



MASTERARBEIT | MASTER'S THESIS

Titel | Title

Reconstruction of coupled gain/loss dynamics in vibrio
genomes

verfasst von | submitted by

Marco Sing B.Sc.

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of
Master of Science (MSc)

Wien | Vienna, 2026

Studienkennzahl lt. Studienblatt | Degree
programme code as it appears on the
student record sheet:

UA 066 875

Studienrichtung lt. Studienblatt | Degree
programme as it appears on the student
record sheet:

Masterstudium Bioinformatik

Betreut von | Supervisor:

Ass.-Prof. Shaul Pollak Pasternak B.Sc. M.Sc. Ph.D.

Acknowledgements

First, I would like to express my deepest gratitude to my family, who have always believed in me and supported my scientific path. They made it possible for me to do my studies. I consider myself fortunate to have this opportunity.

I would also like to sincerely thank my supervisor, Prof. Shaul Pollak. He, together with the entire group, always offered support whenever I needed. I was warmly welcomed into the group, and they strengthened my motivation to continue a scientific career. Within this group, I found not only colleagues, but also friends.

I am grateful to Prof. Martin Polz for providing the dataset, helping define the thesis topic, and for laying the scientific groundwork on which this work builds.

Finally, I thank DOME for offering an environment in which I could write my thesis and improve my knowledge in the field of microbial environmental research.

Kurzfassung

Das Verständnis, welche Gene in mikrobiellen Genomen gemeinsam evolvieren, ermöglicht es, Hypothesen über ihre molekularen und ökologischen Funktionen abzuleiten. Die meisten mikrobiellen Eigenschaften werden durch ein Set von Genen bestimmt, und es wird erwartet, dass diese eine gemeinsame evolutionäre Geschichte teilen. Bei Bakterien mit umfangreicher Rekombination ist es jedoch weiterhin schwierig, echte funktionale Verknüpfungen von solchen zu unterscheiden, die lediglich aus gemeinsamer Abstammung entstehen. Diese Studie präsentiert ein phylogeniebewusstes Framework zur Identifizierung gekoppelter Gen-Gewinne und -Verluste in einem großen Genomdatensatz, unter expliziter Berücksichtigung von Rekombination. Anhand von mehr als 400 *Vibrio*-Genomen wird zunächst eine rekombinationsgefilterte klonale Phylogenie rekonstruiert. Anschließend wird die Evolution der Genfamilien mithilfe von Maximum-Likelihood-Modellen für Gen-Gewinn und -Verlust inferiert. Signifikante Assoziationen zwischen Genfamilien werden danach mit statistischen Verfahren identifiziert. Ein continuous-time Nullmodell wird verwendet, um die Evolutionsdynamik zu simulieren, und ein stratified Test dient dazu, Genpaar-Assoziationen über evolutionäre Linien hinweg zu bestimmen.

Die meisten Gene entwickeln sich unabhängig voneinander, während ein kleiner Anteil kohärente Netzwerke bildet, die Bewegungs-, Verteidigungs- und Ressourcenfunktionen zugeordnet werden können. Dies eröffnet die Möglichkeit, Gene ohne Funktionsannotation funktionell zu annotieren. Das Herausfiltern von Rekombination klärt das evolutionäre Signal und verbessert dadurch die Identifikation gekoppelter Gene. Das Framework bietet einen skalierbaren Ansatz zur Ableitung funktioneller und ökologischer Zusammenhänge direkt aus comparative genomics und ist als wiederverwendbares Python-Modul implementiert, um die Anwendung auf andere Datensätze zu erleichtern.

Abstract

Understanding which genes evolve together allows us to form hypotheses about their molecular and ecological function. Most microbial traits are encoded by sets of genes that function together and are expected to share evolutionary histories. In bacteria with extensive recombination, distinguishing true functional coupling from correlations imposed by shared ancestry remains difficult. This study presents a phylogeny-aware workflow to identify coupled gene gain and loss across large genomic datasets while explicitly accounting for recombination. Using more than 400 *Vibrio* genomes, a recombination-filtered clonal phylogeny is reconstructed and gene-family histories are mapped using maximum-likelihood models of gain and loss. Significant associations between gene families are then detected with a statistical framework combining a continuous-time null model of independent evolution and stratified tests across evolutionary lineages.

Most genes evolve independently, whereas a minority forms coherent networks linked to motility, defense, and resource-use traits, enabling functional predictions for unannotated genes. Removing recombination clarifies evolutionary signal and improves inference of gene coupling. The framework provides a scalable approach to infer functional and ecological relationships directly from comparative genomics, and is implemented as a reusable Python module to facilitate application to other datasets.

Contents

Acknowledgements	1
1 Introduction	10
2 Methods	14
2.1 Genome dataset	14
2.2 Custom scripts and automation	14
2.2.1 Visualization tools	15
2.3 Phylogenetic reconstruction	15
2.3.1 Pan/core genome alignment	15
2.3.2 Recombination inference and masking	16
2.3.3 Final phylogenetic inference	17
2.3.4 Analysis of phylogenetic outliers	19
2.4 Gene gain and loss inference	19
2.4.1 ORF prediction	19
2.4.2 COG annotation and matrix construction	20
2.4.3 Filtering low- and high-prevalence COGs	21
2.4.4 Functional annotation of COGs	21
2.4.5 Maximum likelihood modeling	22
2.4.6 Posterior probability estimation	23
2.5 Coupled gene gain and loss inference	23
2.5.1 Preprocessing and gene-level event expectations	24
2.5.2 Reconstruction of pairwise contingency tables	25
2.5.3 Stratified association testing with the Cochran–Mantel–Haenszel test	26
2.5.4 Multiple-testing correction	28
2.5.5 Parametric CTMC bootstrap null	28
2.5.6 Computational complexity	30
3 Results	31
3.1 Preprocessing	31
3.2 Phylogenetic reconstruction	33
3.3 Maximum-Likelihood inference	35
3.4 Detection of coupled gene gain and loss events	38
3.5 Comparison to CTMC null	42

3.6	Significant COG association network	45
4	Discussion	47
4.1	Evidence for coupled gene gain and loss	47
4.2	Functional patterns in high-degree COGs	49
4.3	Interpretation of significant COG associations	50
5	Conclusion	53
A		55

List of Figures

2.1	Boxplots of genome assembly statistics across 406 <i>Vibrio</i> genomes using a log10 scale, including genome size (A), GC content (B), number of contigs (C), and N50 (D). Some extreme outliers in (C) are removed for clarity.	15
2.2	Distribution of COGs across genomes. The x-axis shows, for each COG, how many genomes contain that gene. The y-axis gives the square-root-transformed number of COGs with that prevalence. Vertical lines at $x = 5$ and $x = 203$ indicate the filtering thresholds.	21
3.1	Final clonal phylogeny reconstructed from 401 <i>Vibrio cyclitrophicus</i> genomes. The tree highlights the sampling day for each genome. It marks those genomes in which recombination segments were detected and removed across all iterations. The outer ring of bars represents the genome sizes.	36
3.2	Distribution of COG gene pairs across bins. Each bin ID represents a quantile of minimum per-gene eventfulness. Bars indicate the total number of COG pairs assigned to each bin.	38
3.3	Distributions of raw and FDR-corrected significance values across all tested COG pairs. .	39
3.4	Effect size versus statistical significance for all tested COG pairs. Each point represents one gene pair. Dotted lines indicate thresholds at $\log(2)$ (two-fold odds ratio) and $q = 0.05$. 40	40
3.5	Median CMH effect size (log odds ratio) per bin.	41
3.6	Expected co-gain versus co-loss events for significantly coupled gene pairs.	41
3.7	Mean number of simulated state transitions per eventfulness bin under the CTMC bootstrap null model. Points indicate the bin-wise mean of total transitions across bootstrap simulations, with error bars showing one standard deviation.	43
3.8	Relationship between CTMC null expectations and empirical significance rates across eventfulness bins.	44
3.9	Examples of connected nodes in the coupling network created with Cytoscape. They all have strong positive effect sizes and couplings are significant. (a) A dense cluster with 46 nodes (COGs) and 1035 edges. All nodes belong to Bin ID 0. (b) A cluster with 21 nodes and 138 edges, with two bin classes: blue nodes are from Bin ID 0 and orange nodes are from Bin ID 5. (c) A smaller cluster with 6 nodes and 15 edges, where all nodes belong to Bin ID 3. (d) A small cluster in high bins, with 4 nodes and 6 edges. All nodes belong to Bin ID 8.	46

Chapter 1

Introduction

Most chemical transformations on Earth are mediated by the molecular machinery of microorganisms, with direct consequences for nutrient cycling, ecosystem function, and climate regulation [24]. Functionally associated genes are often evolutionarily coupled, such that they tend to be gained and lost together in order to preserve shared biological functions [16]. This coupling can arise because genes encoding interacting proteins, metabolic pathways, or ecologically coherent traits are subject to shared selective pressures. Coupled gain or loss can be favored in specific environmental contexts where gene combinations confer ecological advantages [12, 45]. As a consequence, patterns of gene co-evolution can be exploited to identify functional modules and to infer adaptive strategies [11, 3].

Several studies have attempted to identify coupled gene gain and loss using phylogeny-aware computational methods. Cohen and colleagues developed a probabilistic framework in which gene gain and loss are modeled as a continuous-time Markov process evolving along a phylogeny. Model parameters, including gain and loss rates, are estimated using maximum-likelihood inference [10]. By correlating reconstructed gain–loss histories across gene families, they inferred co-evolutionary relationships while controlling for spurious associations using a parametric-bootstrap null model. This framework was later implemented in the software CoPAP, which enables large-scale inference and visualization of gene co-evolution networks [11]. Applying this approach, the authors reported that nearly 40% of tested COGs participated in significant co-evolutionary associations, many of which formed functionally coherent clusters.

In a related but methodologically distinct study, Cordero et al. [13] investigated gene family coevolution at a global scale using ancestral genome reconstructions across a broad prokaryotic phylogeny. Gene family histories were inferred for 4,595 COGs using a maximum-parsimony approach. In this framework, ancestral gene family states are reconstructed along the phylogeny. Maximum parsimony favors the evolutionary scenario that minimizes the total number of gene gain, loss, duplication, and deletion events required to explain the observed gene family sizes in observed genomes. Coevolution was assessed by comparing changes in gene family sizes along phylogenetic branches using two complementary scoring approaches. The first was a quantitative score that measures how strongly changes in the copy number of two gene families are correlated across evolutionary branches. The second was a simpler, non-quantitative sign score that ignores the size of the change and instead asks whether two gene families expand or contract in the same direction on the same branches more often than expected by chance. These comparisons explicitly correct for genome-wide expansion and contraction, that is, global increases or decreases in gene family sizes that are distinct from individual gene gain or loss events.

When benchmarked against KEGG pathways by testing whether high-scoring gene family pairs were enriched for genes participating in the same pathway, the method performed better when gene duplication and deletion events were included, as these processes drive gene family-specific expansion and contraction through changes in copy number within existing families. Their results demonstrate that explicitly modeling gene duplication and deletion improves the detection of functional associations compared to approaches based just on gene gain and loss. At the same time, the study shows that many large-scale patterns observed in coevolutionary networks can in part arise from neutral duplication and deletion processes. In contrast, gene pairs that repeatedly expand or contract together across the phylogeny are more likely to reflect shared biological function. Consequently, coevolutionary networks are most informative when interpreted at the level of individual gene pairs exhibiting coordinated evolutionary changes.

Related phylogenetic approaches have been developed using alternative statistical frameworks. For example, Press et al. [71] employed BayesTraits [3] to model the correlated evolution of gene presence-absence states. BayesTraits is a Bayesian comparative framework that tests whether two binary traits evolve independently or in a statistically dependent manner along a phylogeny by comparing alternative continuous-time Markov models. The explicit aim of this approach was to infer functional associations and to annotate genes of previously unknown function. Applied to bacterial Hsp90 (hsp90A), this analysis identified 327 genes whose gain and loss histories were evolutionarily coupled to hsp90A. These evolutionary associations helped refine existing functional annotations and identify previously unknown Hsp90-associated genes.

A conceptually different phylogeny-aware approach was introduced by Li et al. [54] with CLIME (Clustering by Inferred Models of Evolution). CLIME starts from a predefined set of functionally related genes (e.g., a pathway or protein complex). It then partitions this set into evolutionary modules whose members share similar gain-loss histories along the species tree. Each module is inferred using a tree-based hidden Markov model. In this model, the hidden states represent shared evolutionary histories that give rise to the observed gene presence or absence patterns along the species tree. The model assumes a single ancestral gene gain event followed by lineage-specific loss events. It also explicitly accounts for observation errors in the gene presence-absence data. The tool further expands each inferred module by identifying additional genes in the genome whose evolutionary histories are consistent with the same evolutionary model. Rather than focusing on detecting evolutionary dependence between individual gene pairs, CLIME analyzes the gene set as a whole to uncover coherent evolutionary modules. As a result, biological pathways can be interpreted as collections of distinct evolutionary components instead of as single, uniformly evolving units.

Using a large collection of eukaryotic genomes, Dembech et al. [18] introduced a phylogeny-aware coevolution test based on shared evolutionary transitions rather than reconstructed evolutionary histories. As in the probabilistic framework of Cohen et al. [10] and the ancestral reconstruction approach of Cordero et al. [13], gene presence-absence is interpreted along a phylogeny and coevolution is defined as coordinated changes across branches. However, instead of estimating gain-loss rates or inferring parsimonious ancestral copy-number changes, the method identifies discrete presence-absence transitions for each orthogroup and tests whether two genes change state on the same branches more often than expected by chance using Fishers exact test.

The approach therefore resembles Cordero's branch-wise comparison of changes, but it neither reconstructs a minimal evolutionary scenario nor distinguishes duplication from loss, and unlike CoPAP

it avoids continuous-time Markov modeling entirely. By focusing only on the co-occurrence of events rather than their rates, model complexity is greatly reduced. This allows scaling to 60,912 orthogroups across 1,264 eukaryotic genomes and detection of over 57,000 significant gene pairs. However, this comes at the cost of reduced insight into evolutionary mechanisms compared with rate-based or reconstruction-based methods.

Collectively, these studies demonstrate that explicitly modeling gene gain and loss as stochastic processes on a phylogeny, followed by statistical testing of correlated evolutionary events, is a powerful strategy for identifying functionally linked genes. However, these approaches generally rely on species trees that do not explicitly represent clonal evolutionary history, an assumption that is problematic in recombining bacterial genomes. This limitation is addressed here by performing coevolutionary analyses on a recombination-filtered clonal phylogeny and by providing an easy-to-use, reproducible module for coupled gene gain and loss analysis.

In this context, the genus *Vibrio* represents a powerful model system for studying ecological genomics and the evolutionary consequences of gene gain and loss. Members of this genus are highly diverse marine bacteria and exhibit some of the fastest growth rates reported among free-living bacteria. Their genomes range from approximately 3.8 to 6 Mb and are larger than those of many other common marine bacteria [55]. Members of the genus are found in a wide range of ecological niches, including coastal waters, open oceans, sediments, and host-associated environments [85]. This ecological breadth is reflected at the genomic level. Genomes are shaped by frequent gene gain and loss, horizontal gene transfer, and recombination [12]. Horizontally acquired genes comprise an unusually large fraction of *Vibrio* genomes [55]. Comparative analyses have demonstrated the extent of this variability with a study of seven *Vibrio* species spanning 35 genomes. Only 11.8% of genes were shared across all strains, while 88.2% were variably present [64]. Importantly, this large variation is not limited to differences between species. Such variation can be summarized as phyletic patterns, which record the presence or absence of each gene family across genomes.

Population-genomic studies by Martin F. Polz and colleagues demonstrated that much of the variable gene content in *Vibrio* genomes is concentrated in flexible genomic regions [12, 83]. These regions, including genomic islands, are repeatedly gained and lost across closely related lineages [12]. By comparing closely related strains sampled from the same environments, these studies showed that strains with nearly identical core genomes can differ substantially in their accessory gene content [12, 83]. Many of these regions encode functions related to resource utilization, defense, or interactions with hosts [12, 83]. Polz and colleagues further showed that these genes are not distributed randomly across populations [12]. Particular combinations of genes tend to be associated with specific ecological niches, leading to fine-scale genetic differentiation among co-occurring populations [12, 83]. This pattern suggests that repeated gain and loss of flexible gene sets enables rapid ecological adaptation [12]. However, previous work has largely focused on identifying such regions or linking gene content to ecological niches, rather than systematically reconstructing gene gain and loss histories across entire phylogenies.

This thesis builds on these insights by establishing a robust workflow to detect coupled gene gain and loss across more than 400 genomes. Yu et al. [91] demonstrate at the population level that adaptive structure becomes substantially clearer once the clonal evolutionary backbone is reconstructed and recombination is explicitly separated from vertical inheritance. A central contribution of this work

is the reconstruction of a clonal phylogeny that serves as the basis for mapping gene-family histories. Because ancestral state reconstruction depends critically on tree topology, and recombination has been shown to substantially distort phylogenetic relationships [91]. Removing recombination provides a better suited framework for inferring gain and loss events. The approach is designed to distinguish true functional coupling from spurious correlations arising from shared evolutionary history. Branch-specific gene gain and loss events are reconstructed along the clonal tree, enabling identification of independent co-occurrences across the lineage. Statistical methods are then applied to detect gene pairs whose evolutionary histories are more strongly coupled than expected by chance. Such patterns can indicate functional linkage [10] but may also reflect shared ecological strategies rather than direct molecular interactions, as shown by Pollak et al. [70], who found that genes coevolving with chitinases encode traits underlying public-good exploitation, surface colonization, and resource competition. These relationships can therefore be used both to annotate genes with unknown functions and to infer ecological roles and interaction strategies directly from genomic data.

The workflow is designed to be scalable, statistically robust, and applicable to diverse prokaryotic genome datasets while remaining biologically consistent. In practice, the phylogeny was iteratively refined using *ClonalFrameML* to identify and remove recombination-affected regions, yielding a clonal tree suitable for evolutionary inference. Gene-family evolution on this tree was then modeled in *COUNT*, which estimated maximum-likelihood gain and loss rates. Building on these inputs, a Python module (working name: *ProGeneCoupling*) was implemented to detect significantly coupled gene pairs from a phylogenetic tree and a binary presence–absence matrix. The method combines a continuous-time Markov chain null model of independent evolution with a Cochran–Mantel–Haenszel test across branches, followed by FDR correction to control multiple testing. The results indicate that most genes evolve independently, whereas a subset shows significant associations with large effect sizes. Gene–gene association networks derived from these pairs form coherent clusters, enabling functional assignment of previously uncharacterized genes. Comparison of the identified clusters with published biological knowledge revealed consistent functional relationships, supporting the validity of this workflow. For example, one cluster links a core flagellar gene with an abortive infection (Abi) phage-defense component and several unannotated genes, suggesting coordinated evolution between motility and phage defense and allowing functional hypotheses for the uncharacterized members. The *ProGeneCoupling* module could be extended in the future to incorporate additional steps of the pipeline and may ultimately be released as a tool for annotating unknown genes and identifying shared ecological strategies.

Chapter 2

Methods

2.1 Genome dataset

The data were provided by a group led by Martin F. Polz at the University of Vienna, Center for Microbiology and Environmental Systems Science. He conducted a large-scale sampling campaign in 2010, during which water samples were collected over a three-month period. The resulting isolates represent co-existing ecological populations sampled from Canoe Cove in the Nahant coastal region of the USA ($42^{\circ}25'10.6''\text{N}$, $70^{\circ}54'24.2''\text{W}$), commonly referred to as the Nahant *Vibrio* collection [79]. Co-existing ecological populations are genetically distinct groups of bacteria that live in the same environment at the same time but occupy different ecological niches [79]. The shallow, semi-enclosed coastal system supports dense and genetically structured *Vibrio* populations. Strong seasonal fluctuations in temperature and nutrient availability further reveal ongoing eco-evolutionary dynamics within the community [12].

For the present study, bacteria were collected on seven closely spaced sampling days between August 20 and September 4 (ordinal days 232, 236, 237, 239, 242, 244, and 247), yielding a total of 408 *Vibrio* genomes [60, 45]. Assembly statistics were computed for all genomes (Figure 2.1) and indicate consistently high assembly quality based on established benchmarks [57]. Further details on preprocessing and assembly evaluation are provided in Section 3.1.

The naming convention of *Vibrio* genome sequence files is illustrated by the example `10N.237.46.C11`. The prefix `10N` denotes the sampling year (2010) and location (Nahant), followed by the ordinal sampling day (here, day 237 corresponding to August 22). The subsequent number specifies the size fraction used during isolation. All samples in this study were obtained using a $0.2\ \mu\text{m}$ filter to target bacterial cells. A unique storage identifier was appended to complete the file name [60].

2.2 Custom scripts and automation

All program executions listed here were implemented using Python scripts (Python version 3.9.21). This increases reproducibility and reduces the potential for errors during execution. The scripts forward parameters to the respective programs to be executed. In addition, log files are generated and, in some cases, further statistics are collected to support analysis. Most programs require specific formats, therefore scripts also handle data preparation and parsing. A full list of scripts is provided in Appendix A. Because there is no simple and widely available tool to infer gene gain-loss coupling from evolutionary

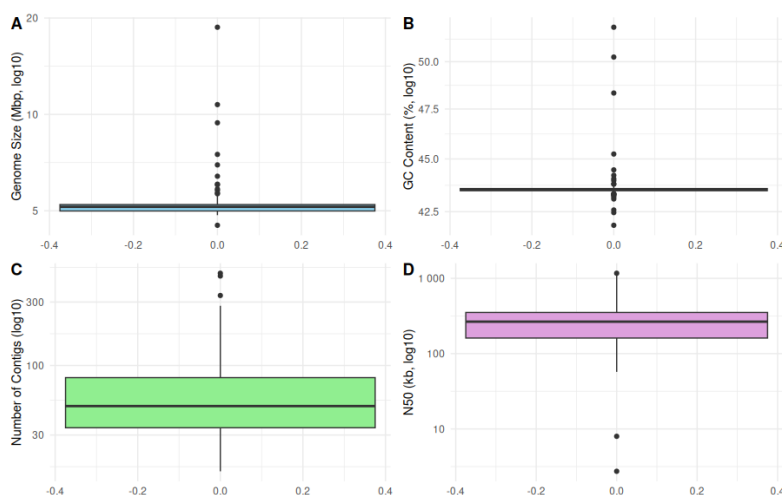


Figure 2.1: Boxplots of genome assembly statistics across 406 *Vibrio* genomes using a log10 scale, including genome size (A), GC content (B), number of contigs (C), and N50 (D). Some extreme outliers in (C) are removed for clarity.

data, a Python module was implemented. The set of scripts that implement the full workflow is also listed in the Appendix. Scripts and a documentation with usage and parameters are available at: <https://gitlab.com/Marsi03/bioinformatics>.

Computational results of this work have been achieved using the Life Science Compute Cluster (LiSC) of the University of Vienna. LiSC provides access to high-performance compute nodes via batch scheduling using the SLURM manager. Jobs are submitted as SLURM batch scripts, and compute partitions are assigned automatically.

2.2.1 Visualization tools

Statistical results produced by the pipeline were post-processed using R (version 4.5.0) [73]. For network-based analyses of coupled gene gain and loss, Cytoscape (version 3.10.3) was used. Cytoscape is an open-source platform for visualizing and analyzing biomolecular interactions [78].

2.3 Phylogenetic reconstruction

2.3.1 Pan/core genome alignment

The core genome was aligned using Parsnp2 (version 2.1.3) [46]. Alignments of conserved core regions provide a robust phylogenetic backbone while reducing the influence of genes introduced through horizontal gene transfer (HGT). Parsnp2, released in 2024, is an improved successor to the original Parsnp tool introduced in 2014 and does not require prior genome annotation. Internally, it estimates pairwise genome distances using the MUMi metric [17], which guides the alignment process. It further infers a maximum-likelihood phylogeny from the resulting alignment using RAXML [81], a widely used program that reconstructs phylogenetic trees by estimating the most likely tree structure and branch lengths given the sequence data. Parsnp2 is optimized for low runtime and memory usage, making it particularly well suited for large genome datasets. To further improve scalability, genomes are divided into fixed-size partitions. Because core regions are shared across all genomes, this strategy reduces memory

requirements and enables efficient parallelization [46].

Before alignment, genome files were decompressed when necessary, and only valid FASTA files were retained for further processing. Parsnp2 requires a reference genome to anchor the alignment. In the Python workflow, an appropriate reference genome was selected prior to execution. Several selection strategies were evaluated, and the best results were obtained using the genome 10N.244.45.B01.fa (length = 5.153 Mbp, 23 contigs). This genome was chosen because it ranked within the top 25% of assemblies with the fewest contigs and had a genome size closest to the median of the dataset.

Parsnp2 was run with default parameters using eight CPU cores. The output included log files, the whole-genome XMFA alignment, and an initial phylogenetic tree in Newick format. Genomes exceeding the reference genome size by more than 1.3× were automatically excluded by Parsnp2.

2.3.2 Recombination inference and masking

Initial detection of recombination areas

To refine the phylogenetic tree and increase its biological relevance, recombination regions were identified and removed. Horizontal recombination transfers homologous sequence tracts between unrelated lineages, making them appear more closely related while making their actual relatives seem more distant, thereby distorting branch lengths and sometimes topology [19]. If recombination is not accounted for, the inferred clonal tree may therefore be biased.

ClonalFrameML (version 1.13) [20] was used in this study. The software was applied to the Parsnp2 tree and the corresponding core genome alignment. The input tree is treated as the clonal genealogy and serves as the ancestral backbone. Given this tree, ClonalFrameML reconstructs ancestral genome sequences at all internal nodes using maximum-likelihood methods. As a result, each branch is associated with a reconstructed ancestral sequence and its corresponding descendant sequence.

For each branch, ClonalFrameML then compares sequences site by site. Evolution along a branch is modeled using a hidden Markov model with two hidden states [22]. One state represents evolution by point mutation, and the other represents sequence changes introduced by recombination. State transition probabilities are estimated using the forward–backward algorithm, which computes the expected usage of each state at every site. These expectations are subsequently used in an expectation–maximization (Baum–Welch) algorithm [4] to iteratively update parameters describing recombination, including how often recombination occurs, how long recombined regions tend to be, and how different imported sequences are from the resident genome, until convergence. After parameter estimation, the Viterbi algorithm [87] is applied to infer the single most likely sequence of hidden states, thereby identifying genomic regions that were likely introduced through recombination [20].

ClonalFrameML outputs an `importation_status.txt` file containing the genome name as well as the start and end positions of regions inferred to result from recombination. Several additional files are produced to support downstream analysis. The `em.txt` file records the progress of the EM algorithm. The `position_cross_reference.txt` file maps inferred recombination events to genomic coordinates. The `ML_sequence.fasta` file contains reconstructed ancestral sequences. The `labelled_tree.newick` file stores the final phylogenetic tree. Importantly, ClonalFrameML modifies only branch lengths and does not change tree topology.

ClonalFrameML was executed using a Python wrapper script. A `.newick` tree file and a `.xmfa` align-

ment file were provided as input. The option `-em true` was used to enable parameter optimization, while `-ignore_incomplete_sites false` ensured that all alignment sites were included in the analysis. For highly gapped alignments, this option can alternatively be set to `true` to exclude sites with missing data. All analyses were performed on a single CPU core of a Linux workstation, as ClonalFrameML does not support multithreading.

Because tree tip labels must exactly match alignment identifiers, sequence headers in the XMFA file were rewritten. This step was necessary because the alignment tool Parsnp2 introduces its own genome labels, which would otherwise be inconsistent with those used in the phylogenetic tree.

Removal of recombinant regions

ClonalFrameML modifies only the branch lengths of the input tree. To obtain a new tree topology, recombination areas in the XMFA alignment file were masked. The masking was performed using dashes (-). Regions defined in the ClonalFrameML output file `importation_status.txt` were used for this purpose. The masking preserved the XMFA block structure and produced a recombination-reduced alignment. This alignment was then used for a subsequent iteration.

2.3.3 Final phylogenetic inference

Maximum likelihood tree with IQ-TREE3

To infer a recombination-free phylogeny with a newly estimated topology, IQ-TREE3 (version 3.0.1) [90] was used. IQ-TREE3 was released in 2025 and is the latest version of the IQ-TREE phylogenomic inference software. It performs phylogenetic tree reconstruction using a maximum-likelihood (ML) framework and is designed to be both fast and computationally efficient. Due to its scalability, IQ-TREE is well suited for large datasets such as the one analyzed here and has been shown to outperform several comparable tools [88].

Prior to tree inference, the XMFA alignment was converted into FASTA format to ensure compatibility with IQ-TREE. Model selection was enabled using the `-m MFP` option, which invokes ModelFinder Plus to identify the best-fitting substitution model according to the Bayesian Information Criterion (BIC) [43]. Selecting an appropriate substitution model is important because model misspecification can bias likelihood estimates and lead to incorrect tree topologies or branch length estimates [52]. Branch support was assessed using two complementary approaches. Ultrafast bootstrap approximation was performed with 1,000 replicates (`-B 1000`) using UFBoot2, which generates bootstrap trees by resampling the alignment [35]. In addition, the Shimodaira–Hasegawa approximate likelihood ratio test (SH-aLRT) was conducted with 1,000 replicates (`-alrt 1000`), providing a likelihood-based measure of support for alternative tree topologies. The combined use of UFBoot and SH-aLRT yields independent support metrics and should result in robustly supported branches.

Multithreading was enabled using automatic thread detection with an upper limit of eight threads (`-T AUTO -ntmax 8`). Among the output files produced by IQ-TREE3, the `.treefile` contains the inferred ML tree in Newick format.

Targeted recombination detection in closely related genomes

Recombination detection methods work by identifying regions that contain more substitutions than expected from normal mutation. If genomes are very similar, recombined segments contain only a few differences and therefore look similar to ordinary mutations. In such cases, the signal required to distinguish recombination from mutation becomes weak, and recombination may not be reliably detected. Additionally, closely related genomes may have diverged only recently, so recombined regions have not yet accumulated enough differences to be detected, or recombination may simply be rare in this group.

In the initial whole-dataset analysis, several clades consisted of very closely related genomes in which recombination was not reliably inferred. These genomes were therefore analysed separately. They were temporarily excluded from the main core genome alignment and re-aligned independently. Restricting the analysis to a low-divergence subset changes the statistical background against which substitutions are evaluated. Substitutions that appear homogeneous across the full dataset may form localized clusters within a closely related clade. As a result, recombination tracts that are masked by high divergence elsewhere in the tree can become detectable when only closely related genomes are considered.

A phylogenetic tree for this subset was inferred with IQ-TREE using the same parameters described in Section 2.3.3. The resulting tree was then used as input for an independent ClonalFrameML run in order to detect recombination events specific to this low-divergence subset.

Recombinant regions inferred in this targeted analysis were masked following the procedure described in Section 2.3.2. The masked subset alignment was subsequently merged back into the global core genome alignment. Finally, ClonalFrameML was executed again on the full dataset using the IQ-TREE3 phylogeny and the updated alignment (parameters identical to Section 2.3.2). This iterative strategy should increase sensitivity to recombination in low-divergence clades while maintaining a consistent genome-wide phylogenetic framework.

Iterative recombination detection

The recombination-detection procedure was then applied iteratively. After each ClonalFrameML run, recombinant regions were masked based on the inferred importation events, and IQ-TREE3 was used to reconstruct a new maximum-likelihood phylogeny. Changes in tree topology were interpreted as evidence for previously undetected recombinant segments. The updated tree was then used as input for the next ClonalFrameML run. This iterative loop comprising ClonalFrameML, masking, and IQ-TREE3 reconstruction was repeated until successive iterations no longer produced substantial topological changes. In practice, three rounds of ClonalFrameML and three corresponding IQ-TREE3 reconstructions were sufficient, after which the inferred phylogeny stabilised. The final result is a clonal phylogeny that appears to have converged within the constraints of the method.

To quantify changes in tree structure across iterations, all pairs of inferred phylogenies were compared using three distance metrics: the Robinson–Foulds (RF) distance [76], the Branch Score Distance (BSD) [49], and the mean branch-length difference. Trees were read from Newick files. Topological distances (RF) were computed using a custom implementation based on Biopython’s `Phylo` module, which extracts all non-trivial splits (i.e. splits for which both partitions contain at least two taxa) and counts those present in only one of the two trees. The RF distance is a purely topological measure, with larger values indicating stronger rearrangements of branching structure. The BSD was computed using PHYLIP’s `treedist`

program and additionally accounts for branch lengths associated with each split. It therefore captures differences in both clade composition and branch-length estimates. Finally, the mean branch-length difference was used to quantify changes in branch-length scaling alone.

Graphical representation of the tree

For the visualization of the final phylogeny, the Interactive Tree Of Life (iTOL, version 7.3) online platform was used [53]. The underlying tree was provided as a Newick file generated in the final IQ-TREE3 inference round. To add features to the visualization, three additional iTOL-compatible annotation files were created. First, a color-strip dataset was constructed. It groups genomes by the ordinal sampling day encoded in their filenames (e.g. 10N.42.xyz.fasta). Second, a simplebar dataset was added to display genome sizes. All lengths were normalized to a maximum bar height of 100. Finally, the ClonalFrameML importation status output was processed to compute the per-genome recombination burden. It is defined as the fraction of each genome affected by recombination. It was exported as a gradient dataset for iTOL.

2.3.4 Analysis of phylogenetic outliers

In the first run with all 408 genomes, two genomes were identified as phylogenetic outliers that distorted the overall tree topology. The two responsible genomes (10N.244.45.D09.fa, 10N.244.45.B06.fa) were removed from the dataset, and the subsequent analysis was carried out with 406 genomes. These two outlier genomes were further analyzed and compared with genomes from other clades. The genomes selected for comparison were randomly chosen from different parts of the tree. Care was also taken to ensure that they were sampled on different days (10N.247.45.A10, 10N.244.46.D10, 10N.239.45.H11, 10N.239.45.F10, 10N.237.45.C05).

A BLASTN (Basic Local Alignment Search Tool for nucleotides) search was used for the characterization of the genomes. For every contig's BLAST result, the number of hits per species was counted. BLAST (BLAST+, version 2.16.0) was run against a locally mirrored copy of the public NCBI nucleotide database (as of 26.04.2025).

2.4 Gene gain and loss inference

2.4.1 ORF prediction

Open reading frames (ORFs) across the *Vibrio* genomes were predicted using Pyrodigal (3.6.3) [51], a python wrapper of Prodigal (PROkaryotic DYnamic programming Gene-finding ALgorithm). Pyrodigal optimized the Prodigal procedure further. Its Python interface allows seamless integration into this analysis pipeline. Prodigal was chosen based on a comprehensive benchmark by Dimonaco et al. [21], which evaluated 15 different coding sequence (CDS) prediction tools using 72 metrics. It included metrics such as the percentage of genes detected or the percentage of perfect matches. Among all tools tested, Prodigal and GeneMarkS [6] ranked as the top performers in 2021. Ultimately Pyrodigal was chosen because its easy to integrate and has the better computational efficiency.

Prodigal uses an unsupervised, data-driven approach. It first identifies potential genes by detecting GC content biases in coding frames. The most likely set of genes is then selected using a dynamic

programming approach. That is done based on features like start codon preference and ribosome binding site motifs. During training, Prodigal scans the genome to learn the preferred RBS motif, such as the Shine-Dalgarno sequence [38].

A representative subset of genomes was selected to train a model. All contigs from the selected genomes were concatenated, and the training set was provided to `pyrodigal.GeneFinder.train()` to infer optimized gene prediction parameters. To construct the training model, training set sizes of 1, 5, and 20 genomes were evaluated. These genomes were selected as the first files (in lexicographic order) from the input dataset. No filtering based on genome length or contig count was applied. The different runs showed that the ORF results, in terms of both total number of predicted genes and their confidence scores, were nearly identical. For further analyses, the model trained with 5 genomes, was selected. The resulting model was saved for reuse.

In the second stage, this trained model was applied to predict the ORFs in all 406 genomes. Each genome file was decompressed and parsed using Biopython's SeqIO. It was then passed to `GeneFinder.find_genes()`. For each predicted gene, metadata such as genomic coordinates, confidence scores, start codon type, GC content, and five internal scores (score, cscore, rscore, tscore, uscore) were recorded. Protein translations were also written to FASTA format.

2.4.2 COG annotation and matrix construction

ORF prediction identifies the genomic locations of protein-coding genes in each genome. Across the 406 genomes analyzed here, this resulted in a very large number of predicted ORFs. Direct comparison of individual ORFs across genomes is neither computationally feasible nor biologically robust. Therefore, similar genes were grouped into gene families for downstream analysis.

Clustering genes into families reduces noise introduced by annotation inconsistencies. Minor differences in gene boundaries or prediction errors are unlikely to affect whether a gene family is scored as present or absent when homologous genes are grouped together. In this study, ORFs were clustered into clusters of orthologous groups (COGs), which represent sets of genes from different organisms that are inferred to share common ancestry and, in many cases, similar biological functions. This clustering relies primarily on sequence similarity [29].

After assigning all ORFs to COGs, a binary presence-absence matrix was constructed. Rows correspond to genomes and columns correspond to COGs, with entries indicating whether at least one member of a given COG is present in a genome.

Gene clustering was performed using FastCOG, an in-house COG identification pipeline written in Julia by Prof. Shaul Pollak based on a COG clustering approach described in Kristensen et al. [48]. The tool takes FASTA files as input, with one file per genome containing all predicted ORFs. Each sequence header encodes the genome identifier, contig, gene number, genomic coordinates, and strand. These FASTA files were generated from the output of the Pyrodigal program.

FastCOG produces multiple output files during clustering. The `parts_clu` directory contains pairwise protein matches grouped by cluster, while individual protein sequences are written to the `parts` directory in FASTA format. All protein sequences are additionally concatenated into a single `prots.faa` file. The complete clustering result is stored in the `cogs.clu` file.

For all subsequent analyses, only the `cogs.clu` file was used. Protein identifiers were parsed to extract the genome, contig, gene number, start and end coordinates, strand, and assigned COG ID. This

information was compiled into a table named which served as the basis for constructing the final binary presence–absence matrix.

2.4.3 Filtering low- and high-prevalence COGs

Most COGs in the dataset are extremely rare, present in only one or a few genomes, or almost universal, as seen in Fig. 2.2. These extremes provide very little usable variation for the analysis. Rare COGs tend to produce unstable and noise-driven correlations. Ubiquitous COGs could appear perfectly correlated with each other simply because they never vary. These patterns do not reflect true shared evolutionary history.

To avoid these artefacts, only COGs that appear in at least five genomes were retained. This follows the rationale of Cohen et al. [11], who showed that COGs with too little variation inflate co-evolution signals by chance. Although their threshold was defined using expected gain/loss events, the underlying principle is the same. COGs that are included in nearly all genomes were also excluded. The upper threshold was chosen to be 203, which means that COGs that appear in more than half of the genomes are removed from further analyses. Analyses should focus on COGs with enough presence-absence variation to be informative. Applying also an effective threshold from the top on, ensured that downstream coupling tests are carried out only on COGs with meaningful evolutionary signal.

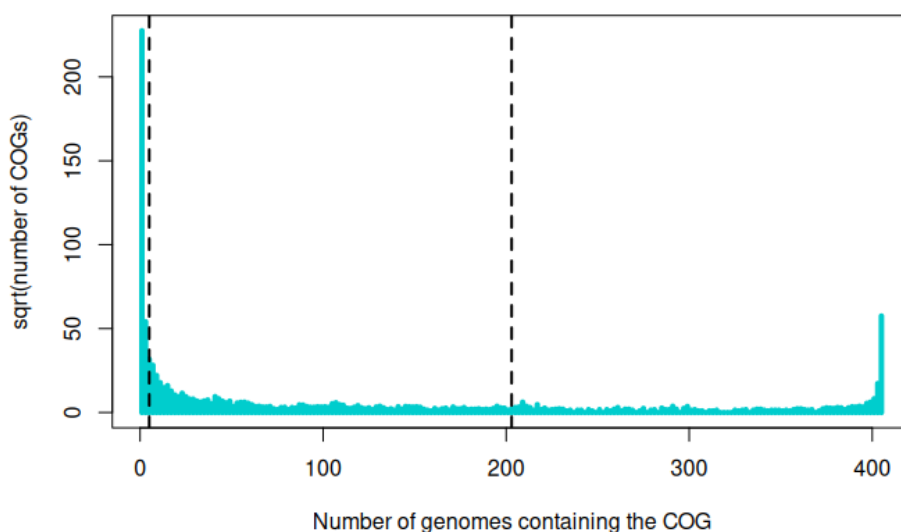


Figure 2.2: Distribution of COGs across genomes. The x-axis shows, for each COG, how many genomes contain that gene. The y-axis gives the square-root-transformed number of COGs with that prevalence. Vertical lines at $x = 5$ and $x = 203$ indicate the filtering thresholds.

2.4.4 Functional annotation of COGs

For downstream analyses, functional annotations were required at the level of COGs. Protein-coding sequences from all genomes were therefore combined into a single FASTA file and annotated using eggNOG-mapper (version 2.1.12). eggNOG-mapper assigns functional annotations by mapping query

proteins to precomputed orthologous groups from the eggNOG database, which is constructed across multiple taxonomic levels. Orthologous group assignments are validated using gene trees to ensure evolutionary consistency [37, 9].

Each COG in this study comprised multiple genes. Functional annotations were consolidated at the COG level by aggregating gene-level annotations. For each COG, the most frequently assigned functional description among its member genes was selected as the representative annotation. To perform this step, a custom parsing script was implemented to process the eggNOG-mapper output and summarize annotations per COG. The script generates a table containing the fields `COG`, `top_description`, `top_count` (number of genes assigned to the most frequent description), `total` (total number of genes in the COG), and `percentage` (fraction of genes supporting the top annotation).

In addition to COG descriptions, KEGG-related annotations were incorporated, including KEGG orthology identifiers as well as pathway and module assignments. The Kyoto Encyclopedia of Genes and Genomes (KEGG) is a curated library that links genes and proteins to metabolic pathways, cellular processes, and higher-level functional systems [44]. These annotations were used to support pathway-level interpretation of coupled gene gain and loss patterns.

2.4.5 Maximum likelihood modeling

A central step in the pipeline for predicting coupled gene gain and loss is maximum-likelihood (ML) modeling. The resulting branch-specific rate estimates are later used to compute per-branch gain and loss probabilities and serve as parameters for the continuous-time Markov chain (CTMC) simulation model. ML inference identifies gain and loss rates that maximize the probability of the observed gene presence-absence patterns. Intuitively, these rates describe how often a gene is expected to be gained or lost along a branch per unit of evolutionary time. For a given set of rates, the likelihood is computed at the tips of the phylogenetic tree, and parameters are iteratively updated until the overall likelihood converges to a maximum [25].

These computations were performed using the program COUNT [15], which models gene family evolution under maximum-likelihood or parsimony frameworks. COUNT can be run either through a graphical user interface or via the command line and requires a working Java installation. It supports three main analysis modes: ancestral gene content reconstruction using Wagner parsimony, parameter optimization for phylogenetic birth-death models, and ancestral gene content inference using posterior probabilities under an optimized birth-death model [14].

In this project, COUNT was used exclusively as a command-line application, with the surrounding workflow implemented in Python. The pipeline programmatically constructs the required command-line arguments and invokes the ML optimization module. The inputs consist of a rooted phylogenetic tree in Newick format and a binary gene presence-absence matrix. To assess computational feasibility, preliminary runs were performed with fewer optimization rounds and a reduced number of gamma-distributed rate categories. As noted by Csűrös [14], full ML optimization can require several days of computation. Subsequently, multiple additional rounds were carried out with progressively increased parameter settings to obtain the final estimates.

The final analysis employed a per-branch model in which each branch has its own gain, loss, and duplication rates. Branch lengths were set to be uniform. Three gamma-distributed rate categories were used for both gain and loss processes. The maximum number of optimization rounds was set to 100, with

a convergence threshold of 0.01, following the recommendations in [14]. The parameter *max_paralogs* was set to 203, defining the maximum total number of gene copies allowed per family across all genomes. Families are retained only if

$$m \geq \sum_{j=1}^n \Phi_{fj},$$

where Φ_{fj} denotes the copy number of family f in genome j . Because the input matrix in this project is binary, a family present in 203 genomes also has a total copy number of 203. As shown in Fig. 2.2, no family exceeds this threshold in the dataset.

Verbose output (-v) was enabled to monitor the progress of the ML optimization in real time, while all other parameters were kept at their default values. The ML run produces a single output file containing the Newick tree annotated with branch-specific gain, loss, and duplication rates. This file was parsed using a custom script to generate a structured CSV file for downstream analyses, including explicit node-parent relationships and the estimated rates for each branch.

2.4.6 Posterior probability estimation

After estimating normalized gain and loss rates with the ML program in COUNT, ancestral gene content was inferred using the Posteriors program. This method reconstructs ancestral states under a phylogenetic birth-death model and is explicitly conditional on the previously estimated model parameters. Consequently, the branch-specific gain and loss rates obtained from the ML step are required to define transition probabilities along each branch.

The Posteriors program integrates over all possible ancestral gene histories and computes, for each branch, the posterior probability of a gene family being gained (absence to presence) or lost (presence to absence). It also reports probabilities for expansions and reductions, corresponding to transitions between different copy numbers [14, 36]. In the present dataset, these quantities are not relevant, because each COG is represented as a binary trait. The input matrix does not contain copy numbers greater than one.

The analysis was run using default parameters and required three inputs: the phylogenetic tree, the presence/absence matrix, and the branch-specific rate file produced by the ML step. Naming internal nodes in the input tree is recommended, as otherwise numerical identifiers are automatically assigned in the output. The results are provided as a single TSV file, which was subsequently parsed to construct branch×COG gain and branch×COG loss matrices. No posterior probability threshold was applied at this stage. Instead, probabilities were later aggregated across branches, allowing each branch to contribute proportionally according to its inferred weight rather than being subject to a hard cutoff.

2.5 Coupled gene gain and loss inference

Although gene-content evolution has been extensively studied, no established software currently infers statistical coupling between gene gain and loss events along a bacterial phylogeny. To address this gap, a Python module was developed, currently referred to by the working name ProGeneCoupling. An overview of all scripts used in this module is provided in Appendix A.1.

The approach combines posterior-weighted reconstruction of pairwise contingency tables, stratified significance testing, and parametric calibration under a continuous-time Markov chain (CTMC) model.

The design prioritizes computational scalability while maintaining biological interpretability.

The module requires three main inputs. First, a rooted phylogenetic tree that is compatible with the gene-content data. In this study, the tree was inferred from a recombination-filtered core-genome alignment. Second, a branch-by-COG matrix of gain and loss posterior probabilities, obtained using the model-based gene family reconstruction tool COUNT. Third, a table of branch-specific rates estimated on the same tree, which was likewise produced by COUNT.

A diagnostics class was added to provide a concise summary for assessing whether results are sensible and well calibrated. The diagnostics generate a `global_summary.json` file describing significance levels, effect sizes, and p-values across all tested gene pairs, as well as a `top_hits.parquet` table containing the strongest gene–gene associations. In addition, a `ctmc_comparison.json` file is written to compare observed significance rates with expectations under the CTMC null model.

A logging class was also implemented to ensure consistent and reproducible recording of progress, warnings, and errors throughout the pipeline. Configuration parameters used for the final analysis are listed in Appendix Table A.2. Although the parameter `n_workers` is included, the final analysis was executed with `n_workers = 1`, as parallel execution is not yet supported in the current implementation.

2.5.1 Preprocessing and gene-level event expectations

The raw posterior gain and loss matrices may include COGs that contribute little evolutionary information. In particular, gene families for which posterior probabilities are close to zero across most or all branches introduce a strong imbalance in downstream analyses. To address this, the preprocessing module computes, for each gene family, the expected number of gain and loss events by summing posterior probabilities across all branches,

$$E_{\text{gain}} = \sum_b P(\text{gain on branch } b), \quad E_{\text{loss}} = \sum_b P(\text{loss on branch } b).$$

Each gene is then assigned an *eventfulness score*, defined as

$$\min(E_{\text{gain}}, E_{\text{loss}}),$$

which captures the extent to which a gene both appears and disappears along the phylogeny. Gene families that experience only gains or only losses have one of the two expectations close to zero and therefore yield low eventfulness scores. Such genes provide limited information for testing coupled gain and loss, as co-evolutionary signals require variation in both directions. COGs whose eventfulness falls below a user-defined threshold are therefore excluded.

This filtering strategy follows the approach introduced by Cohen et al. [10]. In the present analysis, eventfulness filtering had already been applied at an earlier stage (Fig. 2.2), and a minimum threshold of one expected event was retained here to avoid redundant exclusion. For future analyses, applying eventfulness filtering at this stage may be necessary when raw posterior matrices are used as input.

Based on the empirical distribution of eventfulness scores, the preprocessing module constructs quantile-based bin boundaries. In this analysis, ten bins were used, such that COGs were grouped into ten discrete categories with comparable levels of evolutionary activity. These bins are used to calibrate the CTMC-based null model. They are also used to organize diagnostic summaries, ensuring

that differences in baseline gene-level activity are explicitly accounted for. Associations between gene families can be evaluated in the context of their baseline evolutionary activity, allowing relationships to be distinguished from patterns that may arise purely due to overall differences in turnover rates.

In addition to computing event expectations and eventfulness bins, the preprocessing step performs several additional preprocessing steps to prepare the data for downstream analyses. Gain and loss matrices are checked for consistency and alignment, and user-specified COGs can be excluded, for example when particular gene families are known to behave atypically or are biologically irrelevant. A debug mode further allows restriction to a small subset of genes for rapid testing and development, for example by running the pipeline on only 10% of all COGs to verify that the configuration executes correctly and without errors.

2.5.2 Reconstruction of pairwise contingency tables

In this step, branch-wise gain and loss posterior probabilities are converted into 2×2 contingency tables for every considered gene pair (i, j) . These tables summarize, for each pair, how often the two genes exhibit co-occurring gain (or loss) events along the phylogeny, versus how often only one gene changes state.

Let

$$P^{\text{gain}}, P^{\text{loss}} \in [0, 1]^{B \times G}$$

denote the posterior probability matrices returned by COUNT, where B is the number of branches in the phylogeny and G is the number of gene families. The entry $P_{b,i}^{\text{gain}}$ represents the posterior probability that gene i experienced a gain event on branch b . Each posterior is treated as the expectation of a Bernoulli random variable. Joint expectations are approximated by products of marginal posteriors, implicitly assuming conditional independence between genes given the branch and the fitted gain–loss model. The same definition applies to $P_{b,i}^{\text{loss}}$.

For each gene pair (i, j) , the expected cell counts of the gain contingency table are constructed by summing joint probabilities across all branches:

$$\begin{aligned} a_{ij}^{\text{gain}} &= \sum_{b=1}^B P_{b,i}^{\text{gain}} P_{b,j}^{\text{gain}}, & b_{ij}^{\text{gain}} &= \sum_{b=1}^B P_{b,i}^{\text{gain}} (1 - P_{b,j}^{\text{gain}}), \\ c_{ij}^{\text{gain}} &= \sum_{b=1}^B (1 - P_{b,i}^{\text{gain}}) P_{b,j}^{\text{gain}}, & d_{ij}^{\text{gain}} &= B - (a_{ij}^{\text{gain}} + b_{ij}^{\text{gain}} + c_{ij}^{\text{gain}}). \end{aligned}$$

These quantities represent the expected number of branches on which both genes are gained, only gene i is gained, only gene j is gained, or neither gene is gained, respectively. The final cell d_{ij}^{gain} is computed by subtraction because, for each branch, a gene pair can be in exactly one of four mutually exclusive gain states: both genes gain, only gene i gains, only gene j gains, or neither gene gains. As a result, the four contingency cells must collectively sum to the total number of branches B .

An identical construction is applied to the loss posterior matrix P^{loss} , yielding a corresponding 2×2 contingency table for loss events. The gain and loss tables are treated as two strata in the subsequent Cochran–Mantel–Haenszel (CMH) test, allowing statistical association between gene pairs to be assessed while conditioning on event type.

Efficient chunking and vectorized computations are used throughout this step to keep memory usage and runtime manageable for large numbers of gene pairs. The resulting per-pair record contains the eight contingency cells (gain and loss), the pair indices and corresponding COG identifiers, as well as summary statistics including the expected total number of gain and loss events per gene, the minimum eventfulness of the pair, and the associated bin assignments.

2.5.3 Stratified association testing with the Cochran–Mantel–Haenszel test

The Cochran–Mantel–Haenszel (CMH) framework was originally developed to test associations between binary variables while accounting for natural stratification in the data [59]. It was introduced by Mantel and Haenszel for the analysis of case–control studies and builds directly on Cochran’s earlier work on combining information across strata. The central motivation was to avoid bias arising from simply pooling heterogeneous subpopulations into a single contingency table. In case–control settings, confounding variables often induce strata of very different sizes, and the CMH framework addresses this by weighting each stratum appropriately. As a result, the method provides an unbiased test of independence and an estimate of a common odds ratio that reflects association after adjustment for stratification [34].

The CMH test is well suited to the present analysis because gene gain and gene loss events naturally define two biologically meaningful strata. These event types arise from distinct evolutionary processes and often occur at markedly different frequencies along the phylogeny. Importantly, the CMH framework remains reliable even when one stratum is much rarer than the other, which is a common feature of reconstructed gain and loss histories.

In this setting, the null hypothesis states that two genes evolve independently in both the gain and loss strata. Deviations from this null are assessed using a chi-squared test statistic. The implementation follows the standard formulation of the CMH test as presented in Agresti [1]. For computational efficiency, gene pairs are preprocessed in chunks within a dedicated class to construct the corresponding 2×2 contingency tables. These chunks are then passed to a second class that computes the CMH statistic for each gene pair. Intermediate and final results are written to disk to limit peak memory usage.

For every gene pair, two strata are defined: one for gain events and one for loss events. Each stratum $s \in \{\text{gain}, \text{loss}\}$ contributes a contingency table as described in Section 2.5.2:

$$\begin{pmatrix} a_s & b_s \\ c_s & d_s \end{pmatrix}, \quad n_s = a_s + b_s + c_s + d_s.$$

Here, a_s denotes the expected number of branches on which both genes exhibit an event in stratum s , b_s the number of branches on which only the first gene exhibits the event, c_s the number of branches on which only the second gene exhibits the event, and d_s the number of branches on which neither gene exhibits the event.

For each stratum, the row and column margins are computed as

$$r_{1,s} = a_s + b_s, \quad r_{2,s} = c_s + d_s, \quad c_{1,s} = a_s + c_s, \quad c_{2,s} = b_s + d_s.$$

Under the null hypothesis of independence within stratum s , the expected value and variance of the

co-occurrence cell a_s are given by

$$E[a_s] = \frac{r_{1,s} c_{1,s}}{n_s}, \quad \text{Var}(a_s) = \frac{r_{1,s} r_{2,s} c_{1,s} c_{2,s}}{n_s^2 (n_s - 1)}.$$

The expected value gives the number of co-occurring events that would be expected if the two genes evolved independently, given the observed row and column totals. The variance describes how much this number would be expected to fluctuate by chance under this null hypothesis.

Across strata, the CMH test combines the deviations of the observed co-occurrence counts a_s from their expectations $E[a_s]$. These deviations are aggregated and scaled by the total variance to yield a chi-squared test statistic with one degree of freedom. The single degree of freedom arises because, once the marginal totals of the 2×2 tables are fixed, there is only one independent quantity that can vary, namely the number of co-occurring events. An optional continuity correction C can be applied to stabilize the statistic when counts are small. In this implementation, the value of C (typically $C = 0.5$) is user-controlled:

$$X_{\text{CMH}}^2 = \frac{(|\sum_s (a_s - E[a_s])| - C)^2}{\sum_s \text{Var}(a_s)}.$$

Simulation studies on sparse dichotomous data have shown that a fixed continuity correction of $C = 0.5$ can behave poorly when applied per stratum in meta-analysis settings involving many very small studies. These studies nevertheless confirm that the Mantel–Haenszel framework remains among the most stable estimators in such scenarios [42]. In the present context, where expected counts are typically large and only two biologically defined strata are used, the continuity correction serves a mild protective role and is not expected to introduce the biases observed in more extreme settings.

From the resulting chi-squared statistic, a two-sided p -value is computed as

$$p_{\text{CMH}} = \text{erfc}\left(\sqrt{\frac{X_{\text{CMH}}^2}{2}}\right).$$

Because the CMH test statistic follows a chi-squared distribution with one degree of freedom under the null hypothesis, it can be converted into a p -value by evaluating the probability of observing a value at least as large as X_{CMH}^2 . For one degree of freedom, this tail probability can be expressed in closed form using the complementary error function. The erfc formulation is used here for numerical stability, especially for large test statistics.

The resulting p -value quantifies how unlikely the observed deviation from independence is under the null hypothesis of independent gene evolution. Small p -values indicate strong evidence against independence, whereas large p -values indicate that the observed deviations are consistent with random variation. In practice, a small p -value indicates that the two genes tend to be gained or lost together across branches more often than expected by chance, whereas a large p -value suggests no detectable evolutionary coupling beyond random co-occurrence.

In addition to hypothesis testing, the Mantel–Haenszel common odds ratio is estimated as

$$\hat{\theta}_{\text{MH}} = \frac{\sum_s (a_s d_s / n_s)}{\sum_s (b_s c_s / n_s)}.$$

The numerator summarizes concordant patterns, where both genes change or both remain unchanged,

while the denominator summarizes discordant patterns in which only one gene changes. Their ratio yields a pooled odds ratio across strata, meaning a single summary estimate that combines evidence from both the gain and loss strata. The logarithm of this quantity is used for effect-size summaries, capturing both the strength and direction of association. Values of $\hat{\theta}_{MH} > 1$ indicate that the two genes tend to be gained or lost together more often than expected under independence, whereas $\hat{\theta}_{MH} = 1$ indicates no association. The Mantel–Haenszel common odds ratio provides a stable, single measure of evolutionary coupling across gain and loss events.

2.5.4 Multiple-testing correction

False-discovery-rate (FDR) control provides a principled approach for managing the inflation of false positives (Type I errors) that arises in high-dimensional hypothesis testing. When very large numbers of hypotheses are evaluated simultaneously, as is common in genome-scale analyses, false positives are expected to occur even under the global null. As illustrated by Noble [65], extremely small P -values can arise purely by chance when the search space is large, making naive interpretation of individual test results misleading. Classical family-wise error rate controls such as the Bonferroni correction address this problem by strongly limiting false positives, but at the cost of substantial loss of power, often suppressing biologically meaningful signals [5]. In contrast, FDR-based procedures aim to balance sensitivity and specificity by controlling the expected proportion of false discoveries among rejected hypotheses.

The same considerations apply in the present analysis. Testing all possible gene pairs for coordinated gain–loss events results in a high-dimensional multiple-testing scenario, with each gene pair constituting a separate hypothesis. In this setting, individual Cochran–Mantel–Haenszel (CMH) P -values cannot be interpreted in isolation. To account for the resulting multiplicity, FDR control was applied using the Benjamini–Hochberg step-up procedure [5] across the full set of CMH P -values. Results computed in parallel chunks were first aggregated, and the `fdr_bh` implementation from the `statsmodels` package in python was then applied to the complete vector of test statistics.

2.5.5 Parametric CTMC bootstrap null

To complement the Cochran–Mantel–Haenszel (CMH) test, a parametric continuous-time Markov chain (CTMC) bootstrap null model was implemented. The CMH test assumes that observations within contingency-table strata are independent and exchangeable. In the present setting, however, cell entries are not direct event counts but sums of posterior gain and loss probabilities inferred with COUNT, and therefore already incorporate phylogenetic structure. Ancestral states are shared along the tree, making transition probabilities on neighbouring branches statistically dependent. As a result, inferred transitions tend to occur together across related branches. The contingency-table entries therefore reflect correlated expectations rather than independent observations. As a result, the CMH test can detect apparent associations even when genes evolve independently. This could lead to inflated p -values.

The CTMC null model formalizes the appropriate baseline hypothesis: given the observed gain and loss rates and the phylogenetic tree, how many presence–absence transitions are expected if genes evolve independently? Under this null, gene families evolve according to independent two-state gain/loss processes parameterized by the branch-specific rates estimated with COUNT [15]. Continuous-time Markov chains allow transitions to occur at any point along a branch, with branch lengths defining

the time interval over which gains and losses may occur [39]. Simulated presence–absence histories therefore preserve the tree topology, branch lengths, and turnover-rate heterogeneity, but exclude gene–gene coupling by construction.

Observed fractions of significant associations are then compared with expectations obtained from the CTMC simulations. This diagnostic separates significance caused by phylogenetic structure and rate variation from signals consistent with true co-evolution.

To generate the parametric null, gene histories are simulated under a standard two-state CTMC on the phylogeny. The simulation follows the framework described in Ibe [39] and Felsenstein’s formulation [26]. Each gene can occupy state 0 (absent) or 1 (present). Along branch b , transitions occur with gain rate λ_b for $0 \rightarrow 1$ events and loss rate μ_b for $1 \rightarrow 0$ events. For a branch of length t_b , the total transition rate is

$$s_b = \lambda_b + \mu_b,$$

and the transition probabilities are

$$P(0 \rightarrow 1 \mid t_b) = \frac{\lambda_b}{\lambda_b + \mu_b} (1 - e^{-s_b t_b}), \quad P(1 \rightarrow 0 \mid t_b) = \frac{\mu_b}{\lambda_b + \mu_b} (1 - e^{-s_b t_b}).$$

The complementary probabilities are

$$P(0 \rightarrow 0 \mid t_b) = 1 - P(0 \rightarrow 1 \mid t_b), \quad P(1 \rightarrow 1 \mid t_b) = 1 - P(1 \rightarrow 0 \mid t_b).$$

For each simulated gene, the root state is sampled from the stationary distribution

$$\pi_1 = \frac{\lambda_b}{\lambda_b + \mu_b}.$$

Given the state of a parent node, the state of its child is sampled using the CTMC transition probabilities and branch-specific rates. This procedure is repeated recursively until states have been assigned to all nodes in the tree. The number of branches on which the simulated state differs from that of the parent is then counted, yielding the total number of gain and loss events expected under independent evolution. Repeating this procedure for many simulated genes produces a bootstrap null distribution of transition counts.

Memory usage and computational time are reduced by storing large gain and loss posterior matrices in `float32` format and by using vectorized NumPy operations. Single-precision values require half the memory of `float64`, which is important when handling posterior probabilities for many genes and branches, and this reduced precision is sufficient because the values are bounded between 0 and 1. Vectorized NumPy operations further improve performance by replacing explicit Python loops with efficient low-level implementations.

The resulting CTMC null files (JSON format) store, for each branch-length bin, the bin identifier, the mean simulated transition count, its standard deviation, and the number of bootstrap replicates. The CTMC bootstrap null is configurable. It can be enabled or disabled entirely, and the number of bootstrap replicates can be specified by the user. Increasing the number of replicates yields more stable estimates of the mean and variance of transition counts. The random seed can optionally be fixed to ensure reproducibility. In this study, the number of bootstrap replicates was set to 500.

2.5.6 Computational complexity

Algorithmic time complexity is commonly described using Big-O (Bachmann–Landau) notation. It provides an asymptotic characterization of how computational cost scales with input size. By abstracting away constant factors and hardware-dependent runtimes, Big-O notation captures the dominant scaling behavior of an algorithm [58]. This scaling behavior is a key determinant of practical feasibility on large datasets, which is particularly relevant here.

The computational cost of the ProGeneCoupling pipeline is dominated by a small number of critical steps. The most expensive operation is the construction of pairwise contingency tables based on expected gene co-occurrence. This step scales as $O(B \cdot G^2)$, where G denotes the number of gene families (203) and B denotes the number of branches in the phylogeny (800). For each of the $\binom{G}{2}$ gene-family pairs, expected co-occurrence counts are computed across all branches of the tree.

A second critical step is the correction for multiple testing across all tested gene-family pairs. The Benjamini–Hochberg procedure requires sorting the associated p -values, which results in a computational complexity of $O(n \log n)$. The number of hypotheses is

$$n = \binom{G}{2} = \frac{G(G-1)}{2} = O(G^2).$$

Substituting this expression yields an overall complexity of $O(G^2 \log G)$ for the multiple-testing correction step [40]. Memory usage is primarily driven by the step that aggregate results across all gene-family pairs. In particular, storing and concatenating per-pair statistics scales as $O(G^2)$ both in runtime and peak memory usage. But this aggregation step is performed just once during the pipeline.

When the CTMC-based bootstrap null model is enabled, its computational cost scales as $O(R \cdot S \cdot B)$, where R is the number of bootstrap replicates (500), S is the number of simulated genes per replicate (10,000), and B is the number of branches in the phylogeny (800). For each simulated gene, a continuous-time Markov chain is evaluated along all branches of the tree. Consequently, the total computational effort increases linearly with the number of replicates, simulated genes, and branches.

Chapter 3

Results

3.1 Preprocessing

Before performing any large-scale comparative genomic analysis, it is necessary to identify and examine assemblies that deviate strongly from the remainder of the dataset. Such outliers can disproportionately influence downstream results. They may bias reference-based alignments, inflate estimates of gene gain and loss, distort pangenome frequency distributions and alter phylogenetic branch lengths due to elevated divergence. In datasets dominated by closely related strains, even a small number of misclassified genomes can therefore produce systematic errors [68]. For this reason, the preprocessing stage first focuses on detecting and characterizing atypical assemblies and assessing whether they represent biological variation within the focal species or instead belong to a different taxonomic group.

The dataset consists of 406 genomes assigned to *Vibrio cyclitrophicus*. This marine member of the family *Vibrionaceae* was originally isolated from coastal sediments and is known for degradation of polycyclic aromatic hydrocarbons (PAHs) [33]. The species occupies diverse ecological niches and exhibits considerable genomic diversity, making it a suitable model for studying gene-content evolution in coastal marine bacteria [79].

Two assemblies were first identified as clear outliers on the phylogenetic tree. A subsequent BLASTN search (Section 2.3.4) indicated that these genomes (10N.244.45.D09.fa and 10N.244.45.B06.fa) most likely belong to *Vibrio splendidus*, whereas the remaining 406 genomes are assigned to *Vibrio cyclitrophicus*, consistent with the eggNOG-mapper classification.

An overview of the general assembly statistics across the dataset is shown in Figure 2.1 and summarized numerically in Table 3.1. For each metric, the median, interquartile range (IQR; Q3–Q1), minimum, and maximum values are reported. The interquartile range captures the spread of the central 50% of the data. Genome sizes are tightly clustered around the median and are consistent with expectations for *Vibrio* genomes [55]. However, a small number of extreme outliers reach sizes up to 18.7 Mb, suggesting assembly artifacts or contamination rather than biological variation.

GC content is highly consistent across assemblies. GC content is defined as the fraction of guanine and cytosine nucleotides and is typically conserved within evolutionary lineages over long time scales [77]. The uniform GC composition therefore supports that most genomes belong to the same biological group, as contamination would usually introduce clearly divergent values. Only a few extreme outliers are observed.

Assembly contiguity metrics, including N50 and the number of contigs, show reasonable median

values. The N50 statistic represents the weighted median contig length. Low N50 combined with high contig counts indicates fragmentation and can bias gain–loss inference. However, N50 reflects contiguity rather than correctness and correlates only weakly with assembly accuracy [63]. It was therefore treated as a secondary quality indicator relative to overall genomic consistency measures such as GC content.

Taken together, these metrics indicate that most assemblies are reliable. Assemblies with anomalous properties were identified and are summarized in Table A.12. The core-genome alignment program Parsnp2 automatically excluded five genomes (10N.239.45.B10, 10N.244.45.A07, 10N.244.46.E10, 10N.247.45.G07, 10N.247.46.A02). These genomes were not included in downstream analyses, although some additional flagged assemblies remained and may warrant removal in future analyses.

All remaining genomes, each represented as a set of contigs, were aligned to construct a core-genome alignment. The alignment was performed using 10N.244.45.B01 as the reference genome, which served as a stable coordinate system for identifying homologous regions across assemblies. This reference genome has a total length of 5,153,042 bp and consists of 23 contigs. Run statistics for the Parsnp2 program are summarized in Table 3.2.

Metric	Median	IQR (Q3–Q1)	Min	Max	Range
size_bp	5145786	236185	4505851	18716077	14210226
gc_content	43.53	0.10	41.88	51.90	10.02
num_contigs	51	48	16	3264	3248
n50	265684	190809	2740	1167671	1164931

Table 3.1: Genome assembly statistics across 406 *Vibrio* genomes.

Table 3.2: Summary run statistics of the core genome alignment and tree reconstruction.

Metric	Value
Skipped genomes	5
Aligned genomes	401
Intersected core size	2,178,639 bp
Intersected LCBs	2,913
Mean covered bp	3.82 Mb
Runtime	~110 min
Memory usage	~10.6 GB
Tree reconstruction time	~58 min

The tool Pyrodigal was used to predict open reading frames (ORFs) in all assemblies. The model was trained using different numbers of genomes, and the performance of the respective training configurations is summarized in Table 3.3. For each run, ORFs were predicted across all contigs in all genomes, with a runtime of approximately 60 minutes per run. All downstream analyses were then performed using the predictions obtained with the training size T5. The resulting ORF confidence was high.

All predicted ORFs were translated into protein sequences and written to a single combined FASTA file. Each FASTA entry corresponds to one ORF, and the header encodes the genome identifier, contig name, sequence length, start and end coordinates, strand orientation, and an internal flag (e.g. >10N.232.45.E11#contig3#249#301807#302544#-1#00).

The combined protein FASTA file was used as input for FastCOG (2.4.2) to cluster predicted proteins

into homologous gene families. In total, 63,904 distinct COGs were identified. The first quartile, median, and third quartile of the number of genomes per COG were approximately one, meaning at least 75% of gene families occur in only a single genome. The mean number of genomes per COG (3.92) was much higher, reflecting a long right-tailed distribution driven by a small subset of widely distributed genes (Fig. 2.2). The gene frequency spectrum shows a highly open pangenome in which most gene families are strain-specific. Only a small conserved core is shared across the population, consistent with frequent horizontal gene transfer and ecological diversification. This pattern is consistent with theoretical expectations for open bacterial pangenomes and with previous observations [84, 79].

To reduce noise from both extremely rare and nearly ubiquitous gene families, filtering based on COG presence counts was applied. Rows with a sum below 5.0 accounted for 54,663 COGs (85.54%), and rows with a sum greater than 203.0 comprised 4,140 COGs (6.48%). In total, 58,803 of the 63,904 COGs (92.02%) were removed, leaving 5,107 COGs for subsequent analyses. The lower cutoff is consistent with previous work showing that similar thresholds retain biologically meaningful associations [11]. Increasing the number of retained COGs caused prohibitively long runtimes during maximum-likelihood rate inference in COUNT. The upper threshold was therefore chosen as the set of genes present in at most half of the genomes. In a phylogenetic tree this corresponds to gene families that can plausibly switch state along branches, rather than remaining almost always present. Genes present in nearly all genomes would require very few transitions and therefore provide little information, while genes around the midpoint frequency allow gains and losses to be distributed across the tree and can be statistically inferred more reliably. This threshold was chosen based on this theoretical consideration and not from a specific literature reference.

Because most genes undergo only few evolutionary events, most gene pairs fall into low eventfulness categories (Figure 3.2). Genes present in most genomes are nearly static, whereas rare genes provide insufficient information. Consequently, many pairs cannot be identified as associated simply due to insufficient evolutionary signal rather than absence of biological coupling.

For all COGs in the dataset, eggNOG-mapper produced fully consistent functional annotations. All genes within a given COG received the same annotation (100% agreement). This is expected because both the clustering and the annotation rely on sequence homology. Proteins grouped into the same COG share high sequence similarity and typically map to the same orthologous group in eggNOG, resulting in identical transferred annotations. This consistency also indicates that the clustering procedure reliably grouped homologous proteins.

Training size	Total ORFs predicted	Mean ORF confidence (%)
T1	1,865,019	98.89
T5	1,862,725	98.91
T20	1,865,406	98.88

Table 3.3: Results of Pyrodigal ORF prediction using different training set sizes (T1, T5, T20).

3.2 Phylogenetic reconstruction

The phylogenetic reconstruction followed the workflow described in Section 2.3. Briefly, the initial tree inferred with Parsnp2 and RAXML was used as input for ClonalFrameML to identify regions affected

by homologous recombination. This first ClonalFrameML run adjusted branch lengths to account for recombination but did not modify the underlying topology.

The phylogeny was constructed with the explicit aim of minimizing signals driven by recombination (Figure 3.1). A reliable clonal tree is required as input for both COUNT and the Python module ProGeneCoupling. These methods model gene gain and loss events along individual branches from the root to the leaves. Mapping events onto specific branches allows lineage-specific coevolution to be distinguished from simple co-occurrence patterns caused by shared ancestry.

To obtain a recombination-aware topology, IQ-TREE3 was subsequently run on the alignment with recombined regions masked. The updated topology was then supplied to ClonalFrameML, enabling the detection of additional recombination events. This procedure was iterated three times. ClonalFrameML was rerun, the alignment was updated accordingly, and IQ-TREE3 was executed again on the refined masked alignment. Each ClonalFrameML run was performed on a single CPU core and required approximately 10 hours, whereas IQ-TREE3 was run using 8 CPU cores and required roughly 24 hours per iteration.

Reducing the impact of recombination is essential because recombination can generate spurious signals of gene co-gain or co-loss. Genes transferred together on mobile genetic elements may appear repeatedly gained or lost together even if they are not functionally coupled. Masking recombination therefore prevents false inference of coordinated evolution.

Across the dataset, recombination was detected in 182 genomes (Table A.4). In most cases only a very small proportion of the genome was affected, with a median recombination burden of 0.195%. For example, isolate 10N.232.45.E11 (4,978,280 bp) contained eight recombination tracts with a combined length of 9,355 bp, corresponding to 0.188% of the genome. In contrast, isolate 10N.232.45.E12 (5,076,336 bp) carried 46 shorter tracts totaling 12,756 bp, yielding 0.251%. The reported percentage represents the summed length of all inferred recombined segments divided by the genome size. The tract sizes fall within the typical range of a few hundred to a few thousand base pairs described by Didelot and Maiden [19], indicating that the inferred events are consistent with empirically observed bacterial recombination imports.

Recombination signals were predominantly observed along longer branches, for example in the lower part of the phylogeny shown in Figure 3.1. Longer branches represent greater divergence and longer evolutionary time spans, which increase both the opportunity for recombination and the statistical power to detect it. In contrast, closely related genomes connected by very short branches often showed no detected recombination events. This likely reflects either reduced recombination among very similar strains or limited power to distinguish recombination from point mutations at extremely low divergence.

In the coastal *Vibrio cyclitrophicus* populations studied by Shapiro et al. [79], the most closely related isolates differed by only 66 single nucleotide substitutions across a 3.54 Mb core genome, corresponding to approximately 2×10^{-5} substitutions per site and indicating a very recent common ancestor. Despite this near identity, each genome contained about 4,500 bp absent from its closest relative. The fraction of recently replaced sequence is

$$\frac{4500}{3.54 \times 10^6} \approx 1.3 \times 10^{-3} \approx 0.13\%.$$

Thus even extremely recent recombination events affected only a small portion of the genome. In the present dataset, the median recombined fraction (0.195%) is of the same order of magnitude as the $\sim 0.13\%$

reported by Shapiro et al. [79]. This supports the view that bacterial genomes typically accumulate numerous short imports rather than large-scale replacements, consistent with previous observations that closely related isolates rarely exceed 1% recently replaced sequence [20].

A large-scale analysis of 176 bacterial species from the human microbiome [91] similarly inferred that homologous recombination replaces only a small portion of the genome relative to mutation, with estimated intensities ranging from approximately 10^{-5} to 10^{-3} genome fractions per mutation event and a median near 2×10^{-4} . Although these estimates are not directly comparable to tract-based ClonalFrameML measurements, they describe the same biological baseline. Recombination occurs frequently over evolutionary time but affects limited genomic segments at any given divergence level. The recombination burden observed here therefore falls within the range reported across diverse bacterial taxa and supports the conclusion that bacterial evolution proceeds through many small DNA imports rather than genome-wide replacement events.

Together, these observations indicate that recombination in the dataset is detectable but limited in magnitude. Consequently, successive rounds of recombination masking are expected to refine branch lengths and topology only until the remaining phylogenetic structure reflects vertical inheritance. To verify that this point was reached, it was quantified how the tree changed across iterations.

Table 3.4 summarizes how the Robinson–Foulds distance, branch score distance (BSD), and mean branch-length differences changed across successive refinement rounds. Branch lengths and topology stabilized after several iterations, indicating that further application of ClonalFrameML did not lead to additional improvements. The final clonal phylogeny corresponds to the Newick tree produced by the last IQ-TREE3 run (Figure 3.1).

Although the genomes were collected within a relatively short time span, they still contain sufficient genetic variation for reliable phylogenetic inference. This variation most likely represents standing genetic diversity that accumulated long before the sampling period rather than mutations that arose during it. Natural bacterial populations consist of multiple long-coexisting lineages that diverged over extended evolutionary timescales and are simply observed simultaneously when sampled [12]. Consequently, genomes collected at similar time points do not represent the same evolutionary stage.

Comparison	RF	BSD	Branch length diff
Parsnp2 Tree → ClfML Tree 1	1	0.732	-9.934
IQTree Tree 1 → IQTree Tree 2	224	0.001	-0.0004
IQTree Tree 2 → IQTree Tree 3	140	0.001	0
IQTree Tree 3 → IQTree Tree 4	109	0.01	0.134

Table 3.4: Comparison of phylogenetic trees generated during the iterative recombination-aware reconstruction. “ClfML” denotes ClonalFrameML. Each pairwise comparison shows the change between an IQ-TREE3 topology and the new IQ-TREE3 topology after rerunning ClonalFrameML and masking recombination.

3.3 Maximum-Likelihood inference

The maximum-likelihood (ML) estimation performed by COUNT constituted the most computationally demanding step of the analysis. For a dataset with 203 COGs and 800 branches, a single optimization

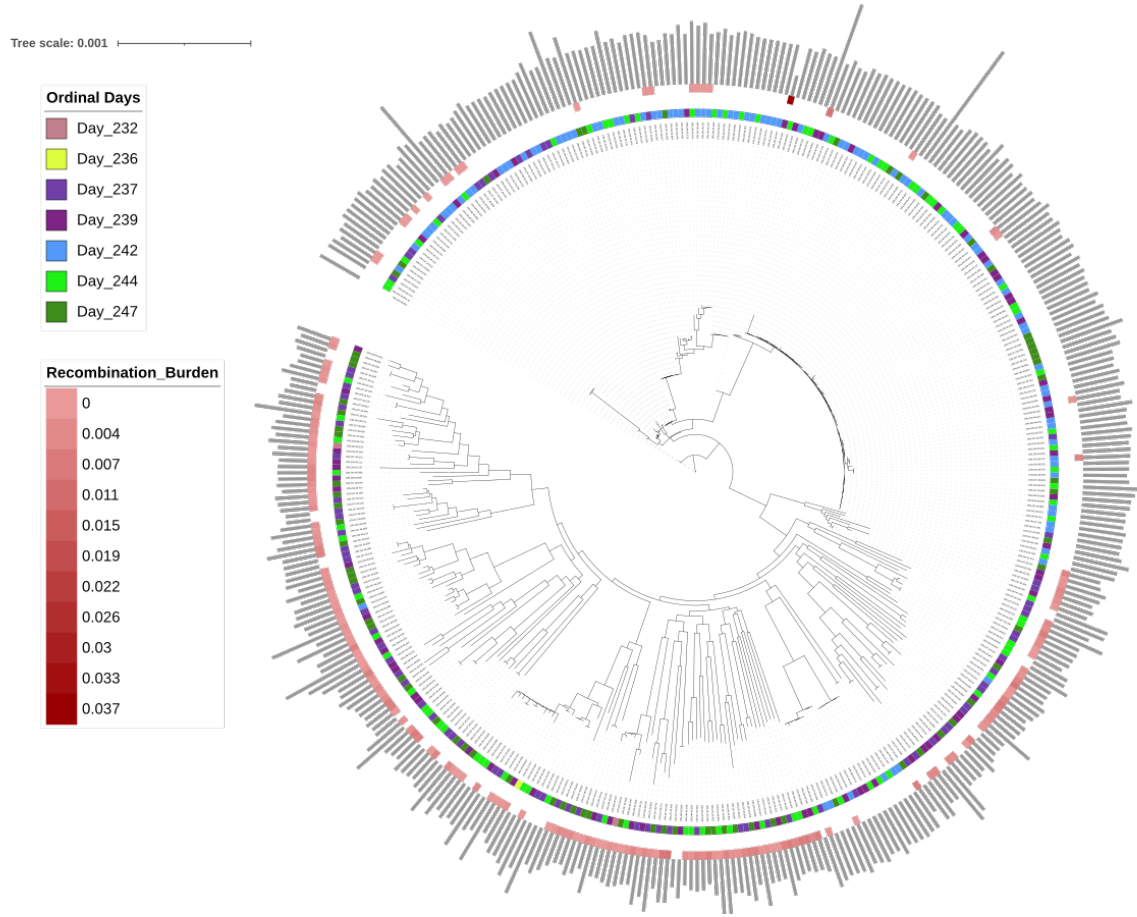


Figure 3.1: Final clonal phylogeny reconstructed from 401 *Vibrio cyclitrophicus* genomes. The tree highlights the sampling day for each genome. It marks those genomes in which recombination segments were detected and removed across all iterations. The outer ring of bars represents the genome sizes.

round required approximately three days of wall-clock time. After two rounds, parameter updates became very small and further improvement was marginal. The workflow was therefore configured to stop after two rounds rather than waiting for the formal convergence criterion. The relative change in log-likelihood after the second round was only 0.28%, indicating an effectively near-converged solution and suggesting that additional iterations would have had negligible influence on downstream inference [14].

During execution, COUNT required approximately 23 GB of RAM. Table 3.5 summarizes the ML parameter estimates from optimization rounds 1 and 2. The root state distribution represents the inferred probability that a gene family was present at the root of the phylogeny. Event-rate parameters quantify the relative frequencies of duplication, loss, and horizontal gene transfer. Branch-length parameters define the evolutionary time scale after rate normalization. The log-likelihood measures how well the model explains the observed gene presence–absence data and tracks improvement across optimization rounds, thus serving as the primary criterion for assessing model fit [14]. Each gene family contributes only a small fraction to the total score.

To capture differences in evolutionary rates across lineages, the optimization divided branches into three discrete gamma-distributed rate categories. This captures major variation in evolutionary dynamics. Additional categories would allow a finer-grained representation of rate heterogeneity but would substantially increase runtime because likelihood calculations must be performed separately for

each category across all branches and gene families.

After normalization, the final model had a mean branch length of 0.0406, a duplication rate of 1.02×10^{-5} , and a loss rate fixed to 1. By model design, the loss rate is uniform across branches. COUNT fixes the loss rate to one because only relative event frequencies are identifiable from presence–absence data [14]. Therefore, loss serves as the reference rate that defines the time scale of the model, and all other parameters should be interpreted as occurring relative to the rate of gene loss.

The duplication rate was allowed to vary during optimization but converged to nearly identical values across the tree, suggesting little evidence for lineage-specific duplication differences. In contrast, horizontal gene transfer was strongly heterogeneous, with inferred transfer rates ranging from 1.01×10^{-5} to 2.74 across branches. Such variability is consistent with ecological and evolutionary dependence of horizontal transfer processes [66, 45]. Comparable likelihood-based studies of bacterial gene content evolution likewise infer high gene gain rates driven predominantly by horizontal transfer [45, 47].

The estimated root prior decreased from 0.10 to 0.055, implying that fewer gene families were inferred to be present in the common ancestor. To match the gene content of present-day genomes, the model inferred more gene gains along the branches, indicating that many genes were acquired later rather than inherited continuously from the ancestor.

Overall, the estimated rates are internally consistent and biologically plausible, indicating an evolutionary pattern dominated by horizontal gene acquisition, while gene duplication contributes comparatively little.

Table 3.5: Summary of COUNT maximum-likelihood (ML) optimization results.

Parameter	Optimization round 1	Optimization round 2
<i>Root state distribution</i>		
Root presence probability (π_0)	0.0551	0.0564
Root prior (initial)	0.10	0.0551
<i>Event rate parameters</i>		
Duplication rate (λ_{dup})	0.5360	0.5517
Loss scaling (optimization coordinate)	0.4470	0.4879
Transfer rate (λ_{hgt})	heterogeneous	heterogeneous
<i>Branch length parameters</i>		
Mean branch length (pre-update)	0.03654	0.04060
Mean branch length (post-update)	0.04060	0.04128
Relative change (%)	+61.33	+1.68
<i>Likelihood</i>		
Log-likelihood (start of round)	225,938.47	225,478.01
Log-likelihood (end of round)	225,478.01	224,837.37
Log-likelihood improvement	28,849.91	640.64
Relative change (%)	-12.8	-0.28

After ML estimation, COUNT was executed in posterior mode to infer branch-specific posterior probabilities of evolutionary events. This step required approximately 1.5 days of runtime and about 20 GB of memory. The output provides, for each branch and gene family (COG), posterior probabilities for gain, loss, expansion, and reduction events. After parsing these results into separate matrices, they served as input for the coupled gain–loss inference. Because these posterior reconstructions depend directly on the previously estimated CTMC rate parameters, near-convergence of the ML optimization

is essential for reliable inference [36].

Interpreting inferred gain events requires care. They should not be confused with the recombination events removed during tree reconstruction. ClonalFrameML filters recombination from core-genome sequences to infer a tree representing vertical ancestry, whereas COUNT analyzes gene presence–absence patterns on this fixed tree. When a gene occurs in distantly related lineages but is absent from intermediate ones, vertical inheritance alone cannot explain the pattern. The likelihood model therefore introduces gene acquisition events along branches to account for the observed distribution. Accordingly, the transfer events inferred by COUNT represent gene content evolution rather than residual recombination signals. Even after removing recombination from the phylogeny, accessory genes are still unevenly distributed, so the model requires many gene gains along the tree.

3.4 Detection of coupled gene gain and loss events

The ProGeneCoupling pipeline was run successfully and completed in approximately 4 hours using the specified parameter settings. With the current implementation and dataset, a memory allocation of about 7 GB is required. After preprocessing and filtering, a total of 5,107 COGs were retained and grouped into 10 eventfulness bins (2.5.1). The quantile-based binning was performed using the per-gene eventfulness measure $\min(E_{\text{gain}}, E_{\text{loss}})$, spanning a range from $\text{edges_min} = 2.03 \times 10^{-4}$ to $\text{edges_max} = 40.12$. A value close to edges_max does not correspond to an observed count of discrete events but reflects an expected number of turnover events accumulated across the entire phylogeny for that gene family. Accordingly, gene families that rarely change state across the tree fall into the lowest bin, whereas highly dynamic genes with frequent gains and losses are placed in the highest bin.

For the downstream coupling analysis, all $\binom{5,107}{2} = 13,038,171$ unique COG (gene) pairs were evaluated. The distribution of gene pairs across eventfulness bins is shown in Figure 3.2. Bin ID 0 contains the largest number of pairs (2,474,265), corresponding to pairs of low-eventfulness genes. In contrast, Bin ID 9 contains the fewest pairs (130,305), representing pairs of highly eventful genes.

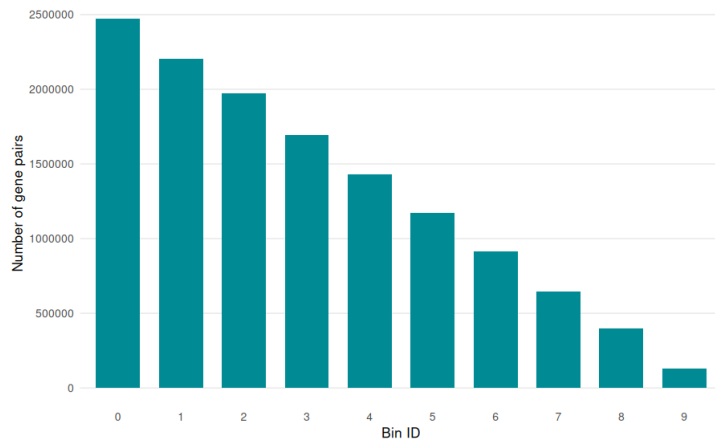
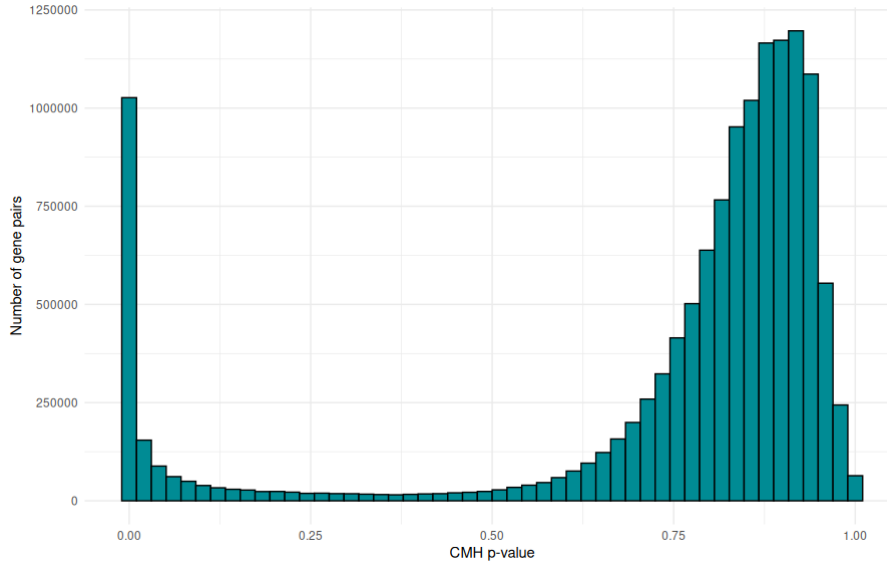


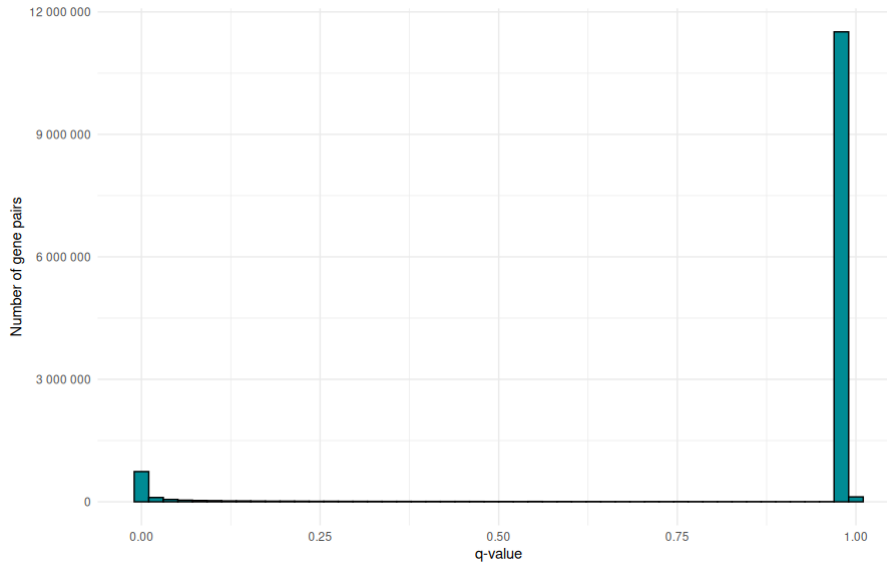
Figure 3.2: Distribution of COG gene pairs across bins. Each bin ID represents a quantile of minimum per-gene eventfulness. Bars indicate the total number of COG pairs assigned to each bin.

Figure 3.3a shows the distribution of raw CMH p -values across all tested COG pairs. The following fractions can be observed: 6.03% of all pairs have $p < 0.001$, 7.85% have $p < 0.01$, and 9.71%

have $p < 0.05$. After Benjamini-Hochberg FDR correction, the distribution of q -values is shown in Figure 3.3b. At an threshold of 0.05, 6.92% of all tested gene pairs remain significant, while 7.60% are significant at $q < 0.10$ and 5.67% at the more stringent threshold $q < 0.01$. The majority of pairs are assigned high q -values, with 89.3% of all pairs having $q > 0.95$.



(a) Distribution of raw CMH p -values.



(b) Distribution of Benjamini-Hochberg FDR-corrected q -values.

Figure 3.3: Distributions of raw and FDR-corrected significance values across all tested COG pairs.

Figure 3.4 illustrates the relationship between statistical significance and effect size for all tested COG pairs. Each point represents a single gene-gene pair. The x -axis shows the estimated Mantel-Haenszel log-odds ratio, $\log(\text{OR}_{\text{MH}})$, while the y -axis reports statistical significance as $-\log_{10}(q)$. The vertical dotted line at $\log(2)$ indicates a two-fold increase in the odds of co-occurring gain or loss events and serves as a reference for biologically meaningful effect sizes. The horizontal dotted line at $-\log_{10}(0.05)$ marks the 5% FDR significance threshold. The majority of non-significant pairs cluster tightly below this cutoff, forming a dense horizontal band.

Gene pairs located in the upper-right quadrant of the plot therefore exhibit both strong statistical support and large positive effect sizes. Out of the 13,038,171 tested COG pairs, 902,633 pairs (6.9%) are significant at $q < 0.05$. In total, 12,433,694 pairs show an absolute log-odds ratio of at least $\log(2)$.

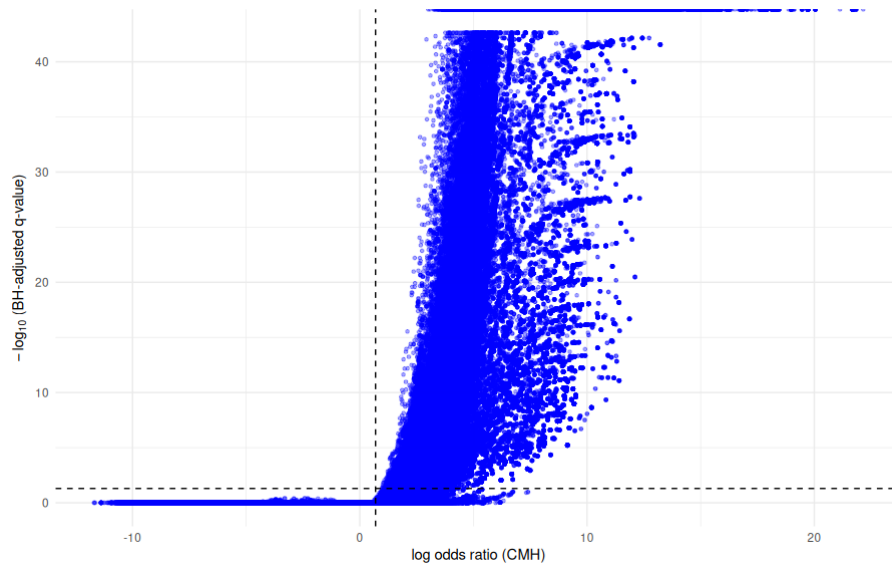


Figure 3.4: Effect size versus statistical significance for all tested COG pairs. Each point represents one gene pair. Dotted lines indicate thresholds at $\log(2)$ (two-fold odds ratio) and $q = 0.05$.

The relationship between gene-pair eventfulness and estimated effect sizes is summarized by a boxplot of Mantel–Haenszel log-odds ratios, stratified by eventfulness bin (Figure 3.5). The plot addresses how effect sizes vary as a function of gene-pair eventfulness. Each box summarizes the distribution of effect sizes within a bin, including the median, the interquartile range (25–75%), and the overall spread across all gene pairs.

The dashed horizontal line indicates the null effect $\log OR = 0$, corresponding to independence between the evolutionary histories of the two genes. Values above zero indicate positive associations: gain or loss events occur on the same phylogenetic branches more often than expected under independence, implying coupled evolution of the gene pair.

Values below zero indicate negative associations. Evolutionary changes in one gene tend to occur on branches where the other gene remains unchanged. The statistic measures co-occurrence of changes rather than their direction. This means that negative values do not show that one gene is gained while the other is lost. Instead, they show that the two genes rarely change on the same branches, suggesting that their evolutionary changes tend to occur separately rather than together.

As an illustration, Bin ID 0 contains 2,474,265 gene pairs with a median log-odds ratio of -4.92 (interquartile range: $[-5.87, -3.64]$). Bin ID 8, comprising 397,573 pairs, shows a median log-odds ratio of -1.00 . In contrast, Bin ID 9 contains 130,305 pairs and has a median log-odds ratio of 0.37 (interquartile range: $[-1.53, 1.17]$).

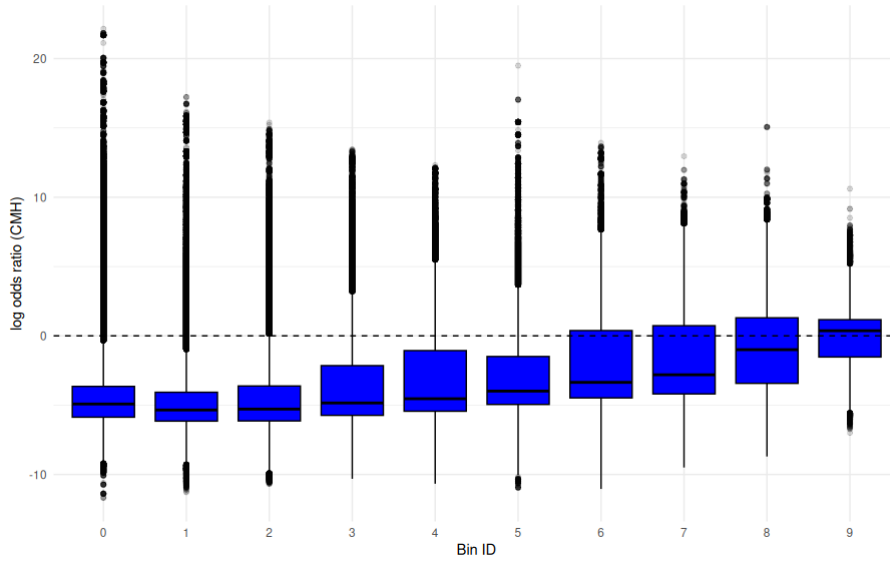


Figure 3.5: Median CMH effect size (log odds ratio) per bin.

While the CMH effect size quantifies whether evolutionary changes co-occur more or less often than expected, it does not indicate which type of event drives the association. A positive odds ratio may arise from repeated joint gains, repeated joint losses, or a mixture of both. To disentangle these, the contingency tables were decomposed into their expected co-gain and co-loss components.

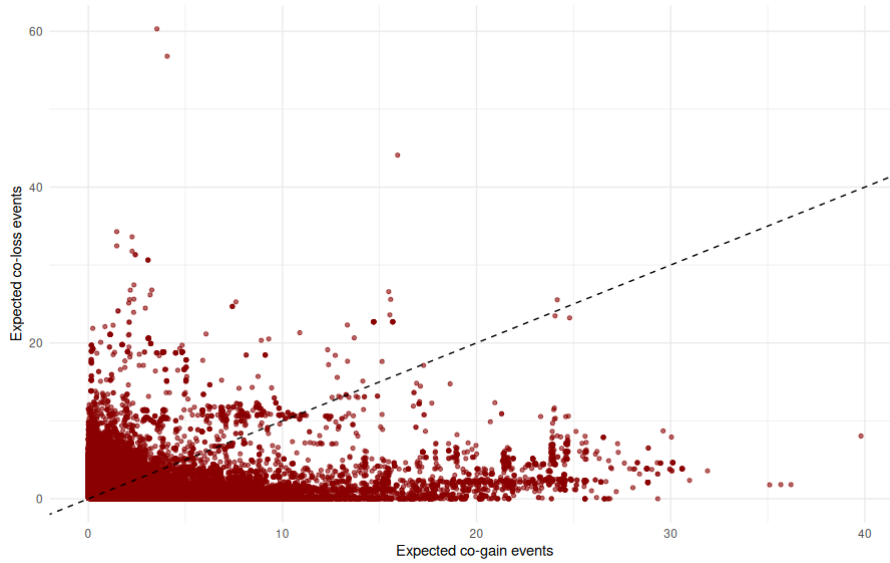


Figure 3.6: Expected co-gain versus co-loss events for significantly coupled gene pairs.

A scatter plot therefore compares, for each significant gene pair ($q_{\text{CMH}} < 0.05$), the expected number of branches with joint gain events (x-axis) to the expected number of branches with joint loss events (y-axis) (Figure 3.6). Each point is derived directly from the corresponding contingency table and represents the expected counts of co-gain and co-loss events across the phylogeny.

The plot assesses whether statistically significant coupling signals are predominantly driven by gain events, loss events, or a combination of both. The dashed diagonal line ($y = x$) indicates equal contributions of co-gain and co-loss. Points below this line correspond to gene pairs dominated by

co-gain events, whereas points above the line indicate dominance of co-loss events.

Across all significant gene pairs, the mean expected number of co-gain events is 1.72, compared to a mean of 0.187 co-loss events. The median expected counts are 1.00 for co-gain and 1.42×10^{-4} for co-loss. Overall, 94.2% of significant gene pairs are gain-dominated.

3.5 Comparison to CTMC null

The continuous-time Markov chain (CTMC) module (see Section 2.5.5) was used to compute a parametric null distribution for gene gain and loss dynamics along the phylogeny. The model addresses a central question: given the observed gain and loss rates and the phylogenetic tree, how many presence–absence transitions are expected if genes evolve independently? The CTMC null excludes gene–gene coupling and provides a baseline expectation directly comparable to the independent gene family models implemented in COUNT [15]. Such explicit null models are necessary because many microbial genes follow largely independent evolutionary trajectories, and associations must therefore be evaluated relative to background evolutionary dynamics rather than absolute event frequencies [2]. In contrast, coupling is assessed using the CMH test, which evaluates co-occurrence patterns while controlling for tree topology, gain and loss rates, and differences in gene eventfulness.

Simulations were performed for 500 independent replicates. The resulting bin-wise transition statistics are summarized in Figure 3.7 as a line plot with error bars. The x-axis denotes the eventfulness bin identifier, while the y-axis shows the mean number of simulated state transitions per gene. Each point represents the bin-wise mean of the total number of presence–absence state changes observed across all bootstrap simulations, and vertical error bars indicate one standard deviation around the mean. So, each dot gives the typical number of changes expected for genes in that bin, while the bars show how consistent that number is across repeated simulations.

The estimated mean number of transitions increases monotonically but slowly, across bins, consistent with the expectation that more eventful genes undergo more evolutionary turnover. In Bin ID 0, the mean transition count is 0.009 with a standard deviation of 0.146. This increases to 0.015 ± 0.187 in Bin ID 1 and 0.025 ± 0.2387 in Bin ID 2. From Bin ID 2 to Bin ID 6 the curve is approximately flat, indicating that transition activity remains nearly constant across these intermediate bins. The highest-eventfulness bin (Bin ID 9) exhibits the greatest transition activity, with a mean of 0.145 and a standard deviation of 0.581. Even in this highest bin the expected number of transitions per gene remains well below one, indicating that gene content evolution is sparse and dominated by few events per lineage, consistent with previous gene gain and loss models [15]. The variability of the null distribution also changes across bins. The standard deviation increases from low bins to intermediate bins. This is reflecting increasing heterogeneity in evolutionary trajectories as genes become more eventful. In the highest bin the variance becomes much larger, which is expected because when many gain and loss opportunities exist along the tree, the outcome depends more strongly on chance.

Overall, simulated presence–absence transitions remain rare across all bins in the custom CTMC implementation. Two methodological aspects could contribute to this. First, the statistic counts only net parent–child state differences rather than the total number of gain and loss events along a branch. Multiple events occurring on the same branch can therefore cancel out and remain unobserved in the final state comparison. Second, the CTMC change probability on a branch is small when $(\lambda + \mu)t$ is

small, where λ and μ are the gain and loss rates and t is the branch length. Short branches or low gain rates therefore produce few observable state flips even under independent evolution.

However, these factors also imply that the null may underestimate the true amount of evolutionary activity. Hidden multiple events, root state assumptions, or mismatches between estimated rates and branch-length scaling could further suppress transition counts.

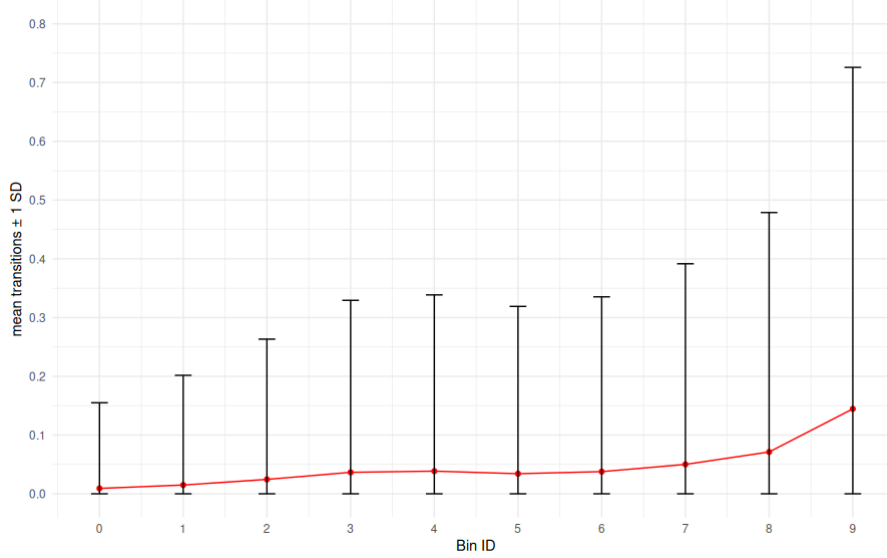


Figure 3.7: Mean number of simulated state transitions per eventfulness bin under the CTMC bootstrap null model. Points indicate the bin-wise mean of total transitions across bootstrap simulations, with error bars showing one standard deviation.

These results define the baseline expectation under independent evolution. Bins with more eventful genes should show slightly more detectable associations simply because more transitions occur. Characters that change state rarely contain little information about correlation, whereas frequently changing characters provide more opportunities for coincidental co-occurrence. This increases detectable associations even under independence [26]. Therefore, a monotonic but gradual increase in significant gene pairs is expected even without biological coupling.

To test whether the empirical data follow this purely statistical expectation, the CTMC-predicted transition activity was directly compared with the observed rate of significant associations across bins. The relationship between CTMC null expectations and empirical results is summarized in Figure 3.8. Each blue point corresponds to a single eventfulness bin. The x-axis shows the mean number of state transitions per gene predicted by the CTMC bootstrap null model for that bin, while the y-axis reports the fraction of gene pairs within the same bin that are significant in the real data, defined as the proportion of tested pairs with FDR-corrected q -values below the chosen threshold. Point sizes are proportional to the number of gene pairs tested in each bin, such that larger points correspond to bins containing more gene pairs. Across bins, the number of tested gene pairs decreases from 2.47 million in Bin ID 0 to 0.13 million in Bin ID 9. A smooth curve summarizes the overall relationship between CTMC-predicted transition activity and observed significance rates.

Across bins, higher mean transition counts are accompanied by higher observed fractions of significant gene pairs, as expected under the null. In the lowest eventfulness bins, mean CTMC-predicted transition counts increase gradually from 0.009 in Bin ID 0 to 0.015 in Bin ID 1 and 0.025 in Bin ID 2,

with corresponding fractions of significant gene pairs of 4.7%, 5.3%, and 7.3%, respectively. Intermediate bins show continued increases in mean transition counts, accompanied by significance rates between 6.5% and 10.5%. Bin ID 8 exhibits a mean transition count of 0.071 and a significance rate of 10.8%. The highest eventfulness bin (Bin ID 9) has the largest mean transition count (0.145) and the highest fraction of significant pairs (11.4%).

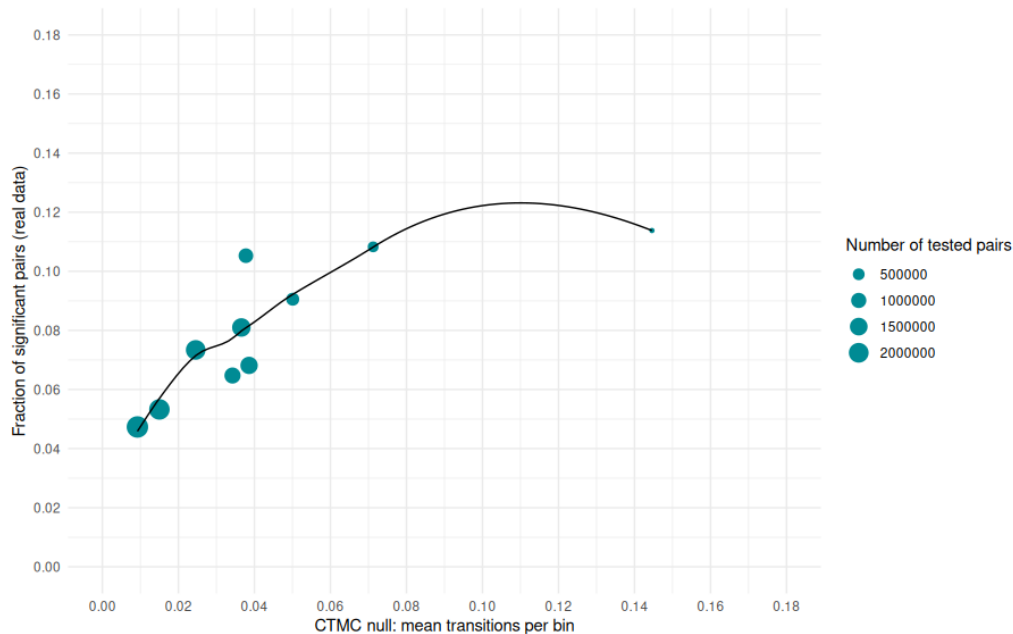


Figure 3.8: Relationship between CTMC null expectations and empirical significance rates across eventfulness bins.

However, the increase does not follow the smooth pattern expected from independent evolution alone [26]. Early bins show a relatively strong rise in significance despite only small increases in transition counts, whereas later bins exhibit only minor changes despite substantially higher expected transition numbers. For example, doubling the expected number of transitions from Bin ID 8 (0.071) to Bin ID 9 (0.145) produces only a marginal increase in significant pairs (from 10.8% to 11.4%). In addition, bins with similar null expectations differ noticeably in observed significance. Thus, transition frequency explains part of the signal but not its magnitude or structure. Low-event genes likely lack statistical information, so even weak dependencies remain undetectable [18]. This produces a steep initial rise once occasional turnover occurs. At higher eventfulness, additional gains and losses increasingly obscure consistent co-occurrence patterns, leading to a saturation of detectable associations. Moreover, individual bins probably mix functionally unrelated mobile genes with tightly linked pathway components, so turnover rate alone cannot predict the probability of association [83][13]. For example, transposases or phage defense genes may be gained and lost often, but mostly independently in different lineages [66][27]. They therefore produce many transitions without showing consistent pairwise associations. In contrast, genes that belong to the same metabolic pathway or protein complex may change state less frequently, yet they tend to appear or disappear together because incomplete pathways are disadvantageous [72]. When both gene types fall into the same eventfulness bin, they have similar numbers of transitions but very different chances of co-occurring, which increases variation beyond the null expectation.

Taken together, the CTMC null provides the number of associations expected if genes evolved

independently. Under this baseline, gene turnover is sparse and increases only gradually with eventfulness, and no bin shows anomalously high transition counts. This supports the CTMC null as a stable reference and explains why some increase in detectable associations is expected simply from increasing information content. However, the empirical significance rates vary across bins in a way not predicted by transition frequency alone, indicating that either unmodeled heterogeneity or genuine evolutionary dependence contributes to the observed signal. Here, unmodeled heterogeneity denotes hidden lineage or genomic factors that induce correlated gain and loss patterns among independent genes. This could generate apparent associations. CMH-significant pairs should therefore be interpreted as probabilistic evidence for coupling. Significance reflects a level of confidence rather than definitive proof. Strong and consistently enriched associations are plausible candidates for linked evolutionary histories, whereas marginal signals are more likely influenced by residual model mismatch and should be treated cautiously.

3.6 Significant COG association network

All significant couplings ($q_{\text{CMH}} \leq 0.05$) were interpreted as edges in an undirected network. Nodes correspond to COGs, and an edge indicates a statistically significant coupled gain-loss signal between two COGs. The results produced by the ProGeneCoupling pipeline were imported into R from the file `results.parquet`. From this table, only statistically significant associations were retained. For each remaining pair, gene identifiers, effect sizes, and bin identifiers were extracted.

The significant pairs were ranked by decreasing effect size, and only the top 10^4 associations were retained, as larger networks were impractical to load into Cytoscape. Functional annotations were added by joining an external gene annotation table. The resulting network decomposed into 170 disconnected subnetworks, ranging from large, densely connected components to small components consisting of only a single edge.

Among the multiple connected components present in the resulting coupling network, four clusters of different sizes and bin compositions were selected for further inspection and are shown in Figure 3.9. An overview of the nodes and edges for cluster (b) is provided in Table A.8. Corresponding summaries for clusters (c) and (d) are given in Tables A.6 and A.7, respectively. The table for cluster (a) is omitted due to its size.

In addition, the degree of each COG was computed. The degree is defined as the total number of significant couplings in which a given COG participates. No further filtering by effect size, bin membership, or event type was applied at this stage. A ranked table of COGs by degree was constructed, containing the following columns:

- COG: identifier of the gene family.
- n_connections: number of significant gene-gene couplings that involve the COG (degree).
- top_description: short functional annotation of the gene family (from eggNOG).
- KEGG_ko: associated KEGG orthology identifier, if available.
- KEGG_Pathway: corresponding KEGG pathway annotation, if available.

Table A.5 lists the 50 COGs with the highest number of connections among all significant couplings. These highly connected nodes represent gene families that repeatedly co-occur.

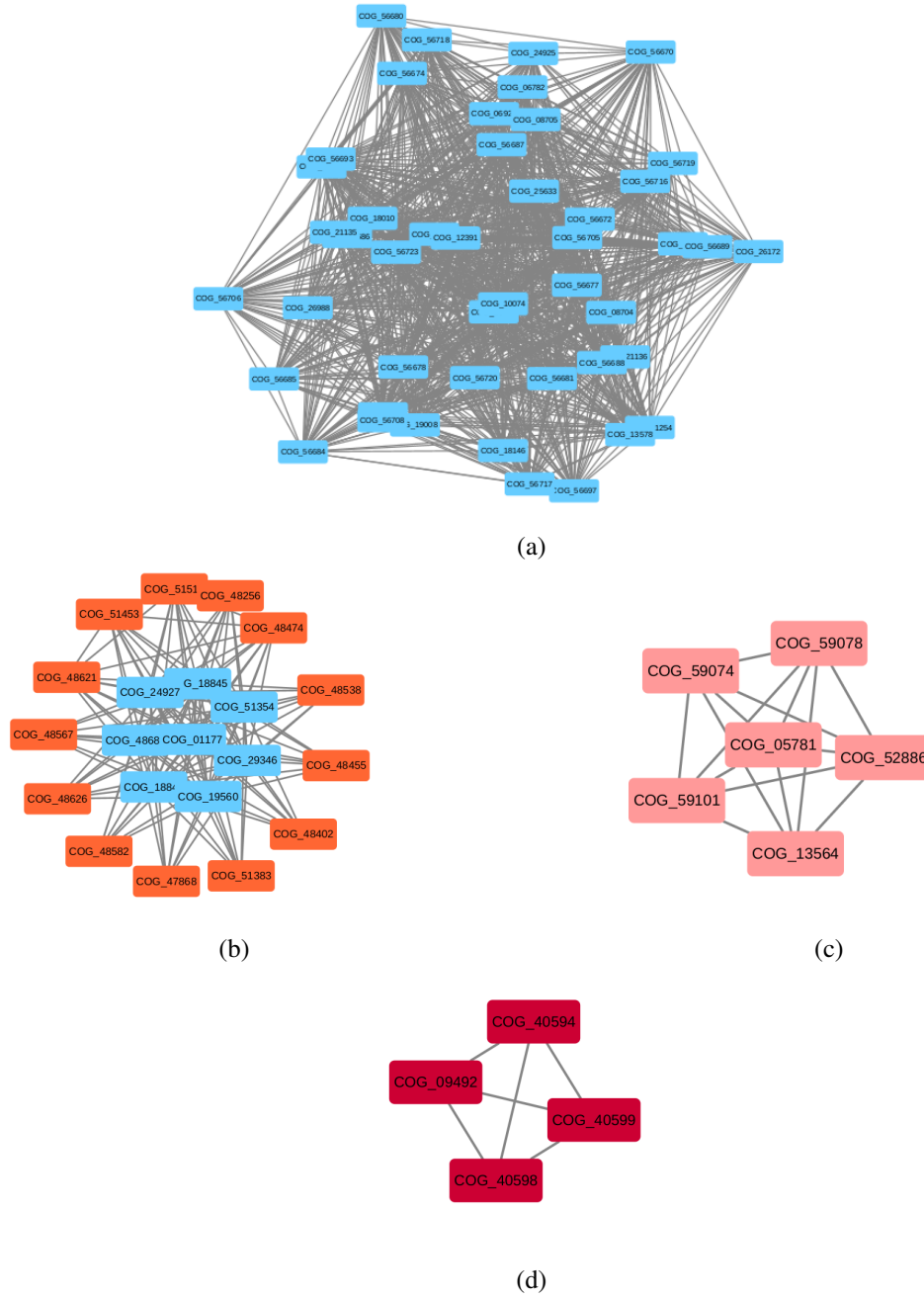


Figure 3.9: Examples of connected nodes in the coupling network created with Cytoscape. They all have strong positive effect sizes and couplings are significant. (a) A dense cluster with 46 nodes (COGs) and 1035 edges. All nodes belong to Bin ID 0. (b) A cluster with 21 nodes and 138 edges, with two bin classes: blue nodes are from Bin ID 0 and orange nodes are from Bin ID 5. (c) A smaller cluster with 6 nodes and 15 edges, where all nodes belong to Bin ID 3. (d) A small cluster in high bins, with 4 nodes and 6 edges. All nodes belong to Bin ID 8.

Chapter 4

Discussion

4.1 Evidence for coupled gene gain and loss

To determine whether genes evolve independently or in a coordinated manner, gene pairs were tested for consistent co-occurrence of gain and loss events across the phylogeny using a stratified Mantel–Haenszel framework. This section first evaluates the statistical support for coupling. It then examines how the strength and interpretation of the signal depend on the number of events and on whether the changes are gains or losses.

The distribution of CMH p -values shows a strong enrichment of very small values, indicating that a subset of gene pairs undergoes coordinated gain and loss (Figure 3.3a). In contrast, most gene pairs have p -values close to 1, consistent with independent evolutionary histories. After FDR correction, the q -value distribution becomes clearly bimodal. Only a small fraction of gene pairs remains significant, whereas the majority shifts to very high q -values. Weak and inconsistent associations are therefore filtered out, and the remaining pairs represent robust coupling signals.

Overall, 5.67% of all tested gene pairs remain significant at $q < 0.01$. Coupled gene evolution is therefore not ubiquitous but confined to a distinct minority of gene pairs. The CMH framework detects recurring co-evolutionary patterns while controlling false positives [59, 1].

The relationship between effect size and statistical significance provides further insight. Many gene pairs show large estimated log odds ratios but are not statistically significant, demonstrating that large effect sizes alone are insufficient evidence for coupling. Significant associations appear only when the signal is consistently observed across many evolutionary events. This behavior is expected because the Mantel–Haenszel test evaluates whether an association is stable across multiple strata.

As the number of gain and loss events increases, uncertainty in the odds ratio decreases. Consequently, moderate effect sizes can become highly significant when supported by sufficient evolutionary signal, whereas large effect sizes based on few events remain unreliable. With limited data, effect size estimates are unstable and may appear large by chance [30].

Most gene pairs have negative log-odds ratios, meaning their evolutionary changes usually occur on different branches rather than together (Figure 3.5). However, the distributions also contain numerous strong positive outliers far above the null line, indicating a subset of gene pairs whose changes often happen on the same branches and therefore are likely truly coupled.

Binning gene pairs by their minimum expected number of events clarifies that the strongly negative medians in the lowest bins are primarily a consequence of sparse data. When events are rare, coincident

changes are unlikely even under independence, which produces extreme negative estimates and wide spreads. These bins therefore provide less reliable evidence for coupled evolution.

As bin ID increases, fewer gene pairs remain but the estimates stabilize. Distributions narrow, become more symmetric, and shift toward zero. Only in the highest bin does the median become positive, and from intermediate bins onward an increasing fraction of pairs shows positive association, consistent with repeated concordant gain or loss events across the phylogeny. Consequently, meaningful inference about coupled evolution should focus on the higher eventfulness bins, where the signal is informative, while pairs from the lowest bins should be interpreted cautiously. To understand what type of evolutionary change generates these positive associations, the coupled events must be separated into joint gains and joint losses.

Figure 3.6 shows a clear skew toward co-gain events. Significant gene pairs co-occur on the same branches far more often through joint gains than through joint losses. This imbalance is visible in both mean and median values, indicating that the detected coupling signal is primarily driven by repeated acquisition rather than repeated deletion. Such a pattern is consistent with prokaryotic genome evolution, where horizontal gene transfer strongly shapes gene content [12]. Supporting this interpretation, the maximum-likelihood analysis in COUNT inferred substantial heterogeneity only for the transfer rate, whereas duplication showed little variation and loss was constrained to be uniform. Coupled events therefore concentrate in branches where genes enter genomes together.

Classical gene-content parsimony analyses assume that gene losses are more frequent than gains and therefore penalize gain events more strongly than losses. Mirkin et al. [62] formalized this idea using a weighted parsimony framework, in which evolutionary scenarios are scored by minimizing the total number of events while assigning a higher cost to gene gains. Under this assumption, a gene observed across distantly related taxa is best explained by ancestral presence followed by multiple independent losses rather than by repeated independent acquisitions.

This assumption is appropriate for datasets containing deeply diverged species. For example, if the same gene is found in mammals, plants, and bacteria, it is more plausible that the gene existed in a very ancient ancestor and was subsequently lost in many descendant lineages than that the identical gene was acquired independently many times in each group.

In contrast, the present dataset contains very closely related genomes. In this setting, patchy gene distributions are more plausibly explained by repeated recent acquisitions from a shared external gene pool than by numerous independent deletions. The loss-dominated assumption underlying classical parsimony reconstructions therefore does not necessarily apply, and gain events can become the main source of observable gene content variation.

Moreover, gains and losses are not inferred symmetrically from presence-absence data. A gain corresponds to a direct state transition, whereas a loss can only be inferred when ancestral presence is reconstructed with high confidence, and the model further restricts variation in loss rates. Consequently, heterogeneity across the phylogeny is preferentially attributed to gains rather than coordinated losses. This explains why most statistically supported coupling events arise from co-gains rather than co-losses: genes tend to enter genomes together along particular branches, while losses occur more diffusely and therefore provide weaker and less consistent evidence for coordinated evolution.

Reliable evidence for coupled gene evolution is therefore obtained primarily for gene pairs with sufficient evolutionary signal. Pairs supported by many gain or loss events yield stable effect-size

estimates and consistent associations across the phylogeny, whereas pairs with few events produce noisy and uncertain signals. Effect size and statistical significance must thus be interpreted jointly, as large estimated effects based on sparse data are unlikely to be reliable.

Overall, the results indicate that most detectable coupling signals originate from genes repeatedly gained together in particular lineages, and that detectability strongly depends on the number of observed evolutionary events and on model assumptions. Grouping gene pairs by event counts and applying a stratified statistical test reduces false positives and focuses inference on consistent associations while avoiding over-interpretation of poorly supported cases.

4.2 Functional patterns in high-degree COGs

Genes with the highest number of evolutionary couplings are listed in Table A.5. They fall into a small number of functional categories that are well known to be frequently gained and lost together during bacterial evolution.

A large fraction of these genes is annotated as mobile genetic elements and defense systems. This group includes transposases, integrases, and toxin–antitoxin modules (e.g. COG_18121, COG_10620, COG_18506, COG_09269, COG_16112, COG_33939, COG_15859, COG_03354, COG_39393). Transposases and integrases are core components of mobile genetic elements, such as plasmids and genomic islands [27]. Because these elements are often gained or lost as entire DNA blocks, the genes they carry tend to show many evolutionary couplings.

COG_18121 has the highest number of connections in this dataset and is annotated as ParE, a toxin belonging to a toxin–antitoxin system. ParE inhibits essential cellular processes but is normally neutralized by its corresponding antitoxin. When a plasmid carrying the system is lost, the unstable antitoxin is degraded, freeing the toxin and killing the plasmid-free cell. These systems are not required for normal cellular function [32, 67], but instead promote the maintenance of plasmids or genomic islands. Accordingly, the presence of ParE mainly signals mobile DNA, as these genes are rarely found in isolation.

Many other high-degree COGs are associated with transporters, regulatory proteins, and conditionally beneficial metabolic pathways [61]. This includes ABC transporters, amino acid and polyamine transport systems, as well as sulfur and amino acid metabolism (e.g. COG_21044, COG_25951, COG_16443, COG_55769). These genes are often organized in operons [72], which facilitates their coordinated gain and loss and makes them evolutionarily flexible.

Several highly connected COGs are involved in cell surface and envelope modification, including genes related to lipopolysaccharide and peptidoglycan biosynthesis (e.g. COG_33920, COG_60199). These processes directly affect the structure and chemistry of exposed cell surface components [74]. Such genes tend to evolve rapidly due to strong selective pressures from bacteriophages and host interactions. Phages initiate infection by binding to specific surface structures, and even small changes can block adsorption and prevent infection [8].

Finally, there is a strong enrichment of poorly characterized genes among the high-degree COGs. This is expected, as mobile and accessory genomic regions are less well annotated than core genomic regions. Accessory and mobile genes evolve rapidly and are lineage-specific, which limits homology-based functional inference and results in poorer annotation, as discussed by McInerney et al. [61].

Overall, COGs with many evolutionary couplings primarily reflect genome plasticity and modular gain and loss of accessory genes, rather than essential core cellular functions.

4.3 Interpretation of significant COG associations

Figure 3.9 shows four clusters of genes with significant coupling, each associated with different Bin IDs. A large number of such clusters were identified, ranging from densely connected clusters with many associated COGs to smaller clusters with only a few connections. The figure highlights selected representative examples that are examined here in more detail. The focus is on cluster (b) first, followed by clusters (c) and (d). Cluster (a) is not analyzed further because of its high connectivity. It is included to illustrate that very large, highly connected clusters were also detected.

Cluster (b) is notable because it contains genes from low- and high-Bin categories. Genes assigned to Bin ID 0 and Bin ID 5 are included. Interactions are predominantly observed between high-Bin genes (orange) and low-Bin genes (blue), whereas interactions between high-Bin genes themselves are less (Figure 3.9). In contrast, low-Bin genes show extensive interactions with one another. This interaction pattern gives rise to a star-like topology, in which low-Bin genes occupy central positions and are surrounded by multiple high-Bin genes.

Inspection of Tables A.8–A.11 show that individual annotations of high-Bin genes are heterogeneous, they can be grouped together to a small number of categories. Collectively, functions are of regulatory and signal-processing nature. At a general level, these genes encode proteins associated with signal transduction, regulatory ATPases, and also poorly characterized accessory domains. The ones annotated only by domains of unknown function show the same consistent coupling patterns to low-Bin genes. Examples of the annotated COGs are histidine kinase-related ATPases (COG_51511), AAA ATPase-containing proteins (COG_48538), and AIPR proteins (COG_48621). Regulatory ATPases are proteins that use hydrolysis to control regulatory processes. They are able to switch states and remodel protein complexes [31]. A similar functional principle will be introduced for GTP-binding proteins that are part of cluster (c).

Low-Bin genes in this cluster form a tightly connected core that is itself highly clustered. Histidine kinases (COG_29346), response regulators (COG_01177), MCP signaling domains (COG_51354), diguanylate cyclases (COG_19560), and diguanylate phosphodiesterases (COG_24927) are all strongly coupled to one another (effect sizes ~ 17.9). It indicates that they tend to be gained and lost as a complete unit. This low-Bin core represents a conserved backbone. All participate in cellular signal processing. External or internal signals are sensed, transmitted, and converted into cellular responses. Together, they allow the cell to detect environmental changes and adjust behaviors, including movement-related responses [82] [89].

High-Bin genes do not show comparably strong or dense coupling among themselves. Instead they are connected to this low-Bin regulatory core. For example, AAA ATPase domain proteins (COG_48538) and DUF-containing proteins (COG_47868, COG_51453) are repeatedly coupled to MCP signaling domains and histidine kinases. This suggests that these proteins help control chemotaxis signaling, for example by stabilizing signaling complexes, switching them between active and inactive states, or adjusting signal strength [23].

Overall, the observed coupling structure in this cluster supports a model in which a stable low-Bin

regulatory core is extended by lineage-specific high-Bin components. They are evolutionary linked to the core machinery, but not essential [28]. High-Bin COGs are a group of genes that were gained and lost many times during evolution. They are evolutionarily dynamic rather than stably conserved. These gene families are expected to belong to the flexible genome and track ecological selection pressures rather than long-term conservation [12]. This allows *Vibrio* lineages to gain or lose regulatory functions in response to a changing environment.

Cluster (c) is discussed next. It consists of a small set of COGs that show nearly perfect evolutionary coupling. Every pair inside the cluster has exactly the same effect size. Consequently, there are no branches where one gene changes state without the others. They are repeatedly gained and lost together across the phylogeny. The details are reported in Table A.6.

One component of this cluster is COG_05781. It is annotated as a transcriptional regulator. Proteins in this category bind DNA and control the expression of nearby genes. Regulators of this kind are often specific rather than global [80]. Typically, they are frequently located in the same operon or genomic island as the genes they regulate [72]. As a result, transcriptional regulators show strong co-gain and co-loss patterns with their target genes [10]. COG_52886 is annotated as a GTP-binding protein. These proteins generally function as molecular switches. In other words, they cycle between active and inactive states depending on nucleotide binding [7]. They often play regulatory roles. For example coordinating the activity of protein complexes or responding to environmental pressure [86]. Another member, COG_59078 contains a RadC-type JAB domain, a protein domain that is widely distributed across bacteria. RadC-type JAB domains in bacteria are primarily associated with DNA- and nucleotide-related functions. These domains are often fused to proteins involved in DNA processing. This implies assistance of DNA-related processes by interacting with other proteins [41]. In context of the cluster, it supports the interpretation that the RadC-like JAB protein functions as part of a co-evolving genetic module. Finally, COG_13564 is annotated as a protein of unknown function belonging to the DUF1232 family. COG_59074 and COG_59101 lack annotations.

The functional annotations of the genes in cluster (c) indicate a shared role in a tightly linked genetic module. Regulation, signaling, and DNA-associated functions are combined here. The transcriptional regulator may regulate one or more genes within the cluster. The GTP-binding protein influence the activity of the system by sensing environmental conditions. A suggestion is, that selection acts on this set of genes as a single unit. From the identical effect sizes it follows that they are transferred, maintained, or deleted together. That is likely because of their physical linkage in the same operon. The unannotated COGs show strong and consistent evolutionary coupling with the annotated genes. Although their molecular functions remain unknown, their evolutionary behavior could mean involvement in the same regulatory or DNA-associated processes.

Finally, cluster (d) consists exclusively of Bin ID 8 genes with identical, strong positive effect sizes (Table A.7). As in cluster (c), the effect sizes in this small cluster are uniform and indicate near-perfect evolutionary coupling. All genes are gained and lost together across the phylogeny, with no branches showing independent changes. This suggests again that these COGs form a single functional unit.

A central component of this cluster is COG_09492. It is annotated as flagellin protein, the main structural subunit of the bacterial flagellum. Flagellin is essential for motility and is typically part of a large and regulated gene cluster [56]. The remaining COGs (COG_40594, COG_40598) are missing an annotation. Just COG_40599 is described as an intracellular phage-related Abi (Abortive infection)

component. Abi systems are bacterial defense mechanisms where an infected cell sacrifices itself to stop a virus from replicating and spreading to other cells [50]. There is a perfect coupling between a core motility gene and an Abi-component. These genes are physically linked and are selected as a single unit, similar to the tightly co-evolving module in cluster (c). A plausible biological interpretation is that motility-related functions are coupled to phage defense mechanisms. A reason for this could be that phages use the flagella structures as guides to attach at the cell surface. This potentially increases the infection risk. [75]. Selection therefore favors coupling these genes.

The unannotated COGs in this cluster show the same evolutionary behavior. This strongly implies again functional involvement in the same process.

In summary, the analysis of coupled gene clusters reveals that strongly co-evolving COGs form linked modules. Nearly identical effect sizes within clusters indicate that these genes are repeatedly gained and lost together, or even physical linked. Many genes are in unannotated COGs, but the consistent coupling to well-characterized COGs provides strong evidence that they participate in the same biological processes. This interpretation is consistent with an earlier phylogenetic profiling study, which demonstrated that genes with highly correlated patterns across genomes are frequently functionally linked [69].

Across clusters, the inferred functions are biologically plausible and align with established knowledge. Regulatory proteins, signaling components, motility genes, and defense-related factors are repeatedly found in tightly connected groups. Similar coupling patterns are seen in different clusters. This lead to the conclusion that the method developed here recovers biologically meaningful gene-gene relationships that are supported by existing literature.

Chapter 5

Conclusion

The genus *Vibrio* provided a suitable model for investigating ecological genomics and the evolutionary consequences of gene gain and loss at relatively fine phylogenetic scale. After quality control, only a small fraction of assemblies required exclusion, resulting in a largely consistent dataset of 401 genomes.

The analytical workflow combined established comparative-genomics methods with newly implemented components. A core genome alignment was generated with Parsnp2, followed by iterative reconstruction of a clonal phylogeny using ClonalFrameML and IQ-TREE3. Gene content was determined by predicting open reading frames with Pyrodigal, clustering homologs using FastCOG, and annotating them via eggNOG-mapper. Maximum-likelihood gain and loss rates were estimated with COUNT, which also provided posterior probabilities for ancestral state reconstruction. Based on these reconstructions, a newly developed Python module inferred coupled gain and loss events by constructing pairwise contingency tables and applying stratified association testing via the Cochran–Mantel–Haenszel test. To control for phylogenetic dependence, the statistical framework was complemented by a parametric continuous-time Markov chain bootstrap null model. Despite testing 13,038,171 gene-family pairs, the module remained computationally tractable. In practice, the dominant computational cost of the entire workflow was the maximum-likelihood inference in COUNT.

A central methodological step was the reconstruction of a recombination-aware clonal phylogeny before ancestral state inference. The gain and loss history of each gene family is projected onto this tree, so any topological distortion directly translates into incorrect evolutionary events, for example falsely inferring multiple independent gains instead of a single horizontally transferred acquisition. Recombination was detected in 182 genomes, with a median recombination burden of 0.195%. This is consistent with previous reports for closely related bacteria. Although the affected fraction is small, recombination mainly occurs at informative sites determining short internal branches and therefore alters relationships among very similar strains. In such datasets, even infrequent recombination can bias ancestral reconstructions and inflate apparent gene gain and loss, making explicit correction necessary for reliable evolutionary inference.

The final association analysis yielded numerous significant gene pairs. Controlling the false discovery rate at 0.01, 5.67% of all tested gene pairs showed coupled gain or loss. Several high-degree COGs were detected, including uncharacterized families as well as genes forming known biological systems, supporting a modular organization of genome evolution. Many inferred gene–gene relationships are also consistent with previously reported biological interactions in the literature.

A Comparison with CoPAP [11] shows that the conclusions are consistent despite methodological

differences. CoPAP detected 5,605 significant pairs (0.062%) among 9,065,282 tested pairs at the same FDR threshold, whereas the present workflow identified 739,264 significant pairs (5.67%) among 13,038,171 pairs. Despite the difference in detection rate and the model genomes, the qualitative patterns coincide. Co-evolving genes form non-random, highly structured networks, decompose into functional modules (for example flagellar systems), and frequently link uncharacterized genes to known biological pathways. Thus, the approach recovers biologically meaningful and interpretable gene–gene associations.

Overall, integrating recombination-aware phylogeny reconstruction with probabilistic ancestral state inference and stratified association testing provides a scalable framework for studying genome evolution.

Appendix A

Script	Purpose
<code>__main__.py</code>	Provides the command-line entry point that starts the analysis.
<code>pipeline.py</code>	Implements the workflow that connects configuration parsing, data preparation, contingency construction, CMH testing, multiple-testing correction, optional CTMC null modeling, and diagnostics.
<code>config_manager.py</code>	Loads and validates user configuration parameters. They control analysis behavior, file paths, and runtime options.
<code>logger.py</code>	Defines a centralized logging utility that uses a consistent log formatting.
<code>storage_manager.py</code>	Handles all input and output operations. Takes care of a consistent directory structure and atomic writes to ensure safety in parallel runs.
<code>prep_engine.py</code>	Loads and preprocesses branch-by-COG gain and loss posterior probability matrices and branch-rate estimates.
<code>pair_worker.py</code>	Processes chunks of gene pairs by converting branch-wise gain and loss posterior probabilities into expected 2×2 contingency tables for statistics.
<code>stats_engine.py</code>	Performs stratified Cochran–Mantel–Haenszel testing across gain and loss strata. Computes odds ratios, test statistics, p-values, and Benjamini–Hochberg FDR correction.
<code>null_model_ctcm.py</code>	Implements a parametric continuous-time Markov chain bootstrap to generate null expectations for gene gain and loss coupling.
<code>diagnostics.py</code>	Generates diagnostic summaries of statistical results, top-ranked gene pairs, and observed-versus-null comparisons.
<code>__init__.py</code>	Defines the package namespace and initialization logic. The project can be imported as a Python module.
<code>requirements.txt</code>	Lists the external Python dependencies and version constraints required to run the pipeline.

Table A.1: Overview of custom Python scripts in the ProGeneCoupling project. Each module and its role in the coupled gene gain/loss analysis pipeline are listed.

Parameter	Value
<code>gain_matrix</code>	<code>pro_gene_coupling/input/branch_cog_gain_matrix.csv</code>
<code>loss_matrix</code>	<code>pro_gene_coupling/input/branch_cog_loss_matrix.csv</code>
<code>tree_path</code>	<code>pro_gene_coupling/input/tree.newick</code>
<code>rates_path</code>	<code>pro_gene_coupling/input/branch_rates.csv</code>
<code>out_dir</code>	<code>pro_gene_coupling_results</code>
<code>chunk_size</code>	100000
<code>n_workers</code>	1
<code>rng_seed</code>	17
<code>continuity_correction</code>	false
<code>bootstrap_reps</code>	500
<code>ctcm_bootstrap</code>	true
<code>min_expected_events</code>	1.0
<code>n_bins</code>	10
<code>write_every</code>	10
<code>debug_n_cogs</code>	0
<code>no_diagnostics</code>	false
<code>drop_cogs</code>	ABSENT, Unnamed: 0
<code>drop_cogs_file</code>	(empty)

Table A.2: Configuration parameters used in the final ProGeneCoupling run.

Script	Purpose
run_pyrodigal.py	Predicts open reading frames from bacterial genomes using Pyrodigal. It outputs GFF files, protein translations, and ORF summary statistics.
blastn_for_gene_contigs.py	Assigns contigs to taxa by running BLASTn searches. It summarizes taxonomic hit counts per contig across genomes.
genome_metrics.py	Computes genome assembly statistics including total length, GC content, contig count, and N50 for multiple genomes.
run_parsnp_alignment.py	Performs core genome alignment using Parsnp2 and generates an XMFA alignment.
rename_xmfa_headers.py	Rewrites Parsnp2 generated XMFA headers by replacing numeric taxon indices with genome names.
xmfa_remove_recombination.py	Masks recombinant regions in an XMFA alignment. It uses the ClonalFrameML output to produce a recombination-filtered alignment.
convert_xmfa_to_fasta.py	Converts an XMFA alignment into a concatenated FASTA file.
rename_nwk_labels.py	Cleans Newick tree leaf labels by removing file suffixes.
run_clonalframeml.py	Runs the ClonalFrameML tool to infer recombination events on a phylogeny and logs all details.
compare_trees.py	Compares phylogenetic trees using Robinson–Foulds distance, branch score distance, and total branch-length differences.
iTOL_ordinal_day.py	Generates an iTOL color-strip annotation. It groups genomes by sampling day inferred from filenames.
iTOL_genome_size.py	Creates an iTOL SIMPLEBAR dataset to visualize normalized genome sizes on a phylogenetic tree.
iTOL_recombination_burden.py	Computes per-genome recombination burden from the ClonalFrameML output and generates an iTOL gradient dataset.
build_COG_matrix.py	Constructs protein families (COGs) and outputs a protein-to-COG mapping table.
parse_eggNOG_COG_to_function.py	Assigns functional annotations to COGs by summarizing eggNOG-mapper descriptions across included genes.
summarize_cog_contigs.py	Quantifies per-COG contig fragmentation statistics across genomes.
prepare_count.py	Converts a COG presence/absence matrix into a COUNT compatible format. It also validates consistency with the phylogeny.
run_count_gainloss.py	Runs COUNT to estimate gene gain and loss rates or compute posterior probabilities along a phylogeny.
parse_count_rates.py	Parses COUNT maximum-likelihood output to extract branch-specific gain, loss, duplication, and transfer rates.
parse_count_posteriors.py	Parses COUNT posterior probability output and exports a cleaned matrix.
construct_gain_loss_matrix.py	Constructs branch-by-COG matrices for gain, loss, expansion, and reduction events from COUNT posteriors.
trim_cog_rows_by_event_counts.py	Filters COGs based on total expected event counts. It can use range-based or percentile-based trimming.
generate_presence_absence_heatmap.py	Generates a high-resolution heatmap from a binary COG presence/absence matrix for visualization.
filter_fasta_alignment_by_name.py	Filters a FASTA alignment to retain only sequences that match a predefined list of genome names.

Table A.3: Overview of custom Python scripts helping with genome processing, phylogenetic analysis, gene family inference, and visualizations.

Genome	total recombination burden (%)	Genome	total recombination burden (%)
10N.232.45.E11	0.19	10N.242.46.C07	0.04
10N.232.45.E12	0.27	10N.242.46.D07	0.16
10N.236.47.F07	0.22	10N.242.46.F12	0.03
10N.237.45.A01	0.37	10N.242.46.G12	0.07
10N.237.45.A04	0.56	10N.242.47.A04	0.02
10N.237.45.A06	0.94	10N.244.45.B03	0.66
10N.237.45.A11	0.21	10N.244.45.C09	0.31
10N.237.45.A12	0.14	10N.244.45.D03	0.75
10N.237.45.B11	0.01	10N.244.45.D05	0.02
10N.237.45.B12	0.38	10N.244.45.E03	0.21
10N.237.45.C09	0.72	10N.244.45.E05	0.20
10N.237.45.C11	0.57	10N.244.45.E08	0.45
10N.237.45.D03	0.45	10N.244.45.E10	0.04
10N.237.45.D12	0.65	10N.244.45.E11	0.02
10N.237.45.E04	0.56	10N.244.45.F04	0.09
10N.237.45.E05	0.12	10N.244.45.F05	0.29
10N.237.45.E07	0.25	10N.244.45.H07	0.00
10N.237.45.E08	0.58	10N.244.45.H09	0.81
10N.237.45.E11	0.43	10N.244.46.A02	0.68
10N.237.45.F02	0.14	10N.244.46.A09	0.62
10N.237.45.F06	0.01	10N.244.46.A10	0.20
10N.237.45.F08	0.39	10N.244.46.B08	0.49
10N.237.45.F12	0.42	10N.244.46.B09	0.10
10N.237.45.G11	0.04	10N.244.46.C07	0.10
10N.237.45.G12	0.64	10N.244.46.C09	0.06
10N.237.45.H06	0.12	10N.244.46.C12	0.26
10N.237.46.A04	0.19	10N.244.46.D03	0.14
10N.237.46.A05	0.51	10N.244.46.D09	0.66
10N.237.46.A06	0.47	10N.244.46.D10	0.07
10N.237.46.A09	0.34	10N.244.46.E12	0.06
10N.237.46.A10	0.24	10N.244.46.F08	0.08
10N.237.46.B04	0.41	10N.244.46.F11	0.06
10N.237.46.B05	0.41	10N.244.46.G10	0.26
10N.237.46.B08	0.00	10N.244.46.G12	0.14
10N.237.46.B10	0.01	10N.244.46.H11	0.34
10N.237.46.B11	0.16	10N.244.47.A05	0.05
10N.237.46.C03	0.05	10N.247.45.A05	0.72
10N.237.46.C04	0.20	10N.247.45.A08	0.47
10N.237.46.C10	0.11	10N.247.45.A09	0.87
10N.237.46.C11	0.24	10N.247.45.A10	0.62

Table A.4: Complete percentage of recombination found and removed per genome. In the remaining genomes, no recombination was detected.

COG	n_connections	Top description	KEGG KO	KEGG pathway
COG_18121	1044	ParE toxin of type II toxin–antitoxin system, parDE	ko:K19092	–
COG_21740	1033	Protein of unknown function (DUF2913)	–	–
COG_08058	1032	User locus_tag	–	–
COG_33920	1018	Lipid A core–O-antigen ligase	ko:K02847	map01100
COG_10620	1017	Antitoxin component of a toxin–antitoxin (TA) module	–	–
COG_13931	993	Endoribonuclease L-PSP	ko:K09022	–
COG_03354	969	Integrase core domain	–	–
COG_09269	966	Transposase ISVa5 (IS66 family)	–	–
COG_03124	938	Dehydrogenase	ko:K13877	map00053
COG_21044	938	Binding protein component of ABC transporter	ko:K02055	map02024
COG_06777	924	Sulfate adenylyltransferase subunit 2	ko:K00957	map01130
COG_33377	923	Catalyzes the synthesis of activated sulfate	ko:K00860	map01120
COG_03122	920	FCD domain	–	–
COG_17642	920	Belongs to the DapA family	ko:K21062	map00330
COG_21042	920	Ornithine cyclodeaminase, mu-crystallin homolog	ko:K19742	map00330
COG_16443	916	Spermidine/putrescine transport system, permease	ko:K02053	map02024
COG_33877	915	Belongs to the UPF0319 family	ko:K09909	–
COG_60199	900	Catalyzes cross-linking of the peptidoglycan cell wall	ko:K05515	map01501
COG_00333	891	Transcriptional regulator	–	–
COG_14669	891	Acetyl-CoA carboxylase biotin carboxylase subunit	ko:K01963	map01212
COG_18501	889	AAA domain protein	–	–
COG_16112	887	Transposase	ko:K07497	–
COG_21040	883	–	–	–
COG_18499	879	–	–	–
COG_03120	877	Restriction endonuclease	ko:K07448	–
COG_33939	875	Transposase	ko:K07497	–
COG_21075	871	–	–	–
COG_33951	871	tRNA processing	–	–
COG_33958	871	Imidazole glycerol phosphate synthase (HisH/HisF system)	ko:K02501	map01230
COG_15859	867	Transposase ISVa5 (IS66 family)	–	–
COG_27562	865	Spore coat polysaccharide biosynthesis protein	–	–
COG_28528	865	Cytidyltransferase	ko:K00983	map01100
COG_18506	863	Antitoxin component of a toxin–antitoxin (TA) module	–	–
COG_25951	863	ABC transporter PotABCD, ATP-binding protein	ko:K02052	map02024
COG_15783	858	AAA domain protein	–	–
COG_08906	857	Beta-eliminating lyase family protein	ko:K01667	map00380
COG_48414	852	Transcriptional regulator	–	–
COG_00001	846	Carbohydrate-selective porin, OprB family	ko:K07267	–
COG_12161	846	–	–	–
COG_19878	846	Calcineurin-like phosphoesterase	–	–
COG_39393	846	ParB-like nuclease domain	–	–
COG_55740	846	Domain of unknown function (DUF3440)	–	–
COG_55769	846	Amino acid permease (COG0814)	ko:K03835	–
COG_28224	845	to SP P39213	ko:K07483	–
COG_21039	840	–	–	–
COG_21041	840	Transcriptional regulator	–	–
COG_47945	840	Putative mono-oxygenase YdhR	–	–

Table A.5: Top 50 COGs ranked by number of significant gene–gene couplings (degree) among all FDR-significant pairs ($q_{\text{cmh}} \leq 0.05$).

bin_A	bin_B	COG_A	COG_B	description_A	description_B	effect_size
3	3	COG_52886	COG_59074	GTP binding	-	12.968
3	3	COG_13564	COG_59101	Protein of unknown function (DUF1232)	-	12.968
3	3	COG_13564	COG_59078	Protein of unknown function (DUF1232)	RadC-like JAB domain	12.968
3	3	COG_13564	COG_59074	Protein of unknown function (DUF1232)	-	12.968
3	3	COG_13564	COG_52886	Protein of unknown function (DUF1232)	GTP binding	12.968
3	3	COG_05781	COG_59101	transcriptional regulator	-	12.968
3	3	COG_05781	COG_59078	transcriptional regulator	RadC-like JAB domain	12.968
3	3	COG_05781	COG_59074	transcriptional regulator	-	12.968
3	3	COG_05781	COG_52886	transcriptional regulator	GTP binding	12.968
3	3	COG_05781	COG_13564	transcriptional regulator	Protein of unknown function (DUF1232)	12.968
3	3	COG_59078	COG_59101	RadC-like JAB domain	-	12.968
3	3	COG_59074	COG_59101	-	-	12.968
3	3	COG_59074	COG_59078	-	RadC-like JAB domain	12.968
3	3	COG_52886	COG_59101	GTP binding	-	12.968
3	3	COG_52886	COG_59078	GTP binding	RadC-like JAB domain	12.968

Table A.6: Gene pair couplings in cluster (c). This table reports all statistically significant gene–gene couplings that are part of cluster (c) in Figure 3.9. All COGs belong to Bin ID 3 and all interactions have strong positive effect sizes.

bin_A	bin_B	COG_A	COG_B	description_A	description_B	effect_size
8	8	COG_40594	COG_40598	-	-	15.063
8	8	COG_09492	COG_40599	Flagellin is the subunit protein which polymerizes to form the filaments of bacterial flagella	Intracellular sensor of Lambda phage, Abi component	15.063
8	8	COG_09492	COG_40598	Flagellin is the subunit protein which polymerizes to form the filaments of bacterial flagella	-	15.063
8	8	COG_09492	COG_40594	Flagellin is the subunit protein which polymerizes to form the filaments of bacterial flagella	-	15.063
8	8	COG_40598	COG_40599	-	Intracellular sensor of Lambda phage, Abi component	15.063
8	8	COG_40594	COG_40599	-	Intracellular sensor of Lambda phage, Abi component	15.063

Table A.7: Gene pair couplings in cluster (d). This table reports all statistically significant gene–gene couplings that are part of cluster (d) in Figure 3.9. All COGs belong to Bin ID 8 and all interactions show very strong positive effect sizes.

bin_A	bin_B	COG_A	COG_B	description_A	description_B	effect_size
0	5	COG_18845	COG_47868	-	Protein of unknown function (DUF4435)	13.128
0	5	COG_18845	COG_51511	-	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	13.293
0	5	COG_18845	COG_51453	-	Putative PD-(D/E)XK family member, (DUF4420)	13.293
0	5	COG_18845	COG_48621	-	AIPR protein	13.293
0	5	COG_18845	COG_48474	-	Z1 domain	13.293
0	0	COG_18845	COG_51354	-	Methyl-accepting chemotaxis protein (MCP) signalling domain	17.915
0	0	COG_18845	COG_48681	-	-	17.915
0	0	COG_18845	COG_29346	-	Histidine kinase	17.915
0	0	COG_18845	COG_24927	-	Putative diguanylate phosphodiesterase	17.915
0	0	COG_18845	COG_19560	-	diguanylate cyclase	17.915
0	5	COG_18844	COG_51383	-	-	13.128
0	5	COG_18844	COG_48626	-	-	13.128
0	5	COG_18844	COG_48582	-	-	13.128
0	5	COG_18844	COG_48567	-	-	13.128
5	0	COG_48626	COG_51354	-	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.128
0	5	COG_18844	COG_48538	-	AAA ATPase domain	13.128
5	0	COG_48626	COG_48681	-	-	13.128
5	0	COG_48582	COG_51354	-	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.128
0	5	COG_18844	COG_48455	-	protein homooligomerization	13.128
5	0	COG_48582	COG_48681	-	-	13.128
0	5	COG_18844	COG_48402	-	-	13.128
5	0	COG_48567	COG_51354	-	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.128
0	5	COG_18844	COG_48256	-	-	13.128
5	0	COG_48567	COG_48681	-	-	13.128
0	5	COG_18844	COG_47868	-	Protein of unknown function (DUF4435)	13.128
5	0	COG_48538	COG_51354	AAA ATPase domain	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.128
0	5	COG_18844	COG_51511	-	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	13.293
5	0	COG_48538	COG_48681	AAA ATPase domain	-	13.128
0	5	COG_18844	COG_51453	-	Putative PD-(D/E)XK family member, (DUF4420)	13.293
5	0	COG_48455	COG_51354	protein homooligomerization	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.128
0	5	COG_18844	COG_48621	-	AIPR protein	13.293
5	0	COG_48455	COG_48681	protein homooligomerization	-	13.128
0	5	COG_18844	COG_48474	-	Z1 domain	13.293
5	0	COG_48402	COG_51354	-	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.128
0	0	COG_18844	COG_51354	-	Methyl-accepting chemotaxis protein (MCP) signalling domain	17.915
5	0	COG_48402	COG_48681	-	-	13.128
0	0	COG_18844	COG_48681	-	-	17.915
5	0	COG_48256	COG_51354	-	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.128
0	0	COG_18844	COG_29346	-	Histidine kinase	17.915
5	0	COG_48256	COG_48681	-	-	13.128
0	0	COG_18844	COG_24927	-	Putative diguanylate phosphodiesterase	17.915
5	0	COG_47868	COG_51354	Protein of unknown function (DUF4435)	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.128
0	0	COG_18844	COG_19560	-	diguanylate cyclase	17.915
5	0	COG_47868	COG_48681	Protein of unknown function (DUF4435)	-	13.128
0	0	COG_18844	COG_18845	-	-	17.915

Table A.8: Gene pair couplings in cluster (b), Part 1. This table reports all statistically significant gene–gene couplings that are part of cluster (b) in Figure 3.9. All COGs belong to Bin ID 0 or 5. All interactions have strong positive effect sizes.

bin_A	bin_B	COG_A	COG_B	description_A	description_B	effect_size
0	5	COG_01177	COG_51383	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	-	13.128
0	5	COG_01177	COG_48626	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	-	13.128
0	5	COG_01177	COG_48582	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	-	13.128
0	5	COG_01177	COG_48567	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	-	13.128
0	5	COG_01177	COG_48538	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	AAA ATPase domain	13.128
0	5	COG_01177	COG_48455	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	protein homooligomerization	13.128
0	5	COG_01177	COG_48402	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	-	13.128
0	5	COG_01177	COG_48256	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	-	13.128
0	5	COG_01177	COG_47868	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	Protein of unknown function (DUF4435)	13.128
0	5	COG_01177	COG_51511	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	13.293
0	5	COG_01177	COG_51453	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	Putative PD-(D/E)XK family member, (DUF4420)	13.293
0	5	COG_01177	COG_48621	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	AIPR protein	13.293
0	5	COG_01177	COG_48474	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	Z1 domain	13.293
0	0	COG_01177	COG_51354	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	Methyl-accepting chemotaxis protein (MCP) signalling domain	17.915
0	0	COG_01177	COG_48681	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	-	17.915
0	0	COG_01177	COG_29346	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	Histidine kinase	17.915
0	0	COG_01177	COG_24927	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	Putative diguanylate phosphodiesterase	17.915
0	0	COG_01177	COG_19560	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	diguanylate cyclase	17.915
0	0	COG_01177	COG_18845	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	-	17.915
0	0	COG_01177	COG_18844	Response regulator containing a CheY-like receiver domain and an HTH DNA-binding domain	-	17.915

Table A.9: Gene pair couplings in cluster (b) , Part 2. This table reports all statistically significant gene–gene couplings that are part of cluster (b) in Figure 3.9. All COGs belong to Bin ID 0 or 5. All interactions have strong positive effect sizes.

bin_A	bin_B	COG_A	COG_B	description_A	description_B	effect_size
0	5	COG_51354	COG_51383	Methyl-accepting chemotaxis protein (MCP) signalling domain	-	13.128
0	5	COG_51354	COG_51511	Methyl-accepting chemotaxis protein (MCP) signalling domain	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	13.293
0	5	COG_51354	COG_51453	Methyl-accepting chemotaxis protein (MCP) signalling domain	Putative PD-(D/E)XK family member, (DUF4420)	13.293
0	5	COG_48681	COG_51383	-	-	13.128
0	5	COG_48681	COG_51511	-	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	13.293
0	5	COG_48681	COG_51453	-	Putative PD-(D/E)XK family member, (DUF4420)	13.293
0	0	COG_48681	COG_51354	-	Methyl-accepting chemotaxis protein (MCP) signalling	
0	5	COG_29346	COG_51383	Histidine kinase	-	13.128
0	5	COG_29346	COG_48626	Histidine kinase	-	13.128
0	5	COG_29346	COG_48582	Histidine kinase	-	13.128
0	5	COG_29346	COG_48567	Histidine kinase	-	13.128
0	5	COG_29346	COG_48538	Histidine kinase	AAA ATPase domain	13.128
0	5	COG_29346	COG_48455	Histidine kinase	protein homooligomerization	13.128
0	5	COG_29346	COG_48402	Histidine kinase	-	13.128
0	5	COG_29346	COG_48256	Histidine kinase	-	13.128
0	5	COG_29346	COG_47868	Histidine kinase	Protein of unknown function (DUF4435)	13.128
0	5	COG_29346	COG_51511	Histidine kinase	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	13.293
0	5	COG_29346	COG_51453	Histidine kinase	Putative PD-(D/E)XK family member, (DUF4420)	13.293
5	5	COG_51453	COG_51511	Putative PD-(D/E)XK family member, (DUF4420)	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	17.038
0	5	COG_29346	COG_48621	Histidine kinase	AIPR protein	13.293
5	0	COG_48621	COG_51354	AIPR protein	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.293
0	5	COG_29346	COG_48474	Histidine kinase	Z1 domain	13.293
5	0	COG_48621	COG_48681	AIPR protein	-	13.293
0	0	COG_29346	COG_51354	Histidine kinase	Methyl-accepting chemotaxis protein (MCP) signalling domain	17.915
5	5	COG_48621	COG_51511	AIPR protein	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	17.038
0	0	COG_29346	COG_48681	Histidine kinase	-	17.915
5	5	COG_48621	COG_51453	AIPR protein	Putative PD-(D/E)XK family member, (DUF4420)	17.038
0	5	COG_24927	COG_51383	Putative diguanylate phosphodiesterase	-	13.128
0	5	COG_24927	COG_48626	Putative diguanylate phosphodiesterase	-	13.128
5	0	COG_48474	COG_51354	Z1 domain	Methyl-accepting chemotaxis protein (MCP) signalling domain	13.293
0	5	COG_24927	COG_48582	Putative diguanylate phosphodiesterase	-	13.128
5	0	COG_48474	COG_48681	Z1 domain	-	13.293
0	5	COG_24927	COG_48567	Putative diguanylate phosphodiesterase	-	13.128
5	5	COG_48474	COG_51511	Z1 domain	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	17.038
0	5	COG_24927	COG_48538	Putative diguanylate phosphodiesterase	AAA ATPase domain	13.128
5	5	COG_48474	COG_51453	Z1 domain	Putative PD-(D/E)XK family member, (DUF4420)	17.038
0	5	COG_24927	COG_48455	Putative diguanylate phosphodiesterase	protein homooligomerization	13.128
5	5	COG_48474	COG_48621	Z1 domain	AIPR protein	17.038
0	5	COG_24927	COG_48402	Putative diguanylate phosphodiesterase	-	13.128
0	5	COG_24927	COG_48256	Putative diguanylate phosphodiesterase	-	13.128
0	5	COG_24927	COG_47868	Putative diguanylate phosphodiesterase	Protein of unknown function (DUF4435)	13.128
0	5	COG_24927	COG_51511	Putative diguanylate phosphodiesterase	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	13.293
0	5	COG_24927	COG_51453	Putative diguanylate phosphodiesterase	Putative PD-(D/E)XK family member, (DUF4420)	13.293
0	5	COG_24927	COG_48621	Putative diguanylate phosphodiesterase	AIPR protein	13.293
0	5	COG_24927	COG_48474	Putative diguanylate phosphodiesterase	Z1 domain	13.293
0	0	COG_24927	COG_51354	Putative diguanylate phosphodiesterase	Methyl-accepting chemotaxis protein (MCP) signalling domain	17.915
0	0	COG_24927	COG_48681	Putative diguanylate phosphodiesterase	-	17.915
0	0	COG_24927	COG_29346	Putative diguanylate phosphodiesterase	Histidine kinase	17.915

Table A.10: Gene pair couplings in cluster (b) , Part 3. This table reports all statistically significant gene–gene couplings that are part of cluster (b) in Figure 3.9. All COGs belong to Bin ID 0 or 5. All interactions have strong positive effect sizes.

bin_A	bin_B	COG_A	COG_B	description_A	description_B	effect_size
0	5	COG_19560	COG_51383	diguanylate cyclase	-	13.128
0	5	COG_19560	COG_48626	diguanylate cyclase	-	13.128
0	5	COG_19560	COG_48582	diguanylate cyclase	-	13.128
0	5	COG_19560	COG_48567	diguanylate cyclase	-	13.128
0	5	COG_19560	COG_48538	diguanylate cyclase	AAA ATPase domain	13.128
0	5	COG_19560	COG_48455	diguanylate cyclase	protein homooligomerization	13.128
0	5	COG_19560	COG_48402	diguanylate cyclase	-	13.128
0	5	COG_19560	COG_48256	diguanylate cyclase	-	13.128
0	5	COG_19560	COG_47868	diguanylate cyclase	Protein of unknown function (DUF4435)	13.128
0	5	COG_19560	COG_51511	diguanylate cyclase	Histidine kinase-, DNA gyrase B-, and HSP90-like ATPase	13.293
0	5	COG_19560	COG_51453	diguanylate cyclase	Putative PD-(D/E)XK family member, (DUF4420)	13.293
0	5	COG_19560	COG_48621	diguanylate cyclase	AIPR protein	13.293
0	5	COG_19560	COG_48474	diguanylate cyclase	Z1 domain	13.293
0	0	COG_19560	COG_51354	diguanylate cyclase	Methyl-accepting chemotaxis protein (MCP) signalling domain	17.915
0	0	COG_19560	COG_48681	diguanylate cyclase	-	17.915
0	0	COG_19560	COG_29346	diguanylate cyclase	Histidine kinase	17.915
0	0	COG_19560	COG_24927	diguanylate cyclase	Putative diguanylate phosphodiesterase	17.915
0	5	COG_18845	COG_51383	-	-	13.128
0	5	COG_18845	COG_48626	-	-	13.128
0	5	COG_18845	COG_48582	-	-	13.128
0	5	COG_18845	COG_48567	-	-	13.128
0	5	COG_18845	COG_48538	-	AAA ATPase domain	13.128
0	5	COG_18845	COG_48455	-	protein homooligomerization	13.128
0	5	COG_18845	COG_48402	-	-	13.128
0	5	COG_18845	COG_48256	-	-	13.128

Table A.11: Gene pair couplings in cluster (b) , Part 4. This table reports all statistically significant gene–gene couplings that are part of cluster (b) in Figure 3.9. All COGs belong to Bin ID 0 or 5. All interactions have strong positive effect sizes.

Genome	Assembly issues
10N.244.46.E10	extremely large genome size (10,730,107 bp); very high contig count (3264); very low N50 (7,982)
10N.244.45.A07	extremely large genome size (18,716,077 bp); very high contig count (1948); very low N50 (57,224)
10N.247.46.A02	very high contig count (1744)
10N.239.45.G06	very high contig count (1819); very low N50 (2,740)
10N.247.45.G07	very high contig count (907); low N50 (78,584)
10N.239.45.B10	very high contig count (867)
10N.239.45.B11	very high contig count (494)
10N.244.46.B09	very high contig count (471)
10N.247.45.E01	very high contig count (336)
10N.242.46.B09	very high contig count (281)
10N.239.45.H09	very high contig count (267)
10N.247.46.E05	very high contig count (258)
10N.247.45.H08	very high contig count (253)
10N.239.45.B12	very high contig count (237)
10N.242.45.G06	very high contig count (232)

Table A.12: Genome assemblies with extreme values. Unusually large genome size, very high contig counts, or very low N50 values.

Bibliography

- [1] Alan Agresti. *Categorical Data Analysis*. Wiley, Hoboken, NJ, 3 edition, 2013.
- [2] Eric Baptiste, Maureen A. O’Malley, Robert G. Beiko, Marc Ereshefsky, J. Peter Gogarten, Laura Franklin-Hall, François-Joseph Lapointe, John Dupré, Tal Dagan, Yan Boucher, and William Martin. Prokaryotic evolution and the tree of life are two different things. *Biology Direct*, 4:34, 2009. doi: 10.1186/1745-6150-4-34.
- [3] Daniel Barker, Andrew Meade, and Mark Pagel. Constrained models of evolution lead to improved prediction of functional linkage from correlated gain and loss of genes. *Bioinformatics*, 23(1): 14–20, 2007. ISSN 1367-4803. doi: 10.1093/bioinformatics/btl558. URL <https://doi.org/10.1093/bioinformatics/btl558>. Epub 2006 Nov 7.
- [4] Leonard E. Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The Annals of Mathematical Statistics*, 41(1):164–171, 1970. URL <http://www.jstor.org/stable/2239727>.
- [5] Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.
- [6] John Besemer, Alexandre Lomsadze, and Mark Borodovsky. Genemarks: a self-training method for prediction of gene starts in microbial genomes. implications for finding sequence motifs in regulatory regions. *Nucleic Acids Research*, 29(12):2607–2618, 06 2001. ISSN 0305-1048. doi: 10.1093/nar/29.12.2607. URL <https://doi.org/10.1093/nar/29.12.2607>.
- [7] Henry R. Bourne, David A. Sanders, and Frank McCormick. The GTPase superfamily: conserved structure and molecular mechanism. *Nature*, 349(6305):117–127, 1991. doi: 10.1038/349117a0.
- [8] Nina K. Broeker and Stefanie Barbirz. Not a barrier but a key: How bacteriophages exploit host’s O-antigen as an essential receptor to initiate infection. *Molecular Microbiology*, 105(3):353–357, 2017. ISSN 0950-382X. doi: 10.1111/mmi.13729.
- [9] Carlos P. Cantalapiedra, Ana Hernández-Plaza, Ivica Letunic, Peer Bork, and Jaime Huerta-Cepas. eggno-mapper v2: functional annotation, orthology assignments, and domain prediction at the metagenomic scale. *Molecular Biology and Evolution*, 2021. doi: 10.1093/molbev/msab293.
- [10] Ofir Cohen, Haim Ashkenazy, David Burstein, and Tal Pupko. Uncovering the co-evolutionary network among prokaryotic genes. *Bioinformatics*, 28(18):i389–i394, 2012. ISSN 1367-4803. doi: 10.1093/bioinformatics/bts396.

- [11] Ofir Cohen, Haim Ashkenazy, Eli Levy Karin, David Burstein, and Tal Pupko. Copap: Coevolution of presence–absence patterns. *Nucleic Acids Research*, 41(W1):W232–W237, 06 2013. ISSN 0305-1048. doi: 10.1093/nar/gkt471. URL <https://doi.org/10.1093/nar/gkt471>.
- [12] Otto X. Cordero and Martin F. Polz. Explaining microbial genomic diversity in light of evolutionary ecology. *Nature Reviews Microbiology*, 12(4):263–273, 2014. ISSN 1740-1534. doi: 10.1038/nrmicro3218. URL <https://doi.org/10.1038/nrmicro3218>.
- [13] Otto X. Cordero, Berend Snel, and Paulien Hogeweg. Coevolution of gene families in prokaryotes. *Genome Research*, 18(3):462–468, 2008. doi: 10.1101/gr.6815508.
- [14] Miklós Csűrös. *COUNT: Evolutionary Analysis of Phylogenetic Profiles and Other Numerical Characters – User’s Guide*. Department of Computer Science and Operations Research, Université de Montréal, Montréal, Québec, Canada, July 2009. Version dated July 15, 2009.
- [15] Miklós Csűrös. Count: evolutionary analysis of phylogenetic profiles with parsimony and likelihood. *Bioinformatics*, 26(15):1910–1912, 06 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq315. URL <https://doi.org/10.1093/bioinformatics/btq315>.
- [16] David de Juan, Florencio Pazos, and Alfonso Valencia. Emerging methods in protein co-evolution. *Nature Reviews Genetics*, 14(4):249–261, 2013. ISSN 1471-0064. doi: 10.1038/nrg3414. URL <https://doi.org/10.1038/nrg3414>.
- [17] Marc Deloger, Meriem El Karoui, and Marie-Agnès Petit. A genomic distance based on mum indicates discontinuity between most bacterial species and genera. *Journal of Bacteriology*, 191(1):91–99, 2009. doi: 10.1128/jb.01202-08. URL <https://journals.asm.org/doi/abs/10.1128/jb.01202-08>.
- [18] Elena Dembech, Marco Malatesta, Carlo De Rito, Giulia Mori, Davide Cavazzini, Andrea Secchi, Francesco Morandin, and Riccardo Percudani. Identification of hidden associations among eukaryotic genes through statistical analysis of coevolutionary transitions. *Proceedings of the National Academy of Sciences*, 120(16):e2218329120, 2023. doi: 10.1073/pnas.2218329120. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2218329120>.
- [19] Xavier Didelot and Martin C. J. Maiden. Impact of recombination on bacterial evolution. *Trends in Microbiology*, 18(7):315–322, July 2010. ISSN 0966-842X. doi: 10.1016/j.tim.2010.04.002. Epub 2010 May 6.
- [20] Xavier Didelot and Daniel J. Wilson. Clonalframeml: efficient inference of recombination in whole bacterial genomes. *PLoS Computational Biology*, 11(2):e1004041, 2015. doi: 10.1371/journal.pcbi.1004041. URL <https://doi.org/10.1371/journal.pcbi.1004041>.
- [21] Nicholas J Dimonaco, Wayne Aubrey, Kim Kenobi, Amanda Clare, and Christopher J Creevey. No one tool to rule them all: prokaryotic gene prediction tool annotations are highly dependent on the organism of study. *Bioinformatics*, 38(5):1198–1207, 12 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab827. URL <https://doi.org/10.1093/bioinformatics/btab827>.

- [22] Richard Durbin, Sean R. Eddy, Anders Krogh, and Graeme Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, 1998.
- [23] Joseph J. Falke, Randal B. Bass, Scott L. Butler, Stephen A. Chervitz, and Mark A. Danielson. The two-component signaling pathway of bacterial chemotaxis: A molecular view of signal transduction by receptors, kinases, and adaptation enzymes. *Annual Review of Cell and Developmental Biology*, 13:457–512, 1997. doi: 10.1146/annurev.cellbio.13.1.457.
- [24] Paul G. Falkowski, Tom Fenchel, and Edward F. Delong. The microbial engines that drive earth’s biogeochemical cycles. *Science*, 320(5879):1034–1039, 2008. doi: 10.1126/science.1153213. URL <https://www.science.org/doi/abs/10.1126/science.1153213>.
- [25] Joseph Felsenstein. Evolutionary trees from dna sequences: A maximum likelihood approach. *Journal of Molecular Evolution*, 17(6):368–376, November 1981. doi: 10.1007/BF01734359.
- [26] Joseph Felsenstein. *Inferring phylogenies*. Sinauer, Sunderland, Mass., 2004. ISBN 0878931775.
- [27] Laura S. Frost, Raphael Leplae, Anne O. Summers, and Ariane Toussaint. Mobile genetic elements: the agents of open source evolution. *Nature Reviews Microbiology*, 3(9):722–732, 2005. doi: 10.1038/nrmicro1235.
- [28] Michael Y. Galperin. Bacterial signal transduction network in a genomic perspective. *Environmental Microbiology*, 6(6):552–567, June 2004. doi: 10.1111/j.1462-2920.2004.00633.x.
- [29] Michael Y Galperin, David M Kristensen, Kira S Makarova, Yuri I Wolf, and Eugene V Koonin. Microbial genome analysis: the cog approach. *Briefings in Bioinformatics*, 20(4):1063–1070, 09 2017. ISSN 1477-4054. doi: 10.1093/bib/bbx117. URL <https://doi.org/10.1093/bib/bbx117>.
- [30] Andrew Gelman, John B. Carlin, Hal S. Stern, David B. Dunson, Aki Vehtari, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 3 edition, 2013. doi: 10.1201/b16018.
- [31] Phyllis I. Hanson and Sidney W. Whiteheart. AAA+ proteins: have engine, will work. *Nature Reviews Molecular Cell Biology*, 6(7):519–529, July 2005. doi: 10.1038/nrm1684.
- [32] Finbarr Hayes. Toxins-antitoxins: plasmid maintenance, programmed cell death, and cell cycle arrest. *Science*, 301(5639):1496–1499, 2003. ISSN 0036-8075. doi: 10.1126/science.1088157.
- [33] Brian P. Hedlund and James T. Staley. *Vibrio cyclotrophicus* sp. nov., a polycyclic aromatic hydrocarbon (pah)-degrading marine bacterium. *International Journal of Systematic and Evolutionary Microbiology*, 51(1):61–66, January 2001. ISSN 1466-5026. doi: 10.1099/00207713-51-1-61.
- [34] Lisa Hermans, Geert Molenberghs, Geert Verbeke, Michael G. Kenward, Pavlos Mamouris, and Bert Vaes. Optimal weighted estimation versus cochrane–mantel–haenszel. *Communications in Statistics - Simulation and Computation*, 51(7):3645–3659, 2022. doi: 10.1080/03610918.2020.1720735.

- [35] Diep Thi Hoang, Olga Chernomor, Arndt von Haeseler, Bui Quang Minh, and Le Sy Vinh. Ufboot2: Improving the ultrafast bootstrap approximation. *Molecular Biology and Evolution*, 35(2):518–522, 10 2017. ISSN 0737-4038. doi: 10.1093/molbev/msx281. URL <https://doi.org/10.1093/molbev/msx281>.
- [36] John P. Huelsenbeck, Fredrik Ronquist, Rasmus Nielsen, and Jonathan P. Bollback. Bayesian inference of phylogeny and its impact on evolutionary biology. *Science*, 294(5550):2310–2314, 2001. doi: 10.1126/science.1065889. URL <https://www.science.org/doi/abs/10.1126/science.1065889>.
- [37] Jaime Huerta-Cepas, Damian Szklarczyk, Davide Heller, Ana Hernández-Plaza, Sofia K. Forslund, Helen Cook, Daniel R. Mende, Ivica Letunic, Thomas Rattei, Lars J. Jensen, Christian von Mering, and Peer Bork. eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. *Nucleic Acids Research*, 47(D1):D309–D314, 2019. doi: 10.1093/nar/gky1085.
- [38] Doug Hyatt, Gwo-Liang Chen, Philip F. LoCascio, Miriam L. Land, Frank W. Larimer, and Loren J. Hauser. Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, 11(1):119, 2010. ISSN 1471-2105. doi: 10.1186/1471-2105-11-119. URL <https://doi.org/10.1186/1471-2105-11-119>.
- [39] Oliver C. Ibe. 5 - continuous-time markov chains. In Oliver C. Ibe, editor, *Markov Processes for Stochastic Modeling (Second Edition)*, pages 85–102. Elsevier, Oxford, second edition edition, 2013. ISBN 978-0-12-407795-9. doi: <https://doi.org/10.1016/B978-0-12-407795-9.00005-0>. URL <https://www.sciencedirect.com/science/article/pii/B9780124077959000050>.
- [40] Aurora Ilmannafia, Alfina Berlian Yudianti, Febriana Nur Aini, Faiz Ramadhani Ghilman Nugroho, Muhammad Dava Khoirur Roziqy, and Azis Suroi. Evaluation theoretical efficiency of selection sort by big-o notation analysis. *Journal of Advanced Systems Intelligence and Cybersecurity*, 1 (1), August 2025. URL <https://journal.unesa.ac.id/index.php/jasic/article/view/44481>.
- [41] Lakshminarayan M. Iyer, Dapeng Zhang, Igor B. Rogozin, and L. Aravind. Evolution of the deaminase fold and multiple origins of eukaryotic editing and mutagenic nucleic acid deaminases from bacterial toxin systems. *Nucleic Acids Research*, 39(22):9473–9497, 09 2011. ISSN 0305-1048. doi: 10.1093/nar/gkr691. URL <https://doi.org/10.1093/nar/gkr691>.
- [42] Michael J. Sweeting, Alexander J. Sutton, and Paul C. Lambert. What to add to nothing? use and avoidance of continuity corrections in meta-analysis of sparse data. *Statistics in Medicine*, 23 (9):1351–1375, 2004. doi: <https://doi.org/10.1002/sim.1761>. URL <https://onlinelibrary.wiley.com/doi/abs/10.1002/sim.1761>.
- [43] Subha Kalyaanamoorthy, Bui Quang Minh, Thomas K. F. Wong, Arndt von Haeseler, and Lars S. Jermiin. Modelfinder: fast model selection for accurate phylogenetic estimates. *Nature Methods*, 14(6):587–589, 2017. ISSN 1548-7105. doi: 10.1038/nmeth.4285. URL <https://doi.org/10.1038/nmeth.4285>.

- [44] Minoru Kanehisa and Susumu Goto. Kegg: Kyoto encyclopedia of genes and genomes. *Nucleic Acids Research*, 28(1):27–30, 01 2000. ISSN 0305-1048. doi: 10.1093/nar/28.1.27. URL <https://doi.org/10.1093/nar/28.1.27>.
- [45] Kathryn M. Kauffman, Julia M. Brown, Radhey S. Sharma, David VanInsberghe, Joseph Elsherbini, Martin Polz, and Libusha Kelly. Viruses of the nahant collection, characterization of 251 marine vibriaceae viruses. *Scientific Data*, 5(1):180114, 2018. ISSN 2052-4463. doi: 10.1038/sdata.2018.114. URL <https://doi.org/10.1038/sdata.2018.114>.
- [46] Bryce Kille, Michael G Nute, Victor Huang, Eddie Kim, Adam M Phillippy, and Todd J Treangen. Parsnp 2.0: scalable core-genome alignment for massive microbial datasets. *Bioinformatics*, 40(5):btac311, 05 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btac311. URL <https://doi.org/10.1093/bioinformatics/btac311>.
- [47] Eugene V. Koonin. Horizontal gene transfer: essentiality and evolvability in prokaryotes, and roles in evolutionary transitions. *F1000Research*, 5:1805, 2016. ISSN 2046-1402. doi: 10.12688/f1000research.8737.1. URL <https://doi.org/10.12688/f1000research.8737.1>.
- [48] David M. Kristensen, Lavanya Kannan, Michael K. Coleman, Yuri I. Wolf, Alexander Sorokin, Eugene V. Koonin, and Arcady Mushegian. A low-polynomial algorithm for assembling clusters of orthologous groups from intergenomic symmetric best matches. *Bioinformatics*, 26(12):1481–1487, 05 2010. ISSN 1367-4803. doi: 10.1093/bioinformatics/btq229. URL <https://doi.org/10.1093/bioinformatics/btq229>.
- [49] M K Kuhner and J Felsenstein. A simulation comparison of phylogeny algorithms under equal and unequal evolutionary rates. *Molecular Biology and Evolution*, 11(3):459–468, 05 1994. ISSN 0737-4038. doi: 10.1093/oxfordjournals.molbev.a040126. URL <https://doi.org/10.1093/oxfordjournals.molbev.a040126>.
- [50] Simon J. Labrie, Julie E. Samson, and Sylvain Moineau. Bacteriophage resistance mechanisms. *Nature Reviews Microbiology*, 8(5):317–327, 2010. ISSN 1740-1534. doi: 10.1038/nrmicro2315. URL <https://doi.org/10.1038/nrmicro2315>.
- [51] Martin Larralde. Pyrodigal: Python bindings and interface to prodigal, an efficient method for gene prediction in prokaryotes. *Journal of Open Source Software*, 7:4296, 04 2022. doi: 10.21105/joss.04296.
- [52] Alan R. Lemmon, Emily C. Moriarty, and Jack Sullivan. The importance of proper model assumption in bayesian phylogenetics. *Systematic biology*, 53(2):265–277, 2004. ISSN 1063-5157.
- [53] Ivica Letunic and Peer Bork. Interactive tree of life (itol) v6: recent updates to the phylogenetic tree display and annotation tool. *Nucleic Acids Research*, 52(W1):W78–W82, 04 2024. ISSN 0305-1048. doi: 10.1093/nar/gkac268. URL <https://doi.org/10.1093/nar/gkac268>.
- [54] Yang Li, Sarah E. Calvo, Roe Gutman, Jun S. Liu, and Vamsi K. Mootha. Expansion of biological pathways based on evolutionary inference. *Cell*, 158(1):213–225, 2014. ISSN 0092-8674. doi: 10.1016/j.cell.2014.05.034. URL <https://doi.org/10.1016/j.cell.2014.05.034>.

- [55] Heyu Lin, Min Yu, Xiaolei Wang, and Xiao-Hua Zhang. Comparative genomic analysis reveals the evolution and environmental adaptation strategies of vibrios. *BMC Genomics*, 19(1):135, 2018. ISSN 1471-2164. doi: 10.1186/s12864-018-4531-2. URL <https://doi.org/10.1186/s12864-018-4531-2>.
- [56] Robert M. Macnab. How bacteria assemble flagella. *Annual Review of Microbiology*, 57:77–100, 2003. doi: 10.1146/annurev.micro.57.030502.090832. URL <https://doi.org/10.1146/annurev.micro.57.030502.090832>.
- [57] Tanja Magoc, Stephan Pabinger, Stefan Canzar, Xinyue Liu, Qi Su, Daniela Puiu, Luke J. Tallon, and Steven L. Salzberg. Gage-b: an evaluation of genome assemblers for bacterial organisms. *Bioinformatics*, 29(14):1718–1725, May 2013. ISSN 1367-4803. doi: 10.1093/bioinformatics/btt273. URL <https://doi.org/10.1093/bioinformatics/btt273>.
- [58] Firdous Ahmad Mala and Rouf Ali. The big-o of mathematics and computer science. *Journal of Applied Mathematics and Computation*, 6(1):1–3, 2022. doi: 10.26855/jamc.2022.03.001.
- [59] Nathan Mantel and William Haenszel. Statistical aspects of the analysis of data from retrospective studies of disease. *JNCI: Journal of the National Cancer Institute*, 22(4):719–748, 1959. doi: 10.1093/jnci/22.4.719.
- [60] Antonio Martin-Platero, Brian Cleary, Kathryn Kauffman, Sarah Preheim, Dennis McGillicuddy, Eric Alm, and Martin Polz. High resolution time series reveals cohesive but short-lived communities in coastal plankton. *Nature Communications*, 9, January 2018. doi: 10.1038/s41467-017-02571-4. URL <https://doi.org/10.1038/s41467-017-02571-4>.
- [61] James O. McInerney, Alan McNally, and Mary J. O’Connell. Why prokaryotes have pangenomes. *Nature Microbiology*, 2(4):17040, 2017. ISSN 2058-5276. doi: 10.1038/nmicrobiol.2017.40. URL <https://doi.org/10.1038/nmicrobiol.2017.40>.
- [62] Boris G. Mirkin, Trevor I. Fenner, Michael Y. Galperin, and Eugene V. Koonin. Algorithms for computing parsimonious evolutionary scenarios for genome evolution, the last universal common ancestor and dominance of horizontal gene transfer in the evolution of prokaryotes. *BMC Evolutionary Biology*, 3(1):2, 2003. ISSN 1471-2148. doi: 10.1186/1471-2148-3-2.
- [63] Niranjan Nagarajan and Mihai Pop. Sequence assembly demystified. *Nature Reviews Genetics*, 14(3):157–167, 2013. doi: 10.1038/nrg3367. URL <https://doi.org/10.1038/nrg3367>.
- [64] Suganya Nathamuni, Ashok Kumar Jangam, Vinaya Kumar Katneni, Ashok Selvaraj, Karthic Krishnan, Sujeet Kumar, Satheesha Avunje, Sivamani Balasubramaniam, Monendra Grover, Shankar V. Alavandi, and Vijayan K. Koyadan. Insights on genomic diversity of *Vibrio* spp. through pan-genome analysis. *Annals of Microbiology*, 69(13):1547–1555, 2019. doi: 10.1007/s13213-019-01539-7. URL <https://doi.org/10.1007/s13213-019-01539-7>.
- [65] William S. Noble. How does multiple testing correction work? *Nature Biotechnology*, 27(12):1135–1137, 2009. ISSN 1546-1696. doi: 10.1038/nbt1209-1135. URL <https://doi.org/10.1038/nbt1209-1135>.

- [66] Howard Ochman, Jeffrey G. Lawrence, and Eduardo A. Groisman. Lateral gene transfer and the nature of bacterial innovation. *Nature*, 405(6784):299–304, 2000. doi: 10.1038/35012500.
- [67] Deo Prakash Pandey and Kenn Gerdes. Toxin-antitoxin loci are highly abundant in free-living but lost from host-associated prokaryotes. *Nucleic Acids Research*, 33(3):966–976, 2005. doi: 10.1093/nar/gki201.
- [68] Donovan H. Parks, Maria Chuvpochina, David W. Waite, Christian Rinke, Adam Skarshewski, Pierre-Alain Chaumeil, and Philip Hugenholtz. A standardized bacterial taxonomy based on genome phylogeny substantially revises the tree of life. *Nature Biotechnology*, 36(10):996–1004, October 2018. ISSN 1546-1696. doi: 10.1038/nbt.4229. URL <https://doi.org/10.1038/nbt.4229>.
- [69] Matteo Pellegrini, Edward M. Marcotte, Michael J. Thompson, David Eisenberg, and Todd O. Yeates. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. *Proceedings of the National Academy of Sciences of the United States of America*, 96(8):4285–4288, 1999. doi: 10.1073/pnas.96.8.4285. URL <https://doi.org/10.1073/pnas.96.8.4285>.
- [70] Shaul Pollak, Matti Gralka, Yuya Sato, Julia Schwartzman, Lu Lu, and Otto X. Cordero. Public good exploitation in natural bacterioplankton communities. *Science Advances*, 7(31):eabi4717, 2021. doi: 10.1126/sciadv.abi4717. URL <https://www.science.org/doi/abs/10.1126/sciadv.abi4717>.
- [71] Michael O. Press, Hongyao Li, Nicole Creanza, Gary Kramer, Christine Queitsch, Victor Sourjik, et al. Genome-scale co-evolutionary inference identifies functions and clients of bacterial hsp90. *PLoS Genetics*, 9(7):e1003631, 2013. doi: 10.1371/journal.pgen.1003631. URL <https://doi.org/10.1371/journal.pgen.1003631>.
- [72] Morgan N. Price, Katherine H. Huang, Adam P. Arkin, and Eric J. Alm. Operon formation is driven by co-regulation and not by horizontal gene transfer. *Genome Research*, 15(6):809–819, June 2005. ISSN 1088-9051. doi: 10.1101/gr.3368805.
- [73] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2025. URL <https://www.R-project.org/>.
- [74] Christian R. H. Raetz and Chris Whitfield. Lipopolysaccharide endotoxins. *Annual Review of Biochemistry*, 71:635–700, 2002. doi: 10.1146/annurev.biochem.71.110601.135414. URL <https://doi.org/10.1146/annurev.biochem.71.110601.135414>.
- [75] D. V. Rakhuba, E. I. Kolomiets, E. S. Dey, and G. I. Novik. Bacteriophage receptors, mechanisms of phage adsorption and penetration into host cell. *Polish Journal of Microbiology*, 59(3):145–155, 2010. ISSN 1733-1331. PMID: 21033576.
- [76] D.F. Robinson and L.R. Foulds. Comparison of phylogenetic trees. *Mathematical Biosciences*, 53(1):131–147, 1981. ISSN 0025-5564. doi: [https://doi.org/10.1016/0025-5564\(81\)90043-2](https://doi.org/10.1016/0025-5564(81)90043-2). URL <https://www.sciencedirect.com/science/article/pii/0025556481900432>.
- [77] Eduardo P. C. Rocha and Antoine Danchin. Base composition bias might result from competition for metabolic resources. *Trends in Genetics*, 18(6):291–294, June 2002. ISSN 0168-9525. doi: 10.1016/S0168-9525(02)02690-2.

- [78] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Research*, 13(11):2498–2504, November 2003.
- [79] B. Jesse Shapiro, Jonathan Friedman, Otto X. Cordero, Sarah P. Preheim, Sonia C. Timberlake, Gitta Szabó, Martin F. Polz, and Eric J. Alm. Population genomics of early events in the ecological differentiation of bacteria. *Science*, 336(6077):48–51, 2012. doi: 10.1126/science.1218198. URL <https://www.science.org/doi/abs/10.1126/science.1218198>.
- [80] Kazuyuki Shimizu. Regulation systems of bacteria such as *Escherichia coli* in response to nutrient limitation and environmental stresses. *Metabolites*, 4(1):1–35, 2013. doi: 10.3390/metabo4010001.
- [81] Alexandros Stamatakis. Raxml version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. *Bioinformatics*, 30(9):1312–1313, 01 2014. ISSN 1367-4803. doi: 10.1093/bioinformatics/btu033. URL <https://doi.org/10.1093/bioinformatics/btu033>.
- [82] Ann M. Stock, Victoria L. Robinson, and Peter N. Goudreau. Two-component signal transduction. *Annual Review of Biochemistry*, 69:183–215, 2000. doi: 10.1146/annurev.biochem.69.1.183.
- [83] Alison F. Takemura, Diana M. Chien, and Martin F. Polz. Associations and dynamics of *Vibrionaceae* in the environment, from the genus to the population level. *Frontiers in Microbiology*, 5:38, 2014. doi: 10.3389/fmicb.2014.00038. URL <https://doi.org/10.3389/fmicb.2014.00038>.
- [84] Hervé Tettelin, Vega Massignani, Michael J. Cieslewicz, Claudio Donati, Duccio Medini, Naomi L. Ward, Samuel V. Angiuoli, Jonathan Crabtree, Amanda L. Jones, A. Scott Durkin, Robert T. DeBoy, Tanja M. Davidsen, Marirosa Mora, Maria Scarselli, Immaculada Margarit y Ros, Jeremy D. Peterson, Christopher R. Hauser, Jaideep P. Sundaram, William C. Nelson, Ramana Madupu, Lauren M. Brinkac, Robert J. Dodson, Mary J. Rosovitz, Steven A. Sullivan, Sean C. Daugherty, Daniel H. Haft, Jeremy Selengut, Michelle L. Gwinn, Liwei Zhou, Nikhat Zafar, Hoda Khouri, Diana Radune, George Dimitrov, Kisha Watkins, Kevin J. B. O’Connor, Shannon Smith, Teresa R. Utterback, Owen White, Craig E. Rubens, Guido Grandi, Lawrence C. Madoff, Dennis L. Kasper, John L. Telford, Michael R. Wessels, Rino Rappuoli, and Claire M. Fraser. Genome analysis of multiple pathogenic isolates of *Streptococcus agalactiae*: Implications for the microbial pan-genome. *Proceedings of the National Academy of Sciences*, 102(39):13950–13955, 2005. doi: 10.1073/pnas.0506758102. URL <https://www.pnas.org/doi/abs/10.1073/pnas.0506758102>.
- [85] Fabiano L. Thompson, Tetsuya Iida, and Jean Swings. Biodiversity of vibrios. *Microbiology and Molecular Biology Reviews*, 68(3):403–431, 2004. doi: 10.1128/mmbr.68.3.403-431.2004. URL <https://journals.asm.org/doi/abs/10.1128/mmbr.68.3.403-431.2004>.
- [86] Natalie Verstraeten, Maarten Fauvart, Wim Versées, and Jan Michiels. The universally conserved prokaryotic gtpases. *Microbiology and Molecular Biology Reviews*, 75(3):507–542, September 2011. ISSN 1092-2172. doi: 10.1128/MMBR.00009-11.

- [87] Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967. doi: 10.1109/TIT.1967.1054010.
- [88] Arndt von Haeseler, Heiko A. Schmidt, Minh Q. Bui, and Lam Tung Nguyen. Iq-tree: A fast and effective stochastic algorithm for estimating maximum-likelihood phylogenies. *Molecular Biology and Evolution*, 32:268–274, November 2014. ISSN 0737-4038. doi: 10.1093/molbev/msu300. URL <http://eprints.cs.univie.ac.at/4509/>.
- [89] George H. Wadams and Judith P. Armitage. Making sense of it all: bacterial chemotaxis. *Nature Reviews Molecular Cell Biology*, 5(12):1024–1037, 2004. ISSN 1471-0080. doi: 10.1038/nrm1524. URL <https://doi.org/10.1038/nrm1524>.
- [90] Thomas Wong, Trong Nhan Ly, Huaiyan Ren, Hector Baños, Andrew Roger, Edward Susko, Chris Bielow, Nicola De Maio, Nick Goldman, Matthew Hahn, Gavin Huttley, Rob Lanfear, and Bui Minh. Iq-tree 3: Phylogenomic inference software using complex evolutionary models. 04 2025. doi: 10.32942/X2P62N.
- [91] Xiaoqian Annie Yu, Cameron R. Strachan, Craig W. Herbold, Michaela Lang, Christoph Gasche, Athanasios Makristathis, Nicola Segata, Shaul Pollak, Adrian Tett, and Martin F. Polz. Genome-wide sweeps create fundamental ecological units in the human gut microbiome. *bioRxiv*, may 2024. doi: 10.1101/2024.05.25.595854. URL <https://doi.org/10.1101/2024.05.25.595854>. Preprint.