sd}(t) && \text{where } \lambda \text{ is a scaling factor for slope steepness} \end{aligned} \tag{5.7}$$

In this way, when $\lambda = 1$, the shape of the sigmoid is such that all the trait values are distributed around the inflexion point, mostly within the interval where the slope is substantial, avoiding the left and right plateaus. We tested 9 values of λ to investigate the performance of the inference method as the steepness of the sigmoid function varies. For each value of λ , 10 000 sites were simulated in total, with 1 000 H_A sites associated to the phenotype, and 9 000 H_0 sites not associated to the phenotype.

Evaluated methods

We compared the continuous version of Pelican to the discrete version, applied to the phenotype after discretization of the log body mass into categories. We named the continuous model “C”, and named the discretized models **D***. Model **D2 mean** works with 2 categories of log body mass, based on whether they fall below or above of the mean of the log body mass observed at the tips. Model **D2 median** is similar, except that the median is used instead of the mean, as is model **D2 phylo-mean** that uses the phylogenetic mean, which is the inferred trait value at the root of the tree, assuming a Brownian model for the trait. Model **D3 terciles** discretizes log body mass into 3 categories based on terciles of log body mass observed at the tips. All methods were run on the simulations described above, and their performance was measured using precision-recall area under the curve (AUC) following the methodology of [Boyd et al., 2013].

5.3 Results

5.3.1 Simulations

We simulated sites along a phylogeny of 116 mammalian species using a mutation-selection model. We simulated two types of sites: negative (H_0) sites for which the model of sequence evolution is constant across the phylogeny, and positive (H_A) sites for which the amino acid fitnesses depend upon a continuous trait evolving along the phylogeny. This trait was chosen to be the log of the empirical weights of the species included in the phylogeny. For inference, we compared our new continuous model (C) to discretized models where the log weight has been divided in 2 or 3 categories. Categories were built by using the mean or the median (for two categories, models **D2 mean** and **D2 median**, respectively), or terciles (model **D3 terciles**). All models are implemented in Pelican.

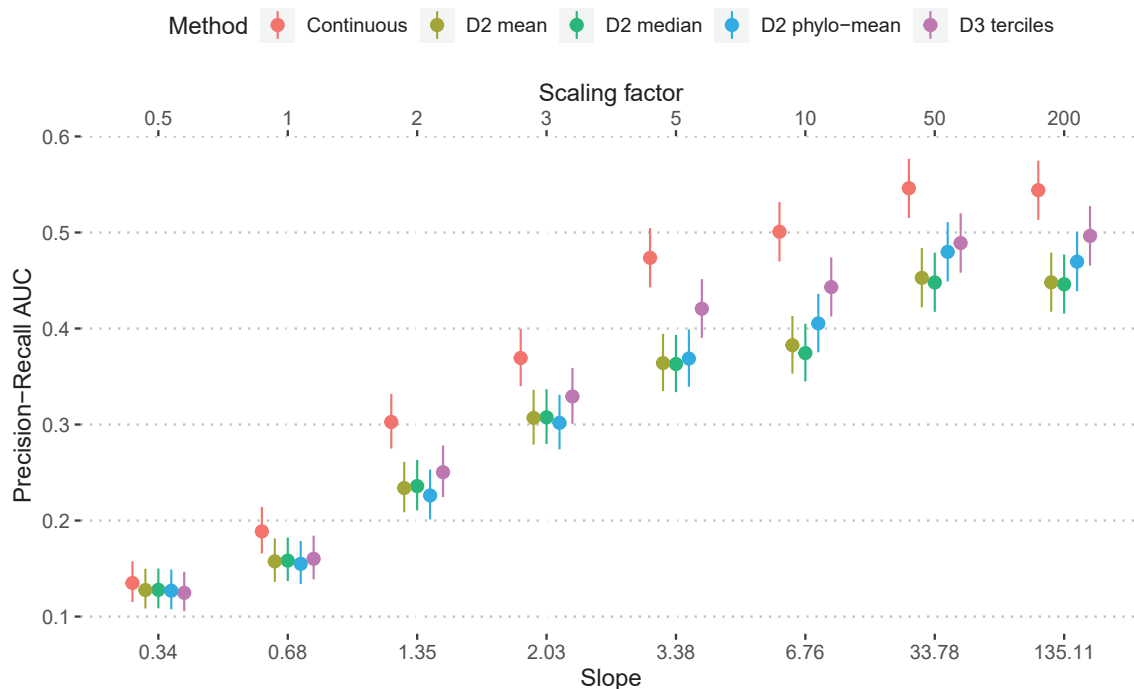


Figure 5.3: Continuous model (C) performs better than Discrete models (D*) across values of the slope binding trait values to fitnesses. Slope values were chosen so that baseline value $S = 0.68$ would make the sigmoid span the range of trait value, and was scaled using several factors (top x -axis).

Results are shown figure 5.3. We investigated the effect of varying the slope of the logistic function linking amino acid preference to the log weight value. Overall performance across all methods improves with the steepness of the slope, as stronger slopes introduce larger fitness differentials during the simulation, leading to a stronger signal in the alignment. We observe that model C performs better than discretized models in all considered situations. Even for large slopes, where the shape of the logistic function is such that it basically becomes a step function, model C outperforms the discretized models. Among discretized models, D3 `terciles` outperforms D2 `mean` and D2 `median`. Comparison of the three models discretized in two categories shows that the choice of the threshold values to build the categories can have a strong effect on the power of the method.

5.3.2 Longevity in mammals

[Farré et al., 2021] analyzed thousands of gene families to look for protein coding sites associated with variation in normalized lifespan across mammalian species. They used a two-step approach to identify candidate sites. First, they looked for sites where all 6 longest-lived species had the exact same amino acid, without gaps. Then among those they selected sites where the amino acids for the 6 shortest-lived species were different from that found in the longest-lived species. With this approach, they identified that three genes, WRN, ZC3HC1, and CASP10, previously linked with lifespan, contained 2, 1 and 6 sites of interest, respectively. We reanalyze these three gene alignments, using the same lifespan data. We use the continuous C model, as well as a discrete model with 2 categories split at the mean of the distribution.

Figure 5.4 shows that the discrete and continuous model often identify similar candidates, as red and black dots for the same site are often close to each other, in particular for low p -values. The C

and D model p -values are correlated, with $R^2 = 0.25$.

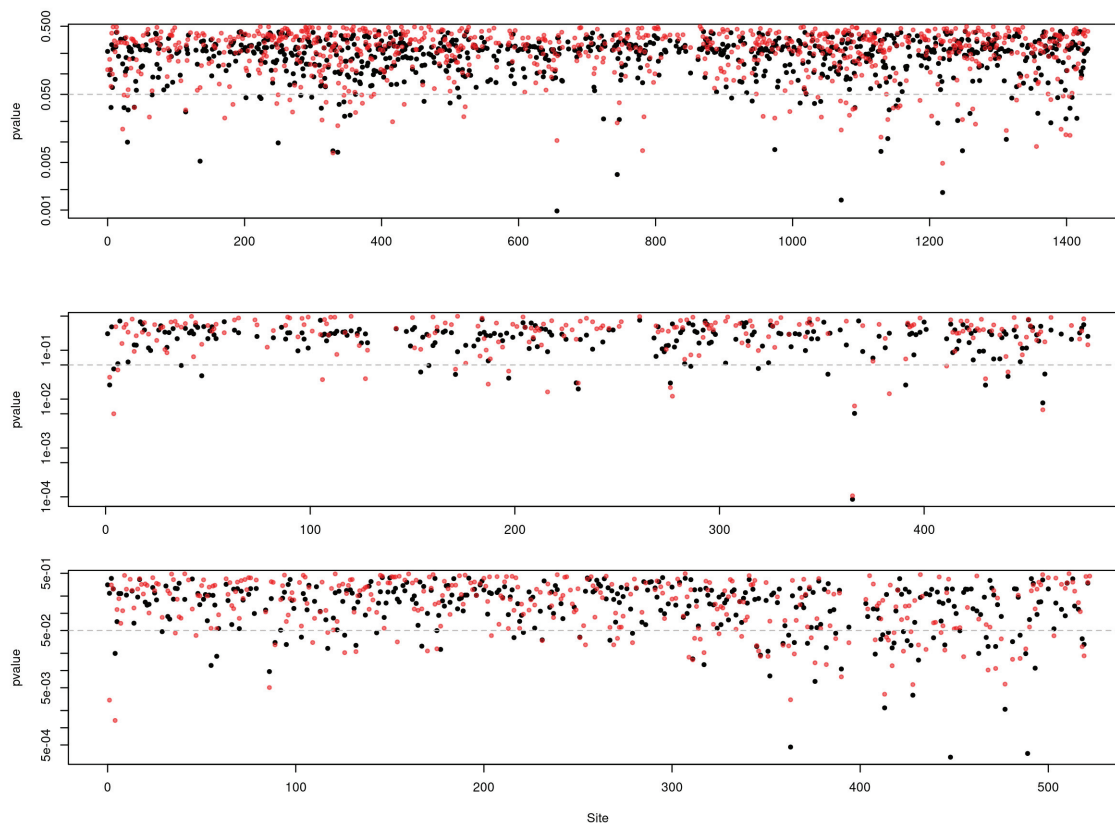


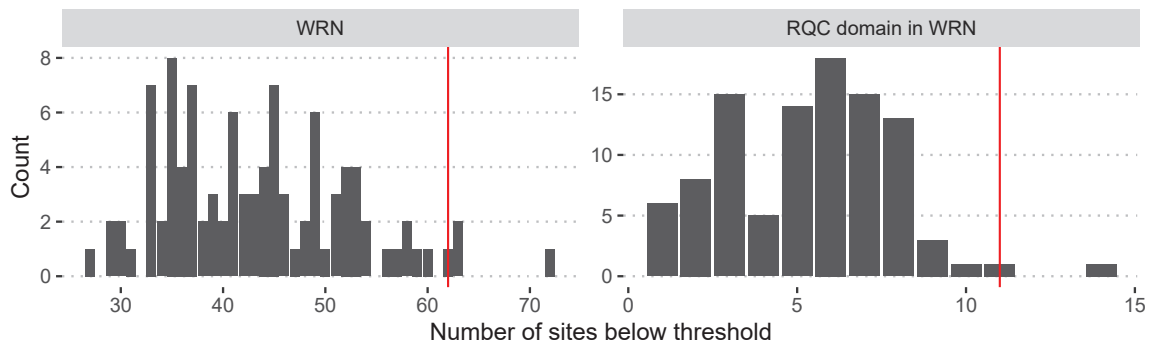
Figure 5.4: P-values for the continuous (red) and discrete (black) models for WRN (top), ZC3HC1 (middle), and CASP10 (bottom) genes. A dashed line indicates the 0.05 p -value threshold.

Estimating false discovery rates using the Benjamini-Hochberg procedure for multiple testing shows that for all genes, very few sites can be considered statistically significant at a 5% FDR threshold, in these data sets. Despite this it is still informative to use p -values to select the most convincing sites, as shown below. To this end we arbitrarily choose to consider sites with p -value < 0.05 .

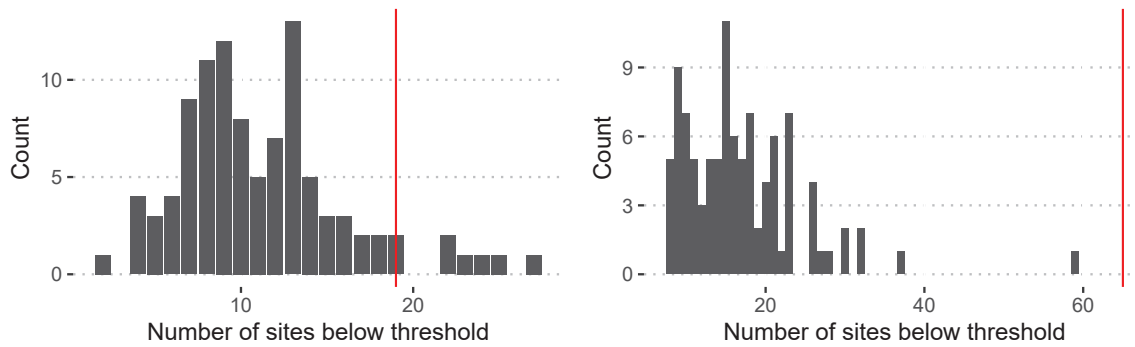
We found that WRN contains 63 sites (out of 1433 in *Homo sapiens*) for which model C's p -value < 0.05 . In itself, this number is not evidence for association with longevity in mammals: at the 5% threshold, under a uniform distribution of p -values, we expect 72 such sites. In addition, the two sites F1018L and N1055S/R/K/I/T identified by [Farré et al., 2021] have larger p -values, at 0.06 and 0.21, respectively. These two sites had been deemed promising because they are part of a RQC domain, in positions 949–1092 [von Kobbe et al., 2003]. RQC domains are involved in DNA unwinding, which is necessary for the repair process ensured by WRN. We observed that 11 of the 102 positions in this domain, *i.e.*, 11%, have p -value < 0.05 , compared to 4% for the rest of the protein (62 sites out of 1331 positions). This is more than the 5 sites expected under a uniform distribution. In comparison, the discrete model identifies 26 sites at the 0.05 level, associates p -values of 0.06 and 0.2 to the previously identified sites, and identified 6% of the RQC sites (6/102) with p -value < 0.05 , compared to 1.5% for the rest of the protein (20 sites out of 1331 positions).

To further evaluate whether the results of model C on the gene were significant, we performed 100 replicate analyses where the lifespan values had been shuffled. The numbers of sites with

Figure 5.5: Distribution of the number of sites with p -value < 0.05 over the three proteins in randomized replicates. The red line indicates the values obtained with the true lifespan data.



(a) Distribution over the entire WRN protein, or within the RQC domain.



(b) Distribution over the entire ZC3HC1 protein.

(c) Distribution over the entire CASP10 protein.

p -value < 0.05 over the entire protein and in the RQC domain are shown figure 5.5a. In both cases, an over-representation of positive sites is found with the true lifespan data. This over-representation is particularly pronounced in the RQC domain. This suggests that adaptation for extended lifespan may have taken place preferentially in the RQC domain.

We found that ZC3HC1 contains 19 sites (out of 482) for which model C's p -value < 0.05 . Interestingly, site T366S/A, which was identified by [Farré et al., 2021], is model C's best candidate, with p -value $= 4.6 \times 10^{-4}$. Comparison of the asymptotic amino acid profiles at this site shows that amino acid S has a much higher frequency in short-lived species whereas it's amino acid T that has the highest frequency in long-lived species (figure 5.6).

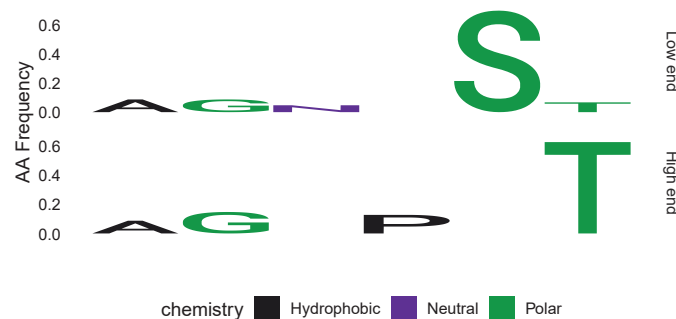


Figure 5.6: Asymptotic amino acid frequency profiles for site 366 in protein ZC3H1. "Low end" corresponds to amino acid frequencies estimated for short-lived species, and "High end" corresponds to amino acid frequencies estimated for long-lived species.

We performed 100 replicate analyses where the lifespan values had been shuffled, and obtained the results shown figure 5.5b. In 6 replicates out of 100, more sites with p -value < 0.05 were found. The discrete model identifies 11 sites, and also has site T366S/A as best candidate, with p -value $= 9 \times 10^{-5}$.

Finally, model C identified 65 positions with p -value < 0.05 in CASP10. This represents 12% of the 521 amino acid positions in the gene. Out of the 6 positions that [Farré et al., 2021] identified, only one has p -value < 0.05 . However, 100 replicate analyses where the lifespan values had been shuffled show that there is a clear excess of positions with p -value < 0.05 (figure 5.5c). The discrete model identifies 64 sites, 5 of which had already been identified by [Farré et al., 2021].

Based on this analysis of three genes, it appears that model C as well as the discrete model identify more sites than the two-step approach used by [Farré et al., 2021], and propose different candidates. Replicate analyses where the lifespan values have been shuffled suggest that, at least for WRN and CASP10, there has been an enrichment for sites associated to lifespan.

5.4 Conclusion

In this article we have presented a new model to search for sites associated with a continuous phenotype. We have implemented this model in our software Pelican, which can also search for sites associated with discrete phenotypes. We have shown on simulations that the continuous model could outperform discretized models. Looking forward, we expect that Pelican will be used in a variety of clades to find new associations between sites in protein sequences and continuous or discrete phenotypes.

5.5 Acknowledgements

We thank Arcadi Navarro Cuartiellas and Gerard Muntané Medina for their insights on this work and for sharing their data.

5.6 Supplementary material

The Orthomam phylogeny and the body-mass trait annotation for each of its species that we used in our simulations are available as supplementary data at <https://doi.org/10.5281/zenodo.7414499>. The corresponding code repository is available at <https://gitlab.in2p3.fr/phoogole/pelican-continuous-benchmark>.

5.7 Additional remarks and discussion

The article draft that makes up the content of this chapter has not been submitted yet, as we think further investigations are needed. The set of positions that we identify do not match those reported in [Farré et al., 2021], and this calls for a more systematic comparison. The encouraging results that we obtained using simulations must be mitigated by recalling the underlying hypothesis of our simulation setting, in which we assume a sigmoid relationship between amino acid fitnesses and trait values. This assumption might not be realistic enough to give us good confidence that the predictions obtained on empirical data are accurate. More work is needed to validate and improve this model, that we lacked the time to do yet.

Reduced throughput compared to the discrete model Even though the continuous model that I described in this chapter only involves two more parameters compared to the discrete model of Pelican, it is quite a lot more costly in computation time. The reason for this is not the extra parameters in the model, but rather the computation of the probability matrices for each trait value. In the discrete model, using the procedure that I described in section 4.3.3, probability matrices are computed by first performing the eigen-decomposition of the rate matrices. This decomposition only has to be done once for each rate matrix, i.e. once for each trait condition, and is re-used at every branch that shares the same trait annotation. In the continuous model, this is not true: each branch in the tree is very likely to have a trait value this is unique in the tree. As a consequence, there is a different rate matrix for each branch in the tree and eigen-decomposition has to be performed for every one of them, increasing the computational cost of fitting the model. This additional cost could be mitigated by identifying regions in the sigmoid curve where the frequency differential between a group of trait values is small (i.e. flat regions, or clumps of trait values that are very similar) and use a single rate matrix for all of these trait values. This approximation could reduce the total number of different rate matrices that have to be decomposed and improve the computation time, hopefully without degrading too much the quality of the fit.

Calibration As would be expected, p -values are not better calibrated than when using the discrete model of Pelican (data not shown).

Ancestral traits inference We use a Brownian model of evolution to reconstruct ancestral phenotypes at maximum-likelihood. This is a naive approach, that is probably too simplistic: it assumes that traits vary with regularity along the tree, and independently between species after they have diverged. Inference of ancestral phenotypes is a research subject in itself, and should be ideally done by integrating both genotype and phenotype data — a problem that is far from being solved at the moment.

 **Chapter 5 summary: *Continuous trait associations***

We built on the discrete model underlying Pelican to provide a variant that would enable the analysis of continuous phenotypic traits, without the need to define a discrete categorization for it, that can be somewhat arbitrary. This variation on the discrete model preserves its general framework: it is a model of amino acid substitutions, shaped by amino acid stationary frequency profiles. Two asymptotic profiles that describe the stationary distribution for extreme trait values are fitted on the data, and frequency profiles for values in-between them are interpolated using a sigmoid function. The shape of this sigmoid function is controlled by two additional parameters per site: a *slope* parameter that determines the steepness of the curve, and a *shift* parameter that is its inflexion point on the axis of trait values. A specific frequency profile for each trait value is thus determined in a continuous fashion.

This continuous model was first evaluated on simulations, using our mutation-selection model which was adapted to simulate on continuous traits, using a similar sigmoid interpolation on amino acid fitness profiles. Comparing the precision-recall of our continuous model to the results obtained with the discrete model using several discretization strategies, we find that the continuous model makes consistently better predictions.

We then compare our findings to previously published results [Farré et al., 2021] that investigate the association of three genes to the longevity of mammal species using an approach based on discretization of the trait with an emphasis on extreme values. Our results differ from those reported in this study, suggesting that more work is currently needed to understand this discrepancy and further validate our approach.

Chapter 6

Gene-level predictions

Pelican implements the site-independent model of TDG09, and therefore produces predictions in the form of p -values at each site in an alignment. In that sense, the result of running Pelican on an alignment is the realisation of multiple independent tests for the same null hypothesis, which is the independence of the site to the phenotype. There is thus an incentive to leverage all of this information to ask other questions. For example, at the level of genes: how many sites are predicted positive within a gene? Which genes are the most likely to be associated to the phenotype? What is their proportion in the genome?

To answer these questions, we can look for patterns among genes that signal an enrichment in low p -values, and would be unexpected under the null hypothesis that the gene is not associated to the phenotype under consideration. Some statistical tools designed for this task exist, and I discuss their application to our specific case, but they are often based on the hypothesis that the p -values are calibrated, i.e. uniformly distributed under the null. This is generally a reasonable assumption, since it is an expected property of p -values, but it is not verified in our case as figure 6.1 illustrates.

This calls for other approaches to work around the lack of calibration and still deliver reliable predictions at the level of genes. I describe and compare in this chapter several approaches to aggregate p -values across genes: conventional ones such as computing a false discovery rate (FDR) and Fisher's method, as well as other more specific methods designed to handle the particularities of our problem. To evaluate the validity of these approaches, we rely once again on simulations of genes under our mutation-selection model (as described chapter 1.3.1): 500 H_0 genes are simulated, each made up of 500 H_0 sites whose fitness profile remains constant; 100 H_A genes are simulated, which contain 490 H_0 sites, and 10 H_A sites whose fitness profile depends on the phenotype at each branch in the tree. Simulations are conducted on the same six empirical phylogenies that we have used so far.

These investigations completely overlook the question of the aggregation of results from the continuous model of Pelican, and focus solely on the discrete model. It would be useful however to question whether the results presented in this chapter remain similar when transposed to the continuous model, so that we may know which aggregation approach is better adapted in this case.

The end of the chapter exposes results obtained from analysing an empirical dataset: the Orthomam database, a curated set of alignments of coding sequences for 116 mammal species. Analyses are conducted looking for associations to several discrete phenotypes, and the resulting lists of best gene candidates are confronted to the literature as well as Gene Ontology databases to evaluate the quality of predictions.

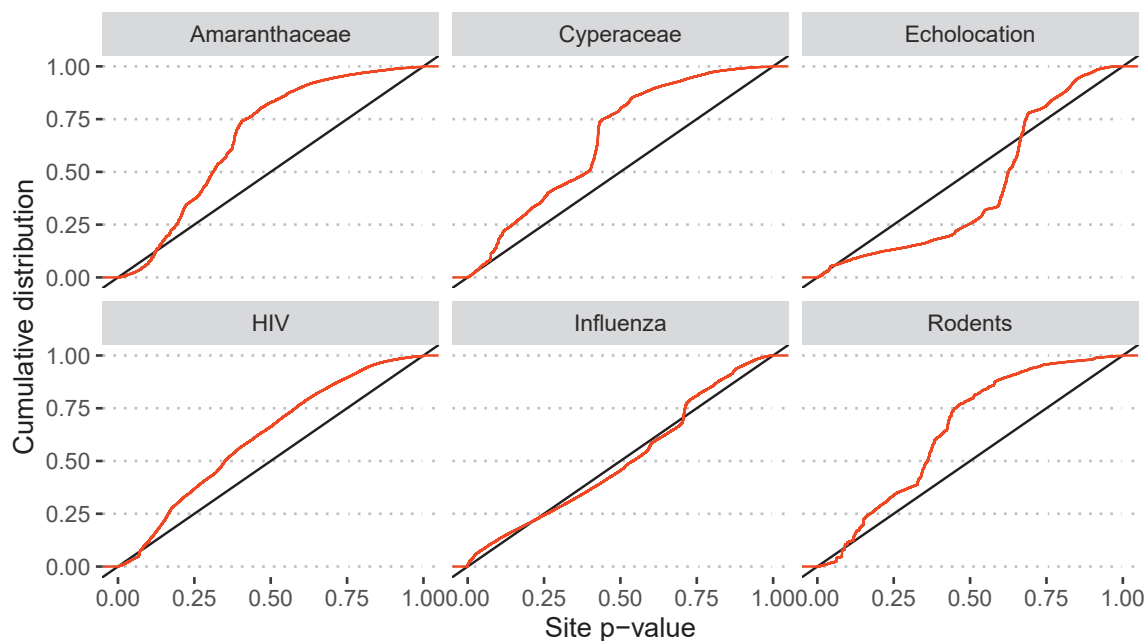


Figure 6.1: Site p -values resulting from Pelican scans are generally not uniformly distributed under its null hypothesis: the cumulative distribution of p -values under the null for a well calibrated method would follow the black straight line in these plots.

Contents

6.1	How many positive sites are there within each gene ?	131
6.2	Which genes are most likely to be associated to the phenotype ?	132
6.2.1	Wilkinson's method	132
6.2.2	Fisher's method and derivatives	133
6.2.3	A mixture model for the distribution of p -values	137
6.2.4	Influenza strikes back	140
6.3	How many genes are associated to a given phenotype ?	143
6.4	Detection of genotype-phenotype associations in the Orthomam database	146
6.4.1	Echolocation	147
6.4.2	Diet	151
6.4.3	Aquatic and marine	153
6.4.4	Subterranean	156
6.4.5	Other phenotypes	159

6.1 How many positive sites are there within each gene ?

Predicting the number of sites within a gene that may be identified as positive can be done by controlling the false discovery rate (FDR) within a gene. The idea is to leverage the independent repetitions of the test performed using Pelican on each site to determine a threshold at which the proportion of false positives is known. The most commonly used approach to achieve this is the Benjamini-Hochberg procedure. It consists in ranking the site p -values in ascending order, denoting k their rank within one gene, and setting a target proportion of false predictions α . We then find the largest p -value such that $p \leq \frac{k}{n}\alpha$, where n is the total number of tests (i.e. the number of non-constant sites in the gene), and reject the null hypothesis for all sites having a p -value lower or equal. The resulting list is expected to contain a proportion α of false predictions.

We must be cautious when using this procedure, as its premise is that p -values are uniformly distributed. To evaluate how the actual quantity of false predictions we obtain compares to the target proportion, we build the distribution of the fraction of false predictions on each simulated dataset at a chosen FDR threshold $\alpha = 0.01$ that is presented figure 6.2.

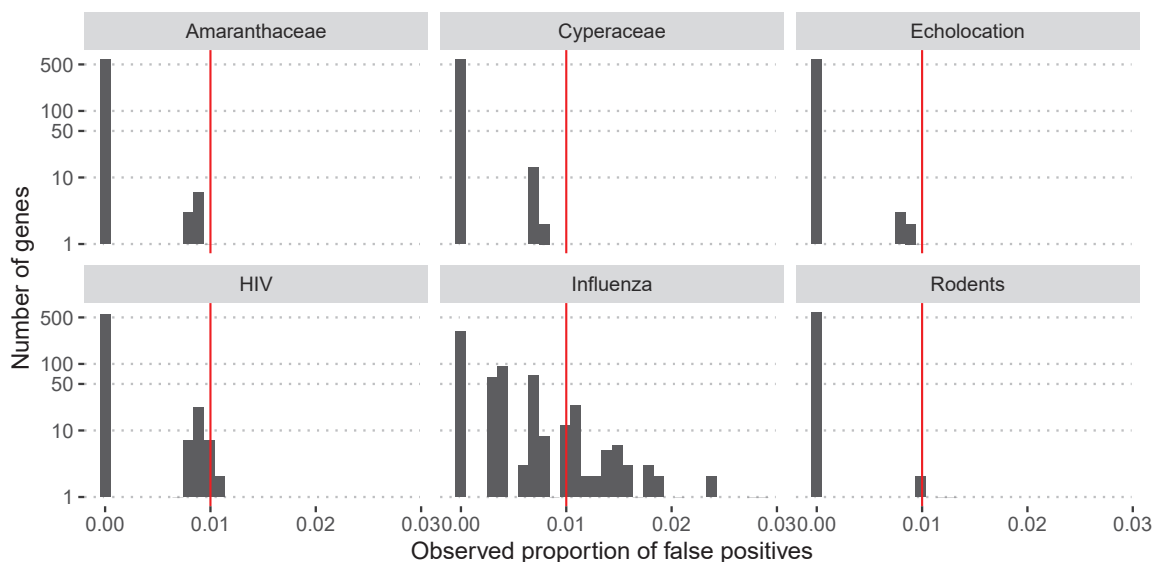


Figure 6.2: Distribution of the false discovery rate for the task of predicting positive sites in a gene, using a threshold corresponding to a target FDR of 0.01 using Benjamini-Hochberg procedure. Each histogram corresponds to a different tree for which 500 genes were simulated. The red vertical line shows the target FDR, and the distribution is expected to be on its left in favorable cases.

These results show that whenever genes include false predictions in their sites, their proportion is generally in adequacy with the threshold that was chosen. It is less true for the Influenza dataset, where the observed false discovery rate is more variable and often exceeds the desired threshold, because of its small enrichment in low p -values under the null — I will come back to this through this chapter. It appears that the Benjamini-Hochberg procedure can generally be applied at the level of sites within each gene to control the false discovery rate, even though site-level predictions are not calibrated. The proportions of p -values below the threshold are generally close enough to the value of the threshold itself. They are not affected by the large deviations from the uniform of higher p -values under the null.

6.2 Which genes are most likely to be associated to the phenotype ?

Although predictions at the level of sites are informative, it would be useful to establish as well a list of genes that are enriched in sites that are likely to be associated to the phenotype under consideration. To that end, we investigate several procedures to aggregate p -values obtained from Pelican at the level of sites, and produce scores at the level of genes. The problem of combining p -values from multiple independent tests is common, e.g. when performing meta-analyses, and several approaches have been reviewed and compared in [Loughin, 2004]. Some of these are evaluated in this chapter, along other approaches tailored for our specific needs.

The ability of each aggregation method to correctly distinguish H_0 genes from H_A genes is estimated using precision-recall AUC. Results are shown in figure 6.3 for each method that we investigated, on all 6 phylogenies. The following sections present each aggregation method with

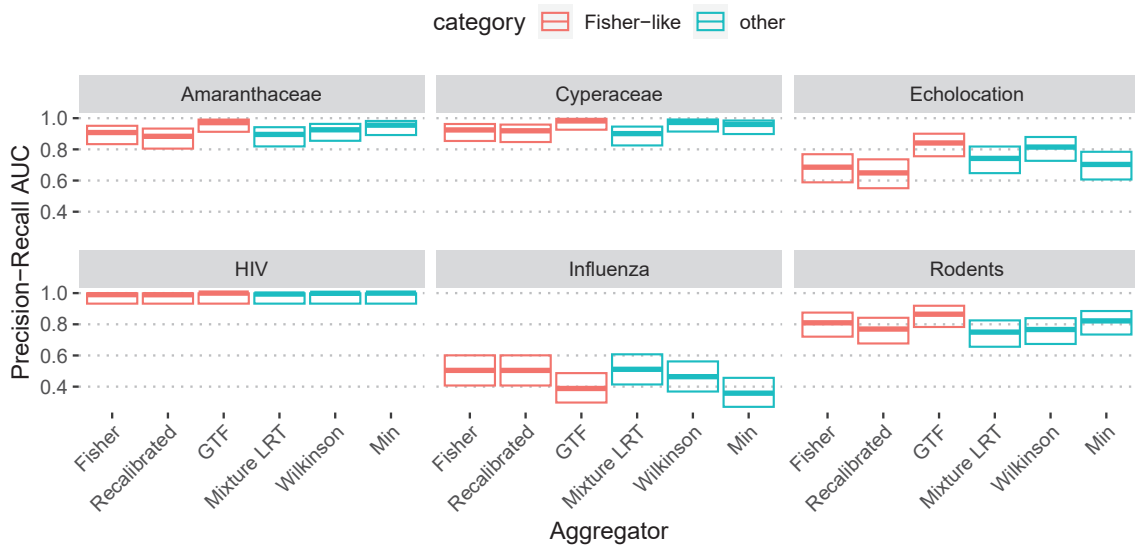


Figure 6.3: Evaluation of several aggregation methods to compute a score on genes from the p -values at each of their sites.

more details, and attempt to provide explanations for discrepancies in their performance.

6.2.1 Wilkinson’s method

Wilkinson’s method [Wilkinson, 1951] is an approach to integrate p -values resulting from multiple independent tests into a single p -value. It has been previously used in a context similar to ours, to compute gene-level p -values from sets of site-level p -values, when testing for evolutionary convergence within sequences [Chabrol et al., 2018]. This approach stems from a simple idea, which consists in counting for each gene the number of sites whose p -value is below a significance threshold γ . The number of sites C_i in gene i that pass the threshold is assumed to follow a binomial distribution, under the null hypothesis that the gene is a negative (i.e. not convergent, or independent from the phenotype under consideration in our case).

$$C_i \sim \text{Binom}(\gamma, l_i) \quad \text{where } l_i \text{ is the length of the gene } i \quad (6.1)$$

A gene-level p -value can then be computed in a straightforward manner, as the fraction of the density of the binomial distribution that lies above the value of C_i .

Of note, this approach does require that we choose a significance threshold on site-level p -values, which is problematic since they are not well calibrated. This implies that the threshold must be set somewhat arbitrarily and does not accurately represents the risk of type 1 error. Moreover, as would be expected, the choice of the significance threshold has a noticeable impact on the prediction of genes using this method, and the optimal value is not constant across datasets as shown in figure 6.4.

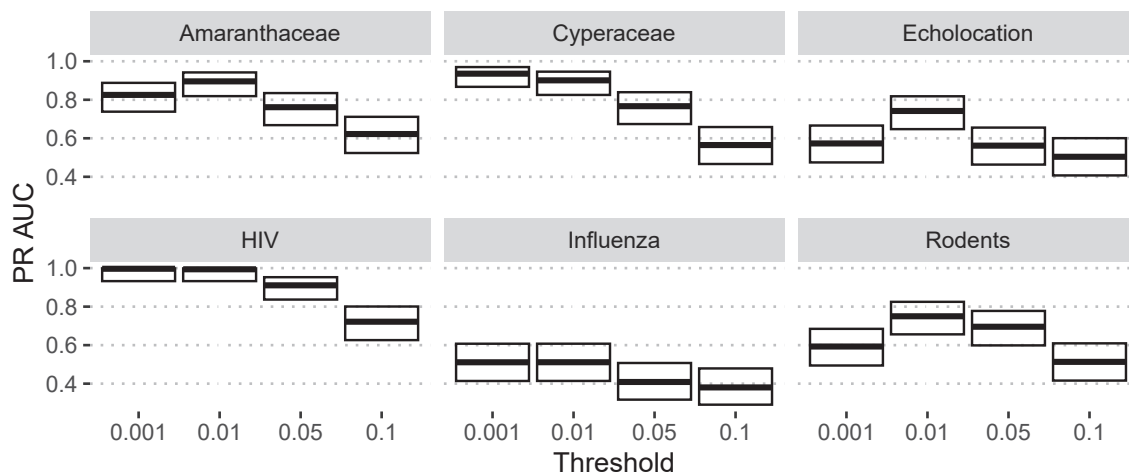


Figure 6.4: The choice of the significance threshold for site-level p -values is determinant for the precision-recall performance at the level of genes when using Wilkinson’s method in our context. The optimal threshold varies across datasets.

Based on these results, the value $p = 0.01$ for the significance threshold seems to be the best compromise across datasets, and was chosen to compare this approach to other aggregation strategy (figure 6.3 at the beginning of this section). This is a somewhat dubious way to proceed, as it gives a slight advantage to this method that will not be accessible when working with real datasets.

6.2.2 Fisher’s method and derivatives

Fisher’s method

Fisher’s method [Fisher, 1932] provides a score that combines p -values from n independent tests all having the same null hypothesis into one test statistic, that we call S . This test statistic follows a chi-square distribution with $2n$ degrees of freedom under the null hypothesis that all p -values are uniformly distributed. This null hypothesis can also be reformulated a bit more clearly as “all independent tests within the group fail to reject their own null hypothesis”.

$$S = -2 \sum_{i=1}^n \log(p_i) \sim \chi_{2n}^2 \quad (6.2)$$

The main limitation when applying Fisher’s method in our context arises from the deviation of Pelican’s site-wise p -values from the uniform distribution, under its null hypothesis. I have already discussed the calibration issues that are rampant in Pelican’s testing procedure, and exposed that the distribution of p -values under the null hypothesis is generally not uniform. This calls for attempting

to lean towards p -values that are better calibrated, which is the motivation for the approach I describe next.

Empirical correction of p -values

We can use the prior that only a few sites within a dataset are expected to be associated to a phenotype, to take a random sample ($m = 10\,000$) within the global site pool ($n = 300\,000$). Since only 1000 sites were simulated under differential selection, we can make the reasonable assumption that the sample only consists in negative sites. From this sample, we compute an empirical distribution of p -values under the null, which we can leverage to correct the calibration of the p -values globally in the dataset. “Corrected” p -values p^* are then calculated as

$$p^* = \frac{1 + F(p) \times m}{1 + m} \quad (6.3)$$

where F is the empirical cumulative distribution function (ECDF) of the sample of p -values, which we apply to get the percentile in this distribution corresponding to each p -value in the output of Pelican. We use pseudo-counts in this calculation by adding 1 to each term, in order to avoid the generation of p^* strictly equal 0. Fisher’s method, described above, is finally applied on each gene’s p^* .

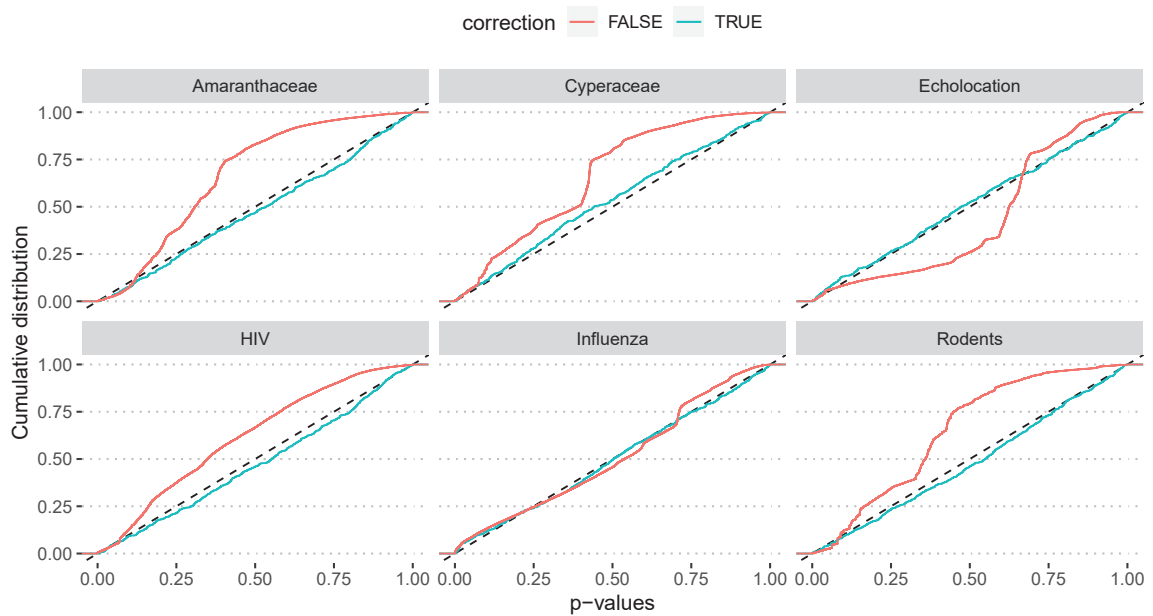
Figure 6.5a shows that the resulting p^* are quite a lot closer to the uniform than the non-transformed p -values when we look at them globally across all genes. The calibration is also improved at the level of each gene, as show in figure 6.5b, but was examined only for a small set of genes and there is no guarantee that it is the general case.

The impact of this in term of precision-recall performance is a bit underwhelming: this correction of site p -values does not improve the gene-level predictions that are made using Fisher’s method, as shown figure 6.3.

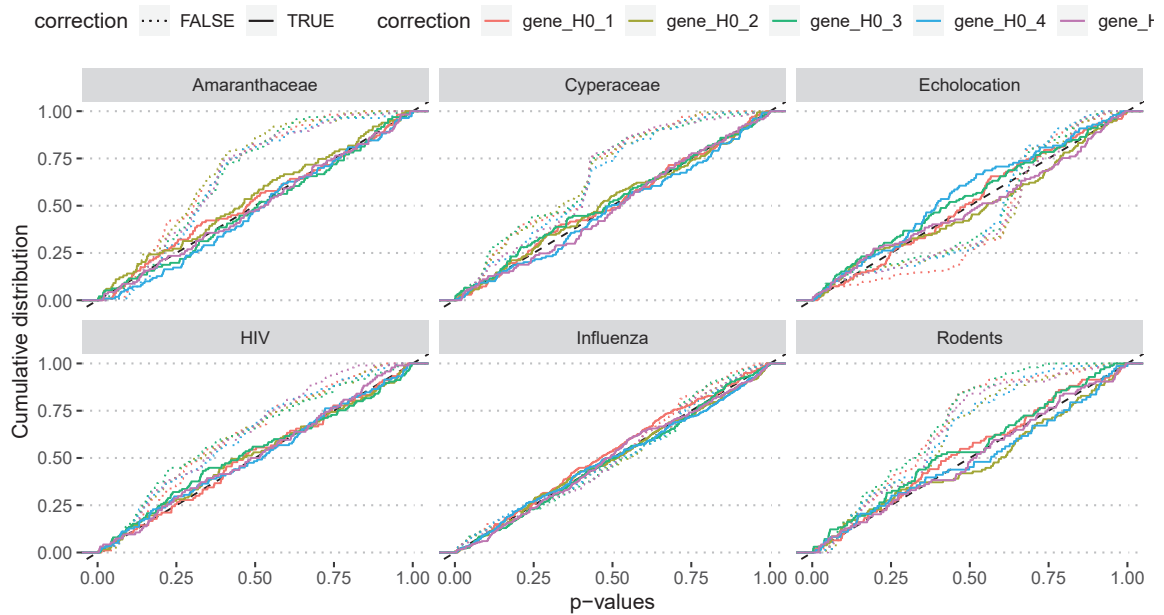
Gene-wise Truncated Fisher’s method

The idea for this approach originally came from our colleague Anamaria Necşulea, who was at that time giving a try at applying Pelican to scan for sites related to a morphological trait among bird species. She found out that the direct application of Fisher’s method did not produce convincing results at the level of genes, and tried a different approach: it consists in taking the k lowest p -values in the global pool of site-wise results, with k being set arbitrarily (e.g. $k = 1000$). A score for each gene is then calculated using equation 6.2, on the restricted set of the lowest p -values — genes having no p -values in this set are assigned a score equal zero. Because the method operates on a filtered set of p -values, there is no way to test for the significance of each gene-level score. However, the precision-recall AUC can be used to evaluate the ranking of the genes according to their assigned score.

Building on this approach, we came up with the Gene-wise Truncated Fisher’s (GTF) method which is a variation that retains the same core idea — looking at the lowest p -values — but allows the calculation of p -values at the level of genes instead of Fisher scores only. The main difference is that we select the k (e.g. $k = 5$) best p -values in each of the genes, instead of filtering across all p -values pooled together. The underlying assumption is that the number of positive sites in each gene is expected to be low: only a few positions within the protein sequence are expected to be



(a) Calibration of p -values across all sites is improved: the distribution of site p -values is closer to the uniform after correction.



(b) Comparison of distributions of p -values for five genes, with or without empirical correction. Calibration of site p -values within genes seems to be improved, but is variable across genes: there is no guarantee that this improvement is systematic across all genes.

Figure 6.5: Calibration of site p -values is improved when transformed using the empirical correction.

associated to a given phenotype¹. In this way, the inspection of the k lowest p -values in a given gene should inform us on its association to the phenotype under consideration. This is simply done by calculating a score G_k^i for each gene i , from the set of its k best p -values, using the Fisher’s score expression from equation 6.2.

$$G_k^i = -2 \sum_{j=1}^k \log(p_{(j)}^i) \quad (6.4)$$

where $p_{(j)}^i$ denotes the j^{th} lowest p -value obtained in gene i . We can not compute a p -value for each gene-level score using a chi-square distribution, as would be done in the Fisher’s method, because only a subset of the site-level p -values are considered in the score. However, assuming that the distribution of site-level p -values under the null hypothesis “There is no positive site in the alignment” is close enough to the uniform distribution — at least locally for the first k of them —, we can determine an empirical null distribution of the score. This can simply be done by drawing n values from the uniform, where n is the number of non-constant sites in the gene, and computing the G-score from the k lowest values that were drawn. Repeating this procedure gives us an empirical distribution of G-scores under the null for each gene, allowing the computation of a p -value at the gene level from the G-score.

The determination of the empirical null distribution can be done more efficiently by leveraging *order statistics* of the uniform distribution. When drawing from a uniform distribution, the m th lowest value $U_{(m)}$ in a sample with size n follows a beta distribution whose parameters depend on m and n

$$U_{(m)} \sim \text{Beta}(m, n + 1 - m) \quad (6.5)$$

This property makes it possible to avoid repeatedly drawing n values from the uniform and sorting them to obtain the k first p -values. Instead, we may directly draw values from k beta distributions while adjusting their parameter m to range in $\{1, \dots, k\}$. The size of the corresponding uniform sample n is still the number of non-constant sites in the gene.

Incidentally, as shown figure 6.6, the empirical null distribution of gene scores is well approximated by the log-normal distribution, with its mean and variance parameters estimated from the sample of null scores. Despite the mathematical dissimilarity between the beta and log-normal distribution, the adequacy between the two have previously been reported [Barrett et al., 1991]. This approximation is useful in two ways: first, the size of the sample required to accurately compute the mean and variance is smaller than what is needed to build the empirical distribution, enabling faster computation of the null distribution; second, the precision of gene-level p -values is less limited when using a known distribution than when using an empirical distribution, where very small non-zero p -values would require a very large sample size.

One potential pitfall in this procedure is obviously that the value of k is set somewhat arbitrarily. It should reflect our prior that in positive genes, only a few positive sites should be present and have low p -values as determined by Pelican. But how many is “few” ? How sensitive is the result to the value that is chosen for k ?

To answer these questions, we measured the precision-recall on our simulation for a range of values for $k \in [1; 30]$, that are shown figure 6.7. The true number of positive sites is 10 in each H_A gene, and we find that the quality of predictions is not very sensitive to using a k value that strays away from this number, if the difference is not too high. As expected, Influenza is the exception once again; see section 6.2.4 for an investigation of what is happening with this dataset.

¹Although this may not always be the case: for example genes in the process of pseudogenization may have a larger proportion of variable sites.

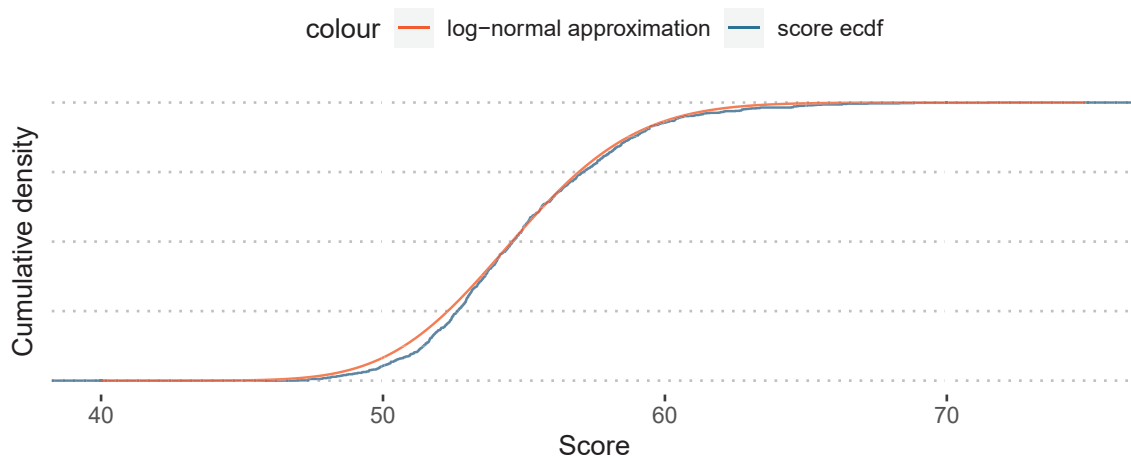


Figure 6.6: The empirical distribution of null scores (blue line) can be approximated well using a log-normal distribution (red line). The mean and variance parameters of the log-normal are estimated from the empirical mean and variance of null scores. The distribution is shown for ($k = 5; n = 500$). Distributions for other combinations of k and n are presented in appendix figure D.1.

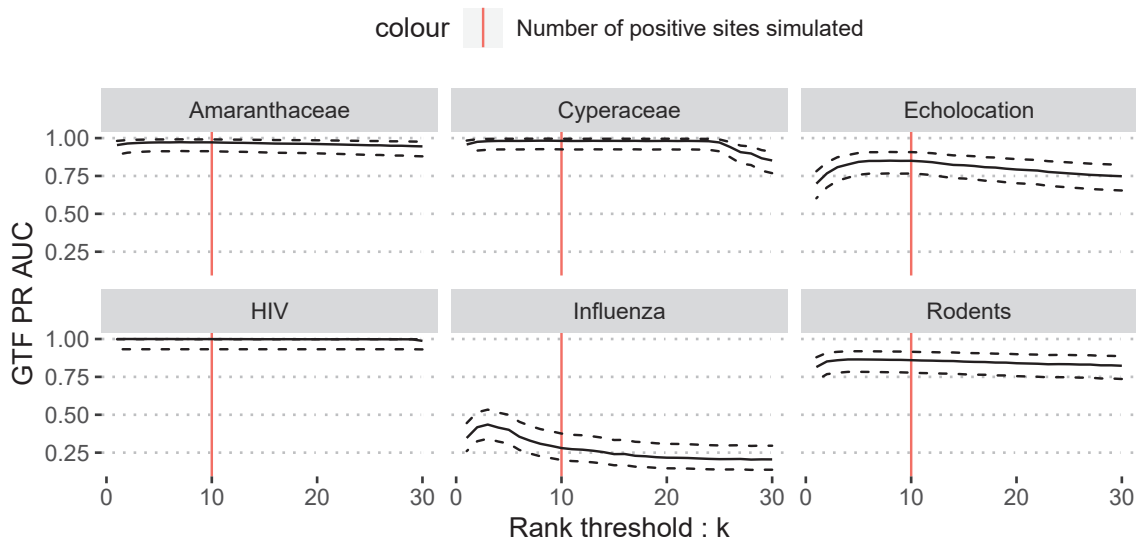


Figure 6.7: Performance of the GTF aggregation method depending on the rank threshold k that is used to compute the score.

6.2.3 A mixture model for the distribution of p -values

We devised a different approach, where we attempt to model explicitly the distribution of p -values. Under this model, the distribution of p -values of sites in a given gene is expected to follow a mixture of uniform and beta distributions, with proportions controlled by a parameter q . The distribution of p -values for positive sites is expected to be enriched in low values, and is represented as a beta distribution, controlled by a shape parameter θ . On the other hand, p -values obtained for negative sites are uniformly distributed.

This model thus makes the assumption that the method generating the p -values is well calibrated, and produces uniform p -values under the null. This hypothesis is generally not verified for Pelican, that produces p -values that are not uniformly distributed, and generally accumulate in the right

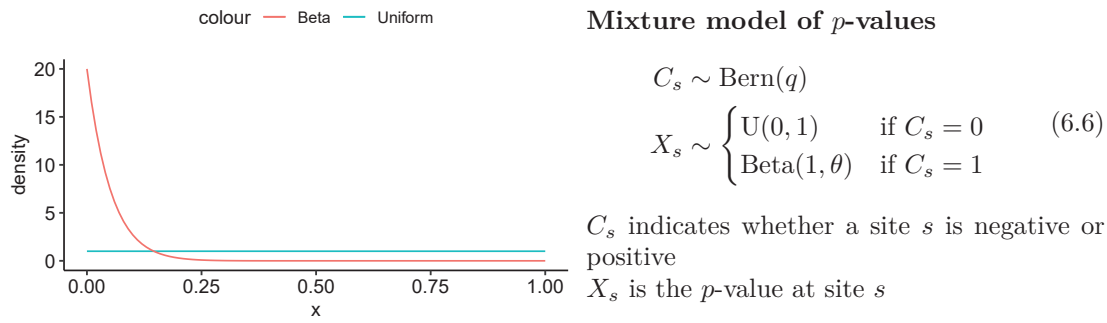


Figure 6.8: The distribution of site p -values is modeled as a mixture between a uniform distribution (H_0 sites) and a beta (H_A sites) whose shape is controlled by a parameter $\theta \gg 1$.

quadrant of the distribution (figure 6.1). Although this implies this is not the “right” model for the distribution of p -values, it could still work in practice since this clump of p -values should be captured under the uniform distribution rather than the beta, assuming its shape is such that most of its mass is accumulated on the left.

The model involves two parameters that must be estimated: the proportion q of positive sites within the gene, and the shape θ of the beta distribution that captures the p -values of positive sites. It also includes a “hidden” state C_s that indicates the category of a site s .

Fitting the model using the EM algorithm

Because we do not know with certainty which category a site belongs to, we must integrate over all possible values of C_s when fitting the model. The expectation-maximisation (EM) algorithm is well suited to fit the model in such situations. It is an iterative algorithm that works in two phases: determine a function for the expectation (E) of the log-likelihood at the current estimate of the model parameters, then compute new parameter estimates that maximize (M) this function. These two steps are repeated until a condition is reached, e.g. when parameter estimates or the log-likelihood remain stable between iterations. In our case, the condition for convergence is that parameter values do not change by more than $1e^{-3}$ between iterations.

The equations for each step are specific to a given model, and their derivation is quite lengthy. For this reason, the complete reasoning leading to the expressions used in both steps is provided in the appendix section D.2.

The expectation step consists in determining a probability distribution for the latent variable C_s , given some parameter values. Because C_s is a binary variable, this simply consists in defining the probability that it is 1 or 0.

$$\pi(C_s = 1) = \frac{q \times \text{dbeta}(X_s, 1, \theta)}{q \times \text{dbeta}(X_s, 1, \theta) + (1 - q)} \quad (6.7)$$

In the maximisation step, parameter values are actualized with new values that maximize the log-likelihood expectation function, assuming that the latent variable is distributed as determined in

the expectation step.

$$\hat{q} = \frac{\sum_s \pi(c_s = 1)}{N}$$

$$\hat{\theta} = - \frac{\sum_s \pi(c_s = 1)}{\sum_s \pi(c_s = 1) \log(1 - X_s)} \quad (6.8)$$

To prevent the beta distribution from capturing high p -values, or even taking a uniform shape ($\theta = 1$), a minimum value is set for θ . In order to avoid setting arbitrary lower bounds on θ , while still guaranteeing that the beta distribution is sufficiently distinct from the uniform, we devised a simple heuristic. It consists in setting a lower bound for θ that is equal to the number of non-constant sites in a gene. The idea is that when the pool of p -values is large, only very low p -values identify positive sites, while this constraint can be relaxed when the pool is smaller. Constant sites are ignored, since the corresponding p -value is always equal 1. This heuristic seems to work well in most situations, as shown figure 6.9, and alleviates the need for finding an optimal bound on θ , which is variable between datasets.

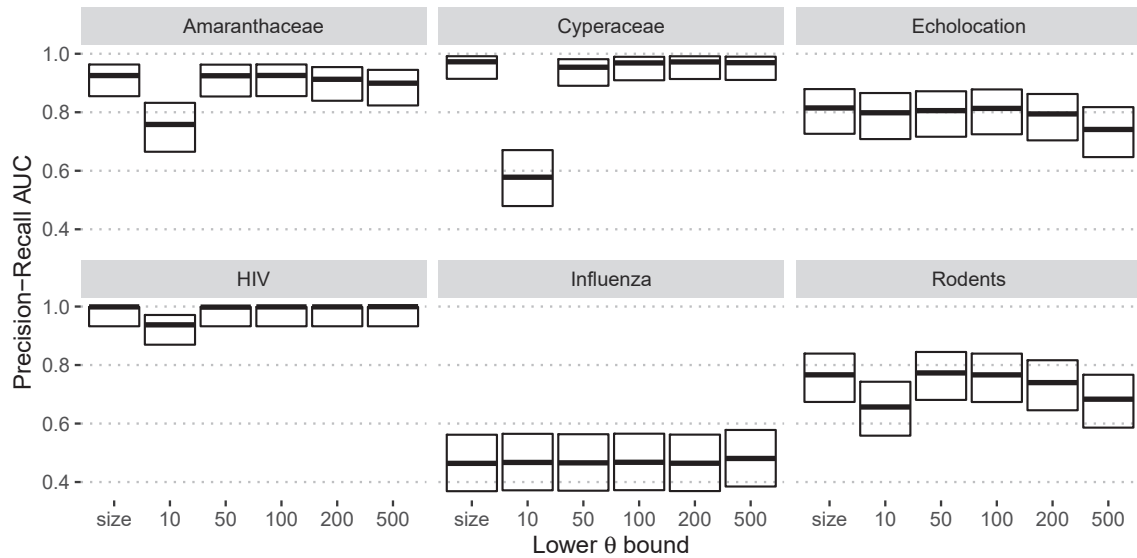


Figure 6.9: Precision-recall performance of the method for several lower bounds on θ . The `size` label denotes the use of the heuristic, which seems to provide optimal performance compared to other strategies where the bound is fixed for all genes.

Hypothesis testing

Once the model is fitted to the data at maximum likelihood, we can then compare its fit to that of a constrained version of itself, where $\theta = 1$ so that all sites have p -values uniformly distributed, regardless of their nature. As the two models are nested within each other, their comparison can be done using a likelihood ratio test (LRT) with 1 degree of freedom, as I previously described in section 4.2.4.

6.2.4 Influenza strikes back

Gene-level predictions on simulations with the Influenza phylogeny are a lot less accurate than those obtained for the other phylogenies, as is shown figure 6.3. This is unexpected, since this dataset seems to fulfill the asymptotic conditions for the likelihood ratio test, that we have identified in section 4.4. Something else must be at play in this case, that could explain the underwhelming quality of predictions in this dataset. Although the Influenza tree is both large and long, and represents a large effective sample size, it was already problematic in one specific situation when simulating relaxed selection, as discussed in section 4.5. Whereas in the current instance, simulations are conducted under differential selection, which suggests that we are still missing part of the explanation. Understanding what causes this situation is important to identify which features of a dataset should be watched out for in the usage of Pelican.

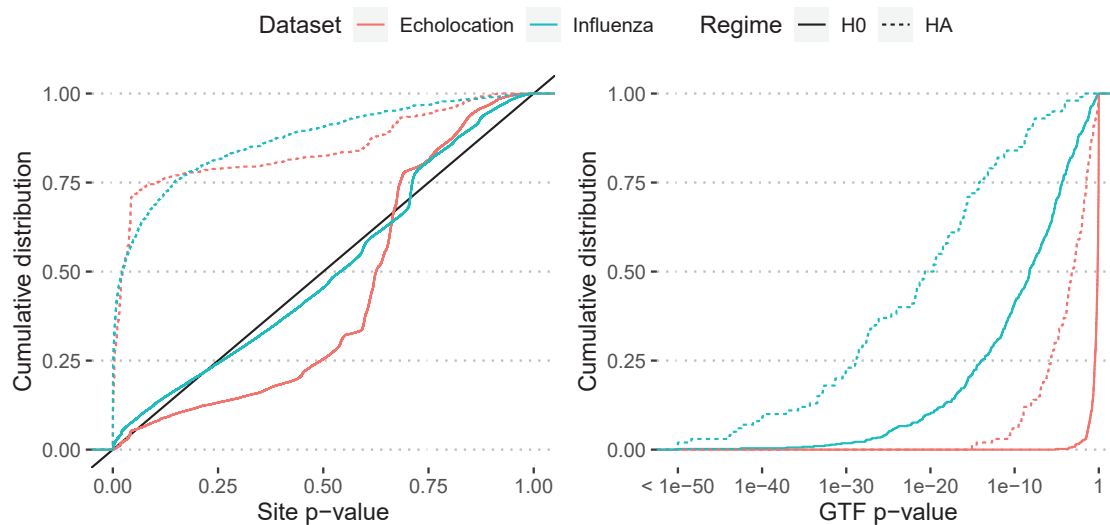


Figure 6.10: Distribution of site p -values (left panel) and gene aggregated p -values (right panel) obtained on the simulated alignments using the Influenza phylogeny. Results on the Echolocation dataset are included for comparison. This highlights that the small enrichment in low p -values at the level of sites translates at the level of genes, resulting in a larger overlap between the distributions of H_0 and H_A genes.

As a first step, let us examine whether the reduced performance results from an overly large number of false positives, or from a situation similar to the relaxed selection experiment where the power of the test is reduced and leads to an increase in the number of false negatives. The left panel of figure 6.10 shows that H_A sites are enriched in low p -values, as expected, and that the distribution of p -values for H_0 sites matches closely the uniform for the Influenza dataset (blue lines). However, if we take a closer look, this distribution under the null has a small but noticeable accumulation of low p -values, which is not present in the case of the echolocation dataset. How does that translate at the level of gene prediction?

The distribution of gene-level p -values, after aggregation of site results using the GTF method, is depicted in the right panel for both the Influenza and Echolocation dataset so that they can be compared. Let us notice that in the echolocation dataset, the overlap between the H_0 and H_A distributions of gene p -values is very limited: the two kinds of genes are well distinguished by the method; in contrast, there is a lot more overlap between the H_0 and H_A distributions on the Influenza dataset. These observations combined suggest that the bad prediction performance on

the Influenza dataset is the result of an enrichment in low p -values among sites, which causes an increased number of false positive gene predictions.

To identify the cause of this enrichment in low p -values under the null, we conducted several experiments that test three hypotheses:

1. the unbalanced annotation of the tree could be the cause of the enrichment in false positive sites. The avian host (background) sub-tree is a lot smaller and shorter than the human host (foreground) sub-tree. A possible consequence is that the foreground sub-tree better reflects the amino acid fitness profile that was used in the simulation under the null, while the background sub-tree carries a more partial signal for the inference of the profile. For example, some amino acids with a relatively low fitness in the simulation profile might not be visible at all in the background sub-tree, even though they are represented in the foreground sub-tree, which induces a difference in the estimated frequency profile for each condition.
2. the distinctive comb-like shape of the Influenza tree, where the length from root to tips increases progressively, could also be a culprit although it is not clear how that would affect the results.
3. Influenza is the only dataset without convergence in its phenotype annotation, and has only one transition occurring at its root. Although we initially dismissed this possibility, as we previously observed that the precision-recall of site predictions was rather insensitive to the number of transitions (see figure 3.5), it remains possible that it may be related to the slight increase in false positive sites that disrupts gene-level predictions.

We tested these hypotheses by building artificial variants of the Influenza tree, which are depicted figure 6.12, and reenacting the simulation-prediction-evaluation pipeline. To test the first hypothesis, we replaced the background sub-tree with a copy of the foreground one: the resulting tree is both large and symmetric. To test the second one, we built an ultrametric version of the tree where all tips are at the same distance from the root. Finally, considering that both features might be the cause of the problem, a phylogeny that is both symmetric and ultrametric was also evaluated. The third hypothesis was tested by changing the annotation of several clades in the foreground sub-tree to background, resulting in a more balanced annotation of the tree and a total of 5 transitions between traits.

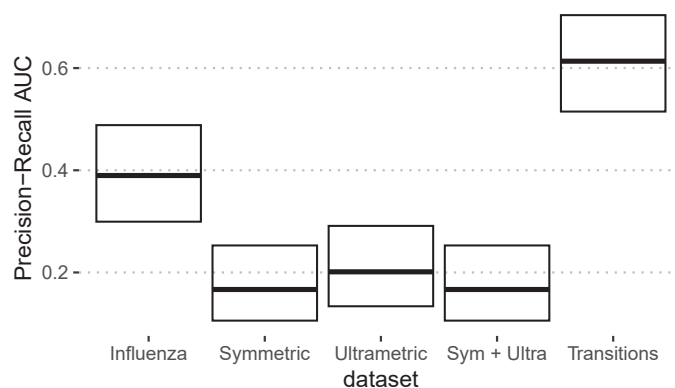


Figure 6.11: Modifications of the tree either decrease the quality of predictions at the level of genes (transformations to symmetric and/or ultrametric tree; hypotheses 1 and 2), or improve them when increasing the number of transitions from 1 to 5, which seems to support our third hypothesis.

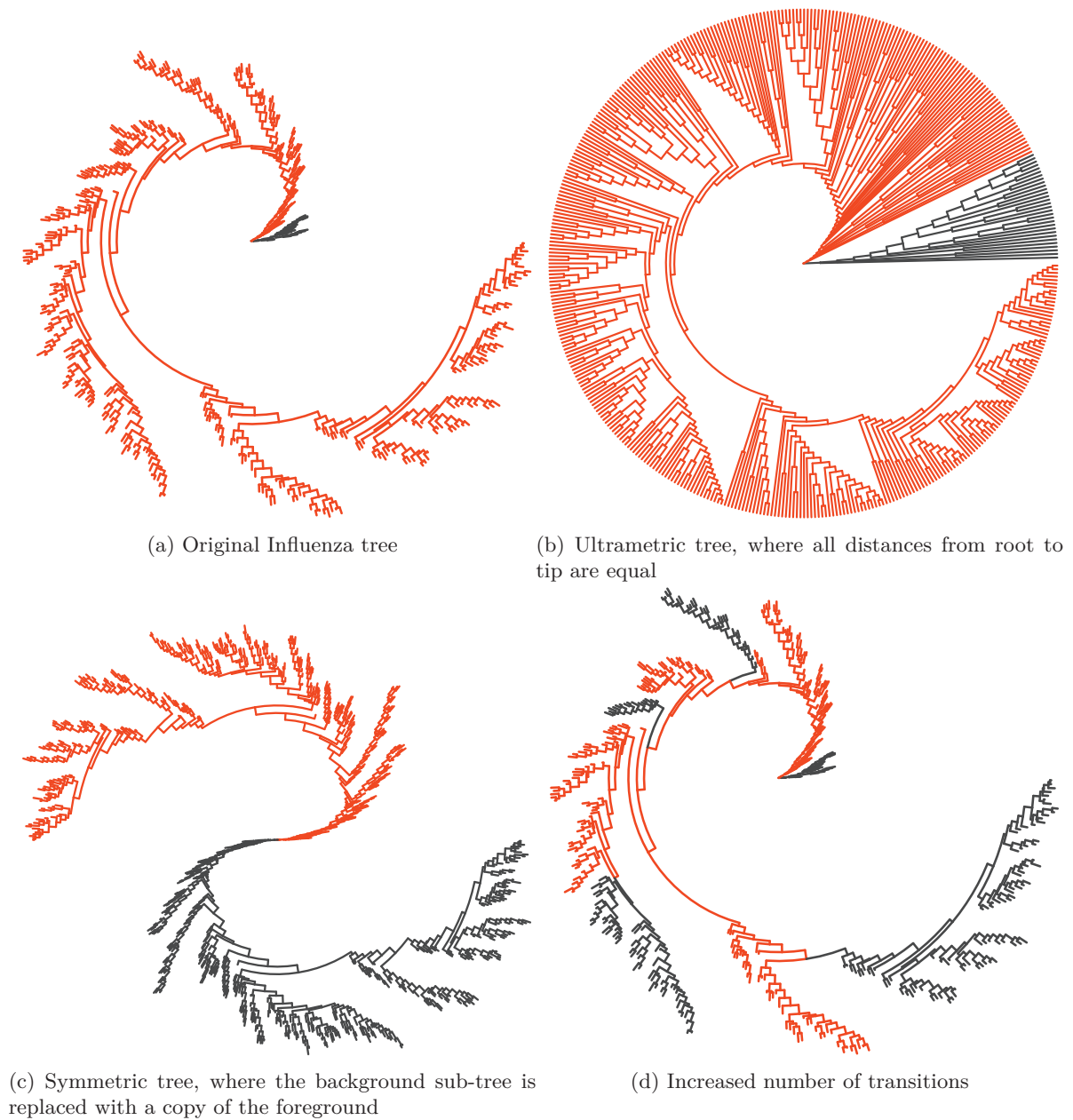


Figure 6.12: Original Influenza tree (a), and modifications of it that were used to investigate the causes of reduced performance in gene-level predictions. Panel (b) is an ultrametric transformation to test the effect of the comb-like shape of the phylogeny. Panel (c) is a change in the topology to investigate whether the issue was the unbalanced annotation of phenotype. Panel (d) introduces additional transition events between phenotypes, to test the hypothesis that the absence of convergence has consequences on the quality of predictions.

We find that gene predictions obtained on simulations using trees with either or both symmetric and ultrametric modifications are even further decreased when compared to those using the original tree, as shown figure 6.11. This suggests that neither of the factors we considered in the first two hypotheses are the cause of the production of false positives. However, precision-recall performance obtained on gene predictions is improved when we introduce additional phenotype transitions.

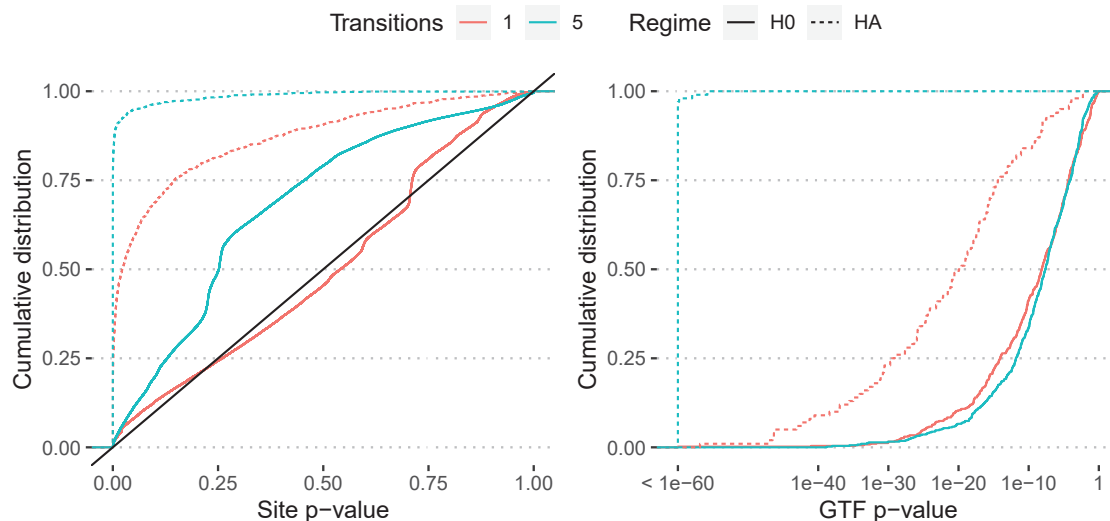


Figure 6.13: Comparison of empirical cumulative distributions of site p -values (left panel) and gene p -values (right panel) between the original dataset with one transition and the modified one with five transitions. The power to reject the null hypothesis is increased with the number of transitions (dashed lines), first at the level of sites, and then translates at the level of genes.

Increasing the number of transitions does not eliminate the false positive predictions at the level of sites, but gives more power to reject the null hypothesis and identify H_A sites, as shown figure 6.13. As a consequence, gene-level p -values better discriminate H_A from H_0 genes. Although the initial observation of the enrichment in low p -values under H_0 is still relevant to understand the causes of the problem, I posit that the more general issue we have with this dataset is a lack of power, due to having a single event of phenotype transition in the tree. This interpretation is consistent with the idea that evidences for adaptation are stronger when they are repeatedly observed through events of convergence.

6.3 How many genes are associated to a given phenotype ?

This question could be answered much like we did at the level of sites within a gene in section 6.1, by setting a threshold that controls the false discovery rate (FDR) using the Benjamini-Hochberg procedure. Again, this builds on the assumption that p -values are calibrated under the null, referring this time to p -values obtained at the level of genes after aggregating site-level p -values. Figure 6.14 illustrates that gene-level p -values are generally not uniformly distributed under the null. However, correcting the site p -values before applying Fisher’s method improves the uniformity of gene p -values under the null (“Corrected” label in figure 6.14), but degrades precision-recall performance (figure 6.3). We therefore choose it as an investigation target, along with the GTF method which provided with the best precision-recall (figure 6.3), for the applicability of the Benjamini-Hochberg procedure on gene-level predictions.

Dataset	Correction + Fisher	GTF
Amaranthaceae	0	0
Cyperaceae	0.00167	0.0233
Echolocation	0	0.0133
HIV	0	0.00167
Influenza	0.245	0.705
Rodents	0	0

Table 6.1: Observed false discovery rate at the gene level for each dataset using either Fisher’s method with p -value correction or the GTF method, at a threshold $\alpha = 0.01$.

Table 6.1 presents the observed proportion of false discoveries in our simulations, when using a target FDR threshold $\alpha = 1\%$. It appears that the procedure is rather conservative, as the observed FDR is generally lower than the threshold. The number of positive predictions is therefore a lower bound on the number of positive genes. However this is not always the case, when the false positive rate is increased, as in results obtained on the Influenza dataset — and Cyperaceae to a lesser extent.

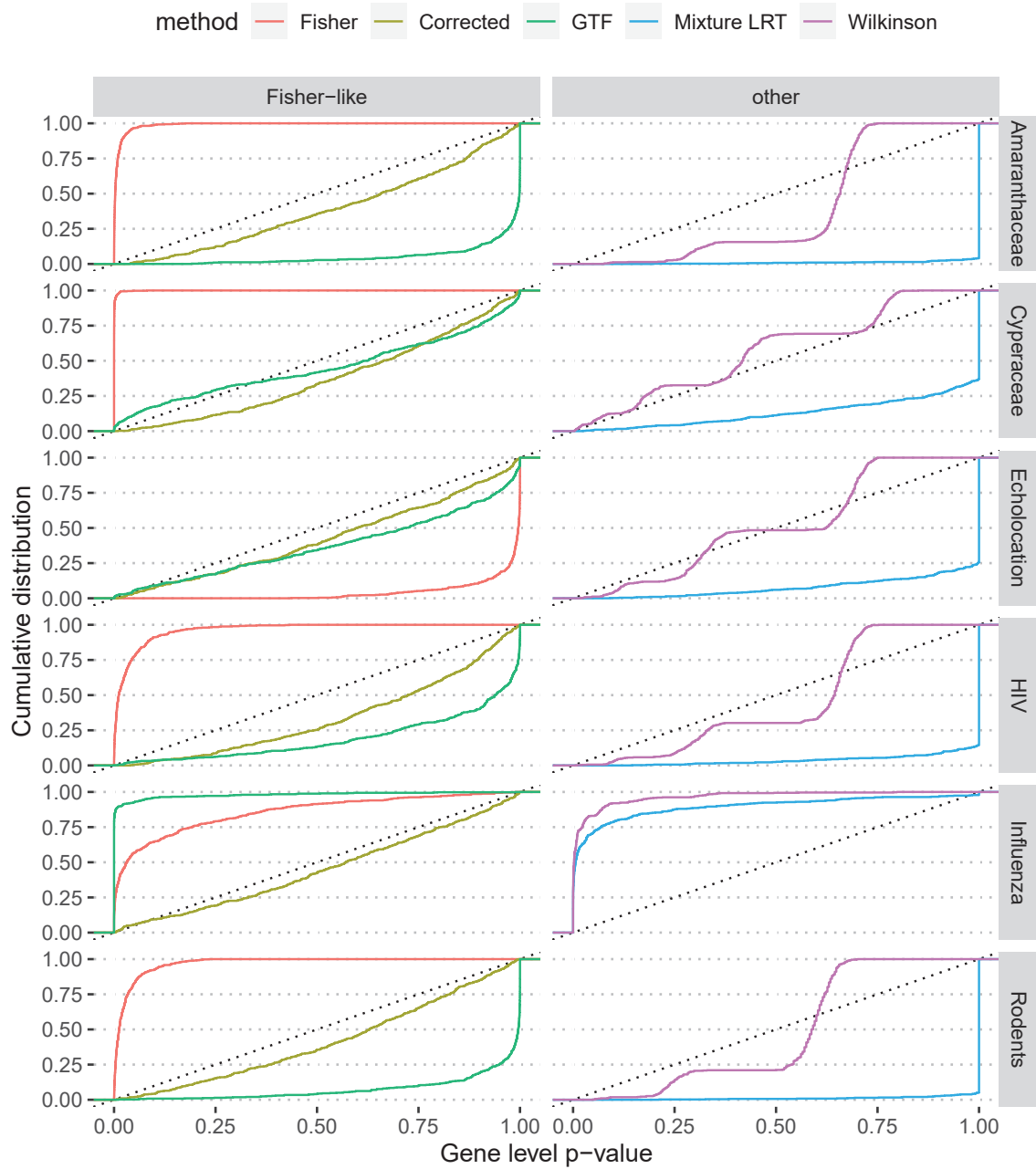


Figure 6.14: Empirical distribution of p -values under the null at the gene level for each aggregation method.

6.4 Detection of genotype-phenotype associations in the Orthomam database

After spending so much effort on validating our methods on simulations, now is (finally!) the time to confront it to empirical data and assess the credibility of results in a realistic setting. For this, we use the Orthomam dataset [Scornavacca et al., 2019]: a carefully reviewed database of 14 509 alignments of coding sequences that spans 116 mammalian species. The corresponding phylogenetic tree is reconstructed at maximum likelihood using the IQ-TREE software [Minh et al., 2020] and the resulting topology is used for all subsequent analyses. We select 8 discrete phenotypic traits as scan targets for genotype associations:

- life in **aquatic**, **marine**, or **subterranean** environments
- **echolocation** using high-frequency vocalizations, that was independently evolved among bats and cetaceans
- diurnal or nocturnal predominance in the **circadian** rhythm
- three-part categorisation of **diet** as herbivore, carnivore or omnivore
- ability to learn and reproduce vocalization (**vocal learning**)
- partition between **domesticated** and wild species

For each phenotype, we manually annotated tips of the tree and reconstructed ancestral traits at maximum parsimony using Fitch’s algorithm. The resulting phylogenies are available in appendix E, and are also displayed in the section for each respective analysis. The duration for a complete scan ranged from 42 to 47 hours per phenotype, using 16 computation cores.

Pelican is run against each trait on every site in the dataset, and gene predictions are performed using the GTF method using the $k = 5$ best p -values obtained for each gene. This method for gene-level aggregation of predictions is described in section 6.2.2. Results presented here include a list of best ranking genes. The corresponding false discovery rate (FDR) values computed using the Benjamini-Hochberg procedure are also provided; the corresponding number of genes below the 0.01 FDR threshold is provided for each dataset. Based on our simulations this estimation of the number of positive genes is expected to be rather conservative. We also conducted gene ontology (GO) enrichment analyses on genes predictions having p -values lower than 0.05, using the package `gprofiler2` [Kolberg et al., 2020] and functional annotations from the Reactome (REAC) database [Fabregat et al., 2018]. We provide for each phenotype a list of REAC terms that are enriched in the corresponding list of genes. In some cases, we also compare our ranking of genes related to a specific REAC pathway between all phenotypes considered, to evaluate whether an enrichment is specific to the phenotype under consideration.

Annotated phylogenies and results obtained from screening Orthomam using Pelican are made available at <https://doi.org/10.5281/zenodo.7501985>. The code repository to download and reproduce the analysis of these results is available at: <https://gitlab.in2p3.fr/phoogle/orthomam-pelican>.

6.4.1 Echolocation

The phylogeny shown figure 6.15 is annotated with the ability to perform echolocation, a trait that is common to some cetacean and bat species. There is an abundant literature on the genomic convergence underlying the independent emergence of echolocation among bats and cetacean, which makes this phenotype a good case study for Pelican and allows us to compare the results we obtain to those previously reported.



Figure 6.15: Orthomam phylogeny annotated with the echolocation trait. See appendix figure E.1 for a larger version including leaf labels.

Table 6.2: Best gene candidates for association to the echolocation trait.

Alignment	p -value	FDR	Functional or adaptive role
PCDH15	3.38×10^{-27}	4.90×10^{-23}	echolocation ^{3,5,7}
ALKAL1	1.84×10^{-17}	1.33×10^{-13}	cytokine ligand
TMC1	6.21×10^{-16}	3.00×10^{-12}	echolocation ^{1,2,3}
LOXHD1	3.33×10^{-15}	1.21×10^{-11}	echolocation ^{1,6}
CDH23	4.81×10^{-14}	1.39×10^{-10}	echolocation ^{3,5}
NKPD1	3.22×10^{-13}	7.78×10^{-10}	NTPase
CA7	1.24×10^{-12}	2.56×10^{-9}	carbonic anhydrase
CHRNA4	1.45×10^{-12}	2.58×10^{-9}	neuronal processing of visual and auditory stimuli ⁴
ZNF536	1.60×10^{-12}	2.58×10^{-9}	neuronal development; vocal learning ⁸
ABTB1	3.47×10^{-12}	5.03×10^{-9}	mediation of protein-protein interactions

¹ [Davies et al., 2012] ² [Dong et al., 2013] ³ [Parker et al., 2013] ⁴ [Espeseth et al., 2007] ⁵ [Shen et al., 2012]

⁶ [McGowen et al., 2020] ⁷ [Davies et al., 2013] ⁸ [Wirthlin et al., 2022]

Table 6.2 shows the best gene candidates that we obtained from the scan of the Orthomam database. Among them, we find that the best ranking genes were already reported to be associated with high-frequency hearing and to have convergently evolved in echolocating mammals. We also find other genes that were not identified to be associated to echolocation in the literature that we know of, but some of them have functional roles that are related to auditory or sensory perception,

and are plausible candidates. In total, 271 genes are predicted to be associated to this trait at a 1% FDR threshold.

Some genes reported to have an adaptive role with regard to echolocation are missing from this list. SLC26A5 (rank 16; $p = 6.21 \times 10^{-11}$) codes for the Prestin protein and is well known for its role in the molecular adaptation to echolocation [Parker et al., 2013, Liu et al., 2010, Li et al., 2010, Shen et al., 2012]. We focus on this gene in the next section, where we compare our predictions at the level of sites to those reported in the literature in the next section. KCNQ4 (rank 149; $p = 1.66 \times 10^{-5}$) [Liu et al., 2011], PJVK (rank 297; $p = 2.76 \times 10^{-4}$) [Davies et al., 2012] and OTOF (rank 304; $p = 3.08 \times 10^{-4}$) [Parker et al., 2013] are among the top 1% genes in our predictions. In contrast, OTOS and OTOG [Dong et al., 2013, Shen et al., 2012] rank 4366 and 13205, and were not identified as positive genes in our analyses. This can be explained by the fact that their adaptive role for echolocation was mediated by an increase in their expression level, instead of modifications in their sequence [Dong et al., 2013]. It is a typical example of adaptation that can not be uncovered using Pelican.

Table 6.3: Genes annotated with a functional role in auditory perception are over-represented among genes associated to the echolocation phenotype.

REAC ID	Term name	p -value	
R-HSA-9659379	Sensory processing of sound	6.04×10^{-8}	***
R-HSA-9662360	Sensory processing of sound, inner hair cells of the cochlea	1.33×10^{-7}	***
R-HSA-9662361	Sensory processing of sound, outer hair cells of the cochlea	1.45×10^{-5}	***
R-HSA-9709957	Sensory Perception	4.18×10^{-5}	***
R-HSA-3000471	Scavenging by Class B Receptors	3.58×10^{-2}	*
R-HSA-9603505	NTRK3 as a dependence receptor	3.89×10^{-2}	*
R-HSA-8964058	HDL remodeling	1.03×10^{-1}	
R-HSA-187015	Activation of TRKA receptors	1.32×10^{-1}	

*** $p < 0.001$ ** $p < 0.01$ * $p < 0.05$

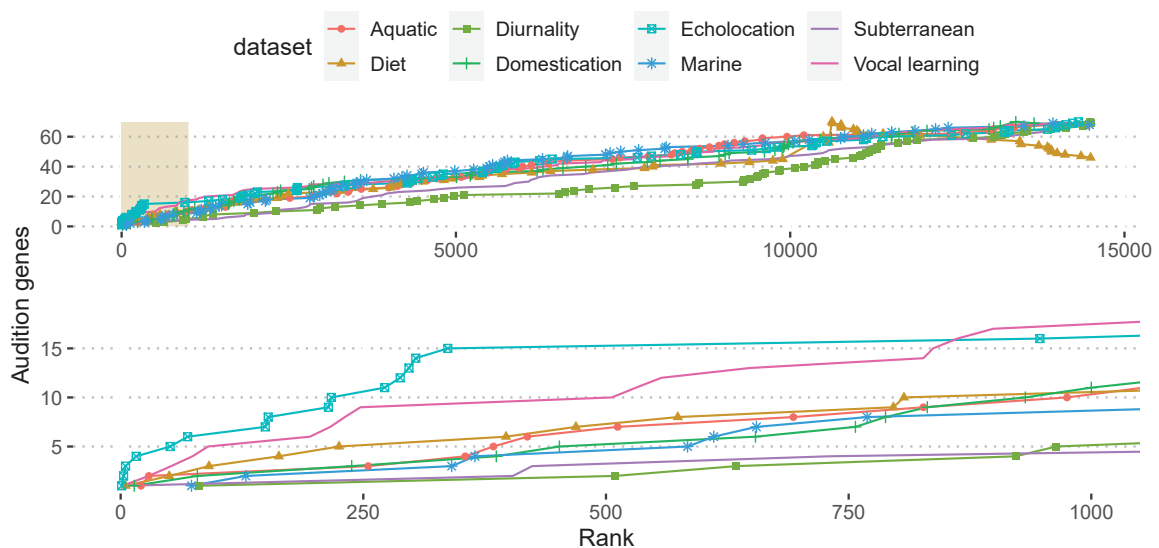


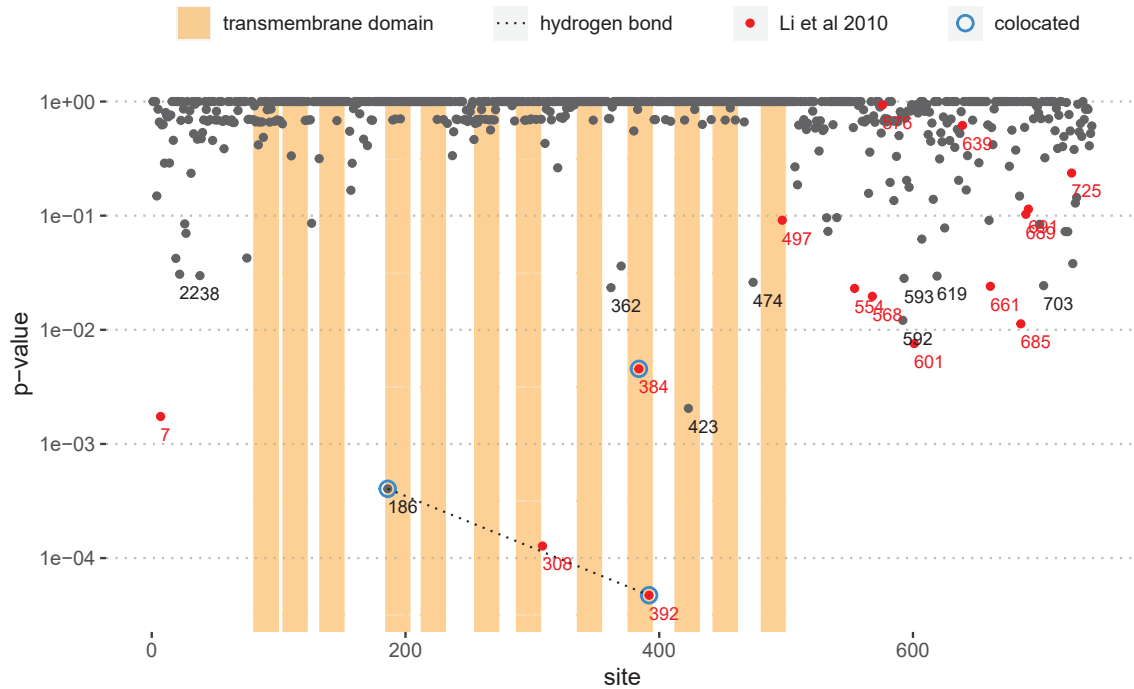
Figure 6.16: Best ranking genes predicted to be associated to echolocation have functional roles in auditory processing. Lower panel is a zoom over the colored area in top panel showing the complete ranking, to highlight the best ranking genes.

GO term enrichment analysis of the ranked list of genes shows that it is strongly enriched in terms related to auditory processing pathways from the REAC database, as shown table 6.3. Comparing the enrichment in genes related to the sound processing pathway across all phenotypes included in our analysis, we find that the echolocation phenotype has a stronger representation among its best ranking genes, as shown figure 6.16.

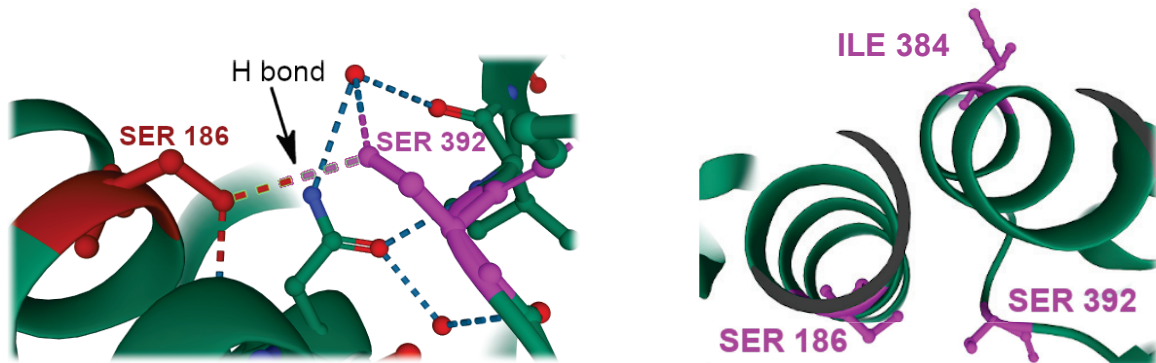
Focus on the Prestin gene (SLC26A5)

We were particularly expecting the presence of the Prestin gene (SLC26A5), which comes at position 16 in our list of predictions, and codes for a transmembrane protein expressed in the inner ear of the mammalian cochlea. Its specific function is motility: using the electric force released from transfers of anions across the membrane, it moves outer hair cells which act as sound amplifiers. There are multiple reports in the literature of this gene having undergone convergent substitutions in echolocating bats and cetaceans [Parker et al., 2013, Shen et al., 2012, Li et al., 2010, Liu et al., 2010]. This is comforting for the validity of our approach, but it also gives us an example of a well-studied gene that can be used to confront our findings to the existing literature. Specifically, we compare the positions that we find associated to the echolocation trait to those reported in [Li et al., 2010], where a list of positions was identified by comparing the phylogeny of mammals to one that is inferred using only an alignment of Prestin. Figure 6.17 shows the p -value for each site in the Prestin, obtained from the application of Pelican. Comparing our results to those of [Li et al., 2010], we find that most sites that have the lowest p -values were also detected positive in their study. The site at position 7 was reported to be involved in the dimerization of the Prestin. Positions 308, 384, 392 and 497 are located in transmembrane domains, and the other highlighted sites are part of a so-called STAS domain that interacts with enzymes transporting anions.

The third “best” site that we detect at position 186 was not identified in the study we are comparing to. However, looking at its location in the 3D structure of the human Prestin (PDB: 7LGU), we find that the residue at this site interacts through a hydrogen bond with the one at position 392, for which there is evidence in the literature. The identification of this position as associated to the function of Prestin for echolocation is thus consistent with the rest of our findings, since it is plausible that one of these two substitutions did occur as an adaptation to the other one. We also notice that the residue at position 384 is located in the same area in the tertiary structure of the protein, although it does not interact with the other two through a hydrogen bond. Nonetheless this may suggest that the conformation of this region is modified in echolocating mammals compared to what it is in the human Prestin structure. Site 423 also exhibits a strong signal according to our results, and is located in a transmembrane domain, but was not identified in the publication we used as reference.



(a) Comparison of site predictions obtained with Pelican on the *Prestin* gene to those reported in [Liu et al., 2010]. A hydrogen interaction between sites 186 and 392 is highlighted, suggesting that they may be co-evolving.



(b) Residues 186 and 392 interact through a hydrogen bond in the human *Prestin*.

(c) Residues 186, 384 and 392 are located in the same area in the tertiary structure.

Figure 6.17: Predictions of sites within the *Prestin* that are associated to the echolocation trait are in good agreement with the literature. The two sites having the strongest signal are in interaction through a hydrogen bond, according to the 3D structure of the human *Prestin* (PDB ID: 7LGU).

6.4.2 Diet

The phylogeny depicted figure 6.18 is annotated with alimentary diets, that fall into three possible categories: herbivory, carnivory or omnivory.



Figure 6.18: Orthomam phylogeny annotated with the diet trait. See appendix figure E.2 for a larger version including leaf labels.

We find that the best ranking genes code for proteins having roles in the digestive process, in particular the metabolism and assimilation of lipids and proteins, as shown table 6.4. Some of them have already been reported to have evolved in association with the herbivore or carnivore diet among some mammal species [Wu, 2022]. Only 2 gene predictions pass the 1% FDR threshold. Gene ontology analysis of our list of genes reveals that it is enriched in genes having functional roles in the digestion, metabolism and absorption of nutrients. Looking at the ranks of genes having functions related to digestion, we find that association to the diet phenotype is over-represented among the best ranking genes compared to other phenotypes, as shown figure 6.19. This strong enrichment in genes associated to digestion might suggest that for this dataset, the number of positive genes estimates at the 1% FDR threshold is overly conservative, and might be related to a lack of power resulting from the tripartite categorization of the phenotype.

Table 6.4: Best gene candidates for association to diet (herbivore, carnivore, omnivore) trait.

Alignment	p -value	FDR	Functional or adaptive role
PNLIP	1.93×10^{-12}	2.80×10^{-8}	pancreatic lipase, carnivory ¹
ENPEP	1.44×10^{-8}	1.05×10^{-4}	protein digestion
CPB1	1.94×10^{-5}	8.17×10^{-2}	protein digestion, carnivory ¹
CPA1	2.25×10^{-5}	8.17×10^{-2}	protein digestion, carnivory ¹
OTOF	9.13×10^{-5}	2.65×10^{-1}	audition
CEL	8.04×10^{-4}	1.00	cholesterol and vitamin assimilation
SLC6A12	1.66×10^{-3}	1.00	GABA transporter
ACSS3	3.41×10^{-3}	1.00	fatty acid metabolism
PLA2G1B	1.31×10^{-2}	1.00	phospholipase, fatty acid digestion ¹
RAPGEF2	1.88×10^{-2}	1.00	cell growth and differentiation

¹ [Wu, 2022]

Table 6.5: Predictions associated to diet are enriched in genes having functional roles in digestion.

Term ID	Term name	p -value	
R-HSA-192456	Digestion of dietary lipid	4.5×10^{-4}	***
R-HSA-2022377	Metabolism of Angiotensinogen to Angiotensins	9.2×10^{-4}	***
R-HSA-8935690	Digestion	6.9×10^{-3}	**
R-HSA-8963743	Digestion and absorption	9.7×10^{-3}	**
R-HSA-2980736	Peptide hormone metabolism	2.4×10^{-2}	*
R-HSA-888593	Reuptake of GABA	4.1×10^{-1}	
R-HSA-975634	Retinoid metabolism and transport	6.4×10^{-1}	
R-HSA-6806667	Metabolism of fat-soluble vitamins	7.0×10^{-1}	

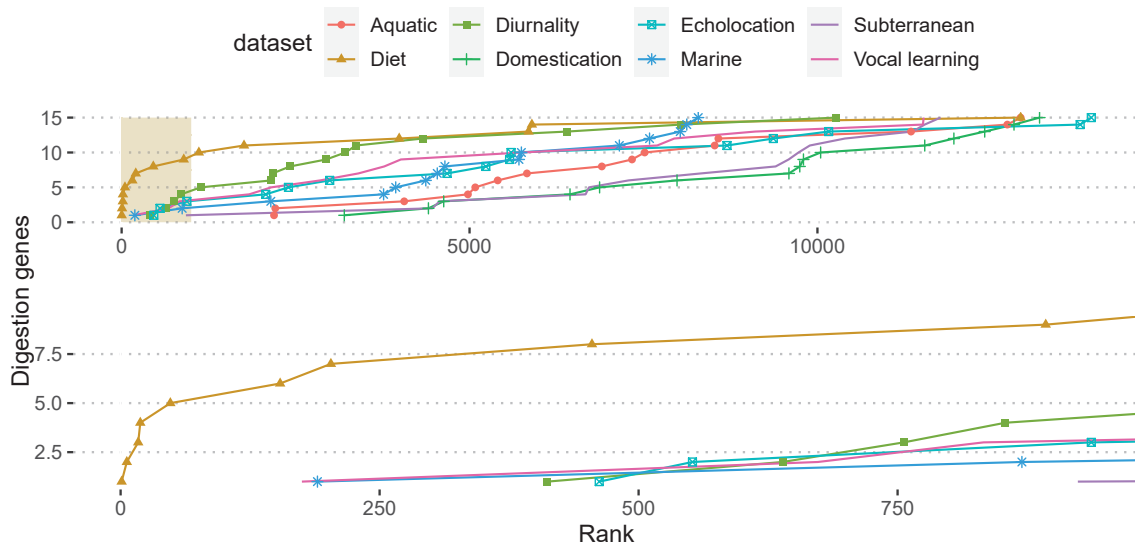
* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$ 

Figure 6.19: Genes having functions related to digestion have higher ranking in the list of predictions for the diet traits than for other phenotypes. Lower panel is a zoom over the colored area in top panel showing the complete ranking, to highlight the best ranking genes.

6.4.3 Aquatic and marine

Annotations of the phylogeny using marine or aquatic traits are quite similar as shown figure 6.20, and the results we obtain from screening the Orthomam database for genotype associations are thus unsurprisingly similar between the two traits.

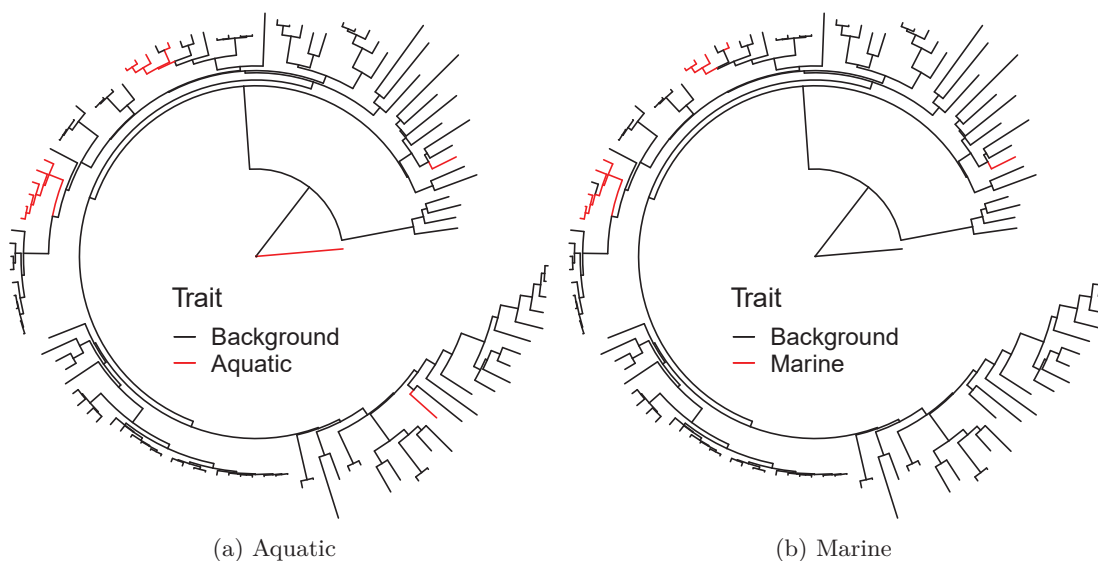


Figure 6.20: Orthomam phylogeny annotated with the aquatic or marine traits. See appendix figures E.3 and E.4 for larger versions including leaf labels.

The list of best ranking genes regarding the adaptation to life in marine environments is presented table 6.6. Those relative to adaptation to life in aquatic environments have less obvious interpretations to me, although some are present in both lists and have functional roles consistent with our expectations; for these reasons and for the sake of brevity, the list of best ranking aquatic genes is shown separately in appendix (table E.6). In total, we find 72 genes related to marine adaptations under the 1% FDR threshold, and 17 genes related to aquatic adaptations. This suggests that the marine annotation might give more power to identify gene associations, as it is more specific than the aquatic annotation.

Alignment	Rank aquatic	p -value	FDR	Functional or adaptive role
SULT1C3	2	1.15×10^{-18}	1.67×10^{-14}	Sulfation of polysaccharides; aquatic adaptation [Hettle et al., 2018]
TGM1	16	1.08×10^{-11}	7.81×10^{-8}	Structural role in the formation of epidermis
KRT80	10	3.08×10^{-10}	1.49×10^{-6}	Keratinization
TRPV3	35	4.19×10^{-10}	1.52×10^{-6}	Sensory perception of temperature; hair formation [Imura et al., 2007]
PERP	30	6.47×10^{-10}	1.87×10^{-6}	Epithelial development [Ihrie et al., 2005]
TUBA3C	4	1.59×10^{-9}	3.84×10^{-6}	Formation of microtubules
MYL1	37	2.35×10^{-9}	4.86×10^{-6}	Muscle contraction [Zhou et al., 2015]
KRT4	5	9.70×10^{-9}	1.76×10^{-5}	Keratinization
VSIG8	25	2.01×10^{-8}	3.02×10^{-5}	Immunoglobulin
S100A5	7	2.19×10^{-8}	3.02×10^{-5}	Calcium-binding protein

Table 6.6: Best ranking genes predicted for association with life in marine environments.

SULT1C3 is the best ranking gene in the list of marine genes, and the second best in the list

of aquatic genes. It encodes a sulfotransferase protein, whose relation with aquatic living is rather obscure at first. However, the introduction of [Hettle et al., 2018] summarizes the important role of sulfation to “regulate the physicochemical properties of the polysaccharide structures, such as gelling and flexibility, thereby enabling adaptation of these polymers to specialized roles in the aquatic environment [Kloareg and Quatrano, 1988]. The common use of sulfate modifications in highly abundant marine polysaccharides suggests this biomass is one of the largest reservoirs of sulfated biomolecules. As a consequence, many marine microbes have acquired the metabolic machinery that allows them to utilize this carbon source [...]. The presence of sulfatases in these marine microbes likely allows them to remove sulfate modifications from these complex polysaccharides, thereby enabling their complete depolymerization and eventual metabolism”. Our identification of this gene is thus consistent with this literature, and suggests that substitutions in SULT1C3 could have played an analogous role in the adaptation of mammals to life in aquatic environment, by either allowing structural modifications of polysaccharides, or their metabolism, or both.

Looking at other genes in our list, we find that a large proportion of them have roles related to skin and hair formation. This observation on the 10 best ranking genes is corroborated by a more formal analysis of gene ontology terms, which reveals an enrichment in functions related to the development of skin (cornification) and hair (keratinization) in the lists of predictions for both phenotypes (aquatic or marine) as presented in tables 6.7 and 6.8. Moreover, predictions on both of these traits include more genes related to the keratinization pathway in their best ranking ones than other phenotypes, as shown figure 6.21. These results are consistent with reports of convergent adaptation in marine mammals in [Chikina et al., 2016], who also identify genes related to these functions. Adaptation of marine mammals is also reported to have left genomic signatures in the α -keratin genes [Sun et al., 2017], although we do not identify as many genes from this family, possibly because Pelican lacks power to detect relaxed selection.

Finally, we notice that the first GO term that is enriched in the predictions from the marine phenotype is *Muscle contraction*, which is corroborated by reports of convergent adaptation in several genes associated to synaptic transmission and muscle contraction in [Zhou et al., 2015].

Table 6.7: Functional enrichments for gene predictions associated to life in aquatic environments.

Term ID	Term name	p -value	
R-HSA-6809371	Formation of the cornified envelope	1.4×10^{-4}	***
R-HSA-1266738	Developmental Biology	2.2×10^{-3}	**
R-HSA-6805567	Keratinization	4.4×10^{-3}	**
R-HSA-112316	Neuronal System	2.7×10^{-2}	*
R-HSA-9619483	Activation of AMPK downstream of NMDARs	4.1×10^{-2}	*
R-HSA-9663891	Selective autophagy	4.6×10^{-2}	*
R-HSA-1296072	Voltage gated Potassium channels	9.1×10^{-2}	
R-HSA-397014	Muscle contraction	9.3×10^{-2}	

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Table 6.8: Functional enrichments for gene predictions associated to life in marine environments.

Term ID	Term name	p -value	
R-HSA-397014	Muscle contraction	4.7×10^{-5}	***
R-HSA-6809371	Formation of the cornified envelope	1.9×10^{-4}	***
R-HSA-390522	Striated Muscle Contraction	4.5×10^{-4}	***
R-HSA-6805567	Keratinization	7.1×10^{-4}	***
R-HSA-9709957	Sensory Perception	1.7×10^{-2}	*
R-HSA-1266738	Developmental Biology	6.0×10^{-2}	
R-HSA-9619483	Activation of AMPK downstream of NMDARs	8.5×10^{-2}	
R-HSA-217271	FMO oxidises nucleophiles	9.3×10^{-2}	

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

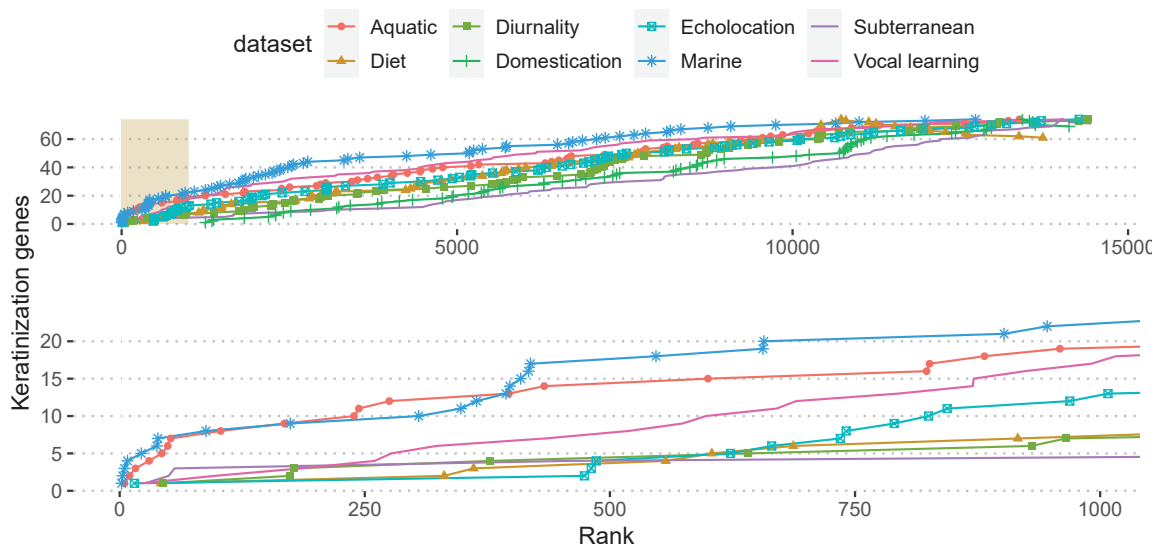


Figure 6.21: Best ranking predictions of genes associated to life in aquatic and/or marine environments are enriched in functions related to keratinization (e.g. hair formation).

6.4.4 Subterranean

We screen mammal genomes for adaptations to life in subterranean environments, using the Orthomam phylogeny with annotations displayed figure 6.22, and present our list of best ranking genes in table 6.9. Among our predictions, 10 of them pass the 1% FDR threshold.

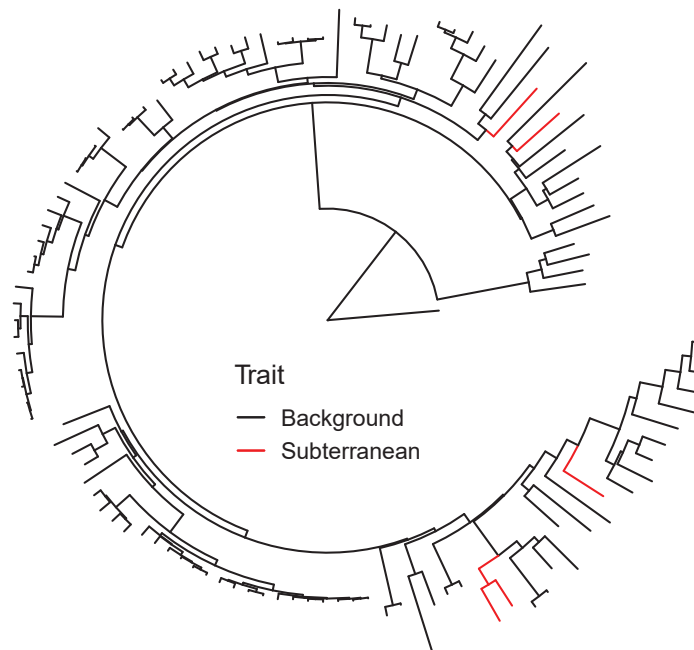


Figure 6.22: Orthomam phylogeny annotated with the subterranean trait. See appendix figure E.5 for a larger version including leaf labels.

Table 6.9: Best ranking predictions associated to the subterranean living phenotype.

Alignment	p -value	FDR	Functional or adaptive role
MITD1	9.56×10^{-9}	1.39×10^{-4}	Mitosis and cell differentiation
CRYBA1	2.57×10^{-7}	1.86×10^{-3}	Crystallin ^{1*}
ALAD	3.89×10^{-7}	1.88×10^{-3}	Heme synthesis
GNAT1	3.37×10^{-6}	8.58×10^{-3}	Visual transduction ^{1*}
TMIE	3.57×10^{-6}	8.58×10^{-3}	Auditory perception
CRYBB3	3.78×10^{-6}	8.58×10^{-3}	Crystallin ^{1*}
CRYGC	4.15×10^{-6}	8.58×10^{-3}	Crystallin ^{1*}
RPE65	5.86×10^{-6}	9.34×10^{-3}	Retin [*]
GLRA1	6.06×10^{-6}	9.34×10^{-3}	Nervous system, startle reflexes
NUP98	6.45×10^{-6}	9.34×10^{-3}	Nuclear pore

¹ [Partha et al., 2017] * Related to vision

We find several genes coding for proteins of the crystallin family that are essential components of the eyes of mammals, and more generally genes related to visual perception. Genes involved with visual perception have previously been reported to have undergone convergent regression among underground-dwelling mammals [Partha et al., 2017], consistently with our results. However, genes related to vision are generally higher ranking in the list established by [Partha et al., 2017] than in ours. This is most probably due to the fact that the change of evolutionary dynamics associated to subterranean living for these genes is mostly, if not exclusively, relaxation of the selective pressure

and loss of function. Indeed, the pressure for maintaining functional eyes is less stringent in dark underground environments. This illustrates how approaches based on detecting changes in the substitution rate, such as what is done in [Partha et al., 2017], are more powerful than profile based methods such as Pelican to detect relaxation of selective pressure. This was already suggested by our comparison of detection performance using simulations under relaxed selection in chapter three (figure 3.5). However, this also shows that we still manage to detect some of the genes under relaxed selection using Pelican, although we lack power to assemble a more exhaustive list. Results of gene ontology term analysis presented in table 6.10 display an enrichment, albeit weak, in our list of genes for terms associated to visual perception. This enrichment in vision-related terms is stronger in the subterranean dataset than in any other dataset that we analyzed, as shown figure 6.23.

Table 6.10: Functional enrichments for gene predictions associated to subterranean adaptation.

REAC ID	Term name	p -value
R-HSA-2514859	Inactivation, recovery and regulation of the phototransduction cascade	1.6×10^{-2} *
R-HSA-2514856	The phototransduction cascade	1.8×10^{-2} *
R-HSA-2187338	Visual phototransduction	2.3×10^{-2} *
R-HSA-2485179	Activation of the phototransduction cascade	4.7×10^{-2} *
R-HSA-9709957	Sensory Perception	1.2×10^{-1}
R-HSA-9009391	Extra-nuclear estrogen signaling	1.3×10^{-1}
R-HSA-8939211	ESR-mediated signaling	8.1×10^{-1}
R-HSA-432040	Vasopressin regulates renal water homeostasis via Aquaporins	1.0

* $p < 0.05$

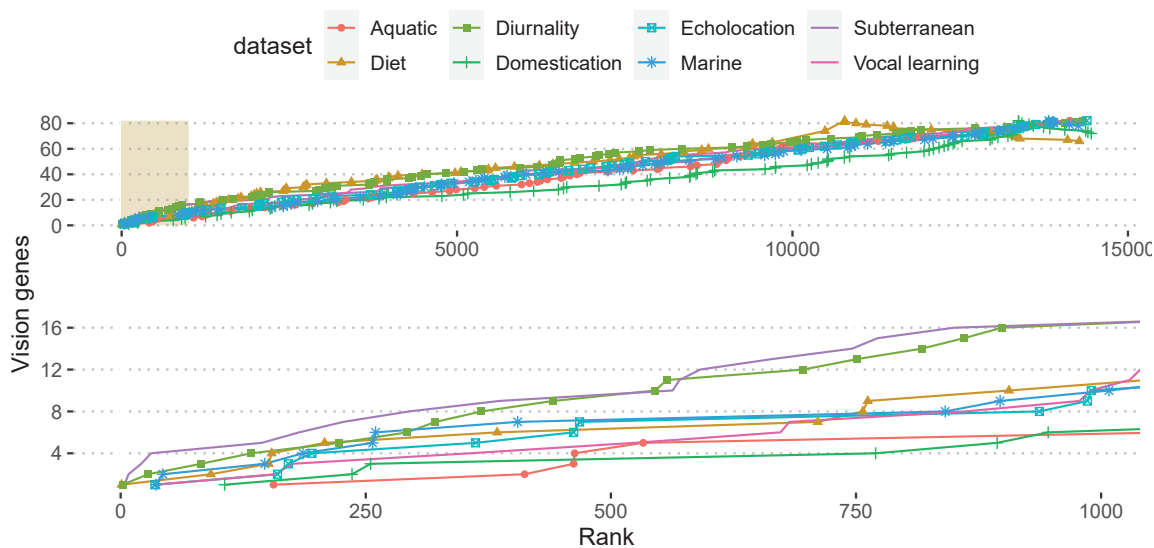


Figure 6.23: Best ranking predictions of genes associated to life in subterranean environments are weakly enriched in functions related to visual perception.

Other predictions in our list have less obvious interpretations, and we did not find any literature that could explain a possible adaptive role to subterranean living. One could speculate that ALAD, related to heme synthesis, could improve O_2 binding in oxygen-deprived underground environments; or that TMIE, related to auditory perception, could suggest a sensory adaptation to compensate for reduced visual acuity; however these are merely hypotheses which should be individually tested. It

is also possible that our list of predictions contains more false positives than we obtained with other phenotypes, as the quantity of subterranean species in the tree is limited to five. We suspect this could favor the existence of sites where the amino acid composition differs between trait conditions, as a consequence of random sampling effects — the importance of such unbalance in the annotation is not clear yet. The lack of GO enrichment for terms other than vision-related ones seems to corroborate this interpretation.

6.4.5 Other phenotypes

Our results on the last three phenotypes that we considered are less striking, possibly because of the choice of the phenotype themselves. Although I do not present these results in detail here, they are available in appendices.

Diurnal/nocturnal

Gene predictions associated to the diurnal/nocturnal annotation do not exhibit a clear pattern of adaptation, however the first and second best ranking genes have functional roles that could be related to adaptation to nocturnal life. The first gene is PITPNC1, and has a key role in the generation of heat from brown adipose tissue [Tang et al., 2022]; it might be that substitutions in this gene could improve the adaptation of nocturnal species to lower night-time temperatures, although we did not find literature that corroborate this hypothesis. The second one is RHO, and codes for a rhodopsin protein which is involved in vision in dimly-lit environments [Sugawara et al., 2010]. Other genes in the list have less clear functional interpretations with regard to this phenotype, and it is possible that the definition of the trait that we used here is not specific enough to discover a larger set of genes. We report 4 gene predictions under the 1% FDR threshold. GO analysis did not present significant enrichment for any term. The annotated phylogeny, along with the list of best ranking genes and GO enrichment results are available in appendix section E.6.

Vocal learning

Vocal learning is the ability to learn and reproduce vocalisations, a trait that is shared by several mammal groups, mainly primates, cetaceans, pinnipeds and bats [Janik and Knörnschild, 2021]. However the underlying mechanisms for the production of vocalisations may strongly differ between these groups, e.g. rely on different organs and have widely variable acoustic signatures. Vocal learning is also better represented as a spectrum than as a clear separation between vocal learners and non-learners. With that in mind, we nonetheless attempted to search for some common molecular bases shared across vocal learning mammals that are considered as a homogeneous group. The list of best ranking genes displays some genes associated to auditory perception like PCDH15 and LOXHD1. The first gene codes for an olfactory receptor, which is a bit puzzling. In the literature, FOXP2 is the first gene that was reported to be involved in language development in humans [Webb and Zhang, 2005]; however this gene comes only in position 859 ($p = 4.76 \times 10^{-2}$) in our list. However, ZNF536 (rank 40; $p = 2.52 \times 10^{-7}$) ranks higher in the list, and is corroborated by experimental evidence [Wirthlin et al., 2022]. Looking at the GO enrichment of terms in our list, we find that it is significantly enriched in genes having functions in sound processing and sensory perception. We suspect that these results may be steered by the fact that the majority of vocal learning species are also echolocators, which may act as a confounding factor in this analysis. Our predictions include 133 genes under the 1% FDR threshold. The annotated phylogeny, along with the list of best ranking genes and GO enrichment results are available in appendix section E.7.

Domestication

The best ranking gene in our list of predictions is DNAJB1, and codes for a heat-shock protein (HSP) that is involved in the response to temperature variation. This finding can be related to existing literature on the importance of heat stress on the growth, reproduction and milk production of livestock [Archana et al., 2017]. The next two genes are involved in glycogen metabolism and

response to stress through corticoid signaling, which might be related to domestication — although the link is not clear and quite speculative as there is no literature to support this. Other genes in the best ranking list are more difficult to relate to the trait under consideration. In total, 46 gene predictions pass the 1% FDR threshold. GO enrichment analysis provides for a large set of significantly enriched terms, but that do not have clear ties with the domesticated trait. It is likely that this phenotype annotation mixes domesticated species that were not selected for the same set of traits, e.g. the desired traits are widely different between dogs and cattle. The annotated phylogeny, along with the list of best ranking genes and GO enrichment results are available in appendix section [E.8](#).

In this chapter, I presented our explorations to perform inference at the level of genes using predictions made with Pelican on sites within simulated genes. First, we used the Benjamini-Hochberg procedure to estimate the number of positive sites within genes while controlling the false discovery rate (FDR). Although Pelican is not calibrated, we empirically confirm that on average the proportion of false positives matches the expectation or is more conservative (figure 6.2).

We then investigated several methods to aggregate site-level p -values to produce gene-level ones, and compare the precision-recall of their predictions on genes (figure 6.3). Wilkinson's method based on a binomial distribution had good precision-recall, but was too sensitive to a significance threshold that has to be chosen on site p -values. Fisher's method relies on the hypothesis that p -values are uniformly distributed under the null, which is not true in our case. We compared its performance to a variant where site p -values are corrected empirically beforehand. Empirical correction of p -values only marginally improves the performance, but noticeably improves the calibration of gene-level predictions (figure 6.14). We also considered another variation on Fisher's method that focuses on the distribution of the lowest p -values, and that we name the Genewise Truncated Fisher (GTF) method. This approach has improved precision-recall compared to Fisher's method and has more power to distinguish positive from negative genes. As an alternative method, we designed a mixture model of site p -values, that distinguishes their distribution between positive and negative sites. It is fitted at maximum likelihood using an efficient expectation-maximisation algorithm. Its predictive performance was however not on par with the GTF method, which remains our best candidate so far.

In the results of our benchmark, predictions were degraded on simulations conducted on the Influenza phylogeny. We highlighted that this bad performance was caused by a small accumulation of low site p -values under the null, despite that their distribution is quite close to the uniform. This led to a large amount of false positives, i.e. negative genes that are not well distinguished from positive ones. To better understand this issue, we conducted several experiments using modified versions of the Influenza phylogeny (figure 6.12). Our results suggest that the main problem could be a lack of power of Pelican on this dataset due to it having only one transition between phenotypes, unlike the others (figure 6.13). This interpretation would be consistent with the expectation that cases of molecular convergence provide stronger evidence for directional selection, due to repetitions in the pattern of substitution.

The second part of this chapter focuses on an analysis of empirical data from the Orthomam database. The dataset consists in 14508 coding sequence alignments for 116 species of mammals, in which we searched for genes associated to several discrete phenotypes: echolocation, diet, aquatic and marine life, subterranean life, diurnal/nocturnal life, vocal learning, and domestication. We reported our predictions of genes associated to each trait using Pelican combined to the GTF method for gene aggregation. We compared them to the literature, to find that our results generally had some level of agreement with it. We also compared our predictions of sites within the Prestin gene associated to the echolocation trait, and found them to be corroborated by previous studies, and to be consistent when confronted to the

tertiary structure of the protein. The support for our finding was not equally good across traits, which could be possibly explained by either an improper choice or annotation of some phenotypes, shortcomings inherent to our methodology, or incomplete literature on the subject. However, our approach appears to produce convincing results for most of the phenotypic traits that we considered, and is further validated by gene ontology (GO) enrichment analysis of our list of predictions.

Chapter 7

Relevance and limitations of the approach

In this chapter, I intent to take a step back and try to draw conclusions from all the results that have been presented so far in this work. I think that at least two elements naturally appear as important points to be discussed: the application scope for Pelican, i.e. identify the main characteristics of settings in which Pelican is expected to give the best results, and those where more precautions should be taken; then have some hindsight on the results obtained from Pelican, both at the level of sites and genes, and the kind of biological interpretations that we can draw from them. This is also the occasion to discuss with more depth the hypotheses that are made in the model of TDG09 and Pelican, with regard to actual biological processes that they aim to represent. I address as well the question of inputs, specifically the uncertainty that they convey, and a few ideas on how to account for it in some cases.

I strive throughout this chapter to identify general guidelines for achieving good predictions using this tool, “good” in the sense that they should give reliable biological insights. Accounting for the limitations that are identified throughout this chapter and this thesis as a whole, I propose some possible improvements on Pelican to make the method either faster, more robust, more accurate, or better adapted to model a wider diversity of situations. Some other research perspectives are also given, that stray further away from the original model and could be fruitful for a better identification or understanding of relations between genotypes and phenotypes.

Contents

7.1	What problem are we solving ?	164
7.2	On the interpretation of Pelican results	165
7.2.1	Pelican identifies sites whose evolution <i>correlates</i> to variations of the phenotype	165
7.2.2	On the interpretation of estimated parameters	166
7.2.3	On the choice of the phenotypic trait	166
7.2.4	The issue of calibration	167
7.3	On input data, and uncertainties	168
7.3.1	Uncertainties in the sequences, and their use as representations of the genotype of species	168
7.3.2	Uncertainties in the alignment	169
7.3.3	Uncertainties in phylogenetic trees	169
7.3.4	Uncertainties in the phenotype	170
7.4	Hypotheses of the Pelican model, and their biological implications	171
7.4.1	Adaptation happens through coding sequences modifications	171
7.4.2	Sites evolve independently from each other	171
7.4.3	Similar traits involve similar genes across species	172
7.4.4	Modeling substitutions at the level of amino acids	172
7.5	Perspectives	173

7.1 What problem are we solving ?

We propose Pelican as a method to detect genotype-phenotype associations, which is based on a model of amino acid preferences that drive the evolution of protein sites. It is simple and fast enough to enable the analysis of genome scale datasets in a timely manner. However, it must be clarified that we do not claim that it is the fastest method available, as we did not perform an exhaustive benchmark of every available model implementations. Specifically, alternative implementations of the underlying model of PAML, such as FastCodeML [Valle et al., 2014] or SlimCodeML [Schabauer et al., 2012], might compare favorably to our proposed method in terms of speed.

We do claim nonetheless that Pelican fills a gap in proposing a readily applicable implementation of a profile-based evolution model [Tamuri et al., 2009, Tamuri et al., 2012, Parto and Lartillot, 2017]. Existing profile methods are computationally costly, and difficult to apply at the genome scale, which is enabled by Pelican. The models underlying profile methods rely on a different approach to detect variation of the selection dynamics compared to d_N/d_S models: instead of comparing substitution rates, they infer amino acid preferences — either as fitness or frequency vectors — which gives an estimation of the *direction* of the selection pressure. Although we shall be cautious regarding the confidence that we can give to these estimates, they could be exploited to obtain additional insights on the process of adaptation.

7.2 On the interpretation of Pelican results

7.2.1 Pelican identifies sites whose evolution *correlates* to variations of the phenotype

This point could be quickly summarized as the common trite that “correlation does not equal causation”, but let us go into more details on what that means in our case. Pelican’s predictions are indeed correlative by nature: it does not model the evolution of the phenotype at all, instead phenotypic traits — both extent and ancestral — are treated as input data. In contrast, an investigation of the causal relationship between the genotype and phenotype should model causality in some way, for example by modeling variations of a trait as a response to genotypic changes. The implication for Pelican is that it detects any site that shows a correlative pattern with the phenotype annotation that its underlying model is able to capture, but it does not say anything on the *nature* of the relationship between a given site and the phenotype. The expectation is that most sites discovered by Pelican have functional roles with regard to the phenotype under consideration, but there are several ways in which this might not be the case.

For example, every echolocating species in the Orthomam phylogeny is either a bat or a cetacean. There is evidence in the literature for the loss of umami perception — the taste of amino acids [Nelson et al., 2002]— in both these groups. The taste receptor TAS1R1 is “absent, unamplifiable, or pseudogenized” in bat species [Zhao et al., 2012], and under relaxed selection in cetaceans [Zhu et al., 2014], although it is quite conserved among vertebrates [Shi and Zhang, 2006]. This would be a good example of two entirely different functions (echolocation and umami perception) that coincide in the phenotype annotation. The convergent loss of taste would act as a confounding variable and could thus be a source of false interpretations if predictions were not checked carefully enough. Luckily, if I may say so, two factors protect us from finding TAS1R1 as a false positive: first, among the majority of echolocating species in Orthomam, a sequence for TAS1R1 is not even found in the alignment¹; second, the power to detect relaxed selection using Pelican is reduced compared to its ability to identify directional selection, and would further reduce the significance of sites in pseudogenes such as TAS1R1. As a result, TAS1R1 has rank 2806 (among 14509) in our list of predictions and was not identified as a candidate for association to the echolocation trait.

Nonetheless, this type of situation may arise in less favourable cases, and the possibility that another confounding trait may alter the results should be always considered. The case of ZNF536, which we identified to be associated both to echolocation and vocal learning, might be another example of this. There is experimental evidence for this gene to be involved in vocal learning [Wirthlin et al., 2022], and finding it among genes associated to echolocation might simply be explained by the fact that most vocal learners are also echolocators in the Orthomam dataset. However, we can not rule out that this gene may have a role in auditory perception in general and also has an adaptive function in performing echolocation.

Beyond this issue of colliding phenotypes, other sources of confusion are plausible, that tie to molecular mechanisms. I have already discussed one of them in this thesis: GC-biased gene conversion (gBGC) produces a signal in sequences that is akin to that of positive selection, and was found to increase the proportion of false discoveries in all methods that we evaluated in [chapter three](#). Instead of repeating here the discussion on the effect of this confounding factor, I propose to discuss another one that may also lead to false discoveries when scanning alignments with Pelican. In

¹Our model does not include insertion and deletion events yet, so that species without a sequence in the alignment are removed altogether.

comparison to the subject of the previous paragraph, *genetic hitchhiking* [Barton, 2000] has to do with an unaccounted for correlation between genes, instead of phenotypes. When an advantageous mutation appears and becomes fixed in the population, in a process sometimes called selective sweep, the frequency of other alleles close enough in the sequence — in “genetic linkage” — may also increase. As a consequence, these hitchhiker genes show patterns of positive selection in alignments of their sequences. If they are genetically linked to genes having a functional role with regard to the phenotype under consideration, they might be captured by Pelican, even though they have no functional association to the phenotype.

Now, let us consider the case when a position in a coding sequence was found to be associated with the phenotype by Pelican, e.g. by finding that it belongs to a protein that has a known functional role relevant for the phenotype under consideration. Even assuming that it is a signature for adaptation, the most information that we can draw from this is that there is an association between this gene and the phenotypic trait, and identify positions within the genes that are of particular importance in this association. But we can not say anything on the causal link between the two, in terms of evolutionary history. Did substitutions in the gene lead to the emergence of the trait ? Or did they occur subsequently to a primitive version of the trait, improving its viability ? These are questions that can not be answered solely from the application of this method.

7.2.2 On the interpretation of estimated parameters

There are two kinds of parameter estimates in the model of Pelican: the scale σ that controls the rate of evolution at one site, and frequency profiles π that give a direction for the substitution process.

A benefit of profile methods at large compared to d_N/d_S methods, is that they not only enable the identification of selective pressure changes, but also give a direction for it. This is a useful information, that can be interpreted in terms of functional changes related to the phenotype [Parto and Lartillot, 2017].

In the case of Pelican, we should keep in mind that the estimated profiles are not equivalent to amino acid fitnesses: in the best case, they are a reflection of the fitness profiles, that is distorted by the mutational and other non-adaptive processes. Moreover, they are generally incomplete, because of the sparse specification of the model (section 4.3.4), which does not consider unobserved amino acids that would have non-zero frequencies given an infinite sample size. That said, one might nonetheless exploit these frequency estimates at positive sites to make comparisons of the general direction of the substitution process between trait conditions. We have not identified yet a good metric that could allow the comparison of profiles. Kullback-Leibler divergence is a potential candidate, a measure of the divergence between two distributions that is difficult to interpret in our case and is not symmetric; it also does not behave well when some frequencies are null. A measure of angle between profiles such as the cosine function may also be considered. Profiles could also be characterized by physico-chemical properties of their amino acid content: for example, identifying at a site that the profile at condition A favors polar residues, while profile at condition B favors non-polar ones, could give insights on the kind of adaptation that occurred when confronted to the protein structure. More work is needed to validate the inference of profiles, and their interpretation.

7.2.3 On the choice of the phenotypic trait

The choice of the phenotypic trait affects the kind of results that can be expected from Pelican. There is of course the case of biological functions that are implemented in very different ways: for example the ability to fly that is shared between, say, birds and insects relies on entirely different

biological structures and is very unlikely to involve similar genes — not even considering that their genomes may have diverged too far to identify and align enough gene families.

A less obvious case is that of morphological structures in general, the evolution of which is posited to be driven mainly by variations in the regulation of gene expression (e.g. [Prud'homme et al., 2007]). This kind of adaptation would be undetectable by Pelican, which only focuses on coding sequences and does not account for transcriptional information. However, there is one instance where Pelican was successfully used to detect genomic substitutions correlated to a morphological traits. Some species of birds exhibit helmet-like structures that stem from their upper beak or their skull, and are a research interest of our colleague Anamaria Necşulea who applied Pelican to one of her dataset. She found that among the best ranking sites she obtained, some of them belonged to genes known to be involved with cranio-facial development.

This is also an interesting case regarding the choice of trait categories: bird species in this dataset could be assigned in binary categories (“helmet” or “no helmet”), or to a finer level in three categories depending on the helmet stemming from the beak or the skull. The latter strategy produced more convincing results when looking at the functional role of the best ranking genes. This highlights that the delimitation of categories is important when annotating discrete phenotypic traits. In that regard, I would recommend that attention is paid to the sample size available within each category, and that the number of species is not too unbalanced across categories. One should also keep in mind that the partition of traits in more than two categories makes the hypothesis that is tested less obvious, and may also reduce the power of this test. In our genomic screen for adaptation to different alimentary diets among mammals, we considered three categories: herbivore, carnivore and omnivore. By doing this, we actually test on each site that at least one condition has a different substitution dynamics from the others, with the null hypothesis that all conditions share a common substitution process. The power to reject this null hypothesis when molecular adaptation occurred in only one of the condition is decreased when compared to a more specific model where this condition is contrasted to the rest of the tree: the model with three categories involves more parameters, and thus requires a stronger signal to reject the homogeneous model in favor of the heterogeneous model. It is thus best suited to detect sites where *each* condition has a specific substitution dynamics².

Finally, a complementary approach could be implemented in our testing framework to compare a tripartite model to a bipartite one and answer more specific questions. For instance, a null model with a partition between aquatic and terrestrial clades could be confronted to a full model with a partition between freshwater, saltwater and terrestrial species. In this setting, both models are nested, as the reduced model could be represented using the full model by constraining amino acid profiles in the freshwater and saltwater conditions to be equal. This approach might be well suited to test for associations specific to life in freshwater or saltwater, excluding those related to adaptation to aquatic environments in general, and might be worth exploring.

7.2.4 The issue of calibration

We dedicated a lot of efforts to characterize the statistical behaviour of Pelican, and particularly on the calibration of p -values, because this an essential prerequisite to enable the determination of significance thresholds. We conclude that, although the distribution of p -values is generally not uniform under the null hypothesis, the deviation from it is low enough in particular when it comes to small p -values. This allows fixing a false discovery rate (FDR) threshold at the level of site

²Furthermore, the categories we chose for this trait may be too naive, as the omnivorous diet may involve adaptations at the intersection of the carnivorous and herbivorous ones.

predictions that actually informs on the number of false positives; regarding gene-level predictions, deviation from the uniform is generally larger, and tends to be conservative, although it is not always the case.

More attempts could be made at improving the calibration of the method, possibly using a procedure based on permutations of observations and simulations, as in [Saputra et al., 2021]. We resorted to a similar (but coarser) approach in [chapter five](#) to estimate the number of positive sites in genes associated to longevity. However, it is not clear whether this method actually ensures calibration across sites, as it might be that it guarantees instead p -values uniformity under the null across alternative annotations of the phylogeny at *one* site.

7.3 On input data, and uncertainties

In the current implementation of Pelican, several kinds of data used by the methods are treated as input, and are assumed to be accurate. However, most of these elements are themselves the result of algorithmic or statistical treatments, and carry their own uncertainties that should be quantified and accounted for further down in any analysis pipeline. Ideally, investigations in the field of molecular evolution should integrate every level of the analysis of sequences as “documents of evolutionary history” [Boussau and Daubin, 2010], from the building of alignments, the reconstruction of phylogenies, to the inference of evolutionary processes. Gene families are generally identified based on sequence similarity, a signal that can be altered by gene duplications or losses, or even non-homologous events such as exon shuffling (e.g [Long et al., 2003]). Sequence alignment then identifies homologous sites within homologous sequences, a process that depends on the phylogenetic relationship between sequences, which is often roughly estimated for that need. In turn, the inference of phylogenetic relationships between species in the form of phylogenetic trees, is performed from the alignment of their genome sequences: the cyclic dependency between the two procedures becomes obvious. This illustrates that gene family annotations, sequence alignments and phylogenetic trees are nothing but estimates, that should be jointly optimized, instead of successively reaching for individual optima. In practice, this is generally unreasonable, due to the computational complexity involved: the typical workflow is rather a step-by-step process, where the product of each stage (e.g the alignment) is used as input for subsequent stages (e.g. phylogenetic inference), carrying over uncertainties unaccounted for at each stage of the analysis.

7.3.1 Uncertainties in the sequences, and their use as representations of the genotype of species

Starting from the most elementary level of complexity in our inputs, that is amino acid or nucleotide sequences, I see at least two possible sources of noise in the data. First, even though sequencing technologies have tremendously improved as a whole in the recent years, the signal for identifying a position in the sequence is not always perfectly unambiguous, and does not indicate clearly the nature of the position. Assuming that we can assign a probability to the possible states that could be signaled as such, this is then easily accounted for due to the stochastic approach in Felsenstein’s algorithm. Since leaf states are represented by a vector of probabilities, when they are known with certainty they are represented as one-hot encoding of the observed state, but encoding uncertainty in the leaf state would be achievable if coupled with a model of sequencing error.

Second, in all of this work, we assume that the genome of a species could be represented as a single sequence. Since it is well known that there is genotypic variability within species and populations,

this is a gross simplification that completely overlooks intra-specific polymorphism. Therefore, the implicit hypothesis that is made is that each sequence we input to the method is a good consensus representation of the genotype of the species. This might be a reasonable assumption regarding sites that are highly conserved within the species, but not so much on highly polymorphic sites. Moreover, polymorphism is a richer information than a single consensus sequence, which could be exploited to improve the quality of predictions. Simple approaches have been proposed, such as the McDonald-Kreitman test [McDonald and Kreitman, 1991], but more complex models can be used to achieve this. This research path was beyond the scope of this thesis, but is a subject of interest in the community : some polymorphism aware models of evolution have been proposed (e.g. PoMo [De Maio et al., 2013]) and are an active area of research [Borges et al., 2022, Wilson et al., 2011, Mugal et al., 2020].

7.3.2 Uncertainties in the alignment

This latter point addressed the question of the content of the alignment, but we also must be wary of the quality of the alignment itself. It depends of course on the quality of each individual sequence (e.g. the quality of the assembly, and the amount of effort dedicated to its validation and curation), which I already discussed. It is also impacted by the degree of similarity between the sequences, as well as the choice of the algorithm used to perform their alignment — and the interaction between the two. Highly divergent sequences can be more difficult to align, as alignment algorithms typically require some degree of conservation to identify common patterns across the sequences. These misaligned sites may turn to be a source of misleading signal for positive selection. On another note, pseudogenes in particular might be difficult to identify and position in an alignment. A short review of these issues, with relevant references to the literature, can be found in [Kosiol and Anisimova, 2019b].

Another obstacle in the inference of multiple sequence alignments is that they generally require a guide phylogeny, and that, in turn, they are used to perform inference of phylogenetic trees. This creates a circular dependency that is generally not properly resolved, as the inference is performed step by step. However, approaches have been proposed to perform joint inference of alignments and phylogenies, as in e.g. [Pečerska et al., 2021].

7.3.3 Uncertainties in phylogenetic trees

Since we did not observe the history of speciation events, phylogenetic trees are necessarily the result of an inference from observable data, in this case sequence alignments mainly³. The consequence is that the reconstruction of the phylogeny adds uncertainty to what was carried over from the sequencing and alignment steps. So we do not know whether the tree we are working with is an accurate depiction of the evolutionary relationship between the species under consideration: we assume it is the best one, but we have no idea if other candidates could also be credible alternatives.

Since Pelican’s model is certainly not well-suited to infer phylogenies — if only because it operates at the amino acid level, and independently on each site — thus preventing the joint estimation of the tree and model parameters, other strategies must be resorted to. A possible alternative would consist in integrating over a distribution of possible trees, by running the method against these different trees and aggregating the results, accounting for the likelihood of each tree. The cost of such an approach would nonetheless be increased with the number of alternative trees that are considered, with no guarantees that this would have a noticeable effect on the quality of the predictions. This might hinder the applicability of such an approach at larger scales.

³Although other kinds of data, e.g. the fossil registry, can be accounted for.

7.3.4 Uncertainties in the phenotype

For our usage, phylogenies are also enriched with an additional feature: each of the branches is annotated with a phenotypic trait. These annotations are used when fitting the heterogeneous model of Pelican to determine the substitution process on each branch, and are thus critical for the application of the method. I identify two sources of variance for phenotypic traits: variability of the trait within each species between individuals, which can be measured; and then uncertainty on ancestral traits, which can not be observed anymore — fossils excepted — and must be inferred from data on extant species using statistical methods.

Much like in the previous paragraph regarding other tree features, the issue arises that we may be giving too much credit to a single estimate among many other possibilities, that are completely overlooked. Currently, we do not address this in a satisfactory manner, and assume that ancestral traits were reconstructed using a perfectly reliable method: only point estimates can be treated, without the possibility to account for confidence intervals or probability distributions on traits. In the case of discrete traits, we could handle their annotation in the tree as probability distributions instead of point estimates, by integrating over each possible state during the application of the pruning algorithm. The expression of the likelihood at a node n , conditionally to its state being x and the state at child c being y , assumes that the trait Z_{nc} on the branch connecting n and c with length t_{nc} is known with 100% confidence

$$\begin{aligned} L_n(x) &= \prod_{c \in \mathcal{C}(n)} \sum_{y \in \mathcal{k}} \mathbb{P}[y|x, t_{nc}] L_c(y) \quad \text{where } \mathcal{C}(n) \text{ is the set of children of node } n \\ &= \prod_{c \in \mathcal{C}(n)} \sum_{y \in \mathcal{k}} \left(\sum_{z \in \mathcal{T}} \mathbb{P}[y|x, t_{nc}, Z_{nc} = z] \mathbb{P}[Z_{nc} = z] \right) L_c(y) \end{aligned} \tag{7.1}$$

where \mathcal{T} is the set of trait modalities.

In this expression, $\mathbb{P}[Z_{nc} = z]$ acts as a prior probability, which is equal 1 since we currently consider only one possibility. But we could also use an annotation of traits as probability distributions to provide us with a different prior at each branch, and integrate over each possible state. At the time of writing, we are considering to implement this strategy to handle discrete trait distributions in the near future. It would not be applicable to continuous traits, as the integration over all possible states is more difficult and costly for continuous distributions.

Another kind of uncertainty we should deal with is that of the position of transitions between phenotype traits. In the current implementation, traits are annotated on branches using the value that is found at the child node, which implies that when a change of the trait occur in the tree it happens at one node and that a trait is constant along a given branch. This is overly simplistic, as variations of the trait may actually occur at any point within a branch, and not only at its extremities. This could be addressed by introducing pseudo-nodes within branches at positions where transitions were inferred to occur, allowing to switch between substitution processes at the adequate time, but I expect that their position would be difficult to infer.

These solutions do not however handle the problem of phenotype inference: they merely allow more flexibility to handle uncertainties in the estimates obtained beforehand. Our software implements rough approaches for the inference of ancestral traits, using Fitch's algorithm to reconstruct discrete traits at maximum parsimony, and a Brownian evolution model for continuous traits at maximum likelihood. These features are provided for convenience, but I am convinced that other approaches better adapted to each dataset exist and should be privileged, associated to human expertise whenever possible.

7.4 Hypotheses of the Pelican model, and their biological implications

At this point in the discussion, it might be useful to explicitly revisit the hypotheses that are made in the model of Pelican. The intent is to examine critically the modeling choices that were made, their scope and expected consequences on the results produced by this method.

7.4.1 Adaptation happens through coding sequences modifications

The scope of this work was restricted beforehand to identifying sites within genomes that are associated to a phenotypic trait. Pelican is the answer we came up with, that is based on a site-independent model of amino acid substitution. As a consequence, a variety of biological processes apart from substitutions are not considered, even though they could have a relationship with the phenotype under consideration. As I mentioned in the introduction, the equivalence between genotype and phenotype is only very partial: phenotypes are not completely determined by a single genotype, but are the result of complex interactions between genes and their environment at large, including other genes. In particular, genes are not expressed in the same way across species, individuals, developmental stages and tissues, and the expression of a gene is determinant of its functional effect at the scale of an organism.

Let us illustrate this with an example, drawn from our search for associations to the ability to echolocate in the Orthomam database. When confronting our results to the literature on the subject of the genomic bases for echolocation, we note that two genes (OTOS and OTOG) were found to be over-expressed among echolocating species [Dong et al., 2013], but were not discovered by Pelican ($p = 0.616$ and $p = 1$, respectively). These two genes code for proteins involved in the auditory system, which increases the credibility of their adaptive role with regard to the echolocation trait. However, this adaptation does not manifest through modifications of the sequence, but rather through variation of their expression level, which can not be uncovered using our approach.

7.4.2 Sites evolve independently from each other

Pelican, like a majority of models of sequence evolution, is built on the hypothesis that sites and genes evolve independently for each other. This hypothesis contradicts the current knowledge of molecular evolution, that establishes the existence of evolutionary interactions between genes, a process known as *epistasis*. This also includes the dependence between sites within genes, as mutations at one site may alter the tertiary structure of the protein, which is the product of complex biochemical interactions between amino acids. As the tertiary structure is determinant for the functional role of the protein, which may affect the fitness of the individual, these interactions between amino acids manifest an evolutionary dependence between codons in the sequence.

The simplifying hypothesis of site-independence is generally made purposefully, because it is a convenient one: it greatly simplifies the model and the process of fitting it to the data. In our case in particular, because of the constraint on the scalability of the method that should be applicable to analyses of large datasets in reasonable time, making the hypothesis of site-independence is somewhat necessary. However, modeling the co-evolution of sites could help to provide richer interpretations in some cases. For instance, I have discussed in section 6.4.1, on the analysis of molecular convergence in the Prestin gene, how the two sites having the most signal are in interaction in the tertiary structure of the human protein. Although we may compare the amino acid frequency profiles between

echolocating and non-echolocating species, it would be even more informative to examine what *combinations* of amino acids are found across echolocating species.

7.4.3 Similar traits involve similar genes across species

This hypothesis is not explicit in the model of Pelican, but rather has to do with the approach in general: we assume that an association can be made between a gene family and a phenotypic trait, on the basis of the molecular divergence observed between species. However, the evolutionary history of a gene may include events that makes the composition of orthologous gene families more difficult: for instance, duplications of genes may occur without speciation, which sometimes allows copies of the gene to be repurposed to other functions, or degenerate into pseudogenes. Identifying which copies of a gene should be compared between species is thus a challenge in itself, and is expected to have an impact on our predictions. Fortunately, we have not been confronted to this problem in our analyses of empirical data, because orthologous gene families in Orthomam had already been carefully delimited. However, it is expected that this issue will commonly appear when working with other datasets, and can not easily be accounted for in our model as we do not know a priori which copies are under selective pressure to maintain a function related to the phenotype under consideration.

Situations also exist where a similar trait is observed across different species, but has independent molecular bases. A striking example of this is the biosynthesis of caffeine, that has convergently evolved across numerous plant species. [Huang et al., 2016] reveal that several plant species like cacao, citrus, guaraná, coffee and tea have independently evolved to produce the exact same caffeine molecule, but that the underlying metabolic pathway varies across species. This implies that, if we were to study the molecular convergence of caffeine biosynthesis, it would be particularly important to distinguish clades on the basis of the metabolic pathway involved in caffeine production: gathering clades without this distinction would blend the signal for molecular adaptation between groups and is likely to yield misleading or partial results. In such a situation, a reasonable approach would be to take advantage of the categorisation into multiple conditions that is enabled in Pelican.

7.4.4 Modeling substitutions at the level of amino acids

Our work that I presented in [chapter three](#) shows that for the task of detecting sites whose substitution history is associated to a phenotype, modeling the substitution process directly at the level of amino acids appears to be sufficient and gives similar performance to that of codon-based models. Amino acid substitution models are simpler and involve reduced computational costs, mainly because the dimension of the state space for substitutions is smaller compared to codon models. In contrast to codon-based models that have the potential to distinguish selective from non-selective effects that shape the substitution process (e.g. [Ratnakumar et al., 2010]), modeling substitutions at the amino acid level makes it difficult to account for confounding factors. In [chapter three](#) we simulated a worst-case scenario for the effect of gBGC (figure 3.6), where its presence coincides perfectly with the phenotype annotation, and showed that it had a strong negative effect on prediction performance. This confounding effect is thus expected to be weaker when performing analysis of empirical datasets. The results we obtained on the Orthomam database might be affected by gBGC which could be a source of false predictions; to evaluate which genes are more susceptible to this effect, our results could be confronted to an estimation of the quantity of recombination hotspots for each gene within species in Orthomam reported in [Galtier, 2021].

7.5 Perspectives

In my opinion, an immediate improvement on the model could be achieved by incorporating insertion and deletion of amino acids, possibly following the methodology from [Rivas and Eddy, 2008] or [Bouchard-Côté and Jordan, 2013, Jowkar et al., 2022]. For now, gaps within sequence alignments are ignored altogether by trimming the tree at each alignment site to only include species where an amino acid is present. This is expected to cause a loss of power for the method, among other possible statistical problems, by reducing the sample size as well as removing a signal that could be informative on the evolutionary dynamics at one site. This is a motivation for explicitly modeling indels in Pelican, as it should contribute to make the model better adapted to handle such common situations.

We have paid attention in the software implementation of Pelican to make it a pleasant experience for the user, but it is still in an early stage and could be further improved; efforts still need to be made to improve the documentation and general accessibility. Some quality-of-life features for the command line interface, such as command auto-completion, are missing yet; they require little development efforts and could facilitate the use of the software. We also have been considering for a while the idea of including tools for the visualisation and interpretation of results, in the form of a local web application. We imagine it could allow the exploration of gene and site predictions across runs and a clear representation of positive sites within proteins. Such representation would be particularly helpful if it could be confronted to the tertiary structure of said protein, either using databases of known structures (e.g. Protein Data Bank) or resorting to inference using AlphaFold 2 [Jumper et al., 2021]. Visualisation of estimated amino acid profiles would also be a desirable feature, which could be enriched with physicochemical properties: e.g. this could inform on which kind of amino acid was favored in sites associated to the phenotype. The definition of a metric measuring the divergence between amino acid profiles could also be useful, in that it would summarize an effect size of the variation in amino acid preferences between phenotypes at one site. This would also be an opportunity to investigate in more details the results that we obtained from scanning the Orthomam database for associations to various phenotypes.

In these investigations on Orthomam, we have found that Pelican produces convincing results; the method was also extensively validated using simulations. But it would be comforting to corroborate that the method works well on a diversity of empirical datasets, involving other tree topologies, species and alignments, to further validate the robustness of the method. For example, we have identified with our experiments on simulations with the Influenza phylogeny, a case where our predictions did not distinguish well positive from negative sites (and then genes). Our results suggest that this stemmed from the absence of phenotype convergence in the tree, that resulted in a loss of power to identify positive sites in the simulated alignments. However, this seems to contradict our previous results that showed that the detection performance of Pelican was rather insensitive to the number of transitions. Further research would be needed to understand better the effect of the number of transitions, and to make it clear which features of a dataset might be problematic; it is however a challenging task to come up with an exhaustive knowledge of unfavorable instances. It would be particularly important to further validate the continuous model of Pelican in a variety of settings: at the moment, it was only applied to the analysis of a few genes, and should be confronted to other empirical datasets. Moreover, the results we obtained were not consistent with those reported in a previous study of these genes and their effect on the longevity of mammal species [Farré et al., 2021], and should be more thoroughly examined to understand the source of this discrepancy.

Going forward, some other methodological developments seem promising to further increase the throughput of the method. Exploiting the highly parallel nature of fitting the model on each site in a set of alignments using computations on the GPU might further increase the throughput of the method, as is suggested in section 4.7. Alternatively, another approach to estimate parameters of the model at maximum likelihood using distributions of substitution histories, also known as substitution mappings [Nielsen, 2002, Rodrigue et al., 2008, de Koning et al., 2010], is expected to drastically decrease the cost of evaluating the likelihood at one parameter point, thus improving the overall speed of the method. Importantly, the majority of the computation time in this case would be the generation of substitution mappings, a cost that could be mitigated by the fact that these mappings might be reused for different phenotype annotations of the same dataset. We have been working on a prototype implementation using substitution mappings that has been achieving promising results. I also speculate that the generation of substitution mappings could be accomplished using computation on GPU to considerably increase the throughput. GPU computation has also been reported to have a lesser carbon footprint than CPU for some bioinformatics applications [Grealey et al., 2022], but that is not always the case. A comparison of the carbon cost efficiency between potential implementations would also be relevant in the choice of future development directions.

Finally, I would find it interesting to explore variations of our approach that integrate other biological variables that could be relevant. Specifically, I expect that a structurally aware extension of the model that incorporates information on the side-chain angle of amino acid [Perron et al., 2019] could improve its ability to detect meaningful substitutions with regard to the phenotype under consideration. Perron et al. have shown that their model performs better than models using exclusively amino acid alphabets for a variety of inference tasks in a phylogenetic context, which hopefully could be translated to our setting.

In the long run, I hope that this work could provide a basis to perform large scale, systematic functional annotations of genomes based on predicted associations to a range of phenotypic traits. It would certainly not be as trustworthy as experimental evidence, but could be utilized to orient the focus on genes candidate for further investigation, to answer questions on their functional or adaptive role. More specifically, this could be useful to provide functional annotations of genes in non-model species, for which knowledge on the genome is generally more scarce.

Bibliography

- [Abhiman and Sonnhammer, 2005] Abhiman, S. and Sonnhammer, E. L. L. (2005). Large-scale prediction of function shift in protein families with a focus on enzymatic function. *Proteins*, 60(4):758–768. [53](#)
- [Anderson et al., 1999] Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., and Sorensen, D. (1999). *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, third edition. [71](#), [77](#), [88](#)
- [Archana et al., 2017] Archana, P., Aleena, J., Pragna, P., Vidya, M., Abdul Niyas, P., Bagath, M., Krishnan, G., Manimaran, A., Beena, Kurien, E., Veerasamy, S., and Bhatta, R. (2017). Role of Heat Shock Proteins in Livestock Adaptation to Heat Stress. *Journal of Dairy, Veterinary & Animal Research*, 5(1). [159](#)
- [Ayres et al., 2012] Ayres, D. L., Darling, A., Zwickl, D. J., Beerli, P., Holder, M. T., Lewis, P. O., Huelsenbeck, J. P., Ronquist, F., Swofford, D. L., Cummings, M. P., Rambaut, A., and Suchard, M. A. (2012). BEAGLE: An Application Programming Interface and High-Performance Computing Library for Statistical Phylogenetics. *Systematic Biology*, 61(1):170–173. [112](#)
- [Bacaër, 2011] Bacaër, N. (2011). *A Short History of Mathematical Population Dynamics*. Springer London, London. [24](#)
- [Barrett et al., 1991] Barrett, A., Normand, M., and Peleg, M. (1991). A ‘log—beta’ vs. the log—normal distribution for particle populations with a wide finite size range. *Powder Technology*, 66(2):195–199. [136](#)
- [Barton, 2000] Barton, N. H. (2000). Genetic hitchhiking. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences*, 355(1403):1553–1562. [166](#)
- [Bel et al., 2016] Bel, L., Daudin, J., Etienne, M., Lebarbier, E., Mary-Huard, T., Robin, S., and Vuillet, C. (2016). *Le Modèle Linéaire et ses Extensions*. Edition Ellipses. [40](#)
- [Besnard et al., 2009] Besnard, G., Muasya, A. M., Russier, F., Roalson, E. H., Salamin, N., and Christin, P.-A. (2009). Phylogenomics of c4 photosynthesis in sedges (cyperaceae): Multiple appearances and genetic convergence. *Molecular Biology and Evolution*, 26(8):1909–1919. [16](#), [33](#), [61](#), [195](#)
- [Bloom, 2014] Bloom, J. D. (2014). An experimentally informed evolutionary model improves phylogenetic fit to divergent lactamase homologs. *Molecular Biology and Evolution*, 31(10):2753–2769. [55](#), [65](#), [117](#)
- [Bloom, 2017] Bloom, J. D. (2017). Identification of positive selection in genes is greatly improved by using experimentally informed site-specific models. *Biology Direct*, 12(1):1. [29](#), [31](#), [65](#)

- [Bolívar et al., 2019] Bolívar, P., Guéguen, L., Duret, L., Ellegren, H., and Mugal, C. F. (2019). GC-biased gene conversion conceals the prediction of the nearly neutral theory in avian genomes. *Genome Biology*, 20(1):5. 55, 57
- [Borges et al., 2022] Borges, R., Boussau, B., Höhna, S., Pereira, R. J., and Kosiol, C. (2022). Polymorphism-aware estimation of species trees and evolutionary forces from genomic sequences with revbayes. *Methods in Ecology and Evolution*, 13(11):2339–2346. 169
- [Bouchard-Côté and Jordan, 2013] Bouchard-Côté, A. and Jordan, M. I. (2013). Evolutionary inference via the Poisson Indel Process. *Proceedings of the National Academy of Sciences*, 110(4):1160–1166. 173
- [Boussau and Daubin, 2010] Boussau, B. and Daubin, V. (2010). Genomes as documents of evolutionary history. *Trends in Ecology & Evolution*, 25(4):224–232. 168
- [Boussau and Gouy, 2006] Boussau, B. and Gouy, M. (2006). Efficient likelihood computations with nonreversible models of evolution. *Systematic Biology*, 55(5):756–768. 80
- [Boyd et al., 2013] Boyd, K., Eng, K. H., and Page, C. D. (2013). *Area under the Precision-Recall Curve: Point Estimates and Confidence Intervals*, volume 7908 of *Lecture Notes in Computer Science*, page 451–466. Springer Berlin Heidelberg, Berlin, Heidelberg. 35, 58, 122
- [Bridgham et al., 2006] Bridgham, J. T., Carroll, S. M., and Thornton, J. W. (2006). Evolution of hormone-receptor complexity by molecular exploitation. *Science (New York, N.Y.)*, 312(5770):97–101. 53
- [Bulmer, 1986] Bulmer, M. (1986). Neighboring base effects on substitution rates in pseudogenes. *Molecular Biology and Evolution*, 3(4). 32
- [Chabrol et al., 2018] Chabrol, O., Royer-Carenzi, M., Pontarotti, P., and Didier, G. (2018). Detecting the molecular basis of phenotypic convergence. *Methods in Ecology and Evolution*, 9(11):2170–2180. 132
- [Chikina et al., 2016] Chikina, M., Robinson, J. D., and Clark, N. L. (2016). Hundreds of genes experienced convergent shifts in selective pressure in marine mammals. *Molecular Biology and Evolution*, 33(9):2182–2192. 16, 68, 154
- [Chor and Snir, 2007] Chor, B. and Snir, S. (2007). Analytic solutions of maximum likelihood on forks of four taxa. *Mathematical Biosciences*, 208(2):347–358. 86
- [Christin et al., 2007] Christin, P.-A., Salamin, N., Savolainen, V., Duvall, M. R., and Besnard, G. (2007). C4 Photosynthesis evolved in grasses via parallel adaptive genetic changes. *Current biology*, 17(14):1241–7. 53
- [Crow, 1987] Crow, J. F. (1987). Population genetics history : a personal view. *Annual Review of Genetics*, 21(1):1–22. 24
- [Davies et al., 2013] Davies, K. T., Maryanto, I., and Rossiter, S. J. (2013). Evolutionary origins of ultrasonic hearing and laryngeal echolocation in bats inferred from morphological analyses of the inner ear. *Frontiers in Zoology*, 10(1):2. 147, 215

- [Davies et al., 2012] Davies, K. T. J., Cotton, J. A., Kirwan, J. D., Teeling, E. C., and Rossiter, S. J. (2012). Parallel signatures of sequence evolution among hearing genes in echolocating mammals: an emerging model of genetic convergence. *Heredity*, 108(5):480–489. [147](#), [148](#), [215](#)
- [Davydov et al., 2016] Davydov, I. I., Robinson-Rechavi, M., and Salamin, N. (2016). State aggregation for fast likelihood computations in molecular evolution. *Bioinformatics*. [90](#)
- [Davydov et al., 2019] Davydov, I. I., Salamin, N., and Robinson-Rechavi, M. (2019). Large-Scale Comparative Analysis of Codon Models Accounting for Protein and Nucleotide Selection. *Molecular Biology and Evolution*, 36(6):1316–1332. [55](#)
- [Dayhoff et al., 1978] Dayhoff, M., Schwartz, R., and Orcutt, B. (1978). 22 a model of evolutionary change in proteins. *Atlas of protein sequence and structure*, 5:345–352. [81](#)
- [de Koning et al., 2010] de Koning, A. P. J., Gu, W., and Pollock, D. D. (2010). Rapid Likelihood Analysis on Large Phylogenies Using Partial Sampling of Substitution Histories. *Molecular Biology and Evolution*, 27(2):249–265. [174](#)
- [De Maio et al., 2013] De Maio, N., Schlötterer, C., and Kosiol, C. (2013). Linking great apes genome evolution across time scales using polymorphism-aware phylogenetic models. *Molecular Biology and Evolution*, 30(10):2249–2262. [169](#)
- [Dong et al., 2013] Dong, D., Lei, M., Liu, Y., and Zhang, S. (2013). Comparative inner ear transcriptome analysis between the Rickett’s big-footed bats (*Myotis ricketti*) and the greater short-nosed fruit bats (*Cynopterus sphinx*). *BMC Genomics*, 14(1):916. [147](#), [148](#), [171](#), [215](#)
- [dos Reis, 2015] dos Reis, M. (2015). How to calculate the non-synonymous to synonymous rate ratio of protein-coding genes under the Fisher–Wright mutation–selection framework. *Biology Letters*, 11(4). [54](#)
- [Duchemin et al., 2022] Duchemin, L., Lanore, V., Veber, P., and Boussau, B. (2022). Evaluation of methods to detect shifts in directional selection at the genome scale. *Molecular Biology and Evolution*. [116](#), [118](#), [120](#)
- [Duret et al., 2002] Duret, L., Semon, M., Piganeau, G., Mouchiroud, D., and Galtier, N. (2002). Vanishing gc-rich isochores in mammalian genomes. *Genetics*, 162(4):1837–1847. [31](#)
- [Dutheil et al., 2012] Dutheil, J. Y., Galtier, N., Romiguier, J., Douzery, E. J., Ranwez, V., and Boussau, B. (2012). Efficient Selection of Branch-Specific Models of Sequence Evolution. *Molecular Biology and Evolution*, 29(7):1861–1874. [54](#)
- [Espeseth et al., 2007] Espeseth, T., Endestad, T., Rootwelt, H., and Reinvang, I. (2007). Nicotine receptor gene CHRNA4 modulates early event-related potentials in auditory and visual oddball target detection tasks. *Neuroscience*, 147(4):974–985. [147](#), [215](#)
- [Fabregat et al., 2018] Fabregat, A., Jupe, S., Matthews, L., Sidiropoulos, K., Gillespie, M., Garapati, P., Haw, R., Jassal, B., Korninger, F., May, B., Milacic, M., Roca, C. D., Rothfels, K., Sevilla, C., Shamovsky, V., Shorsler, S., Varusai, T., Viteri, G., Weiser, J., Wu, G., Stein, L., Hermjakob, H., and D’Eustachio, P. (2018). The Reactome Pathway Knowledgebase. *Nucleic Acids Research*, 46(D1):649–655. [146](#)

- [Farré et al., 2021] Farré, X., Molina, R., Barteri, F., Timmers, P. R. H. J., Joshi, P. K., Oliva, B., Acosta, S., Esteve-Altava, B., Navarro, A., and Muntané, G. (2021). Comparative Analysis of Mammal Genomes Unveils Key Genomic Variability for Human Life Span. *Molecular Biology and Evolution*, 38(11):4948–4961. [114](#), [116](#), [120](#), [123](#), [124](#), [125](#), [126](#), [128](#), [173](#)
- [Felsenstein, 1981] Felsenstein, J. (1981). Evolutionary trees from dna sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376. [66](#), [80](#), [83](#), [116](#)
- [Felsenstein, 1985] Felsenstein, J. (1985). Phylogenies and the comparative method. *The American Naturalist*, 125(1.):1–15. [79](#), [120](#)
- [Fisher, 1922] Fisher, R. A. (1922). On the dominance ratio. *Proceedings of the Royal Society of Edinburgh*, 42:321–341. [24](#)
- [Fisher, 1932] Fisher, R. A. (1932). *Statistical methods for research workers*, 4th ed. Statistical methods for research workers, 4th ed. Oliver & schä, Oxford, England. [133](#)
- [Fitch, 1971] Fitch, W. M. (1971). Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topology. *Systematic Biology*, 20(4):406–416. [33](#), [79](#)
- [Flaxman et al., 2020] Flaxman, S., Mishra, S., Gandy, A., Unwin, H. J. T., Mellan, T. A., Coupland, H., Whittaker, C., Zhu, H., Berah, T., Eaton, J. W., Monod, M., Imperial College COVID-19 Response Team, Perez-Guzman, P. N., Schmit, N., Cilloni, L., Ainslie, K. E. C., Baguelin, M., Boonyasiri, A., Boyd, O., Cattarino, L., Cooper, L. V., Cucunubá, Z., Cuomo-Dannenburg, G., Dighe, A., Djaafara, B., Dorigatti, I., van Elsland, S. L., FitzJohn, R. G., Gaythorpe, K. A. M., Geidelberg, L., Grassly, N. C., Green, W. D., Hallett, T., Hamlet, A., Hinsley, W., Jeffrey, B., Knock, E., Laydon, D. J., Nedjati-Gilani, G., Nouvellet, P., Parag, K. V., Siveroni, I., Thompson, H. A., Verity, R., Volz, E., Walters, C. E., Wang, H., Wang, Y., Watson, O. J., Winskill, P., Xi, X., Walker, P. G. T., Ghani, A. C., Donnelly, C. A., Riley, S., Vollmer, M. A. C., Ferguson, N. M., Okell, L. C., and Bhatt, S. (2020). Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe. *Nature*, 584(7820):257–261. [231](#)
- [Galtier, 2021] Galtier, N. (2021). Fine-scale quantification of GC-biased gene conversion intensity in mammals. *Peer Community Journal*, 1. [172](#)
- [Galtier and Duret, 2007] Galtier, N. and Duret, L. (2007). Adaptation or biased gene conversion? Extending the null hypothesis of molecular evolution. *Trends in Genetics*, 23(6):273–277. [31](#)
- [Gillespie, 1976] Gillespie, D. T. (1976). A general method for numerically simulating the stochastic time evolution of coupled chemical reactions. *Journal of Computational Physics*, 22(4):403–434. [29](#), [71](#), [117](#)
- [Glémin et al., 2015] Glémin, S., Arndt, P. F., Messer, P. W., Petrov, D., Galtier, N., and Duret, L. (2015). Quantification of GC-biased gene conversion in the human genome. *Genome Research*, 25(8):1215–1228. [72](#)
- [Goldman and Yang, 1994] Goldman, N. and Yang, Z. (1994). A codon-based model of nucleotide substitution for protein-coding dna sequences. *Molecular Biology and Evolution*. [53](#), [116](#)
- [Grealey et al., 2022] Grealey, J., Lannelongue, L., Saw, W.-Y., Marten, J., Méric, G., Ruiz-Carmona, S., and Inouye, M. (2022). The Carbon Footprint of Bioinformatics. *Molecular Biology and Evolution*, 39(3). [174](#)

- [Gu et al., 2013] Gu, X., Zou, Y., Su, Z., Huang, W., Zhou, Z., Arendsee, Z., and Zeng, Y. (2013). An update of DIVERGE software for functional divergence analysis of protein family. *Molecular Biology and Evolution*, 30(7):1713–1719. [53](#)
- [Guindon, 2003] Guindon, S. (2003). *Méthodes et algorithmes pour l’approche statistique en phylogénie*. PhD thesis, Université de Montpellier. [85](#)
- [Guindon et al., 2004] Guindon, S., Rodrigo, A. G., Dyer, K. A., and Huelsenbeck, J. P. (2004). Modeling the site-specific variation of selection patterns along lineages. *Proceedings of the National Academy of Sciences of the United States of America*, 101(35). [54](#)
- [Guéguen and Duret, 2018] Guéguen, L. and Duret, L. (2018). Unbiased Estimate of Synonymous and Nonsynonymous Substitution Rates with Nonstationary Base Composition. *Molecular Biology and Evolution*, 35(3):734–742. [55](#), [67](#), [69](#)
- [Halabi et al., 2021] Halabi, K., Karin, E. L., Guéguen, L., and Mayrose, I. (2021). A Codon Model for Associating Phenotypic Traits with Altered Selective Patterns of Sequence Evolution. *Systematic Biology*, 70(3):608–622. [68](#)
- [Haldane, 1927] Haldane, J. B. S. (1927). A mathematical theory of natural and artificial selection, part v: Selection and mutation. *Mathematical Proceedings of the Cambridge Philosophical Society*, 23(7):838–844. [24](#)
- [Halpern and Bruno, 1998] Halpern, A. L. and Bruno, W. J. (1998). Evolutionary distances for protein-coding sequences: modeling site-specific residue frequencies. *Molecular Biology and Evolution*, 15(7):910–917. [25](#), [26](#), [54](#), [55](#), [117](#)
- [Hasegawa et al., 1985] Hasegawa, M., Kishino, H., and Yano, T.-a. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *Journal of Molecular Evolution*, 22(2):160–174. [25](#)
- [Hettle et al., 2018] Hettle, A. G., Vickers, C., Robb, C. S., Liu, F., Withers, S. G., Hehemann, J.-H., and Boraston, A. B. (2018). The Molecular Basis of Polysaccharide Sulfatase Activity and a Nomenclature for Catalytic Subsites in this Class of Enzyme. *Structure*, 26(5):747–758.e4. [153](#), [154](#), [220](#), [222](#)
- [Ho and Hurst, 2022] Ho, A. T. and Hurst, L. D. (2022). Unusual mammalian usage of tga stop codons reveals that sequence conservation need not imply purifying selection. *PLOS Biology*, 20(5). [67](#)
- [Hou et al., 2022] Hou, Y., Zhao, S., Liu, Q., Zhang, X., Sha, T., Su, Y., Zhao, W., Bao, Y., Xue, Y., and Chen, H. (2022). Ongoing positive selection drives the evolution of sars-cov-2 genomes. *Genomics, Proteomics & Bioinformatics*. [16](#)
- [Huang et al., 2016] Huang, R., O’Donnell, A. J., Barboline, J. J., and Barkman, T. J. (2016). Convergent evolution of caffeine in plants by co-option of exapted ancestral enzymes. *Proceedings of the National Academy of Sciences*, 113(38). [172](#)
- [Ihrle et al., 2005] Ihrle, R. A., Marques, M. R., Nguyen, B. T., Horner, J. S., Papazoglu, C., Bronson, R. T., Mills, A. A., and Attardi, L. D. (2005). Perp Is a p63-Regulated Gene Essential for Epithelial Integrity. *Cell*, 120(6):843–856. [153](#), [222](#)

- [Imura et al., 2007] Imura, K., Yoshioka, T., Hikita, I., Tsukahara, K., Hirasawa, T., Higashino, K., Gahara, Y., Arimura, A., and Sakata, T. (2007). Influence of TRPV3 mutation on hair growth cycle in mice. *Biochemical and Biophysical Research Communications*, 363(3):479–483. [153](#), [222](#)
- [Ishiwata-Endo et al., 2020] Ishiwata-Endo, H., Kato, J., Stevens, L. A., and Moss, J. (2020). ARH1 in Health and Disease. *Cancers*, 12(2):479. [220](#)
- [Janik and Knörnschild, 2021] Janik, V. M. and Knörnschild, M. (2021). Vocal production learning in mammals revisited. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 376(1836). [159](#)
- [JCBN, 1984] JCBN (1984). Nomenclature and Symbolism for Amino Acids and Peptides. Recommendations 1983. *European Journal of Biochemistry*, 138(1):9–37. [77](#)
- [Jones et al., 2019] Jones, C. T., Susko, E., and Bielawski, J. P. (2019). Looking for Darwin in Genomic Sequences: Validity and Success Depends on the Relationship Between Model and Data. In Anisimova, M., editor, *Evolutionary Genomics: Statistical and Computational Methods*, Methods in Molecular Biology, pages 399–426. Springer, New York, NY. [54](#), [55](#), [69](#)
- [Jones et al., 2016] Jones, C. T., Youssef, N., Susko, E., and Bielawski, J. P. (2016). Shifting balance on a static mutation–selection landscape: A novel scenario of positive selection. *Molecular Biology and Evolution*. [26](#)
- [Jones et al., 2017] Jones, C. T., Youssef, N., Susko, E., and Bielawski, J. P. (2017). Shifting Balance on a Static Mutation–Selection Landscape: A Novel Scenario of Positive Selection. *Molecular Biology and Evolution*, 34(2):391–407. [54](#)
- [Jones et al., 2009] Jones, K. E., Bielby, J., Cardillo, M., Fritz, S. A., O’Dell, J., Orme, C. D. L., Safi, K., Sechrest, W., Boakes, E. H., Carbone, C., Connolly, C., Cutts, M. J., Foster, J. K., Grenyer, R., Habib, M., Plaster, C. A., Price, S. A., Rigby, E. A., Rist, J., Teacher, A., Bininda-Emonds, O. R. P., Gittleman, J. L., Mace, G. M., and Purvis, A. (2009). PanTHERIA: a species-level database of life history, ecology, and geography of extant and recently extinct mammals: Ecological Archives E090-184. *Ecology*, 90(9):2648–2648. [120](#)
- [Jowkar et al., 2022] Jowkar, G., Pečerska, J., Maiolo, M., Gil, M., and Anisimova, M. (2022). ARPIP: Ancestral Sequence Reconstruction with Insertions and Deletions under the Poisson Indel Process. *Systematic Biology*. [173](#)
- [Jukes et al., 1969] Jukes, T. H., Cantor, C. R., et al. (1969). Evolution of protein molecules. *Mammalian protein metabolism*, 3:21–132. [81](#)
- [Jumper et al., 2021] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589. [70](#), [173](#)
- [Kapralov et al., 2012] Kapralov, M. V., Smith, J. A. C., and Filatov, D. A. (2012). Rubisco evolution in c4 eudicots: An analysis of amaranthaceae sensu lato. *PLoS ONE*, 7(12). [16](#), [33](#), [61](#), [197](#)

- [Kimura, 1962] Kimura, M. (1962). On the probability of fixation of mutant genes in a population. *Genetics*, 47(6):713–719. [24](#), [25](#), [26](#)
- [Kimura, 1983] Kimura, M. (1983). *The Neutral Theory of Molecular Evolution*. Cambridge University Press, Cambridge. [72](#), [117](#)
- [Kloareg and Quatrano, 1988] Kloareg, B. and Quatrano, R. (1988). Structure of the cell walls of marine algae and ecophysiological functions of the matrix polysaccharides. *OCEANOGRAPHY AND MARINE BIOLOGY: AN ANNUAL REVIEW.*, 26:259–315. [154](#)
- [Kolberg et al., 2020] Kolberg, L., Raudvere, U., Kuzmin, I., Vilo, J., and Peterson, H. (2020). gprofiler2 – an R package for gene list functional enrichment analysis and namespace conversion toolset g:Profiler. *F1000Research*, 9. [146](#)
- [Korber et al., 2020] Korber, B., Fischer, W. M., Gnanakaran, S., Yoon, H., Theiler, J., Abfalterer, W., Hengartner, N., Giorgi, E. E., Bhattacharya, T., Foley, B., Hastie, K. M., Parker, M. D., Partridge, D. G., Evans, C. M., Freeman, T. M., de Silva, T. I., Angyal, A., Brown, R. L., Carrilero, L., Green, L. R., Groves, D. C., Johnson, K. J., Keeley, A. J., Lindsey, B. B., Parsons, P. J., Raza, M., Rowland-Jones, S., Smith, N., Tucker, R. M., Wang, D., Wyles, M. D., McDanal, C., Perez, L. G., Tang, H., Moon-Walker, A., Whelan, S. P., LaBranche, C. C., Saphire, E. O., and Montefiori, D. C. (2020). Tracking Changes in SARS-CoV-2 Spike: Evidence that D614G Increases Infectivity of the COVID-19 Virus. *Cell*, 182(4):812–827.e19. [53](#)
- [Kosakovsky Pond et al., 2011] Kosakovsky Pond, S. L., Murrell, B., Fourment, M., Frost, S. D., Delport, W., and Scheffler, K. (2011). A random effects branch-site model for detecting episodic diversifying selection. *Molecular Biology and Evolution*, 28(11):3033–3043. [54](#)
- [Kosiol and Anisimova, 2019a] Kosiol, C. and Anisimova, M. (2019a). Selection Acting on Genomes. In Anisimova, M., editor, *Evolutionary Genomics: Statistical and Computational Methods*, pages 373–397. Springer New York, New York, NY. [54](#)
- [Kosiol and Anisimova, 2019b] Kosiol, C. and Anisimova, M. (2019b). Selection acting on genomes. In *Evolutionary Genomics: Statistical and Computational Methods*, volume 1910 of *Methods in Molecular Biology*, pages 386–387. Springer New York, New York, NY. [169](#)
- [Kosiol et al., 2008] Kosiol, C., Vinar, T., da Fonseca, R. R., Hubisz, M. J., Bustamante, C. D., Nielsen, R., and Siepel, A. (2008). Patterns of positive selection in six Mammalian genomes. *PLoS genetics*, 4(8). [54](#)
- [Lanave et al., 1984] Lanave, C., Preparata, G., Saccone, C., and Serio, G. (1984). A new method for calculating evolutionary substitution rates. *Journal of Molecular Evolution*, 20(1):86–93. [27](#), [80](#), [117](#)
- [Le and Gascuel, 2008] Le, S. Q. and Gascuel, O. (2008). An improved general amino acid replacement matrix. *Molecular Biology and Evolution*, 25(7):1307–1320. [78](#), [81](#), [117](#)
- [Lefébure et al., 2017] Lefébure, T., Morvan, C., Malard, F., François, C., Konecny-Dupré, L., Guéguen, L., Weiss-Gayet, M., Seguin-Orlando, A., Ermini, L., Sarkissian, C. D., Charrier, N. P., Eme, D., Mermillod-Blondin, F., Duret, L., Vieira, C., Orlando, L., and Douady, C. J. (2017). Less effective selection leads to larger genomes. *Genome Research*, 27(6):1016–1028. [16](#)

- [Leroy et al., 2021] Leroy, X., Doligez, D., Frisch, A., Garrigue, J., Rémy, D., and Vouillon, J. (2021). The ocaml system: Documentation and user’s manual. *INRIA*, 3:42. [71](#)
- [Levine and Casella, 2001] Levine, R. A. and Casella, G. (2001). Implementations of the monte carlo em algorithm. *Journal of Computational and Graphical Statistics*, 10(3):422–439. [49](#)
- [Li et al., 2010] Li, Y., Liu, Z., Shi, P., and Zhang, J. (2010). The hearing gene Prestin unites echolocating bats and whales. *Current Biology*, 20(2). [53](#), [148](#), [149](#)
- [Liu and Nocedal, 1989] Liu, D. C. and Nocedal, J. (1989). On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1-3):503–528. [109](#)
- [Liu et al., 2010] Liu, Y., Cotton, J. A., Shen, B., Han, X., Rossiter, S. J., and Zhang, S. (2010). Convergent sequence evolution between echolocating bats and dolphins. *Current Biology*, 20(2). [16](#), [148](#), [149](#), [150](#)
- [Liu et al., 2011] Liu, Z., Li, S., Wang, W., Xu, D., Murphy, R. W., and Shi, P. (2011). Parallel evolution of KCNQ4 in echolocating bats. *PloS one*, 6(10). [148](#)
- [Long et al., 2003] Long, M., Betrán, E., Thornton, K., and Wang, W. (2003). The origin of new genes: glimpses from the young and old. *Nature Reviews Genetics*, 4(11):865–875. [168](#)
- [Loughin, 2004] Loughin, T. M. (2004). A systematic comparison of methods for combining p-values from independent tests. *Computational Statistics & Data Analysis*, 47(3):467–485. [132](#)
- [Magomedova et al., 2019] Magomedova, L., Tiefenbach, J., Zilberman, E., Le Billan, F., Voisin, V., Saikali, M., Boivin, V., Robitaille, M., Guerousov, S., Irimia, M., Ray, D., Patel, R., Xu, C., Jeyasuria, P., Bader, G. D., Hughes, T. R., Morris, Q. D., Scott, M. S., Krause, H., Angers, S., Blencowe, B. J., and Cummins, C. L. (2019). ARGLU1 is a transcriptional coactivator and splicing regulator important for stress hormone signaling and development. *Nucleic Acids Research*, 47(6):2856–2870. [229](#)
- [Marcovitz et al., 2019] Marcovitz, A., Turakhia, Y., Chen, H. I., Gloudemans, M., Braun, B. A., Wang, H., and Bejerano, G. (2019). A functional enrichment test for molecular convergent evolution finds a clear protein-coding signal in echolocating bats and whales. *Proceedings of the National Academy of Sciences*, 116(42). [53](#)
- [Mayrose, 2004] Mayrose, I. (2004). Comparison of Site-Specific Rate-Inference Methods for Protein Sequences: Empirical Bayesian Methods Are Superior. *Molecular Biology and Evolution*, 21(9):1781–1791. [86](#)
- [McDonald and Kreitman, 1991] McDonald, J. H. and Kreitman, M. (1991). Adaptive protein evolution at the Adh locus in Drosophila. *Nature*, 351(6328):652–654. [169](#)
- [McGowen et al., 2020] McGowen, M. R., Tsagkogeorga, G., Williamson, J., Morin, P. A., and Rossiter, a. S. J. (2020). Positive Selection and Inactivation in the Vision and Hearing Genes of Cetaceans. *Molecular Biology and Evolution*, 37(7):2069–2083. [147](#), [215](#)
- [Merényi et al., 2020] Merényi, Z., Prasanna, A. N., Wang, Z., Kovács, K., Hegedüs, B., Bálint, B., Papp, B., Townsend, J. P., and Nagy, L. G. (2020). Unmatched Level of Molecular Convergence among Deeply Divergent Complex Multicellular Fungi. *Molecular Biology and Evolution*, 37(8):2228–2240. [53](#)

- [Meunier et al., 2005] Meunier, J., Khelifi, A., Navratil, V., and Duret, L. (2005). Homology-dependent methylation in primate repetitive DNA. *Proceedings of the National Academy of Sciences*, 102(15):5471–5476. [55](#), [57](#), [72](#)
- [Minh et al., 2020] Minh, B. Q., Schmidt, H. A., Chernomor, O., Schrempf, D., Woodhams, M. D., von Haeseler, A., and Lanfear, R. (2020). IQ-TREE 2: New Models and Efficient Methods for Phylogenetic Inference in the Genomic Era. *Molecular Biology and Evolution*, 37(5):1530–1534. [146](#)
- [Mitchell et al., 2019] Mitchell, J. D., Allman, E. S., and Rhodes, J. A. (2019). Hypothesis testing near singularities and boundaries. *Electronic journal of statistics*, 13(1):2150. [71](#)
- [Morange, 2000] Morange, M. (2000). Georges Canguilhem et la biologie du XXe siècle/ Georges Canguilhem and twentieth-century biology. *Revue d'histoire des sciences*, 53(1):83–106. [13](#)
- [Moretti et al., 2014] Moretti, S., Laurency, B., Gharib, W. H., Castella, B., Kuzniar, A., Schabauer, H., Studer, R. A., Valle, M., Salamin, N., Stockinger, H., and Robinson-Rechavi, M. (2014). Selectome update: quality control and computational improvements to a database of positive selection. *Nucleic Acids Research*, 42(Database issue). [53](#), [54](#)
- [Mugal et al., 2020] Mugal, C. F., Kutschera, V. E., Botero-Castro, F., Wolf, J. B. W., and Kaj, I. (2020). Polymorphism Data Assist Estimation of the Nonsynonymous over Synonymous Fixation Rate Ratio for Closely Related Species. *Molecular Biology and Evolution*, 37(1):260–279. [169](#)
- [Muntané et al., 2018] Muntané, G., Farré, X., Rodríguez, J. A., Pegueroles, C., Hughes, D. A., de Magalhães, J. P., Gabaldón, T., and Navarro, A. (2018). Biological Processes Modulating Longevity across Primates: A Phylogenetic Genome-Phenome Analysis. *Molecular Biology and Evolution*, 35(8):1990–2004. [116](#)
- [Murrell et al., 2012a] Murrell, B., de Oliveira, T., Seebregts, C., Kosakovsky Pond, S. L., Scheffler, K., on behalf of the Southern African Treatment, and Consortium, R. N. S. (2012a). Modeling hiv-1 drug resistance as episodic directional selection. *PLoS Computational Biology*, 8(5). [33](#), [54](#), [55](#), [61](#)
- [Murrell et al., 2012b] Murrell, B., de Oliveira, T., Seebregts, C., Kosakovsky Pond, S. L., Scheffler, K., on behalf of the Southern African Treatment, and Consortium, R. N. S. (2012b). Modeling hiv-1 drug resistance as episodic directional selection. *PLoS Computational Biology*, 8(5). [198](#)
- [Murrell et al., 2015] Murrell, B., Weaver, S., Smith, M. D., Wertheim, J. O., Murrell, S., Aylward, A., Eren, K., Pollner, T., Martin, D. P., Smith, D. M., Scheffler, K., and Kosakovsky Pond, S. L. (2015). Gene-Wide Identification of Episodic Selection. *Molecular Biology and Evolution*, 32(5):1365–1371. [54](#)
- [Nagylaki, 1983] Nagylaki, T. (1983). Evolution of a finite population under gene conversion. *Proceedings of the National Academy of Sciences*, 80(20):6278–6281. [31](#)
- [Nelder and Mead, 1965] Nelder, J. A. and Mead, R. (1965). A Simplex Method for Function Minimization. *The Computer Journal*, 7(4):308–313. [71](#), [86](#)
- [Nelson et al., 2002] Nelson, G., Chandrashekar, J., Hoon, M. A., Feng, L., Zhao, G., Ryba, N. J. P., and Zuker, C. S. (2002). An amino-acid taste receptor. *Nature*, 416(6877):199–202. [165](#)

- [Nguyen et al., 2015] Nguyen, L.-T., Schmidt, H. A., von Haeseler, A., and Minh, B. Q. (2015). IQ-TREE: A Fast and Effective Stochastic Algorithm for Estimating Maximum-Likelihood Phylogenies. *Molecular Biology and Evolution*, 32(1):268–274. [33](#)
- [Nielsen, 2002] Nielsen, R. (2002). Mapping Mutations on Phylogenies. *Systematic Biology*, 51(5):729–739. [174](#)
- [Nielsen et al., 2006] Nielsen, R., Bauer DuMont, V. L., Hubisz, M. J., and Aquadro, C. F. (2006). Maximum Likelihood Estimation of Ancestral Codon Usage Bias Parameters in *Drosophila*. *Molecular Biology and Evolution*, 24(1):228–235. [25](#)
- [Nielsen et al., 2005] Nielsen, R., Bustamante, C., Clark, A. G., Glanowski, S., Sackton, T. B., Hubisz, M. J., Fledel-Alon, A., Tanenbaum, D. M., Civello, D., White, T. J., J. Sninsky, J., Adams, M. D., and Cargill, M. (2005). A Scan for Positively Selected Genes in the Genomes of Humans and Chimpanzees. *PLoS Biology*, 3(6). [54](#)
- [Parker et al., 2013] Parker, J., Tsagkogeorga, G., Cotton, J. A., Liu, Y., Provero, P., Stupka, E., and Rossiter, S. J. (2013). Genome-wide signatures of convergent evolution in echolocating mammals. *Nature*, 502(7470):228–231. [147](#), [148](#), [149](#), [215](#)
- [Partha et al., 2017] Partha, R., Chauhan, B. K., Ferreira, Z., Robinson, J. D., Lathrop, K., Nischal, K. K., Chikina, M., and Clark, N. L. (2017). Subterranean mammals show convergent regression in ocular genes and enhancers, along with adaptation to tunneling. *eLife*, 6. [16](#), [156](#), [157](#), [223](#)
- [Partha et al., 2019] Partha, R., Kowalczyk, A., Clark, N. L., and Chikina, M. (2019). Robust Method for Detecting Convergent Shifts in Evolutionary Rates. *Molecular Biology and Evolution*, 36(8):1817–1830. [53](#)
- [Parto and Lartillot, 2017] Parto, S. and Lartillot, N. (2017). Detecting consistent patterns of directional adaptation using differential selection codon models. *BMC Evolutionary Biology*, 17(1):147. [26](#), [29](#), [51](#), [164](#), [166](#)
- [Parto and Lartillot, 2018] Parto, S. and Lartillot, N. (2018). Molecular adaptation in rubisco: Discriminating between convergent evolution and positive selection using mechanistic and classical codon models. *PLOS ONE*, 13(2). [16](#), [26](#), [29](#), [33](#), [54](#), [55](#), [58](#), [67](#), [68](#), [75](#), [116](#)
- [Paszke et al., 2019] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc. [71](#), [105](#)
- [Patwa and Wahl, 2008] Patwa, Z. and Wahl, L. (2008). The fixation probability of beneficial mutations. *Journal of The Royal Society Interface*, 5(28):1279–1289. [24](#)
- [Penn et al., 2008] Penn, O., Stern, A., Rubinstein, N. D., Dutheil, J., Bacharach, E., Galtier, N., and Pupko, T. (2008). Evolutionary modeling of rate shifts reveals specificity determinants in HIV-1 subtypes. *PLoS computational biology*, 4(11). [53](#)
- [Perron et al., 2019] Perron, U., Kozlov, A. M., Stamatakis, A., Goldman, N., and Moal, I. H. (2019). Modeling structural constraints on protein evolution via side-chain conformational states. *Molecular Biology and Evolution*, 36(9):2086–2103. [174](#)

- [Pečerska et al., 2021] Pečerska, J., Gil, M., and Anisimova, M. (2021). Joint Alignment and Tree Inference. preprint, Bioinformatics. 169
- [Pouyet et al., 2016] Pouyet, F., Bailly-Bechet, M., Mouchiroud, D., and Guéguen, L. (2016). SENCA: A Multilayered Codon Model to Study the Origins and Dynamics of Codon Usage. *Genome Biology and Evolution*, 8(8):2427–2441. 65
- [Prud’homme et al., 2007] Prud’homme, B., Gompel, N., and Carroll, S. B. (2007). Emerging principles of regulatory evolution. *Proceedings of the National Academy of Sciences*, 104(suppl_1):8605–8612. 167
- [Pupko and Galtier, 2002] Pupko, T. and Galtier, N. (2002). A covarion-based method for detecting molecular adaptation: application to the evolution of primate mitochondrial genomes. *Proceedings. Biological Sciences*, 269(1498):1313–1316. 53
- [R Core Team, 2021] R Core Team (2021). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 73
- [Ratnakumar et al., 2010] Ratnakumar, A., Mousset, S., Glémin, S., Berglund, J., Galtier, N., Duret, L., and Webster, M. T. (2010). Detecting positive selection within genomes: the problem of biased gene conversion. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 365(1552):2571–2580. 55, 57, 64, 67, 172
- [Rey et al., 2018] Rey, C., Guéguen, L., Sémon, M., and Boussau, B. (2018). Accurate Detection of Convergent Amino-Acid Evolution with PCOC. *Molecular Biology and Evolution*, 35(9):2296–2306. 51, 58
- [Rey et al., 2019] Rey, C., Lanore, V., Veber, P., Guéguen, L., Lartillot, N., Sémon, M., and Boussau, B. (2019). Detecting adaptive convergent amino acid evolution. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 374(1777). 16, 26, 31, 33, 43, 44, 50, 51, 58, 61, 65, 72, 75, 117, 194
- [Risso et al., 2017] Risso, V. A., Martinez-Rodriguez, S., Candel, A. M., Krüger, D. M., Pantoja-Uceda, D., Ortega-Muñoz, M., Santoyo-Gonzalez, F., Gaucher, E. A., Kamerlin, S. C. L., Bruix, M., Gavira, J. A., and Sanchez-Ruiz, J. M. (2017). De novo active sites for resurrected Precambrian enzymes. *Nature Communications*, 8. 53
- [Rivas and Eddy, 2008] Rivas, E. and Eddy, S. R. (2008). Probabilistic Phylogenetic Inference with Insertions and Deletions. *PLoS Computational Biology*, 4(9). 173
- [Rodrigue, 2013] Rodrigue, N. (2013). On the Statistical Interpretation of Site-Specific Variables in Phylogeny-Based Substitution Models. *Genetics*, 193(2):557–564. 55, 69
- [Rodrigue and Lartillot, 2016] Rodrigue, N. and Lartillot, N. (2016). Detecting adaptation in protein-coding genes using a bayesian site-heterogeneous mutation-selection codon substitution model. *Molecular Biology and Evolution*, 34(1):11. 27
- [Rodrigue and Lartillot, 2017] Rodrigue, N. and Lartillot, N. (2017). Detecting Adaptation in Protein-Coding Genes Using a Bayesian Site-Heterogeneous Mutation-Selection Codon Substitution Model. *Molecular Biology and Evolution*, 34(1):204–214. 55

- [Rodrigue et al., 2020] Rodrigue, N., Latrille, T., and Lartillot, N. (2020). A Bayesian Mutation–Selection Framework for Detecting Site-Specific Adaptive Evolution in Protein-Coding Genes. *Molecular Biology and Evolution*, 38(3):1199–1208. [55](#)
- [Rodrigue et al., 2008] Rodrigue, N., Philippe, H., and Lartillot, N. (2008). Uniformization for sampling realizations of Markov processes: applications to Bayesian implementations of codon substitution models. *Bioinformatics*, 24(1):56–62. [174](#)
- [Rodrigue et al., 2010] Rodrigue, N., Philippe, H., and Lartillot, N. (2010). Mutation-selection models of coding sequence evolution with site-heterogeneous amino acid fitness profiles. *Proceedings of the National Academy of Sciences*, 107(10):4629–4634. [26](#), [29](#), [54](#), [55](#), [117](#)
- [Rousselle et al., 2019] Rousselle, M., Laverré, A., Figuet, E., Nabholz, B., and Galtier, N. (2019). Influence of Recombination and GC-biased Gene Conversion on the Adaptive and Nonadaptive Substitution Rate in Mammals versus Birds. *Molecular Biology and Evolution*, 36(3):458–471. [55](#), [67](#)
- [Rubinstein et al., 2011] Rubinstein, N. D., Doron-Faigenboim, A., Mayrose, I., and Pupko, T. (2011). Evolutionary models accounting for layers of selection in protein-coding genes and their impact on the inference of positive selection. *Molecular Biology and Evolution*, 28(12):3297–3308. [65](#)
- [Saputra et al., 2021] Saputra, E., Kowalczyk, A., Cusick, L., Clark, N., and Chikina, M. (2021). Phylogenetic Permutations: A Statistically Rigorous Approach to Measure Confidence in Associations in a Phylogenetic Context. *Molecular Biology and Evolution*, 38(7):3004–3021. [168](#)
- [Saunders and Green, 2007] Saunders, C. T. and Green, P. (2007). Insights from Modeling Protein Evolution with Context-Dependent Mutation and Asymmetric Amino Acid Selection. *Molecular Biology and Evolution*, 24(12):2632–2647. [55](#)
- [Schabauer et al., 2012] Schabauer, H., Valle, M., Pacher, C., Stockinger, H., Stamatakis, A., Robinson-Rechavi, M., Yang, Z., and Salamin, N. (2012). Slimcodeml: An optimized version of codeml for the branch-site model. In *2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum*, page 706–714, Shanghai, China. IEEE. [88](#), [164](#)
- [Scornavacca et al., 2019] Scornavacca, C., Belkhir, K., Lopez, J., Dernas, R., Delsuc, F., Douzery, E. J. P., and Ranwez, V. (2019). OrthoMaM v10: Scaling-Up Orthologous Coding Sequence and Exon Alignments with More than One Hundred Mammalian Genomes. *Molecular Biology and Evolution*, 36(4):861–862. [33](#), [61](#), [120](#), [146](#), [196](#)
- [Shen et al., 2012] Shen, Y.-Y., Liang, L., Li, G.-S., Murphy, R. W., and Zhang, Y.-P. (2012). Parallel Evolution of Auditory Genes for Echolocation in Bats and Toothed Whales. *PLoS Genetics*, 8(6). [147](#), [148](#), [149](#), [215](#)
- [Shi and Zhang, 2006] Shi, P. and Zhang, J. (2006). Contrasting Modes of Evolution Between Vertebrate Sweet/Umami Receptor Genes and Bitter Receptor Genes. *Molecular Biology and Evolution*, 23(2):292–300. [165](#)
- [Smith et al., 2019] Smith, P., Godde, N., Rubio, S., Tekeste, M., Vladar, E. K., Axelrod, J. D., Henderson, D. J., Milgrom-Hoffman, M., Humbert, P. O., and Hinck, L. (2019). VANGL2 regulates luminal epithelial organization and cell turnover in the mammary gland. *Scientific Reports*, 9(1):7079. [220](#)

- [Spielman and Wilke, 2015] Spielman, S. J. and Wilke, C. O. (2015). The Relationship between dN/dS and Scaled Selection Coefficients. *Molecular Biology and Evolution*, 32(4):1097–1108. [54](#), [65](#), [66](#)
- [Spielman and Wilke, 2016] Spielman, S. J. and Wilke, C. O. (2016). Extensively Parameterized Mutation–Selection Models Reliably Capture Site-Specific Selective Constraint. *Molecular Biology and Evolution*, 33(11):2990–3002. [55](#), [65](#), [69](#)
- [Studer et al., 2008] Studer, R. A., Penel, S., Duret, L., and Robinson-Rechavi, M. (2008). Pervasive positive selection on duplicated and nonduplicated vertebrate protein coding genes. *Genome Research*, 18(9):1393–1402. [54](#)
- [Sugawara et al., 2010] Sugawara, T., Imai, H., Nikaido, M., Imamoto, Y., and Okada, N. (2010). Vertebrate Rhodopsin Adaptation to Dim Light via Rapid Meta-II Intermediate Formation. *Molecular Biology and Evolution*, 27(3):506–519. [159](#), [225](#)
- [Sun et al., 2017] Sun, X., Zhang, Z., Sun, Y., Li, J., Xu, S., and Yang, G. (2017). Comparative genomics analyses of alpha-keratins reveal insights into evolutionary adaptation of marine mammals. *Frontiers in Zoology*, 14(1):41. [154](#)
- [Suzuki et al., 2009] Suzuki, Y., Gojobori, T., and Kumar, S. (2009). Methods for Incorporating the Hypermutability of CpG Dinucleotides in Detecting Natural Selection Operating at the Amino Acid Sequence Level. *Molecular Biology and Evolution*, 26(10):2275–2284. [55](#)
- [Tamuri, 2021] Tamuri, A. U. (2021). A mutation–selection model of protein evolution under persistent positive selection. *Molecular Biology and Evolution*, 39(1):9. [29](#), [32](#), [56](#), [57](#), [64](#), [68](#), [69](#), [73](#)
- [Tamuri et al., 2012] Tamuri, A. U., dos Reis, M., and Goldstein, R. A. (2012). Estimating the distribution of selection coefficients from phylogenetic data using sitewise mutation–selection models. *Genetics*, 190(3):1101–1115. [25](#), [26](#), [29](#), [54](#), [55](#), [117](#), [164](#)
- [Tamuri et al., 2009] Tamuri, A. U., dos Reis, M., Hay, A. J., and Goldstein, R. A. (2009). Identifying changes in selective constraints: Host shifts in influenza. *PLoS Computational Biology*, 5(11). [16](#), [17](#), [33](#), [51](#), [53](#), [55](#), [57](#), [58](#), [61](#), [66](#), [71](#), [75](#), [80](#), [116](#), [118](#), [164](#), [199](#), [207](#), [208](#)
- [Tamuri et al., 2014] Tamuri, A. U., Goldman, N., and Reis, M. d. (2014). A Penalized Likelihood Method for Estimating the Distribution of Selection Coefficients from Phylogenetic Data. *Genetics*. [55](#), [69](#)
- [Tang et al., 2022] Tang, G., Ma, C., Li, L., Zhang, S., Li, F., Wu, J., Yin, Y., Zhu, Q., Liang, Y., Wang, R., Huang, H., Zhao, T.-J., Yang, H., Li, P., and Chen, F.-J. (2022). PITPNC1 promotes the thermogenesis of brown adipose tissue under acute cold exposure. *Science China Life Sciences*, 65(11):2287–2300. [159](#), [225](#)
- [Tavaré et al., 1986] Tavaré, S. et al. (1986). Some probabilistic and statistical problems in the analysis of dna sequences. *Lectures on mathematics in the life sciences*, 17(2):57–86. [27](#), [80](#), [117](#)
- [Teufel et al., 2018] Teufel, A., Ritchie, A., Wilke, C., and Liberles, D. (2018). Using the mutation–selection framework to characterize selection on protein sequences. *Genes*, 9(8):409. [25](#)
- [Thiltgen et al., 2017] Thiltgen, G., dos Reis, M., and Goldstein, R. A. (2017). Finding Direction in the Search for Selection. *Journal of Molecular Evolution*, 84(1):39–50. [70](#)

- [Thorne et al., 2007] Thorne, J. L., Choi, S. C., Yu, J., Higgs, P. G., and Kishino, H. (2007). Population genetics without intraspecific data. *Molecular Biology and Evolution*, 24(8):1667–1677. [27](#)
- [Tian et al., 2019] Tian, Y., Wen, H., Qi, X., Zhang, X., and Li, Y. (2019). Identification of mapk gene family in *Lateolabrax maculatus* and their expression profiles in response to hypoxia and salinity challenges. *Gene*, 684:20–29. [220](#)
- [Tsai et al., 2015] Tsai, C.-H., Li, C.-H., Liao, P.-L., Cheng, Y.-W., Lin, C.-H., Huang, S.-H., and Kang, J.-J. (2015). NcoA2-Dependent Inhibition of HIF-1 Activation Is Regulated via AhR. *Toxicological Sciences*, 148(2):517–530. [220](#)
- [Uffelmann et al., 2021] Uffelmann, E., Huang, Q. Q., Munung, N. S., de Vries, J., Okada, Y., Martin, A. R., Martin, H. C., Lappalainen, T., and Posthuma, D. (2021). Genome-wide association studies. *Nature Reviews Methods Primers*, 1(1):59. [38](#)
- [Valle et al., 2014] Valle, M., Schabauer, H., Pacher, C., Stockinger, H., Stamatakis, A., Robinson-Rechavi, M., and Salamin, N. (2014). Optimization strategies for fast detection of positive selection on phylogenetic trees. *Bioinformatics*, 30(8):1129–1137. [164](#)
- [Veber, 2017] Veber, P. (2017). pveber/bistro: bistro 0.3.0. [36](#)
- [von Kobbe et al., 2003] von Kobbe, C., Thomä, N. H., Czyzewski, B. K., Pavletich, N. P., and Bohr, V. A. (2003). Werner Syndrome Protein Contains Three Structure-specific DNA Binding Domains. *Journal of Biological Chemistry*, 278(52). [124](#)
- [Webb and Zhang, 2005] Webb, D. M. and Zhang, J. (2005). FoxP2 in Song-Learning Birds and Vocal-Learning Mammals. *Journal of Heredity*, 96(3):212–216. [159](#)
- [Wertheim et al., 2015] Wertheim, J. O., Murrell, B., Smith, M. D., Kosakovsky Pond, S. L., and Scheffler, K. (2015). RELAX: Detecting Relaxed Selection in a Phylogenetic Framework. *Molecular Biology and Evolution*, 32(3):820–832. [68](#)
- [Whelan and Goldman, 2001] Whelan, S. and Goldman, N. (2001). A general empirical model of protein evolution derived from multiple protein families using a maximum-likelihood approach. *Molecular Biology and Evolution*, 18(5):691–699. [78](#), [81](#), [117](#)
- [Wickham, 2016] Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. [73](#)
- [Wilkinson, 1951] Wilkinson, B. (1951). A statistical consideration in psychological research. *Psychological bulletin*, 48(3):156–158. [132](#)
- [Wilks, 1938] Wilks, S. S. (1938). The Large-Sample Distribution of the Likelihood Ratio for Testing Composite Hypotheses. *The Annals of Mathematical Statistics*, 9(1):60 – 62. [82](#), [120](#), [207](#)
- [Wilman et al., 2014] Wilman, H., Belmaker, J., Simpson, J., de la Rosa, C., Rivadeneira, M. M., and Jetz, W. (2014). EltonTraits 1.0: Species-level foraging attributes of the world’s birds and mammals: *Ecological Archives* E095-178. *Ecology*, 95(7):2027–2027. [120](#)
- [Wilson et al., 2011] Wilson, D. J., Hernandez, R. D., Andolfatto, P., and Przeworski, M. (2011). A Population Genetics-Phylogenetics Approach to Inferring Natural Selection in Coding Sequences. *PLoS Genetics*, 7(12). [169](#)

- [Wirthlin et al., 2022] Wirthlin, M. E., Schmid, T. A., Elie, J. E., Zhang, X., Shvareva, V. A., Rakuljic, A., Ji, M. B., Bhat, N. S., Kaplow, I. M., Schäffer, D. E., Lawler, A. J., Annaldasula, S., Lim, B., Azim, E., Zoonomia Consortium, Meyer, W. K., Yartsev, M. M., and Pfenning, A. R. (2022). Vocal learning-associated convergent evolution in mammalian proteins and regulatory elements. preprint. [147](#), [159](#), [165](#), [215](#)
- [Wright, 1931] Wright, S. (1931). Evolution in mendelian populations. *Genetics*, 16(2):97–159. [24](#), [25](#)
- [Wu, 2022] Wu, Y. (2022). Diet evolution of carnivorous and herbivorous mammals in Laurasiatheria. *BMC Ecology and Evolution*, 22(1):82. [151](#), [152](#), [217](#)
- [Yang, 2006] Yang, Z. (2006). *Computational Molecular Evolution*. Oxford University Press. [71](#), [84](#), [88](#)
- [Yang, 2007] Yang, Z. (2007). PAML 4: Phylogenetic Analysis by Maximum Likelihood. *Molecular Biology and Evolution*, 24(8):1586–1591. [51](#), [54](#), [57](#), [78](#), [116](#)
- [Yang and Nielsen, 2008] Yang, Z. and Nielsen, R. (2008). Mutation-selection models of codon substitution and their use to estimate selective strengths on codon usage. *Molecular Biology and Evolution*, 25(3):568–579. [25](#), [26](#), [27](#), [29](#), [30](#), [53](#), [54](#), [55](#), [65](#), [117](#)
- [Yang et al., 2005] Yang, Z., Wong, W., and Nielsen, R. (2005). Bayes empirical bayes inference of amino acid sites under positive selection. *Molecular Biology and Evolution*, 22(4):1107–1118. [54](#), [57](#), [70](#)
- [Yu, 2020] Yu, G. (2020). Using ggtree to visualize data on tree-like structures. *Current Protocols in Bioinformatics*, 69(1). [73](#)
- [Zeldovich et al., 2005] Zeldovich, K. B., Berezovsky, I. N., and Shakhnovich, E. I. (2005). Protein and DNA sequence determinants of thermophilic adaptation. *PLoS Computational Biology*, preprint(2006). [112](#)
- [Zhang et al., 2014] Zhang, G., Li, C., Li, Q., Li, B., Larkin, D. M., Lee, C., Storz, J. F., Antunes, A., Greenwold, M. J., Meredith, R. W., Ödeen, A., Cui, J., Zhou, Q., Xu, L., Pan, H., Wang, Z., Jin, L., Zhang, P., Hu, H., Yang, W., Hu, J., Xiao, J., Yang, Z., Liu, Y., Xie, Q., Yu, H., Lian, J., Wen, P., Zhang, F., Li, H., Zeng, Y., Xiong, Z., Liu, S., Zhou, L., Huang, Z., An, N., Wang, J., Zheng, Q., Xiong, Y., Wang, G., Wang, B., Wang, J., Fan, Y., da Fonseca, R. R., Alfaro-Núñez, A., Schubert, M., Orlando, L., Mourier, T., Howard, J. T., Ganapathy, G., Pfenning, A., Whitney, O., Rivas, M. V., Hara, E., Smith, J., Farré, M., Narayan, J., Slavov, G., Romanov, M. N., Borges, R., Machado, J. P., Khan, I., Springer, M. S., Gatesy, J., Hoffmann, F. G., Opazo, J. C., Håstad, O., Sawyer, R. H., Kim, H., Kim, K.-W., Kim, H. J., Cho, S., Li, N., Huang, Y., Bruford, M. W., Zhan, X., Dixon, A., Bertelsen, M. F., Derryberry, E., Warren, W., Wilson, R. K., Li, S., Ray, D. A., Green, R. E., O’Brien, S. J., Griffin, D., Johnson, W. E., Haussler, D., Ryder, O. A., Willerslev, E., Graves, G. R., Alström, P., Fjeldså, J., Mindell, D. P., Edwards, S. V., Braun, E. L., Rahbek, C., Burt, D. W., Houde, P., Zhang, Y., Yang, H., Wang, J., Avian Genome Consortium, Jarvis, E. D., Gilbert, M. T. P., and Wang, J. (2014). Comparative genomics reveals insights into avian genome evolution and adaptation. *Science (New York, N.Y.)*, 346(6215):1311–1320. [53](#), [54](#)

- [Zhang et al., 2005] Zhang, J., Nielsen, R., and Yang, Z. (2005). Evaluation of an improved branch-site likelihood method for detecting positive selection at the molecular level. *Molecular Biology and Evolution*, 22(12):2472–2479. [54](#), [57](#), [70](#)
- [Zhao et al., 2012] Zhao, H., Xu, D., Zhang, S., and Zhang, J. (2012). Genomic and Genetic Evidence for the Loss of Umami Taste in Bats. *Genome Biology and Evolution*, 4(1):73–79. [165](#)
- [Zhen et al., 2012] Zhen, Y., Aardema, M. L., Medina, E. M., Schumer, M., and Andolfatto, P. (2012). Parallel Molecular Evolution in an Herbivore Community. *Science (New York, N.Y.)*, 337(6102):1634–1637. [53](#)
- [Zhou et al., 2015] Zhou, X., Seim, I., and Gladyshev, V. N. (2015). Convergent evolution of marine mammals is associated with distinct substitutions in common genes. *Scientific Reports*, 5(1). [153](#), [154](#), [222](#)
- [Zhou and Stephens, 2012] Zhou, X. and Stephens, M. (2012). Genome-wide efficient mixed-model analysis for association studies. *Nature Genetics*, 44(7):821–824. [40](#), [58](#)
- [Zhu et al., 2014] Zhu, K., Zhou, X., Xu, S., Sun, D., Ren, W., Zhou, K., and Yang, G. (2014). The loss of taste genes in cetaceans. *BMC Evolutionary Biology*, 14(1):218. [165](#)

Appendices

A	Summarized remarks on the usage of Pelican	192
B	Empirical phylogenies	194
C	Evaluation of methods: supplementary material	200
C.1	Synthetic trees	201
C.2	Study of the calibration of the methods	204
C.3	Benchmark results with confounding factors	205
C.4	Evaluation of Pelican using different degrees of freedom in the LRT	207
D	Gene aggregation	209
D.1	Null distribution for the Gene-wise Truncated Fisher (GTF) method	210
D.2	Mixture model: derivation of EM equations	211
E	Scans of Orthomam for genotype-phenotype associations	214
E.1	Echolocation	215
E.2	Diet	217
E.3	Adaptation to life in aquatic environments	219
E.4	Adaptation to life in marine environments	221
E.5	Adaptation to life in subterranean environments	223
E.6	Diurnality and nocturnality	225
E.7	Vocal learning	227
E.8	Domestication	229
F	Other work	231

Appendix A

Summarized remarks on the usage of Pelican

Dataset features Pelican detects genotype-phenotype associations within coding sequences by comparing two alternative models (4.2.3) using a likelihood ratio test (4.2.4). This test relies on an asymptotic condition that requires the sample size (here, the number of sequences) to be large enough for it to be accurate. Because we work in a phylogenetic context, sequences are not independent observations; this correlation tends to decrease when the divergence time between them increases. The interaction between these two elements constitutes an effective sample size, that should ideally be as high as possible: trees both large and long are expected to give the best results (4.4). We have also shown evidence that trait annotations with a single transition might be problematic (6.2.4), even when the effective sample size is large, and advise for caution when working with such phylogenies — particularly regarding false positives.

Like any other approach in comparative genomics, the quality of results obtained from Pelican is sensitive to the quality of the alignment (7.3.2), that should be carefully checked beforehand. The same goes for the phylogeny (7.3.3), and the afferent phenotype annotation (7.3.4), that are also expected to determine the confidence that can be put into the predictions. Special care should be devoted to reviewing trait annotations, and in the case of discrete traits the number of relevant categories should be considered: one case have been reported where three categories were more realistic than two, even though it was not immediately obvious, and improved the quality of results.

Gene-level predictions (6) Because of the difficulties stemming from the lack of calibration of Pelican, there is no completely satisfying solution to make gene-level predictions yet. Nonetheless, it appears that the GTF method (6.2.2) can be recommended, as it gives predictions reliable enough, both on simulations and empirical data (6.4). It is however not well calibrated, thus making the application of statistical tools to compute false discovery rates at the gene level unreliable. Importantly, because constant sites in alignments are not tested, they must not be included in the gene aggregation. An R package implementation of GTF is available at <https://github.com/lsdch/gtfisher>. Example usage (note the filtering clause for non constant sites):

```
site_pvals = readr::read_tsv("/path/to/pelican/output/all_sites.tsv")
gene_pvals = gtfisher::gtf_predict(site_pvals, alignment, aagtr_pval, naa > 1)
```

Multinomial filter feature It is provided as a way to quickly scan large datasets for sites associated to a phenotype, at the expense of a loss in sensitivity (4.6). In this setting, it is discouraged to directly attempt aggregation of site results at the level of genes, because only a few sites are actually tested using Pelican. Instead, a strategy that consists in identifying a subset of best candidate sites in a filtered run, perform an unfiltered scan of genes that they belong to, and then aggregate predictions on these genes seems like a reasonable approach. Recommended values for the filtering threshold range from $1e^{-3}$ to $1e^{-1}$ depending on the desired stringency and speed increase, but should be adapted to each individual case.

Appendix B

Empirical phylogenies

Empirical phylogenies as first described in [chapter one](#), then used for simulating datasets so that performance of detection methods can be evaluated in [chapter two](#) and [chapter three](#).



Figure B.1: Rodents tree annotated with adaptation to life in arid environments [Rey et al., 2019]

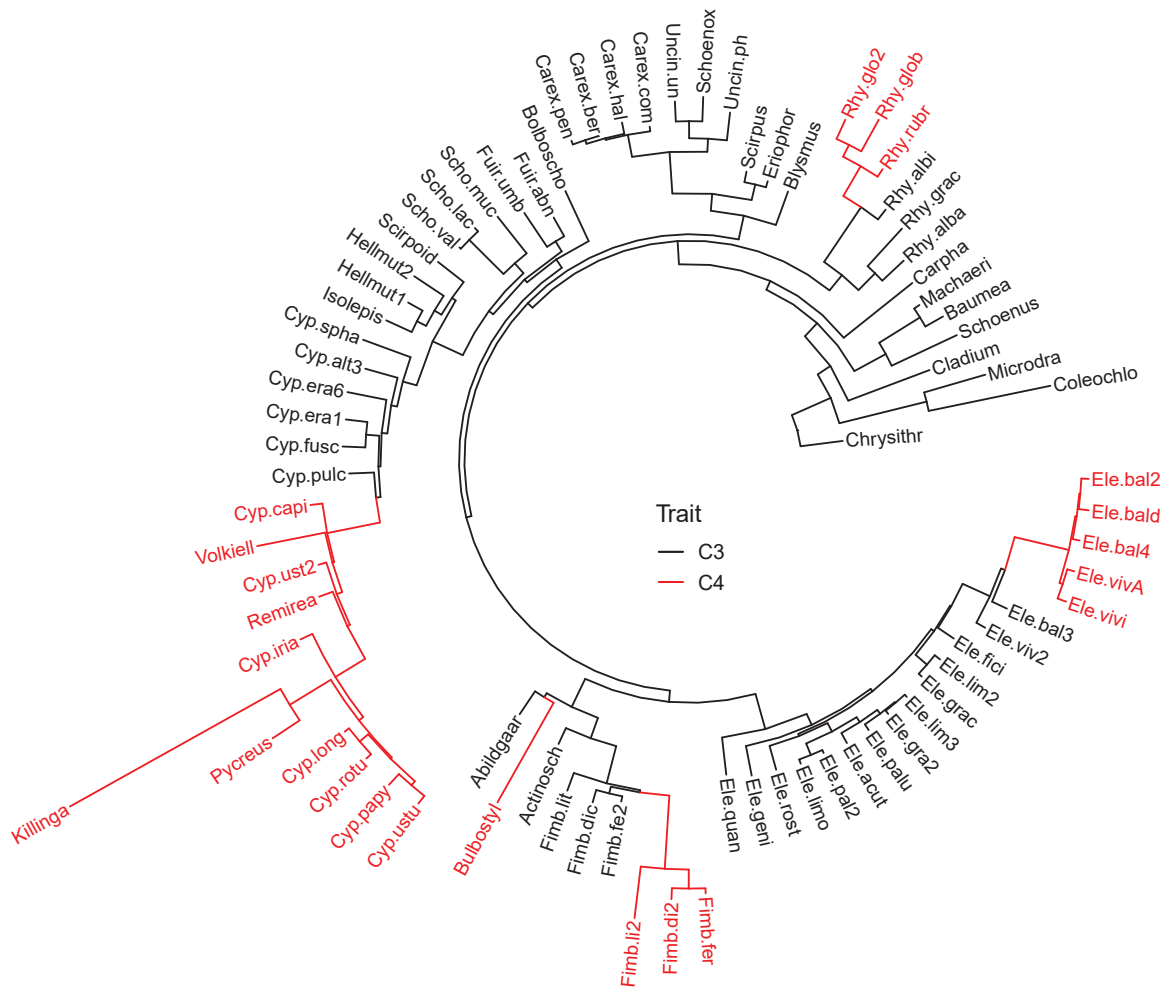


Figure B.2: *Cyperaceae* tree annotated with photosynthesis type [Besnard et al., 2009]

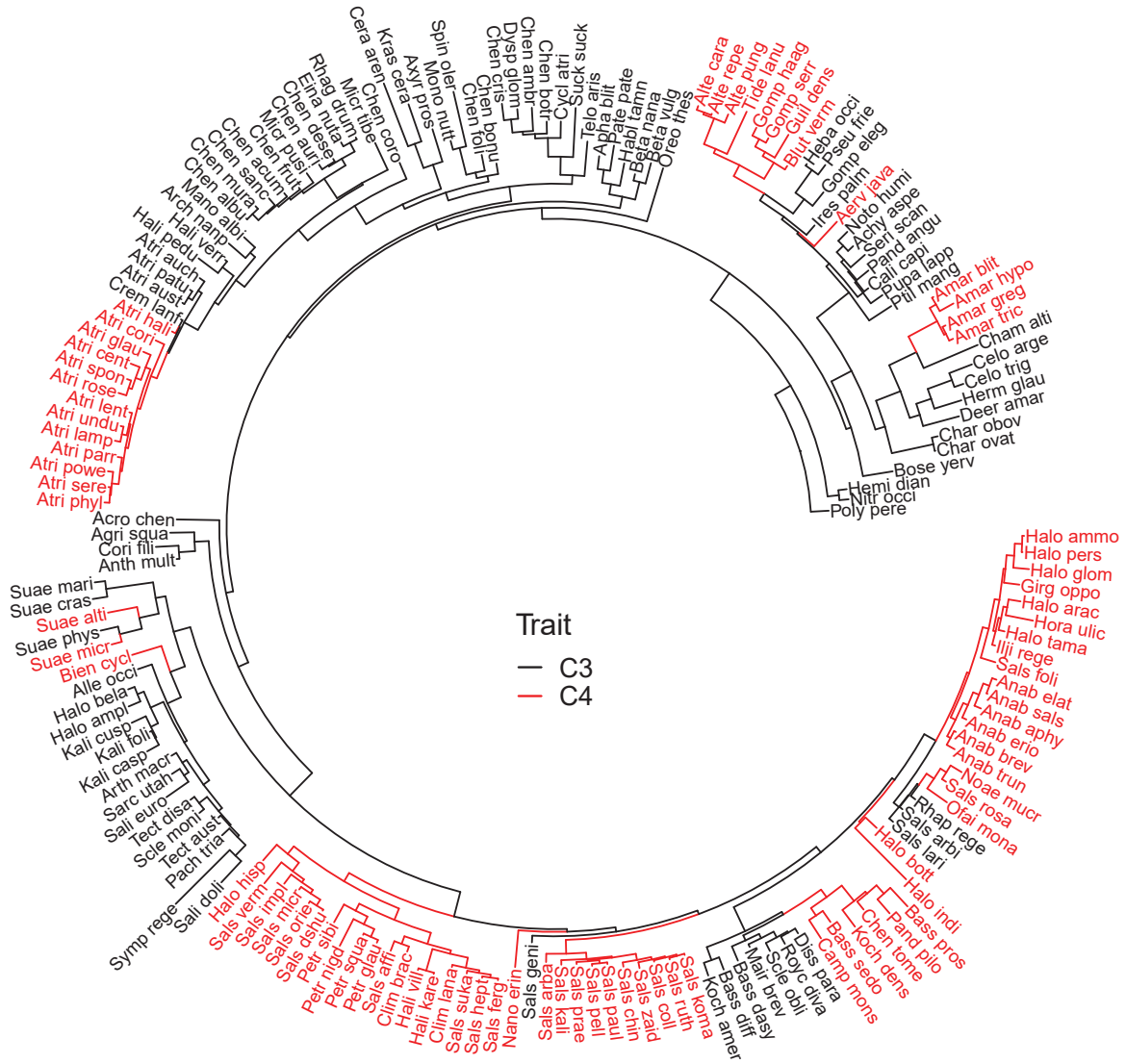


Figure B.4: *Amaranthaceae* tree annotated with photosynthesis type [Kapralov et al., 2012]

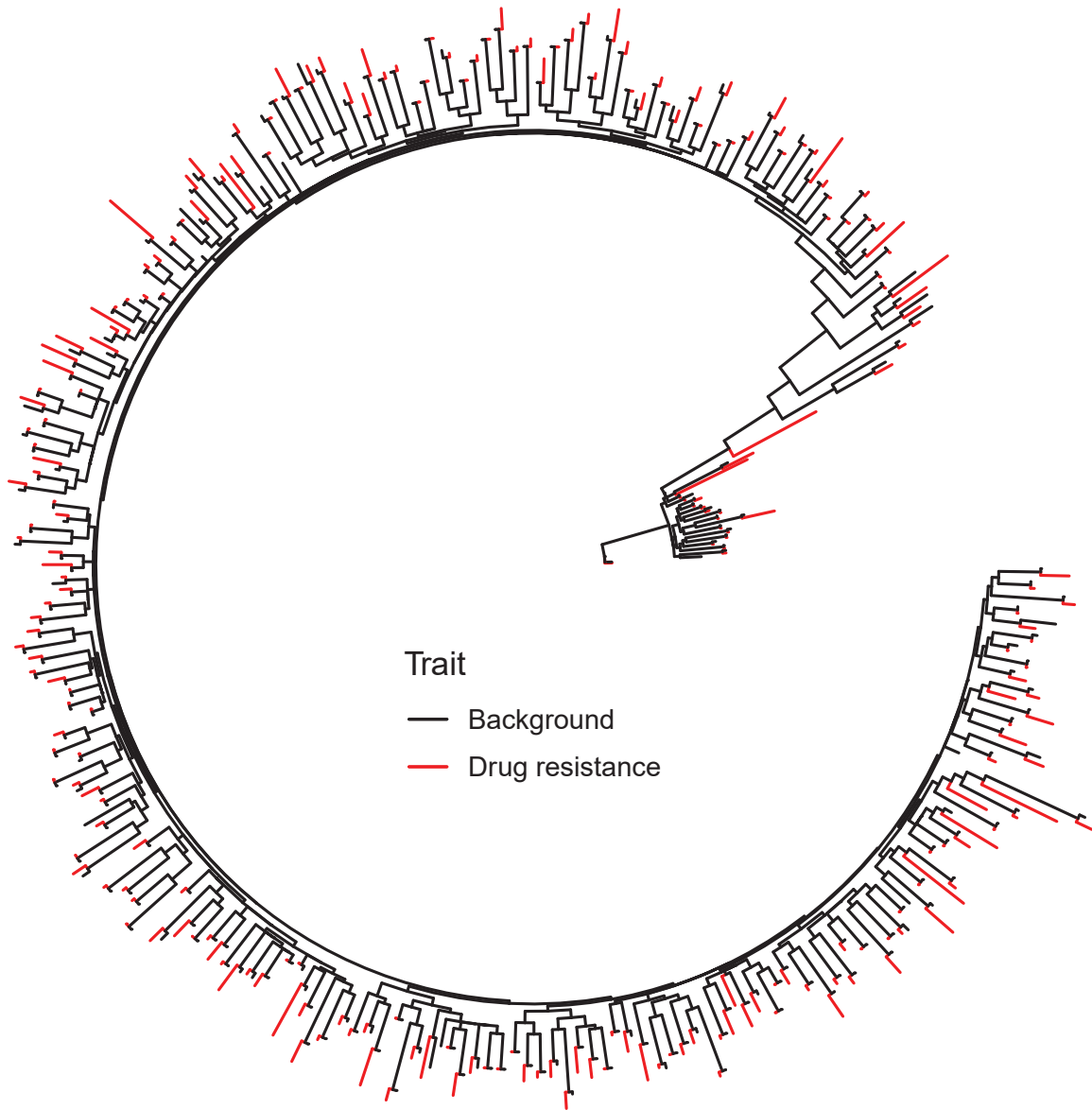


Figure B.5: HIV Reverse-transcriptase tree with annotation for drug resistance treatment [Murrell et al., 2012b]

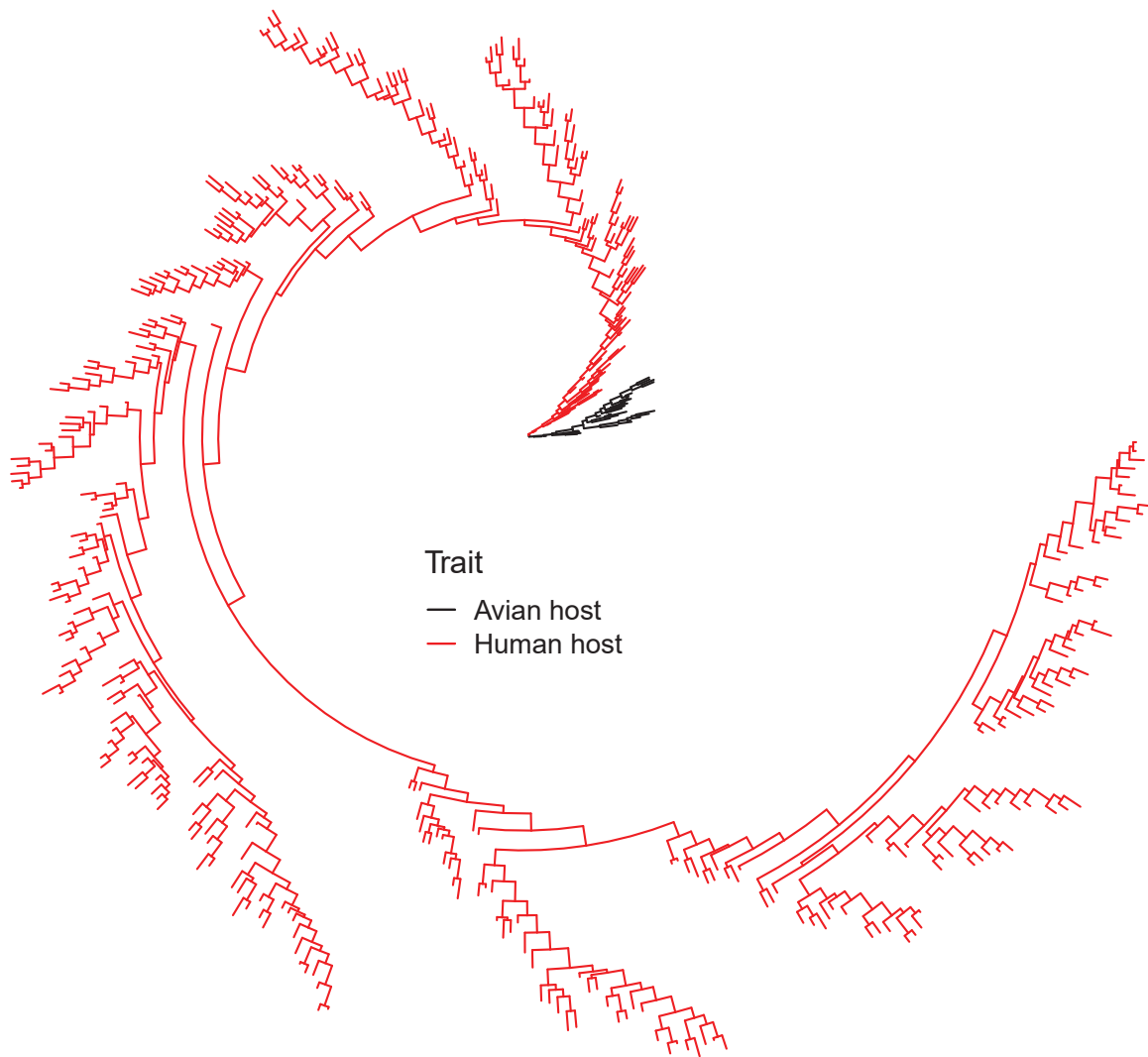


Figure B.6: Influenza H1 segment tree annotated with host species [Tamuri et al., 2009]

Appendix C

Evaluation of methods: supplementary material

Contents

C.1 Synthetic trees	201
C.2 Study of the calibration of the methods	204
C.3 Benchmark results with confounding factors	205
C.4 Evaluation of Pelican using different degrees of freedom in the LRT	207

C.1 Synthetic trees

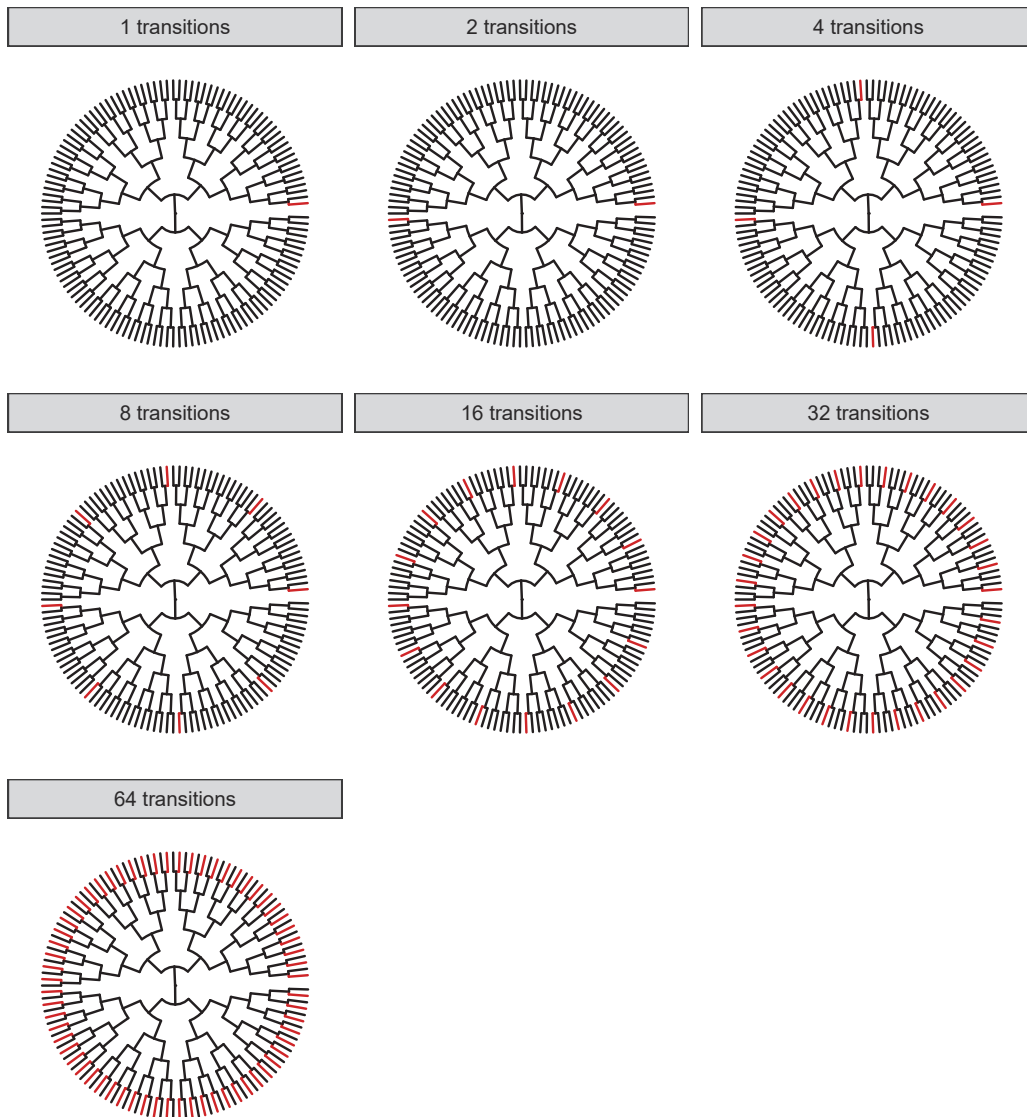


Figure C.1: Synthetic trees with variable number of transitions on terminal branches

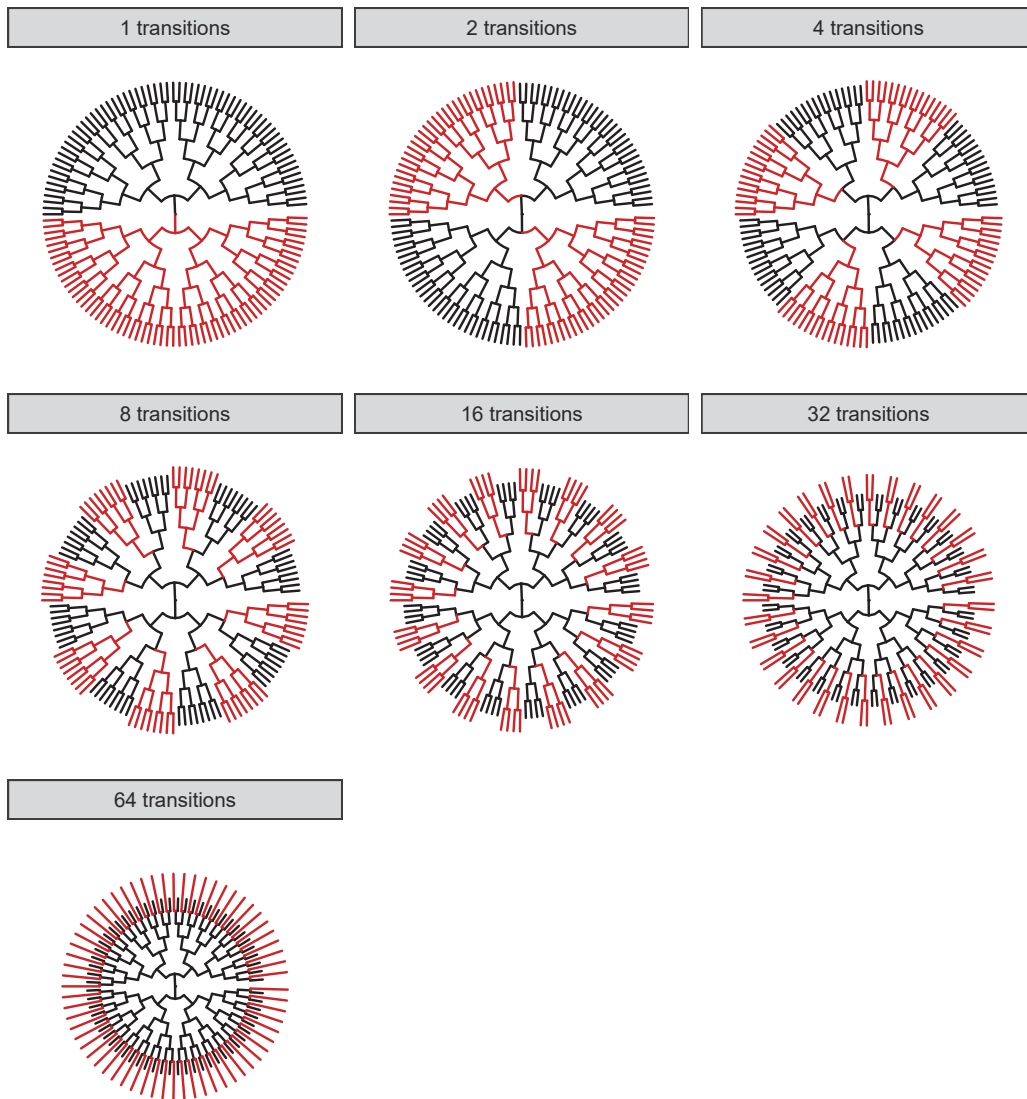


Figure C.2: Synthetic trees ensuring a constant amount of time is spent in a condition

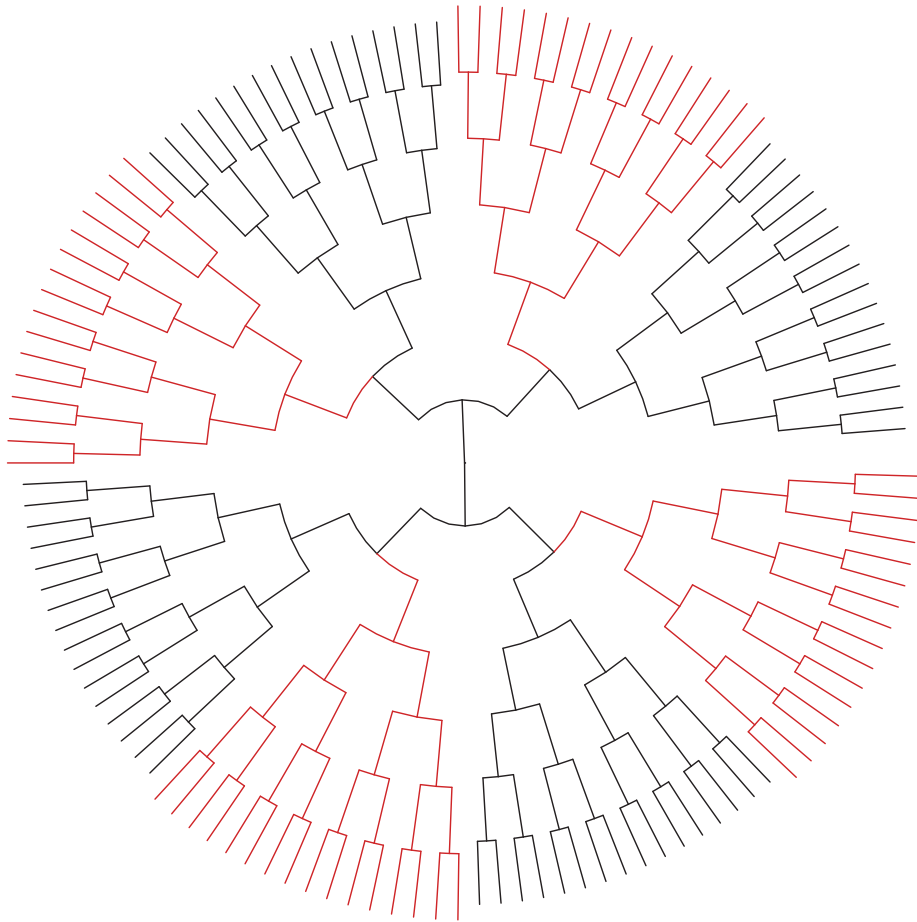


Figure C.3: Tree topology and annotation used in the experiment where variation in branch lengths is investigated.

C.2 Study of the calibration of the methods

Dataset	Diffsel	Gemma	Multinomial	PCOC	Pelican	TDG09	codeml
<i>Cyperaceae</i>	0.00	0.02	0.07	0.00	0.02	0.03	0.00
<i>Amaranthaceae</i>	0.00	0.00	0.05	0.00	0.01	0.02	0.00
Rodents	0.00	0.00	0.00	0.00	0.00	0.01	0.00
Echolocation	0.00	0.01	0.02	0.00	0.01	0.02	0.00
HIV	NA	0.00	0.00	NA	0.01	0.01	0.00
Influenza	NA	0.04	0.23	0.01	0.05	NA	0.02

Table C.1: Observed false positive rate at the 0.05 threshold. A well calibrated method should yield a proportion of 0.05 false positives.

Dataset	Diffsel	Gemma	Multinomial	PCOC	Pelican	TDG09	codeml
<i>Cyperaceae</i>	0.00	0.08	0.24	0.00	0.06	0.12	0.00
<i>Amaranthaceae</i>	0.00	0.02	0.24	NA	0.02	0.10	0.00
Rodents	0.00	0.02	0.01	0.00	0.02	0.11	0.00
Echolocation	0.00	0.05	0.10	0.00	0.05	0.10	0.00
HIV	NA	0.00	0.00	NA	0.02	0.04	0.00
Influenza	NA	0.07	0.41	0.01	0.09	NA	0.25

Table C.2: Observed false positive rate at the 0.05 threshold, after removing constant sites from the data. A well calibrated method should yield a proportion of 0.05 false positives.

C.3 Benchmark results with confounding factors

Evaluations on empirical phylogenies in the presence of confounding factors are presented here. They include persistent positive selection (PPS) and non selective forces: GC-biased gene conversion (gBGC) and CpG hypermutability. HIV and Influenza datasets were excluded from this analysis, because of their larger size and the associated computational costs.

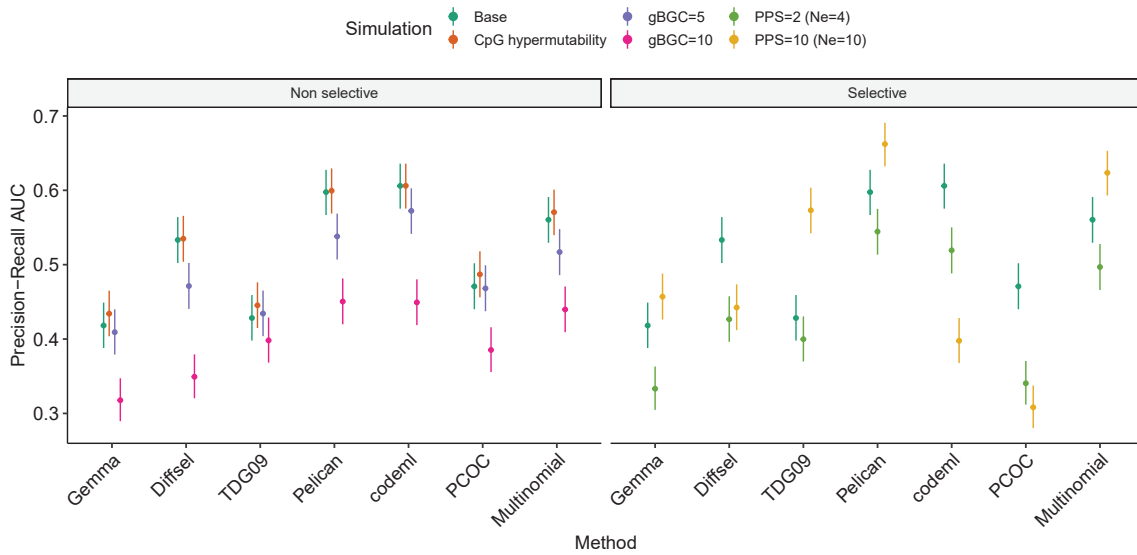


Figure C.4: Precision-recall AUC for all methods in the presence of confounding factors measured on the Rodents dataset.

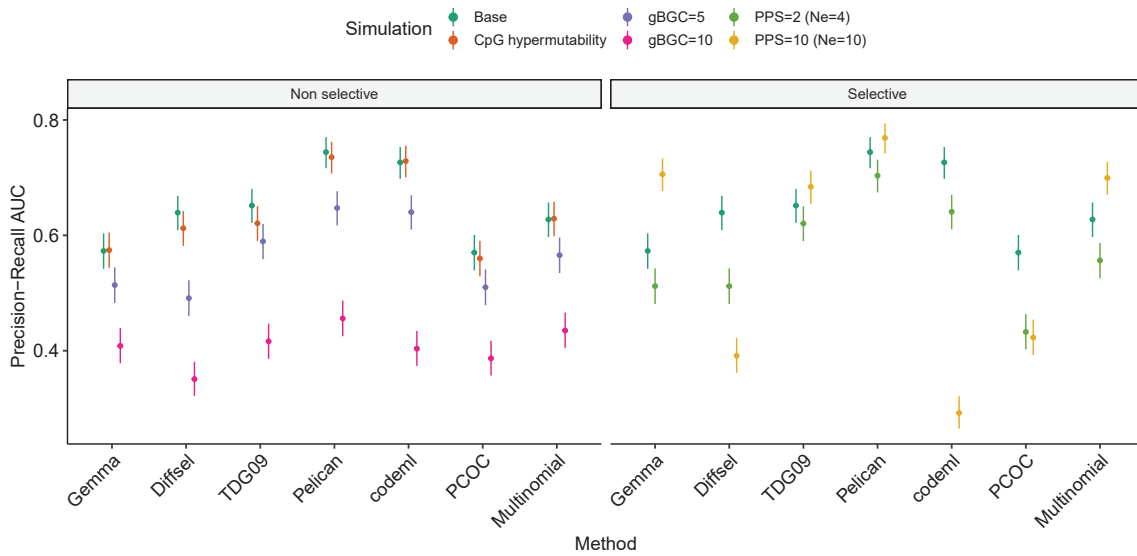


Figure C.5: Precision-recall AUC for all methods in the presence of confounding factors measured on the *Cyperaceae* dataset.

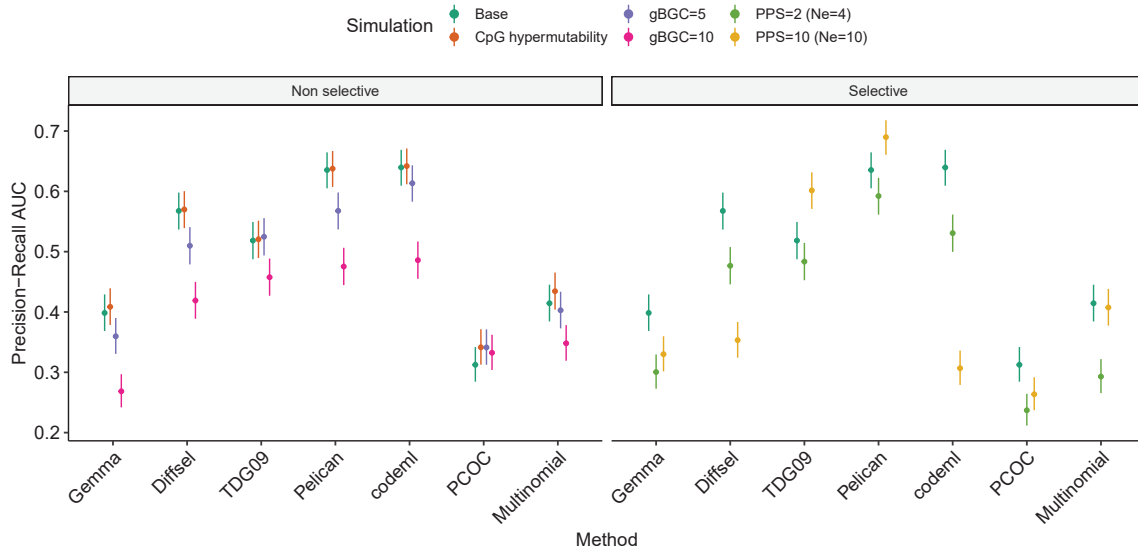


Figure C.6: Precision-recall AUC for all methods in the presence of confounding factors measured on the *Amaranthaceae* dataset.

Persistent positive selection is harder to distinguish from episodic positive selection

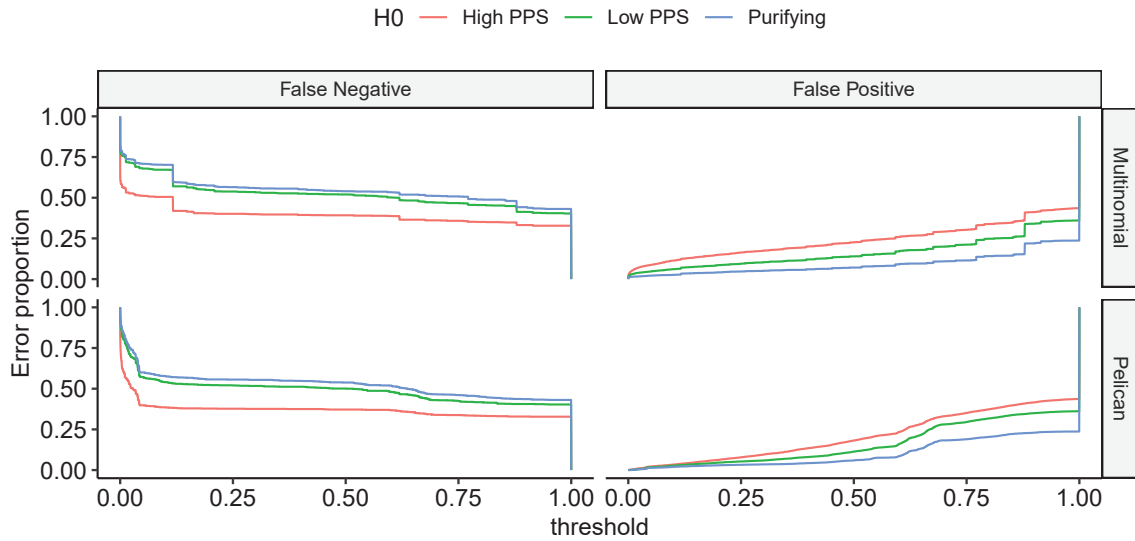


Figure C.7: Evaluation of the confusions made by two methods when detecting episodic positive selection on a background of persistent positive selection (PPS) or purifying selection. PPS can be more easily mistaken for episodic positive selection, resulting in an increased quantity of false positives, and a reduction in the number of false negatives compared to those measured when detecting against a background of purifying selection. Results are extracted from the experiments conducted on the Orthomam phylogeny annotated with echolocation phenotypes, where 9000 H_0 and 1000 H_A sites are simulated.

C.4 Evaluation of Pelican using different degrees of freedom in the LRT

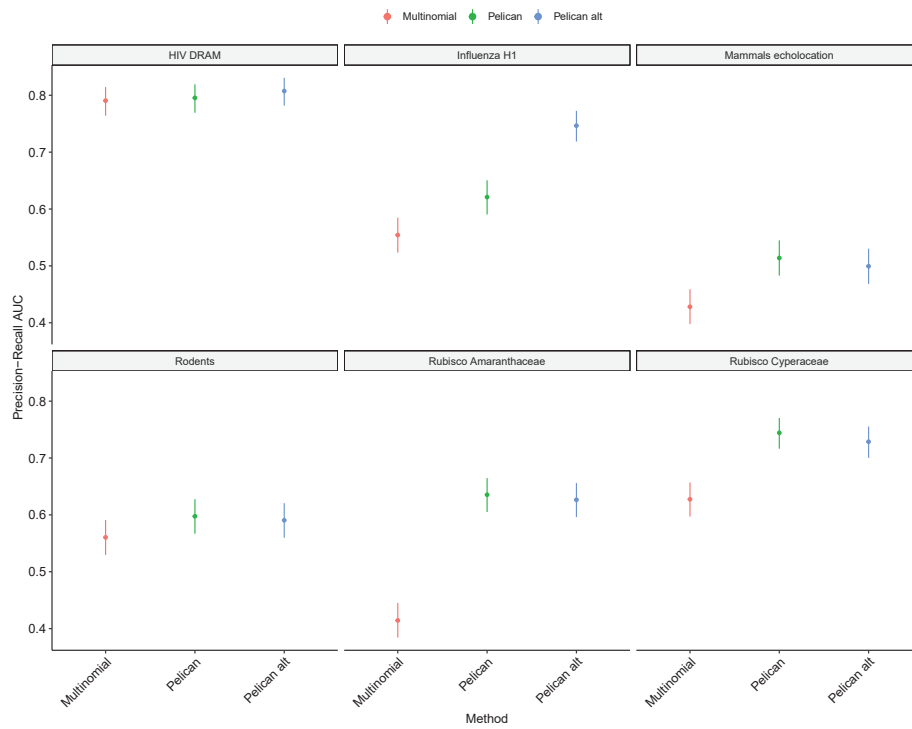
In TDG09 [Tamuri et al., 2009], the degrees of freedom for the LRT were computed as the number of amino acid types observed at one site minus one, corresponding to the number of additional adjustable parameters in the alternative (H_A) model compared to H_0 . Our implementation of this model, named Pelican, uses the same specification for the computation of degrees of freedom.

In response to the observation that Pelican shows strongly decreased performance for detecting relaxations of selective pressure (main fig. 3.5), we investigated a different specification for the computation of the degrees of freedom as the difference between the sum of the numbers of amino acid types observed in each condition, and the total number of amino acid types observed at one site:

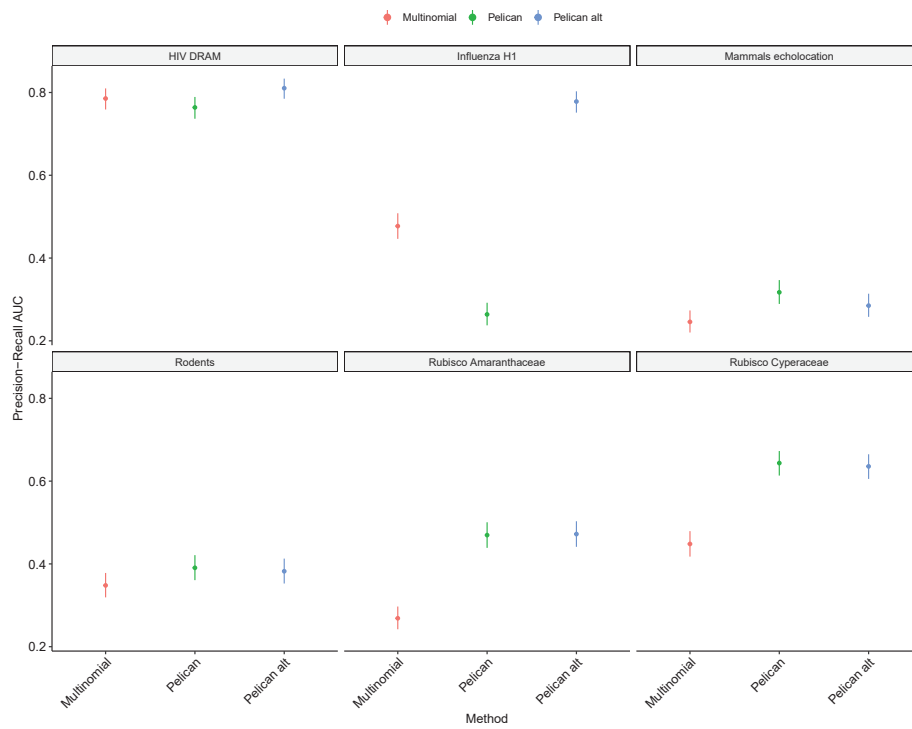
$$\text{df} = \left(\sum_c |AA_c(\text{site})| \right) - |AA(\text{site})| \quad \text{where } c \text{ is a condition}$$

We name this alternative specification `Pelican alt`. Both specifications were evaluated on all empirical phylogenies (sup. fig. C.8).

The alternative specification appears to be a good trade-off, as performance is slightly decreased on some datasets (e.g Echolocation), but strongly increased on others (e.g Influenza, HIV). However, it differs from the specification in the original implementation of the model [Tamuri et al., 2009], and from the usual specification of a LRT [Wilks, 1938], and needs to be more thoroughly tested. Altogether, these results suggest that further improvements can be made to Pelican by improving how the LRT is computed.



(a) Differential selection



(b) Relaxed selection

Figure C.8: Evaluation of Pelican using two different specifications for the computation of degrees of freedom in the likelihood ratio test. **Pelican** is the version that was evaluated throughout the main manuscript, and uses the original specification from [Tamuri et al., 2009]. **Pelican alt** uses a different specification that is defined in the manuscript (Material and methods).

Appendix D

Gene aggregation

Supplementary material for [chapter six](#): Gene-level predictions.

Contents

D.1 Null distribution for the Gene-wise Truncated Fisher (GTF) method .	210
D.2 Mixture model: derivation of EM equations	211
D.2.1 Expectation step	211
D.2.2 Maximisation step	213

D.1 Null distribution for the Gene-wise Truncated Fisher (GTF) method

The Genewise Truncated Fisher (GTF) method is an aggregation method of site p -values as gene p -values and is described in section 6.2.2.

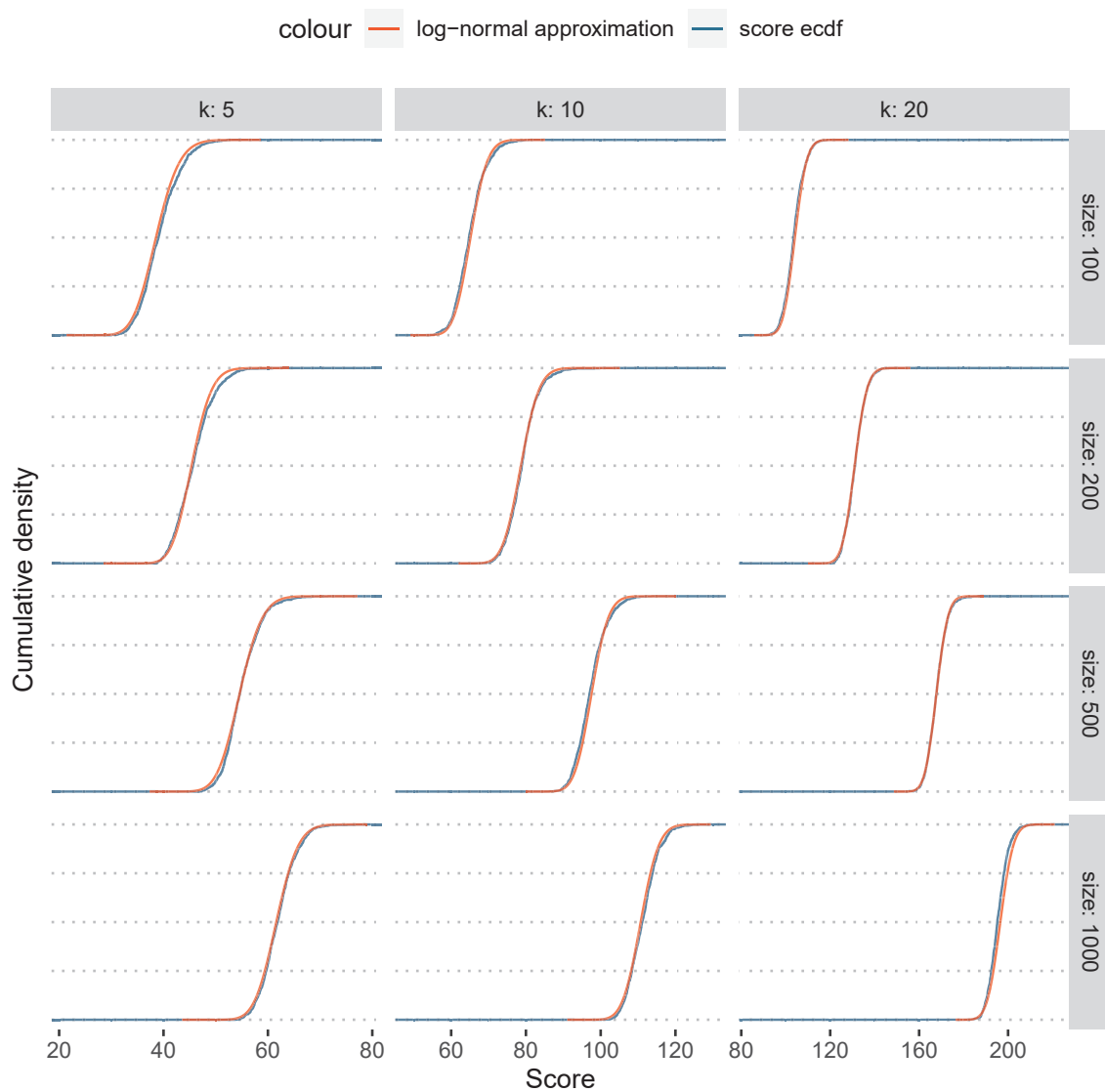


Figure D.1: The empirical cumulative distribution function (ECDF) of the truncated Fisher score used in the GTF method is well approximated by a log-normal distribution, with mean and variance parameters estimated from simulated samples. k is the maximum rank that is considered in the computation of the score, and `size` is the simulation sample size.

D.2 Mixture model: derivation of EM equations

In this model all sites are independent, identically distributed. It can be applied gene by gene, or genome-wide, irrespective of the gene they are in.

$$C_s \sim \text{Bern}(q)$$

$$x_s \sim \begin{cases} \text{U}(0, 1) & \text{if } C_s = 0 \\ \text{Beta}(1, \theta) & \text{if } C_s = 1 \end{cases}$$

where C_s indicates whether site s is undergoing a shift in directional selection or not, and x_s is the p-value distribution at site s .

D.2.1 Expectation step

Jensen's inequality provides a lower bound on the likelihood function:

$$\begin{aligned} \log L(q, \theta) &\geq \mathbb{E}_{C \sim \pi}[\log P(X, C = c|q, \theta)] = f_\pi(q, \theta) \\ f_\pi(q, \theta) &= \mathbb{E}_{C \sim \pi}(\log P(X, C|q, \theta)) \\ &= \sum_c \pi(c) \log P(X, C = c|q, \theta) \end{aligned}$$

The likelihood of X conditionally to the latent variable C being equal c is:

$$\begin{aligned} \log P(X, C = c|q, \theta) &= \log(P(X|C = c, q, \theta)P(C = c|q, \theta)) \\ &= \log(P(X_1, \dots, X_N|C = c, q, \theta)P(C_1 = c_1, \dots, C_N = c_N|q, \theta)) \\ &= \sum_s \log(P(X_s|C = c, q, \theta)P(C_s = c_s|q)) \\ &= \sum_s \log(\text{dunif}(X_s, 0, 1)^{(1-c_s)} \times \text{dbeta}(X_s, 1, \theta)^{c_s} q^{c_s} (1-q)^{(1-c_s)}) \\ &= \sum_s (1-c_s)(\log \text{dunif}(X_s, 0, 1) + \log(1-q)) + c_s(\log \text{dbeta}(X_s, 1, \theta) + \log q) \\ &= \sum_s (1-c_s) \log(1-q) + c_s(\log \text{dbeta}(X_s, 1, \theta) + \log q) \end{aligned}$$

The posterior distribution of C at site s is:

$$\begin{aligned}
 \pi(C_s = c_s) &= P(C_s = c_s | X, q, \theta) \\
 &= P(C_s = c_s | X, q, \theta) \\
 &= \frac{P(X | C_s = c_s, q, \theta) P(C_s = c_s | q, \theta)}{P(X | q, 1, \theta)} \\
 &\propto P(X | C_s = c_s, q, \theta) P(C_s = c_s | q, \theta) \\
 &= P(X | C_s = c_s, q, \theta) q^c (1 - q)^{1-c} \\
 &= \left(\prod_t P(X_t | C_s = c_s, q, \theta) \right) q^c (1 - q)^{1-c} \\
 &= \left(P(X_s | C_s = c_s, q, \theta) \prod_{t \neq s} P(X_t | C_s = c_s, q, \theta) \right) q^c (1 - q)^{1-c} \\
 &= \left(P(X_s | C_s = c_s, q, \theta) \underbrace{\prod_{t \neq s} P(X_t | q, \theta)}_{\text{does not depend on } c_s} \right) q^c (1 - q)^{1-c} \\
 &\propto P(X_s | C_s = c_s, q, \theta) q^c (1 - q)^{1-c} \\
 &= (dunif(X_s, 0, 1)^{1-c} dbeta(X_s, 1, \theta)^c) q^c (1 - q)^{1-c} \\
 &= ((1 - q) dunif(X_s, 0, 1))^{1-c} (q dbeta(X_s, 1, \theta))^c \\
 &= (1 - q)^{1-c} (q dbeta(X_s, 1, \theta))^c
 \end{aligned}$$

$$\pi(C_s = 1) = \frac{q dbeta(X_s, 1, \theta)}{q dbeta(X_s, 1, \theta) + (1 - q)}$$

The lower bound on the likelihood function is then:

$$\begin{aligned}
 f_\pi(q, 1, \theta) &= \sum_c \pi(c) \log P(X, C = c|q, \theta) \\
 &= \sum_c \pi(c) \sum_s \log P(X_s, C_s = c_s|q, \theta) \\
 &= \sum_s \sum_c \pi(c) \log P(X_s, C_s = c_s|q, \theta) \\
 &= \sum_s \sum_{r \in \{0,1\}} \sum_{c, c_s=r} \pi(c) \log P(X_s, C_s = r|q, \theta) \\
 &= \sum_s \sum_{r \in \{0,1\}} \log P(X_s, C_s = r|q, 1, \theta) \left(\sum_{c, c_s=r} \pi(c) \right) \\
 &= \sum_s \sum_{r \in \{0,1\}} \log P(X_s, C_s = r|q, 1, \theta) \left(\sum_c \pi(c_s = r) \pi(c_{(s)}|c_s = r) \right) \\
 &= \sum_s \sum_{r \in \{0,1\}} (\log P(X_s, C_s = r|q, 1, \theta)) \pi(c_s = r) \underbrace{\left(\sum_c \pi(c_{(s)}|c_s = r) \right)}_{=1} \\
 &= \sum_s \pi(c_s = 0) (\log(1 - q) + \log d_{unif}(X_s, 0, 1)) + \pi(c_s = 1) (\log q + \log dbeta(X_s, 1, \theta)) \\
 &= \sum_s \pi(c_s = 0) \log(1 - q) + \pi(c_s = 1) (\log q + \log dbeta(X_s, 1, \theta))
 \end{aligned}$$

D.2.2 Maximisation step

Find optima where the derivative is null:

$$\begin{aligned}
 \frac{\partial f_\pi}{\partial q} &= \sum_s -\frac{\pi(c_s = 0)}{1 - q} + \frac{\pi(c_s = 1)}{q} \\
 \frac{\partial f_\pi}{\partial \theta} &= \sum_s \pi(c_s = 1) (\log(1 - X_s) + \psi(1 + \theta) - \psi(\theta)) \\
 &= \sum_s \pi(c_s = 1) \left(\log(1 - X_s) + \frac{1}{\theta} \right)
 \end{aligned}$$

Parameter estimates:

$$\begin{aligned}
 (1 - q) \sum_s \pi(c_s = 1) &= \sum_s \pi(c_s = 0) q \Rightarrow \hat{q} = \frac{\sum_s \pi(c_s = 1)}{\sum_s \pi(c_s = 0) + \sum_s \pi(c_s = 1)} \\
 &\Rightarrow \hat{q} = \frac{\sum_s \pi(c_s = 1)}{N} \\
 - \sum_s \pi(c_s = 1) \log(1 - X_s) &= \frac{1}{\theta} \sum_s \pi(c_s = 1) \Rightarrow \hat{\theta} = - \frac{\sum_s \pi(c_s = 1)}{\sum_s \pi(c_s = 1) \log(1 - X_s)}
 \end{aligned}$$

Appendix E

Scans of Orthomam for genotype-phenotype associations

Supplementary data for the analyses conducted on the Orthomam database in [chapter six: Gene-level predictions](#) section 6.4, scanning for genes associated to several phenotypic traits.

Contents

E.1 Echolocation	215
E.2 Diet	217
E.3 Adaptation to life in aquatic environments	219
E.4 Adaptation to life in marine environments	221
E.5 Adaptation to life in subterranean environments	223
E.6 Diurnality and nocturnality	225
E.7 Vocal learning	227
E.8 Domestication	229

E.1 Echolocation

Supplementary material for section 6.4.1.

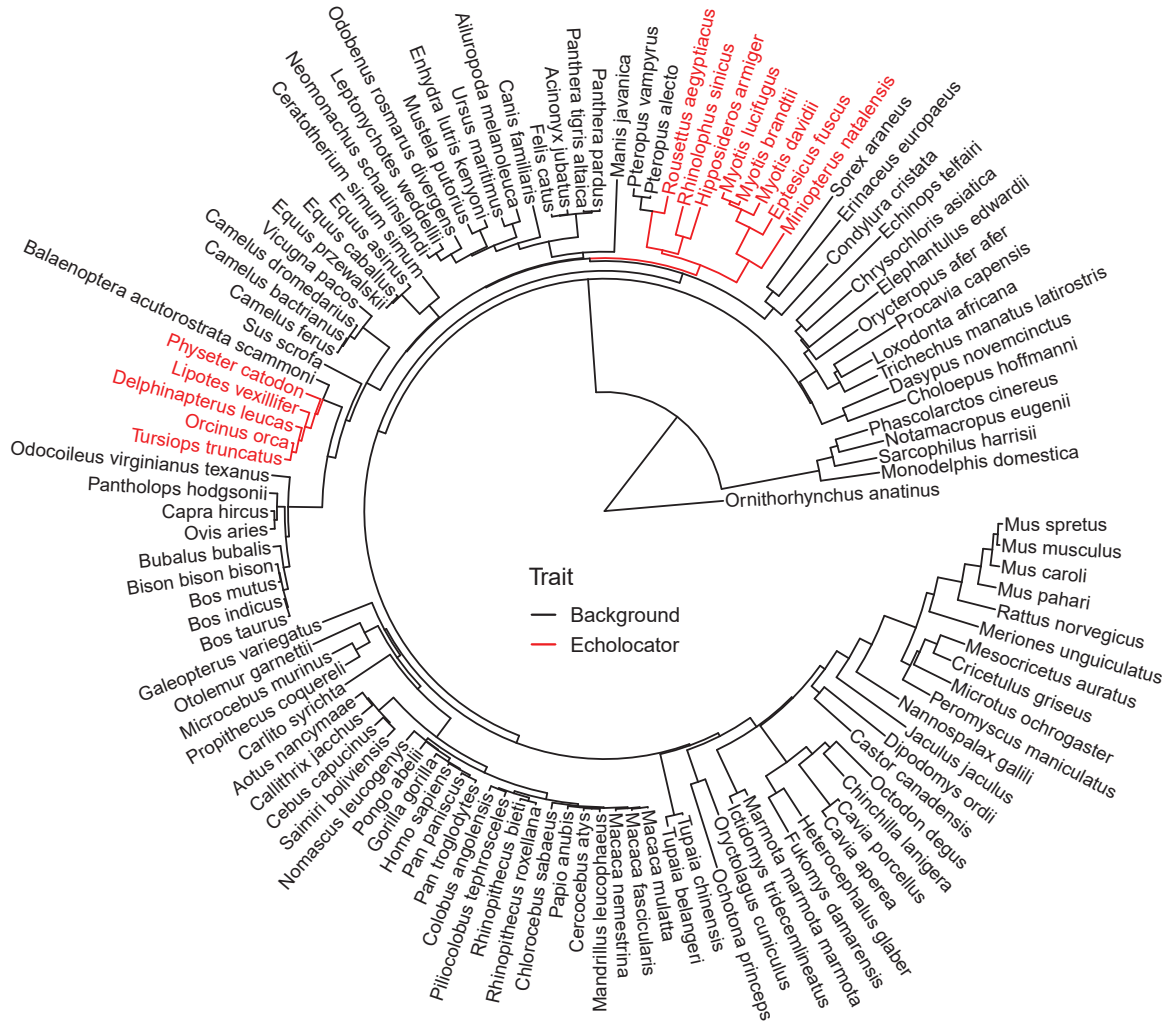


Figure E.1: Orthomam phylogeny with echolocation trait annotation.

Table E.1: Best gene candidates for association to the echolocation trait.

Alignment	p -value	FDR	Functional or adaptive role
PCDH15	3.38×10^{-27}	4.90×10^{-23}	echolocation ^{3,5,7}
ALKAL1	1.84×10^{-17}	1.33×10^{-13}	cytokine ligand
TMC1	6.21×10^{-16}	3.00×10^{-12}	echolocation ^{1,2,3}
LOXHD1	3.33×10^{-15}	1.21×10^{-11}	echolocation ^{1,6}
CDH23	4.81×10^{-14}	1.39×10^{-10}	echolocation ^{3,5}
NKPD1	3.22×10^{-13}	7.78×10^{-10}	NTPase
CA7	1.24×10^{-12}	2.56×10^{-9}	carbonic anhydrase
CHRNA4	1.45×10^{-12}	2.58×10^{-9}	neuronal processing of visual and auditory stimuli ⁴
ZNF536	1.60×10^{-12}	2.58×10^{-9}	neuronal development; vocal learning ⁸
ABTB1	3.47×10^{-12}	5.03×10^{-9}	mediation of protein-protein interactions

¹ [Davies et al., 2012] ² [Dong et al., 2013] ³ [Parker et al., 2013] ⁴ [Espeseth et al., 2007] ⁵ [Shen et al., 2012]

⁶ [McGowen et al., 2020] ⁷ [Davies et al., 2013] ⁸ [Wirthlin et al., 2022]

Table E.2: Genes annotated with a functional role in auditory perception are over-represented among genes associated to the echolocation phenotype.

REAC ID	Term name	p -value	
R-HSA-9659379	Sensory processing of sound	6.04×10^{-8}	***
R-HSA-9662360	Sensory processing of sound, inner hair cells of the cochlea	1.33×10^{-7}	***
R-HSA-9662361	Sensory processing of sound, outer hair cells of the cochlea	1.45×10^{-5}	***
R-HSA-9709957	Sensory Perception	4.18×10^{-5}	***
R-HSA-3000471	Scavenging by Class B Receptors	3.58×10^{-2}	*
R-HSA-9603505	NTRK3 as a dependence receptor	3.89×10^{-2}	*
R-HSA-8964058	HDL remodeling	1.03×10^{-1}	
R-HSA-187015	Activation of TRKA receptors	1.32×10^{-1}	

*** $p < 0.001$ ** $p < 0.01$ * $p < 0.05$

E.2 Diet

Supplementary material for section 6.4.2.

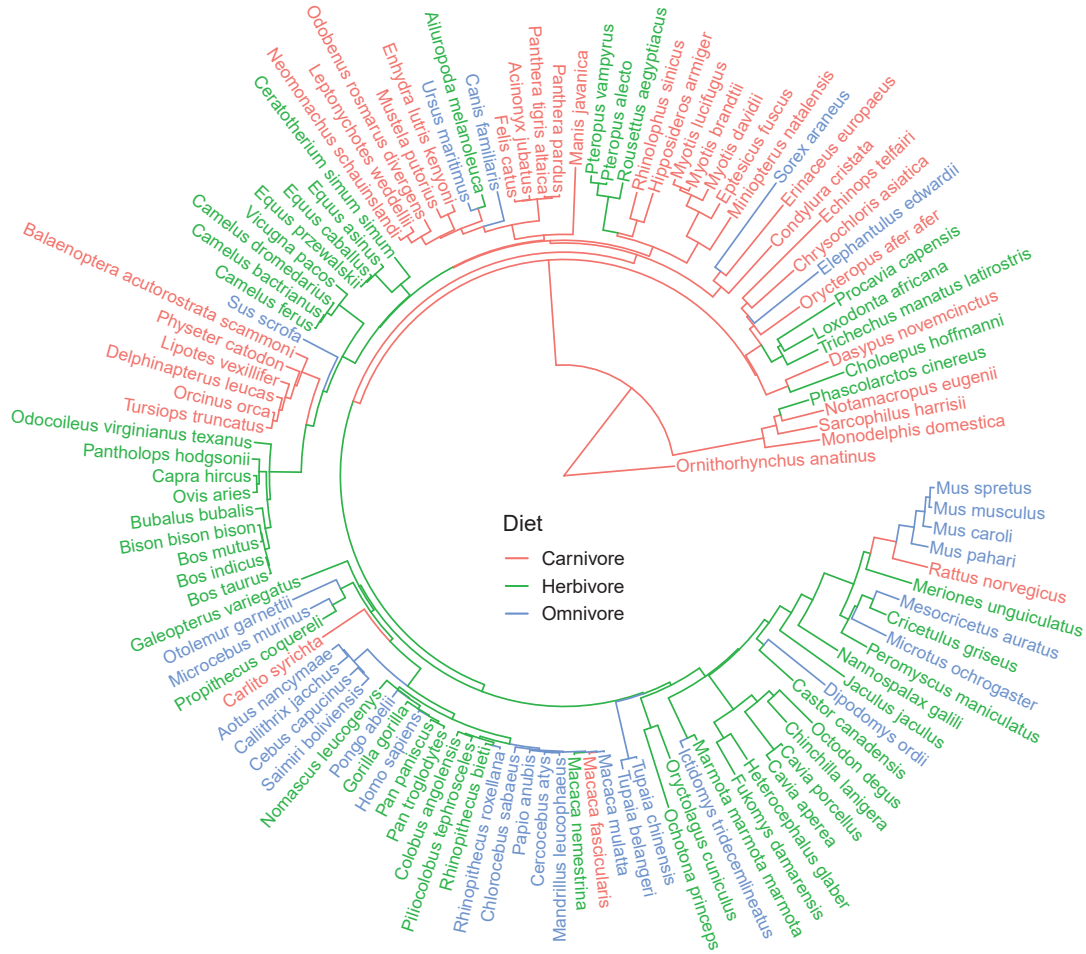


Figure E.2: Orthomam phylogeny with diet trait annotation

Table E.3: Best gene candidates for association to the diet (herbivore, carnivore, omnivore) trait.

Alignment	p -value	FDR	Functional or adaptive role
PNLIP	1.93×10^{-12}	2.80×10^{-8}	pancreatic lipase, carnivory ¹
ENPEP	1.44×10^{-8}	1.05×10^{-4}	protein digestion
CPB1	1.94×10^{-5}	8.17×10^{-2}	protein digestion, carnivory ¹
CPA1	2.25×10^{-5}	8.17×10^{-2}	protein digestion, carnivory ¹
OTOF	9.13×10^{-5}	2.65×10^{-1}	audition
CEL	8.04×10^{-4}	1.00	cholesterol and vitamin assimilation
SLC6A12	1.66×10^{-3}	1.00	GABA transporter
ACSS3	3.41×10^{-3}	1.00	fatty acid metabolism
PLA2G1B	1.31×10^{-2}	1.00	phospholipase, fatty acid digestion ¹
RAPGEF2	1.88×10^{-2}	1.00	cell growth and differentiation

¹ [Wu, 2022]

Table E.4: Predictions associated to diet are enriched in genes having functional roles in digestion.

Term ID	Term name	p -value	
R-HSA-192456	Digestion of dietary lipid	4.5×10^{-4}	***
R-HSA-2022377	Metabolism of Angiotensinogen to Angiotensins	9.2×10^{-4}	***
R-HSA-8935690	Digestion	6.9×10^{-3}	**
R-HSA-8963743	Digestion and absorption	9.7×10^{-3}	**
R-HSA-2980736	Peptide hormone metabolism	2.4×10^{-2}	*
R-HSA-888593	Reuptake of GABA	4.1×10^{-1}	
R-HSA-975634	Retinoid metabolism and transport	6.4×10^{-1}	
R-HSA-6806667	Metabolism of fat-soluble vitamins	7.0×10^{-1}	

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

E.3 Adaptation to life in aquatic environments

Supplementary material for section 6.4.3.

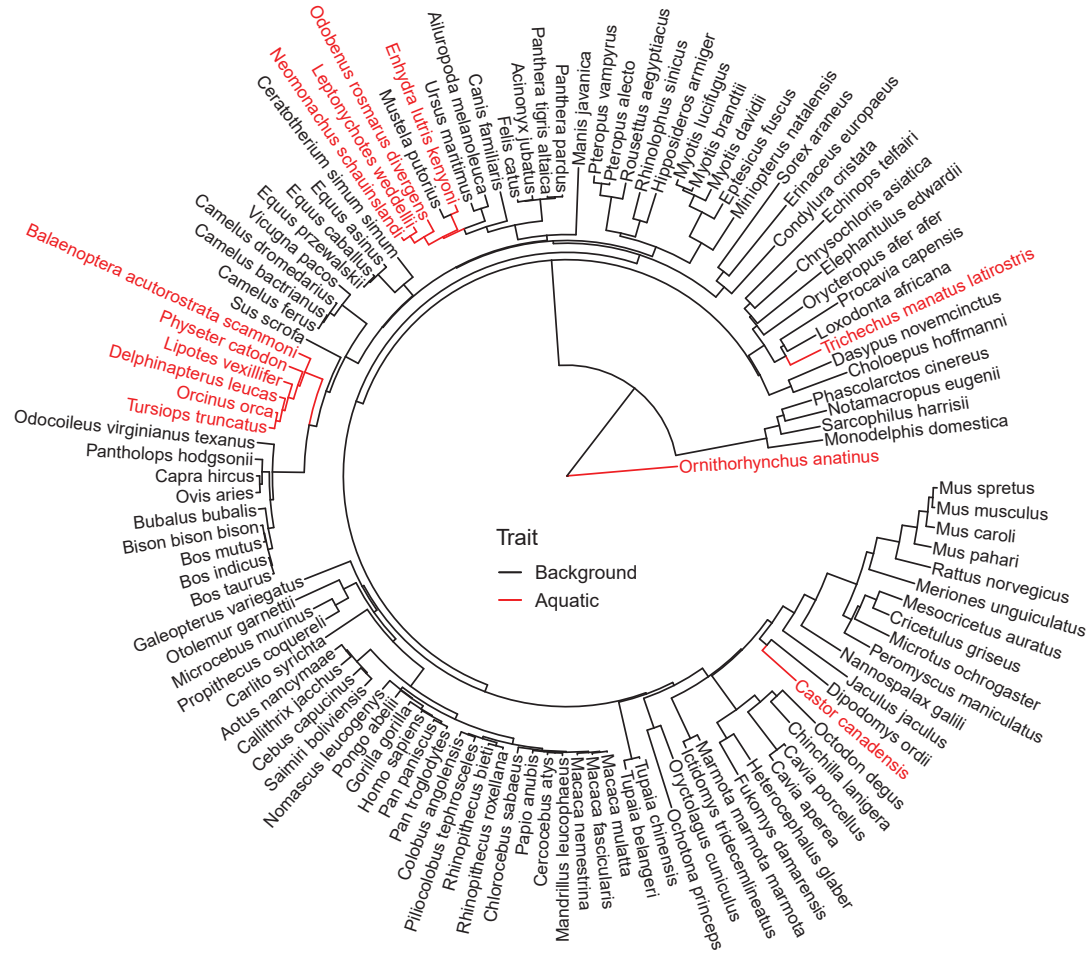


Figure E.3: Orthomam phylogeny with adaptation to aquatic environment trait annotation.

Table E.5: Functional enrichments for gene predictions associated to life in aquatic environments.

Term ID	Term name	<i>p</i> -value
R-HSA-6809371	Formation of the cornified envelope	1.4×10^{-4} ***
R-HSA-1266738	Developmental Biology	2.2×10^{-3} **
R-HSA-6805567	Keratinization	4.4×10^{-3} **
R-HSA-112316	Neuronal System	2.7×10^{-2} *
R-HSA-9619483	Activation of AMPK downstream of NMDARs	4.1×10^{-2} *
R-HSA-9663891	Selective autophagy	4.6×10^{-2} *
R-HSA-1296072	Voltage gated Potassium channels	9.1×10^{-2}
R-HSA-397014	Muscle contraction	9.3×10^{-2}

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Table E.6: Best ranking gene predictions associated to life in aquatic environments.

Alignment	p -value	FDR	Functional or adaptive role
NCOA2	3.87×10^{-11}	5.61×10^{-7}	Transcription coactivator; response to hypoxia [Tsai et al., 2015]
SULT1C3	2.27×10^{-9}	1.65×10^{-5}	Sulfation of polysaccharides; adaptation to aquatic environments [Hettle et al., 2018]
VANGL2	9.23×10^{-9}	4.46×10^{-5}	mammary glands development [Smith et al., 2019]
TUBA3C	1.54×10^{-8}	5.59×10^{-5}	Formation of microtubules
KRT4	1.08×10^{-7}	3.13×10^{-4}	Keratinization
ADPRH	1.88×10^{-7}	4.52×10^{-4}	membrane repair, immunity, tumor suppression [Ishiwata-Endo et al., 2020]
VPS33B	2.18×10^{-7}	4.52×10^{-4}	vesicle mediated protein sorting
S100A5	7.66×10^{-7}	1.39×10^{-3}	calcium binding
MAPK8	8.74×10^{-7}	1.41×10^{-3}	MAP kinase; osmotic regulation [Tian et al., 2019]
KRT80	3.11×10^{-6}	4.51×10^{-3}	Keratinization

E.4 Adaptation to life in marine environments

Supplementary material for section 6.4.3.

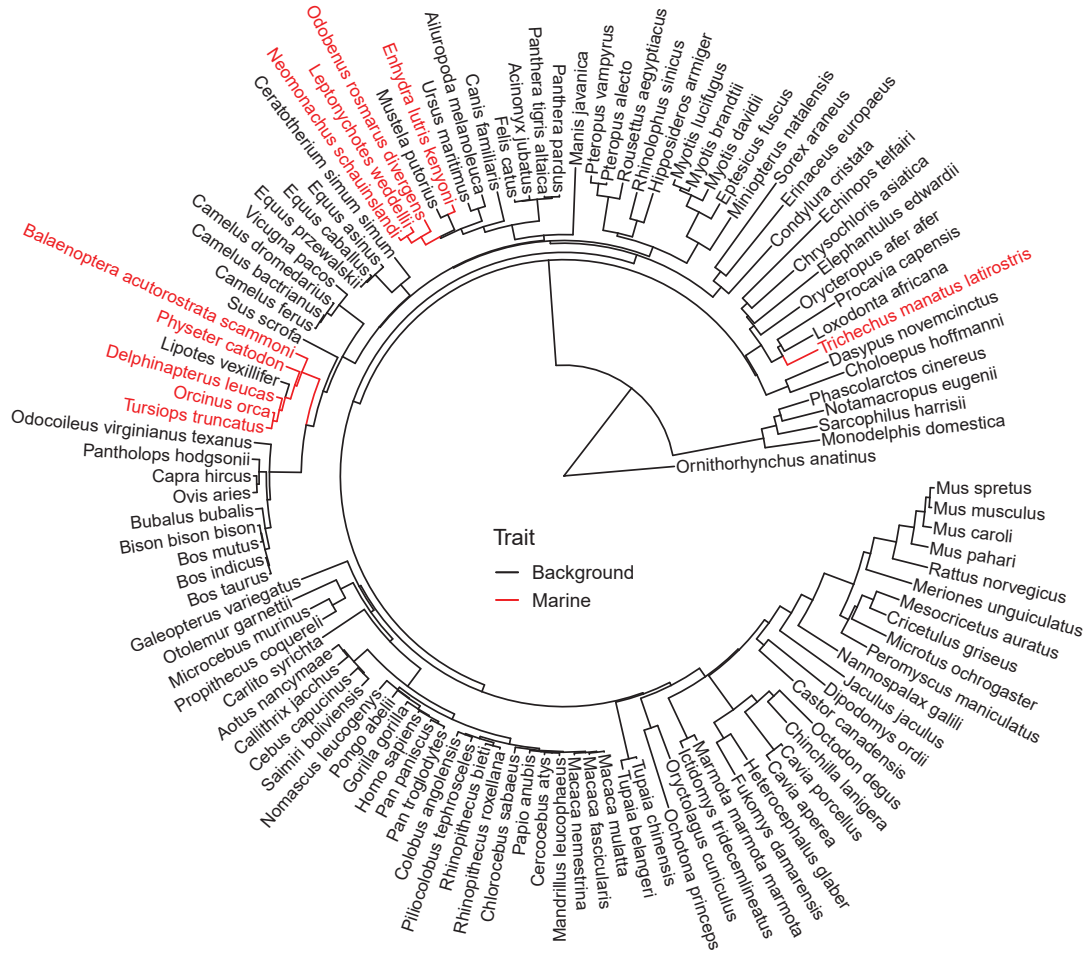


Figure E.4: Orthomam phylogeny with adaptation to marine environment trait annotation.

Table E.7: Functional enrichments for gene predictions associated to life in marine environments.

Term ID	Term name	<i>p</i> -value	
R-HSA-397014	Muscle contraction	4.7×10^{-5}	***
R-HSA-6809371	Formation of the cornified envelope	1.9×10^{-4}	***
R-HSA-390522	Striated Muscle Contraction	4.5×10^{-4}	***
R-HSA-6805567	Keratinization	7.1×10^{-4}	***
R-HSA-9709957	Sensory Perception	1.7×10^{-2}	*
R-HSA-1266738	Developmental Biology	6.0×10^{-2}	
R-HSA-9619483	Activation of AMPK downstream of NMDARs	8.5×10^{-2}	
R-HSA-217271	FMO oxidises nucleophiles	9.3×10^{-2}	

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

Table E.8: Best ranking genes predicted for association with life in marine environments.

Alignment	Rank aquatic	p -value	FDR	Functional or adaptive role
SULT1C3	2	1.15×10^{-18}	1.67×10^{-14}	Sulfation of polysaccharides; aquatic adaptation [Hettle et al., 2018]
TGM1	16	1.08×10^{-11}	7.81×10^{-8}	Structural role in the formation of epidermis
KRT80	10	3.08×10^{-10}	1.49×10^{-6}	Keratinization
TRPV3	35	4.19×10^{-10}	1.52×10^{-6}	Sensory perception of temperature; hair formation [Imura et al., 2007]
PERP	30	6.47×10^{-10}	1.87×10^{-6}	Epithelial development [Ihrie et al., 2005]
TUBA3C	4	1.59×10^{-9}	3.84×10^{-6}	Formation of microtubules
MYL1	37	2.35×10^{-9}	4.86×10^{-6}	Muscle contraction [Zhou et al., 2015]
KRT4	5	9.70×10^{-9}	1.76×10^{-5}	Keratinization
VSIG8	25	2.01×10^{-8}	3.02×10^{-5}	Immunoglobulin
S100A5	7	2.19×10^{-8}	3.02×10^{-5}	Calcium-binding protein

E.5 Adaptation to life in subterranean environments

Supplementary material for section 6.4.4.

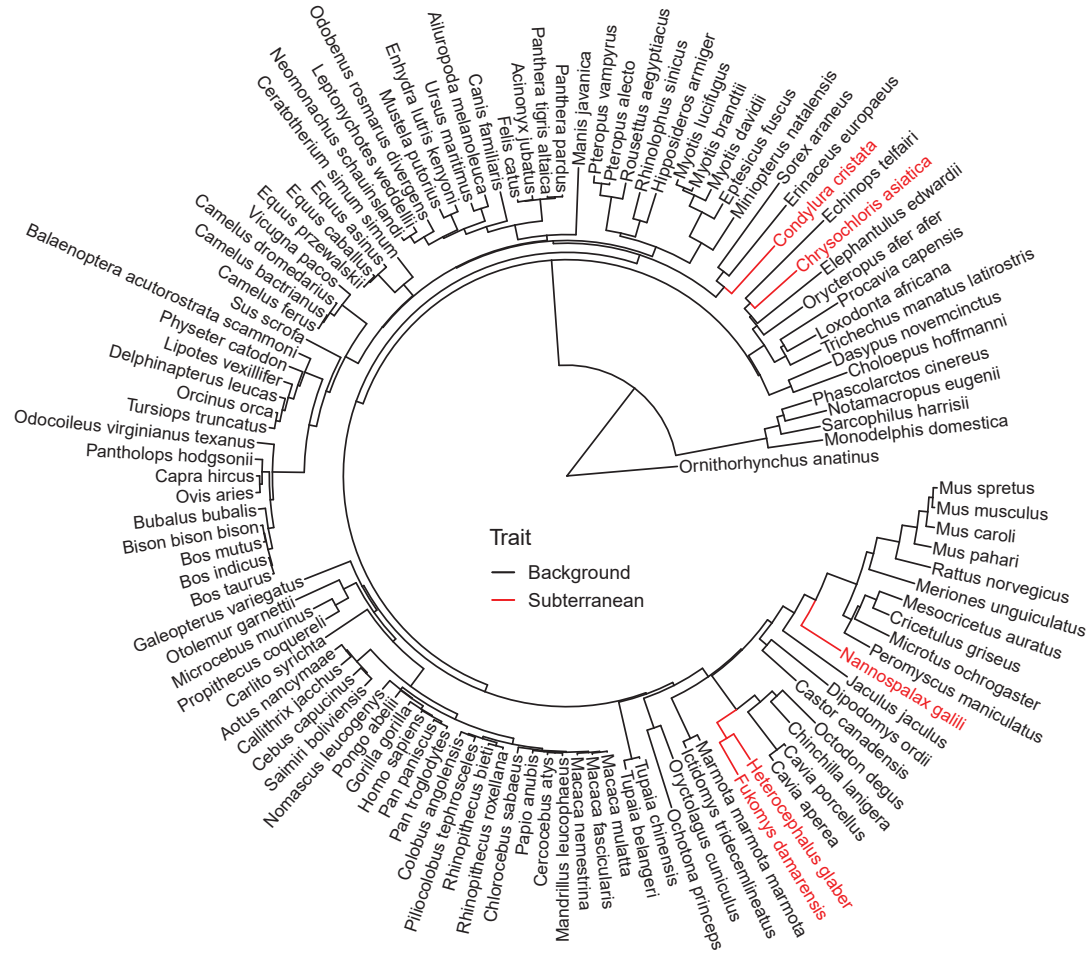


Figure E.5: Orthomam phylogeny with subterranean trait annotation.

Table E.9: Best ranking predictions associated to the subterranean living phenotype.

Alignment	<i>p</i> -value	FDR	Functional or adaptive role
MITD1	9.56×10^{-9}	1.39×10^{-4}	Mitosis and cell differentiation
CRYBA1	2.57×10^{-7}	1.86×10^{-3}	Crystallin ^{1*}
ALAD	3.89×10^{-7}	1.88×10^{-3}	Heme synthesis
GNAT1	3.37×10^{-6}	8.58×10^{-3}	Visual transduction ^{1*}
TMIE	3.57×10^{-6}	8.58×10^{-3}	Auditory perception
CRYBB3	3.78×10^{-6}	8.58×10^{-3}	Crystallin ^{1*}
CRYGC	4.15×10^{-6}	8.58×10^{-3}	Crystallin ^{1*}
RPE65	5.86×10^{-6}	9.34×10^{-3}	Retin [*]
GLRA1	6.06×10^{-6}	9.34×10^{-3}	Nervous system, startle reflexes
NUP98	6.45×10^{-6}	9.34×10^{-3}	Nuclear pore

¹ [Partha et al., 2017] * Related to vision

Table E.10: Functional enrichments for gene predictions associated to life in subterranean environments.

REAC ID	Term name	<i>p</i> -value
R-HSA-2514859	Inactivation, recovery and regulation of the phototransduction cascade	1.6×10^{-2} *
R-HSA-2514856	The phototransduction cascade	1.8×10^{-2} *
R-HSA-2187338	Visual phototransduction	2.3×10^{-2} *
R-HSA-2485179	Activation of the phototransduction cascade	4.7×10^{-2} *
R-HSA-9709957	Sensory Perception	1.2×10^{-1}
R-HSA-9009391	Extra-nuclear estrogen signaling	1.3×10^{-1}
R-HSA-8939211	ESR-mediated signaling	8.1×10^{-1}
R-HSA-432040	Vasopressin regulates renal water homeostasis via Aquaporins	1.0

* $p < 0.05$

E.6 Diurnality and nocturnality

Supplementary data for associations to diurnal/nocturnal life in section 6.4.5.

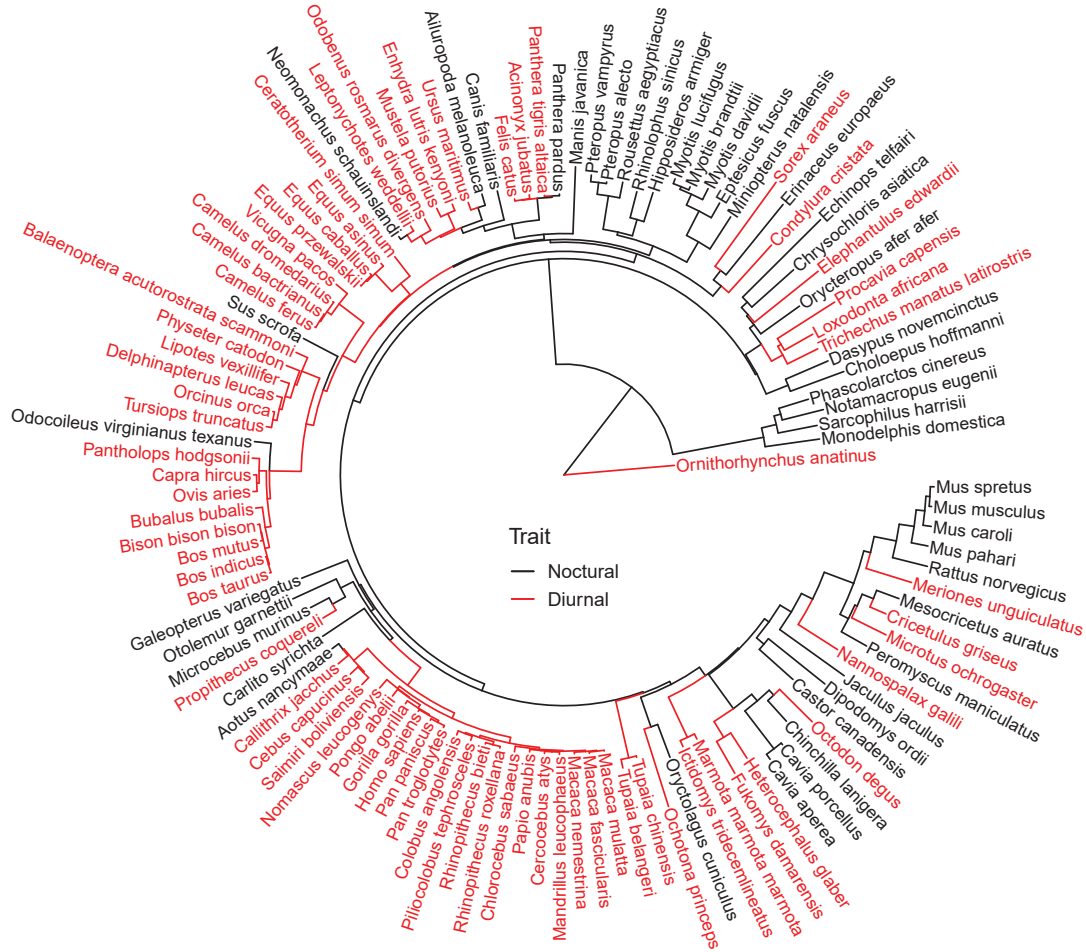


Figure E.6: Orthomam phylogeny with diurnal/nocturnal trait annotation.

Table E.11: Best ranking gene predictions associated to diurnal/nocturnal life.

Alignment	p -value	Functional or adaptive role
PITPNC1	1.83×10^{-10}	thermogenesis from adipose tissue [Tang et al., 2022]
RHO	1.30×10^{-9}	rhodopsin, dim light vision [Sugawara et al., 2010]
GADD45B	1.60×10^{-7}	DNA damage repair
AKR1E2	2.62×10^{-6}	ketone metabolism; involvement in cataract formation
KCNAB1	3.45×10^{-6}	potassium ion channel
ATP4B	7.68×10^{-5}	potassium ion channel
PINLYP	8.66×10^{-5}	phospholipase inhibiting
COMMD9	2.16×10^{-4}	sodium ion transport
GPT	2.61×10^{-4}	glucose and AA metabolism
CELF3	2.66×10^{-4}	involved in language disorders

Table E.12: Functional enrichments for gene predictions associated to diurnal/nocturnal life.

REAC ID	Term name	<i>p</i> -value
R-HSA-8964540	Alanine metabolism	2.0×10^{-1}
R-HSA-419771	Opsins	4.1×10^{-1}
R-HSA-2485179	Activation of the phototransduction cascade	5.0×10^{-1}
R-HSA-2514859	Inactivation, recovery and regulation of the phototransduction cascade	5.3×10^{-1}
R-HSA-2514856	The phototransduction cascade	5.6×10^{-1}
R-HSA-6782210	Gap-filling DNA repair synthesis and ligation in TC-NER	6.1×10^{-1}
R-HSA-6782135	Dual incision in TC-NER	6.3×10^{-1}
R-HSA-8939211	ESR-mediated signaling	8.7×10^{-1}

E.7 Vocal learning

Supplementary data for associations to vocal learning in section 6.4.5.

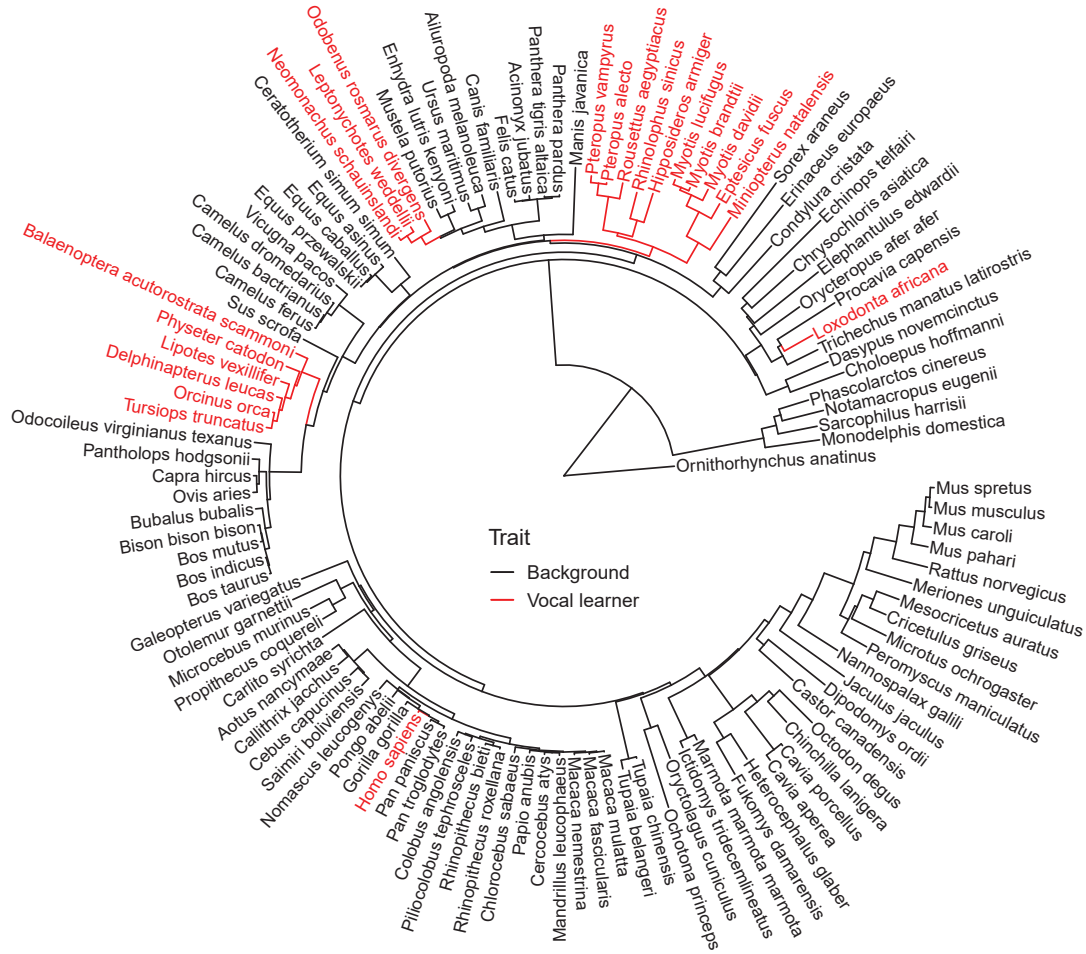


Figure E.7: Orthomam phylogeny with vocal learning trait annotation.

Table E.13: Best ranking genes predictions associated to vocal learning.

Alignment	<i>p</i> -value	FDR	Functional or adaptive role
OR52N1	3.48×10^{-17}	5.05×10^{-13}	olfactory receptor
PCDH15	2.36×10^{-13}	1.71×10^{-9}	retinal and cochlear function
LOXHD1	8.17×10^{-13}	3.95×10^{-9}	auditory perception
NOS1AP	1.49×10^{-12}	5.40×10^{-9}	neuronal nitric oxide synthesis regulation
PPFIA4	1.68×10^{-11}	4.88×10^{-8}	cytoskeleton
KDM5B	1.37×10^{-10}	3.32×10^{-7}	tumor suppression; cognitive disorders
FCRLB	1.69×10^{-10}	3.49×10^{-7}	immune response
CA3	8.38×10^{-10}	1.39×10^{-6}	carbonic anhydrase
VSIG8	8.65×10^{-10}	1.39×10^{-6}	immune response
AMIGO1	3.63×10^{-9}	5.26×10^{-6}	cell adhesion

Table E.14: Functional enrichments for gene predictions associated to vocal learning.

REAC ID	Term name	p -value	
R-HSA-9659379	Sensory processing of sound	2.1×10^{-3}	**
R-HSA-9662360	Sensory processing of sound, inner hair cells of the cochlea	7.5×10^{-3}	**
R-HSA-9662361	Sensory processing of sound, outer hair cells of the cochlea	9.1×10^{-3}	**
R-HSA-9709957	Sensory Perception	3.7×10^{-2}	*
R-HSA-112316	Neuronal System	5.1×10^{-2}	
R-HSA-2142753	Arachidonic acid metabolism	9.0×10^{-2}	
R-HSA-8978868	Fatty acid metabolism	1.6×10^{-1}	
R-HSA-112314	Neurotransmitter receptors and postsynaptic signal transmission	1.6×10^{-1}	

* $p < 0.05$ ** $p < 0.01$ *** $p < 0.001$

E.8 Domestication

Supplementary data for associations to domestication in section 6.4.5.

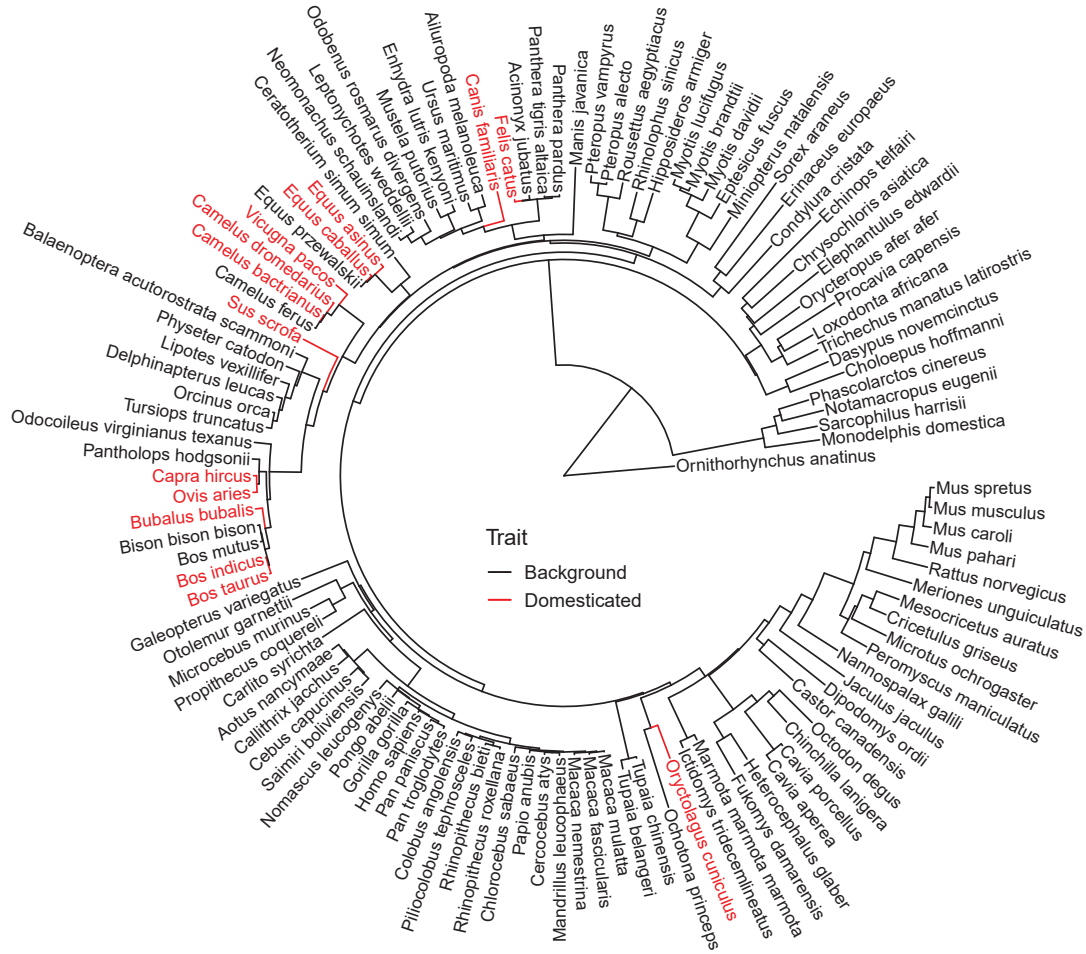


Figure E.8: Orthomam phylogeny with domesticated trait annotation.

Table E.15: List of best ranking gene predictions associated to domestication.

alignment	p -value	FDR	Functional or adaptive role
DNAJB1	2.19×10^{-13}	3.17×10^{-9}	heat shock protein
PPP1CB	2.52×10^{-12}	1.83×10^{-8}	protein phosphatase; glycogen metabolism
ARGLU1	7.47×10^{-12}	3.29×10^{-8}	embryonic development and stress response [Magomedova et al., 2019]
VDAC3	1.13×10^{-11}	3.29×10^{-8}	anion transport
BPIFB4	1.14×10^{-11}	3.29×10^{-8}	longevity
NXPB1	2.15×10^{-11}	5.20×10^{-8}	neuronal signaling
GPM6B	4.59×10^{-11}	9.51×10^{-8}	nervous system development
CUL1	9.94×10^{-11}	1.80×10^{-7}	protein degradation and ubiquitination
GNAL	8.09×10^{-10}	1.30×10^{-6}	olfactory perception
C1QBP	3.16×10^{-9}	4.59×10^{-6}	multifunctional protein

Table E.16: Functional enrichments for gene predictions associated to domestication.

REAC ID	Term name	p -value	
R-HSA-8951664	Neddylation	9.1×10^{-6}	***
R-HSA-9010553	Regulation of expression of SLITs and ROBOs	1.2×10^{-5}	***
R-HSA-195721	Signaling by WNT	2.0×10^{-5}	***
R-HSA-376176	Signaling by ROBO receptors	2.3×10^{-5}	***
R-HSA-983168	Antigen processing: Ubiquitination & Proteasome degradation	5.0×10^{-5}	***
R-HSA-3858494	Beta-catenin independent WNT signaling	9.6×10^{-5}	***
R-HSA-983169	Class I MHC mediated antigen processing & presentation	1.2×10^{-4}	***
R-HSA-1234176	Oxygen-dependent proline hydroxylation of Hypoxia-inducible Factor Alpha	3.4×10^{-4}	***

*** $p < 0.001$

Appendix F

Other work

The first year of my PhD coincided with the onset of the Covid-19 pandemics, during which we joined our efforts to the numerous research initiatives on the subject. The following article is a small study of the spreading dynamics of SARS-CoV-2, early 2020 in France. We adapted a Bayesian model of epidemiology published by [Flaxman et al., 2020] that investigates the effect of non-pharmaceutical interventions across countries, to analyse mortality data in hospitals due to Covid-19, across regions in France. Although the original model distinguished several types of sanitary measures (e.g. schools closing, lockdown...), policies in France were limited to enforcing a full lockdown. We find that the lockdown had a noticeable effect in reducing the mortality. We also looked for a signal for increased mortality due to maintaining municipal elections right before the lockdown was decreed, but did not find evidence in the data for such an effect. However, since every day of delay for enacting the lockdown would result in a strongly increased mortality over the period we considered, the municipal elections could have been indirectly a cause for an increased number of casualties if they were a reason for delaying the application of the lockdown.



Peer Community Journal

Section: Mathematical & Computational Biology

RESEARCH ARTICLE

Published
2022-01-12

Cite as

Louis Duchemin, Philippe
Veber and Bastien Boussau
(2022) *Bayesian investigation of
SARS-CoV-2-related mortality in
France*, Peer Community
Journal, 2: e6.

Correspondence

bastien.boussau@univ-lyon1.fr

Peer-review

Peer reviewed and
recommended by
PCI Mathematical &
Computational Biology,
<https://doi.org/10.24072/pci.mcb.100001>



This article is licensed
under the Creative Commons
Attribution 4.0 License.

Bayesian investigation of SARS-CoV-2-related mortality in France

Louis Duchemin¹, Philippe Veber¹, and Bastien Boussau¹

Volume 2 (2022), article e6

<https://doi.org/10.24072/pcjournal.84>

Abstract

The SARS-CoV-2 epidemic in France has focused a lot of attention as it has had one of the largest death tolls in Europe. It provides an opportunity to examine the effect of the lockdown and of other events on the dynamics of the epidemic. In particular, it has been suggested that municipal elections held just before lockdown was ordered may have helped spread the virus. In this manuscript we use Bayesian models of the number of deaths through time to study the epidemic in 13 regions of France. We found that the models accurately predict the number of deaths 2 to 3 weeks in advance, and recover estimates that are in agreement with recent models that rely on a different structure and different input data. In particular, the lockdown reduced the viral reproduction number by $\approx 80\%$. However, using a mixture model, we found that the lockdown had had different effectiveness depending on the region, and that it had been slightly more effective in decreasing the reproduction number in denser regions. The mixture model predicts that 2.08 (95% CI: 1.85-2.47) million people had been infected by May 11, and that there were 2567 (95% CI: 1781-5182) new infections on May 10. We found no evidence that the reproduction numbers differ between week-ends and week days, and no evidence that the reproduction numbers increased on the election day. Finally, we evaluated counterfactual scenarios showing that ordering the lockdown 1 to 7 days sooner would have resulted in 19% to 76% fewer deaths, but that ordering it 1 to 7 days later would have resulted in 21% to 266% more deaths. Overall, the predictions of the model indicate that holding the elections on March 15 did not have a detectable impact on the total number of deaths, unless it motivated a delay in imposing the lockdown.

¹Université de Lyon, Université Lyon1, CNRS, Laboratoire de Biométrie et Biologie Evolutive, UMR5558, F-69622 Villeurbanne, France



Peer Community Journal is a member of the
Centre Mersenne for Open Scientific Publishing
<http://www.centre-mersenne.org/>

e-ISSN 2804-3871



Contents

1	Introduction	2
2	Material and methods	3
	2.1 Models.....	3
	2.2 Data.....	5
	2.3 Choice of interventions.....	6
	2.4 Simulations to estimate effect sizes	7
	2.5 Implementation	7
	2.6 Availability.....	7
3	Results.....	7
	3.1 Evaluation of Model 1 and of the efficiency of the lockdown.....	7
	3.2 Effect of week-ends.....	9
	3.3 Effect of the elections.....	10
	3.4 Evidence for heterogeneity between regions in the efficacy of the lockdown...	13
	3.5 Status of the epidemic on May 11.....	15
	3.6 Counterfactual investigation of alternative lockdown enforcements.....	16
4	Discussion.....	16
5	Conclusion	18
6	Supplementary material availability	18
7	Acknowledgements.....	18
8	Conflict of interest disclosure	18
	References	18

1. Introduction

The World Health Organization (WHO) declared a pandemic of coronavirus disease 2019 (SARS-CoV-2) on March 11, 2020 following its spread to 114 countries (World Health Organization, 2020) with an estimated 118,000 cases at the time. In France, a first patient was diagnosed with the disease on January 24th 2020 (Bernard Stoecklin et al., 2020). By May 1st, the number of SARS-CoV-2 related deaths in France was 24,594 (French Government, 2020). On March 17 at noon, a lockdown was enforced that required a self-authorization to leave home. This lockdown followed a series of less severe measures such as the prohibition of gatherings above 100 people (March 13) and school closures (March 14).

These measures surrounded already planned nation-wide municipal elections on Sunday March 15. With enforced distancing measures in polling stations, they were maintained, which led to criticism (Cédric Pietralunga, Alexandre Lemarié, Olivier Faye, 2020), as this could have favored the spread of the virus by increasing the number of contacts on a week-end day. It is therefore of interest to investigate whether these elections did have an effect on SARS-CoV-2 related deaths in France.

There has also been suggestions that different parts of France may have adhered to the lockdown requirements with different observance. Behaviours susceptible to favour the spread of the virus may have been more widespread in some regions than in others. In particular, newspapers reported that large numbers of people were not following the strict lockdown rules and instead spent time outside, typically on the banks of the Seine river, in Paris (Elsa Ponchon, 2020). If such differences between regions were true, one might expect to see an effect on region-wise numbers of SARS-CoV-2 related deaths. In particular, the Île-de-France (Paris) region would be expected to show higher mortality rates.

The lockdown was eventually lifted on May 11, when the authorities estimated that the epidemic was sufficiently under control. Given the importance of such a decision, it is important to assess the state of the epidemic on May 11 using several methodological approaches.

Various approaches have been used to monitor the epidemic. Most are compartmental models, which include Susceptible Infected Removed/Recovered (SIR) or Susceptible Exposed Infected Removed/Recovered (SEIR) models. Such models can be used in a deterministic framework, as in (Magal and Webb, 2020; Massonnaud et al., 2020; Roux et al., 2020; Sofonea et al., 2020), can be used for performing simulations by including stochasticity through resampling steps in an otherwise deterministic framework (Neher et al., 2020), or can be used in a completely stochastic framework, as in (Flaxman et al., 2020; Salje et al., 2020). Deterministic models have small computational requirements, but probabilistic approaches lend themselves to statistical inference, e.g. Bayesian inference.

In this paper we used Bayesian inference to study SARS-CoV-2 related deaths in France. We build upon work by Flaxman et al. (Flaxman et al., 2020) to investigate heterogeneity of the viral reproduction number R_t due to both temporal (lockdown, week-ends, election day) and spatial variations (inter-regional heterogeneity), and to evaluate the status of the epidemic when the lockdown was lifted on May 11.

Flaxman et al. proposed a Bayesian method to estimate decreases of the reproduction number (R_t) of the virus due to various interventions such as school closures and lockdowns among 11 countries. We adapted this model from its released version 2. Version 2 improves upon version 1 by accounting for the fact that R_t decreases as the pandemic progresses because a larger portion of the population has been infected and can no longer be infected. We applied the model to the 13 French regions and notably computed region-wise Infection Fatality Rates (IFR) by taking into account region-specific demographic data. First, we investigated the ability of the model to predict the progression of the epidemic in France. Second, we examined the effect of the lockdown on the reproduction number of the disease. Third, we examined the ability of the model to detect two types of temporal heterogeneities: week-ends, during which a smaller portion of workers go to work, and March 15th election day. We used simulations to assess the effect size necessary for the model to detect these heterogeneities, and then applied the model to the empirical data. Fourth, we developed a mixture model to study potential heterogeneities among regions. We found that this model had a better fit than the first model. Fifth, we used both model 1 and the mixture model to assess the total number of infections as of May 11, and the new infections on that day. Finally, we investigated counterfactual scenarios in which the lockdown is imposed 1 to 7 days before or after the actual date.

2. Material and methods

2.1. Models.

2.1.1. Basic model. Here we present the version 2 of the model by Flaxman et al. (Flaxman et al., 2020) briefly, and direct the interested reader to the original publication for more details. We have kept the original authors' symbols for clarity. Version 2 models the evolution of the number of deaths day by day by assuming a discrete renewal process, where portions of the population are susceptible, infected, or recovered/dead. This process describes the evolution of the number of infections over time, and serves as an input to a model of the time between infection and death. In the original model, heterogeneities between countries were induced by different input parameter values. For instance, each country had its own population size. All the countries however shared the same estimated parameter values, apart from parameters setting the number of seed infections, which describe the numbers of infections happening during the first 6 days of the epidemic in a given country, and are necessary to initiate the epidemic. The model accounted for variations in the reproduction number of the virus due to non-pharmaceutical interventions. It estimated parameter values for each of the interventions, which were shared by all countries.

More specifically, deaths on a given day are the consequence of infections that took place some *infection-to-death* time in the past. The model allows for variation across individuals in this *infection-to-death* time by assigning it a probabilistic distribution π . In practice π is the convolution of two Gamma distributions whose parameters are obtained from the literature. That is, the *infection-to-death* time is modeled as the sum of two independent random times

: the incubation period, and the time between onset of symptoms and death. Both time components are Gamma distributed. The observed daily numbers of deaths $D_{t,m}$ on day t for region m are drawn from a negative binomial distribution with parameters that vary day by day:

$$D_{t,m} \sim \text{NegativeBinomial}(d_{t,m}, d_{t,m} + \frac{d_{t,m}^2}{\psi})$$

where $\psi \sim \text{Normal}^+(0, 5)$ is a half-Normal distribution. $d_{t,m} = \sum_{\tau=0}^{t-1} c_{\tau,m} \pi_{t-\tau,m}$ is the expected number of deaths on day t for region m . It is a discrete sum of the number of new infections $c_{\tau,m}$ per day τ and region m since the first day of data, times the probability $\pi_{t-\tau,m}$ that people infected on that day τ die on day t . The number $c_{\tau,m}$ of new infections on day τ and region m is the result of a discrete renewal process. This process depends first on a distribution g of time between infection and the ability to infect other individuals, and second on a country-specific reproduction number $R_{t,m}$. g is set to be a Gamma distribution with parameters fixed. $R_{t,m}$ models the average number of secondary infections at time t for country m . It depends on:

- the population size of the country: $R_{t,m}$ will tend to be larger in larger populations as there are more people to infect. However, as the number of infected and recovered individuals increases in a country, $R_{t,m}$ decreases because there are fewer individuals to infect. This is handled in the version 2 model deterministically based on population sizes given as input to the model.
- the age structure of the country to account for the variable susceptibility of the different age classes in a population. $R_{t,m}$ will tend to be larger in countries with older populations. This is handled in the version 2 model deterministically based on *infection fatality ratios* (IFR) given as input to the model.
- non-pharmaceutical interventions such as a lockdown. By reducing the number of contacts between individuals, these interventions will tend to reduce $R_{t,m}$. The effect of each intervention is quantified by a single parameter that we seek to estimate from the data. It is assumed to be homogeneous over all days during which it is enforced.

2.1.2. *Model extensions.* Our models reproduce the general structure of the version 2 model. However we applied it to French regions, with changes in the type and number of interventions, and, in one case, allowing for different estimated parameter values for different regions.

We used four models: one model where only the lockdown is included, one model with lockdown and week-ends, one model with lockdown and election day, and one mixture model with lockdown allowing for heterogeneities among regions in the efficiency of the lockdown.

- (1) Model with lockdown. The model with lockdown is basically the same as in (Flaxman et al., 2020) except that a single intervention was considered. Lockdown was considered to have an homogeneous effect throughout all m regions and from its start to its end. It was assumed to have an effect on the reproduction number $R_{t,m}$ of the virus according to equation 1:

$$(1) \quad R_{t,m} = R_{0,m} \times e^{-I_t \times \alpha_{lockdown}}$$

where $R_{0,m}$ stands for the reproduction number at day 0 in region m and incorporates demographic parameters, and I_t stands for an indicator function for day t taking value 1 on lockdown days and 0 otherwise.

The prior distribution of $\alpha_{lockdown}$ is a Gamma distribution of shape 0.1667 and rate 1.0, shifted to the left to allow for decreasing or increasing effects with about a 50/50 chance. For this intervention, large decreasing effects are expected, so the distribution was mirrored around 0 by taking its negative, leading to the prior shown in 2.

$$(2) \quad \alpha_{lockdown} = Z - \frac{\log(1.05)}{6} \text{ where } Z \sim \Gamma(0.1667, 1)$$

- (2) Model with lockdown and week-ends. The second model builds upon the first model by including the influence of week-ends. These were modelled as an additional intervention with the same prior as for the lockdown, assuming less work on week-ends compared to weekdays should induce lower reproduction numbers (Eq. 3). However, let it be clear

that this model is not intended to explain the irregularities in mortality *reporting* during week-ends.

$$(3) \quad R_{t,m} = R_{0,m} \times e^{-I_{t,lockdown} \times \alpha_{lockdown} - I_{t,weekends} \times \alpha_{weekends}}$$

- (3) Model with lockdown and election day. The third model builds upon the first model and includes the influence of the election day. On this single day, another intervention is added, with a prior very similar to that used for the two other interventions, except that we expect here an *increase* of the reproduction number. Therefore, we used the same prior as for the other interventions except for the negative sign, yielding equation 4.

$$(4) \quad \alpha_{elections} = Z - \frac{\log(1.05)}{6} \text{ where } Z \sim \Gamma(0.1667, 1)$$

$$(5) \quad R_t = R_{0,m} \times e^{-I_{t,lockdown} \times \alpha_{lockdown} - I_{t,elections} \times \alpha_{elections}}$$

- (4) Model with heterogeneity among regions.

The fourth model builds upon the first model but allows for heterogeneity among regions with two categories. These two categories of regions are allowed to differ in how much the lockdown changed the transmissibility of SARS-CoV-2. To this end, we implemented a mixture model on $\alpha_{lockdown}$ parameters, with two categories, resulting in two parameters, $\alpha_{lockdown}^1$ and $\alpha_{lockdown}^2$. A region m can choose between the two possible values, and this is indicated with a Bernoulli distributed variable $C_m \in \{1, 2\}$. We called θ the parameter of the Bernoulli distribution, and chose a uniform prior for it. In summary:

$$\theta \sim \text{Beta}(1, 1)$$

$$C_m \sim \text{Bern}(\theta)$$

Then we defined $R_{t,m}$, the reproduction number for region m as:

$$(6) \quad R_{t,m} = R_{0,m} \times e^{-I_{t,lockdown} \times \alpha_{lockdown}^{C_m}}$$

We draw both $\alpha_{lockdown}^k$ values from the same prior distributions as for the first and second models, but enforce that $\alpha_{lockdown}^2$ is larger than $\alpha_{lockdown}^1$ by using a dedicated variable type in Stan (Stan Development Team, 2019). Since Stan does not handle mixture models explicitly, we encoded a marginalized version of our model as proposed in Stan's manual and developed a posterior decoding method (described in Supplementary Material 4.1) to extract results for individual regions.

2.2. Data.

2.2.1. Mortality data.

Mortality data per region were downloaded on May 11 2020 from two sources: OpenCovid (OpenCOVID19 contributors, 2020), and Santé Publique France (SPF) (French Ministry of Health, 2020). OpenCovid is a citizen-based initiative, whose aim is to assemble and provide data sets to study the epidemic in France and abroad. SPF is a governmental agency that provides data related to the epidemic at national and sub-national levels. Both datasets were merged into one, prioritizing data from SPF on the days when observations from both sources were available..

Data for regions Guadeloupe, Guyane, La Réunion, Martinique, and Mayotte, which have low mortality numbers in the studied period, were not included in this analysis. The first day for which we have data in all regions is February 15. The amount of missing data from this day onward is low: 14 days at most for regions Île-de-France, Occitanie and Pays de la Loire, and 10.92 days on average (fig. 1)..

2.2.2. *Infection Fatality Ratios.* Infection Fatality Ratios (IFRs) provide the probability of death given infection, and vary depending on the age of the infected individual. Based on data from China, IFRs were estimated for 9 age classes: [0 – 9], [10 – 19], ..., [70 – 79], [80 <] by (Verity et al., 2020). Those estimates cannot be used directly for French regions as many parameters susceptible to affect IFRs differ between the two countries. However Flaxman et al. (Flaxman

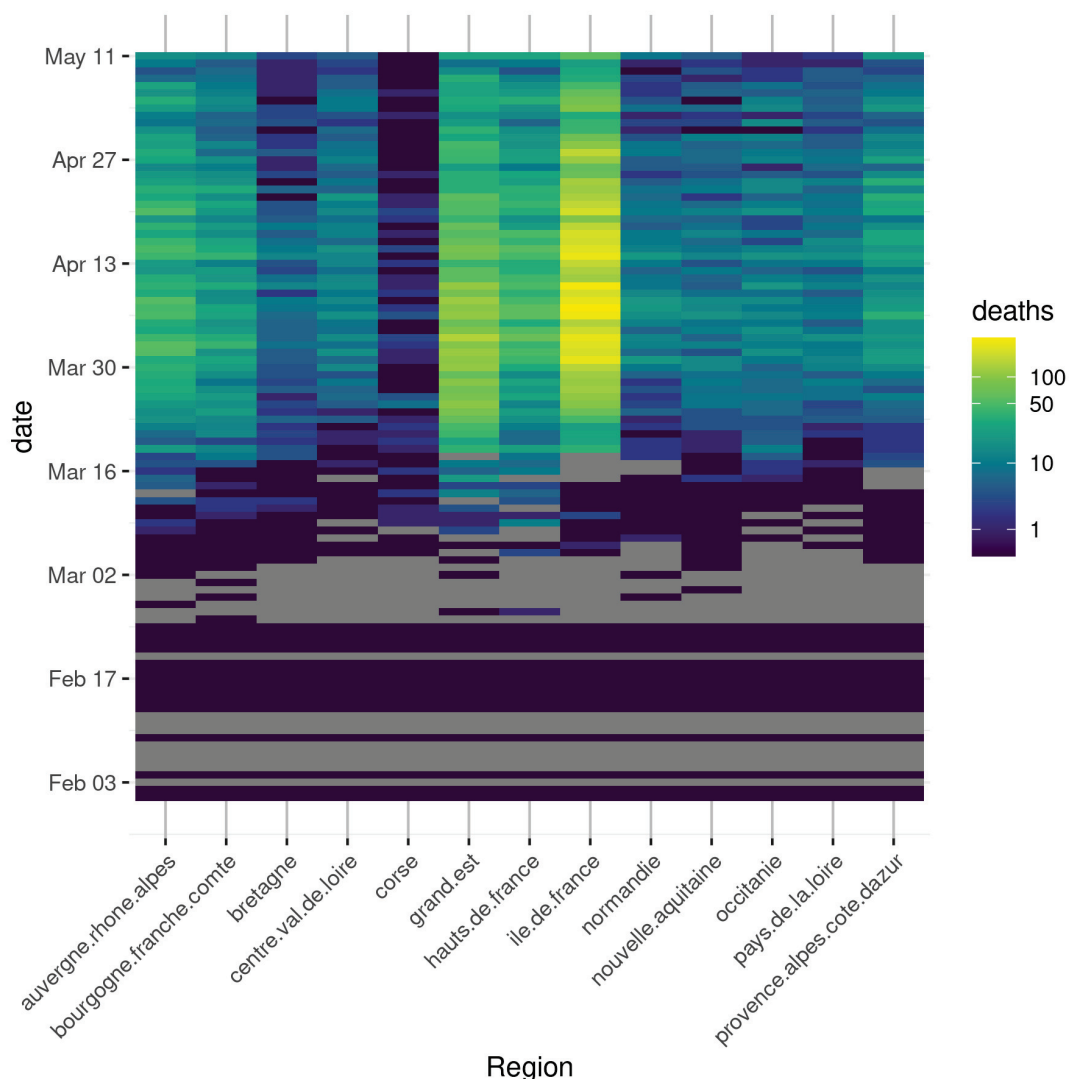


Figure 1 – Mortality data for 13 regions in France, from the first day when all regions have data. Colors are scaled as log mortality for a given day and region. Gray tiles indicate missing data. All data from March 19th onwards originate from the SPF dataset.

et al., 2020) estimated country-specific Case Fatality Rates (CFRs), providing the probability of death given a diagnosed infection. We used the country-wise CFRs for China (0.0138) and France (0.011526) to scale the Chinese age-specific IFRs. More specifically, we use proportionality to scale all Chinese age-specific IFRs by $0.011526/0.0138$ to obtain French age-specific IFRs. Finally, we obtain region-wise IFRs by computing the sum of the French age-specific IFRs weighted by the population size of the corresponding age class in each region.

2.3. Choice of interventions.

In (Flaxman et al., 2020), different interventions had been used: school closure ordered, case-based measures such as self-isolation, public events banned, social distancing encouraged, lockdown decreed. In France, these different interventions happen in close temporal proximity, at the same time in all regions, between March 13 and March 17. This makes identifying their individual contributions very challenging. Therefore we chose to only use one intervention, the full lockdown, on March 17. We also considered two additional events, that were treated in the model as additional interventions: week-ends and the election day, as each could have an effect

on the viral reproduction number. In particular, week-ends may decrease R_t because more businesses are closed on week-ends, and the election day may increase R_t by gathering many voters in polling stations.

2.4. Simulations to estimate effect sizes.

We investigated the ability of the model to detect the effect of one-day events, like the elections, or of week-ends, depending on the size of the effect.

To do so, we relied on simulations reproducing the model's dynamics, and accounting for the effect of the events to be investigated (elections or week-ends) as described in section 2.1.2. Each simulation was initialized with parameters sampled from a previous fit of the model. The reference model used to sample these parameters accounted for the lockdown effect, and was fitted on mortality data up to May 11, yielding 2000 samples of parameter values. 500 sets of parameters were randomly sampled from this pool in order to run 500 simulations per conditions.

Conditions were defined as a fold-change applied to the adjusted R_t during the elections or week-end days. With our prior hypotheses that week-ends would cause a decrease in R_t , we ran simulations assuming fold-changes : 1 (no change), 0.9, 0.75, 0.5. Similarly, to evaluate the consequences of a putative R_t spike during the elections, we ran simulations with fold-changes : 1, 1.25, 1.5, 2. We then compared the simulated mortality between conditions to evaluate the possibility to retrieve such a change in R_t from mortality observations.

2.5. Implementation.

The models described in paragraph 2.1.2 were encoded using Stan's probabilistic language (Stan Development Team, 2019), as variants of the code developed by Flaxman et al. (Flaxman et al., 2020) (version 2). Inference was performed using Stan via the R library rstan. Stan implements a variant of Markov Chain Monte Carlo (MCMC) inference algorithms, called Hamiltonian Monte Carlo (HMC). Given a model with unknown parameters and data, this algorithm generates a sequence of parameter values whose distribution converges to the posterior distribution of the parameters given the data. In our inferences, we used 4 independent chains. We discarded the initial 2000 iterations of each chain (burnin) and used the next 4000 iterations for our posterior sample. Convergence of the chains was assessed by checking the Rhat statistic which is based on comparing inter-chain to intra-chain variance, as recommended in Stan's manual.

2.6. Availability.

The code used for the experiments is available at https://gitlab.in2p3.fr/boussau/corona_french_regions

3. Results

We first investigate whether model 1 can capture the major trends of the epidemic in the French regions. Second, we use it to evaluate the efficacy of the lockdown. Third, we study the ability of models 2 and 3 (section 2.1.2) to identify changes in the reproduction number due to the elections or to week-ends, both on simulated and empirical data. Fourth, we investigate potential differences among regions in the efficacy of the lockdown. Fifth, we study counterfactual scenarios where the lockdown is enforced a few days before or after March 17 to evaluate the effect on the total number of deaths.

3.1. Evaluation of Model 1 and of the efficiency of the lockdown.

3.1.1. *Model fit.* (Flaxman et al., 2020) investigated the fit of their model by cross validation. To do so, they pruned from their data set 3 days for which they have data and compared the inferred numbers of deaths to the empirical numbers of deaths. They repeated this procedure several times. The model was found to behave well, with a correlation of 93% between the inferred and empirical country-wise numbers of deaths. We challenged our model a bit further by predicting the number of deaths in the 13 regions of France after hiding large parts of the data. Each run was performed by removing the k last weeks of data, with k ranging from 0 to 8, and comparing the inferred and empirical numbers of deaths up to May 11 when the lockdown

was lifted. Remaining data points used for estimation after removing those weeks are referred to as "prefix" in this section.

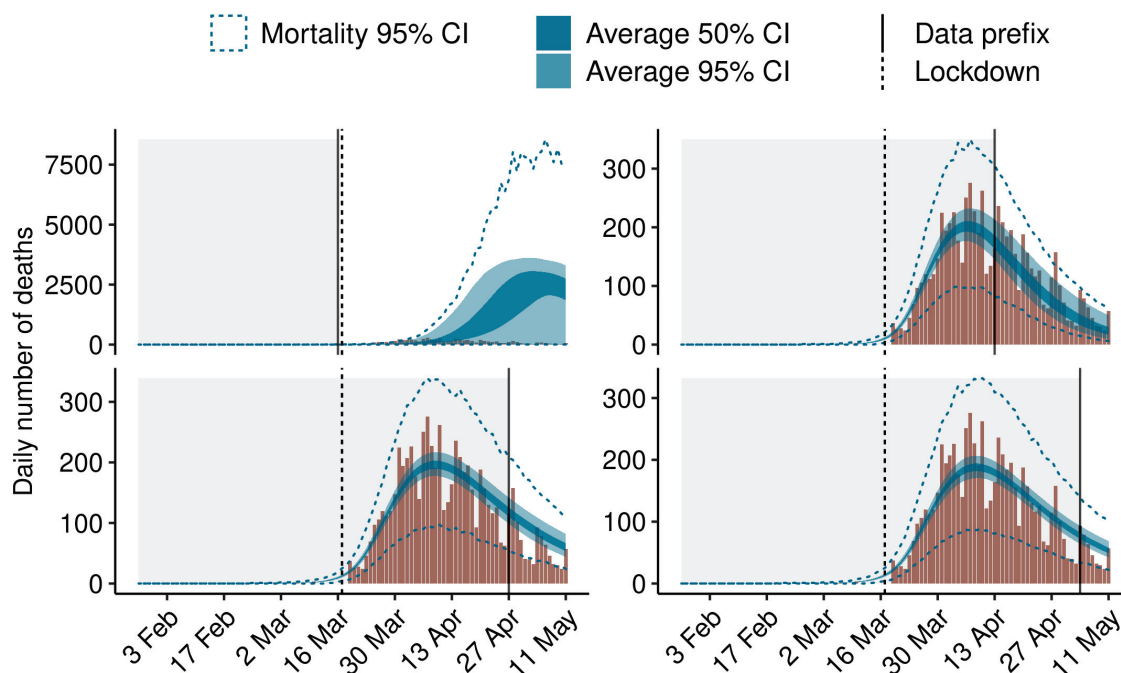


Figure 2 – Model fits using prefixes of data for region Île-de-France. The dashed vertical line corresponds to March 17, when the lockdown was enforced. Data right of the plain vertical line were hidden from the model. The observed numbers of deaths are represented with a brown histogram, and the predictions of the model are in blue. Dark blue ribbons correspond to the 50% credibility intervals and light blue ribbons to the 95% credibility intervals of the expected numbers of death. Blue dashed lines represent the 95% credibility interval of the predicted numbers of deaths $D_{t,m}$ (see section 2.1).

Fig. 2 shows the results when different numbers of days are given as input for region "Île de France". Data for other regions are presented in Supp. Mat. and show the same trends. The data shows weekly trends of low numbers of deaths on week-ends compared to high numbers just after the week-ends. This can be explained by the fact that the counts provided by French public health agencies are based on the date each event was reported, and not the date it occurred (Luc Peillon, 2020). However in practice there is always a latency between the events occurring during the treatment process (e.g hospitalization, admission in ICU, decease) and their reporting. This latency is longer during the week-ends, possibly because of reduced workforce, leading to increased numbers reported on the following Monday. The model does not explicitly handle under-reporting and instead smoothes these irregularities out.

The model both predicts the expected numbers of deaths per day and the actual numbers of deaths, which are simulated thanks to a negative binomial distribution around the expected numbers of deaths. The model performs poorly when the last 8 weeks of data are held out (upper left panel), and vastly overestimates the numbers of deaths. This is likely due to the fact that with such an early censoring of the data, no information about the lockdown is given to the model. The three other panels show that when 4 or more additional weeks of data are provided, the model does a good job at predicting the dynamics of the epidemic. These 4 additional weeks provide the data necessary for the model to estimate the effect of the lockdown on the reproduction number.

For instance, the model estimates that in total there had been 6231 deaths [CI: 5456-7160] in region "Île de France" when all the data up to May 11 is used, 6502 deaths [CI: 5698-7403]

when the data stops one week before May 11 (bottom right panel), 6829 deaths [CI: 5908-7882] when the data stops two weeks before May 11 (bottom left panel), and 5894 deaths [CI: 4854-7443] when the data stops four weeks before May 11 (top right panel). The actual total number of deaths on May 11 in this region is 6643, which is in the credibility interval for all estimates.

To focus on the predictive ability of the model, i.e. its ability to estimate the number of deaths for unobserved weeks, we computed the total squared error only on the last unobserved week of data, and varied the prefix size. With a prefix that stops right before this last week, the total squared error is 12350 (95% CI : 7051-25307). If the prefix stops 2 weeks before the last week, it is 14956 (95% CI : 8036;35293), and 18001 (95% CI : 11420;27495) if the prefix stops four weeks before the last week. The error made by the model when predicting 4 weeks in advance is thus 45% worse than when predicting one week in advance. We conclude from the above that the model can be used to predict the number of deaths several weeks in advance while keeping a useful level of accuracy.

Figure 3 presents fitted mortality for three regions, using data up to May 11. Equivalent figures for all regions in this analysis are provided as supplementary material.

If we focus on the total number of deaths in France using data up to May 11, we observe that the model is able to reproduce the trends in the observed numbers very accurately, making errors ranging from 0.86% (9750 estimated deaths for 9834 observed in data) to 6.70% (7300 estimated for 7824 observed) per day over the month of April (Fig. 4). This shows that the inability of the model to capture weekly irregularities in the reporting of deaths has not had a noticeable effect on the estimation of the total numbers of deaths through time.

Overall, the model appears to capture well the dynamics of the epidemic in French regions. In the following, we use the model to investigate whether particular events in the pandemics in France have left a footprint in the number of deaths.

3.1.2. Reduction of viral transmissibility due to the lockdown. Model 1 allows estimating the effect of the lockdown on the reproduction number of the virus. This is done through a parameter $\alpha_{lockdown}$ whose prior distribution is a shifted Gamma (see section 2.1). The posterior distribution clearly differs from the prior distribution meaning that there is information in the data to estimate the $\alpha_{lockdown}$ parameter value (Supplementary Figure 11).

As shown Fig. 5, the reproduction number in Île-de-France decreases markedly with the lockdown, shifting from about 3.58 (95% CI : 3.34 - 3.86) before the lockdown to 0.69 (95% CI : 0.65 - 0.73) after the lockdown, i.e. a reduction of 80.78%.

At the national level, the average R_t among regions weighted by their population size is 3.34 (95% CI : 3.19 - 3.51) before lockdown and decreasing to 0.65 (95% CI : 0.62, 0.67) after.

3.2. Effect of week-ends.

Model 2 combines the effects of the lockdown and of week-ends. First we investigated what effect size would be necessary to detect an effect of week-ends on viral transmissibility, and then we assessed whether week-ends had had a detectable impact on viral transmissibility.

3.2.1. Effect size required to observe an effect of week-ends. Fig. 6 shows the effect on mortality in Île-de-France through time and total mortality at national scale of decreases in R_t due to a reduction of contacts between individuals on week-ends, when fewer workers are active. They reveal that a R_t fold change of around 0.75 seems necessary for it to have a detectable impact on the number of deaths, because the distributions obtained with an R_t fold change of 0.9 overlap largely with the distributions obtained without a fold change. In terms of contacts, this would mean that there should be 25% fewer contacts during week-ends than during a week-day for the effect to be detectable. Simulation results for all regions are available as supplementary material.

3.2.2. No detectable effect of week-ends on viral spread. The model finds little effect of changes of individual behaviour on week-ends on the dynamics of the number of deaths through time. Fig. 7 shows that the resulting posterior of R_t looks very similar to the posterior obtained without accounting for behavioural changes on week-ends (see Supplementary Figure 14 for comparison

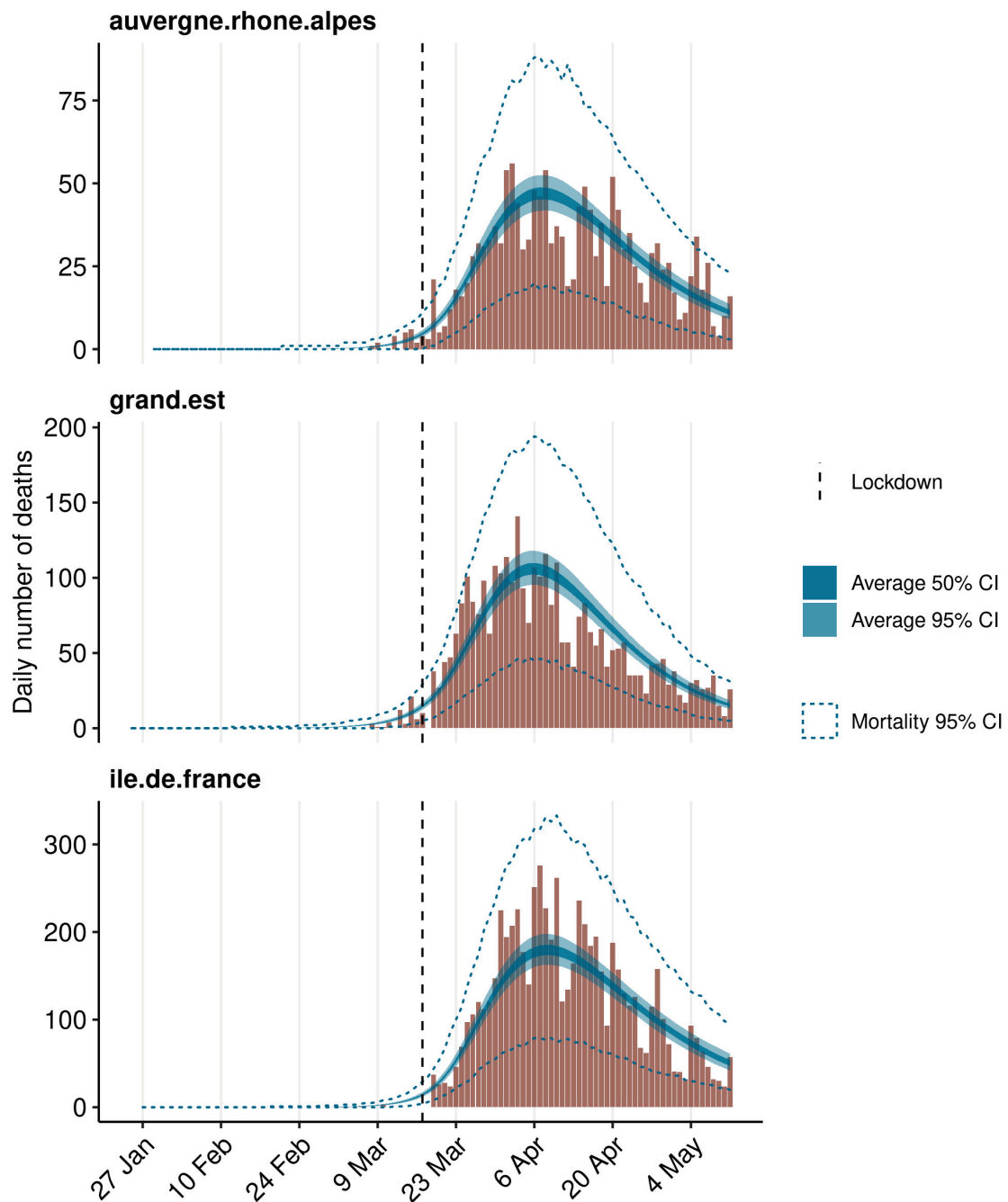


Figure 3 – Model fit on the complete dataset for three different regions.

with base model R_t). The associated posterior distribution of $\alpha_{weekend}$ is presented in Supplementary Figure 12.

3.3. Effect of the elections.

The first round of voting in the municipal elections took place on Sunday March 15, just two days before the nation-wide lockdown was enacted. The voter turnout amounted to 41.6%. Following measures were enforced : safety distancing, and a maximum of three voters were allowed at once in polling stations ; hydroalcoholic gel was available in every polling station, and masks were mandatory ; voters were encouraged to bring their own pen and ballot paper which

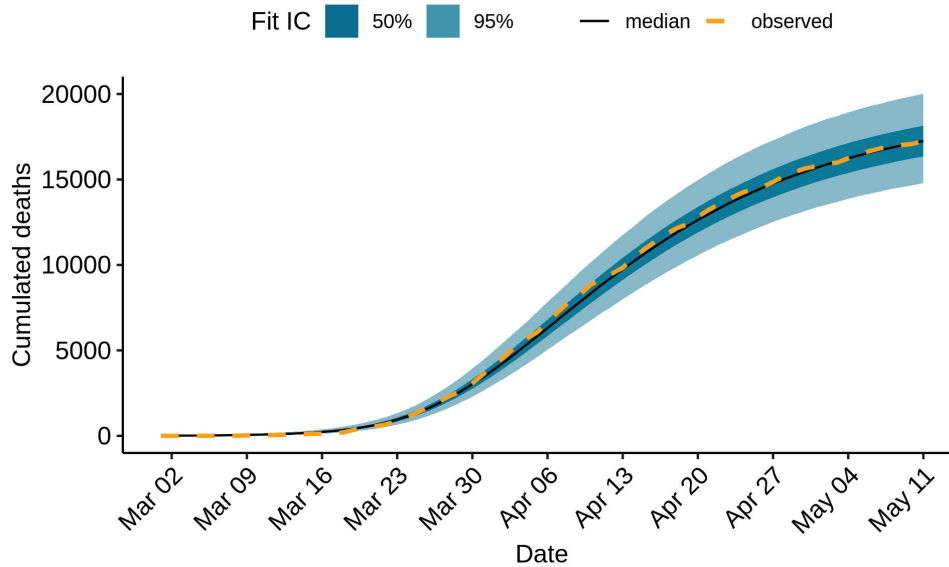


Figure 4 – Cumulated mortality over time, fitting data up to May 11.

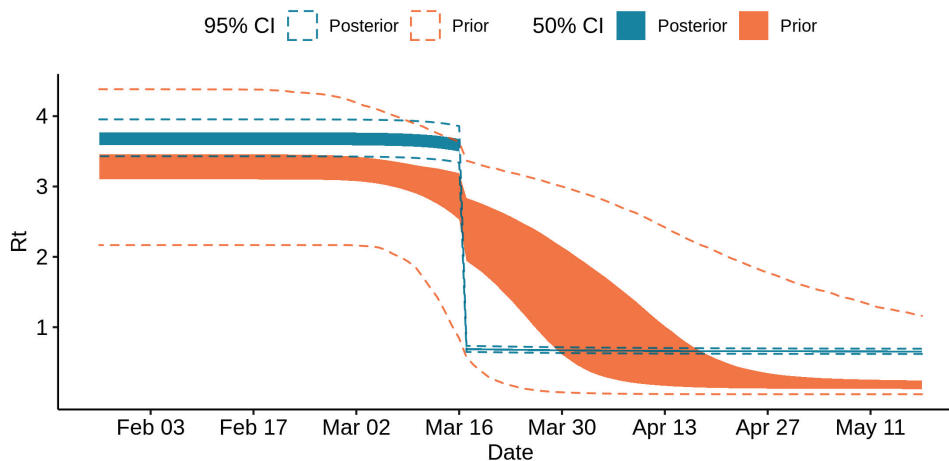


Figure 5 – Prior and posterior samples of R_t in region Île-de-France.

was sometimes sent by mail. Even with these precautions, such an event is expected to increase the number of contacts that occurs during the day, as well as the reproduction rate.

Model 3 combines the effects of the lockdown and of the election day. First we investigated what effect size would be necessary to detect an effect of the election day on viral transmissibility. Using simulations, we investigated different fold change values for the R_t parameter. Second, we assessed whether the election day had had a detectable impact on viral transmissibility using the French mortality data.

3.3.1. *Effect size required to observe an effect of the election day.* Fig. 8 suggest that in order to detect an increase of the transmission rate R_t on the election day based on mortality data, this effect would have to be a change in R_t of at least a factor 2. This suggests that a model based of the number of deaths through time could only detect strong increases of R_t during the election day. Additional simulation results for all regions are presented as supplementary material.

3.3.2. *No detectable effect of the election day on viral spread.* The model finds no evidence for an increase in the number of contacts during election day on the dynamics of the number of deaths

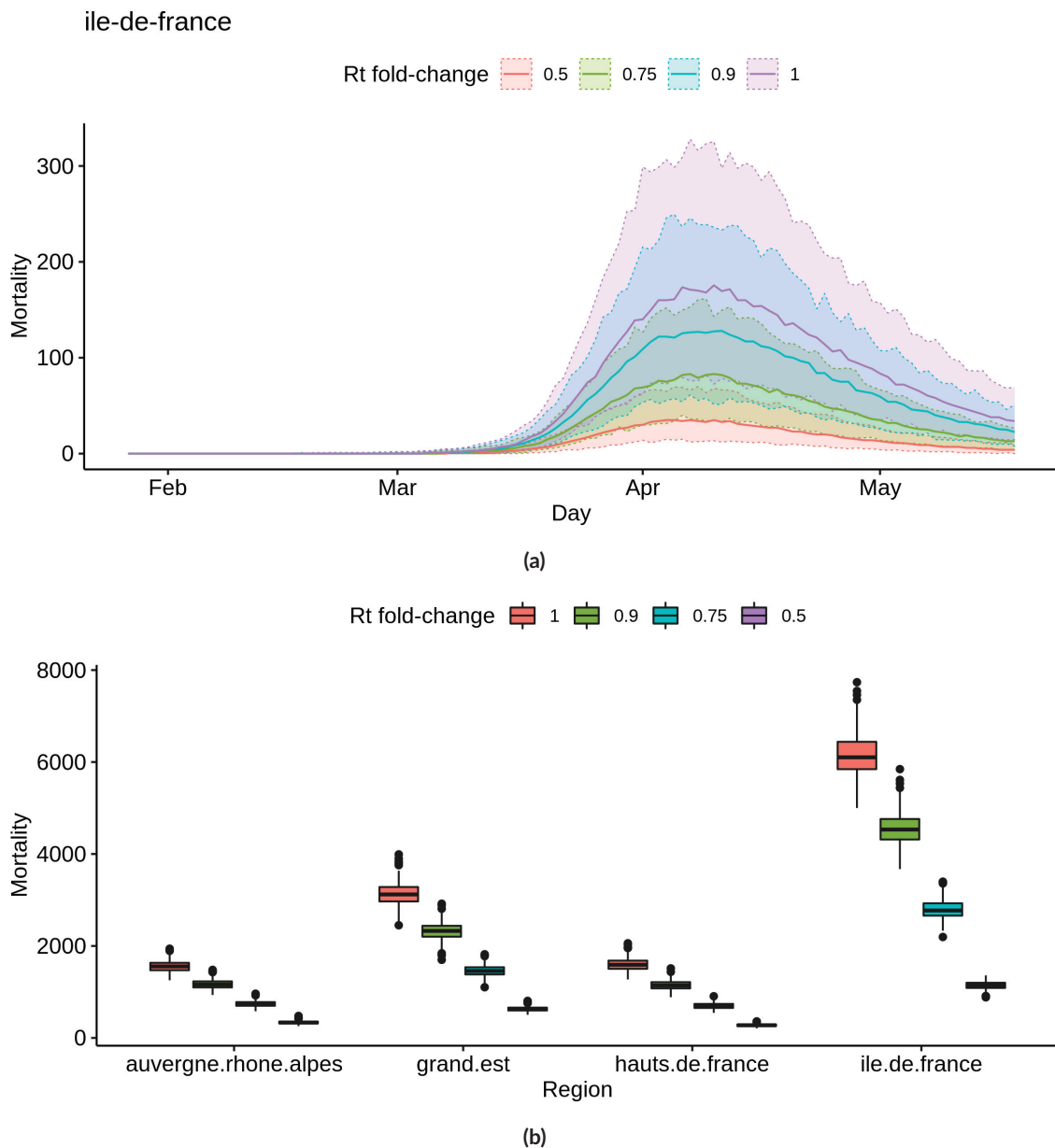


Figure 6 – Simulated distributions of deaths, assuming different effect sizes of week-ends on R_t . (a) Simulated distribution of deaths through time in region Île-de-France. Median values are represented with a solid line, and shaded areas correspond to 95% credibility intervals. (b) Distributions of the total numbers of deaths in four regions. Each box shows (from top to bottom) the 3rd quartile, median and 1st quartile of the distribution. The vertical line on top of each box extends up to the largest value of the sample no further from the 3rd quartile than 1.5 times the inter-quartile difference; larger values are then represented as dots and can be interpreted as possible outliers. The vertical lines below each box are constructed in an analogous way for low values.

through time. Fig. 9 shows that the resulting posterior on the R_t value is much flatter on March 15 than the prior. The associated posterior distribution of $\alpha_{elections}$ is presented in Supplementary Figure 13.

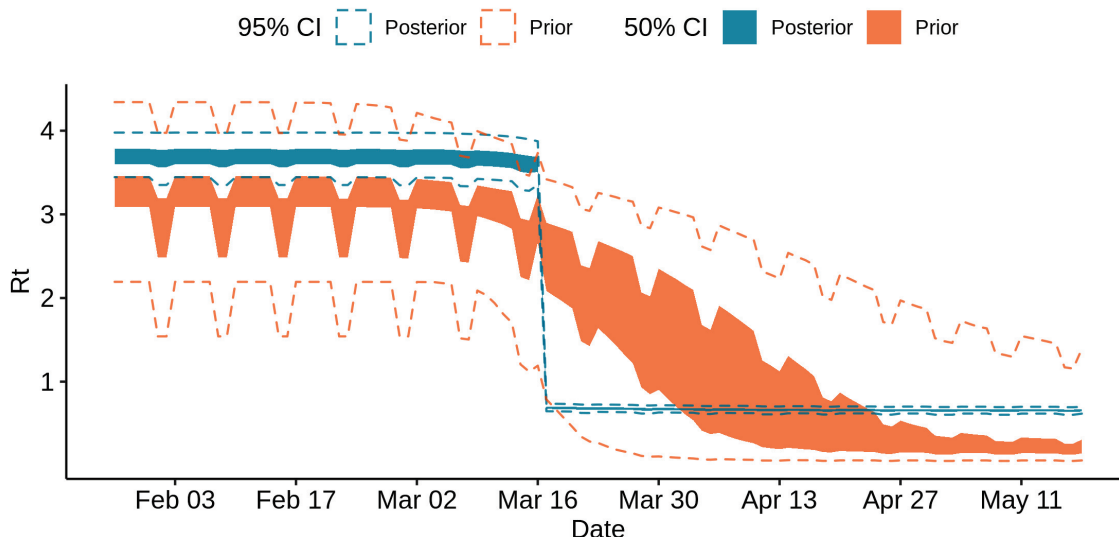


Figure 7 – Prior and posterior samples of R_t in region Île-de-France

3.4. Evidence for heterogeneity between regions in the efficacy of the lockdown.

It has been suggested that the lockdown may have not been applied as strictly in different French regions. To investigate this, we used a mixture model to allow for two categories of reduction of the transmissibility due to the lockdown. We estimated two $\alpha_{lockdown}$ values, one for each category of the mixture, and estimated a proportion θ_i associated to each category. We found that the two categories almost had the same share among the 13 regions, with $\theta_1 = 0.52$ and $\theta_2 = 0.48$; comparison between the prior and the posterior distributions indicates that the data informed the model (Supplementary Figure 7). The corresponding reduction factors were $\alpha_{lockdown}^1 = 1.57$ (95 %C.I. 1.46 - 1.65) and $\alpha_{lockdown}^2 = 1.79$ (95% C.I. 1.67 - 1.94). We used posterior decoding to assign to each region a distribution of the R_t fold change due to the lockdown (Fig. 10), defined as $\exp(-\alpha)$ in equation 1. The distributions appear to be bimodal, which is expected given the underlying two categories of $\alpha_{lockdown}$ used in the mixture model. The sizes of the modes vary depending on the region, which reveals that the two $\alpha_{lockdown}$ values fit the regions differently. The lower R_t fold change fits best the regions Île de France or Corse, while the higher R_t fold change fits best Hauts de France and Occitanie.

Median fold changes vary between 0.174 for Île de France and 0.207 for Hauts de France. Île de France is the region where the lockdown has had the strongest effect on the R_t , contrary to expectations based on news reports. We used a linear model to investigate the relations between R_t fold change as a dependent variable and regional population size, population density, and difference between pre- and post-lockdown population sizes as explanatory variables. This difference between pre- and post-lockdown population sizes is due to migrations between regions during the few days surrounding the lockdown decree. Our linear model has an adjusted R^2 of 0.45. For each variable included in the model, we asked whether the corresponding coefficient in the linear regression was significantly different from 0. The most significant association we found was with population density, with a p-value of 0.02 and a negative correlation.

We compared the ajustement of the mixture model compared to that of model 1 by computing sums of squared errors over each day up to May 11. Squared errors are calculated for each sample between daily numbers of deaths and the numbers of deaths as predicted by each model. We found that the mixture model has a smaller error at 257950 (95% CI : 193776-345351) than model 1 at 283397 (95% CI : 211504-379692), representing a reduction of about 9% (Supplementary Figure 9). The reduction in error made by using a mixture model also varies depending on the region (Supplementary Figure 10), with the largest improvement observed in Île de France.

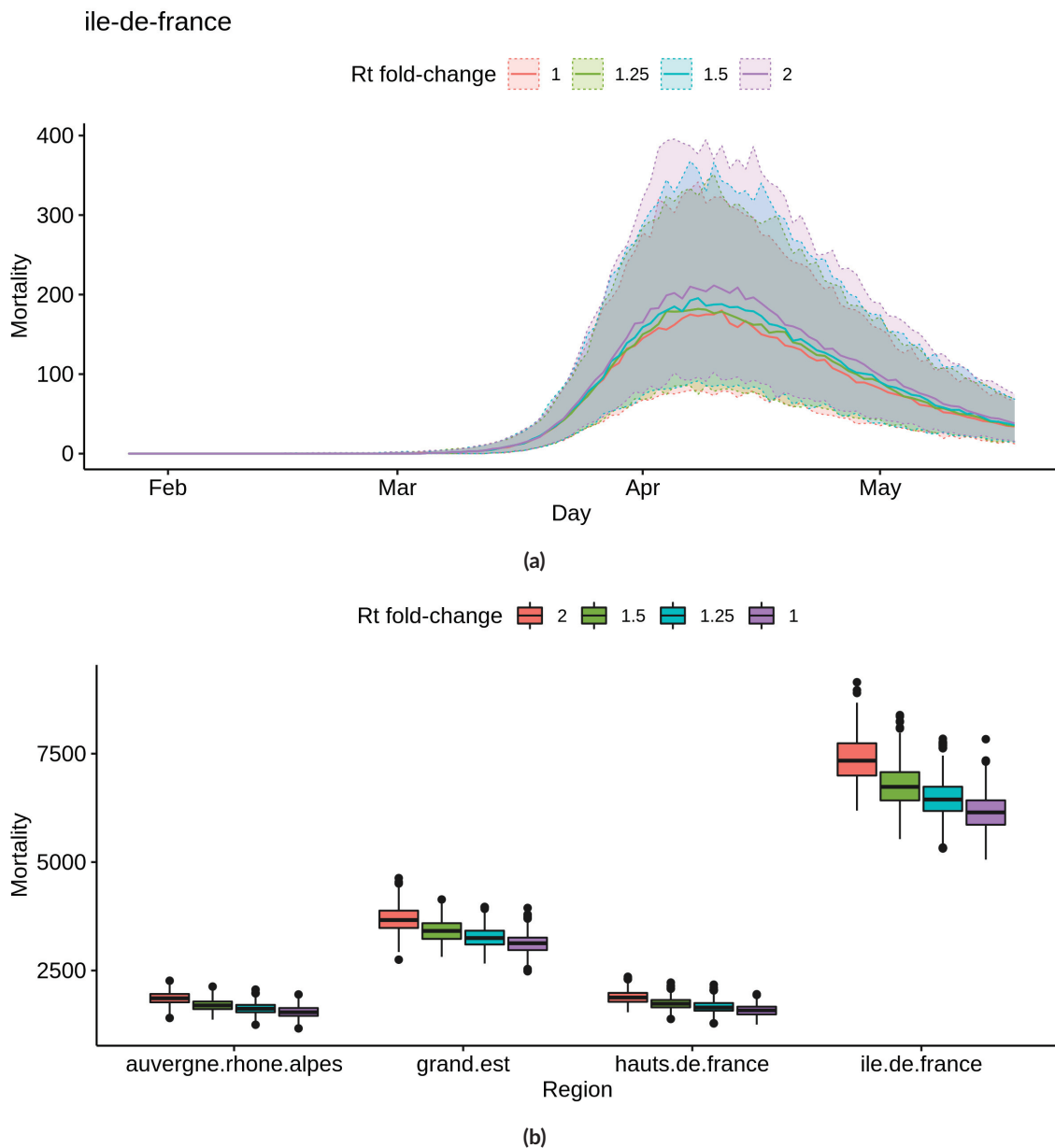


Figure 8 – Simulated distributions of deaths count, assuming different effect sizes of the election day on R_t . (a) Simulated distribution of deaths through time in region Île-de-France. Median values are represented with a solid line, and shaded areas correspond to 95% credibility intervals. (b) Distributions of the total numbers of deaths in four regions. See Figure 6b for details on the representation.

There is support in the data for using a mixture model as shown by the difference between posterior and prior distributions (Supplementary Figure 7).

However, predictions on the last week of data when fitting on the corresponding prefix of data are not enhanced through the mixture model with total squared error equal 12975 (95% CI : 7335 ; 34699) when compared to model 1 (12350 [95% CI : 7051 ; 25307]). A more thorough evaluation of prediction performances, such as cross-validation, would be necessary to conclude on the general predictive capacity of both models.

The estimates of the national average reproduction number according to the mixture model are 3.25 (95% CI : 3.10 - 3.44) before lockdown and 0.63 (95% CI : 0.59 - 0.67) after.

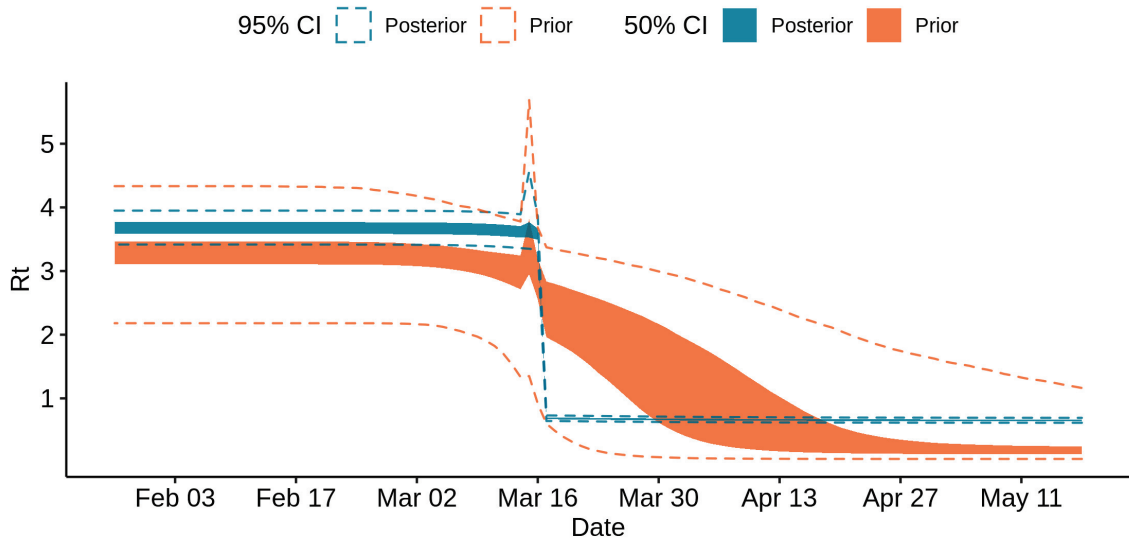


Figure 9 – Prior and posterior samples of R_t in region Île-de-France

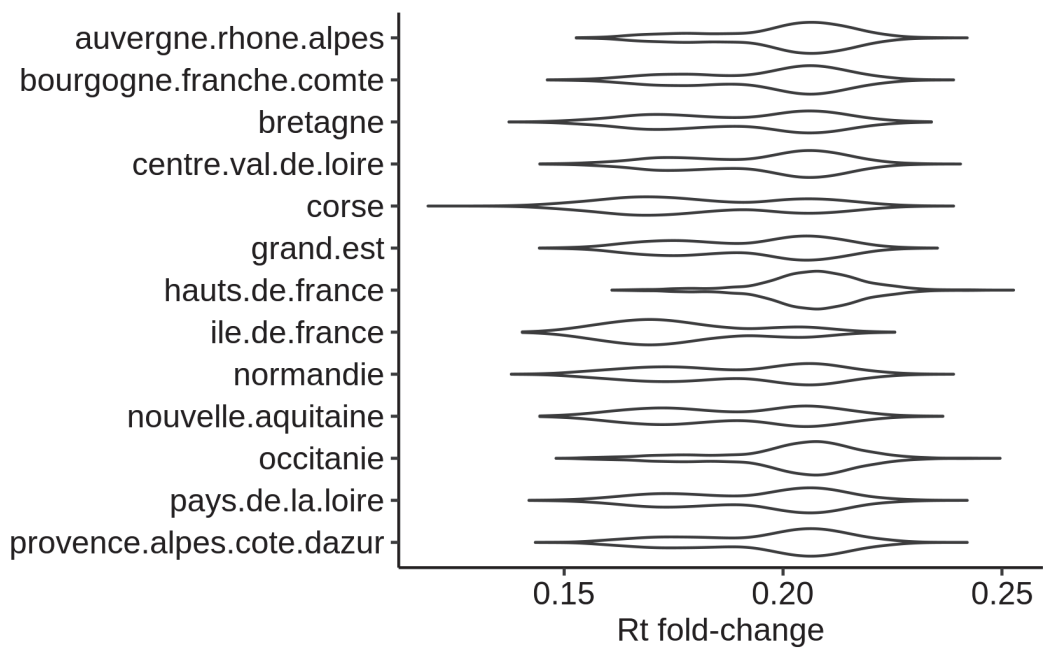


Figure 10 – Posterior distribution of R_t fold change per region

3.5. Status of the epidemic on May 11.

We used both the mixture model and model 1 to assess the status of the epidemic on May 11, the day before the lockdown was lifted. Model 1 estimates that on May 11 2.09 (95% CI : 1.69-2.66) million people had been infected. This represents 3.22% (95% CI: 2.61-4.09) of the population. Further, the model estimates that there were 2793 (95% CI : 1761-4543) new infections on May 11.

The mixture model estimates that until this date 2.08 (95% CI : 1.85-2.47) million people had been infected, representing 3.20% (95% CI : 2.85-3.81) of the population. According to this model there were 2567 (95% CI : 1781-5182) new infections on May 11.

3.6. Counterfactual investigation of alternative lockdown enforcements.

We used our models to investigate the effect of putting the lockdown in place either earlier or later than the actual lockdown date on March 17. To do so, we assessed the total number of deaths predicted by the model as of May 11, a quantity that is well estimated by model 1 and by the mixture model as seen on Fig. 4. For the mixture model, Fig. 11 shows that delays in starting the lockdown result in excess deaths: from 21% (3575) additional deaths for one day of delay to 266% (45932) for 7 days of delay. Conversely, an earlier lockdown results in lower numbers of deaths, 76% (13044) fewer deaths for 7 days, and 19% (3204) for one day. For model 1, the trend is very similar with respectively: 21% (3666), 262% (45172), 75% (12997), 18% (3098).

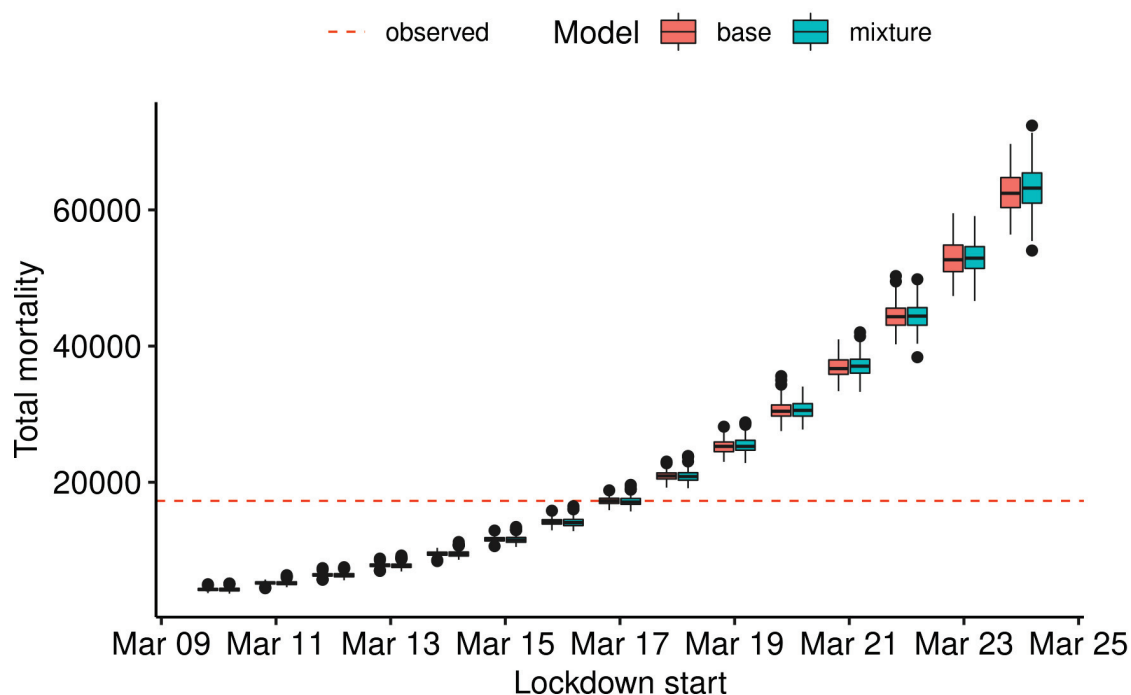


Figure 11 – Effect of different lockdown dates in counterfactual scenarios. Both models were used to predict the total number of deaths on May 11 if the lockdown was put in place up to 7 days before or 7 days after the actual lockdown date on March 17.

4. Discussion

In this manuscript, we studied the ability of a Bayesian model to fit the mortality data of the SARS-CoV-2 epidemic in France. These mortality data are incomplete, as they only include the numbers of deaths in hospitals of patients positive for the virus. In particular, they do not include deaths at home, or deaths in retirement facilities. Such input data also neglect other potentially useful sources of information, such as the number of cases, or the number of hospitalizations. Despite their shortcomings, numbers of deaths in hospitals have been widely used to study the epidemic in France and in other countries as it unfolded, notably because they were more readily available than other statistics.

We assessed the ability of our model to predict the number of deaths based on censoring of the data, and found that the model was able to accurately predict the number of deaths weeks in advance (Fig. 2).

We further explored the ability of our model using solely the number of deaths through time to detect the effect of week-ends or of single-day events, such as the election day. Week-ends would need to incur a decrease of about 20% in e.g. the number of contacts to be detectable by the model. This was not found in the empirical data. The difference between week days and week-end days is probably weaker during lockdown, because fewer people go to work on any day during the lockdown. A single-day event would need to e.g. multiply the number of contacts on that day by a factor of 2 to be detectable; expectedly, the model found no evidence for such a large effect of the elections on the number of deaths. Accordingly, another study using admissions and deaths together with regional participation to the election has also found an absence of evidence that the elections had had a detectable impact on viral spread (Zeitoun et al., 2020).

We investigated whether the lockdown had had different effects on the reproduction number in the 13 French regions. Our mixture model identified differences between regions, with Île de France showing the largest effect of the lockdown. This heterogeneity is not significantly correlated to differences in pre-lockdown R_0 , population sizes, areas, or the difference between the number of inhabitants pre and post lockdown. However, it is weakly negatively correlated to population density: the lockdown tends to be more efficient in denser regions.

Estimates obtained with the mixture model differ slightly from those obtained with model 1. For instance, nationally the average reproduction number is a bit smaller before and after lockdown (3.25 vs 3.34, and 0.62 vs 0.65). These estimates of the reproduction number can be compared to the values estimated by other groups. We focus on two works: those of (Salje et al., 2020) and (Sofonea et al., 2020).

(Salje et al., 2020) and (Sofonea et al., 2020) found results that are a bit different from ours, in particular for the reproduction number before the lockdown. The former estimated a reproduction number of 2.90 (95% CI:2.80-2.99) before the lockdown, and of 0.67 (95% CI:0.65-0.68) after the lockdown, and the latter a reproduction number of 2.99 (95% likelihood interval 2.59-3.39), and "between 21.3 and 27.1% of its value after the lockdown", *i.e.* between 0.64 and 0.81. Our credibility intervals thus overlap with the intervals of (Sofonea et al., 2020). This is interesting as (Sofonea et al., 2020) used a different model from ours, that did not take into account heterogeneities between regions, but that is based on a probabilistic fine-grain compartmental model. (Salje et al., 2020) used a Bayesian model similar to ours, except that they used both hospitalization and deaths data, but did not model the saturation of the population as the epidemic progresses and the proportion of susceptible individuals decreases in the population, and did not use a mixture model to account for heterogeneities in the lockdown efficacy between regions.

A source of difference between our model, the model of (Sofonea et al., 2020), and theirs is the values of the Infection Fatality Ratios that were used. They based their IFR on the data from the Diamond Princess cruise ship, while (Sofonea et al., 2020) and we based ours on data from Wuhan, in China. As a result, their average IFR, nation-wide, is 0.7, while ours is 0.99. We performed a test by scaling down our IFRs by multiplying them by 0.7/0.99 in model 1. We find reproduction numbers in our results are virtually unchanged by this scaling of the IFR.

Values of the reproduction number in turn affect the estimates of the total number of infected people and the total number of new infections on May 11. (Salje et al., 2020) estimate that 2.8 (range : 1.8-4.7) million people have been infected by May 11, when the lockdown was lifted, and that there were 3900 (range 2600-6300) new infections on May 11. A series of sensitivity analyses yielded a larger range of values, notably between 1700 and 9600 new infections on May 11. These values are consistent with our estimates of the number of new infections on May 11. However, the mixture model infers that only 2.08 million people had been infected by May 11 (vs 2.09 for model 1), with 2567 new infections (vs 2794 for model 1). The difference in the total number of infections with (Salje et al., 2020) is likely explained by our higher IFR: fewer infections are required to explain a given number of deaths. Indeed, down-scaling our IFRs resulted in an increase of the total number of infections to 2.71 millions (95% CI : 2.19 - 3.49) as of May 10 for model 1, closer to the estimate reported by (Salje et al., 2020). Better estimates of regional IFRs might be obtained by updating the work of (Roques et al., 2020) with more data. However, the better fit of the mixture model over model 1 suggests that the total

number of infections is probably overestimated by model 1 and by (Salje et al., 2020). Overall, this comparison with (Salje et al., 2020) and (Sofonea et al., 2020) suggests that the estimates of key parameters of the epidemic are similar across a range of models and data sources, even if they do not fully agree.

Our study of counterfactual scenarios suggests that imposing the lockdown early results in fewer deaths, and imposing the lockdown late results in more deaths, which is unsurprising given the dynamics of any epidemic. It can be put in perspective with our study of the effect of the elections on the French epidemic. Although holding the elections on Sunday March 15th did not leave a noticeable footprint in the number of deaths, it may have caused a delay in imposing the lockdown. For instance, and according to the projections of our mixture model, setting up the lockdown on Friday March 13 instead of Tuesday March 17 would have resulted in 50% fewer deaths nationwide (8557 fewer deaths as of May 11, while the estimate according to model 1 is 55% (9466 fewer deaths as of May 11)).

5. Conclusion

We used Bayesian models of the number of SARS-CoV-2 related deaths through time to study the epidemic, assess the influence of various events, and evaluate counterfactual scenarios. We found that the model accurately predicts the number of deaths a few weeks in advance, and recovers estimates that are in agreement with recent models that rely on a different structure and different input data. We also found evidence for heterogeneity between regions in the efficacy of the lockdown on epidemic spread. The predictions of the model indicate that holding the elections on March 15 did not have a detectable impact on the total number of deaths, unless it motivated a delay in imposing the lockdown.

6. Supplementary material availability

Supplementary material is publicly available on medRxiv (<https://doi.org/10.1101/2020.06.09.20126862>), under the "Supplementary Material" section.

7. Acknowledgements

The authors would like to thank (Flaxman et al., 2020) for making their model implementation open-source, thus allowing us to extend and modify it for the purpose of this research. We would also like to thank Wayne Landis and an anonymous reviewer for their fruitful comments on an earlier version of the manuscript.

A preprint version of this article has been peer-reviewed and recommended by Peer Community In Mathematical & Computational Biology (<https://doi.org/10.24072/pci.mcb.100001>).

8. Conflict of interest disclosure

The authors of this article declare that they have no financial conflict of interest with the content of this article. None of them is a *PCI Math Comp Biol* recommender.

References

- Bernard Stoecklin S et al. (2020). *First cases of coronavirus disease 2019 (COVID-19) in France: surveillance, investigations and control measures, January 2020*. *Eurosurveillance* **25**, 2000094. <https://doi.org/10.2807/1560-7917.ES.2020.25.6.2000094>.
- Cédric Pietralunga, Alexandre Lemarié, Olivier Faye (Mar. 2020). *Coronavirus : l'exécutif mis sous pression pour avoir maintenu le premier tour des élections municipales*. https://www.lemonde.fr/politique/article/2020/03/15/coronavirus-1-executif-mis-sous-pression-pour-avoir-maintenu-le-premier-tour-des-municipales_6033154_823448.html.
- Elsa Ponchon (Mar. 2020). *Coronavirus à Paris : maintenant, on ne rigole plus avec le confinement*. <http://www.leparisien.fr/paris-75/coronavirus-a-paris-maintenant-on-ne-rigole-plus-avec-le-confinement-20-03-2020-8284737.php>.

- Flaxman S et al. (Aug. 2020). *Estimating the effects of non-pharmaceutical interventions on COVID-19 in Europe*. en. *Nature* **584**, 257–261. ISSN: 0028-0836, 1476-4687. <https://doi.org/10.1038/s41586-020-2405-7>.
- French Government (2020). *COVID-19 map and data in France*. <https://www.gouvernement.fr/info-coronavirus/carte-et-donnees>.
- French Ministry of Health (2020). *Données hospitalières relatives à l'épidémie de COVID-19*. <https://www.data.gouv.fr/fr/datasets/donnees-hospitalieres-relatives-a-lepidemie-de-covid-19/>.
- Luc Peillon (Apr. 2020). *Coronavirus à Paris : maintenant, on ne rigole plus avec le confinement*. https://www.liberation.fr/checknews/2020/04/07/covid-19-pourquoi-les-chiffres-des-deces-et-des-hospitalisations-sont-toujours-plus-eleves-le-lundi_1784460.
- Magal P, Webb G (Mar. 2020). *Predicting the number of reported and unreported cases for the COVID-19 epidemic in South Korea, Italy, France and Germany*. en. *medRxiv*. Publisher: Cold Spring Harbor Laboratory Press, 2020.03.21.20040154. <https://doi.org/10.1101/2020.03.21.20040154>.
- Massonnaud C et al. (Mar. 2020). *COVID-19: Forecasting short term hospital needs in France*. en. *medRxiv*. <https://doi.org/10.1101/2020.03.16.20036939>.
- Neher RA et al. (Mar. 2020). *Potential impact of seasonal forcing on a SARS-CoV-2 pandemic*. en. *Swiss Medical Weekly* **150**. <https://doi.org/10.4414/smw.2020.20224>.
- OpenCOVID19 contributors (2020). *COVID19 epidemic french national data*. <https://github.com/opencovid19-fr/data>.
- Roques L et al. (May 2020). *Using early data to estimate the actual infection fatality ratio from COVID-19 in France*. en. *medRxiv*. Publisher: Cold Spring Harbor Laboratory Press. <https://doi.org/10.1101/2020.03.22.20040915>.
- Roux J et al. (Apr. 2020). *COVID-19: One-month impact of the French lockdown on the epidemic burden*. en. *medRxiv*. <https://doi.org/10.1101/2020.04.22.20075705>.
- Salje H et al. (May 2020). *Estimating the burden of SARS-CoV-2 in France*. en. *Science*. ISSN: 0036-8075, 1095-9203. <https://doi.org/10.1126/science.abc3517>.
- Sofonea MT et al. (May 2020). *Epidemiological monitoring and control perspectives: application of a parsimonious modelling framework to the COVID-19 dynamics in France*. en. *medRxiv*. <https://doi.org/10.1101/2020.05.22.20110593>.
- Stan Development Team (2019). *RStan: the R interface to Stan*. R package version 2.19.1. URL: <http://mc-stan.org/>.
- Verity R et al. (Mar. 2020). *Estimates of the severity of COVID-19 disease*. en. *medRxiv*. <https://doi.org/10.1101/2020.03.09.20033357>.
- World Health Organization (2020). *WHO Director-General's opening remarks at the media briefing on COVID-19 - 11 March 2020*. <https://www.who.int/dg/speeches/detail/who-director-general-s-opening-remarks-at-the-media-briefing-on-covid-19---11-march-2020>.
- Zeitoun JD et al. (May 2020). *Reciprocal association between participation to a national election and the epidemic spread of COVID-19 in France: nationwide observational and dynamic modeling study*. en. *medRxiv*. <https://doi.org/10.1101/2020.05.14.20090100>.

